A Bioengineering Approach to Quantifying the Progression of Amyotrophic Lateral Sclerosis by Electromyography using Novel Wireless Instrumentation and Data Processing Methods

Aidan Paul Wickham

Department of Bioengineering

Imperial College London

A thesis submitted for the degree of

Philosophiæ Doctor (PhD)

June 2019

Imperial College London

Abstract

This thesis presents the research undertaken towards developing new instrumentation used for capturing fasciculation potentials (FPs) by electromyography (EMG), so that their waveform morphology may be used as a clinical biomarker for tracking the progression of Amyotrophic Lateral Sclerosis (ALS).

Evidence in literature has shown that the fasciculation potential, caused by the instability of a diseased motor neuron's membrane potential, will increase in complexity and variability as the disease progresses.

A 'data-processing pipeline' has been developed in order to process data from multiple-channel surface EMG recordings taken from ALS patients. The pipeline conditions the signals, detects fasciculations, and groups them by the motor unit they originated from (based on the similarity of their morphology) so that variability between waveforms can be parameterised. The data-processing pipeline uses a combination of novel algorithms and frameworks, and machine learning methods (i.e. clustering) to detect and group the FP waveforms. The data-processing pipeline has been developed into a stand-alone application.

A multiple-channel wireless device for capturing EMG signals at high resolution and sampling rate has been developed. The device captures only the FP from the EMG signal and saves it locally to an SD card. The device captures fasciculation potentials across eight channels at 16-bit resolution with a sampling rate of approximately 4 kHz per channel.

A new route for prototyping transdermal, microneedle devices has been developed using a shrinkable resin technology. The reason for using transdermal electrodes for EMG recordings is to reduce skin-electrode contact impedance by painlessly penetrating through the highly resistive first layer of skin. The method has been used to produce microneedles approximately 300 μm in height. The microneedle arrays produced have been made conductive and functionalised into a biosensor for measuring lactate.

Declaration of Originality

The work in this thesis was carried out at the Department of Bioengineering at Imperial College London under the supervision of Professor Martyn G. Boutelle and Professor Emmanuel M. Drakakis. I confirm that the work presented in this thesis is my own. Where information has been derived from other sources, I confirm that this has been indicated in this thesis.

AIDAN PAUL WICKHAM

Declaration of Copyright

The copyright of this thesis rests with the author. Unless otherwise indicated, its contents are licensed under a Creative Commons Attribution-Non Commercial 4.0 International Licence (CC BY-NC).

Under this licence, you may copy and redistribute the material in any medium or format. You may also create and distribute modified versions of the work. This is on the condition that: you credit the author and do not use it, or any derivative works, for a commercial purpose.

When reusing or sharing this work, ensure you make the licence terms clear to others by naming the licence and linking to the licence text. Where a work has been adapted, you should indicate that the work has been changed and describe those changes.

Please seek permission from the copyright holder for uses of this work that are not included in this licence or permitted under UK Copyright Law.

Publications

The publications arising from this thesis are:

- A. P. Wickham, M. A. Booth, S. A. N. Gowers, C. L. Leong, M. L. Rogers, I. C. Samper and M. G. Boutelle. Chemical monitoring in clinical settings: recent developments toward real-time chemical monitoring of patients, Analytical Chemistry, 2017 90(1) 2-18. DOI: 10.1021/acs.analchem.7b04224

- J. Bashford, A. Wickham, R. Iniesta, E. Drakakis, M. Boutelle, K. Mills and C. Shaw. SPiQE: An automated analytical tool for detecting and characterising fasciculations in amyotrophic lateral sclerosis, Clinical Neurophysiology, 2019 130(7) 1083-1090. DOI: 10.1016/j.clinph.2019.03.032

- J. Bashford, A. Wickham, R. Iniesta, E. Drakakis, M. Boutelle, K. Mills and C. Shaw. The surface potential quantification engine integrates active voluntary identification to enhance fasciculation analysis in amyotrophic lateral sclerosis. Muscle & Nerve. *Submitted (2019)*

Acknowledgements

I would like to thank the following for their help and guidance with the work presented in this PhD thesis:

My supervisors, Martyn Boutelle and Manos Drakakis, for their advice, support, and guidance throughout this PhD.

Dr. James Bashford, Prof. Chris Shaw, and Prof. Kerry Mills for the invaluable collaborative effort presented in this thesis. Your clinical expertise of ALS and collection of patient data has been instrumental in getting the data processing work presented in this thesis to its current point.

All the patients at King's College Hospital who have volunteered their time to take part in the longitudinal study and collection of EMG data.

Sally, Michelle, Isabelle, Mars, Agnes, and Rob of the Boutelle lab group for their patience, advice, and inspiration in the lab, and of course for all the laughter.

Georgios and Drakakis lab group members for their invaluable help and advice when I was first starting out with circuit board design.

My family for their inspiration, constant intrigue into my work, and their never-ending support. Without you all I would never have reached this point. Thank you.

Brook, you may be one of the few people to ever read this thesis from cover to cover! Your dedication to helping me succeed will never ever be forgotten. Thank you.

To my father,

Thank you for the inspiration to study Engineering

To my mother,

Thank you for the support that has enabled me to do so

"What is the definition of an idiot? A 12-year old boy who can't do algebra"

- Anthony Wickham

Contents

LIST OF FIGURES **19**

LIST OF TABLES **25**

LIST OF ABBREVIATIONS **27**

CHAPTER 1 INTRODUCTION **31**

1.1 Amyotrophic Lateral Sclerosis **31**

1.2 Pathophysiology **32**

1.2.1 The motor unit 32

1.2.2 Motor unit disease and death 32

1.3 Clinical Assessment of ALS **34**

1.4 Fasciculation Potentials **35**

1.5 Electromyography **37**

1.6 Thesis Objectives **40**

1.7 Thesis Outline **41**

CHAPTER 2 MATERIALS AND METHODS **43**

2.1 Data Processing **43**

2.2 Data Collection **44**

2.2.1 Patient cohort and longitudinal study 44

2.2.2 High-density surface EMG 44

2.2.3 Patient preparation and recording setup 45

2.2.4 Sensor recycling 47

2.3 Laboratory Methods **47**

2.3.1 PDMS casting 47

2.3.2 Polyester casting resin 48

2.3.3 3D model design and printing 48

2.3.4 Microneedle array fabrication 48

2.4 Electronic device design and testing **48**

2.4.1 Device fabrication 48

2.4.2 Device testing 49

2.5 Hardware Programming 49
2.5.1 Arduino programming and bootloader 49
2.5.2 'SDFat' library 51

CHAPTER 3 DATA PROCESSING PIPELINE PART 1 53
3.1 Data capture 54
3.2 Signal conditioning and channel inspection 55
3.2.1 Signal conditioning 56
3.2.2 50 Hz noise removal 57
3.2.2.1 Linear regression subtraction 59
3.2.2.2 Frequency removal and inverse 62
3.2.2.3 Wavelet filtering 63
3.2.2.4 Empirical mode decomposition 64
3.2.2.5 Notch filter 69
3.2.3 Channel inspection 71
3.3 Fasciculation Potential Detection 76
3.3.1 Probabilistic spike detection 78
3.3.1.1 Refining 82
3.3.1.2 Probabilistic spike detection performance 84
3.3.2 Channel reduction 93
3.3.3 Voluntary motor unit activity 97
3.4 Conclusion of Part 1 99

CHAPTER 4 DATA PROCESSING PIPELINE PART 2 103
4.1 Grouping Fasciculation Potential Morphologies 103
4.1.1 Clustering 104
4.1.1.1 Feature extraction 108
4.1.1.2 Polarity 111
4.1.1.3 Principle Component Analysis 112
4.1.1.4 'Cluster and seed' framework 116
4.1.2 Matched filtering 122
4.1.2.1 Grouping FPs using matched filtering 125
4.1.2.2 Shuffling framework 131
4.1.3 Comparison of clustering and matched filtering methods 135
4.1.3.1 Problems with clustering and matched filtering 135
4.1.3.2 Combining clustering and matched filtering 137

4.1.3.3 Rejected FPs ... 139
4.1.4 Alignment and template creation ... 139
4.1.5 Template grouping ... 142
4.1.5.1 Manual grouping ... 145
4.2 Pipeline Output ... 147
4.2.1 Parameterisation of waveform ... 148
4.3 Conclusion of Part 2 ... 151

CHAPTER 5 SURFACE POTENTIAL QUANTIFICATION ENGINE ... 155
5.1 Channel Selection ... 156
5.2 Load Data and Condition Signals ... 157
5.3 Fasciculation Potential Detection ... 158
5.4 Clustering ... 159
5.5 Matched Filtering and Alignment ... 160
5.6 Template Grouping ... 161
5.7 Conclusion ... 163

CHAPTER 6 WIRELESS MULTIPLE CHANNEL DEVICE ... 165
6.1 Device Design – Version 1 ... 165
6.1.1 Requirements ... 165
6.1.2 Device overview ... 166
6.1.3 Analogue front-end ... 167
6.1.3.1 Buffer ... 167
6.1.3.2 Pre-amplification ... 168
6.1.3.3 Combined analogue front-end - LMP7702 ... 169
6.1.4 Analogue-to-digital converter selection – ADS1298 ... 172
6.1.4.1 Programmable-gain amplifier ... 174
6.1.4.2 Reference voltage ... 176
6.1.4.3 Data format ... 178
6.1.4.4 ADS1298 configuration ... 178
6.1.4.5 Data retrieval ... 181
6.1.5 Microcontroller ... 182
6.1.5.1 ATMEGA328p ... 183
6.1.5.2 Arduino bootloader and clock rate ... 184
6.1.6 Data transmission and storage ... 185
6.1.6.1 Wireless protocol – Bluetooth ... 185

6.1.6.2 Removable storage card ... 187
6.1.7 Power management ... 188
6.1.7.1 Battery charger ... 188
6.1.7.2 Voltage regulators ... 189
6.1.8 Data rate and data management ... 191
6.1.8.1 Device resolution ... 191
6.1.8.2 Data rate ... 191
6.1.9 Printed circuit board layout – Device version 1 ... 192
6.1.10 Wireless device performance ... 195
6.1.10.1 Bluetooth streaming tests ... 195
6.1.10.2 SD card problems ... 199
6.2 Device Design – Version 2 ... 202
6.2.1 Microcontroller – ATMEGA2560 ... 203
6.2.1.1 ATMEGA2560 ... 203
6.2.2 Printed circuit board layout – Device Version 2 ... 204
6.2.3 Device version 2 characterisation ... 206
6.2.3.1 Buffer writing speed ... 206
6.2.3.2 Circular buffer array ... 207
6.2.3.3 SD card writing tests ... 209
6.2.3.4 Bluetooth streaming and SD card testing ... 213
6.2.3.5 EMG signal test ... 214
6.3 Arduino Programming ... 216
6.3.1 Data handling ... 216
6.3.2 Device set up ... 218
6.3.2.1 SPI transfer ... 218
6.3.2.2 Set up protocol ... 218
6.3.3 Data capture loop ... 218
6.4 Conclusion ... 221

CHAPTER 7 TRANSDERMAL ELECTRODES ... 223
7.1 Skin-Electrode Impedance ... 224
7.1.1 Surface EMG electrode model ... 225
7.1.2 Transdermal/microneedle array electrodes ... 229
7.2 Current Methods of Microneedle Array Fabrication ... 230
7.3 Iterative Casting-Shrinkage Cycles ... 231
7.3.1 Method ... 232

7.4 Results and Discussion 238
7.4.1 Microneedle characterisation 238
7.4.2 Skin penetration tests 243
7.5 Gold Coating 249
7.6 Lactate Sensor Fabrication 253
7.6.1 Platinum coating 253
7.6.1.1 Platinum black nanoparticle electrodeposition 254
7.6.2 Lactate biosensor fabrication 256
7.6.2.1 Biocompatibility 258
7.7 Individually Addressable Microneedle Arrays 259
7.8 Conclusion 265

CHAPTER 8 CONCLUSIONS AND FUTURE WORK 269
8.1 Overview 269
8.2 Future Work 272

BIBLIOGRAPHY 275

APPENDIX A 281

APPENDIX B 285

APPENDIX C 288

PERMISSIONS 292

CONTENTS

List of Figures

Figure 1-1: The motor unit....32
Figure 1-2: Reinnervation of denervated muscle fibres. 33
Figure 1-3: Morphology comparison between fasciculation potentials from benign fasciculation syndrome (BFS) patients and ALS patients....37
Figure 1-4: Change in MUAP morphology across muscle volume. 39
Figure 1-5: High-density surface EMG (HD-sEMG) arrays.....40
Figure 2-1: High-density surface EMG grid electrodes used in the longitudinal study.....45
Figure 2-2: HD-sEMG electrolyte gel and double sided adhesive backing application. 46
Figure 2-3: Using Arduino hardware as an In-System-Programmer (ISP).....50
Figure 3-1: Overview of the data-processing pipeline.....53
Figure 3-2: Placement of 64-channel surface EMG grid electrodes. 55
Figure 3-3: Signal-conditioning flowchart. 56
Figure 3-4: Effect of high-pass filter frequency on the surface EMG signal.....57
Figure 3-5: Frequency spectrum of band-passed surface EMG recording.....58
Figure 3-6: Unwanted ringing effect caused by 50 Hz notch filter.....59
Figure 3-7: Regression-subtraction method performed on noisy EMG recording.....60
Figure 3-8: Regression-subtraction applied across multiple signal bins. 61
Figure 3-9: Effect of regression-subtraction applied across multiple time-bins.....62
Figure 3-10: Wavelet filtering used to remove noise from EMG signal.....64
Figure 3-11: First and second iteration of the empirical mode decomposition noise removal method.....65
Figure 3-12: Formation of Intrinsic Mode Function using EMD noise-removal method.....66
Figure 3-13: Frequency power spectrum of each IMF found in the EMG signal. 67
Figure 3-14: 50 Hz noise accounted by several IMFs using EMD noise-removal method. ..67
Figure 3-15: Frequency power spectrum and time series of the reconstructed EMG signal formed using EMD noise-removal method.....68
Figure 3-16: Original EMG signal affected by 50 Hz noise and the resulting notch filtered signal.....70
Figure 3-17: EMG signals with low 50 Hz noise levels are less distorted if left unfiltered..71
Figure 3-18: HD-sEMG grid electrode by TMSi. 72
Figure 3-19: Three channels from same recording session displaying very different 'activity' levels (band-pass filtered 20-500 Hz without 50 Hz notch filter). 73
Figure 3-20: Bar chart demonstrating excluded channels with high levels of 'activity'. 75
Figure 3-21: High level of activity caused in a non-perimeter channel likely due to insufficient application of electrolytic gel. 76
Figure 3-22: Amplitude threshold can exclude legitimate FPs. 77
Figure 3-23: Spike detection flow chart. 77
Figure 3-24: Histogram and probability function created from the values within an EMG recording.....79
Figure 3-25: Probability scores for signal epochs. 80
Figure 3-26: Conjoining adjacent epochs that have passed the probability threshold to form the fasciculation potential.....81

Figure 3-27: Steps taken using the 'probabilistic' spike detection method. 81
Figure 3-28: Refining stage used to conjoin separated epochs. 83
Figure 3-29: Manually detected FPs from two 60-second periods of EMG data recorded from two different patients, combined into the same time-series. 85
Figure 3-30: Receiver Operator Characteristic (ROC) curve comparing the False Positive Rate (FPR) and True Positive Rate (TPR) of the spike detection algorithm when compared to labelled data for a range of probability thresholds. 87
Figure 3-31: ROC curve with refined epochs. 88
Figure 3-32: ROC curves for different amplitude thresholds. 90
Figure 3-33: ROC curve for the spike detection algorithm for a range of probability thresholds. 92
Figure 3-34: FPs detected and rejected using probabilistic spike detection algorithm. 93
Figure 3-35: FP morphology changes depending on position on recording electrode. 94
Figure 3-36: A fasciculation potential captured by HD-sEMG at four different time-points (FP1, FP2, FP3, FP4). 95
Figure 3-37: Example of forming the 'super-channel'. 96
Figure 3-38: FP occurrence map. 97
Figure 3-39: Voluntary motor unit activity that could be mistaken for FPs. 98
Figure 4-1: Grouping FP morphologies flowchart. 104
Figure 4-2: Example of clustered data. 104
Figure 4-3: Fuzzy c-means clustering example. 106
Figure 4-4: Hard K-means clustering example. 108
Figure 4-5: Scatter plot of amplitude and area. 109
Figure 4-6: Extracted features from FPs. 110
Figure 4-7: Splitting FP polarity before clustering. 111
Figure 4-8: Clustering flowchart. 112
Figure 4-9: ALS patient data used to demonstrate principle component analysis. 113
Figure 4-10: PCA of features extracted from FPs. 114
Figure 4-11: Correlation between different features. 115
Figure 4-12: K-means clustering of principle components from dataset. 116
Figure 4-13: 'Cluster and seed' framework for the first four iterations. 118
Figure 4-14: Final iteration of 'cluster and seed' framework. 121
Figure 4-15: Formation of matched filter. 123
Figure 4-16: Matching criteria of impulse response. 124
Figure 4-17: Matched filtering method used to find a waveform of similar morphology to the target template in the unknown signal. 125
Figure 4-18: Checking u and r thresholds of a database of FPs. 127
Figure 4-19: Flowchart for grouping FPs using matched filtering. 129
Figure 4-20: Cross-compare FP IDs and combine groups. 130
Figure 4-21: Flowchart for grouping FPs using matched filtering and shuffling framework. 132
Figure 4-22: Grouping matched FPs using shuffling framework. 134
Figure 4-23: Number of FPs in temporary database during shuffling rounds. 135
Figure 4-24: Errors caused by clustering due to similarity of FP features. 136
Figure 4-25: Combining clustering and matched filtering. 137

Figure 4-26: Number of FPs in the temporary database after each FP removal stage of the matched filtering shuffling framework, for one shuffling round. 138
Figure 4-27: Alignment of matching FPs. 140
Figure 4-28: Alignment of grouped FP waveforms. 141
Figure 4-29: Further grouping of FPs using averaged templates and matched filtering... 143
Figure 4-30: Matched filtering is used to further group the averaged FP group templates. 144
Figure 4-31: Manual grouping of templates. 146
Figure 4-32: Group of FPs outputted by data processing pipeline. 147
Figure 4-33: Grouped FPs aligned by matched filtering and by start point. 149
Figure 4-34: Grouped FPs aligned by start point showing time variations. 150
Figure 4-35: Template creation and firing rate calculation of individual motor unit. 151
Figure 4-36: Full-data processing pipeline summary flowchart. 154
Figure 5-1: START panel of Surface Potential Quantification Engine (SPiQE). 156
Figure 5-2: Channel select panel of SPiQE. 157
Figure 5-3: Load data and condition signals panel for SPiQE. 158
Figure 5-4: Spike detection panel for SPiQE. 159
Figure 5-5: Clustering panel for SPiQE. 160
Figure 5-6: Matched filtering and alignment panel for SPiQE. 161
Figure 5-7: Template grouping panel for SPiQE. 162
Figure 6-1: Device design overview. 166
Figure 6-2: Op-amp configuration for a unity-gain buffer/voltage follower. 167
Figure 6-3: The op-amp configuration for a non-inverting amplifier. 169
Figure 6-4: Analogue Front-End (AFE) of each sensor input. 169
Figure 6-5: Dual precision amplifier LMP7702 used for the AFE of the device. 170
Figure 6-6: Diagram of amplification of a large FP. 171
Figure 6-7: Output of signals amplified by the AFE. 172
Figure 6-8: Key analogue input and digital input/output pins of the ADS1298. 173
Figure 6-9: Programmable gain amplifier of ADS1298. 174
Figure 6-10: Instrumentation amplifier op-amp configuration of ADS1298. 175
Figure 6-11: Complete AFE op-amp configuration of device. 176
Figure 6-12: Effect of ADS1298 reference voltage. 177
Figure 6-13: Write register (WREG) command example. 179
Figure 6-14: ADS1298 start-up configuration and procedure. 181
Figure 6-15: SPI data output for retrieving data. 182
Figure 6-16: Communication ports (SPI, USART, I²C) and interrupt pins of the ATMEGA328p. 183
Figure 6-17: Key pin connections of ATMEGA328p to peripheral devices. 184
Figure 6-18: Maximum clock frequency vs. operating voltage of ATMEGA328p. 185
Figure 6-19: SPI connection between microcontroller and peripheral devices. 188
Figure 6-20: Schematic of the device power supply and battery charger. 189
Figure 6-21: TPS7A7001 schematic for producing +3.3 V and +2.5 V. 190
Figure 6-22: PCB layout of 8-channel wireless device. 194
Figure 6-23: Assembled device version 1. 195
Figure 6-24: Wireless device Bluetooth test compared to PowerLab. 196

Figure 6-25: Device Bluetooth test of 20 μV_{p-p} and 2 μV_{p-p} sine waves compared to PowerLab. 197
Figure 6-26: Loss of data as a result of low Bluetooth sampling rate. 198
Figure 6-27: Writing delay for one 512-byte block. 200
Figure 6-28: Plan for storing and writing multiple 512-byte blocks to an SD card. 201
Figure 6-29: Writing delay for two 512-byte blocks. 202
Figure 6-30: ATMEGA2560 pin connections to peripheral devices. 203
Figure 6-31: PCB layout of 8-channel wireless device version 2. 205
Figure 6-32: Assembled device version 2 206
Figure 6-33: Demonstration of a circular-buffer array storing the phrase 'HELLOWORLD' in an 8-element array. 208
Figure 6-34: Schematic of circular-buffer array and buffer blocks stored in SRAM. 209
Figure 6-35: SD card writing test with 100 Hz and 500 Hz sine waves. 210
Figure 6-36: Histogram of FP length 212
Figure 6-37: Read and write time of an FP to the SD card. 213
Figure 6-38: Pre-recorded signal and FPs detected by wireless device. 214
Figure 6-39: FPs captured by device from pre-recorded EMG recording. 215
Figure 6-40: FP captured by device overlaid on input signal. 216
Figure 6-41: Data is stored in the SRAM using a series of nested structures. 217
Figure 6-42: Flowchart of data capture loop. 220
Figure 7-1: Diagram of skin layers. 224
Figure 7-2: Electrical equivalent circuit of electrode-electrolyte contact. 226
Figure 7-3: Full electrical equivalent model of wet sEMG skin-electrode contact electrode. 228
Figure 7-4: Microneedle array penetrating through the *stratum corneum*. 229
Figure 7-5: Casting-shrinkage method used to create microfluidic chips 232
Figure 7-6: 3D printed master used for initial experiments into producing MNAs using the casting-shrinkage method 233
Figure 7-7: Shrunken PDMS replicas distorted by air bubbles. 234
Figure 7-8: Custom vacuum chamber for degassing PU-water mixture and mould. 235
Figure 7-9: PU cast before and after shrinking process 236
Figure 7-10: Casting-shrinkage cycle schematic 237
Figure 7-11: PDMS casts are taken of the shrunken casts. 238
Figure 7-12: The shrunken microneedles after one cycle, measured using a laser profilometer. 239
Figure 7-13: First, second, and third cycle casts of 2x2 array of 4x4 conical microneedles. 240
Figure 7-14: SEM image of third cycle cast of conical microneedles. 241
Figure 7-15: SEM images of different microneedle geometry and MNA layouts. 242
Figure 7-16: Skin biopsy cross-section method 244
Figure 7-17: 20 µm skin slice taken from human cadaver skin biopsy penetrated by microneedles measuring approximately 572 µm in height (5X magnification). 245
Figure 7-18: 20 µm skin slice taken from human cadaver skin biopsy penetrated by microneedles measuring approximately 572 µm in height (10X magnification). 245
Figure 7-19: Schematic of puncture depth variation between skin biopsy cross-sections. 246

Figure 7-20: 20 μm skin slice taken from human cadaver skin biopsy penetrated by microneedles measuring approximately 572 μm in height showing broken s*tratum corneum* (5X magnification). 246
Figure 7-21: 20 μm skin slices taken from human cadaver skin biopsy - control test MNA penetration test from the third cycle of the cast-shrinking process (microneedles measuring approximately 300 μm in height). 248
Figure 7-22: First cycle cast-shrinkage MNAs coated with chromium and gold. 249
Figure 7-23: Cyclic Voltammetry (CV) setup and results of Au-coated MNA. 250
Figure 7-24: Insulating the base of the Au-coated MNA. 251
Figure 7-25: CVs of Au vs. Ag|AgCl in 1.5 mM ferrocene monocarboxylic acid for the insulated and uninsulated Au-coated MNAs. 252
Figure 7-26: Hydrogen peroxide calibration on 50 μm Pt and Au electrodes. 254
Figure 7-27: Platinum black nanoparticle deposition onto gold surface. 255
Figure 7-28: Exposed gold of MNAs coated with Pt black nanoparticles. 256
Figure 7-29: Lactate biosensor MNA layering. 257
Figure 7-30: Current response for lactate biosensor MNA. 258
Figure 7-31: Model of 3D printed master used for the individually addressable MNA and the resulting PU cast and PDMS mould from the second cycle of the cast-shrinkage process. 260
Figure 7-32: SEM images of the first and second-cycle casts of the individually addressable MNAs. 260
Figure 7-33: Steps 1-4 of coating individually addressable microneedles. 261
Figure 7-34: Fabricating Au coated individually addressable microneedles. 261
Figure 7-35: Au coated individually addressable microneedles. 262
Figure 7-36: CVs of Au vs. Ag|AgCl in 1.5 mM ferrocene monocarboxylic acid of an individually addressable microneedle. 263
Figure 7-37: Conical electrode, inlaid vs. stilted. 265

LIST OF FIGURES

List of Tables

Table 2-1: Equipment used for testing and characterising assembled devices 49
Table 2-2: SPI pins for Arduino Uno 50
Table 3-1: Confusion matrix of positive and negative data-points. 86
Table 3-2: Number of manually detected FPs (labelled dataset) that pass through the amplitude threshold (ampth) at different threshold values. 91
Table 4-1: Results of using matched filtering to match five FPs of matching morphology. 128
Table 6-1: ADC resolution of data for internal reference voltages 2.4 V and 4 V. 177
Table 6-2: Output code of converted data at 24-bit resolution (two's complement)..... 178
Table 6-3: Configuration data to be written to ADS1298 on power-up. 180
Table 6-4: Comparison of wireless protocols ZigBee, Bluetooth (classic), and Wi-Fi. 186
Table 6-5: Resistor values required for producing +3.3 V and +2.5V. 190
Table 6-6: Signal-to-noise ratio (SNR) values of signals recorded by wireless device and PowerLab. 197
Table 6-7: Comparison of quasi-continuous sampling for device versions 1 and 2..... 207
Table 6-8: Effect of ADS1298 data rate on device SD card sampling rate. 211
Table 7-1: Qualitative assessment of various fabrication methods used for producing microneedle arrays. 230
Table 7-2: Measurements of microneedle height and base width after 1st and 2nd cycles of the cast-shrinking process. 240
Table 7-3: Puncture depths of microneedles from the 2nd and 3rd cycles of the casting-shrinkage method. 247
Table 7-4: Approximate calculated surface area of exposed Au on MNA. 253
Table 7-5: Rise time of the first five steps in the calibration of the Lactate biosensor MNA. 258

List of Abbreviations

ALS	Amyotrophic lateral sclerosis
ADC	Analogue to digital converter
AFE	Analogue front end
Ag	Silver
Ag|AgCl	Silver-silver chloride
ALS-FRS	ALS- function rating scale
Ampth	Amplitude threshold
Au	Gold
AUC	Area under curve
BFS	Benign fasciculation syndrome
BSA	Bovine serum albumin
CE	Counter electrode
Closeth	Closeness threshold
Cr	Chromium
CS	Chip select
CSV	Comma separated value
CV	Cyclic voltammetry
DAQ	Data acquisition
DC	Direct current
DIN	Data in
DOUT	Data out
DRDY	Data ready
DRIE	Deep reactive ion etching
ECG	Electrocardiogram
EEG	Electroencephalogram
EMD	Empirical mode decomposition
EMG	Electromyography
Fc	Ferrocene
Fibs	Fibrillations
FPGA	Field programmable gate array
FPR	False positive rate
FPs	Fasciculation potentials
HD-sEMG	High-density surface electromyography
I/O	In/out
I^2C	Two-wire serial interface
IC	Integrated circuit
IDE	Integrated development environment
IMF	Intrinsic mode function
IPA	Isopropyl alcohol

ISI	Inter-spike-interval
ISP	In system programmer
KCL	King's College London
LOx	Lactate oxidase
LSB	Least significant bit
MEKP	Methyl ethyl ketone peroxide
MISO	Master in slave out
MNA	Microneedle array
MND	Motor neuron disease
MOSI	Master out slave in
mPD	M-phenylenediamine
MRC	Medical Research Council
MSB	Most significant bit
MU	Motor unit
MUAP	Motor unit action potential
MUNE	Motor unit number estimation
MUNIX	Motor unit number index
nEMG	Needle electromyography
OCT	Optimal cutting temperature
Op-amp	Operational amplifier
Opcode	Operation code
PBS	Phosphate buffered saline
PC	Principle component
PCA	Principle component analysis
PCB	Printed circuit board
PDMS	Polydimethylsiloxane
PEG-DE	Poly(eythylene glycol) diglycidyl ether
PGA	Programmable gain amplifier
Probth	Probability threshold
PSW	Positive sharp wave
Pt	Platinum
PU	Polyurethane
RC	Resistor-capacitor
RDATA	Read data
RDATAC	Read data continuous
RE	Reference electrode
RMS	Root mean square
ROC	Receiver operator characteristic
RREG	Read register
SCLK	Serial clock
SD	Secure digital
SDATAC	Stop data continuous

SEM	Scanning electron microscope
sEMG	Surface electromyography
SNR	Signal to noise ratio
SPI	Serial peripheral interface
SPiQE	Surface potential quantification engine
SPS	Samples per second
SRAM	Static read access memory
TMSi	Twente Medical Systems International
TNR	True negative rate
TPR	True positive rate
UI	User-interface
USART	Universal synchronous/asynchronous receiver/transmitter
USB	Universal serial bus
V_{p-p}	Peak to peak voltage
WE	Working electrode
WREG	Write register
ZCP	Zero crossing point

Chapter 1
Introduction

1.1 Amyotrophic Lateral Sclerosis

Amyotrophic Lateral Sclerosis (ALS), or Motor Neuron Disease (MND) is a neurodegenerative disorder. As the disease progresses, the patient will suffer from muscle atrophy and a loss of motor function as it attacks motor neurons in the brain and in the spinal cord. As the disease progresses further, the patient will approach complete paralysis and as the muscles associated with breathing become affected, the patient will require permanent breathing apparatus. ALS eventually results in death, typically occurring 3 years from symptom onset;[1] the cause of death is most commonly due to respiratory failure or pneumonia, caused by the weakening of respiratory muscles.[2]

At the time of writing, Riluzole is currently the only licensed drug available for ALS in Europe, which only shows a small survival benefit, prolonging life by a median of 2-3 months.[3,4] There is an urgency to discover novel biomarkers that can be monitored throughout the course of the disease, which could be used in clinical trials to speed up the discovery of new drugs and therapeutics to slow down or halt the progression of ALS.[5]

1.2 Pathophysiology

1.2.1 The motor unit

The motor unit (MU) consists of the motor neuron's cell body (soma), its axon, and all the muscle fibres that its terminal branches innervate. The muscle fibres associated to a motor unit are typically dispersed throughout the muscle, rather than MU's muscle fibres being bundled closely together. The number of muscle fibres per MU will depend on the size and function of the muscle. For example, the MU of muscle requiring dexterity and specificity may only innervate 10 muscle fibres, whereas the MU of a muscle requiring low dexterity and large amounts of power (e.g. a leg muscle) can innervate over 1000.[6] The depolarisation of the motor neuron causes the muscle fibres of the motor unit to contract; the electrical waveform of this is known as the motor unit action potential (MUAP).

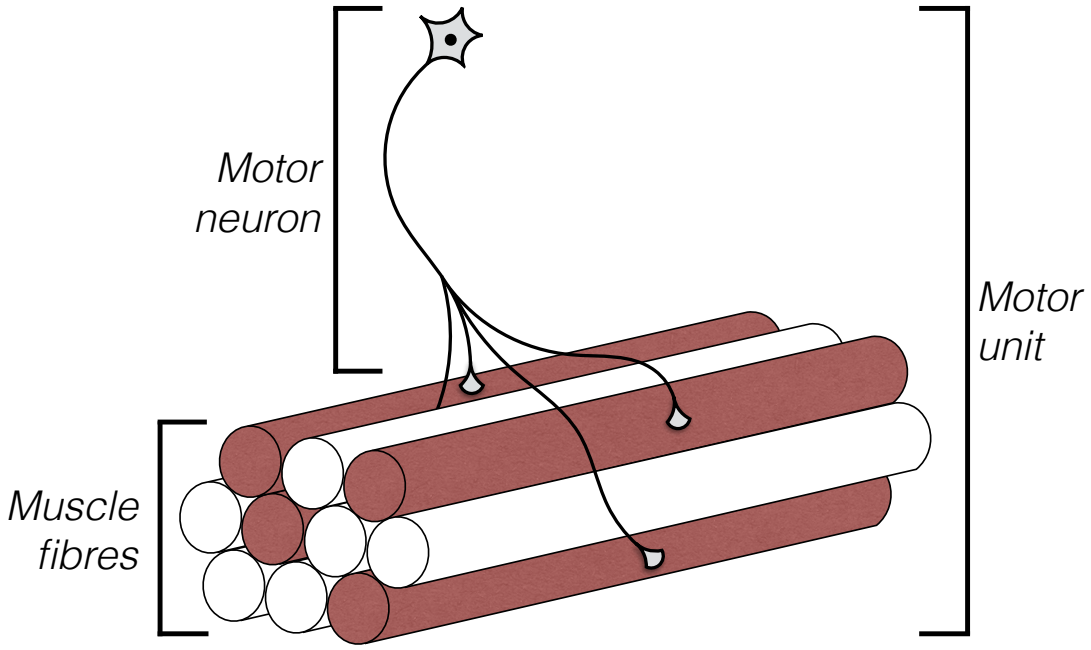


Figure 1-1: The motor unit. *The motor unit comprises of a motor neuron and the muscle fibres it innervates. The discharge of the motor neuron will cause the innervated muscle fibres to contract.*

1.2.2 Motor unit disease and death

As the disease progresses, and ailed motor neurons eventually die, the associated muscle fibres in the MU will become denervated. Remaining motor units will branch into the denervated muscle fibres, reinnervating them, in order to

preserve muscle power. As the process of the motor neuron degeneration is gradual in ALS, the reinnervation of muscle fibres by adjacent motor neurons is also likely to be a gradual process. As a result, the MUAPs of MUs that are acquiring denervated muscle fibres will gradually increase in amplitude, duration, and complexity.[7,8] At this point, the patient will begin to lose dexterity and specificity of motor action due to less motor units innervating more muscle fibres. Due to the nature of the disease process, eventually the reinnervating MU's motor neuron will also become diseased, resulting in mass denervation and a loss of muscle power in the patient.

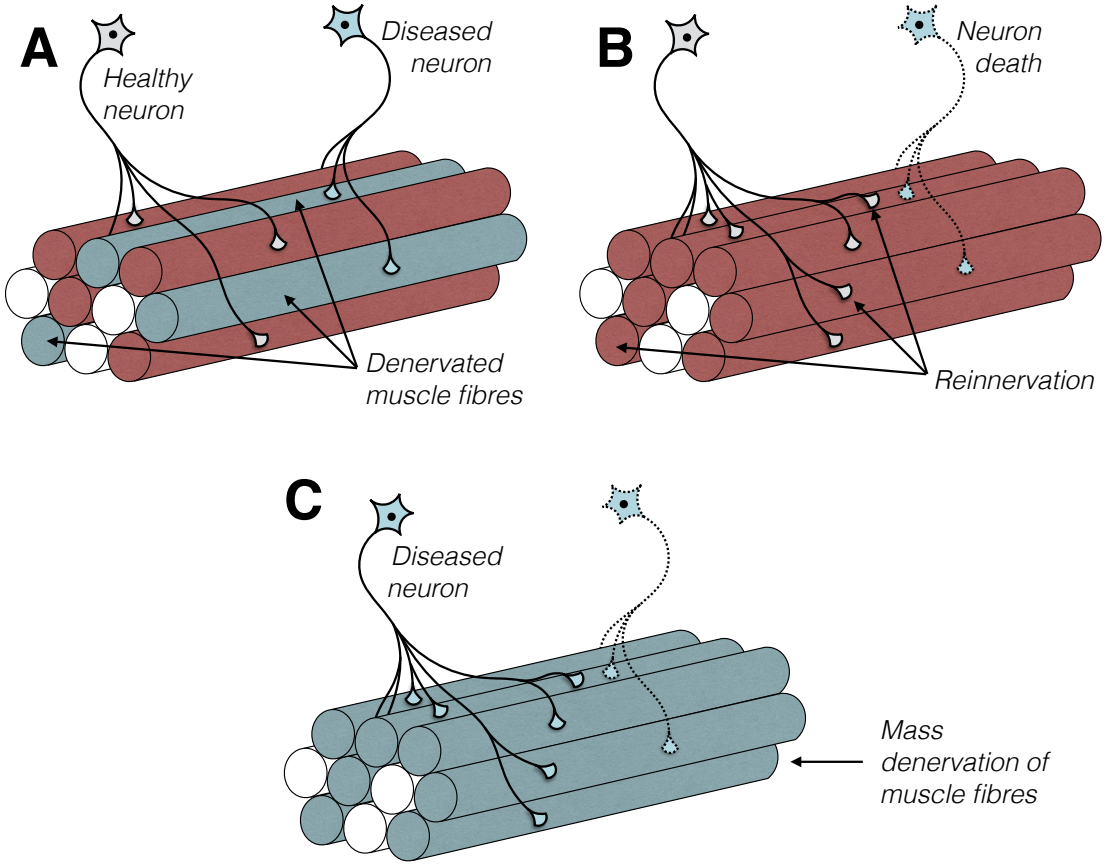


Figure 1-2: Reinnervation of denervated muscle fibres. *(A) Diseased motor neuron results in denervation of muscle fibres associated to that motor unit. (B) Healthy motor neuron reinnervates denervated muscle fibres to preserve muscle power. (C) When the reinnervating motor neuron also eventually becomes diseased, the muscle fibres are denervated, resulting in a loss of muscle power.*

1.3 Clinical Assessment of ALS

A common way of measuring the progression of ALS is by using a range of different functional rating scales and muscle power tests. These ratings can provide the clinician and patient with an indication of the current stage of the disease. The ALS function rating scale (ALS-FRS) assesses the patient's ability to carry out specific every-day tasks and bodily functions, such as handwriting or swallowing.[9] For example, for the assessment of '*handwriting*', the patient will be scored a 4 if '*normal*' handwriting is demonstrated, and a 0 if the patient is '*unable to grip pen'*.[9] The Norris scale also assesses patient functionality, but on a more discrete scale from 0 to 3.[10] The score given to each test in the functional rating scales are totalled together, to give the patient an overall score. The Medical Research Council (MRC) scale measures patient muscle power on a scale from 0-5; 5 indicating the muscle has a normal level of strength and 0 indicating the muscle has lost all power. Muscle power is assessed by the clinician pushing/pulling on different limbs and asking the patient to resist against it, testing different muscle groups around the body. The power scores for each muscle group are totalled together to give an overall MRC score for the patient.

These scales help to provide an indication of how the disease has progressed, however as they are based on the opinion of the assessing clinician there is a possibility they could be subject to bias. Also, using these discrete scales means that valuable continuous information of the patient's condition can be missed; for example, a patient can rapidly decline in their scaled assessment with comparatively little drop in functionality or muscle power.

Respiratory function is also used as an assessment of disease progression by measuring the patient's vital lung capacity; this is a continuous measurement, and taken by the patient blowing into a device, which records the measurements as a percentage. However it is also debated how reliable this measure is; as the patient's facial muscles deteriorate, they will struggle to purse their lips around the tube used for blowing into the device, which may produce an unreliable result for respiratory function.

There are a number of pathophysiological biomarkers that can be used to assess disease progression: motor unit number estimation (MUNE) and motor unit number index (MUNIX) attempt to estimate the number of motor neurons innervating a muscle or group,[11,12] transcranial magnetic stimulation can be used to measure cortical excitability,[13] and excitability testing of peripheral nerves.[14] The difficulty with using these pathophysiological biomarkers is that they are complex to perform and require time and experience to learn properly,[5] limiting their use as a measurement of disease prognosis in clinical trials.[15]

For these reasons, the development of potential drugs and therapeutics is hampered due to the difficulty of precisely measuring the progression of the disease. There is need for a robust, quantifiable, continuous measure of ALS prognosis that does not require large amounts of experience or specialised training to perform, and new instrumentation that can facilitate such a measure.

1.4 Fasciculation Potentials

As the motor neuron becomes diseased, it will struggle to maintain its membrane potential, resulting in spontaneous discharges known as 'fasciculations', which cause the muscle fibres belonging to the neuron's MU to contract.[16] These MU contractions or fasciculations can present themselves as visible muscle twitches in MU's close to the surface of the skin, however fasciculations in deeper MUs must be detected via electromyogram (see Section 1.5). The electrical waveform of a fasciculation is known as a 'fasciculation potential' (FP). Another neurophysiological indicator of ALS are Fibrillations (Fibs), which occur from individual denervated muscle fibres; Fibs cannot be seen visually, but can be detected using electromyography (EMG).[17]

FPs are considered to be the clinical and neurophysiological hallmark of ALS.[7,18–22] FPs can be used as an early marker for ALS, compared to Fibs which occur in the later stages of the disease[23] (due to the denervation of muscle fibres from dead motor neurons). The presence of fasciculations will precede loss of muscle power,[24] before the neuron has started to denervate the muscle fibres of its MU, and therefore will precede any changes that may be recorded using the rating scales and techniques described in Section 1.3. Consequently, FPs are

considered an *'early marker of ALS'*[23] and their presence is emphasised in the Awaji electrodiagnostic criteria for the diagnosis of ALS.[25,26]

Although FPs are a well recognised feature of ALS, their presence is not enough to confirm diagnosis of ALS alone.[25] For example, patients with benign fasciculation syndrome (BFS) will express FPs with a stable, simple waveform morphology in likeness to a MUAP (see Figure 1-3). FPs in ALS patients, however, are recognised for their increased jitter, complex morphology, longer duration, slower firing frequency, and conduction block.[23,25,27–31]

As the disease progresses and the reinnervation of muscle fibres from denervated MUs starts to take place from branching motor neurons (see Figure 1-2), the instability of FPs and the complexity of their waveforms will increase.[23,28] Investigations by De Carvalho & Swash have found the morphology of the FPs to have increased in amplitude, complexity, duration, and polyphasicity in patients with a lower MRC score (and thus have progressed further with the disease), as can be seen in Figure 1-3.[23] These findings match with the explanation of reinnervating motor units described in Figure 1-2. This suggests that not only can FPs be used as an early marker for ALS, but the changes observed in their morphology and the instability of their waveform can be used as a method of tracking disease progression.

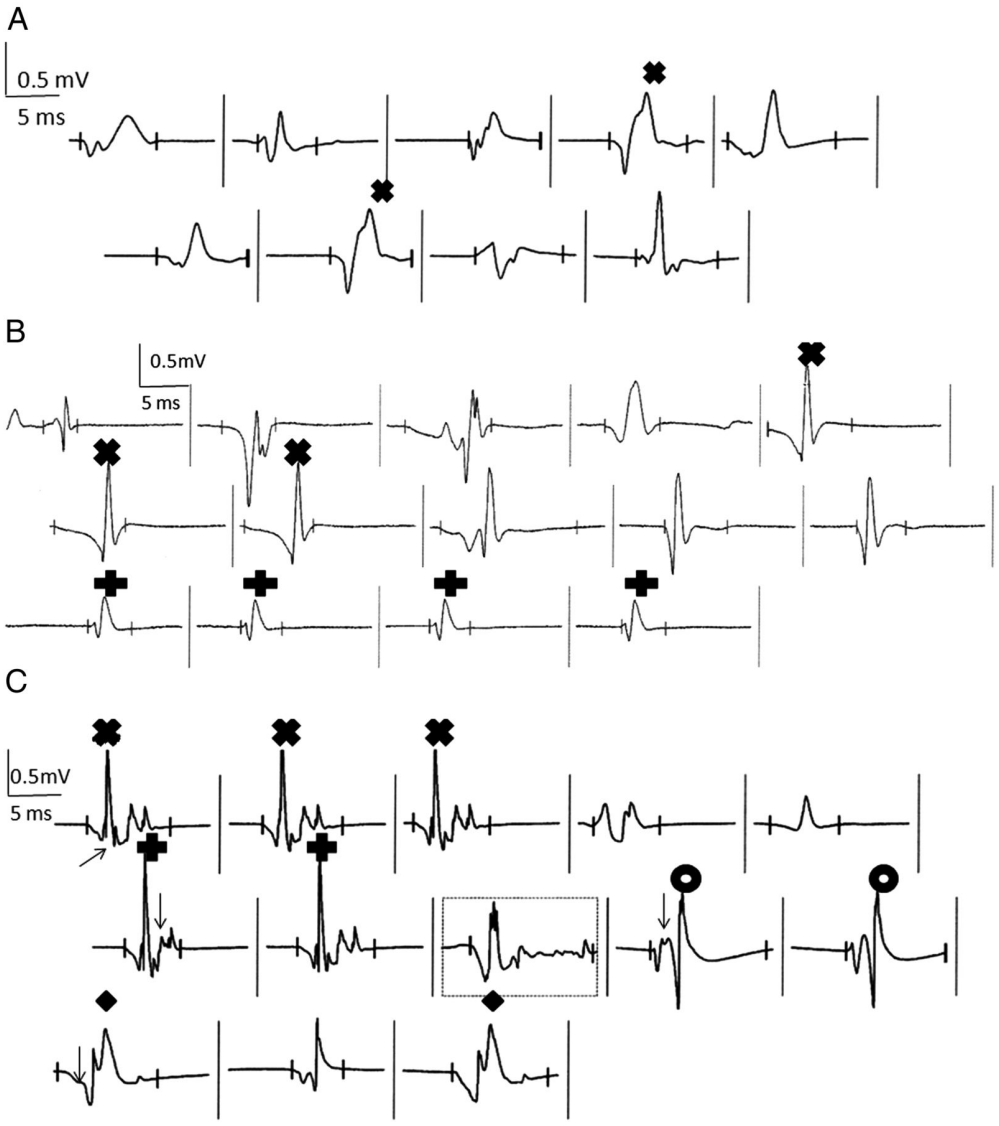


Figure 1-3: Morphology comparison between fasciculation potentials from benign fasciculation syndrome (BFS) patients and ALS patients. *Manually extracted FPs from BFS and ALS patients, detected using needle EMG. (A) FPs from a patient with BFS, showing simple morphology similar to a MUAP. (B) FPs from ALS patient with MRC score of 5, showing an increased amplitude and complexity compared to FPs from a BFS patient. (C) FPs from ALS patient with MRC score of 4, showing further morphology complexity and increased polyphasicity. (Matching FPs are marked with a symbol, and unstable components are marked with an arrow). (Image taken from de Carvalho et al. 2013).*[23]

1.5 Electromyography

Electromyography (EMG) is the recording of electrical activity from muscle tissue. It can be performed using penetrative needle electrodes or by using non-penetrative surface electrodes.

Investigations into neuromuscular diseases and disorders such as ALS often use needle EMG (nEMG) electrodes.[6–8,16,18,19,23,25,26,32–37] The benefit of using nEMG is that the electrode tip can be inserted directly into the MU that is fasciculating, meaning a high quality recording of the FP waveform, that is unlikely to be affected by noise. The disadvantage of using nEMG is that the electrode must penetrate through the skin and into the muscle of the patient, which is likely to cause the patient pain and discomfort; the nEMG electrode may need to be taken out and reinserted multiple times until the electromyographer has the electrode in the location they wish to investigate. As the needle is inserted it will cause damage to the muscle, resulting in unwanted discharges.[6] Any movement from the muscle or the needle during the recording can cause further damage to the muscle, resulting in further discharges that will appear as noise and artefacts on the recorded signal. Recording sessions are likely to be long in duration, and therefore it is likely the patient will move during the recording period.

The alternative to nEMG is to use surface EMG (sEMG) electrodes, which do not penetrate into the muscle but are positioned on the surface of the patients skin. This is the great advantage of sEMG compared to nEMG; the application of the surface electrode is painless and can be repositioned if necessary without causing the patient any additional discomfort. SEMG electrodes are also easier to apply than nEMG electrodes, lowering the skill level required by the investigator. The disadvantage of sEMG is that is highly susceptible to noise;[38] the top layer of skin is highly resistive, meaning the skin-electrode contact can have a very large impedance resulting in a very noisy recording. This impedance can be reduced by preparing the patient's skin by shaving, cleaning, and abrading away dead skin cells.[39] However, this is not always successful, often causing noise on the sEMG recording. Despite this, sEMG has an important value as patients are more likely to take part in studies and retain their involvement if there are to be repeat recording sessions, if the application of electrodes is painless.

Penetrative nEMG does have the advantage that the needle electrode can be used to target individual MUs by inserting the needle directly into the MU of interest, whereas recording from the skin surface will detect muscle activity

from the entire muscle volume. The positioning of the sEMG electrode relative to the fasciculating MU's innervation zone/motor endplate region will also affect the polarity and amplitude of the FP that is recorded,[21,22,40] as shown in Figure 1-4.

Figure 1-4: Change in MUAP morphology across muscle volume. *The location of the sEMG electrode relative to the innervation zone of the motor unit will affect the morphology of the MUAP. MUAP recorded by a 10x10 array of sEMG electrodes (bipolar derivation, inter-electrode distance = 5 mm). The innervation zone is located at approximately the 3rd row from the bottom, 4th column from the left. (Image taken from Stegeman et al. 2012).*[21]

High-density surface EMG (HD-sEMG) grid arrays are becoming an increasingly popular tool for recording EMG from both healthy patients, but also in particular from investigations into FPs from patients with ALS.[21,22,40–46] Using HD-sEMG electrode arrays, like those shown in Figure 1-5, ensures there is an electrode ideally positioned for recording FPs from each fasciculating MU in the muscle of interest. The use of HD-sEMG grid arrays also improves the likelihood that FPs will be detected as multiple MUs can be recorded from at the same time, compared to using a single electrode.[33,46] Additional spatial information gained from HD-sEMG can provide insight into physiological characteristics such as motor unit territory, conduction velocity, and innervation zone localisation, that individual nEMG and sEMG electrodes cannot.[46]

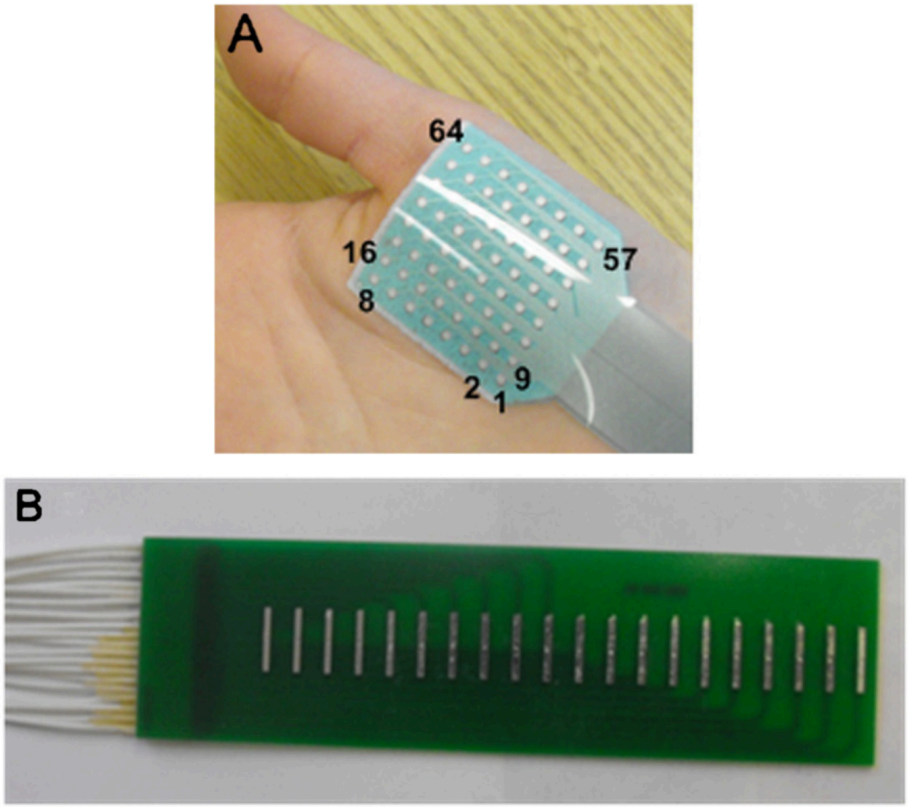


Figure 1-5: High-density surface EMG (HD-sEMG) arrays. *(A) Flexible HD-sEMG grid array consisting of 64 channels (8x8 matrix), manufactured by TMSi, Netherlands. (B) HD-sEMG linear array consisting of 20-channels, manufactured by Zhou et al.*[46] *(Images reproduced from Zhou 2012).*[46]

1.6 Thesis Objectives

This thesis focuses on improving the instrumentation for the detection of FPs as a way of monitoring the progression of ALS, by way of HD-sEMG grid arrays. FPs are a biomarker for ALS, and their morphology has been shown to increase in instability and complexity as the disease progresses. New instrumentation is required in order to efficiently use FP morphology as a measure of disease prognosis.

EMG recording sessions might need to be very long in duration in order to ensure enough FPs have been captured to form a measure of their waveform variability, and thus a measure of disease progression. Manually detecting FPs from high-density channel recordings will be an extremely arduous task (especially if the recordings are very long in duration). For this reason, automatic

processing is required to detect FPs from the HD-sEMG recordings. The number of FPs detected from a recording session could potentially be very large, and so automatic processing will also be required to group the FPs by the MU that they have originated from.

If recording sessions are to take place regularly, then a new recording system will be required that will maximise comfort for the patient. This system should make monitoring from the patients possible without the need of a clinician present to set it up, and should be compact enough to be a step towards a wearable device.

As previously mentioned, sEMG is very susceptible to noise. This can cause problems with the detection of FPs and the assessment of their morphology. Despite this, surface electrodes are preferable to penetrative needle EMG due to their painless application. Investigations into how to fabricate an electrode that maximises the benefits of both sEMG and nEMG are made in this thesis.

1.7 Thesis Outline

Chapter 2 describes the common materials and methods used in this thesis, as well as the software packages used for data analysis and electronic design and programming. This chapter also describes the recording setup and patient preparation used in the EMG studies of patients with ALS.

Chapter 3 describes the first half of the data processing pipeline: signal conditioning and fasciculation potential detection. This chapter describes the steps taken to condition the signal so that FPs can be detected with confidence. Due to surface EMG's high susceptibility to noise, several methods for the removal of mains line noise (50 Hz) have been investigated. HD-sEMG grid electrodes can often have poor connections and so low-quality channels are automatically checked for using a novel measure termed 'channel activity'. FPs are detected using a novel algorithm termed the 'probabilistic spike detection' method. Rather than selecting an arbitrary detection threshold value, this method provides a rational basis for selecting a threshold for FPs, which is a variant to factors such as grid, electrode, skin, gel, patient, and operator.

Chapter 4 describes the second half of the data processing pipeline: algorithms and frameworks developed for grouping FPs by their morphology, and thus by the motor unit they have originated from. Using an iterative framework, termed 'cluster-and-seed', the principle components of FPs are grouped together using K-means clustering. The morphology of the clustered FPs are then checked using an algorithm called 'matched filtering'; the FP clusters are split into smaller groups or are rejected from the analysis. The FP groups produced by the data-processing pipeline can be examined by clinicians and investigators, who can extract an abundance of information related to the variability of the waveform morphology.

Chapter 5 describes the stand-alone application that has been developed using the algorithms and frameworks developed as part of the data-processing pipeline described in Chapters 3 and 4. The application is named 'Surface Potential Quantification Engine' (SPiQE), after the collaboration between the data-processing pipeline I have developed and the collection of HD-sEMG data by clinicians at King's College London.

Chapter 6 describes the development of a compact, multiple-channel wireless device. The developed device captures and stores just the FP from the incoming EMG signal, not capturing the rest of the 'empty signal' (EMG signal not containing FPs). The device stores FPs at high resolution and high sampling rate to a local SD card.

Chapter 7 describes the development of a new prototyping route for producing transdermal devices/microneedle arrays using a shrinking resin technology. A range of different geometries and microneedle profiles has been experimented with using this method. The microneedle arrays have undergone penetration testing using human cadaver skin-biopsies. The microneedles have been made conductive and functionalised into a biosensor for measuring lactate.

Chapter 8 concludes the thesis and describes possible future work.

Chapter 2
Materials and Methods

In this chapter, I present the common materials, methods, software, and hardware that are frequently used throughout this thesis.

2.1 Data Processing

All data processing and analysis in this thesis has been performed using *MATLAB* (MATLAB R2014B, MathWorks). MATLAB is a scripting programming language allowing for the manipulation, analysis, and presentation of data. MATLAB script/code is written within the MATLAB development environment, which gives users access to a host of libraries containing many common and useful functions.

MATLAB also provides the means to develop a user-interface (UI) for code developed in the MATLAB environment. The MATLAB Compiler allows for UIs to be exported as stand-alone applications that can be shared and used by parties without the need to purchase or have installed MATLAB; users will have to install the free to use 'MATLAB Runtime' to use the stand-alone applications.

2.2 Data Collection

2.2.1 Patient cohort and longitudinal study

This PhD thesis has partly been a collaboration with clinical investigators Prof Chris Shaw and Dr James Bashford at King's College London (KCL). The team of clinicians at KCL have used their expertise and access to patients to form a cohort of Amyotrophic Lateral Sclerosis (ALS) patient volunteers to collect EMG data from. Patient recordings took place at monthly intervals (or as close to a month as possible, depending on the health/mobility of the patient) as part of a 'longitudinal study'. Repeated recording sessions as part of a longitudinal study mean that any changes in FP occurrence/morphology can be tracked, and thus the patient's prognosis can be measured. The collected data has received ethical approval from North of Scotland Research Ethics Service (ref. 15/NS/0103, approved October 2015) and by Nottingham 1 Research Ethics Service (ref. 17/EM/0221, approved June 2017).

2.2.2 High-density surface EMG

All EMG recordings from patients presented in this thesis were recorded using high-density surface EMG (HD-sEMG) grid electrodes, developed by TMSi (Twente Medical Systems International B.V). The HD-sEMG grid electrodes are arranged in an 8x8 grid (64 individually addressable EMG channels) and have been used previously in literature to detect FPs.[46] The grid electrodes are made of a Kapton film, the same material used to make flexible circuit boards, which allows the grid to bend and form around the curvature of the patient's body (see Figure 2-1). The grid electrode is available in a 4 mm and 8.5 mm inter-electrode distance. The 8.5 mm grid electrodes were chosen as to cover the large muscle bulk of the muscles used for recording (*biceps brachii* and *gastrocnemius medial*). A specialised, shielded cable connects the 64 electrode channels to the TMSi data acquisition system, the 'Refa', for amplifying and sampling the incoming EMG signals. The Refa does not use any electronic hardware filters (e.g. band-pass, notch filter etc.) so the full frequency range of the originally sampled data is available for the investigator to analyse. The Refa has monopolar inputs, but is configured as an average reference amplifier; all input channels are averaged

and subtracted from each individual input. The Refa requires it to be connected to a neutral site on the patient (i.e. a bony prominence) to act as a reference point for the amplifier and ADC. The Refa will record data in a 'poly5' data format, which can be converted into a '.mat' file for processing in MATLAB.

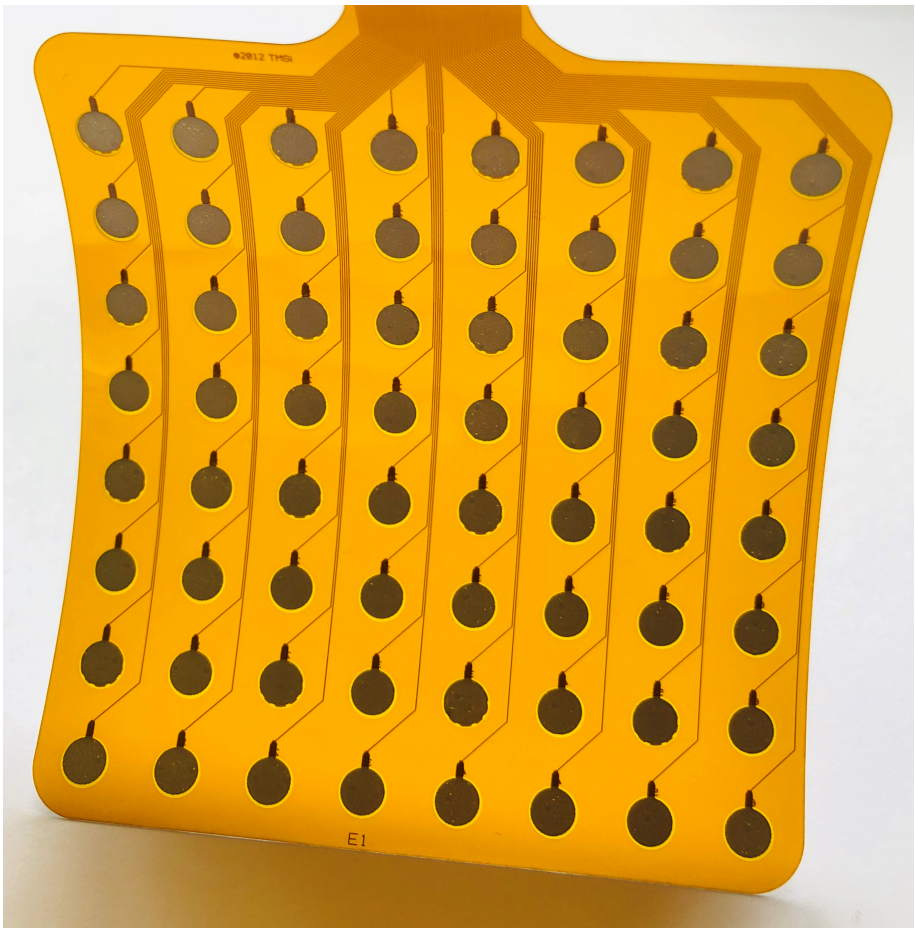


Figure 2-1: High-density surface EMG grid electrodes used in the longitudinal study. *The grid electrodes offer an 8x8 array of surface EMG electrodes, mounted on a Kapton film that makes it flexible. Shown is the 8.5mm inter-electrode distance version of the grid electrode that is used in the longitudinal study. The HD-sEMG grid array is manufactured by TMSi, Netherlands.*

2.2.3 Patient preparation and recording setup

Patients were asked to lie down and relax, avoiding any muscle contractions in their arms or legs. The patient's skin must be properly prepared before the sensor grid is applied in order to reduce the impedance of the surface electrode-skin contact. The skin is cleaned using alcohol wipes and, if necessary, shaved. An abrasive, exfoliating cream is used on the skin at the site where the HD-sEMG electrode grid is to be applied to try and remove the highly resistive *stratum corneum* (top layer of skin consisting of dead skin cells and oils). A double-sided

adhesive sticker (see Figure 2-2(B)) is attached to the HD-sEMG grid electrode to fix it to the patient's skin for the duration of the recording session. Before fixing the grid electrode to the patient, a coating of conductive, electrolytic gel is applied across the grid (see Figure 2-2(C)). The sticker's second backing is removed, and the gel-coated electrode grid is fixed to the patient's prepared skin (see Figure 2-2(D)).

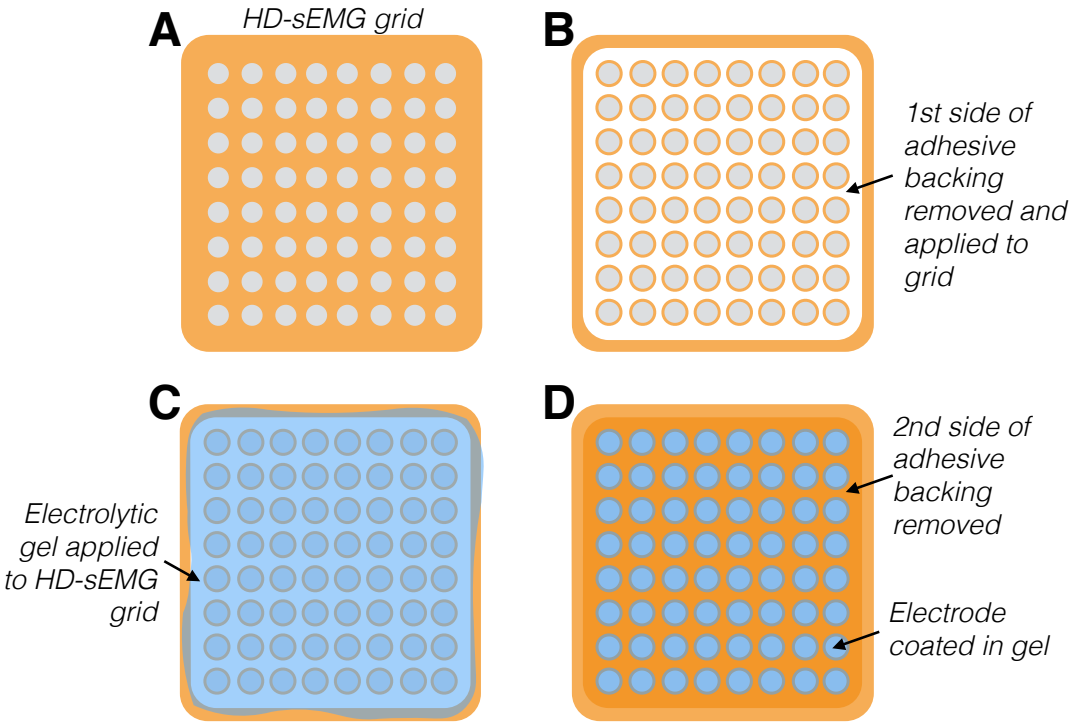


Figure 2-2: HD-sEMG electrolyte gel and double sided adhesive backing application. *(A) Collect HD-sEMG electrode and make sure patient's skin has been prepared for recording. (B) Remove first side of double-sided adhesive backing and apply to the grid. (C) Coat electrode in electrolyte gel. (D) Remove second side of backing, leaving electrodes covered in gel and HD-sEMG grid ready to be adhered to the patient's skin.*

Recordings are taken from the *biceps brachii* and the *gastrocnemius medial* muscles, bilaterally. An additional electrode is placed over the elbow bone and anklebone to provide an electrical reference for the Refa amplifier. Measurements from the corners of the grid electrode with reference to the location of physical landmarks (e.g. bony prominences) are made and photographs are taken in order to ensure that the grid electrode is positioned as close to the same location as possible in future recordings.

2.2.4 Sensor recycling

The HD-sEMG grids used are classed by the manufacturer (TMSi) as being 'semi-reusable'. The adhesive backing of the HD-sEMG is removed using isopropyl alcohol (IPA) and any excess glue is cleaned from the sensors using paper towelling. A log is kept of how many times a sensor grid has been used.

2.3 Laboratory Methods

2.3.1 PDMS casting

Polydimethylsiloxane (PDMS) is a flowable, clear silicone elastomer that cures into a solid upon the addition of a curing agent. PDMS in its cured state is pliable, making it easy to handle and manipulate without damaging it when mould making or releasing it from a cast. PDMS is commonly used as the material from which to fabricate microfluidic chips due to flexibility, transparency, low cost, and ability to produce replicas of microstructures[47]. To create the PDMS cast, the base polymer (part A) is added to the curing/cross-linking agent (part B) at a ratio of 10:1, A:B, by weight (Sylgard 184). Lower ratios (i.e. greater amount of curing agent) will result in a stiffer, less flexible cast. The cure time of the mixture can be controlled by temperature, for example the mixture will take 48 hours to cure at 25°C and just 10 minutes to cure at 150°C.

The base polymer and curing agent are combined and the resulting mixture is poured into the mould. The mixture is then placed in a vacuum chamber and degassed to remove air bubbles that may have been incorporated into the mixture during the mixing process. Degassing the polymer mixture also ensures that any air bubbles trapped/sticking to the surface of the microstructures of the mould are removed. Once the mixture is fully degassed (this may require removing it from the vacuum chamber and visually inspecting it for the presence of air bubbles), the mixture is placed in the oven to cure at 65°C for 4 hours.[48] Higher temperatures are not used to avoid damaging or deforming the structure of the mould, despite the slower cure time. When cured, the mould is removed from the oven. Care is taken to remove the PDMS cast from the mould without damaging it. A scalpel, or other sharp instrument, may be used to free the PDMS cast from the walls of the containing mould.

2.3.2 Polyester casting resin

When casting of a material of higher tensile strength/stiffness is required, a 'water-clear' polyester casting resin has been used (Easy Composites) instead of PDMS. The polyester casting resin is also flowable and will cure when combined with the catalyst MEKP (methyl ethyl ketone peroxide). The casting resin is prepared in a similar way to PDMS; after mixing it is poured into the mould and degassed using a vacuum chamber. The recommended amount of catalyst by the manufacturer is 0.3-2% by weight of resin, however the polyester casting resin has a relatively fast gel time compared to PDMS and will become highly viscous in a short space of time. If the mixture is too viscous, it will become very difficult to remove any air bubbles that have become trapped between the resin and the surface of the mould. To slow down the resin's gel time, only a very small amount of catalyst is used: 0.2-0.3% catalyst by weight of resin.

2.3.3 3D model design and printing

All 3D designs in this thesis have been designed using *AutoCAD* (AutoCAD 2017, AutoDesk). The 3D printer used in this project is a Stratasys Objet30 Pro (resolution: 600 dpi x 600 dpi x 900 dpi). Designs were printed using 'VeroClear' material (acrylic – polymethyl methacrylate) in a glossy finish. After printing, the printed parts were placed in the oven at 70°C for 24 hours to ensure any solvents that may interfere with the PDMS casting process described in Chapter 7 have evaporated off.

2.3.4 Microneedle array fabrication

The microneedle arrays (MNAs) described in this thesis were fabricated using casting-shrinkage cycles.[49] A shrinking polyurethane (PU) polymer is used to isometrically shrink casts of 3D-printed masters. The full details of this method for producing MNAs is detailed in Chapter 7.

2.4 Electronic device design and testing

2.4.1 Device fabrication

The electronic devices described in Chapter 6 have been designed using *Altium* (Altium Designer 2016, Altium); software allowing for the schematic capture and

printed circuit board (PCB) design. The Gerber files produced for the completed PCB designs were sent to Elecrow (www.elecrow.com, Shenzhen, China) for manufacture. I assembled the components and PCBs using reflow soldering.

2.4.2 Device testing

The performance of the devices was tested using the equipment shown in Table 2-1. All device testing was performed in a standard laboratory setting, using an Agilent signal generator (see Table 2-1) to generate waveforms. The minimum amplitude the signal generator can produce is 20 mV_{p-p} (peak to peak amplitude), and so signal attenuators have been used to create signals of smaller amplitudes. It is possible to load pre-recorded signals onto the Agilent signal generator, which can be 'played-back' via the generator's output; the Agilent will up-sample the signal to 5 MHz, giving the appearance of a near-continuous signal.

Table 2-1: Equipment used for testing and characterising assembled devices

Equipment	Name and model
Signal Generator	Agilent 33220A
Data Acquisition System	Powerlab (ADInstruments)
Digital Oscilloscope	Agilent MSO-X 2002A

2.5 Hardware Programming

2.5.1 Arduino programming and bootloader

In this thesis, the microcontrollers used in the devices described in Chapter 6 have all been bootloaded with the *Arduino* bootloader. Arduino is an open-source electronics platform, which allows for the simple programming of Arduino hardware products. The Arduino integrated development environment (IDE) makes writing programs for hardware fast and efficient; the IDE has many integrated libraries containing useful functions, which can save the user time. Code written in the Arduino IDE can be easily uploaded to the target hardware over the serial port (e.g. USB cable, paired Bluetooth connection) of the programming computer. The Arduino IDE, however, can only program microcontrollers that have the *Arduino bootloader* burnt to it; microcontrollers

bought from suppliers will not be bootloaded with the Arduino bootloader on delivery. The bootloading can be done using a dedicated in-system-programmer (ISP), or it is possible to configure existing Arduino hardware to act as an ISP.

First the 'ArduinoISP' program is uploaded to the Arduino device being used as the ISP. After this, the Arduino acting as the ISP needs to be wired to the target microcontroller using the wiring diagram shown in Figure 2-3. In the example shown, an 'Arduino Uno' development board is used as the ISP with its serial peripheral interface (SPI) port (see Appendix B) connected to the SPI port of the target microcontroller.

Table 2-2: SPI pins for Arduino Uno

SPI line	Pin number
SCLK	13
MISO	12
MOSI	11
CS	10

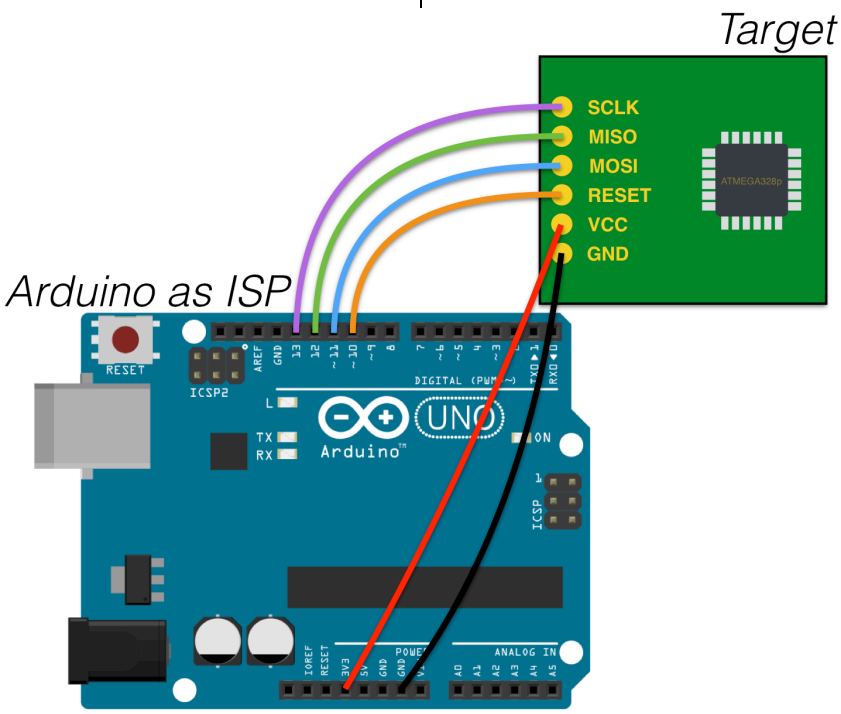


Figure 2-3: Using Arduino hardware as an In-System-Programmer (ISP). *The Arduino hardware (Arduino Uno) acting as an ISP is used to burn the Arduino bootloader onto the target microcontroller so that it can be programmed using the Arduino IDE. The bootloader is burnt over the SPI ports of the devices. Power is supplied to the target device from the Arduino hardware. (Image adapted from Fritzing component image).*

Once connected, the target board microcontroller is selected in the Arduino IDE. The next step is to burn the bootloader to the target microcontroller by pressing *'Burn bootloader'*. The Arduino IDE will report if the bootloading was successful or not.

2.5.2 'SDFat' library

As will be discussed in Chapter 6, the speed at which data can be written to the secure digital (SD) card is of the utmost importance. The Arduino IDE has many libraries in it, including an SD library for writing to and reading from SD cards. However, Arduino's SD library is very slow at writing data and is not suitable for use when attempting to log data at very high speeds. Therefore, an open-source library named 'SDFat' (copyright © Bill Greiman, 2018), which makes it possible to write and read SD files at impressive speeds, has been used in thesis instead.

Chapter 3
Data Processing Pipeline Part 1

In this chapter I describe the first half of the data-processing steps I have taken in order to process the surface electromyography (EMG) data collected using high density-surface EMG (HD-sEMG) grid electrodes from patients enrolled in the longitudinal study at King's College London (KCL) (see Section 2.2.1). I have developed a 'data-processing pipeline', a paradigm by which to sort fasciculation potentials (FPs) by their motor unit. The pipeline takes the patient EMG recordings through the necessary signal conditioning steps, spike detection, and finally groups FPs by their morphology, as demonstrated in Figure 3-1. This chapter covers the steps taken to capture the data, condition the signal, and detect FPs.

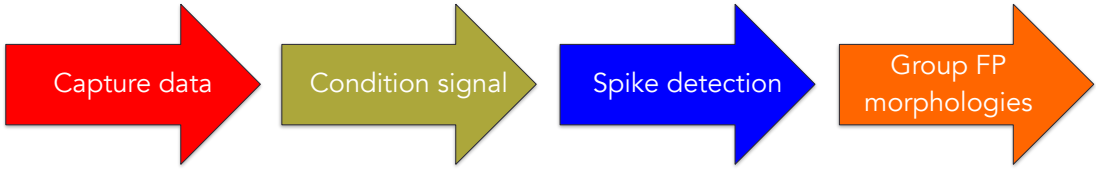


Figure 3-1: Overview of the data-processing pipeline. *From data capture to signal conditioning, spike detection, and finally grouping of the FP morphologies.*

A data-processing pipeline is necessary so that EMG data collected from ALS patients can be processed and the necessary analytics outputted to the clinician in a usable format. My assumption in developing this pipeline is the following hypothesis: Different fasciculating motor units (MUs) will express individual

waveform morphologies, and so by grouping FPs by waveform, a measure of the FP variability can be produced for that MU. MUs will begin to fasciculate at different instances throughout the lifetime of the disease as it spreads through the patient's body. The frequency of fasciculation occurrence from each MU will vary over the course of the disease, becoming more frequent as the motor neuron becomes hyperexcitable and then less frequent as the motor neuron approaches death.[50,51] Therefore, by grouping FPs by waveform and thus MU, it is possible to track the progression of the disease in individual MUs by examining both the changes in FP morphology and the frequency of FP occurrence.

3.1 Data capture

The data used in this thesis to form and test the data-processing pipeline is from a longitudinal study being conducted by clinical collaborators at KCL. This has meant that the data-processing pipeline could be written and tested while in production using 'real-world' patient data without having to rely on the use of simulated data;[52–56] real patient recordings can often contain high levels of noise, voluntary muscle contractions, and unpredictable artefacts that simulated data cannot, and sometimes possibly does not wish to describe. In this study, EMG data has been captured using 64-channel HD-sEMG grid electrodes (TMSi Refa, see Section 2.2.2). The recordings are monopolar, sampled at 2048 Hz, 16-bit resolution, and are 30 minutes in duration. The HD-sEMG electrode is coated in an electrolyte gel to improve the skin-electrode contact,[57] and an adhesive backing is attached to the grid electrode before it is applied to the prepared patient's skin (see Section 2.2.3). In this study, four muscles are recorded from: bilaterally, the *gastrocnemius medial* (calf muscle) and *biceps brachii* (bicep muscle), as shown in Figure 3-2. Photographs are taken of the grid electrode placement and measurements are taken of the electrode's distance to prominent anatomical landmarks, in order to make sure the electrode is placed in the same location each recording session. An additional sEMG electrode is placed over a bony prominence close to the recording site, such as the elbow or ankle, to act as a patient reference (see Figure 3-2(B)).

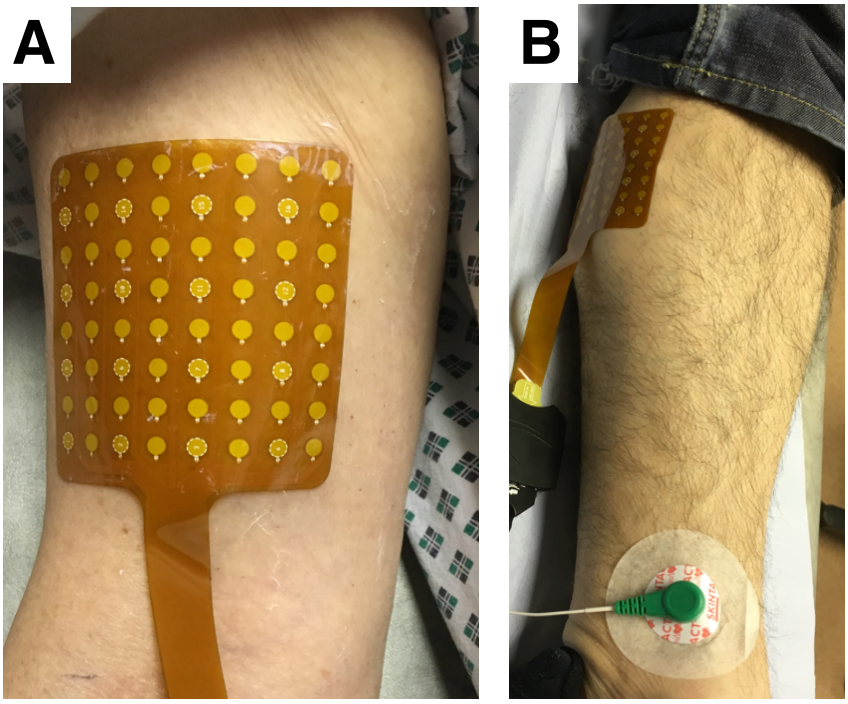


Figure 3-2: Placement of 64-channel surface EMG grid electrodes. *(A) Biceps brachii grid electrode placement. (B) Gastrocnemius grid electrode placement and patient reference over anklebone.*

The EMG recordings produce a very large raw data file size due to the data resolution, number of channels, and recording duration. Data recorded at 16-bit resolution and 2048 Hz, over 64 channels, for 30 minutes (1800 s), and from 4 different muscles will produce a proprietary binary file of 1.9 GB (see Equation (3-1)). When saved in a more generally readable format, such as *CSV* or *.mat* (MATLAB file type), the file sizes are much larger at over 7 GB. This emphasises the importance of pipeline efficiency for rapid processing.

$$Binary\ file\ size = 16\ bit * 2048\ Hz * 64\ channels * 1800\ \text{s} * 4\ \text{limbs} = 1.9\ \text{GB} \quad \textbf{(3-1)}$$

3.2 Signal conditioning and channel inspection

Before the FPs can be extracted, the signal must be properly conditioned in order to remove noise and other unwanted frequencies. Each channel must be inspected for signal quality in case of a poor skin-electrode contact or electrode disconnection. The steps taken to condition and check the 64 channels of the HD-

sEMG grid electrode are shown in Figure 3-3. I investigated a portion of this signal conditioning processing during the MRes phase of this project.

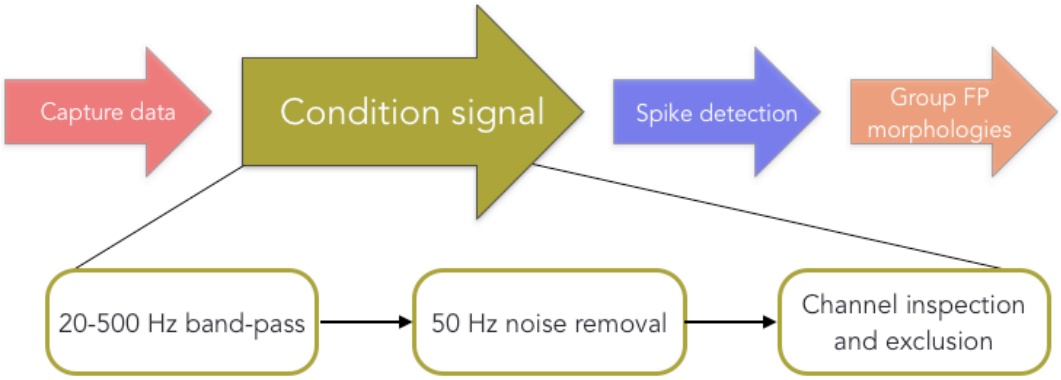


Figure 3-3: Signal-conditioning flowchart. *To condition the signals, they are band-pass filtered from 20 to 500 Hz and then 50 Hz mains line noise is removed. Each channel on the HD-sEMG grid electrode is inspected for signal quality, to ensure the signal has a good skin-electrode contact.*

3.2.1 Signal conditioning

The data recorded by the TMSi Refa is completely 'raw' i.e. the signal has not been passed through any digital or electronic hardware filters in order to remove any undesirable frequencies. The first step in the data-processing pipeline is to condition the signal by putting it through a band-pass filter with cut-off frequencies of 20 Hz and 500 Hz. This removes the low and high frequency components in the signal that do not occur in the FP; frequencies lower than 20 Hz are too slow to be a component of an FP, frequencies higher than 500 Hz are too fast. It is common practice to use a band-pass filter over a frequency range similar to this when examining fasciculations in EMG data.[22,23,38,40,41,43–46,57–59] Motion artefacts in sEMG signals, caused by movement between the sensor and skin, are also removed when using the high-pass cut-off frequency of 20 Hz;[38] a cut-off frequency higher than 20 Hz will remove frequencies belonging to the sEMG signal, resulting in disturbances in FP morphology, which was demonstrated in the MRes phase of this project (see Figure 3-4).[60]

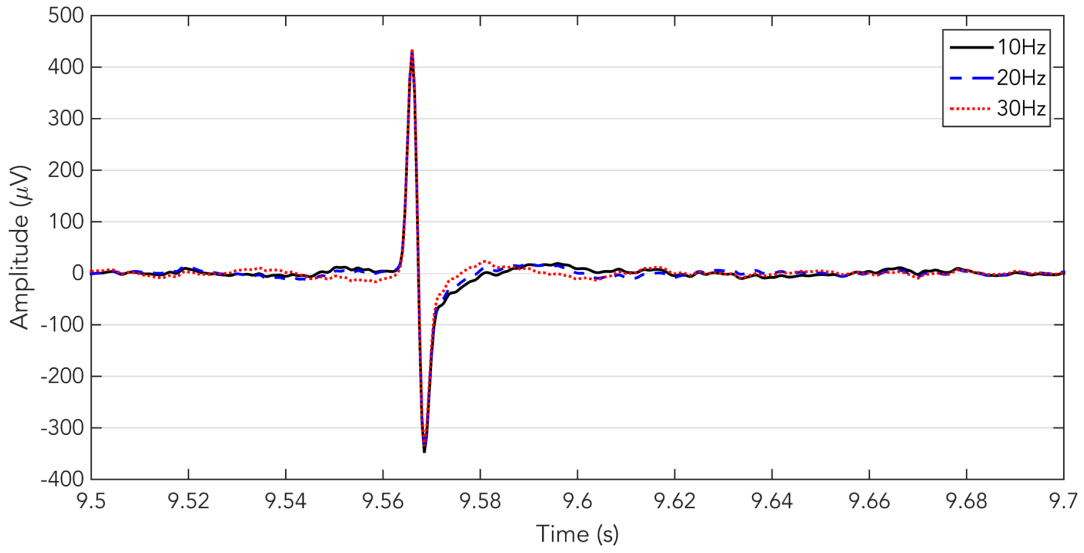


Figure 3-4: Effect of high-pass filter frequency on the surface EMG signal. *A cut-off frequency of 30Hz is too high as it removes frequencies belonging to the EMG signal, resulting in distortions of the FP morphology. Recording was taken from a benign fasciculation syndrome patient. (Image taken from MRes phase of project; Wickham 2015, Department of Bioengineering, Imperial College London).*[60]

In literature, several different values for the low-pass cut-off frequency are used when investigating sEMG; for example, 400 Hz,[22,38,40,57] 450 Hz,[57,59] and 500 Hz.[43–46,54,56] As we are specifically interested in FPs, it may be of interest to preserve higher frequency components of the signal that could show FP waveform variability, and thus be indicative of disease progression. For that reason, the low-pass cut-off frequency was set at 500 Hz, which also follows the practice of other researchers who have also investigated FPs recorded using sEMG.[43–46]

3.2.2 50 Hz noise removal

Surface EMG is inherently susceptible to noise due to the large impedance that can occur at the skin-electrode interface (see Section 1.5). As a result, sEMG recordings can contain large levels of 50 Hz mains line noise that dominates the frequency spectrum as shown in Figure 3-5.

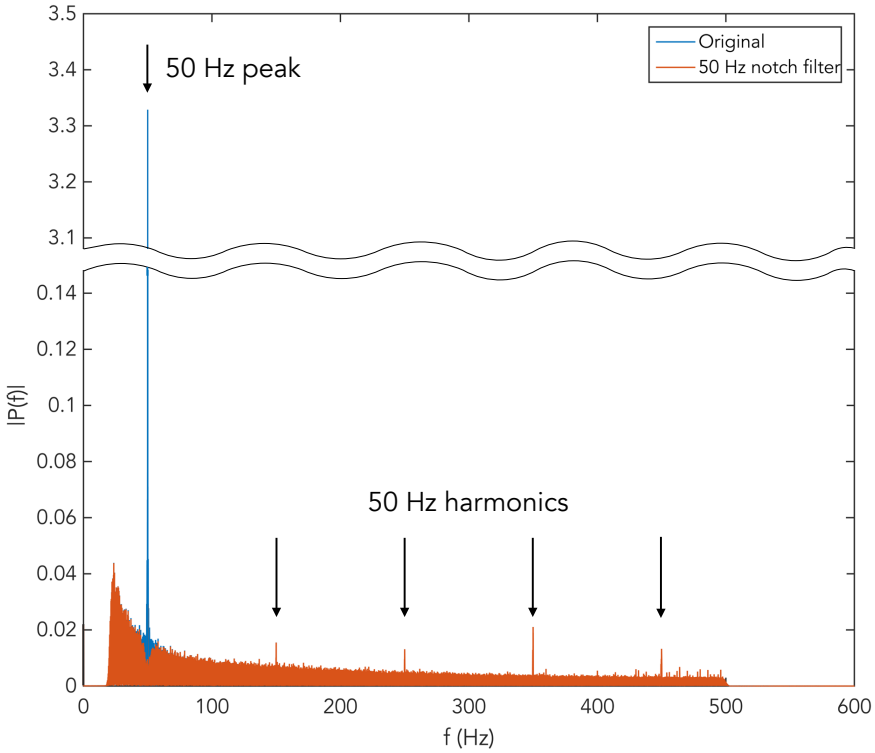


Figure 3-5: Frequency spectrum of band-passed surface EMG recording. *EMG recording from ALS patient band-passed from 20-500 Hz. The spectrum of the original signal is dominated by the presence of a 50 Hz signal (blue), which can be removed from the spectrum using a notch filter (red). However, the positioning of the 50 Hz frequency within the bulk of the spectrum means its removal will likely lead to distortions in the underlying signal. The 50 Hz noise in this sEMG recording has also resulted in visible peaks in the frequency spectrum at its 3rd, 5th, 7th, and 9th harmonics.*

The problem with removing 50 Hz noise from sEMG signals, e.g. with a notch filter, is that it lies in the middle of the bulk of the frequency spectrum, meaning its removal from the signal is likely to result in the distortion of the underlying signal; a portion of the frequency at 50 Hz will exist naturally in the EMG signal, meaning the total removal of it will affect the signal's morphology as shown in Figure 3-6.

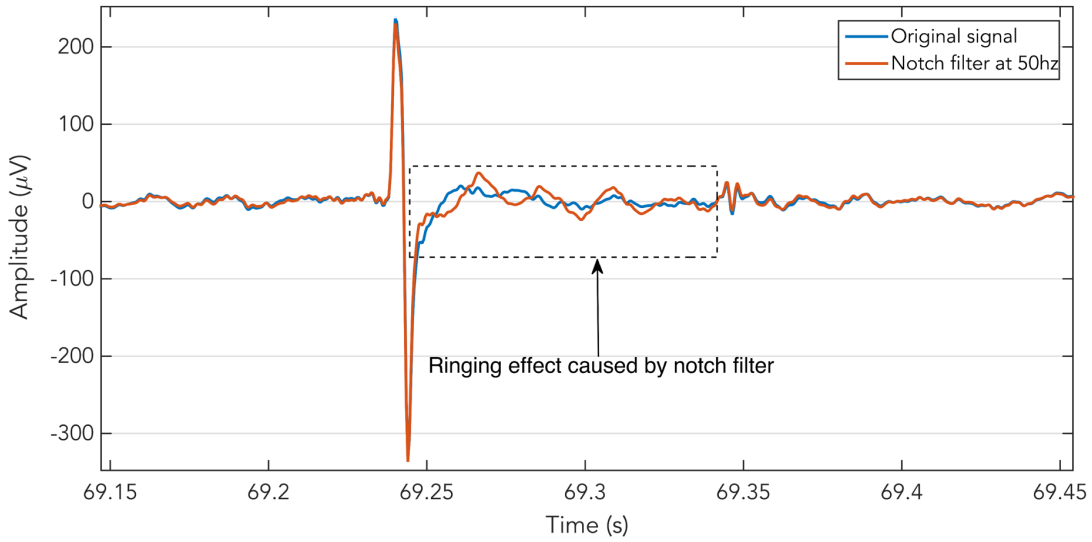


Figure 3-6: Unwanted ringing effect caused by 50 Hz notch filter. *Application of a notch filter at 50 Hz has resulted in an unwanted ringing effect, distorting the morphology of the fasciculation potential. The distortion is due to notch filter completely removing the spectral component of 50 Hz that belongs to the underlying EMG signal. (Image taken from MRes phase of project; Wickham 2015, Department of Bioengineering, Imperial College London).*[60]

For this reason, the removal (and prevention) of 50 Hz mains line noise from EMG signals is a popular topic in literature.[38,56,61–65] Several methods have been explored and developed in hope to successfully de-noise the patient recordings without disturbing the morphology of the FPs:

1. *Linear regression subtraction*
2. *Frequency removal and inverse*
3. *Wavelet filtering*
4. *Empirical mode decomposition*
5. *Notch filtering*

3.2.2.1 Linear regression subtraction

A method described by Baratta et al. is to linearly regress a 50 Hz sine wave against a period of the signal that does not contain fasciculations, and then subtract it from the EMG signal.[61] The regression of the sinusoids in Equation (3-2) and (3-3) give the coefficients for offset and amplitude, and the phase shift that best describe the 50 Hz noise signal in Equation (3-4) and (3-5):

$$X_1(t) = \sin(2\pi f_{50} t) \quad \textbf{(3-2)}$$

$$X_2(t) = \cos(2\pi f_{50} t) \quad \textbf{(3-3)}$$

$$Y_1(t) = a + bX_1(t) + e_1(t) \quad \textbf{(3-4)}$$

$$Y_2(t) = c + dX_2(t) + e_2(t) \quad \textbf{(3-5)}$$

The sinusoids are then subtracted to remove the 50 Hz noise from the signal. The problem that I have found with using this method is that it assumes the mains line noise to be of *"constant amplitude and phase"*,[56] which is an unrealistic expectation when dealing with real-world signals. Even if the patient stays perfectly still during the entire recording session, there are many variables that affect the noise signal making it likely the noisy signal will vary in amplitude, offset, and phase. This method may work well on simulated noisy EMG data,[56] but when tested on real-world recordings such as those taken at KCL, the method performed poorly (see Figure 3-7).

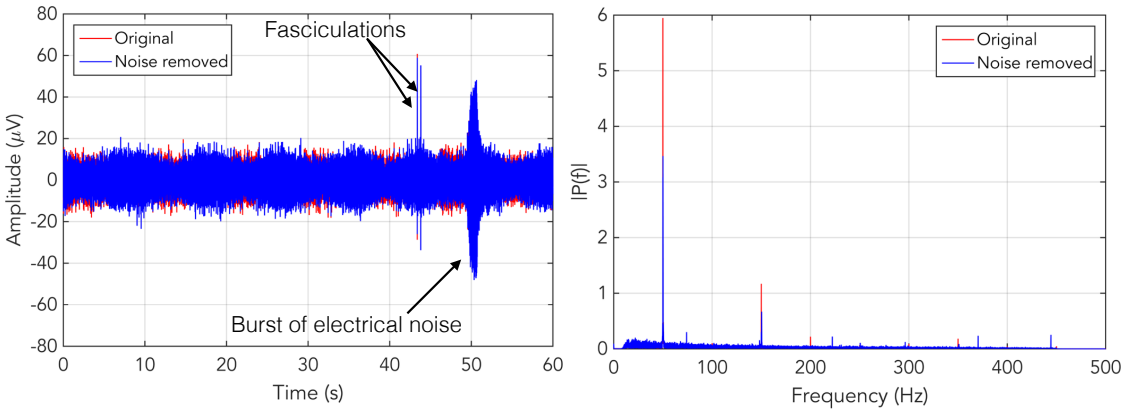


Figure 3-7: Regression-subtraction method performed on noisy EMG recording. *Due to the variation in mains line noise (amplitude, offset, phase) the method has not been successful in completely removing 50 Hz noise from the signal. (A) Time series of EMG data showing the original and processed signals. (B) The power frequency spectrum of the processed signal showing the 50 Hz noise and its harmonics have been reduced, but not removed completely from the signal. (Image taken from MRes phase of project; Wickham 2015, Department of Bioengineering, Imperial College London).*[60]

Instead of using a single quiet period of signal (no fasciculations), I developed the method further by splitting the signal into multiple time-bins and performing the regression-subtraction method on each bin individually, shown in Equations (3-6) to (3-8). This approach aims to better solve the varying-noise throughout the signal. For *N* number of bins:

$$Y_1(t_n) = a_n + b_n X_1(t_n) + e_1(t_n) \quad \textbf{(3-6)}$$

$$Y_2(t_n) = c_n + d_n X_2(t_n) + e_2(t_n) \quad \textbf{(3-7)}$$

$$EMG' = EMG - \sum_{n=1}^{N} b_n X_1(t_n) - \sum_{n=1}^{N} d_n X_2(t_n) \quad \textbf{(3-8)}$$

Using multiple bins when regressing X_1 and X_2 against the EMG signal to account for changes in amplitude and phase of the mains line noise throughout the recording is an effective way to reduce the 50 Hz frequency, as shown in Figure 3-8. However, applying the regression-subtraction method in this way can have a detrimental effect on the signal, especially if it is highly populated with FPs. When applied to a signal containing many FPs, the FPs will inevitably be incorporated into the bins used to find the regression coefficients, resulting in the unwanted, unavoidable effect seen in Figure 3-9.

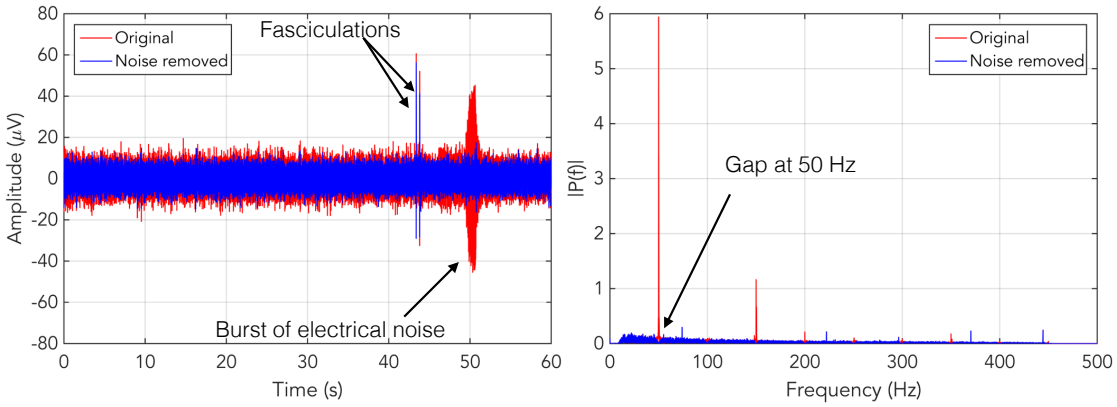


Figure 3-8: Regression-subtraction applied across multiple signal bins. *The regression-subtraction method is used on multiple bins over the signal length to account for the any change in amplitude, offset, and phase of the mains line noise, resulting in a significant reduction in the 50 Hz frequency. (Image taken from MRes phase of project; Wickham 2015, Department of Bioengineering, Imperial College London).*[60]

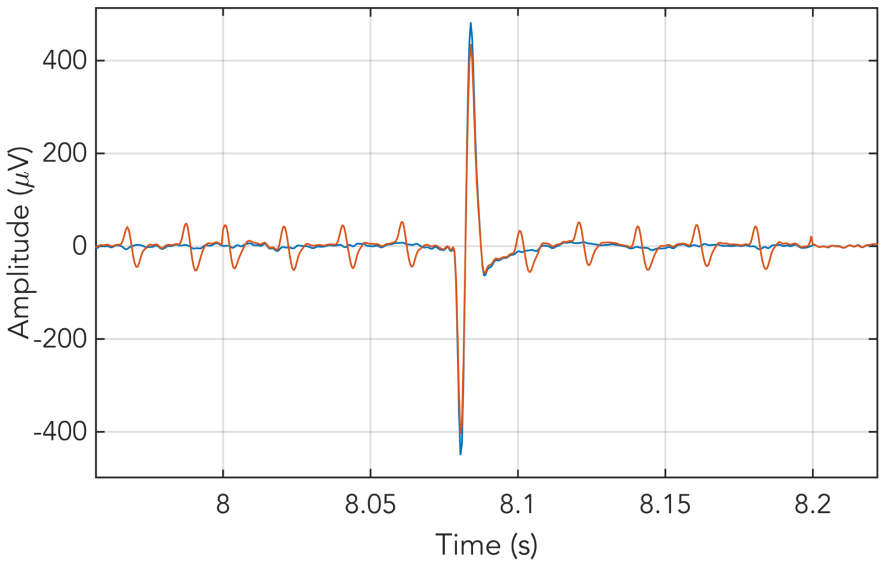


Figure 3-9: Effect of regression-subtraction applied across multiple time-bins. *EMG recordings are likely to contain many FPs, making it very likely each time-bin will contain a fasciculation. In this example, the EMG signal (blue) contains 149 FPs over a 60 s period. The 50 Hz spectral component of the FPs has contributed to the regression coefficients, resulting in distortions (orange). (Image taken from MRes phase of project; Wickham 2015, Department of Bioengineering, Imperial College London).*[60]

This method requires the regression to be performed against a 'silent time period'[61] of the signal, however due to the spontaneous, random nature of fasciculations it can be difficult to determine what sections of the signal truly are 'silent' especially when FPs could potentially be hidden in the signal by the affecting noise.[66] In some patients, fasciculations can be so frequent that it is not possible to find a period of signal long enough that does not contain FPs to regress the sinusoids against.

3.2.2.2 Frequency removal and inverse

A method developed in the MRes phase of the project was to try and remove noise by analysis of the frequency domain rather than the time domain. To do so, the EMG signal was transformed into a frequency spectrum using the discrete Fourier transform function *fft* in MATLAB. Large spikes in the frequency domain caused by noise at 50 Hz and its harmonics (i.e. 100 Hz, 150 Hz, 200 Hz etc.) are set to an average of frequencies surrounding them, 'smoothing' the spikes over. The spectrum is then inversed using the *ifft* MATLAB function to return it to the time domain. This method attempts to remove the noise component from the

signal without completely removing the frequency components that would normally exist in the underlying signal. This method is computationally less demanding than linear regression-subtraction, but requires signal-specific parameters that need fine-tuning, making it unideal for automatic processing.

3.2.2.3 Wavelet filtering

Literature has described the use of wavelet filtering as a method of detecting, extracting, and resolving superimposed motor unit action potentials (MUAPs) from EMG signals.[34,36,53,67] FPs occur due to the involuntary discharge of the motor neuron causing a contraction in the MU, and so their waveform morphology is very similar to that of an MUAP (which is voluntary). The method works by removing background noise and outputting a signal that contains a close approximation to the morphology of the FPs, based on the wavelet shape used. The method could be used to improve the temporal resolution of the signal, removing white Gaussian noise and other background noise.[53,67] Ren et al. model the signal as a summation of noise, W, with the pure signal, S, where the wavelet coefficients a and τ describe the wavelet scale and time index, respectively:[53]

$$X_{\varphi}(a,\tau) = S_{\varphi}(a,\tau) + W_{\varphi}(a,\tau) \quad \textbf{(3-9)}$$

It was investigated in the MRes phase of the project whether wavelet filtering could successfully be used to remove FPs in the presence of mains line noise. Here, the *wden* MATLAB function is used to perform the signal de-noising. The wavelets used in this method are both from the Daubechies family, based on literature using wavelet filtering to extract MUAPS from EMG signals. Daubechies2[67] (db2) and Daubechies5[53] (db5) are used, as shown in Figure 3-10.

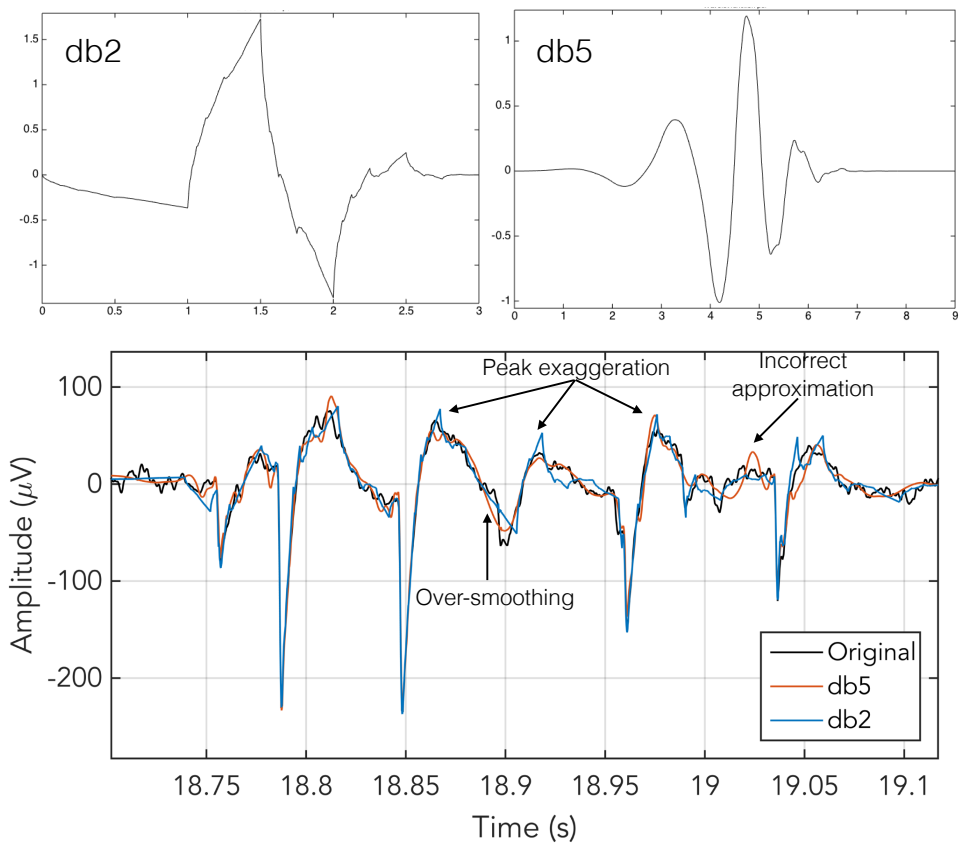


Figure 3-10: Wavelet filtering used to remove noise from EMG signal. *Wavelets db2 and db5 are used to form an approximation of the de-noised FPs. The approximation oversimplifies the morphology of the FPs by smoothing out high frequency components of the signal and exaggerates certain features such as peak height. (Image taken from MRes phase of project; Wickham 2015, Department of Bioengineering, Imperial College London).*[60]

The disadvantage of this method is the incorrect assumption that FPs (and MUAPs) are *"almost the same shape"*[67] as the wavelet morphology - the FP's morphology can be highly unpredictable and complex and should be preserved without approximating it to the shape of the wavelet. The approximation of the FPs to the wavelet causes incorrect peaks, smoothing of high frequency components, and small artefacts that did not belong to the original signal.

3.2.2.4 Empirical mode decomposition

Empirical mode decomposition (EMD) is a method used to break down a signal into several components called intrinsic mode functions (IMFs). Each IMF found in the data represents an oscillatory mode or harmonic component of the signal that can be isolated and removed, for example 50 Hz noise. The method works by first identifying all the local extrema (peaks and troughs) in the signal. A cubic

spline is fitted to the local maxima and local minima of the signal, *X*. The mean of the maxima and minima splines is found, m_1, which is then subtracted from the signal to form h_1. The process is repeated using h_1 as the new signal, finding a new mean spline m_{11}. The process repeats itself until the algorithm reaches a stopping criterion (e.g. standard deviation between h_{k-1} and h_k), at which point h_{1k} is defined as the first IMF, c_1, and is extracted from the signal:

$$h_1 = X(t) - m_1 \quad \textbf{(3-10)}$$

$$h_{11} = h_1 - m_{11} \quad \textbf{(3-11)}$$

$$h_{1k} = h_{1(k-1)} - m_{1k} \quad \textbf{(3-12)}$$

$$X(t) = c_1 + r_1 \quad \textbf{(3-13)}$$

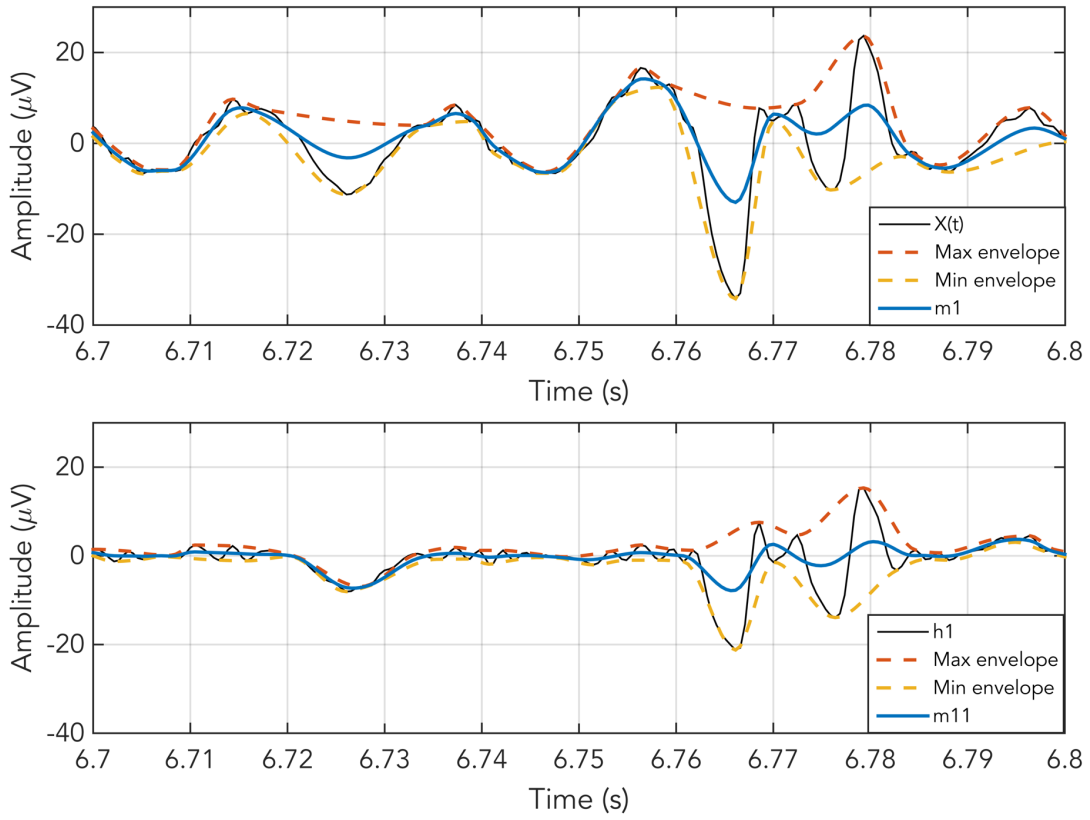


Figure 3-11: First and second iteration of the empirical mode decomposition noise removal method. *(Top) The envelope of the local maxima and local minima is formed on the original signal X(t), which is averaged to find the mean m_1. (Bottom) The mean of the extrema envelopes m_1 is subtracted from the original signal to form h_1 and the process repeats, forming a new mean m_{11}.*

The residual component, r_1, remaining after the first IMF, c_1, is subtracted from the signal is now treated as the new data series. The process repeats until all

IMFs are found. EMD does not require each IMF to be of constant amplitude and frequency, and so EMD can account for changes in amplitude, offset, and phase that other methods like linear regression subtraction cannot (see Section 3.2.2.1). The IMF shown in Figure 3-12 has captured the majority of the 50 Hz signal, however there are sections of noise in the EMG signal that the IMF has not captured.

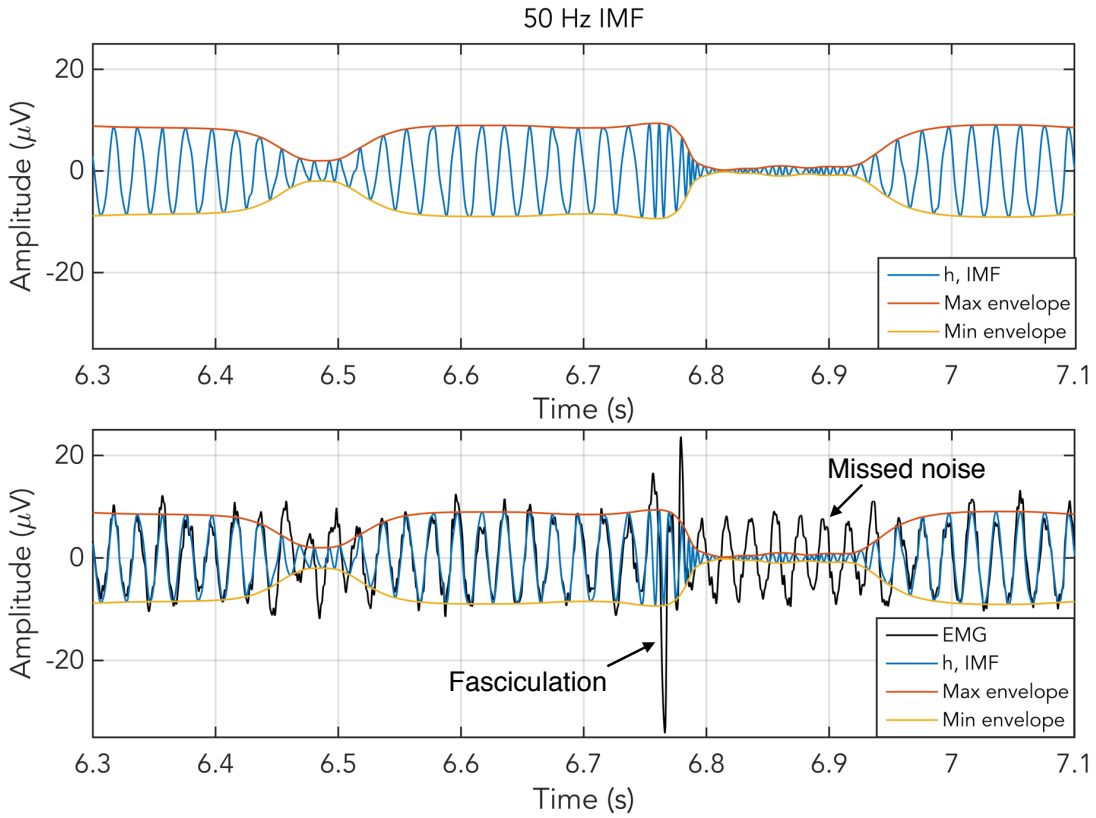


Figure 3-12: Formation of Intrinsic Mode Function using EMD noise-removal method. *The top image shows the final iteration of h (h_{4-74}) which then forms an IMF belonging to the EMG signal, including the local maxima and local minima envelopes. The bottom image includes the original EMG signal for comparison against the 50 Hz IMF. This IMF has not successfully captured the entire 50 Hz noise signal.*

On inspection of the frequency spectrum produced for each IMF, EMD has split the 50 Hz noise signal across several IMFs (IMF3 - IMF6), shown in Figure 3-13. The 50 Hz peak is most dominant in IMF4, the IMF shown in Figure 3-12, but also has a dominant peak in IMF6 followed by IMF5 and IMF3. Figure 3-14 overlays IMF4 and IMF6 with the original EMG signal to show how the majority of the 50 Hz noise is now described by these two modes. A possible explanation for several IMFs isolating different amounts of the 50 Hz noise signal is because the

noise is not a perfect sinusoid; it is affected by changes in phase as well as high frequency components/white noise.

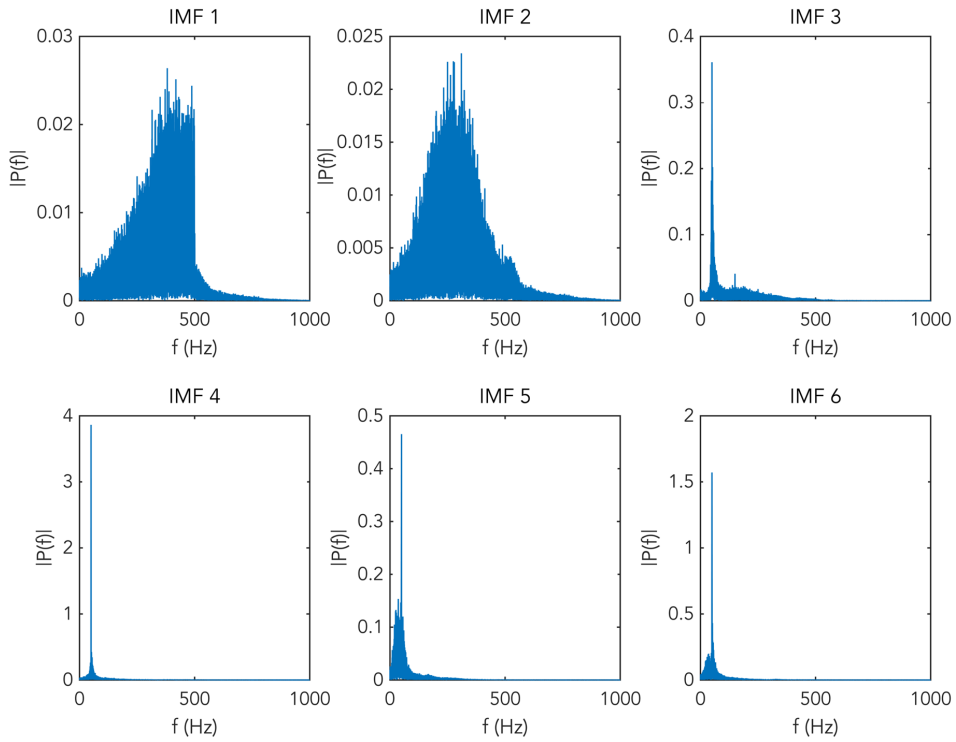


Figure 3-13: Frequency power spectrum of each IMF found in the EMG signal. *The frequency spectrum for IMF4 is heavily dominated by the 50 Hz noise signal, however it is not the only IMF found to have a peak at 50 Hz in its frequency spectrum; IMF6, IMF5, and IMF3 also have a large peak at 50 Hz.*

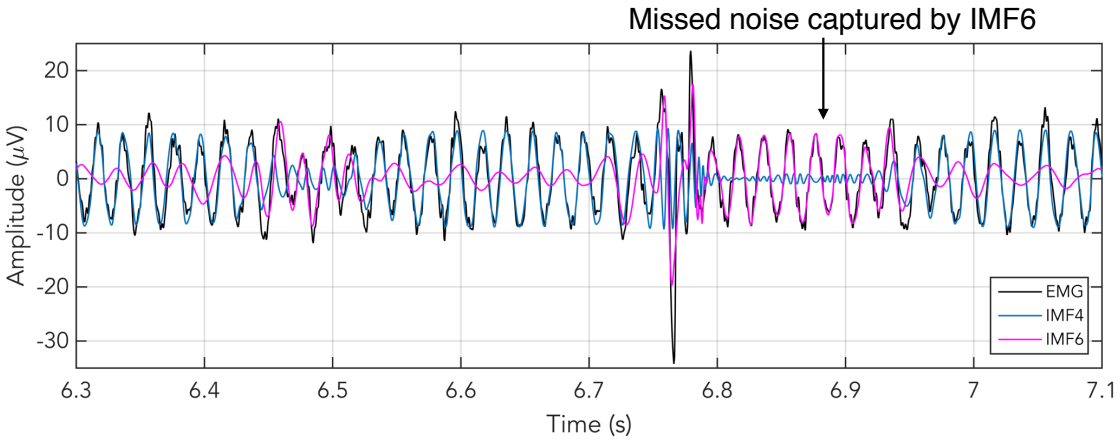


Figure 3-14: 50 Hz noise accounted by several IMFs using EMD noise-removal method. *The unaccounted-for noise that was not captured by IMF4 (blue) has been partially accounted for by IMF6 (pink).*

The IMFs responsible for the 50 Hz noise can be identified and removed from the signal. However, as it can be seen in the IMF frequency spectrums in Figure 3-13,

the IMFs that are responsible for the majority of the 50 Hz noise in the signal also carry other frequencies in their spectrums. Excluding these IMFs entirely would have an effect on the underlying EMG signal. Alternatively, the responsible IMFs with dominating peaks at 50 Hz can be notch filtered and then recombined with the remaining IMFs. In the example shown in Figure 3-15, a notch filter is applied to IMF4 and IMF6 that carry the majority of the 50 Hz noise (see Figure 3-14). The notched IMF4 and IMF6 are then recombined with the other IMFs to reconstruct the EMG signal. The frequency spectrum in Figure 3-15 shows that the peak at 50 Hz has been significantly reduced using the EMD method without removing the entire 50Hz frequency from the signal, as a regular notch filter would have done.

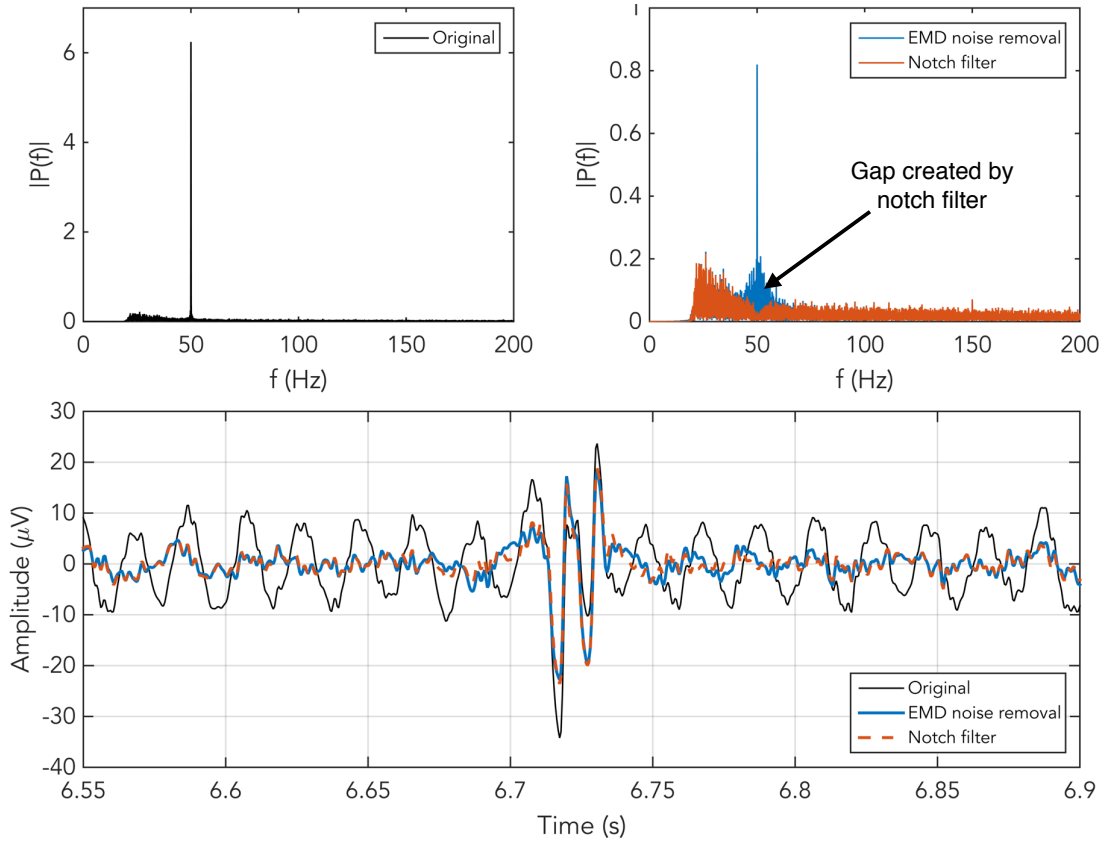


Figure 3-15: Frequency power spectrum and time series of the reconstructed EMG signal formed using EMD noise-removal method. *A notch filter at 50 Hz is applied to IMF4 and IMF6, which are then recombined with the other IMFs to reconstruct the EMG signal. The 50 Hz noise has been greatly reduced using the EMD method, as seen in the frequency spectrum (top-right) compared to the original frequency spectrum (top-left). The new EMD signal is compared to the original EMG signal and the notch filtered signal in the time series (bottom).*

The EMD noise removal method removes a significant amount of 50 Hz noise from the signal, as shown in the time-series in the bottom of Figure 3-15. However, the resulting signal is very similar to that produced when using a notch filter to completely remove the 50 Hz frequency. The EMD noise removal method is very computationally expensive, taking approximately 5 minutes per channel for a 30-minute recording (over 5 hours for all 64 channels) on a standard desktop computer (Macbook Pro 2013, 2.3 GHz Intel Core i7, 16 GB RAM).

3.2.2.5 Notch filter

Removing noise from EMG data without distorting the underlying signal is a difficult challenge, and one that has been extensively researched in literature. There are methods that work well theoretically using simulated data[56] but when applied to a real EMG recording they don't perform well, methods that require case-by-case fine-tuning, or methods that are too computationally expensive. Using a notch filter is a fast, efficient method for removing the 50 Hz frequency from the signal. Despite the distortion it may cause to the underlying signal, it is the fastest, most automatic method I have investigated for removing noise from the EMG recording. The distortion it causes to the fasciculations is not ideal, however it is no more damaging than other noise-removal methods that have been investigated.

From this point onwards, the notch filter has been used to remove 50 Hz noise from the EMG recordings, despite the effect it may have on the morphology of FPs. The effect of mains line noise on the signal can be so great that it can be much more beneficial to have some unwanted distortion of the EMG signal than not to be able to detect FPs at all. The FP in Figure 3-16 has been uncovered by the removal of 50 Hz noise using a notch filter in MATLAB; the effect of 50 Hz noise can be so great on a signal that without removing the noise this FP is unlikely to have been detected at all.

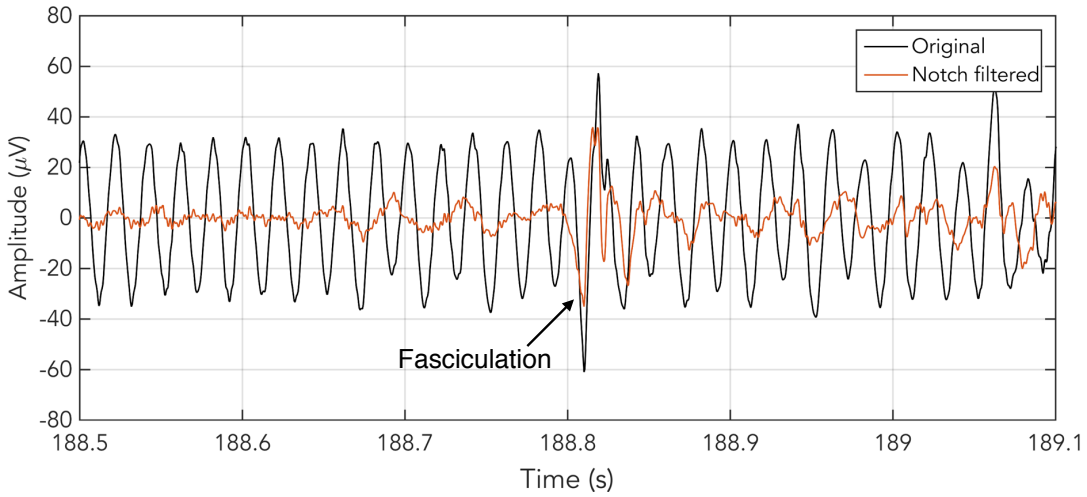


Figure 3-16: Original EMG signal affected by 50 Hz noise and the resulting notch filtered signal. *Despite the removal of the 50 Hz frequency component of the underlying signal, the fast, efficient removal of 50 Hz from the signal has revealed the fasciculation potential previously covered by noise.*

It should be noted that in cases where the EMG signal has been minimally affected by 50 Hz noise, applying a notch filter can have a more detrimental effect than if it were left unfiltered; if the 50 Hz noise on the signal is sufficiently low, it is best left unfiltered than risk 'ringing distortions', as seen in Figure 3-17.

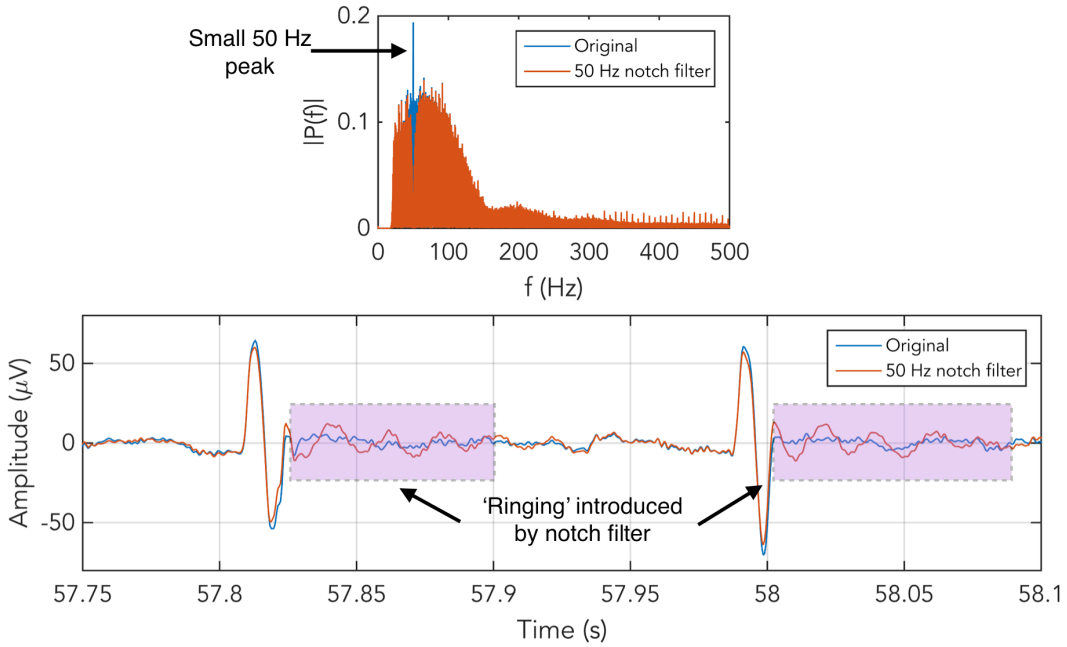
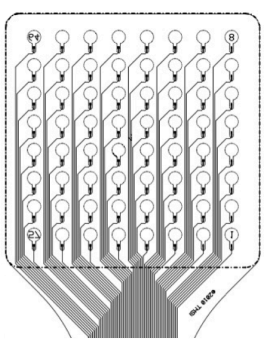


Figure 3-17: EMG signals with low 50 Hz noise levels are less distorted if left unfiltered. *The peak at 50 Hz in the frequency spectrum (top) for this recording is very small; applying a notch filter has caused a distorting ringing effect in the time series (bottom) that is not seen in the original, unfiltered signal. In cases where EMG signals are affected by low levels of 50 Hz noise, the overall distortion to the signal may be less if the signal is left unfiltered (by 50 Hz notch).*

3.2.3 Channel inspection

The HD-sEMG grid electrode is a very useful piece of instrumentation, as discussed previously in Section 1.5. However, with the great number of channels available, it is likely that some will have a poor skin-electrode contact and record a very noisy signal. The Kapton film backing of the grid electrode is flexible, however the film's mechanical stiffness will cause the outer edges of the grid to pull away from the patient's skin. As a result, the outer channels of the grid (red-grid, see Figure 3-18) are likely to be the most affected by noise. This can be partly managed with filtering, however if the skin-electrode contact is extremely poor (or non-existent) the electrode will not have detected any activity from the muscle and will pick up false, noisy signals. Often the inner most-channels (green-grid, see Figure 3-18) will have a good skin-electrode connection. However, on occasion some electrodes may not be properly coated in conductive gel, resulting in a pocket of air between the electrode and skin causing a poor skin-electrode contact, and therefore producing very high levels of noise on the signal.



8	16	24	32	40	48	56	64
7	15	23	31	39	47	55	63
6	14	22	30	38	46	54	62
5	13	21	29	37	45	53	61
4	12	20	28	36	44	52	60
3	11	19	27	35	43	51	59
2	10	18	26	34	42	50	58
1	9	17	25	33	41	49	57

Figure 3-18: HD-sEMG grid electrode by TMSi. *Each channel is numbered in the grid (left). Red grid (8x8 channels), orange grid (6x6 channels), green grid (4x4 channels). It is common for the outer channels of the red grid to have a poor or no connection due to the mechanical stiffness of the grid electrode pulling the channels way from the patient's skin. (Image adapted from HD-sEMG drawing, TMSi).*[68]

Manually inspecting each channel for quality is time consuming and not feasible if this pipeline is to automatically process EMG signals from ALS patients. I have developed a new method to inspect each channel for quality using the discrete Fourier transform, where a measure termed '*activity*' is produced for the recording (signal + noise) of each channel. The method is based on the knowledge that fasciculations are sporadic and relatively infrequent, and so the ideal EMG recording would be a silent signal containing just FPs. This ideal, zero-noise recording would therefore have a relatively low 'activity' compared to a noisy channel with a bad contact/connection. The measure of 'activity' is calculated by integrating the Fourier frequency spectrum of the signal to find its area-under-the-curve value. Noisy channels will contain larger levels of frequencies, relative to non-noisy channels on the same grid, giving them higher measures of 'activity'. This method is also an effective way of detecting 'dead channels' – channels that have recorded no data at all; normally due to a poor connection, these channels will have recorded a single offset value. Band-passing this offset-value will cause very large rippling peaks at the beginning and end of the signal that will result in a very high level of 'activity'. Figure 3-19 compares three channels from the same recording: a clean channel, a noisy channel, and a disconnected 'dead' channel. The frequency spectrums for the three channels are computed, which demonstrates the different levels of frequency contribution in each channel recording; both Channel 31 and Channel 36 have been affected by

50 Hz noise, but Channel 31 considerably more so. Channel 31 also demonstrates 1/f 'pink' noise. Channel 36 has an extremely large peak at 0 Hz due to the constant offset value recorded and a secondary peak at 20 Hz caused by the high-pass filter portion of the band-pass filter applied in the signal conditioning stage of the data-processing pipeline (see Section 3.2.1).

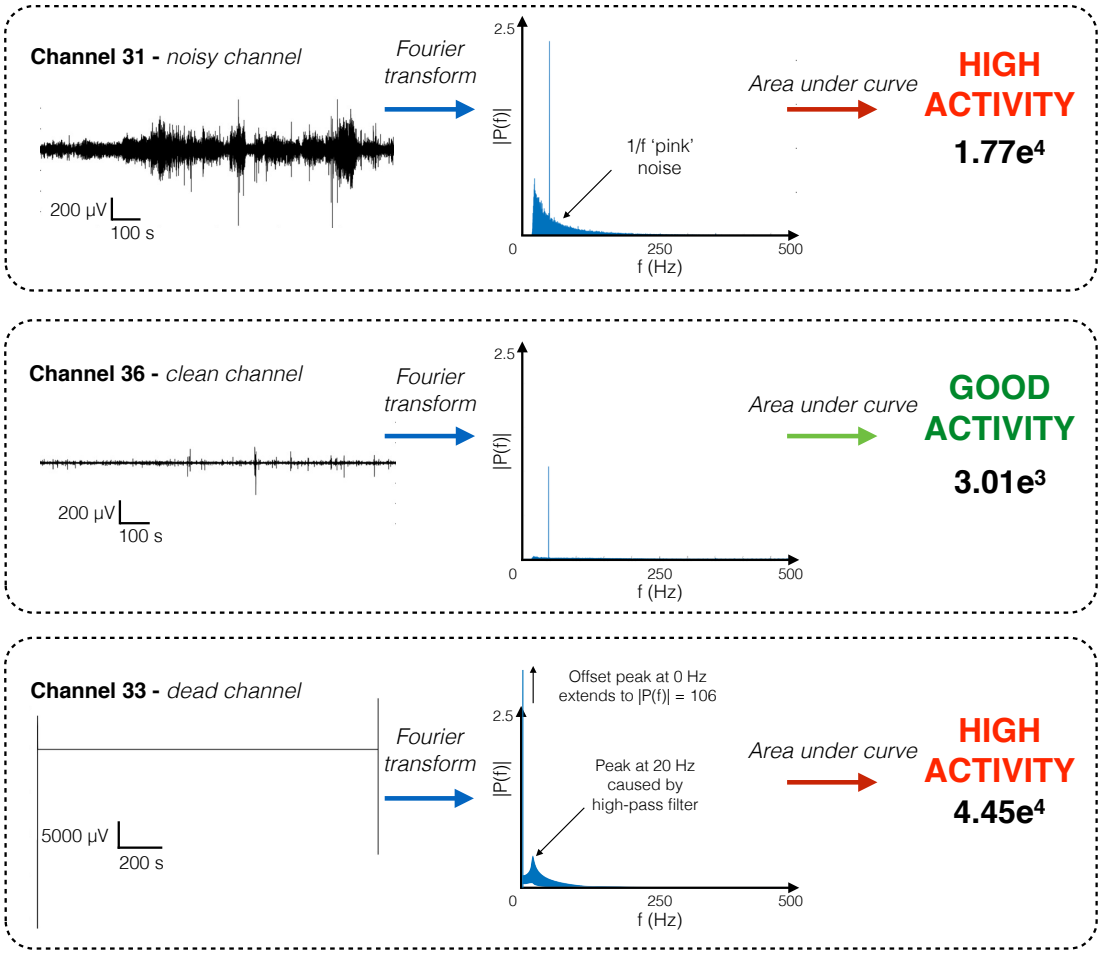


Figure 3-19: Three channels from same recording session displaying very different 'activity' levels (band-pass filtered 20-500 Hz without 50 Hz notch filter). *Channel 31 (perimeter of orange grid) has displayed unusually high amounts of electrical activity in its time-series compared to Channel 36 (inner channel of green grid). Channel 33 (perimeter of red-grid) is a 'dead' channel, which has recorded a single offset value; when band-passed, this causes extremely large rippling peaks at the beginning and end of the signal. Integrating the frequency spectrum of each channel, to find an area-under-curve value, gives the channel's 'activity' level; Channel 31 and Channel 33 are found to have a considerably higher level of activity than their surrounding channels (i.e. Channel 36) and are therefore excluded from further analysis under the assumption they have either a poor skin-electrode contact or are completely disconnected.*

Channels with abnormally high activity levels are removed and ignored for the rest of the data-processing pipeline. Figure 3-20 shows the different activity levels found across the 64-channel sensor grid; as previously mentioned, it is often the outer channels of the grid that have a poor/no skin-electrode contact and thus a high activity level. The criterion for excluding channels from the data-processing pipeline is if its level of activity is greater than the median value (Q_2) plus 1.5 times the inter-quartile-range (Q_3-Q_1):

$$Outlier > Q_2 + 1.5(Q_3 - Q_1) \quad \textbf{(3-14)}$$

Instead of using median and interquartile range to detect outliers, a combination of mean and standard deviation can be used to exclude noisy channels (e.g. > $\mu + 2\sigma$). However, this method can be problematic, as channels with extremely high measures of activity (outliers) will cause an increase in the values for mean and standard deviation, thus causing the exclusion threshold to shift and be unrepresentative of the bulk of the data. Using the median and interquartile range instead will mean that very large outliers will not affect the exclusion threshold set in Equation (3-14).

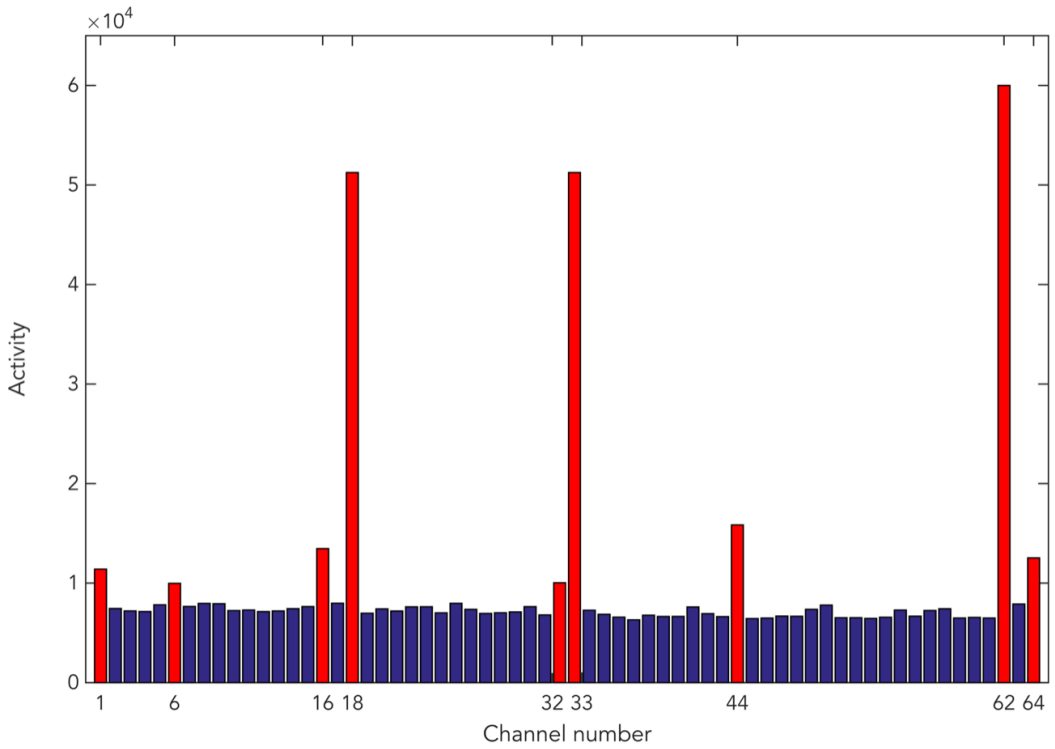


Figure 3-20: Bar chart demonstrating excluded channels with high levels of 'activity'. *Channels are excluded (red) if their activity level is greater than $Q_2 + 1.5(Q_3-Q_1)$. Channels 1, 6, 16, 18, 32, 33, 62, and 64 are all channels on the outer perimeter of the grid electrode that have a poor skin-electrode connection. Channel 44 is a non-perimeter channel that has a high activity level, which is possibly due to a poor connection caused by a deficit of electrolytic gel resulting in an air bubble.*

As previously mentioned, air bubbles can become trapped between the electrode and skin if insufficient amounts of electrolytic gel are applied to the grid. This results in a very high impedance at the skin-electrode interface, causing a very high measure of activity for that electrode. Figure 3-21 shows the effect of a poor skin-electrode contact at Channel 30, likely due to a pocket air trapped between the electrode and skin.

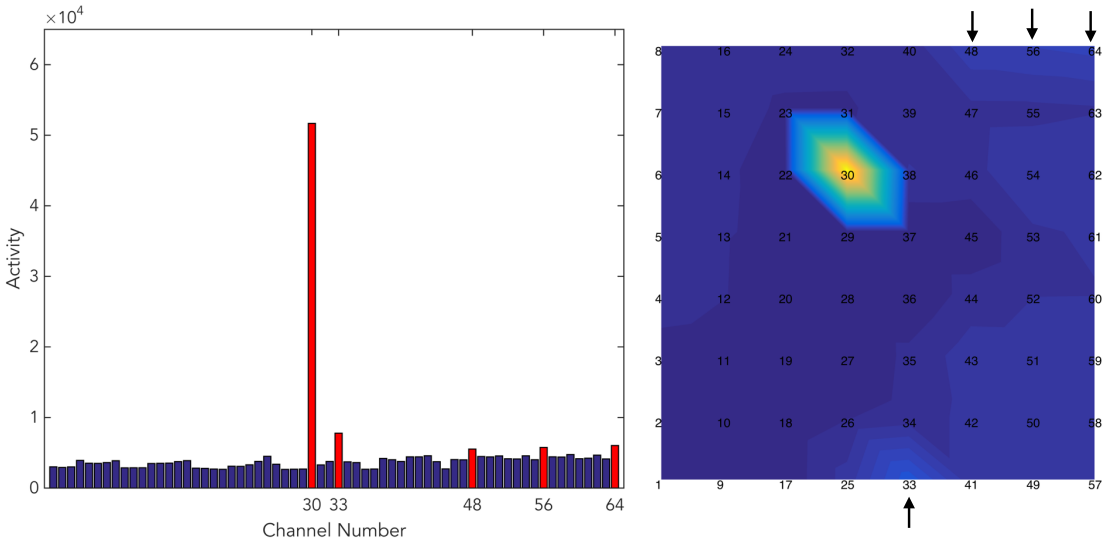


Figure 3-21: High level of activity caused in a non-perimeter channel likely due to insufficient application of electrolytic gel. *Poor application of gel can cause air between the skin and electrode, resulting in a high electrode impedance and thus a high level of 'activity'. The activity in Channel 30 is clearly abnormal compared to the other channels, likely due to an air bubble. Channels 33, 48, 56, and 64 are all perimeter channels and are also flagged as having high levels of activity, probably due to either poor grid application, or failure of the adhesive layer to the skin.*

3.3 Fasciculation Potential Detection

The FPs ideally need to be extracted automatically from EMG signal. Manual detection of FPs from the EMG time series is not uncommon,[16,19,23,33] however as the recordings I wish to process are likely to take place frequently, be very long in duration, and be over many channels (i.e. using HD-sEMG), the manual detection of FPs from the time-series would be a very arduous task. A difficulty of automatically detecting FPs is that they occur spontaneously in the time series with an unpredictable waveform and amplitude. Many methods of FP detection either rely on the manual detection of fasciculations or are based on some form of amplitude 'thresholding';[16,22,23,31,32,35,43,45,46,53] when the EMG signal goes above or below a certain value/threshold in order to detect when a fasciculation has occurred. The problem with this method is that often the EMG signal will contain noisy artefacts or voluntary muscle contractions; detection methods based on arbitrary threshold values can lead to the false detection of FPs. In addition to this, when using a set value for amplitude threshold, legitimate FPs that just fall

short of the threshold value will be wrongfully rejected. For example, the FP in Figure 3-22 would be rejected using the 50 μV amplitude threshold used by de Carvalho & Swash,[23] based on the smallest accepted amplitude when defining a MUAP by Rosenfalck.[69]

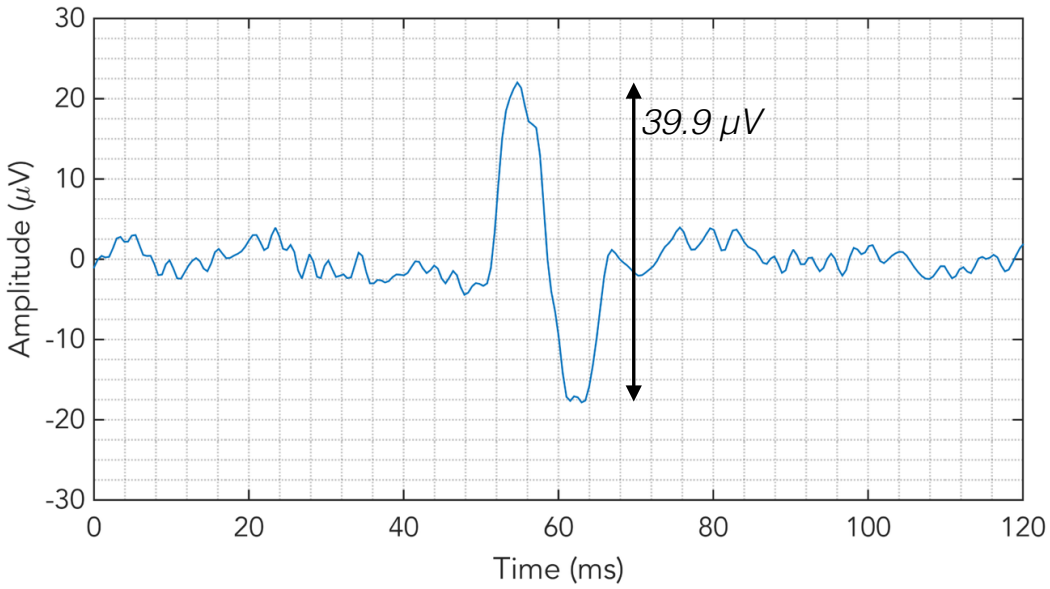


Figure 3-22: Amplitude threshold can exclude legitimate FPs. *When using an amplitude threshold, legitimate FPs that have an amplitude just short of the threshold can be wrongfully rejected.*

The detection of FPs follows the steps shown in Figure 3-23: a novel detection method for spikes/FPs based on probability, the refinement of those detected FPs, consolidating the 64 channels of the HD-sEMG grid into a single *'super-channel'*, and the removal of voluntary motor unit action potentials (MUAPs).

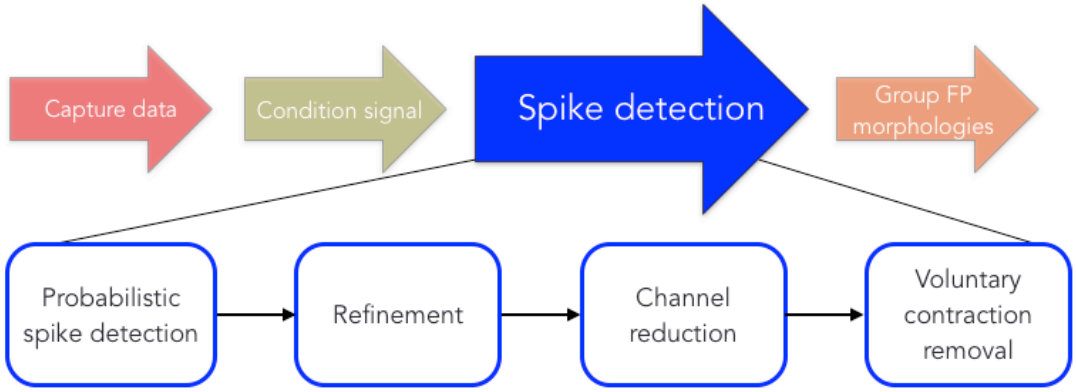


Figure 3-23: Spike detection flow chart. *Flow diagram showing steps taken in FP detection: probabilistic spike detection, refinement, channel reduction and voluntary contraction removal.*

3.3.1 Probabilistic spike detection

The approach taken in this thesis is to use a method I have developed termed '*probabilistic spike detection'.* This method provides a rational basis for selecting a threshold for spikes, which is a variant to factors such as grid, electrode, skin, gel, patient, and operator. I began development of this method of FP detection in the MRes phase of this project, but I have since developed, refined, and tested this method further in my PhD phase. The advantage of this approach is that during fasciculation detection, the baseline level of noise/disturbances in the signal is accounted for regardless of the inputted signal, rather than just using an arbitrary, pre-defined amplitude threshold value. This approach ensures that the FPs are detected with a confidence level that is specific to the analysed signal, rather than a 'one-size-fits-all' approach that is taken by some threshold-based detection methods. The efficacy of this probabilistic spike detection method has been demonstrated in partnership with my clinical collaborators at KCL.[70]

Step 1:

A histogram is created using ***every single data-point*** in the signal; this demonstrates how data is distributed in the signal, where each bin of the histogram represents a single microvolt step between the most positive and most negative values found in the signal. The bins closest to 0 μV will have the highest frequencies in the histogram, making it very tall and narrow in appearance, as shown in Figure 3-24. This is due to the signal being AC-coupled when band-pass filtered; despite any noise, motion artefacts, voluntary contractions, or very large fasciculations, the signal will have a mean value of approximately 0 μV. Additionally, the amount of 'empty' signal (not containing fasciculations) in a recording will vastly outweigh the amount of signal containing true FPs, which explains why the bins of data centred around 0 μV will have very high frequencies compared to the rest of the histogram. Notice how the example histogram shown in Figure 3-24 is asymmetric; this particular signal has values that are more negative than positive, likely due to a large negative phase of a FP. This method of probabilistic spike detection takes into account what the actual values of the signal are being presented, rather than

what values of an FP are to be expected (cf. using arbitrary amplitude thresholds).

Step 2:

The histogram is then divided by the sum of the bin centre frequencies (the total number of data points in the EMG signal) to form a probability density function (PDF). The PDF is normalised against its maximum value and subtracted from 1, producing the graph on the right of Figure 3-24 that has probability values between 0 and 1 (a zero value meaning a spike has not occurred). A 'probability threshold' is set, and values over this in this new probability function (see Figure 3-24) are considered to be potentially belonging to an FP.

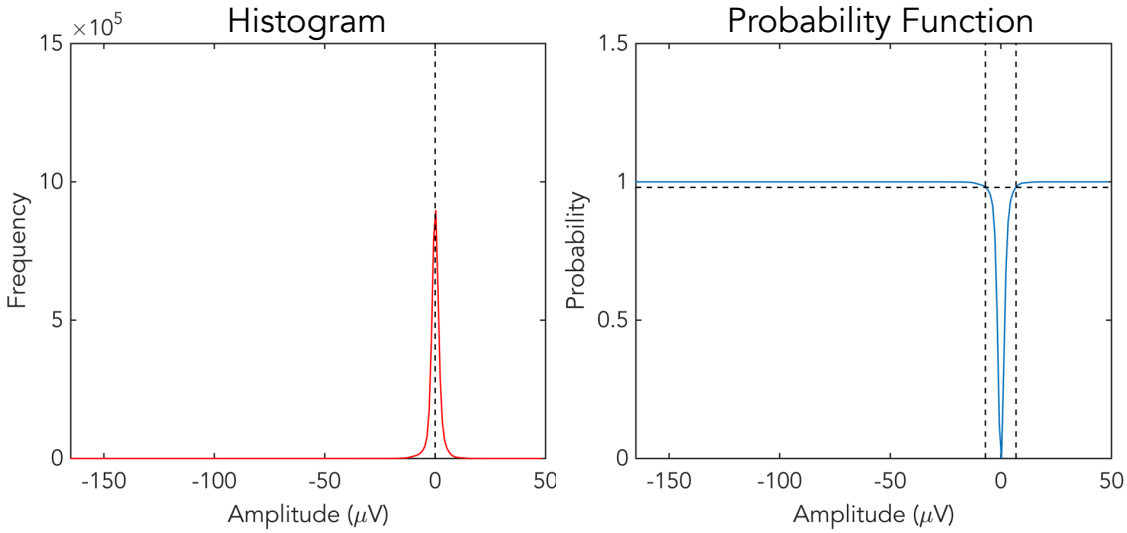


Figure 3-24: Histogram and probability function created from the values within an EMG recording. *The histogram (left) shown is asymmetric due to the signal possessing values of greater negative values that positive. The histogram is inverted and normalised in order to create the probability function (right). In this example, the probability threshold is set at 0.98 corresponding to approximately ±7 μV; values greater than this are considered to be potential components of an FP.*

Step 3:

Returning to the original data in the time domain, the zero-crossing-points (ZCPs) of the signal are found; locations where the signal crosses 0 μV, moving from positive to negative or from negative to positive. The length of signal between each ZCP is termed as an 'epoch'. The maximum value (or minimum value, if negative) of each epoch is found. These values are then inputted into the probability function to give a score for how likely it is that epoch belongs to a

spike;[i] all data-points within an epoch will receive the same probability score as its maximum/minimum value, as shown in Figure 3-25. Whether or not these epochs are further evaluated for belonging to a spike will depend on the probability threshold.

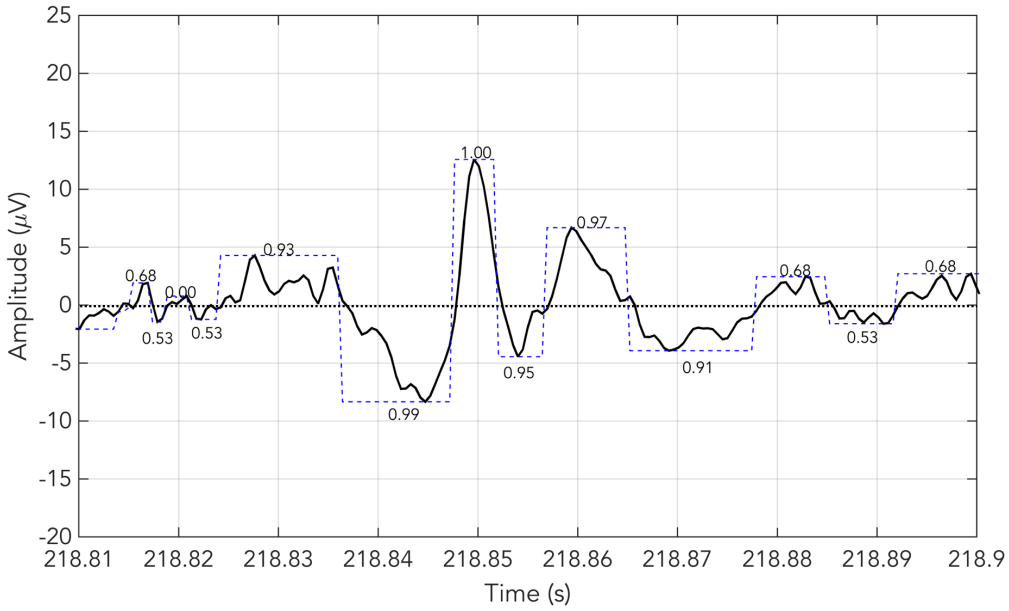


Figure 3-25: Probability scores for signal epochs. *Each 'epoch' of data (length of signal between zero crossing points) is given a probability score of how likely it is a component of an FP. The probability score given to each epoch is determined by the probability function, using the epoch's maximum absolute value. The probability scores given to each epoch in this example have been rounded to two decimal places.*

If the epoch passes the probability threshold it has differed sufficiently from the level of baseline noise/activity and is considered a potential phase belonging to a spike. Epochs in close proximity to each other are conjoined to form the full waveform, and the time-stamp (location) of the spike is found. Figure 3-26 demonstrates how the individual epochs, found using 'probabilistic thresholding', are stitched to the adjacent epochs to form the FP shown.

[i] *The term 'spike' is used here as the detection algorithm has not yet determined whether the data truly belongs to a fasciculation potential; the spike could be of other neurophysiological origin.*

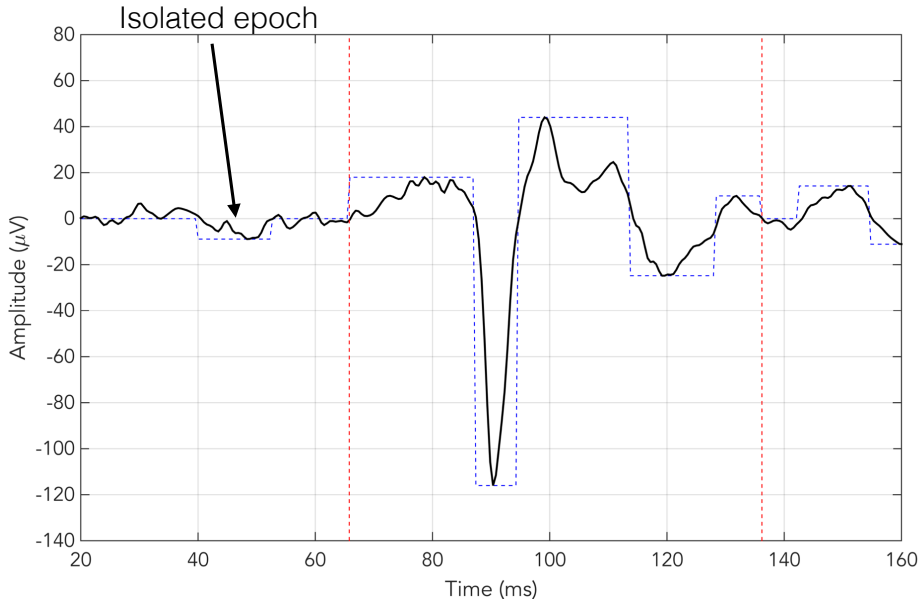


Figure 3-26: Conjoining adjacent epochs that have passed the probability threshold to form the fasciculation potential. *Epochs below the probability threshold are rejected. The FP start and end points (red dashed lines) are defined by the ZCPs of the first and last epochs that form the FP. The FP detected in this example has been formed from four epochs stitched together and has a duration of 70.4 ms and an amplitude of 159.3 μV.*

Figure 3-27 summarises the stages required to detect FPs using this probabilistic spike detection method.

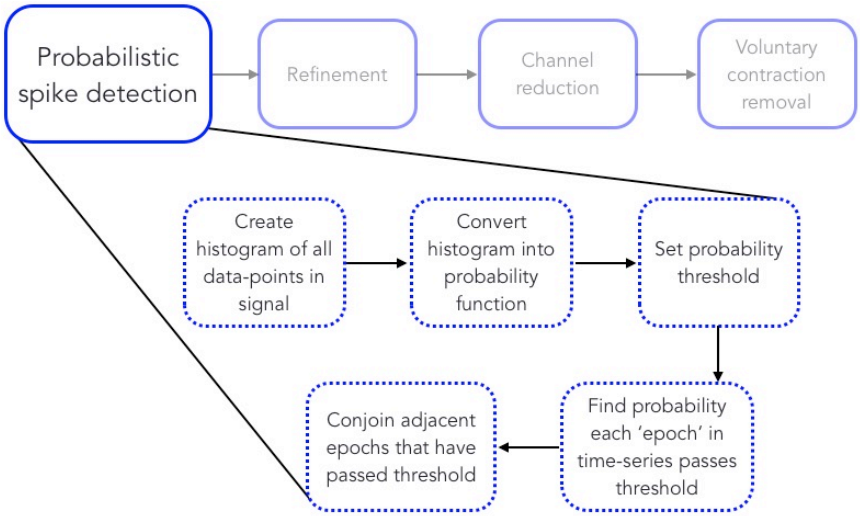


Figure 3-27: Steps taken using the 'probabilistic' spike detection method. *The method provides a rational basis for the detection of FPs depending on the actual values of the signal, rather than choosing an arbitrary amplitude threshold.*

3.3.1.1 Refining

Not all epochs belonging to an FP will be directly adjacent to one another - some FPs will contain epochs of data that are separated/isolated from the rest of the waveform. This can be due to noise on the data, causing very short, low amplitude epochs that fail the probability threshold, but can also be due to the increasing polyphasicity of the fasciculation causing the morphology to erratically move above and below the ZCP. For this reason, after the initial conjoining of adjacent epochs, a 'closeness threshold' is used to conjoin epochs that are within a particular distance of each other, as demonstrated in Figure 3-28. Criteria for rejection include: not enough phases (single phases cannot be an FP, such as the labelled isolated epoch in Figure 3-26), too many phases or too long in duration (likely noisy disturbance/motion artefact, or a full muscle contraction), or too low amplitude (cannot confidently distinguish between genuine FP event and stochastic artefacts/deviations in the signal).

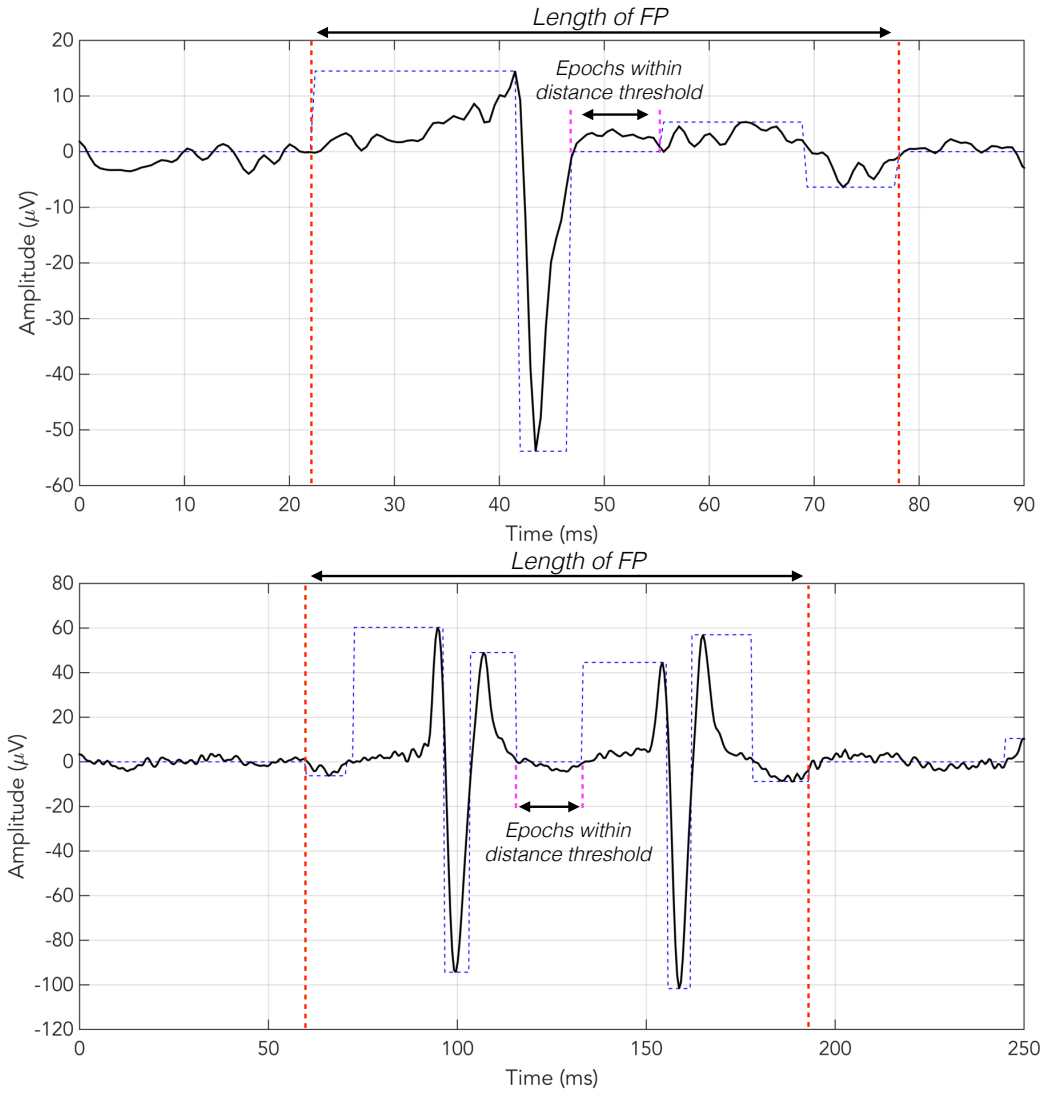


Figure 3-28: Refining stage used to conjoin separated epochs. *The refining stage has conjoined epochs, which are within a distance threshold of one another, into the FP bracketed by the red dotted lines. (Top) Epochs have been conjoined together to complete the full waveform of this FP. (Bottom) Two FPs of matching morphology within a distance threshold have been conjoined to form a doublet, a waveform characteristic with motor neuron hyperexcitability.*

The probabilistic spike detection method has taken account for baseline noise and disturbances in the signal. However, there may be sporadic fluctuations in the signal baseline that can have the appearance of a spike. Therefore, there is an opportunity for the user to set an amplitude threshold in the refining stage, should they wish to. By setting an amplitude threshold, any fluctuations in the signal baseline that could have been detected by the algorithm can be eliminated, thus reducing the risk of detecting false positives (see Section 3.3.1.2). It is also a way of eliminating low-amplitude neurological waveforms that are not of

interest such as cross-talk of voluntary MU activity (see Section 3.3.3) or FPs from neighbouring muscles.

3.3.1.2 Probabilistic spike detection performance

Receiver operator characteristic (ROC) analysis was performed to test the sensitivity and selectivity of the 'probabilistic spike detection' method for many different probability thresholds. To assess the sensitivity (true positive rate, TPR) and selectivity (true negative rate, TNR) of the spike detection algorithm, FPs needed to be detected manually from the EMG recordings. ROC analysis can be a tool for choosing the best thresholds for the automatic spike detection algorithm (i.e. probability threshold, closeness threshold, amplitude threshold).

To test the detection method's effectiveness, FPs were detected manually using two 60-second periods (245,760 data-points in total) of EMG recordings from two different patients. Multiple patients were used for the ROC analysis to increase the variability/type of FP morphology the spike detection algorithm would capture (i.e. amplitude, duration, polyphasicity etc.). Using two different recordings also increases the variability of the baseline noise the spike detection algorithm has to cope with when identifying FPs. The periods of data, the recording sessions, the limb, and the patients the recordings were taken from were all chosen completely at random. The data had already been pre-processed for signal quality (20-500 Hz bandpass and 50 Hz notch filter) and once selected, was visually inspected to ensure the signal period did not contain any large segments of unwanted activity (i.e. muscle contractions). The channels the recordings were taken from were checked to ensure they did not have excessively high levels of 'activity' (see Section 3.2.3). Over the two 60-second periods, 165 and 111 FPs were manually detected, respectively (276 FPs in total), and the precise time-stamps of each fasciculation onset and offset were recorded; a process that was extremely labour and time intensive. Both recordings and all manually detected FPs were re-inspected to eliminate any errors that could occur from manual detection. Only periods of signal that could be said to confidently be FPs were selected in the manual detection; fluctuations in the signal amplitude that did not differ enough from the baseline noise were

not selected. The 60-second periods and the manually detected FPs are shown in Figure 3-29.

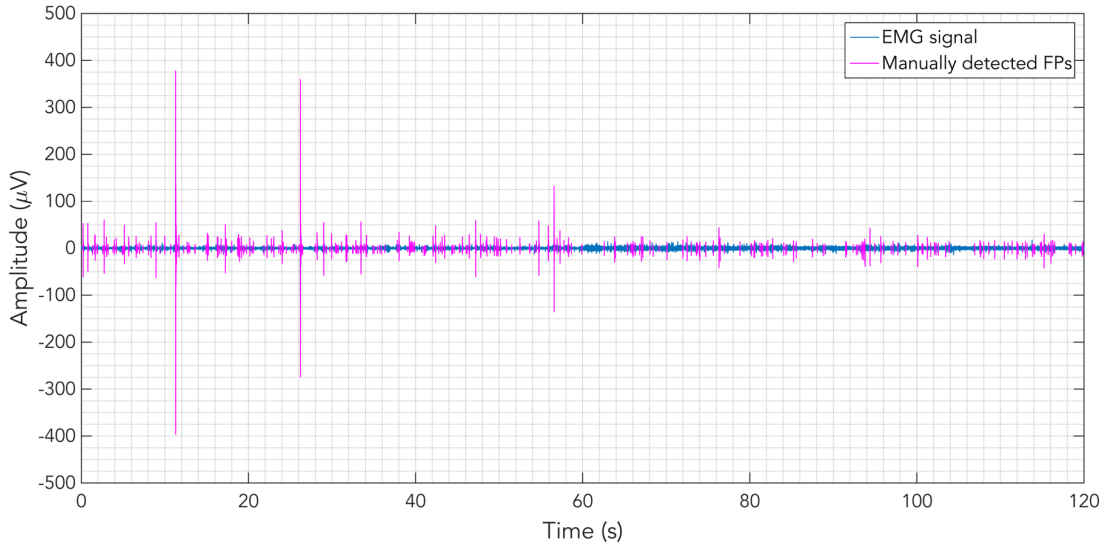


Figure 3-29: Manually detected FPs from two 60-second periods of EMG data recorded from two different patients, combined into the same time-series. *The patients, limbs, recording sessions, and periods of data were all chosen completely at random. Neurological waveforms believed to be FPs were identified visually by scanning through the time-series and manually noting the time-stamps of FP onset and offset. The signal has already been pre-processed (20-500 Hz band-pass, 50 Hz notch filter) and has passed the channel inspection for activity level.*

The manually detected FPs are used to form the 'labelled' dataset; all data-points belonging to manually detected FP are labelled 'positive' (21,373 data-points), and all other data-points belonging to the rest of the signal are labelled as 'negative' (224,389 data-points). Due to the sporadic nature of fasciculations, the proportion of data-points labelled as 'negative' is much greater than those labelled 'positive'. Using the automated probabilistic spike detection algorithm, the 'predicted' dataset of positive/negative data-points is also formed. Predicted-positive data-points that match with the labelled-positive data-points are classed as 'true-positives'. Similarly, predicted-negative data-points that match with the labelled-negative data-points are classed as 'true-negatives'. True-positives and true-negatives are indications the algorithm is working as intended. Table 3-1 details the confusion matrix for identifying true-positives, true-negatives, false-positives, and false negatives determined by the predicted and labelled datasets.

Table 3-1: Confusion matrix of positive and negative data-points. *Data-points that have been correctly identified by the spike detection algorithm as belonging to an FP are classed as a true-positive. Correct identification of data-points not belonging to FP (i.e. the rest of the signal) are classed as true-negatives. Labelled-positive and labelled-negative data-points that are incorrectly identified by the spike detection algorithm are classed as false-negatives and false-positives, respectively.*

		Labelled	
		Positive	Negative
Predicted	Positive	*True Positive*	*False Positive*
	Negative	*False Negative*	*True Negative*

'False-negatives' occur when the spike detection algorithm has determined a data-point as negative but has actually been labelled positive during the manual detection. The missed detection of FPs is undesirable, but manageable as long as there are enough FPs captured to produce the required analytics. 'False-positives' occur when the spike detection algorithm has incorrectly identified data-points as belonging to an FP, when actually those data-points were labelled negative during the manual detection. The true-positive-rate (TPR) can be calculated using the number of true positives and labelled-positives from the manually-detected dataset (see Equation (3-15)) and likewise for the false-positive-rate (FPR) using the false-positives and labelled negatives (see Equation (3-16)).

$$TPR = \frac{True\ positives}{Labelled\ positives} = \frac{True\ positives}{True\ positives + False\ negatives} \quad \textbf{(3-15)}$$

$$FPR = \frac{False\ positives}{Labelled\ negatives} = \frac{False\ positives}{False\ postives + True\ negatives} \quad \textbf{(3-16)}$$

For a data-processing pipeline examining FP morphology, such as the one described in this thesis, false-positives are *extremely undesirable* and strict ruling should be applied to ensure the number of false-positives is kept as low as possible; false-positives (e.g. noise that has been classified as an FP) can interfere

with FP classification and measurement of morphology variability that takes place later in the data-processing pipeline (see Chapter 4). Using strict ruling can potentially lower the TPR, as less labelled positives will be correctly identified by the algorithm. However, this is seen as acceptable trade-off in order to keep the FPR as low as possible (i.e. to keep the specificity of the detection method high).

To form the ROC curve shown in Figure 3-30, the spike detection algorithm is run for a range of probability thresholds between 0 and 1. The ROC curve is run without the refine function (see Section 3.3.1.1) and so still includes isolated epochs and has no amplitude/polyphasicity cut off. The ideal ROC curve should be as close as possible to the top-left corner of the plot (FPR = 0, TPR = 1). Without the refine function, the ROC curve gives an area-under-curve (AUC) value of 0.947. The probability threshold that gives the smallest Euclidean distance to the top-left corner of the ROC plot is 0.95.

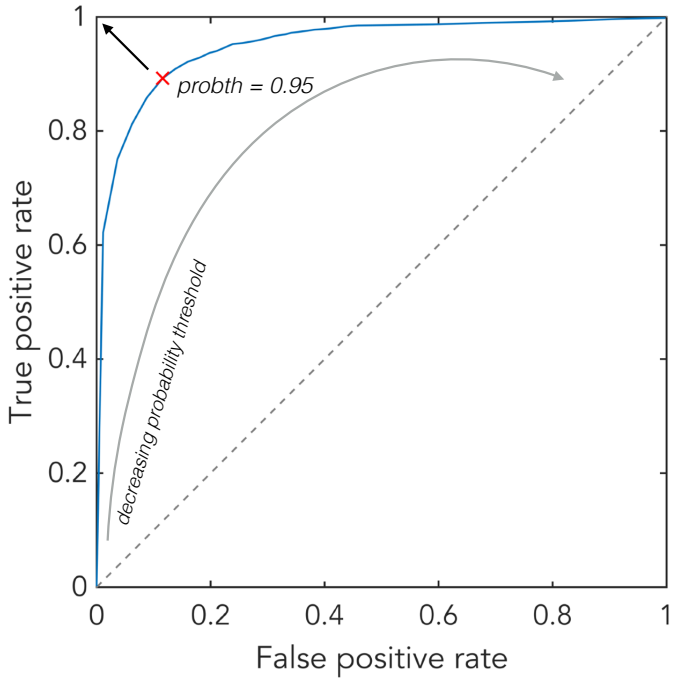
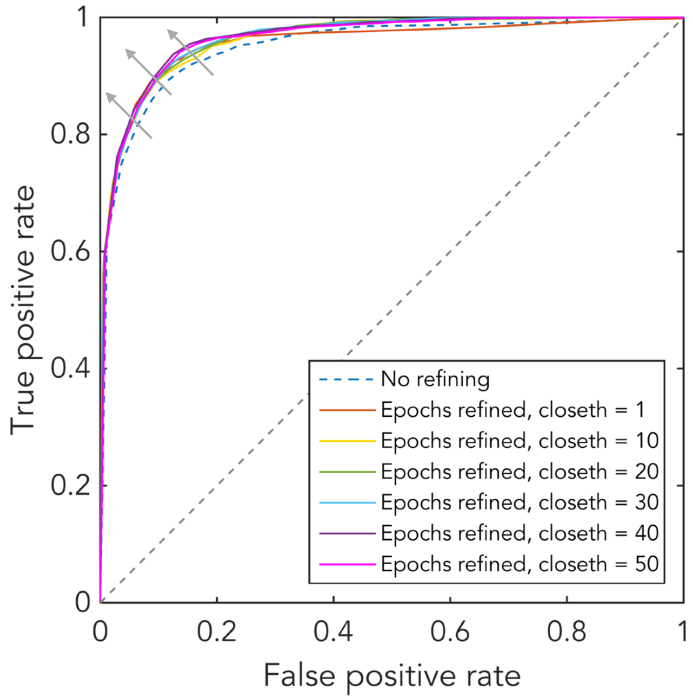


Figure 3-30: Receiver Operator Characteristic (ROC) curve comparing the False Positive Rate (FPR) and True Positive Rate (TPR) of the spike detection algorithm when compared to labelled data for a range of probability thresholds. *The threshold value with the smallest Euclidean distance to the top-left corner is 0.95. The ROC curve's area-under-curve (AUC) value is 0.9470.*

Applying the refine function to remove isolated epochs/conjoin epochs changes the ROC curve very slightly; the curve shifts slightly upwards towards the top-

left corner of the plot as the TPR has increased, as shown in Figure 3-31. This is likely to be due to the refine function conjoining epochs that are within the closeness threshold. Conjoining epochs that belong to the same FP will increase the number of true-positives that would have otherwise resulted in false-negatives. However, wrongfully conjoined epochs could potentially result in an increase in false-positives. Removing isolated, single-phase epochs that have passed the probability threshold, *probth*, with the refine function, caused by sudden fluctuations in the signal baseline, will decrease the number of false-positives. The ROC curve seen in Figure 3-31 uses the refine function to remove isolated epochs and conjoin epochs within closeness thresholds (*closeth*) of 1, 10, 20, 30, 40, and 50 data-points. The AUC values increase with the increase in *closeth* and peak at a threshold of 40 data-points ($AUC_{closeth40}$ = 0.9645) before decreasing again (see Figure 3-31).



Closeth	AUC
1	0.9514
10	0.9618
20	0.9626
30	0.9636
40	0.9645
50	0.9619

Figure 3-31: ROC curve with refined epochs. *The refining function has been applied to remove single-phase, isolated epochs threshold and conjoin epochs within a closeness threshold of each other. The resulting curves have improved the ROC curve with a general shift to the top-left corner of the plot. All the closeness thresholds tested gave a positive improvement to ROC values. A closeness threshold of 40 data points has produced the highest AUC value of 0.9645. The probability threshold value with the smallest Euclidean distance to the top-left corner for closeth = 40 is 0.96. The table on the right hand side of the figure details the AUC values for the different closeth thresholds.*

The user can also apply an amplitude threshold, *ampth*, in the refining function. Using an amplitude threshold can be a useful way of excluding small fluctuations in the signal that could have been potentially been accidentally mistaken for an FP. It is also a way to exclude neurological morphologies that could have possibly been detected by the sensor that the user can't confidently confirm whether or not they are FPs from the studied muscle (e.g. fibrilliations/PSWs, cross-talk of FPs from neighbouring muscles etc.). A range of amplitude thresholds (*ampth*) were applied, using *closeth* = 40, shown in the ROC curves in Figure 3-32. The left-hand ROC curves show a slight initial increase in TPR when *ampth* = 10 μV is applied, however as *ampth* increases the TPR of the ROC curves drops dramatically. This is due to an increase in the number of rejected data-points belonging to FPs while the number of manually detected labelled-positives remains the same. This causes the number of false-negatives to increase, thus explaining the decrease in TPR (see Equation (3-15) and Table 3-1). The right-hand plot in Figure 3-32 shows the ROC curves when the amplitude threshold is applied to not just the automatically detected spikes but to the labelled dataset as well. The FPR and TPR values continue to improve with increasing amplitude threshold; when *ampth* = 60 μV, the ROC curve shows that the spike detection algorithm gives a near perfect result ($AUC_{ampth60} = 0.9960$).

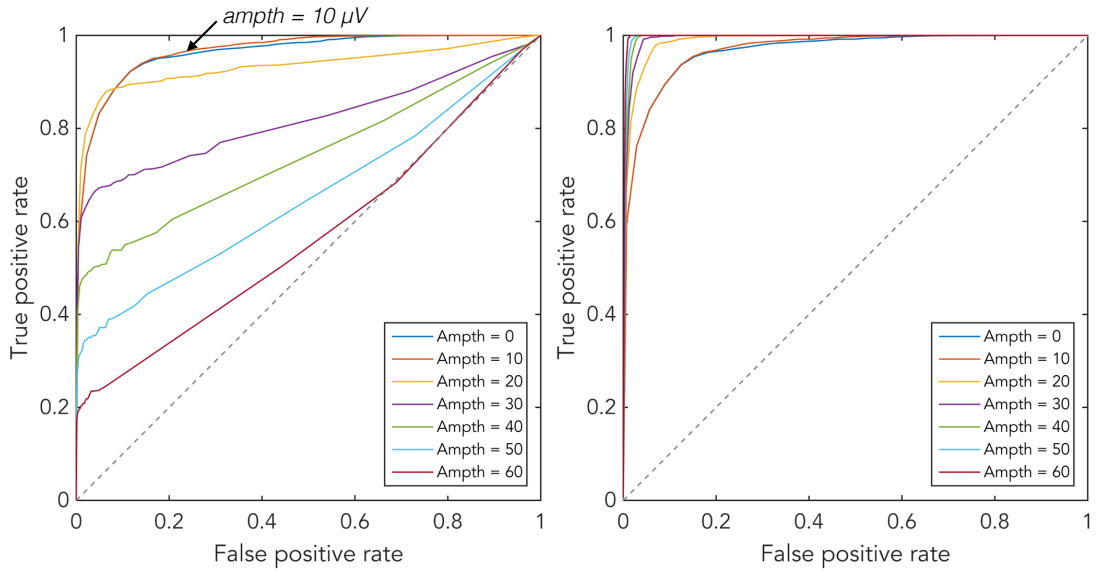


Figure 3-32: ROC curves for different amplitude thresholds. *(Left) The ROC curves for many different amplitude thresholds (ampth) applied in the refining stage. TPR initially increases when ampth = 10 μV, but decreases when ampth increases further; the more FPs that are excluded at the refining stage, the fewer true positives there can be when compared to the labelled dataset. (Right) An amplitude threshold has been applied to both the automatically detected FPs and the manually detected labelled dataset. FPR and TPR values improve considerably (ampth = 60 μV, AUC = 0.9960), however the number of FPs detected has dropped considerably at higher amplitude thresholds.*

However, the higher the amplitude threshold is, the greater the number of true FPs that are rejected from the recording, which could have been used to measure disease progression. For example, out of 276 manually detected FPs, just 22 have passed when using an amplitude threshold of *ampth* = 60 μV. The number of manually detected FPs that pass the amplitude thresholds used in Figure 3-32 are shown in Table 3-2.

Table 3-2: Number of manually detected FPs (labelled dataset) that pass through the amplitude threshold (ampth) at different threshold values.

Ampth (μV)	FPs
0	276
10	276
20	213
30	109
40	60
50	34
60	22

It is therefore logical to choose a lower amplitude threshold, such as 20 μV, so as not to exclude FPs that would have been included if detection were performed manually. The ROC curve for *ampth* = 20 μV still has improved FPR and TPR values (AUC = 0.9863) and includes 213 FPs that have an amplitude over 20 μV from the 276 FPs that were included in the original manually detected dataset. Figure 3-33 shows the improvement that has been made to the spike detection algorithm with the inclusion of the refining stage using the thresholds *ampth* = 20 μV and *closeth* = 40; the probability threshold closest to the top-left corner of the ROC plot is 0.95 (TPR = 0.9594, FPR = 0.0580). A higher probability threshold can be used if the user desires a lower FPR, however this will lower the TPR.[ii] When using these thresholds, the spike detection algorithm has detected 206 FPs, out of the original 276 FPs manually detected (of which 213 have an amplitude of 20 μV or greater).

[ii] *Low FPR is a desirable outcome from the algorithm, and potentially more important than achieving a high TPR. Low FPR ensures that no sections of signal are wrongly identified as FPs.*

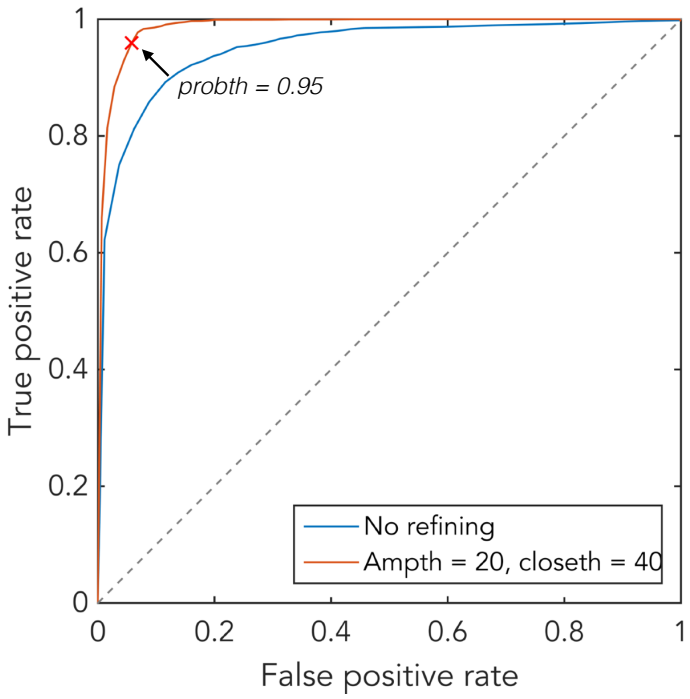


Figure 3-33: ROC curve for the spike detection algorithm for a range of probability thresholds. *The ROC curve has improved with the inclusion of the refining algorithm (ampth = 20 μV, closeth = 40), shifting closer to the top-left corner of the plot (AUC = 0.9863). FPs below ampth = 20 μV are also excluded from the manually detected labelled dataset. The probability threshold value with the smallest Euclidean distance to the top-left corner is 0.95.*

On closer inspection of the waveforms rejected by the detection algorithm using these thresholds (*ampth* = 20 μV, *closeth* = 40), it is possible they do not originate from fasciculations but from other spontaneous neuromuscular activity (and should be rightfully rejected). The morphology of the rejected waveforms in Figure 3-34 is similar to that of fibrillations/PSWs described by Mills[32] (see Section 1.4). However due to the low amplitude of a fibrillation/PSW, it is unlikely it could be detected in the time domain using surface EMG (whilst detectable using needle EMG) unless additional spectral signal analysis techniques were employed.[41] Another explanation for these rejected, repeated, low-amplitude waveforms is that they are of voluntary MU activity cross-talking from a neighbouring muscle; this is unlikely as their inter-spike-interval is greater than 250 ms (see Section 3.3.3). If these waveforms are truly FPs to be detected and analysed, it is likely they will be captured by a channel elsewhere on the HD-sEMG grid at a greater amplitude (see Section 3.3.2) and so will still be included in the data-processing pipeline.

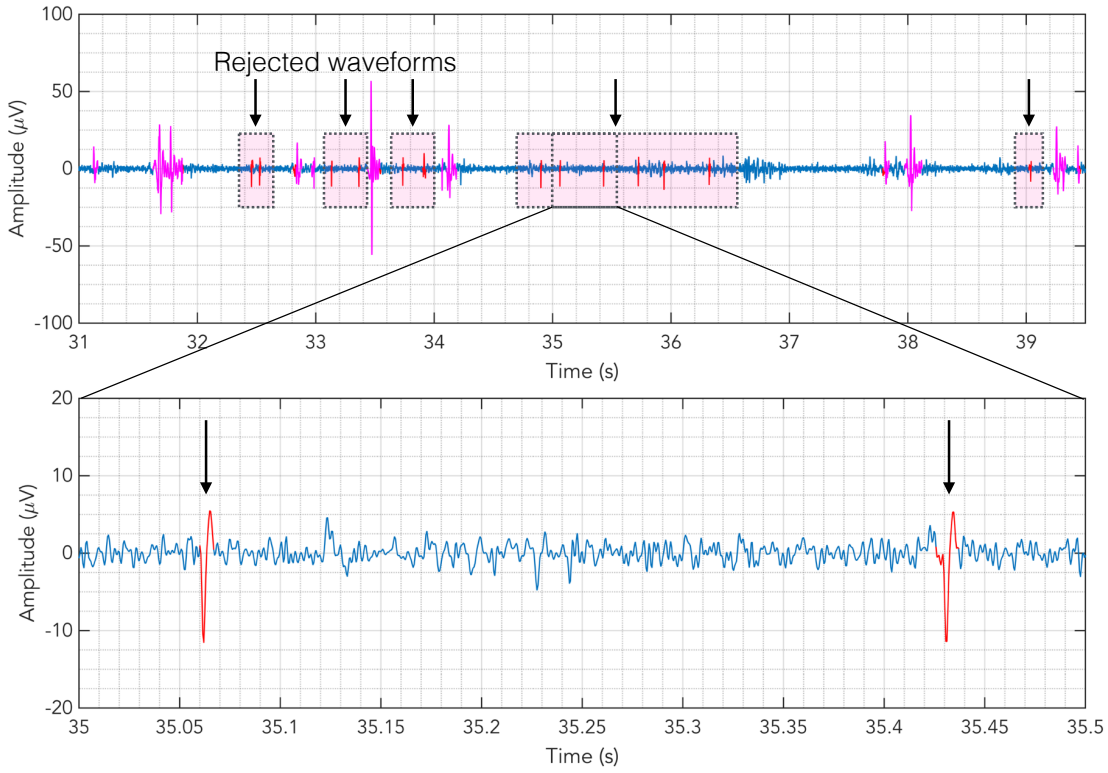


Figure 3-34: FPs detected and rejected using probabilistic spike detection algorithm. *FPs detected by the probabilistic spike detection algorithm (pink) and other low-amplitude neuromuscular waveforms that have been excluded by the refining stage (red) when using a probability threshold (probth) of 0.95, a closeness threshold (closeth) of 40 and an amplitude threshold (ampth) of 20 μV.*

3.3.2 Channel reduction

The muscle fibres of a motor unit can be dispersed throughout the entire muscle volume,[6] meaning that when a motor neuron discharges, its FP could be detected over the entire muscle area. Therefore, when using HD-sEMG grid arrays, the detected FP will not just be localised to a single channel but will spread across the majority, if not all, of the channel recordings. The FP will travel from the motor neuron, via the motor endplate (where the motor neuron innervates the muscle fibre) into the muscle fibres, and then towards the tendons at either the end of the muscle. The polarity of the FP will depend on the potential's direction of travel from the motor endplate relative to the electrode, as demonstrated in the diagram in Figure 3-35.

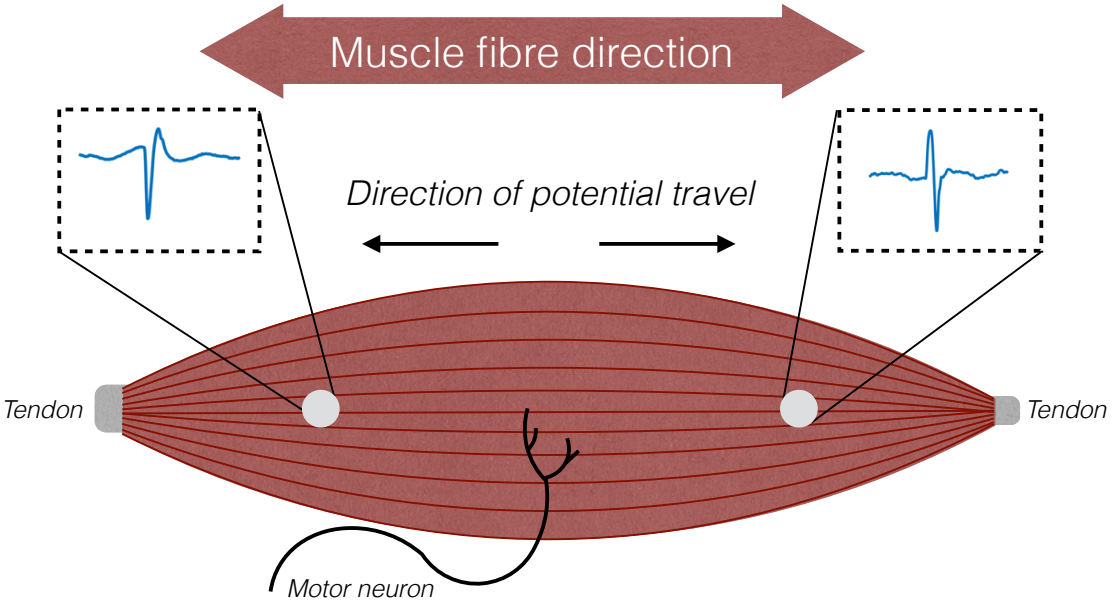


Figure 3-35: FP morphology changes depending on position on recording electrode. *The polarity and morphology of the FP will change depending on where the electrode is relative the motor endplate zone of the motor unit (i.e. where the motor neuron innervates the muscle fibres).*

Figure 3-36 demonstrates the distribution of an FP across the entire HD-sEMG grid. The morphology of an FP will change across the HD-sEMG grid, varying in amplitude and polarity. The polarity reversal seen between the channels in Figure 3-36 (made possible due to the average reference amplifier),[45] makes it possible to identify locations of the motor endplate zones.[22,40] Comparing the morphology of an FP as it is recorded in every single channel would be time consuming and computationally very expensive, and so the abundant amount of information that has been recorded needs to be reduced. The increase of signal amplitude across the grid can be explained by the increasing proximity between the bioelectric source and the electrodes.[40] Therefore, *only the waveform that has the greatest amplitude across all channels* of the HD-sEMG grid for each fasciculation is saved and taken forward in the data-processing pipeline. My rationale for this is that this waveform will be the most representative of a particular FP. The highlighted channel in Figure 3-36 has recorded the FP at the greatest amplitude, which is repeatable for the same fasciculating motor unit recorded at different time points in the recording.

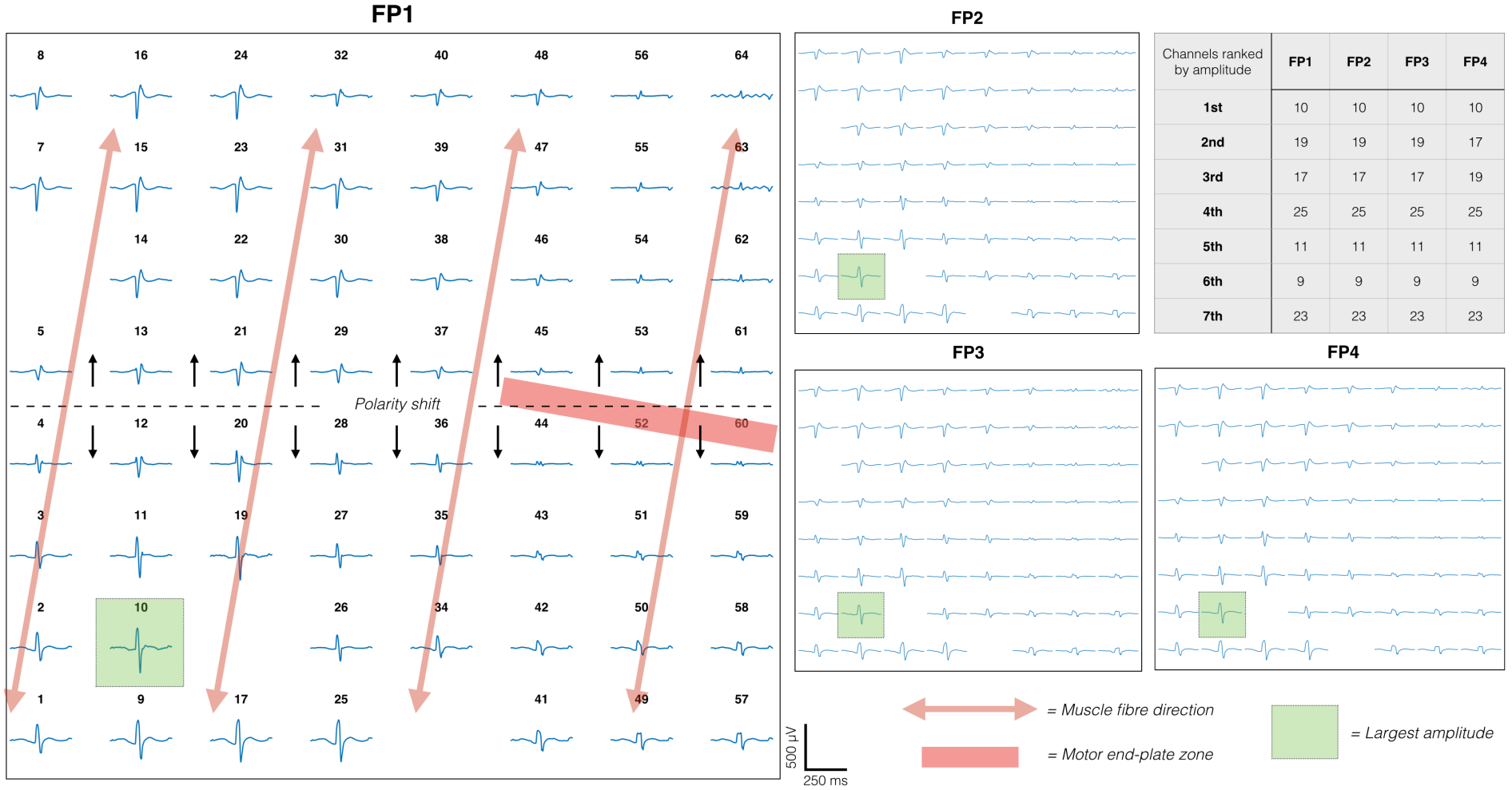


Figure 3-36: A fasciculation potential captured by HD-sEMG at four different time-points (FP1, FP2, FP3, FP4). *Depending on where the electrode is relative to the motor end plate zone (highlighted red region), the recorded FP for each channel will have a differing polarity, morphology, and amplitude. The muscle fibre direction (red arrows) can be determined by examining the change in amplitude and polarity of the waveform across the HD-sEMG grid. Channel 10 (highlighted green) has measured the FP at the greatest amplitude; this waveform is selected out of the 64 for further processing in the pipeline. For the same fasciculating motor unit, at the four different time-points shown, Channel 10 has consistently measured the FP at its greatest amplitude. The table (top-right) shows the channels in order of amplitude for FP1-FP4.*

The FP waveform at each time-stamp, found by the probabilistic spike detection algorithm described in Section 3.3.1, with the greatest amplitude across the 64-channels is extracted (e.g. Channel 10 in Figure 3-36). These representative waveforms for each FP are saved in a single time-series termed *'super-channel'*. The super-channel in Figure 3-37 demonstrates the extracted FP waveforms that have been saved amongst blank data where no FPs have been detected. The use of HD-sEMG grids for EMG studies has many benefits, however the abundance of channels can make visualising the recorded data difficult; storing the FPs in a 'super-channel' can make it easier for the user/clinician to view.

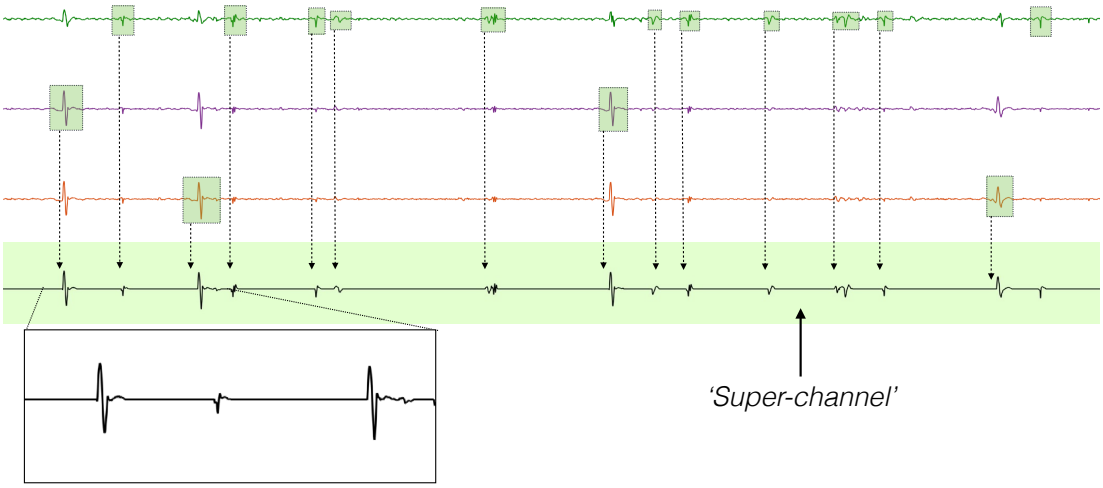


Figure 3-37: Example of forming the 'super-channel'. *The waveform with the greatest amplitude across the HD-sEMG grid for each fasciculation is extracted and saved into a single time-series called 'super-channel'. This example demonstrates this channel-reduction step extracting the largest waveforms from three separate channels and saving them into the super-channel (highlighted green). The data between FPs in the super-channel is left blank (zero), as shown.*

The number of FPs associated with each channel (noting only the channel where the FP amplitude is greatest) is recorded. Mapping this occurrence of fasciculations on the HD-sEMG grid is a useful procedure as the clinician can easily visualise if particular areas of the muscle have large amounts of activity. Comparing these occurrence maps with subsequent recordings can allow the clinician to view how the fasciculation activity changes and how it spreads over the muscle as the disease progresses, possibly observing the denervation and reinnervation of MUs.

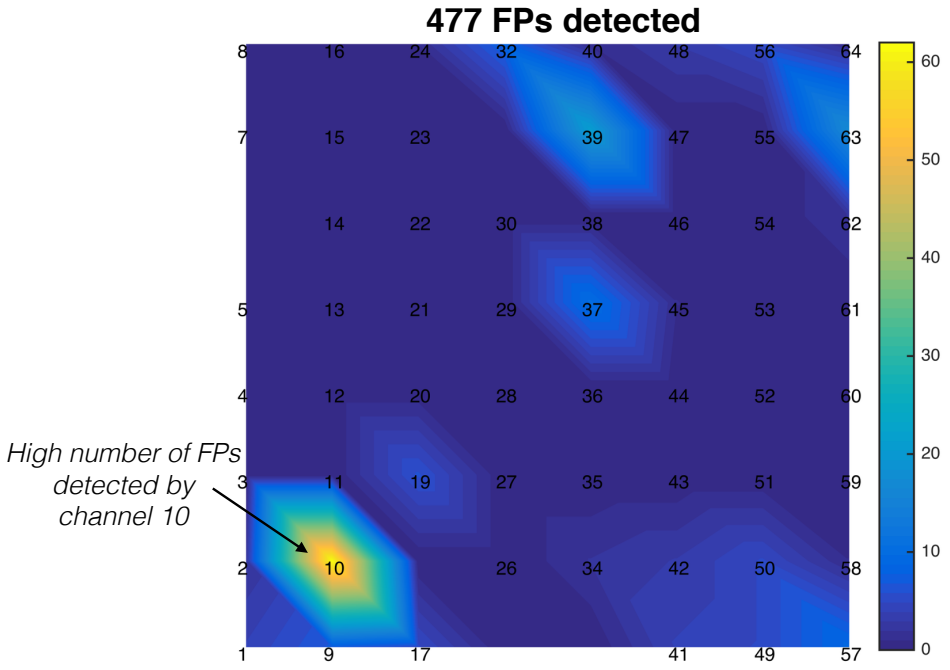


Figure 3-38: FP occurrence map. *The heat map highlights areas/channels of the grid where large amounts of fasciculation activity have been detected – where the amplitude for the FP was greatest. (Note that channels 6, 18, 25, 31, and 33 have been turned off).*

3.3.3 Voluntary motor unit activity

Remaining completely still with relaxed muscles for extended periods of time can be difficult for patients and can result in voluntary MU activity. A full voluntary muscle contraction is fairly simple to detect in an EMG time-series and will appear as a large burst of electrical activity, which is simple to remove. However, very low-level voluntary activity can occur even when the patient believes their muscles are fully relaxed and are not aware of any voluntary muscle activity. This voluntary muscle activity can occur in individual MUs and will appear on the EMG signal as a train of voluntary MUAPs. This can cause difficulties, as the waveform of a voluntary MU contraction can look very similar to that of a fasciculation. Voluntary MUAPs that are mistaken for FPs can disrupt the analytical output of the data-processing pipeline.

The lowest rate that a MU can fire at is 4 Hz (250 ms),[71] and so this is what is used as the criterion for identifying voluntary MU activity; when three or more MUAPs of matching waveform occur successively with an inter-spike-interval of less than 250 ms they are classed as voluntary MU activity and are removed.

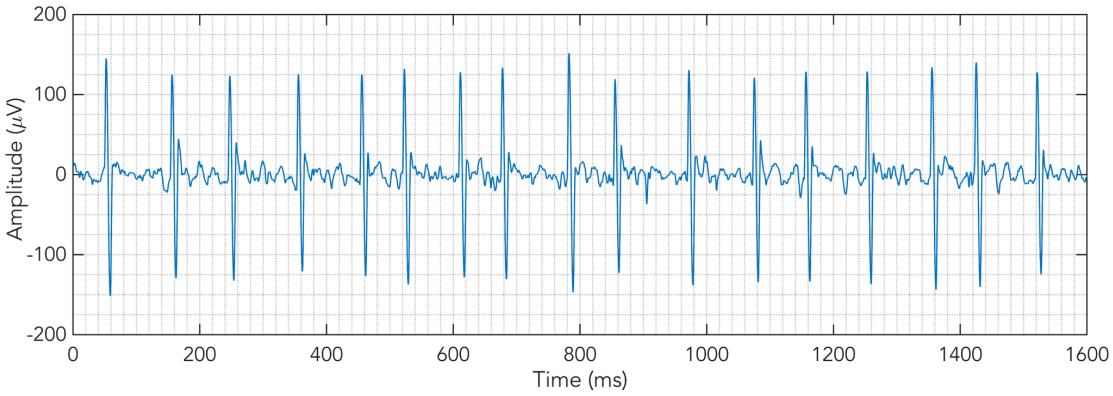


Figure 3-39: Voluntary motor unit activity that could be mistaken for FPs. *The slowest rate voluntary MUAPs can fire at is 4 Hz (inter-spike-interval of 250 ms); if three or more consecutive potentials with matching waveforms occur with an inter-spike-interval of less than 250 ms, these are classed are voluntary MU activity and should be excluded from the automatically detected FPs.*

The problem that occurs is that the pipeline cannot detect if the waveforms that occur successively within 250 ms of each other are matching (i.e. from the same MU) until they have been properly classified by the rest of the data-processing pipeline. One solution is to remove all potentials from the super-channel that have occurred within 250 ms of each other, regardless whether their waveforms match or not. This could, however, remove genuine FPs that have been detected; voluntary contractions can feasibly occur throughout the entire recording session, meaning the recording would be wasted. Another solution is to leave all suspicious waveforms in the super-channel, and once they have been matched/grouped together by morphology, the voluntary MUAPs can be clearly identified and discarded before any analytics are produced. However, as seen in Figure 3-39, voluntary MUAPs are *extremely abundant* compared to FPs; attempting to match all the voluntary MUAPs before removing them could take a large amount of computational effort.

The user therefore has the option to remove suspected voluntary activity based on the inter-spike-intervals of the detected waveforms and risk removing genuine FPs from the analysis, or to leave all detected waveforms in the data-processing pipeline and remove them after they have been grouped together but risking a potentially high computational load.

3.4 Conclusion of Part 1

This chapter has shown the opening steps of the data-processing pipeline. This chapter has covered how the EMG data is recorded from patients, the signal conditioning steps, noise removal, channel inspection, and detection of FPs using novel method of 'probabilistic spike detection'.

64-channel High Density-surface EMG (HD-sEMG) grid electrodes are used to record from ALS patients. Multiple channel EMG recordings can provide a wealth of information, which needs to be managed with efficient signal processing. The first stage in the signal processing is to condition the signal with a band-pass filter from 20-500 Hz; the low-pass filter removes low frequency artefacts such as motion artefacts caused by movement between the sensor and skin, the high-pass component removes unwanted high frequency components. Surface EMG is very susceptible to mains line noise (50 Hz) that is very difficult to remove without disturbing the underlying signal. A common method of 50 Hz noise removal is to use a notch filter. This method is computationally inexpensive, however it can distort the underlying signal as it completely removes the target frequency from the signal, resulting in an unwanted 'ringing' effect on the underlying signal. Several other methods of 50 Hz noise removal have been investigated but have had unsatisfactory results. The noise removal methods investigated either required specific tuning of parameters (thus not lending itself to an automatic process), placed too much computational load on the operating computer, or distorted/approximated the signal in a way that had unsatisfactory results. Empirical mode decomposition (EMD) could provide a way of reducing the levels of 50 Hz noise to a satisfactory level by isolating the harmonic components of the signal responsible for noise. However, this method places a very large computational load on the operating computer, making it unfeasible with current computing but could possibly be viable in the future. Despite any distortion it may cause, the 50 Hz notch filter is chosen as the method to remove mains line noise with due to its simplicity and computational efficiency; the levels of 50 Hz noise on an EMG recording can be so large that it can hide/mask FPs, and so unwanted distortions of the FP waveform are seen as

a necessary sacrifice to make in order to detect the presence of FPs at all. In certain cases, the levels of 50 Hz noise on the EMG recording may be low enough not to require any filtering. In these cases, it is recommended not to apply a notch filter to avoid any distortions this may cause the morphology of the FP. Of course, the best solution would be the prevention of 50 Hz noise in the first place; a solution for this is discussed in Chapter 7.

Channels are inspected for disconnection or poor skin-electrode contact. A new measure termed channel '*activity*' was created; channels with abnormally high levels of *activity* are removed from the data-processing pipeline. Channel activity is calculated by the sum of absolute values (area under the curve) of the EMG signal's Fourier spectrum (calculated using the discrete Fourier transform). Channels with a poor connection will have an abundance of additional frequencies on their signal, giving a high measure of *activity* relative to channels with a good quality recording. Disconnected channels will record a constant offset value, however when band-passed this will result in a large ringing effect at either end of the signal and will also give a high measure for *activity.*

The final stage of the data-processing pipeline uses a novel approach to detecting FPs termed 'probabilistic spike detection'. The method provides a rational basis for selecting a threshold for spikes, depending on factors such as grid, electrode, skin, gel, patient, and operator, rather than selecting an arbitrary amplitude threshold without any insight to the signal's quality or behaviour. The method uses all of the values in the signal to form a probability function of the likelihood a value will occur. Each 'epoch' of data (period of signal between each zero-crossing point) is assigned a probability value based on the maximum/minimum value in that epoch. Epochs over a certain probability threshold are conjoined together if they are adjacent to one another, or within a distance threshold of one another, to form the polyphasic waveforms of a FP. The performance of the probabilistic detection method has been tested and refined using real ALS patient data, from which FPs had been manually detected.

The number of channels in the dataset is consolidated into a single *'super-channel'*, which simplifies the forthcoming analysis in the data-processing pipeline and makes it easier for the user/clinician to view the dataset. The super-

channel is formed by first comparing the onset/offset times of FPs detected across the HD-sEMG grid, and then by extracting the waveform from the channel that has expressed the FP at its greatest amplitude for the given time interval. Once the super-channel is formed, the inter-spike-interval (ISI) between spikes can be checked for voluntary muscle activity; an ISI of 250 ms or less is likely to be due to a voluntary contraction of a MU. Voluntary MUAPs are not of interest in the investigation into FP variability and so should be discarded. The risk of removing the voluntary MUAPs at this stage is that legitimate FPs could accidentally be removed as well. Voluntary MUAPs can be removed at the end of the pipeline, once they have been grouped by their waveform and isolated from the FPs detected in the EMG recording. However, as voluntary MUAPs can be abundant, their presence can put an unnecessary computational load on the rest of the data-processing pipeline.

The next chapter, Data Processing Pipeline Part 2, describes the latter half of the data-processing pipeline in which the processed FPs are grouped by their morphology. Once the FPs are grouped, they are aligned and averaged to form templates of the FPs found in the EMG recording.

Chapter 4
Data Processing Pipeline Part 2

In the previous chapter, Data Processing Pipeline Part 1, I described the methods and processes used in order to capture the data, condition the signal, and detect the presence of FPs in the EMG signal. In this chapter, I describe the latter half of the data-processing pipeline: grouping the FP waveforms by their morphology.

4.1 Grouping Fasciculation Potential Morphologies

Now that the FPs have been detected and extracted from the recordings, the next stage in the data-processing pipeline is to group them by their morphologies. As stated in my hypothesis at the beginning of the previous chapter, different fasciculating motor units (MUs) will express their own individual waveform morphologies. By grouping FPs by waveform, and thus by MU, a measure of the FP variability can be produced for each MU. The grouping of FP morphologies follows the steps shown in Figure 4-1: 'clustering' of FPs based on specific features, examining the similarity of FPs by their morphology using a method calling 'matched filtering', the alignment of grouped FPs, and finally creating of templates from the averages of aligned FPs.

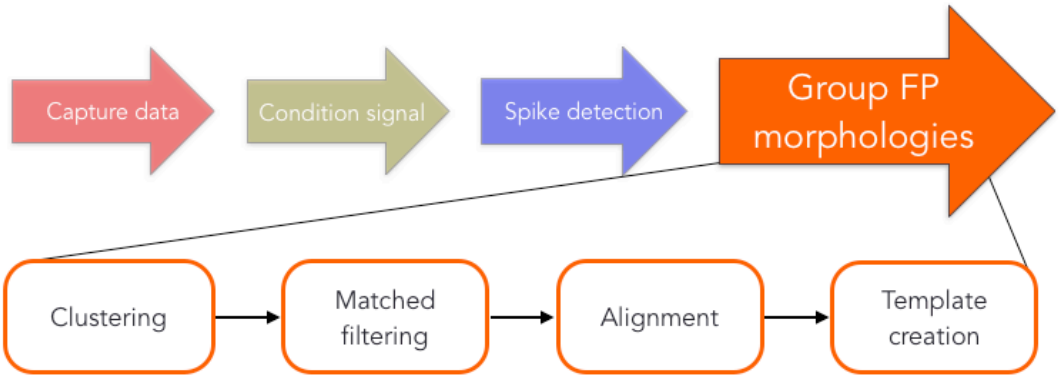


Figure 4-1: Grouping FP morphologies flowchart. *Steps taken in grouping FPs by their morphology: clustering, matched filtering, alignment of grouped FPs, and template creation.*

4.1.1 Clustering

Clustering is a powerful machine-learning tool for the classification/grouping of datasets into smaller groups of 'clusters'. Clustering is an 'unsupervised' machine-learning method, meaning it does not rely on labelled data in order to classify/group data-points. Instead, clustering relies only on the input dataset and the similarities within it from which to draw inferences from. Clustering is a useful explorative tool to use in data analysis when investigating underlying patterns/groupings in a dataset that is unknown to the user, in the absence of *a priori* knowledge. Clustering algorithms will 'cluster' data into groups based on the proximity of their locations in space, as shown in Figure 4-2.

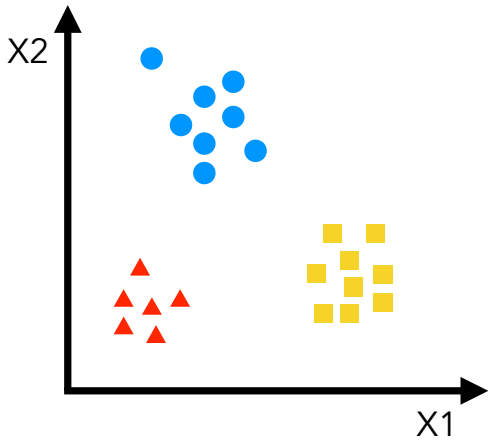


Figure 4-2: Example of clustered data. *The scattered data has been clustered into three groups based on the proximity of their locations in parameter space.*

Clustering is a useful technique for grouping MUAPs/FPs by their morphology, where the user has no *a priori* information of what waveforms the EMG signal contains or how many motor units are firing/fasciculating. Clustering is a

technique that has been used extensively in literature for the grouping of MUAPS and FPs from EMG recordings.[22,34,36,40,43,53,58,72] Two types of clustering that could be used are 'soft' clustering and 'hard' clustering.

Soft clustering

Soft clustering methods such as 'fuzzy c-means' clustering allow data-points to belong to multiple data clusters with varying degrees of membership. For example, one cluster may be 80% responsible for a data-point, and another cluster responsible for the remaining 20%. This type of soft assignment is useful when analysing datasets known to have datapoints that can be shared across several class types e.g. gene features can be soft clustered, as they can be responsible for several different biological processes. Fuzzy c-means clustering aims to minimise the value of the objective function J (see Equation (4-1)), based on the squared distance between each data-point x_j and cluster centre c_i, where D is the number of data-points, C is the number of cluster centres, and μ_{ji} is the membership value of each data-point to each cluster centre. The overlap between clusters is determined by the fuzzy partition coefficient m (where $m > 1$). Fuzzy c-means clustering is an iterative method, with μ_{ji} being updated each iteration (see Equation (4-2)) until J cannot be minimised any further or until an iteration limit is reached. Figure 4-3 demonstrates two clusters of data clustered using fuzzy c-means clustering. Multivariate Gaussians are provided to show the distribution of data and covariance of each data cluster.

$$J = \sum_{j=1}^{D} \sum_{i=1}^{C} \mu_{ji} \left\| x_j - c_i \right\|^2 \quad \textbf{(4-1)}$$

$$\mu_{ji} = \frac{1}{\sum_{k=1}^{N} \left(\frac{\left\| x_j - c_i \right\|}{\left\| x_j - c_k \right\|} \right)^{\frac{2}{m-1}}} \quad \textbf{(4-2)}$$

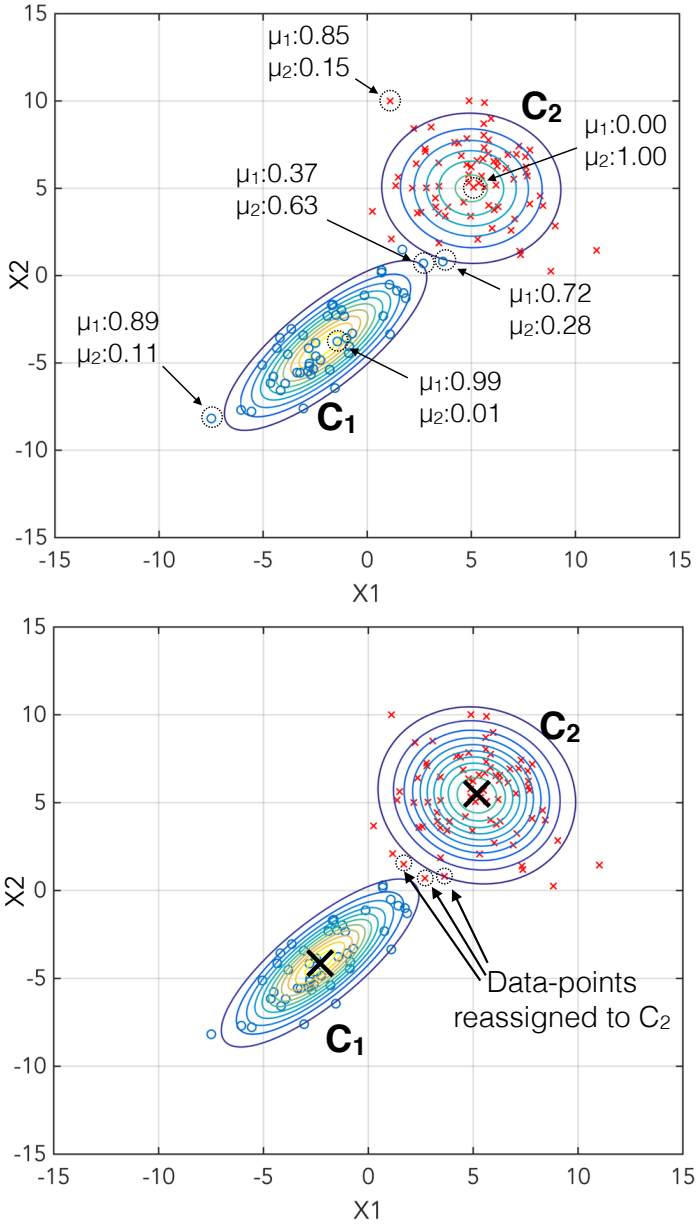


Figure 4-3: Fuzzy c-means clustering example. *Data clustered using fuzzy c-means clustering (m = 2). Some example membership values are labelled (top) to demonstrate how μ varies across the clustering space. The labelled data-points have been reassigned (bottom) from C_1 to C_2 based on the membership values the algorithm has calculated in order to minimise J. Multivariate Gaussians are provided to show the distribution of data and covariance of each data cluster.*

Hard clustering

Hard clustering methods, such as K-means clustering, only allow data-points to be assigned to a single cluster centre. Hard assignment is useful when analysing datasets in which the user knows data-points that can, or must, be strictly assigned to one specific class type. In the K-means clustering algorithm, the number of clusters, *K,* is chosen and each data-point is assigned to the nearest

cluster centre (see Figure 4-4). The algorithm then works iteratively to minimise the total squared Euclidean distance E between each data-point x_j and its assigned cluster centre c_i (see Equation (4-3)). As K-means is a hard clustering method, the membership value of each data-point p_{ij} can only be equal to 0 or 1 (p_{ij} = {0,1}, see Equations (4-4) and (4-5)). These hard membership values mean that only the squared distance between each data-point and its assigned cluster centre will contribute to the total value of *E:*

$$E = \sum_{j=1}^{D} \sum_{i=1}^{K} p_{ji} \left\| x_j - c_i \right\|^2 \quad \textbf{(4-3)}$$

$$p_{ji} = 1, if\ x_j\ assigned\ to\ c_i \quad \textbf{(4-4)}$$

$$p_{ji} = 0, else \quad \textbf{(4-5)}$$

Data-points are assigned to the nearest cluster centre as to minimise *E.* The cluster centre locations are updated on each iteration by taking the average of its assigned data-points. As the cluster centre locations are updated, data-points may be reassigned to different cluster centres, as shown in Figure 4-4. This method iterates until the cluster centre locations stop changing:

1. *Assign each data-point to the closest cluster centre.*
2. *Move each cluster centre to the centre of the data-points assigned to it.*
3. *Re-assign data-points closest to new cluster centre locations.*
4. *Iterate until cluster centre locations stop changing.*

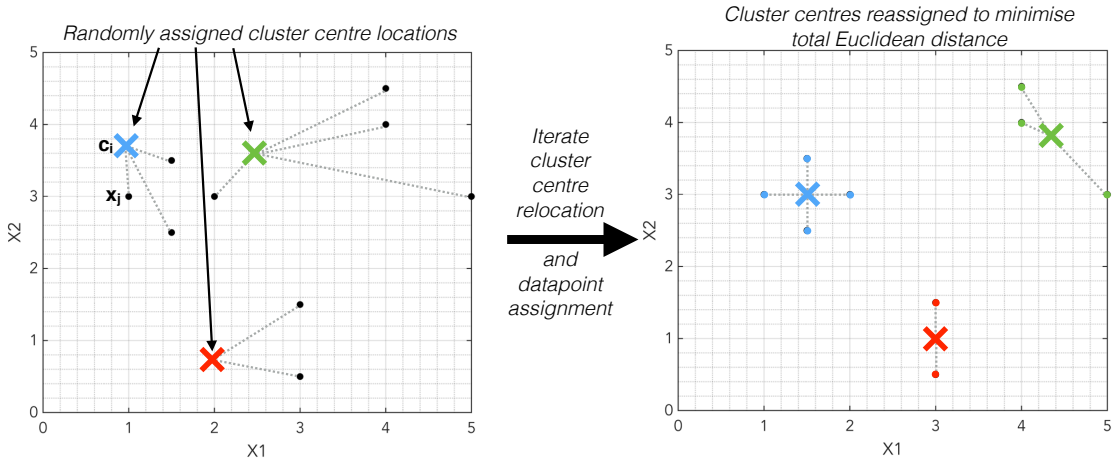


Figure 4-4: Hard K-means clustering example. *Each data-point is assigned to the cluster centre closest to it (Euclidean distance). For example, the data-point x_j is assigned to the cluster centre c_i. Each cluster centre location is moved to the centre of its assigned data-points, and then data-points are reassigned to the closest new cluster centre locations. The process is iterated until the cluster centre locations stop changing.*

In this data-processing pipeline, a hard clustering approach is taken; FPs require 'hard' classification, rather than being assigned a weighting/membership value, in order to clearly separate the different waveform morphologies that can appear in a single patient EMG recording.

4.1.1.1 Feature extraction

At this stage of the pipeline, the FPs at present are still stored in the full complexity of their waveforms. In order to simplify their complexity, certain 'features' of the FP are extracted from each waveform. It is these features that are treated as having a location in space during the clustering process. For example, the *amplitude* and *area* (sum of absolute values) of an FP are two defining features of the waveform that can be used to cluster the data in two dimensions; as shown in Figure 4-5, each FP will be represented as a location in two-dimensional space depending on its amplitude and area.

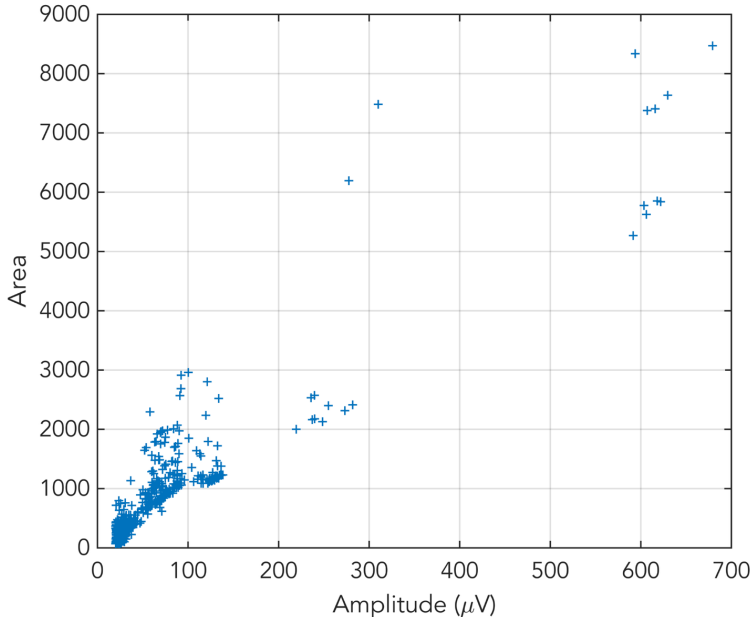
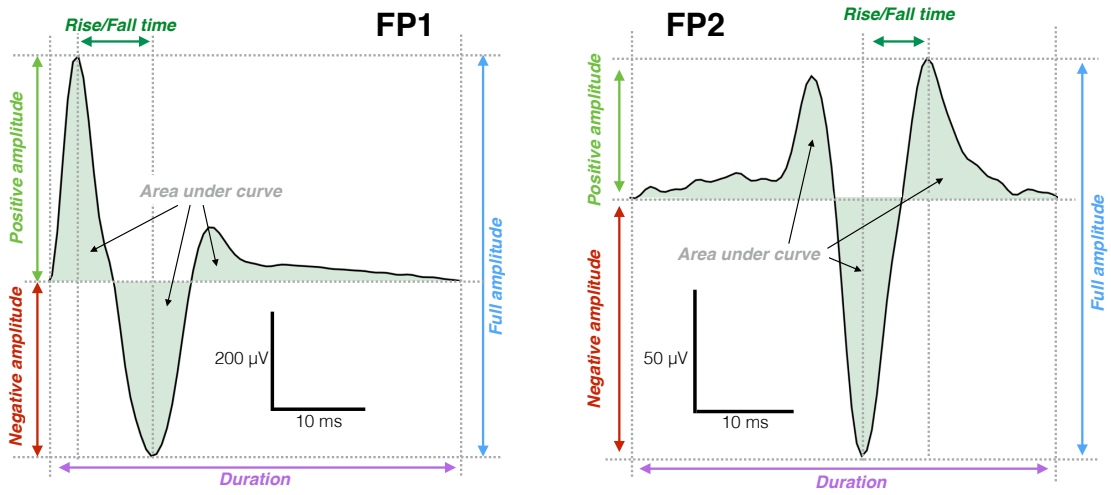


Figure 4-5: Scatter plot of amplitude and area. *The two features 'amplitude' and 'area' have been extracted from the database of FPs, previously detected in the data-processing pipeline, and are scattered against each other.*

However, there are many different features that could be extracted from an FP to describe the complexity of its waveform. The characteristic features extracted from the FP, as shown in Figure 4-6, are as follows:

1. *Amplitude*
2. *Duration*
3. *Area under curve (sum of absolute values)*
4. *Positive amplitude*
5. *Negative amplitude*
6. *Rise/Fall time (time from max positive/negative value to max negative/positive value)*
7. *Positivity. Refers to the ratio of the positive amplitude to the negative amplitude*



	FP1	FP2
Full amplitude	605.9 µV	159.2 µV
Positive amplitude	337.5 µV	54.5 µV
Negative amplitude	-268.4 µV	-104.7 µV
Duration	37.6 ms	41.5 ms
Rise/Fall time	6.8 ms	6.4 ms
Area under curve	5698.1 ms.µV	1813.1 ms.µV
Positivity	1.26	0.52

Figure 4-6: Extracted features from FPs. *Amplitude, duration, area under the curve, positive amplitude, negative amplitude, and the time between the waveform's maximum and minimum points (rise/fall time) are all extracted from the two FP waveforms shown. Another feature termed 'positivity' is calculated as the ratio between the positive and negative amplitude, for example FP1 = 1.26 and FP2 = 0.52.*

A key feature of an FP's waveform is its number of phases; as the disease progresses FP morphology will become increasingly polyphasic.[23] Despite this, *phases* are not extracted as a feature from the FPs. As the number of phases will be a discrete integer (unlike *amplitude* or *duration*), this causes the data-points when clustered to form very definite clusters in space. While this may not immediately appear to be a problem, an issue arises from how phases are defined: sections of the waveform between zero-crossing points. Stochastic noise on the signal can easily affect where and how many times it crosses zero, leading to a false number of phases and thus incorrect clustering.

4.1.1.2 Polarity

Before the clustering process can begin, the detected FPs need to be split into two separate groups depending on their polarity; whether the waveform leads with a dominant positive or negative peak. The reason for this is that FPs originating from different motor units can sometimes have very similar waveforms, and thus have very similar features, but be of opposite polarity (see Figure 4-7). The FP's polarity is due to the location of the recording electrode relative to the position of the MU's innervation zone (see Section 3.3.2). The polarity of an FP is determined in this data-processing pipeline by whether the leading phase (first dominant phase) of the waveform is positive or negative; the leading phase is the first phase of the waveform to have an amplitude that exceeds 33% of the total FP amplitude.

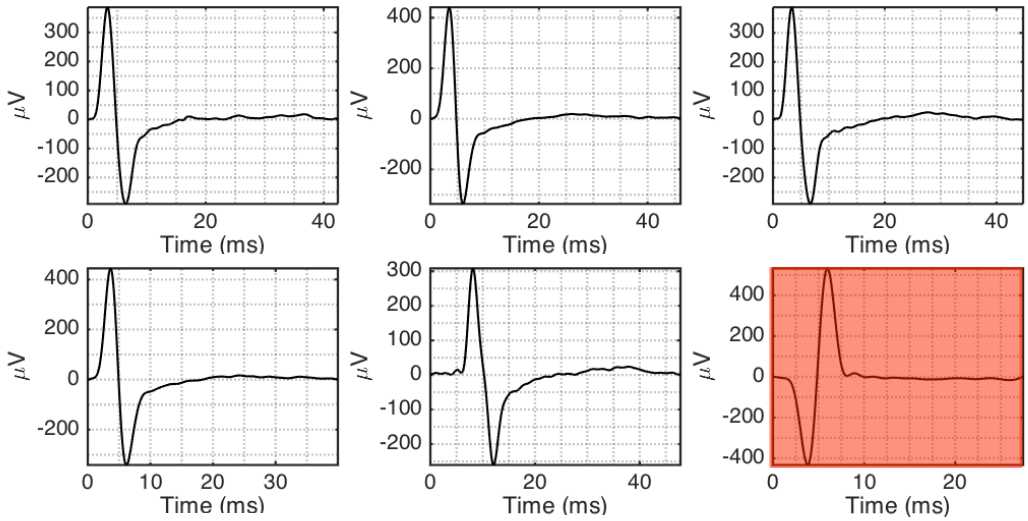


Figure 4-7: Splitting FP polarity before clustering. *The highlighted FP (red) has very similar features/waveform morphology to the non-highlighted FPs despite them having opposite polarities. FPs must be split by their polarity in order to prevent incorrect clustering from occurring. (Image adapted from MRes phase of project; Wickham 2015, Department of Bioengineering, Imperial College London).*[60]

Once the FPs are split by their positive/negative polarity they are treated as separate datasets for the rest of the grouping stages in the data processing pipeline. The polarity split of FPs precedes the Principle Component Analysis (PCA) (see Section 4.1.1.3) and clustering, as shown in Figure 4-8.

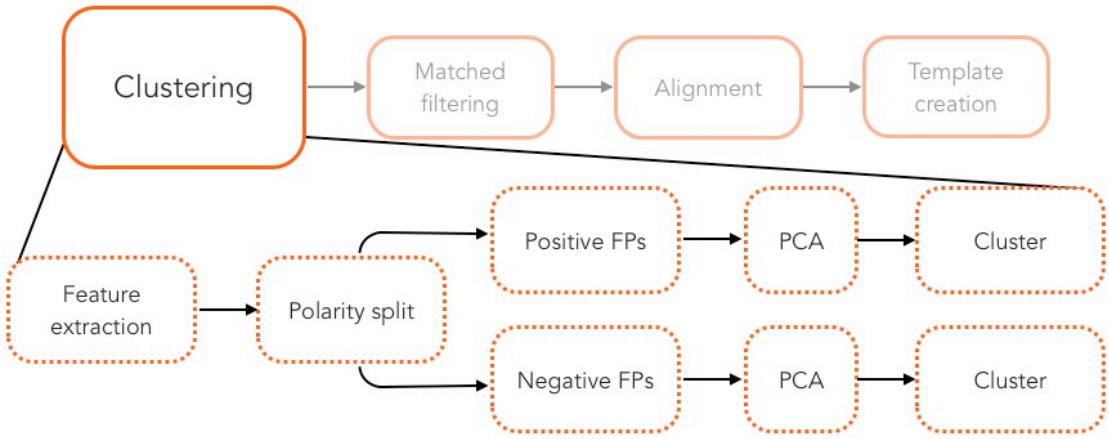


Figure 4-8: Clustering flowchart. *Features are extracted from the detected FPs to simplify their complexity. The FPs are then split by polarity into two sub-groups called 'Positive FPs' and 'Negative FPs'. These sub-groups are kept separate for the remainder of the grouping stages in the data processing pipeline.*

4.1.1.3 Principle Component Analysis

In datasets that have many variables (e.g. detected FPs and their extracted features), it can be difficult for the user to understand the relationships between variables and how they affect each other. In datasets with two or three variables the user could view this relatively easily in 2-D or 3-D space (i.e. *xyz* space), however in datasets where the number of variables extends well beyond this (e.g. >100s of variables), it can quickly become very difficult to visualise these relationships.

Principle Component Analysis (PCA) is a dimensionality reduction algorithm that aims to reduce the dimensions (numbers of features) of a dataset. PCA is a useful way of viewing the relationships between features and removing redundant information. PCA does not discard any of the extracted features, but instead constructs new variables called 'principle components'. The number of principle components (PCs) produced by PCA will be as large as the original number of features, however just the first few PCs will often describe the majority of the variance in the data (often describing >80% of variance). This makes the dataset less complex, as there could be potentially thousands of variables/features to cluster the data with. To demonstrate the use of PCA, the ALS patient EMG recording shown in Figure 4-9 is used. FPs are detected using the probabilistic spike detection algorithm described in Section 3.3 and the

features listed in Section 4.1.1.1 are extracted. The FPs are split by their polarity and the 'positive FPs' (see Figure 4-8) are used in this demonstration. The *pca* function in MATLAB is used to perform the PCA.

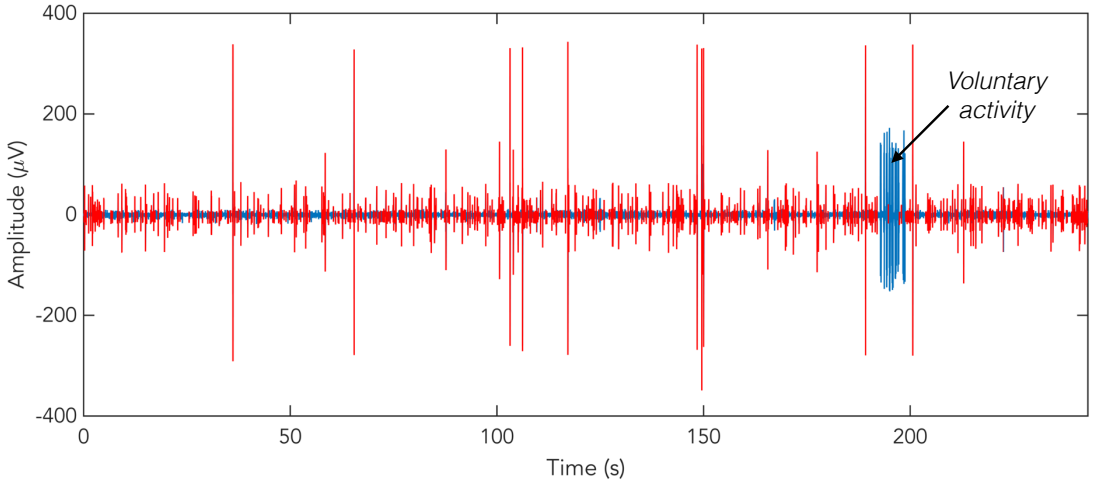


Figure 4-9: ALS patient data used to demonstrate principle component analysis. *The EMG recording (blue) is very abundant with FPs (red) that have been detected using the probabilistic spike detection algorithm. The features extracted from each FP are: amplitude, positive amplitude, negative amplitude, duration, rise/fall time, area under the curve, and positivity.*

PCA will linearly transform the data from the highly dimensional space to the principle component space, as shown in the biplots in Figure 4-10. The vectors in the biplot show how strongly the features have contributed to each principle component, on top of the scattered, linearly transformed data. It can be seen that the features *amplitude*, *max pos, max neg,* and *area* all contribute significantly to the first principle component (PC1); the features *fall/rise time* and *duration* contribute significantly to the second principle component (PC2); and the feature *positivity* contributes the most to the third principle component (PC3). The scree plot at the bottom of Figure 4-10 shows that 99.4% of the variance in the data is explained by the first four principle components, with the first three principle components (PC1, PC2, PC3) explaining 89.9% of the variance.

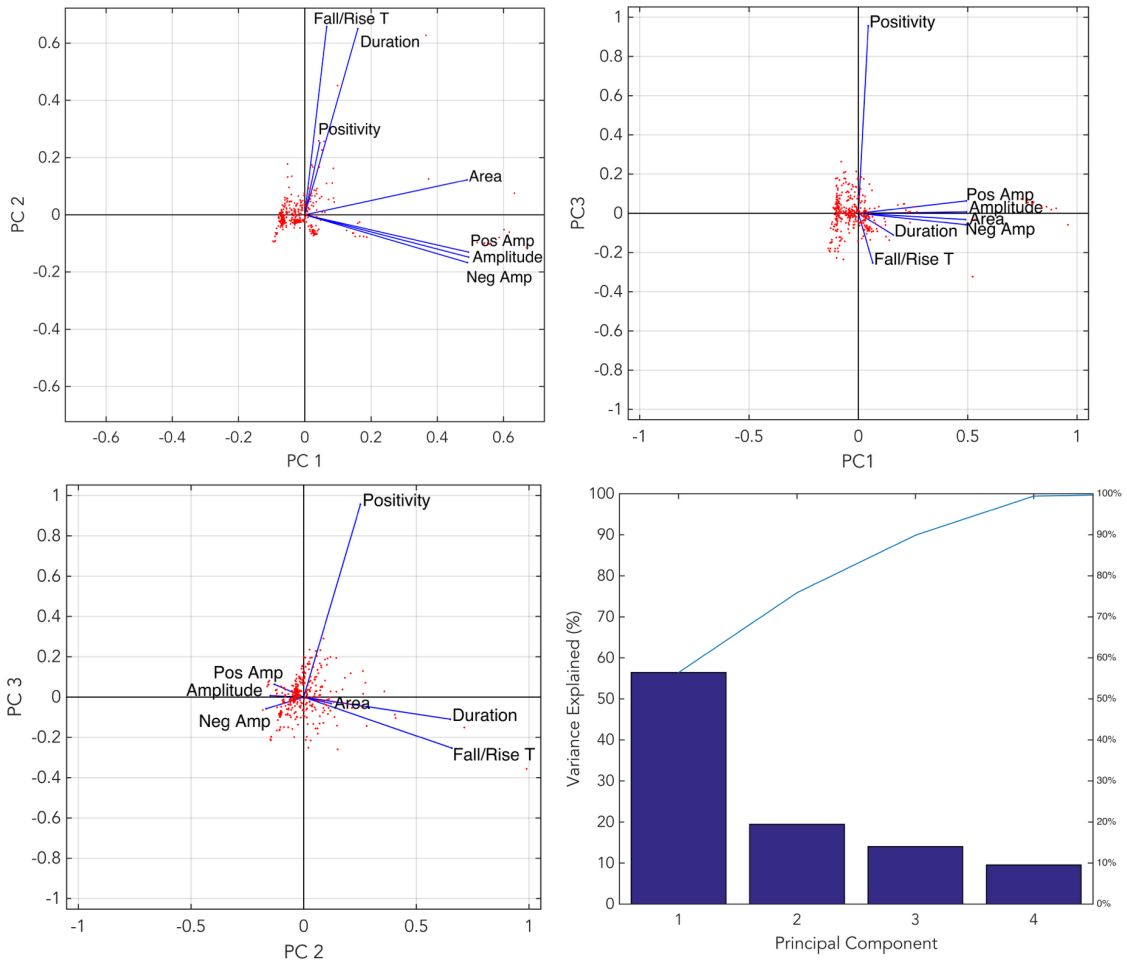


Figure 4-10: PCA of features extracted from FPs. *The data (scattered red) has been linearly transformed by PCA, as shown in the biplots of PC1 vs. PC2 (top-left), PC1 vs. PC3 (top-right), and PC2 vs. PC3 (bottom-left). The vectors (blue) show each feature's contribution to the orthogonal PCs, and the relationships and correlations between them. The scree plot (bottom-right) shows how just the first three PCs are used to explain 89.9% of the variance in the data.*

The position of the vectors can provide more information about the relationships between features; feature vectors in very close proximity, forming a small angle, will have a close relationship and a positive correlation, vectors at 90° to one another will likely show no correlation, and vectors close to 180° will be negatively correlated. For example, the vectors for *amplitude* and *positive amplitude* show a strong correlation (r = 0.997), as do the vectors for *amplitude* and *area under the curve* (r = 0.918), whereas *negative amplitude* and *positivity* show close to no correlation at all (r = -0.018), as shown in Figure 4-11.

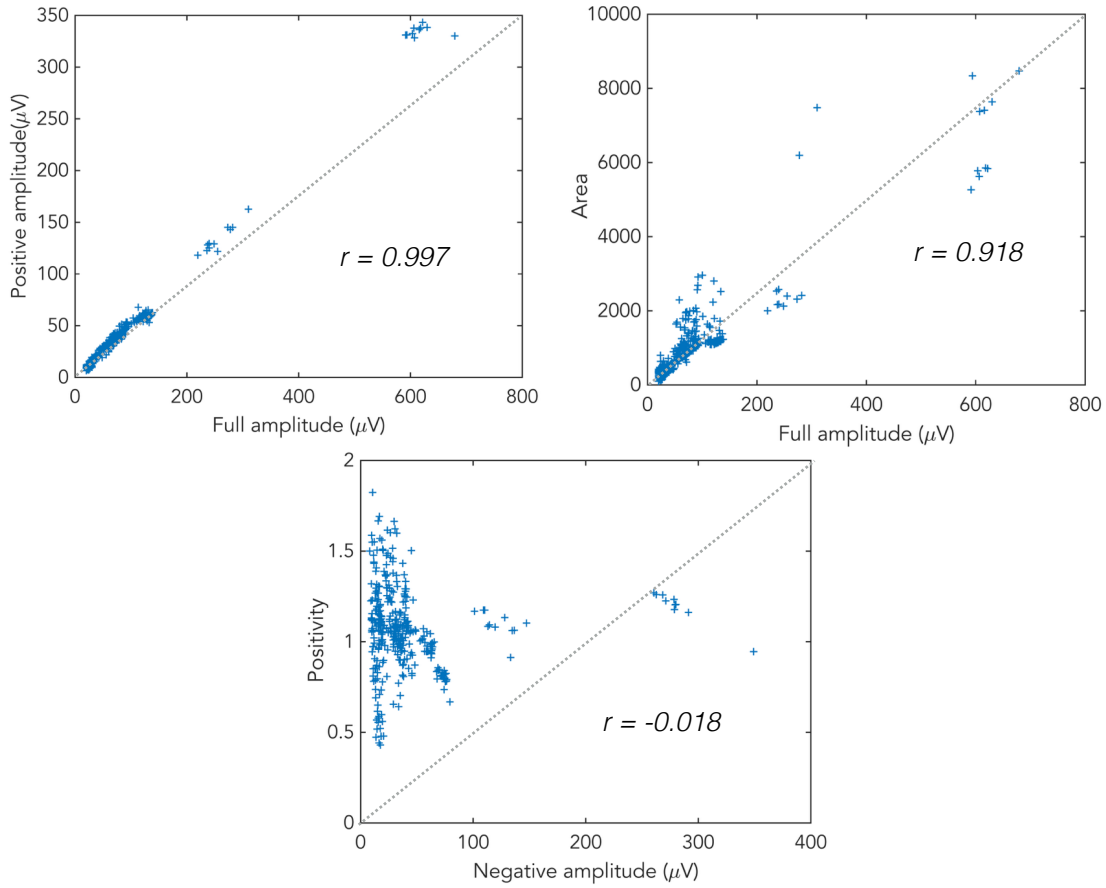


Figure 4-11: Correlation between different features. *PCA has helped to highlight the data that has a strong correlation, such as amplitude and positive amplitude (top-left) and amplitude and area (top-right), and data that does not, such as negative amplitude and positivity. This analysis is useful when investigating if features are dependant on one another or not, and if there is any redundant information in the dataset.*

If one was not to perform PCA, then the clustering process in the next stage could be a very inefficient process, as the user would have to investigate the relationship between each feature themselves. For example, if the features extracted from the 'positive FPs' were to be *amplitude* and *positive amplitude,e* their close relationship would lead to ineffective clustering. Instead, by producing the principle components of the data, the variance of the dataset can be better explained and will result in a more efficient clustering process. Figure 4-12 shows the scattered principle components of the dataset, and the results of running the clustering process using the *kmeans* function in MATLAB.

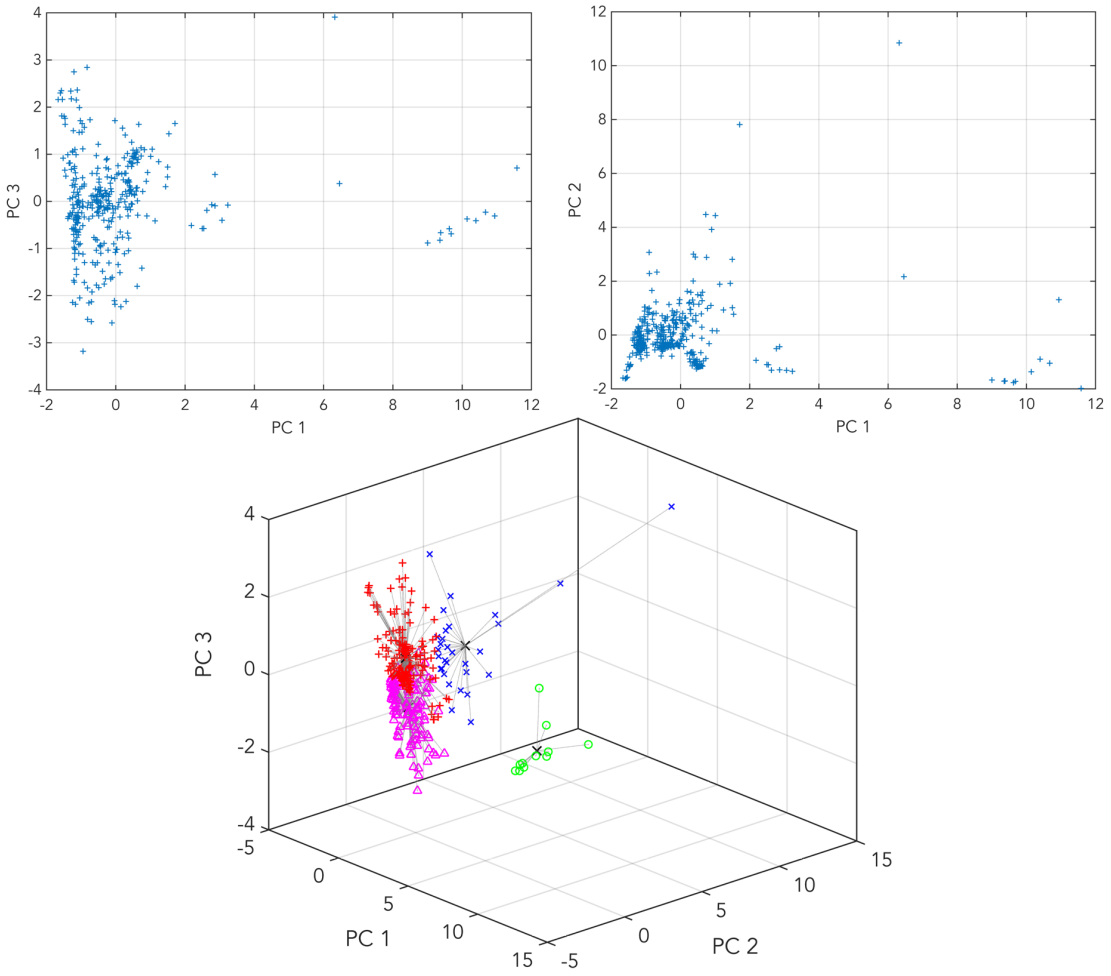


Figure 4-12: K-means clustering of principle components from dataset. *The scattered principle components of the dataset explain a large majority of the variance in the dataset. Using k-means clustering, with four cluster centres (k=4), data has been clustered together.*

4.1.1.4 'Cluster and seed' framework

One common problem when using clustering algorithms is that it requires the user to have prior knowledge of how many classes, or cluster centres (k), to expect in the dataset. Often the user will either have this *a priori* information, by visualising the data in space, or by performing the clustering algorithm, assessing the outcome, and then re-clustering the data with additional cluster centres.[43] This "semi-automatic"[22] approach to clustering is necessary when attempting to cluster MUAPs/FPs from EMG signals as the user will have "no *a priori* knowledge of the MUAP classes composing the EMG signal".[72] In this data-processing pipeline, the preference is to not have to resort to using human-intervention; the grouping of FP waveforms should be as automatic as possible, without the user having to pre-visualise the data or reiterate the clustering

process themselves. In a longitudinal study, such as the one conducted at King's College London (KCL), it could be possible to use the FP dataset from a previous recording session to provide the *a priori* knowledge needed to cluster the FP datasets of successive recordings. However, as these are EMG recordings from ALS patients, fasciculating MUs can die and new MUs can start fasciculating in the time between recording sessions, which would lead to less/more classes of FP waveform.

To overcome the absence of *a priori* knowledge, I have developed an automatic framework that assesses the outcome of the k-means clustering and automatically reseeds additional cluster centres back into the clustering algorithm. The framework is iterative and termed *'cluster and seed'*. The method begins using a single cluster centre. K-means clustering is then performed in 3 dimensions, using the first three principle components of the data (see Section 4.1.1.3); the *kmeans* function in MATLAB is used to perform the clustering algorithm. Data-points that lie outside a Euclidian distance threshold (see Equation (4-6)) from the cluster centre are classed as outliers. The distance threshold magnitude is found as the square root of the squares of the variances of the first three principle components:

$$Thresh_{dist} = \sqrt{{\sigma_{PC1}}^2 + {\sigma_{PC2}}^2 + {\sigma_{PC3}}^2} \quad \textbf{(4-6)}$$

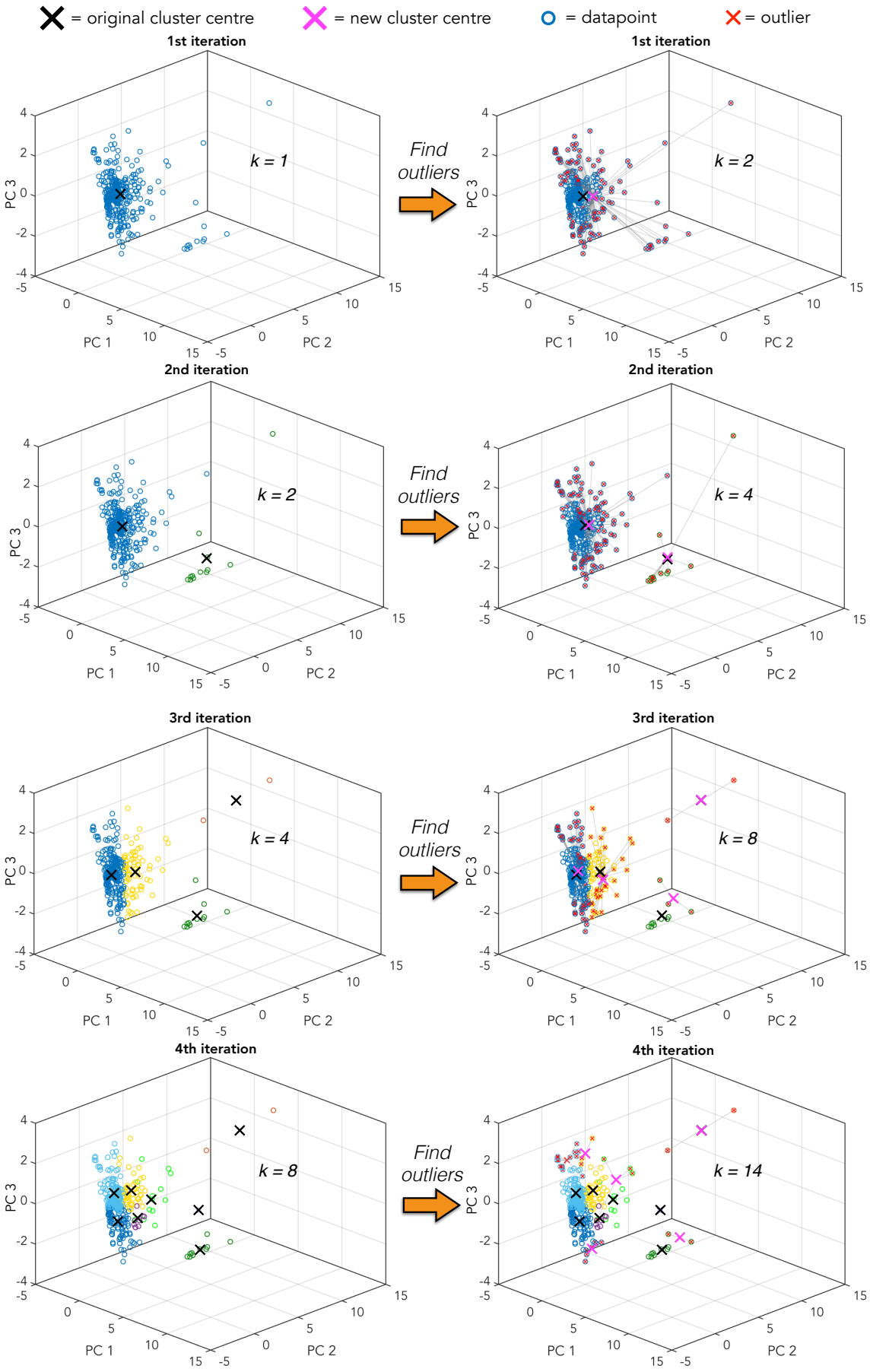
The locations of the outliers of each cluster centre are then averaged to find the mean, which then become the locations of new cluster centres. The locations of the original cluster centres plus the newly formed outlier-cluster centres, are '*seeded*' back into the clustering algorithm. For example, if k=2 and after the initial round of clustering both clusters have outliers, those outliers are averaged and the clustering algorithm is repeated with k=4. The *cluster and seed* framework iterates until all the data-points are within the distance threshold of a cluster centre, or until an iteration limit is reached:

1. *Cluster data points using cluster centre(s) (starting with k = 1).*
2. *Find outliers of each cluster centre (data-points outside distance threshold).*
3. *Average outliers to create new cluster centres.*
4. *'Seed' new cluster centres into k-means clustering algorithm with existing cluster centres.*
5. *Repeat until all data-points are within the distance threshold of a cluster centre.*

Figure 4-13 demonstrates the *cluster and seed* framework iterating on a dataset, to cluster data without the need for *a priori* knowledge or for the user to visualise the data first. The dataset used in Figure 4-13 is the same set of positive FPs used demonstrate PCA (see Section 4.1.1.3).

(Figure on opposite page)

Figure 4-13: 'Cluster and seed' framework for the first four iterations. *In the first iteration the data is clustered with k = 1 and the cluster centre is found (black X). The outliers outside the distance threshold (small red X) are found and then averaged to form a new, additional cluster centre (pink X). The clustering is repeated with k = 2, the outliers of each cluster centre are found and averaged to create two new cluster centres, meaning k = 4 for the next iteration. The process is iterated until no outliers are found.*


= original cluster centre
= new cluster centre
= datapoint
= outlier
1st iteration
k = 1
Find outliers
1st iteration
k = 2
2nd iteration
k = 2
Find outliers
2nd iteration
k = 4
3rd iteration
k = 4
Find outliers
3rd iteration
k = 8
4th iteration
k = 8
Find outliers
4th iteration
k = 14
PC 1
PC 2
PC 3

Continuing with the demonstration of the *cluster and seed* framework shown in Figure 4-13, the algorithm finishes on its sixth iteration when there are no more outliers. Figure 4-14 demonstrates the success of the algorithm at grouping waveforms by their morphology, as shown with the FP waveforms that belong to four different cluster groups. Note, the algorithm has successfully separated out doublet waveforms, often associated with neuromuscular disorders such as ALS. The very tightly clustered groups of FPs shown in orange at the bottom of Figure 4-14 are an extremely repeatable waveform compared to the other more loosely clustered groups of FPs which show some waveform variability. Four of the clusters found have an FP count of 1, meaning these waveforms lay outside the distance threshold of any other data-point in the principle component space.

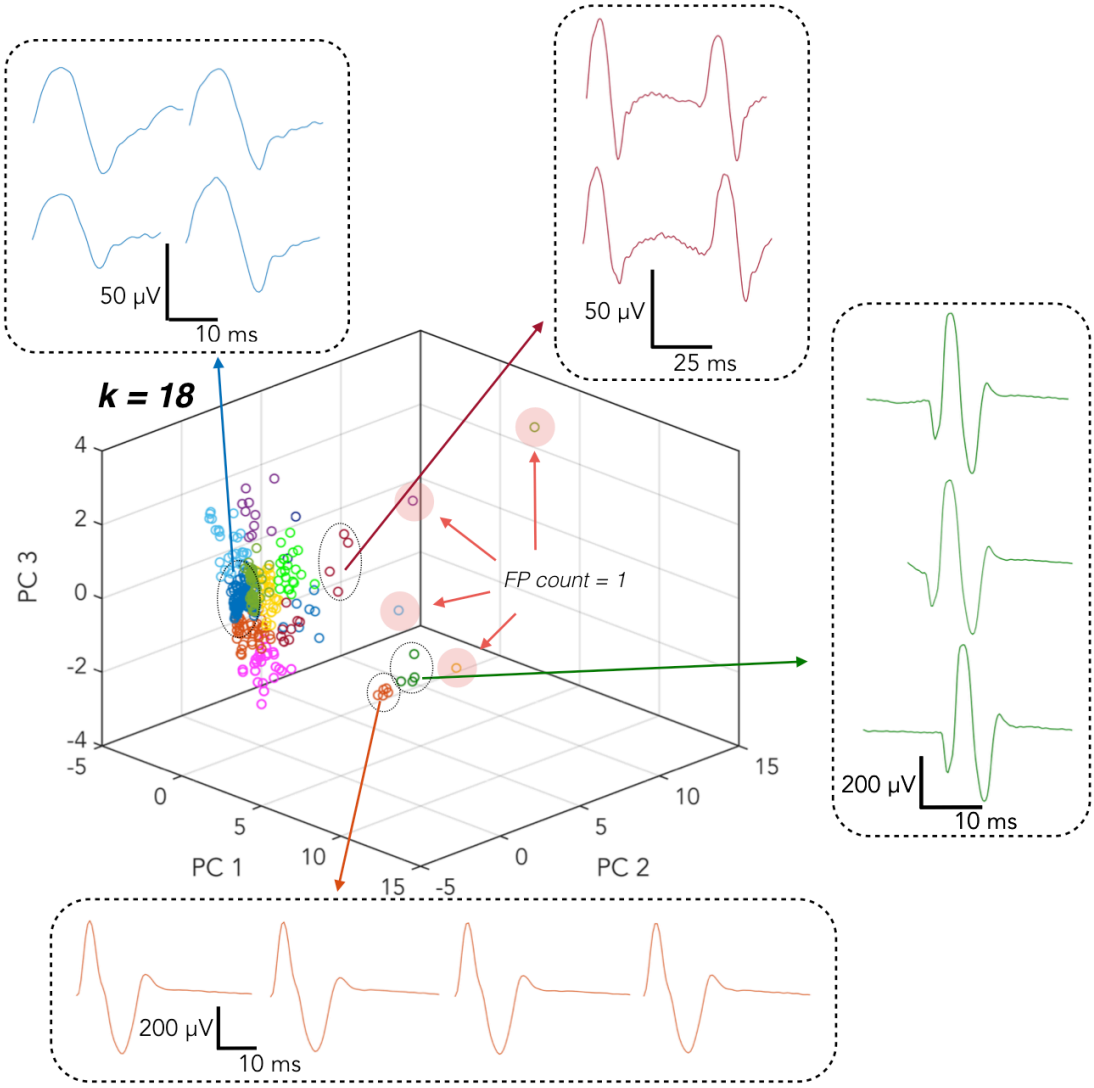


Figure 4-14: Final iteration of 'cluster and seed' framework. *Final (sixth) iteration of the cluster and seed algorithm that has clustered the data into 18 clusters (k = 18), 4 of which are clusters with an FP count of 1 (circled red). Example FPs from four different clusters are shown, demonstrating the success of the clustering algorithm at separating FPs by their morphology.*

Depending on the dataset and how the data-points occupy the principle component space, the *cluster and seed* method can over-cluster the data; the *cluster and seed* algorithm has clustered the data in Figure 4-14 with k = 18. By over-clustering the data, groups of FPs can potentially be split into more groups than necessary. However, this is seen as a preferable option as it will lower the risk of incorrect FP grouping that can occur due to under-clustering. Jahanmiri-Nezhad et al. also opt for over-clustering in preference to under-clustering when grouping FPs from ALS patients, stating "overestimation is better than underestimation"[43] in reference to the number of cluster centres to use.

4.1.2 Matched filtering

Matched filtering is a signal processing technique used to detect the presence of a *known template* in an *unknown signal*. Matched filtering is commonly used in radar, where the reflection of the sent signal (unknown signal) is searched for traces of the original outgoing signal (known template). The method is able to identify common elements between the template and signal, even in the presence of stochastic noise and unknown artefacts.I explored the use of matched filtering for finding the locations of FPs in EMG signals in the MRes phase of this project, and have since developed this technique into a successful framework for grouping FPs by their morphology.

The first step in the developed method is to form the matched filter, h, by time-reversing the known template, T (see Equation (4-7)). The unknown signal, x, is then convolved with the matched filter to produce the impulse response output, y (see Equations (4-8) and (4-9)). A demonstration of this technique, using an FP as a target template and an EMG recording as the unknown signal, is shown in Figure 4-15.

$$h(n) = T(N_T - n + 1) \quad \textbf{(4-7)}$$

$$y(n) = x(n)h(1) + x(n-1)h(2) + \cdots + x(n - N_h + 1)h(N_h) \quad \textbf{(4-8)}$$

$$y(n) = x(n) * h(n) = \sum_{k=-\infty}^{\infty} x(n)h(n-k) \quad \textbf{(4-9)}$$

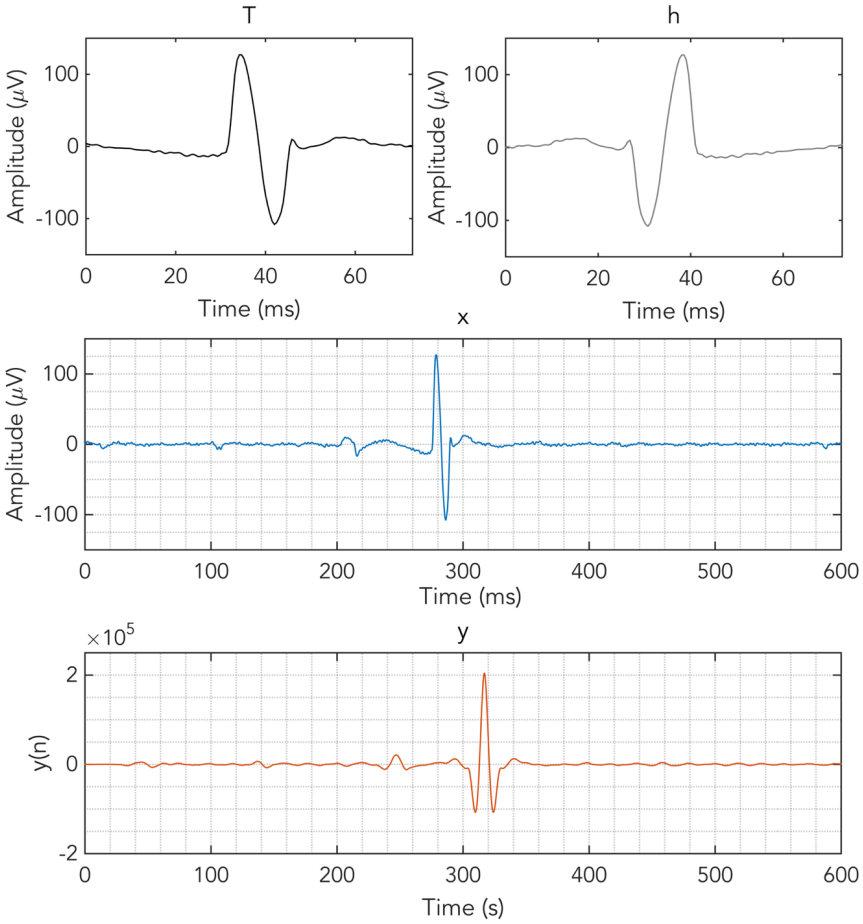


Figure 4-15: Formation of matched filter. *The target template, T, (top-left) is time-reversed to form the matched filter, h (top-right). Convolving the unknown signal, x, (middle) with the matched filter produces the impulse response, y (bottom). If the signal contains any elements of the original template, the resulting impulse response produced will be similar to that shown.*

To determine whether or not the unknown signal contains any elements of the target template, the morphology of the impulse response is examined. If the signal contains an *identical match* of the target template, the peak of the impulse response can be calculated by multiplying the template vector T with its transpose to produce the ideal peak value u:

$$u = TT^T \quad \textbf{(4-10)}$$

$$Peak_{ideal} = u \quad \textbf{(4-11)}$$

The peak value of the impulse response will only be equal *to* u if the signal contains an *identical copy of the template* (see Figure 4-16), which is very unlikely to happen in an unknown signal affected by stochastic noise and unpredictable variations. For this reason, a threshold is set above and below the

value of *u*; impulse response peaks that fall within this boundary of *u* signify a possible match in the unknown signal with the target template.

Another measure that is taken from the impulse response is its symmetry. The impulse response's morphology is similar to that of a 'Mexican-hat wavelet', and if it is a perfect match with the target, it will have a symmetrical impulse response. To check the impulse response's symmetry, the response's waveform is split in half at its peak and the two halves of the waveform are correlated against each other to produce the coefficient *r*, which will be a value between 0 and 1. The top of Figure 4-16 shows how the impulse response from a perfectly matching template has a peak value equal to *u* and a completely symmetrical waveform (see the overlay of the right-half and left-half of the impulse response).

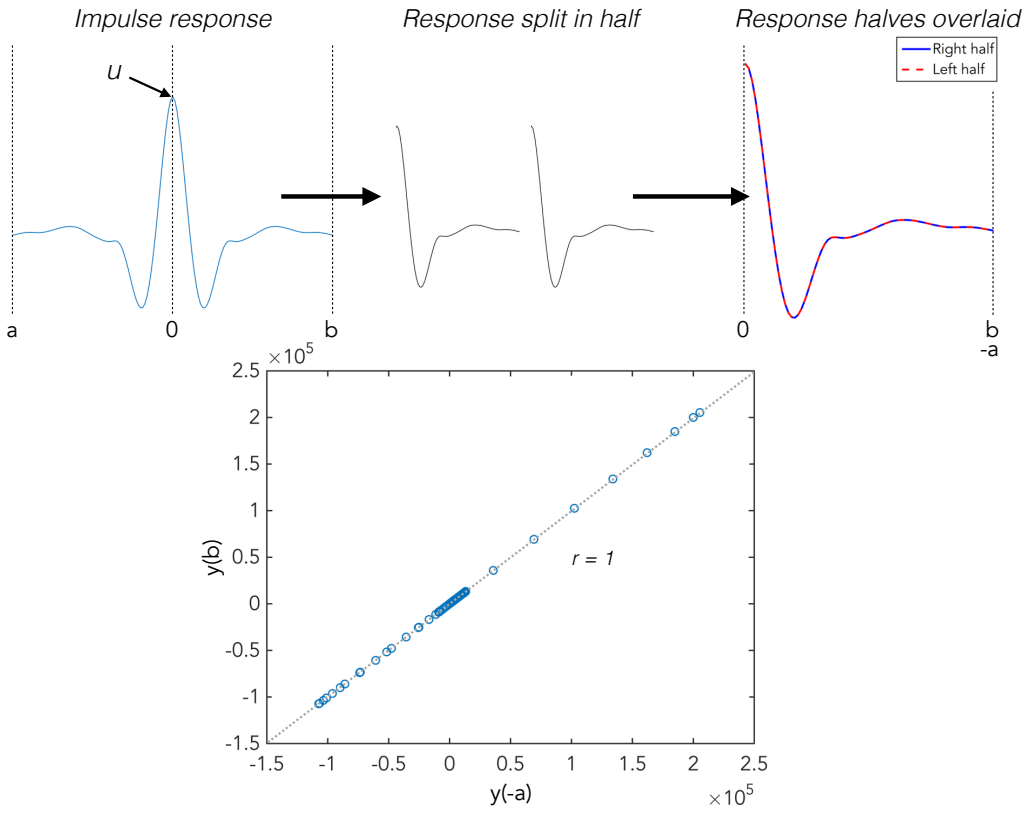


Figure 4-16: Matching criteria of impulse response. *The perfect impulse response (top-left) found when the target template is convolved with an identical match, giving a peak value equal to u and a completely symmetrical waveform. The symmetry is checked by correlating the first half with the second half of the impulse response, giving a value of r = 1 (perfectly symmetrical).*

Again, it is unlikely that two halves of the impulse response will be perfectly correlated ($r = 1$), signifying a perfect match, and so a threshold for *r* is also

applied. Figure 4-17 demonstrates the result of matched-filtering with an unknown signal, showing how a very similar waveform to the target template in the signal has been found and how the values of *u* and *r* differ for these very similar waveforms.

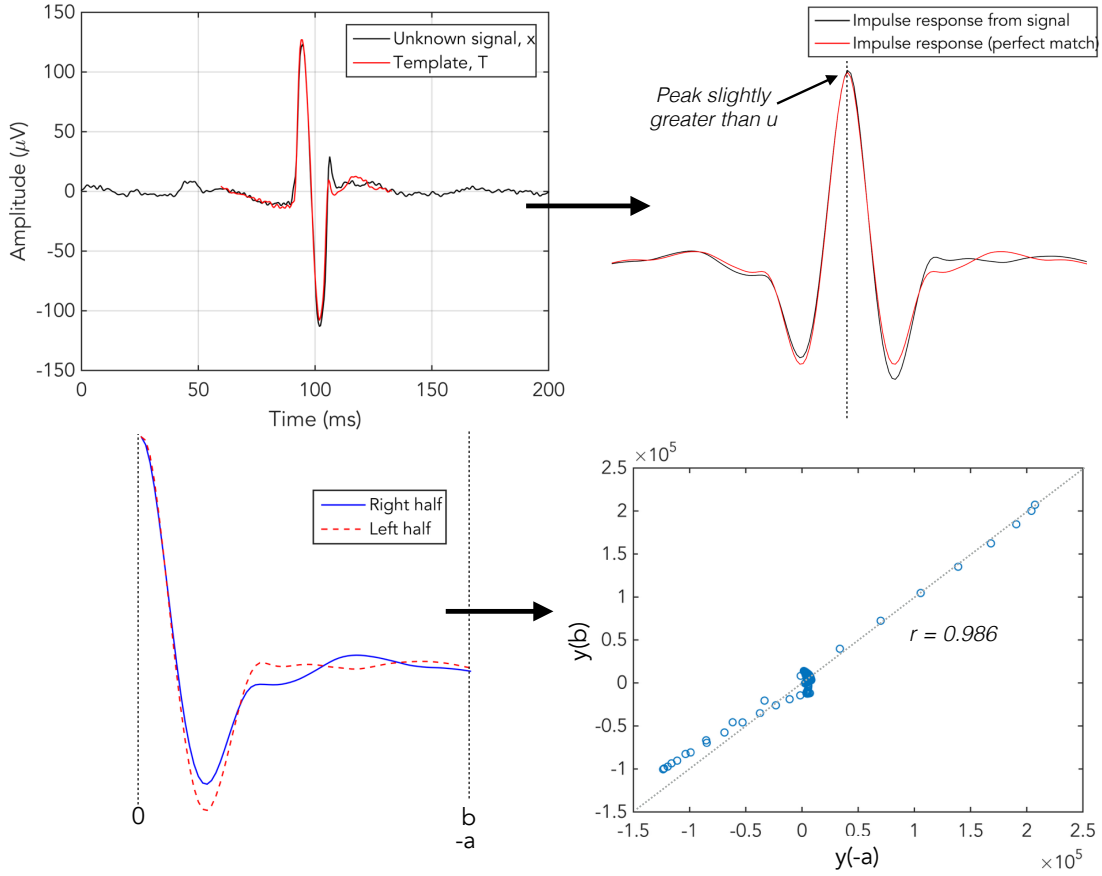


Figure 4-17: Matched filtering method used to find a waveform of similar morphology to the target template in the unknown signal. *By comparing the peak value of the impulse response to the ideal peak value u, followed by measuring the response's symmetry, it can be determined whether or not the unknown signal contains elements of the target template. The peak of the impulse response and the high correlation coefficient, indicates whether or not the unknown signal contains a similar waveform to the target template.*

4.1.2.1 Grouping FPs using matched filtering

The matched filtering method can be very useful for detecting the location of FPs in EMG recordings, by convolving a catalogue of pre-determined FP waveform templates with the EMG signal and examining the locations of peaks in the impulse response. Using a catalogue of pre-recorded FPs/MUAPs for spike detection and decomposition of superimposed signals is not uncommon.[36,41,72] FP waveform morphology tends to be unique to the patient due to their

physiology and their unique denervation and reinnervation of branching motor neurons (previously described in Section 1.2), and so using a catalogue of pre-determined FP waveform templates that are not specific to the patient could be problematic. The clinician/operator will also not know what FP morphologies to expect in the signal (no *a priori* knowledge) when choosing the predetermined template for convolution. In addition to this, the morphology of the motor unit's FP will change and develop as the disease progresses, becoming increasingly complex and increasingly difficult for the clinician/operator to know what template morphology to use for matched filtering.

Instead, the way matched filtering is used in this data processing pipeline is to compare and group the FPs that were detected in the signal using the spike detection algorithm described in Section 3.3. The method works by taking the first FP in the database of detected FPs and time-reversing it to form the matched filter. A fake 'signal' is then formed by concatenating the next FP in the database with empty signals (zeros) either side of it. By convolving the matched filter formed from the first FP with the 'signal' formed from the second FP, an impulse response is produced. The impulse response undergoes the same two tests for peak and symmetry as if it were produced by convolving a template against an unknown signal: first is to check whether the peak of the response is within a percentage threshold of u, followed by checking the symmetry of the impulse response using the correlation coefficient, r. Note, that in the data processing pipeline the peak of u is checked first against the threshold, followed by checking the value of r; finding the peak value first makes it easier to split the waveform in half in order to check its symmetry. After this, the process is repeated with the next FP in the database being concatenated in an empty signal and convolved with the matched filter. The process repeats itself until all the FPs in the database have been convolved with the matched filter. The process is outlined in Figure 4-18. If the FP is deemed to be a match, its ID is saved and the next FP in the database is examined. If it's not a match, it is ignored and the next FP is examined. This process continues, forming groups of matching FPs.

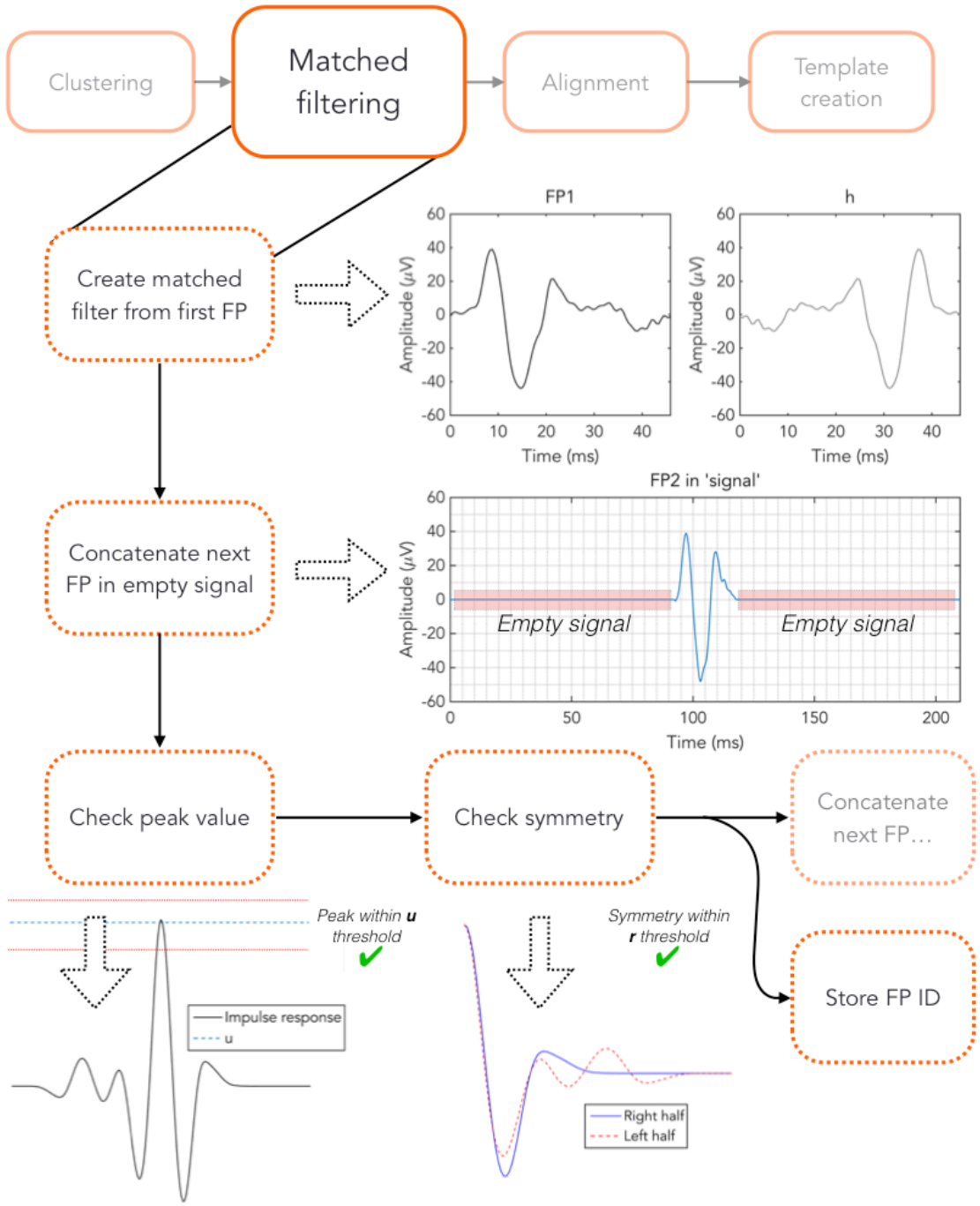


Figure 4-18: Checking u and r thresholds of a database of FPs. *The first FP in the database is time-reversed to form the matched filter. The next FP in the database (found using the spike detection algorithm) is concatenated into an empty signal, and the impulse response is formed as a result of their convolution. The peak of the impulse response is checked against the ideal peak value, u, and the symmetry of the response is checked by correlating the two halves of the response against each other to produce r. If the FP passes both thresholds, it is deemed a match and its ID is stored.*

Strict thresholds are used for both checking the peak value of the impulse response against *u* and when checking the correlation value *r*, to ensure there

aren't any false groupings of FPs. While this approach decreases the number of false positives (incorrect matches) from occurring, it can also increase the number of false negatives (missed correct matches) that occur too. This is due to the morphology of FPs in ALS patients being inherently variable, and becoming increasingly more so as the disease progresses.[23] This defines the key difficulty in grouping FPs by morphology: trying to match FP morphologies by their similarities, whilst they are inherently different and variable. Table 4-1 demonstrates five FPs of matching morphology that have originated from the same MU, however the strict thresholds applied to prevent FP mismatching have resulted in a 'no match' between FPs with slightly different morphologies.

Table 4-1: Results of using matched filtering to match five FPs of matching morphology. *The strict threshold values used for the peak height u and the correlation r of the matched filter's impulse response reduce the risk of false positives but can also increase the chance of false negatives. The five FPs shown have all originated from the same motor unit, however due to the inherent variability in their morphologies, the matched filtering algorithm has resulted in a combination of matches and mismatches. For example, FP3 only matches with FP1 under the strict threshold values, whereas FP1 matches with FP2-FP5.*

	①	②	③	④	⑤
①	N/A	*Match*	*Match*	*Match*	*Match*
②	*Match*	N/A	**No match**	*Match*	*Match*
③	*Match*	**No match**	N/A	**No match**	**No match**
④	*Match*	*Match*	**No match**	N/A	*Match*
⑤	*Match*	*Match*	**No match**	*Match*	N/A

When the entire database of FPs has been examined, the process repeats itself using the next FP in the database to form the matched filter. The process keeps repeating until every FP in the database has been used as a matched filter. To overcome the number of false negatives occurring from using strict thresholds, the groups of FPs formed from using each FP as matched filter are combined based on the common FP IDs that they may share. For example, *within the thresholds set for u and r, if A matches with B, and B matches with C, then A, B, and C must belong to the same FP group.* The use of the matched filtering method for grouping FPs by their morphology is built into the framework shown in Figure 4-19. Combining FP groups by the cross comparison of common FP IDs is demonstrated in the example in Figure 4-20.

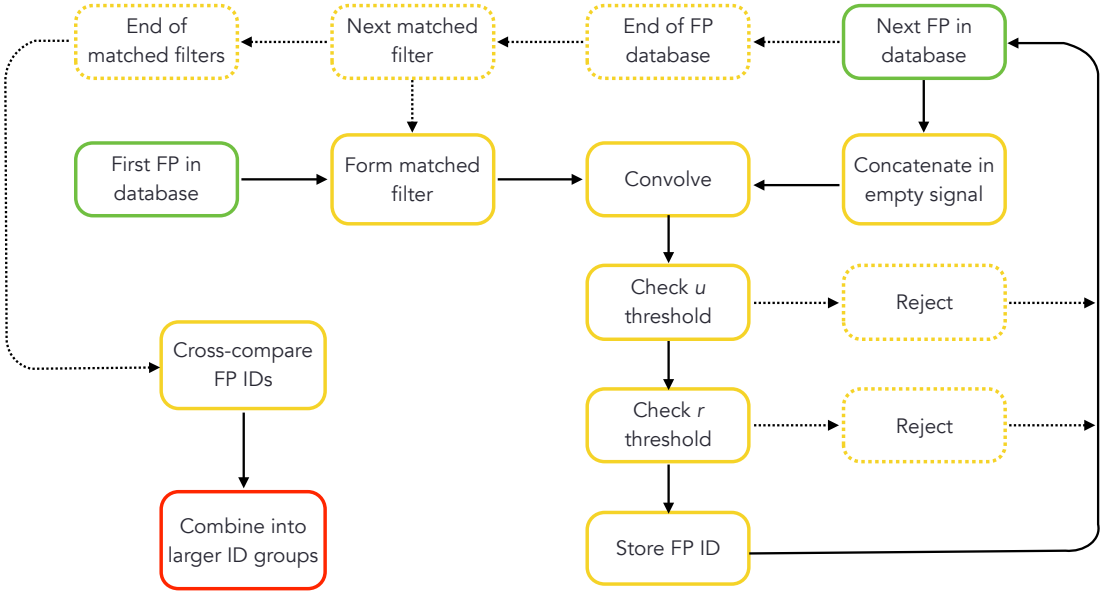


Figure 4-19: Flowchart for grouping FPs using matched filtering. *The algorithm begins with the two green cells in the flowchart. The first FP in the database is used to form the first matched filter, and the next FP in the database is concatenated into an empty signal. The matched filter is convolved with the FP, and its impulse response is examined. If the response fails the threshold checks for u and r, the FP is rejected and the algorithm moves onto the next FP in the database. If it passes the threshold checks, the FP ID is stored and the algorithm moves onto the next FP in the database. This process repeats using the same matched filter until all the FPs in the database have been examined, at which the point the algorithm creates a new matched filter from the next FP in the database. This repeats until all FPs in the database have been used as a matched filter, being compared with all other FPs. At this stage, the groups of FP IDs are compared for any FP ID similarities. The algorithm ends at the red cell in the flowchart, with groups that have common FP IDs being combined into larger ID groups.*

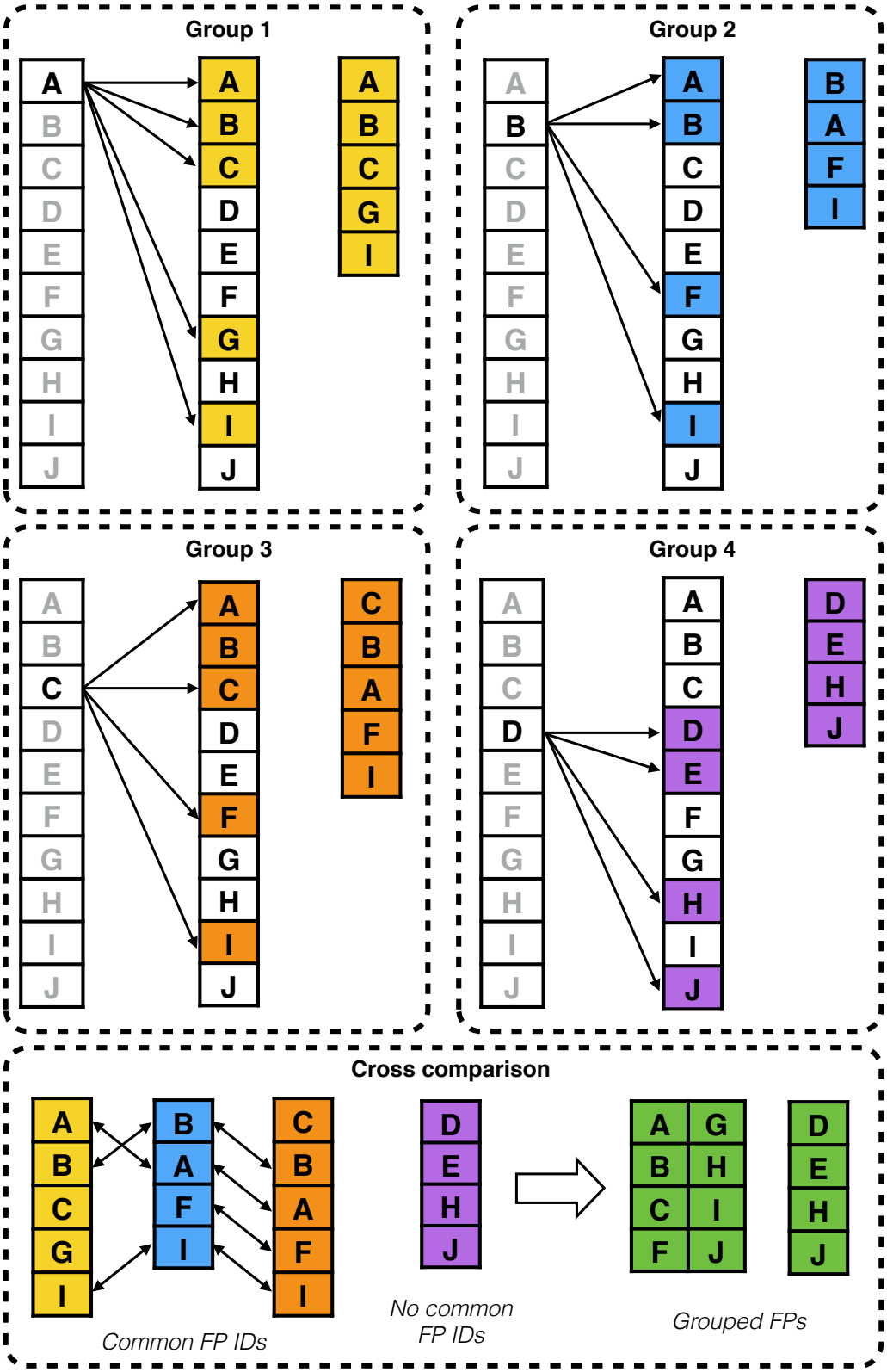


Figure 4-20: Cross-compare FP IDs and combine groups. *The groups of FPs found using each FP in the database (A-J) as a matched filter undergo cross-comparison. Groups with common FP IDs are combined to form larger groups. This example demonstrates how the first four FPs (A-D) are used to find groups 1-4. Groups 1, 2, and 3 are combined to form ABCFGI and group 4 is not combined as none of its FP IDs are shared with any other group.*

4.1.2.2 Shuffling framework

Checking every single FP with every other FP for similar morphologies using matched filtering can be very computationally expensive, particularly when the patient EMG recordings can contain potentially many thousands of fasciculations. Instead, I have developed an additional framework using the matched filtering method for grouping FPs known as 'shuffling'. The shuffling framework works by removing FPs from the database that have been grouped together in order to reduce its size, and then by repeating the matched filtering process but with the FP database in a randomised order.

The shuffling framework begins with a temporary copy of the FP database being made. Next, the first FP from the database is used to form the first matched filter. The FPs in the database found to be a match have their IDs saved and are then removed from a temporary database. The next FP in the temporary database forms the new matched filter, and the matched filtering process continues with the new, shrunken, temporary database. The shrunken size of the new database will speed up the processing time of the next round of matched filtering as there are now less FPs to compare with. When all the FPs are matched, or the remaining FPs cannot be matched, the algorithm moves onto the next round of matched filtering.

As shown in Table 4-1, small variations in FP morphology may cause incorrect mismatching of FPs when using strict thresholds for u and r. Some FPs may not get the opportunity to be grouped with others, as they will have already been removed from the temporary database. To overcome this, the original database of FPs is 'shuffled' (its order randomised) to form a new temporary database before the matched filtering algorithm begins again. This 'Monte-Carlo' approach of randomisation of the database gives FPs an opportunity to match with other FPs that may have been removed from the database in the previous round of matched filtering. Once a 'shuffling limit' has been reached (number of times to shuffle the database and repeat matched filtering, determined by the user), the groups of FP IDs from each round of shuffling are cross-compared and combined into larger ID groups. A flow chart detailing how matched filtering, the removal of FP IDs from the temporary database, and the randomisation of the

database order have been incorporated into the shuffling framework is shown in Figure 4-21. The removal of groups of FPs from the database, the randomisation of the new database in each 'shuffling round', and the comparison of FP IDs is described in the example in Figure 4-22.

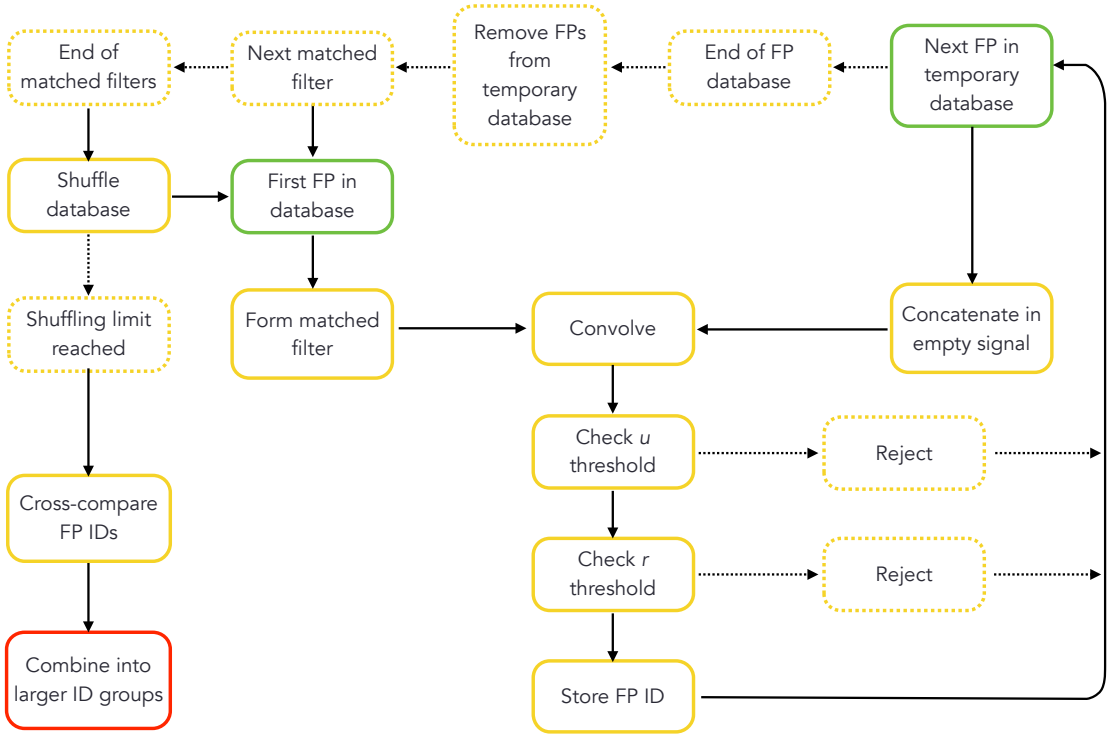
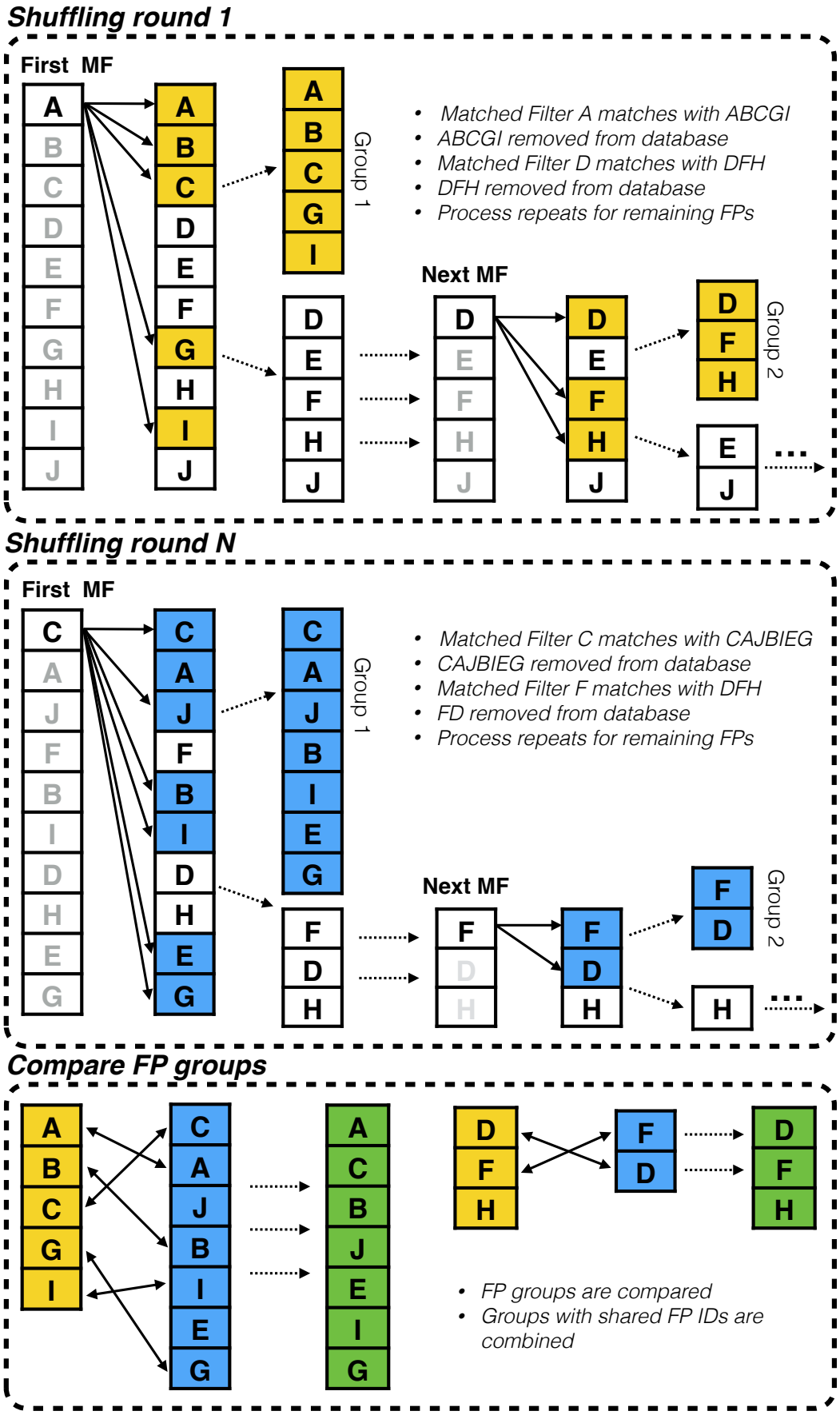


Figure 4-21: Flowchart for grouping FPs using matched filtering and shuffling framework. *The shuffling algorithm begins with the two green cells in the flowchart. The first FP in the database is used to form the first matched filter, and the next FP in the database is concatenated into an empty signal. The matched filter is convolved with the FP, and its impulse response is examined. After every FP in the database has been examined, the IDs of FPs that have passed the threshold checks are stored and then those FPs are removed from the temporary database. The process repeats itself with the next matched filter being formed from the first FP in the shrinking temporary database. When the temporary database is depleted of FPs to create matched filters from, the original database is 'shuffled', or randomised, in order to remove any bias from the next round of matching. The whole process repeats itself until the predetermined 'shuffling limit' has been reached. At this stage, the groups of FP IDs are compared for any FP ID similarities. The algorithm ends at the red cell in the flowchart, with groups that share FP IDs being combined into larger ID groups.*



(Figure caption on next page)

Figure 4-22: Grouping matched FPs using shuffling framework. *In this example, the algorithm starts with the first FP (labelled A), and time-reverses it to form the matched filter (MF). The matched filter (A) is convolved with the remaining FPs in the database (A-J) and its matches (ABCGI in yellow) are removed from the database. The process repeats itself with the remaining unmatched FPs in the temporary database (DEFHJ), with D being used as the matched filter. The new matched filter is convolved with the remaining FPs, until there are no more matches to be made. At this point, the original database is 'shuffled' into a random order to remove any bias from the next round of matched filtering. In this example the matched filter formed from C matches with CAJBIEG and the new matched filter, F, in the reduced database matches with FD. When the shuffling limit has been reached, the groups of matched FPs that share common IDs are combined. In the example, the yellow groups from the first round of shuffling and the blue groups from the nth round of shuffling are combined to form the larger green groups ACBJEIG and DFH.*

Using the original framework (see Figure 4-19 and Figure 4-20) to group FP morphologies will require a large amount of computational effort. For example, a database containing n = 1000 FPs will require 999,000 individual convolutions (calculated using Equation (4-12)), with each FP being used as a matched filter.

$$N_{convolution} = n^2 - n \quad \textbf{(4-12)}$$

The shuffling framework works to reduce the number of convolutions required. Figure 4-23 demonstrates how the number of FPs in the database is reduced as groups of FPs are removed for the first five shuffling rounds. The x-axis 'FP removal stage' refers to the '*Remove FPs from temporary database*' stage in the shuffling framework (refer to Figure 4-21).

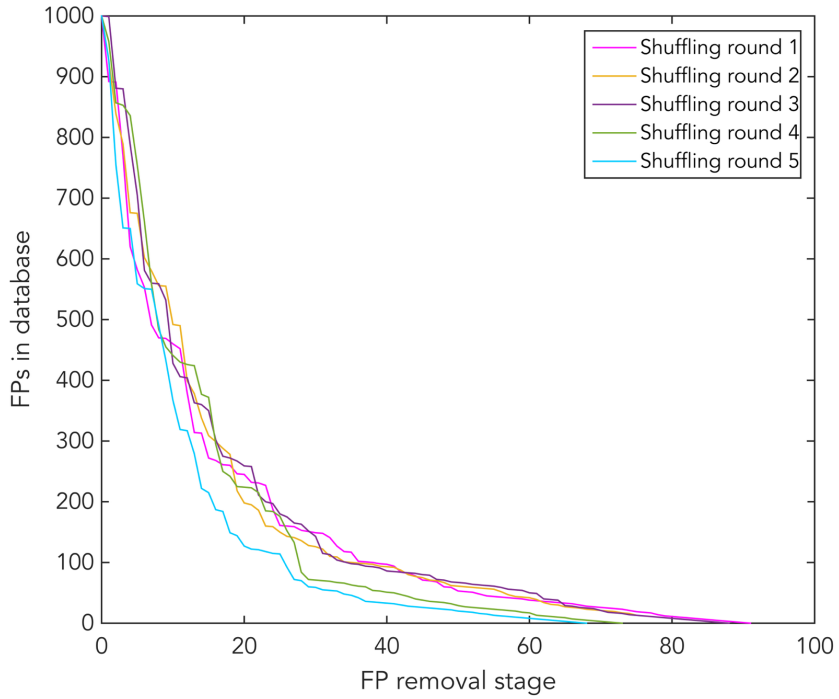


Figure 4-23: Number of FPs in temporary database during shuffling rounds. *The plot demonstrates the number of FPs in the temporary database that remain after each group of matching FPs is removed for a given shuffling round. The results of the first five rounds of shuffling are shown.*

The first five shuffling rounds shown in Figure 4-23 have required a total of 59,463 convolutions. The more shuffling rounds used, the more opportunity FPs have to be convolved with other FPs that may have already been removed from the temporary database. The shuffling framework has been run using a larger number of shuffling rounds ($N_{shuffle}$ = 20), requiring 283,401 convolutions; a fraction of what would have been required if using the original framework previously detailed in Section 4.1.2.1.

4.1.3 Comparison of clustering and matched filtering methods

4.1.3.1 Problems with clustering and matched filtering

Whilst clustering is a proven unsupervised classification method, when it comes to the classification of FPs, clustering it can often make mistakes; FPs of different waveforms that have originated from different MUs can sometimes be incorrectly clustered together. This can often be due to the lack of *a priori* knowledge of how many FP waveform classes to expect in the EMG signal, and thus the number of cluster centres to use. This problem can be mitigated by

using a framework such as the 'cluster and seed' method detailed in section 4.1.1.4, or by an 'overestimation'[43] of the number cluster centres to use. However, despite this, incorrect clustering can still occur due to FPs with different morphologies often sharing similar features such as amplitude, duration, area etc. Figure 4-24 shows a cluster of mismatching FPs as a result of using the 'cluster and seed' framework and the same dataset used in Sections 4.1.1.3 and 4.1.1.4.

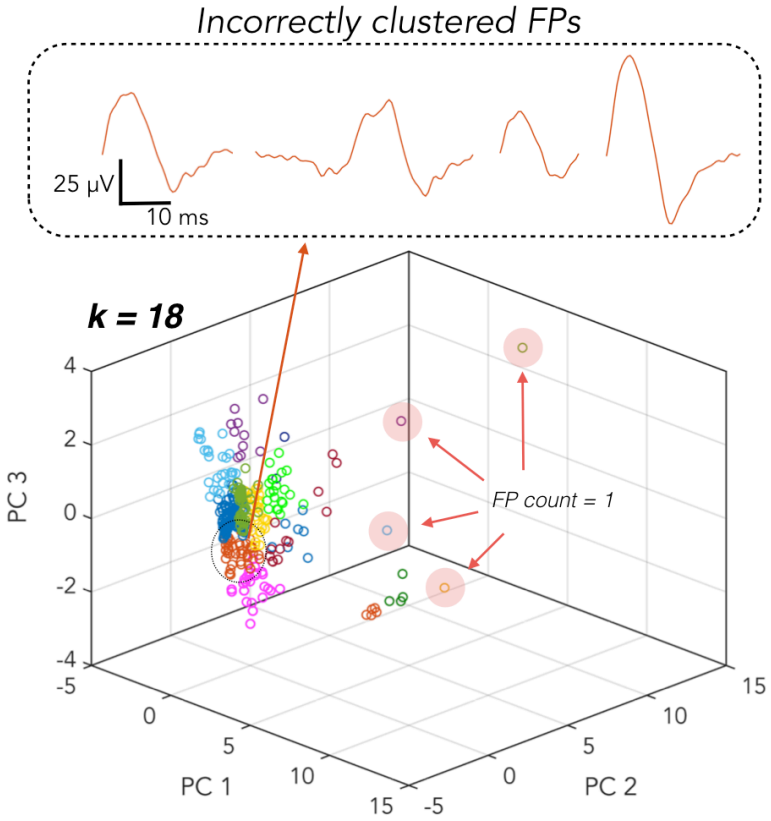


Figure 4-24: Errors caused by clustering due to similarity of FP features. *Despite using the 'cluster and seed' framework to detect outlying data-points and reseed new cluster centres back into the clustering algorithm, the circled cluster contains FP morphologies that have been incorrectly matched. Clustering is a powerful machine learning technique, however, it can result in the incorrect grouping of FPs due to the similarity of their features.*

Matched filtering is a comparison of morphologies rather than specific features. If the thresholds used for *u* and *r* are too liberal (see Section 4.1.2) then matched filtering can also result in incorrect matching of FPs. If the correct thresholds are applied, matched filtering can be a powerful technique for comparing FP morphology. However, the method can be very computationally expensive, even when coupled with the shuffling framework described in Section 4.1.2.2.

Clustering is a fast, computationally inexpensive technique; matched filtering can require significant processing power to convolve the morphologies of potentially thousands of FPs.

4.1.3.2 Combining clustering and matched filtering

The data processing pipeline I have developed draws on the advantages of both clustering and matched filtering in order to correctly group/classify FPs by their waveform. By initially using the k-means 'cluster and seed' algorithm, the FPs are split into smaller groups based on the similarity of the dataset's principle components. Each group formed is then processed using matched filtering, which helps to refine the groups of FPs further by removing any FPs that have been incorrectly clustered as well as splitting the clusters into groups of FPs that have matching morphology. However, this method can potentially lead to many small groups of FPs being generated; a solution to this is described in Section 4.1.4. Figure 4-25 describes how clustering and matched-filtering are combined to group FPs together.

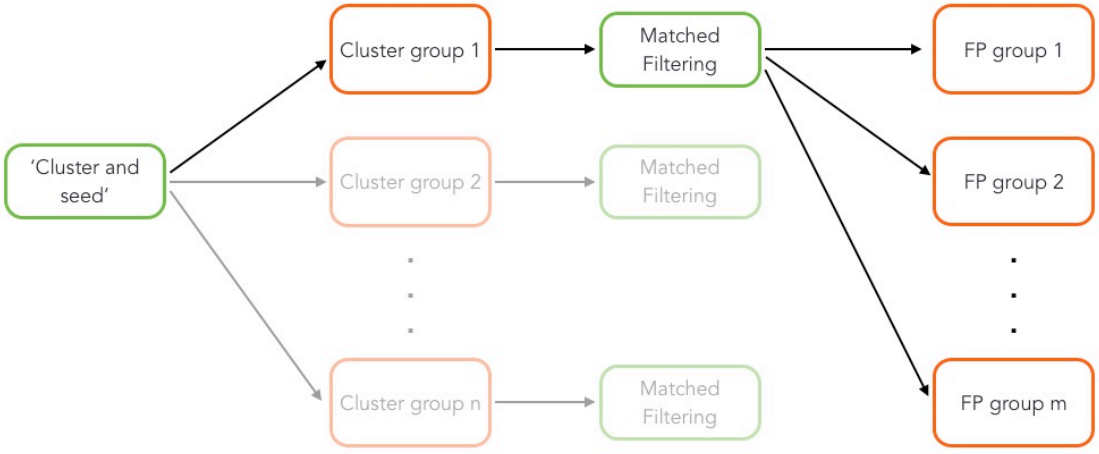


Figure 4-25: Combining clustering and matched filtering. *The k-means 'cluster and seed' algorithm is used to cluster the data into n groups based on the FPs principle features. Each clustered group undergoes matched filtering to further divide the data into groups of matching FPs and remove any FPs that were previously incorrectly clustered.*

The computational load of matched filtering is significantly reduced as the initial dataset has been split into smaller groups by the clustering stage. Using the clustered data from Figure 4-24, it is possible to observe the size of the cluster groups shrink as FPs are removed by the matched filtering and shuffling

framework, and thus calculate the number of convolutions. Figure 4-26 shows the size of the shrinking database for the first round of shuffling with and without the use of clustering. Without clustering, the database of 355 FPs requires a total of 2,605 convolutions (see top of Figure 4-26). With clustering, the dataset of 355 FPs is first split into 18 clusters (four rejected as FP count = 1) which then requires a total of 1,087 convolutions (see bottom of Figure 4-26).

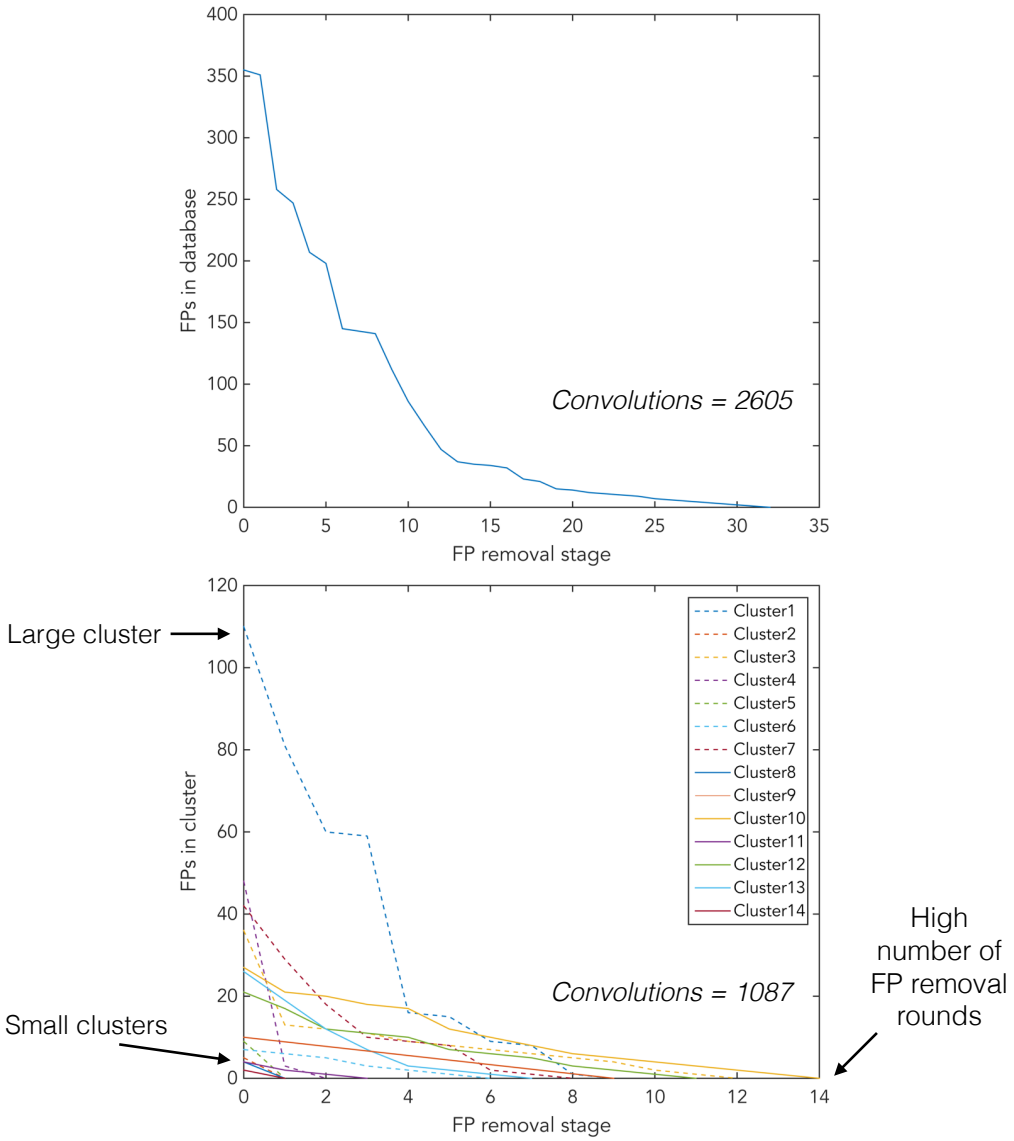


Figure 4-26: Number of FPs in the temporary database after each FP removal stage of the matched filtering shuffling framework, for one shuffling round. *(Top) Without using clustering to first split the database into smaller clusters, the matched filtering shuffling framework requires a total 2605 convolutions. (Bottom) By clustering the database first into smaller groups, the computational load of the matched filtering algorithm is reduced, only requiring 1087 convolutions. Some clusters have required a higher number of FP removal rounds than others, suggesting these clusters contain more mismatching FPs than others.*

4.1.3.3 Rejected FPs

Any unmatched FPs during the grouping stage are rejected by the pipeline on the principle that a solitary FP cannot be analysed for morphology variability if it has no other morphologies to compare it to. These fasciculations occasionally have originated from a MU that has fired just once during a recording session, but often the solitary FPs are rejected in the matched filtering stage because they were previously incorrectly clustered.

The reason for FPs being rejected is often due to noise. For example, during FP detection, noise on the EMG signal can make it difficult to determine the beginning and end of a fasciculation, leading to its duration being measured as longer/shorter than it actually is. Noise can also affect the morphology of the FP in other ways, such as a bigger/smaller amplitude and differing levels of complexity etc. For these reasons, FPs can have distorted features which leads to them being incorrectly clustered. Another reason for FP rejection can be due to an FP being incorrectly split by polarity (see Section 4.1.1.2). This would lead to the incorrect clustering of that data-point and thus the rejection of that FP from its cluster group during the matched-filtering stage.

To ensure FPs are not being wrongfully excluded, all rejected FPs are grouped together and run through the matched filtering algorithm again. The number of rejected FPs should be small compared to total number of detected FPs, and so it will not take long to check their morphologies using matched filtering. If any groups are found in the rejected FPs they are reincluded in the data processing pipeline. Occasionally, noise or other stochastic artefacts in the EMG recording may have been wrongfully detected as an FP using the automatic detection algorithm and should rightfully be rejected by the pipeline.

4.1.4 Alignment and template creation

It is useful to align the grouped FP waveforms so that any changes/fluctuations in morphology can be more easily observed, and any variability calculated. FP waveforms can also be averaged to form a template for that fasciculating motor unit and saved for future use. Instead of manually aligning each group of FPs, an output from the matched filtering algorithm is exploited; the peak of the impulse response is also the location at which the two FP waveforms match best (see

Figure 4-27). FPs of matching morphology will very rarely have the same duration and so aligning them by their onset/offset will not work; the inherent nature of the FPs variability, coupled with the stochastic noise on the signal, can give FPs very different onset/offsets. During the alignment process, each FP is concatenated in an array of empty zeros to account for differences in FP duration.

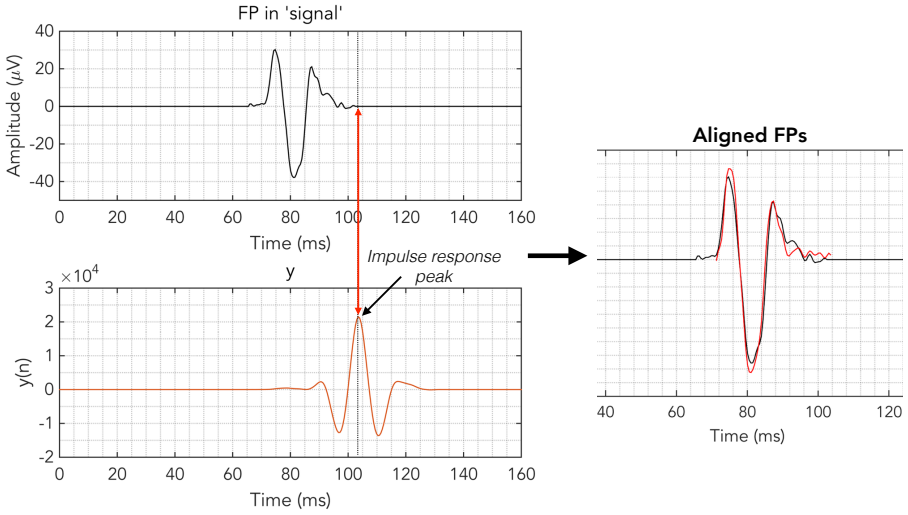


Figure 4-27: Alignment of matching FPs. *The alignment of FPs is done using the peak of the impulse response from matched filtering. The peak of the impulse response indicates the location where the two FP morphologies match best.*

The alignment algorithm is an iterative process. The grouped morphologies are first aligned using the steepest part (maximum gradient) of each waveform, and then averaged to form a 'starting template'. Each FP waveform is then run through the matched filtering algorithm, using the starting template as the matched filter. The locations of the impulse response peaks for each FP are recorded, which are then used to align the FP waveforms with the starting template. The re-aligned FPs are then averaged again to form a new template. This process is repeated, with the alignment of the grouped waveforms improving with each iteration. The iterative process ends when the averaged template stops changing, as shown in Figure 4-28. Note that often, aligning the waveforms at the point of maximum gradient can give a starting template very similar in morphology to the final averaged template.

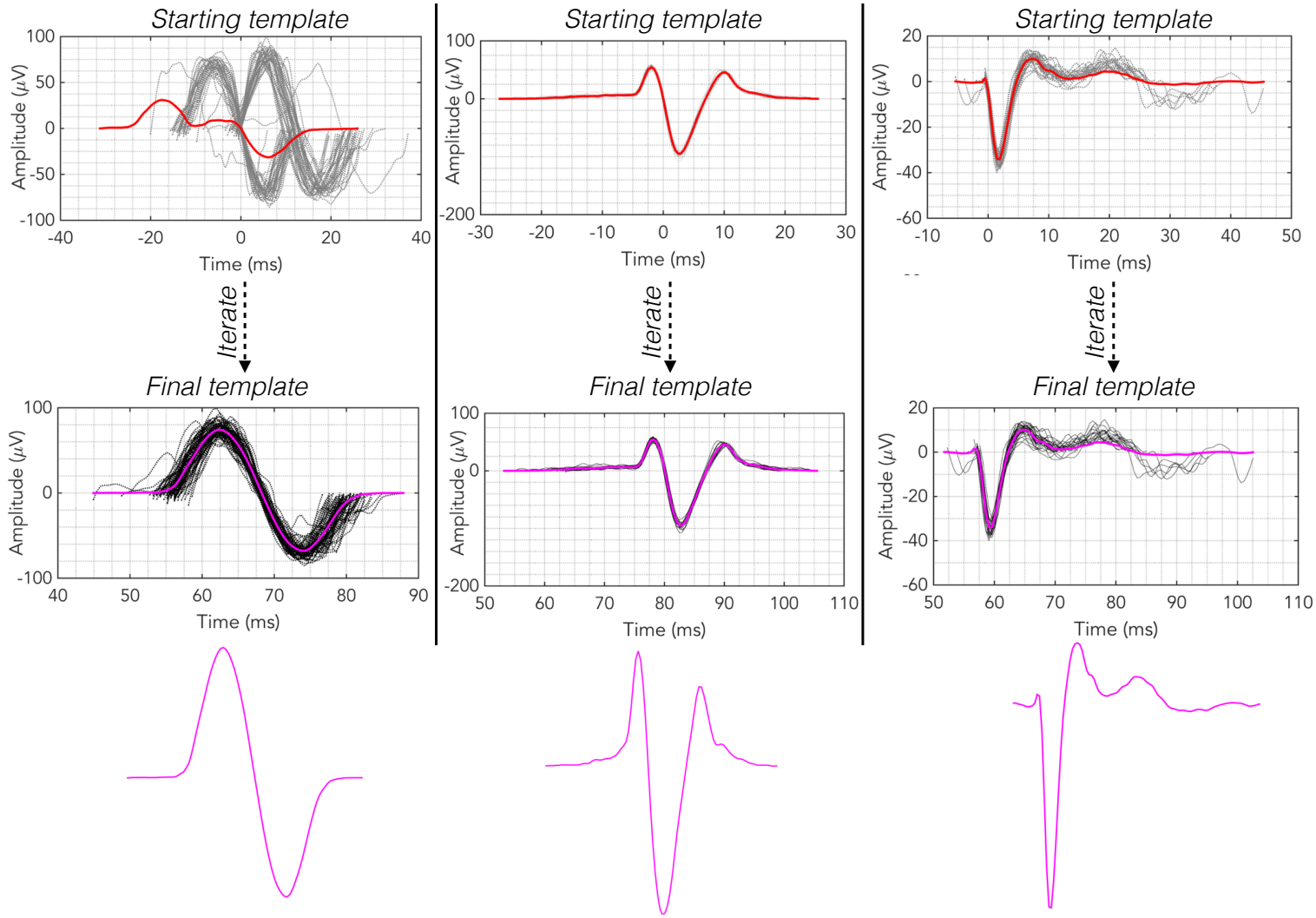


Figure 4-28: Alignment of grouped FP waveforms. *FP waveforms are aligned and averaged to create an average template using the peak of the impulse response outputted from the matched filtering algorithm. The alignment of waveforms is an iterative process; a new averaged template is created each iteration, which the FP waveforms are aligned to. The iterative process ends when the averaged template stops changing, and thus the alignment of the FPs cannot be improved. The examples shown (from left to right) are produced from the alignment of 73, 38, and 33 FPs, respectively.*

The alignment process can be summarised by the following steps:

1. *Align FP waveforms at the maximum- gradient point of each morphology.*
2. *Average FP waveforms to form starting template.*
3. *Convolve each waveform through matched filter using starting template.*
4. *Align FP waveforms using location of impulse response peak.*
5. *Average aligned FP waveforms to form a new template.*
6. *Repeat process. Stop when template stops changing.*

4.1.5 Template grouping

Throughout the stages of FP grouping, a strict approach is taken to minimise the risk of mis-grouping waveforms and maximise the pipeline's specificity, as false grouping of FPs will result in an incorrect assessment of waveform variability. As a result, the templates are often split into smaller groups of FPs than they should be. Once again, matched filtering is used in order find which FP templates match, group them together and thus group the underlying FPs used to create the templates together as well. Strict thresholds are applied to de-risk the incorrect grouping of FPs and so some groups of FPs will remain split into smaller groups than they should be.

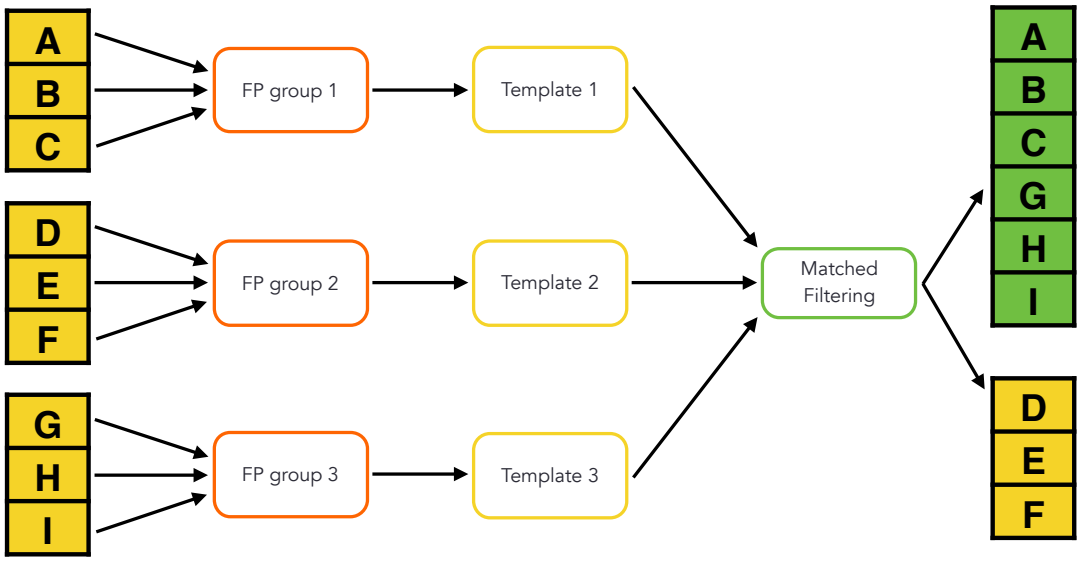


Figure 4-29: Further grouping of FPs using averaged templates and matched filtering. *The templates created from aligning and averaging the groups formed from clustering and matched filtering undergo another round of matched filtering in an attempt to further group FPs that have been incorrectly split into smaller groups. In this example, the FP groups ABC, DEF, and GHI are aligned and averaged to form Template 1, Template 2, and Template 3, respectively. The three templates are passed through the matched filtering algorithm to see if the subgroups of FPs can be grouped further. In the example, FP groups ABC and GHI are grouped together based on the similarity of their template morphologies to form the new FP group ABCGHI.*

Figure 4-30 demonstrates the result of grouping the templates of each FP group using this additional round of matched filtering. Once the templates are grouped together, the IDs of each template are combined to form a large group/class of FPs. For example, in Figure 4-30, Group 1 has been formed from 16 FP templates, combining the FPs used to form each template into a larger group containing 84 FPs.

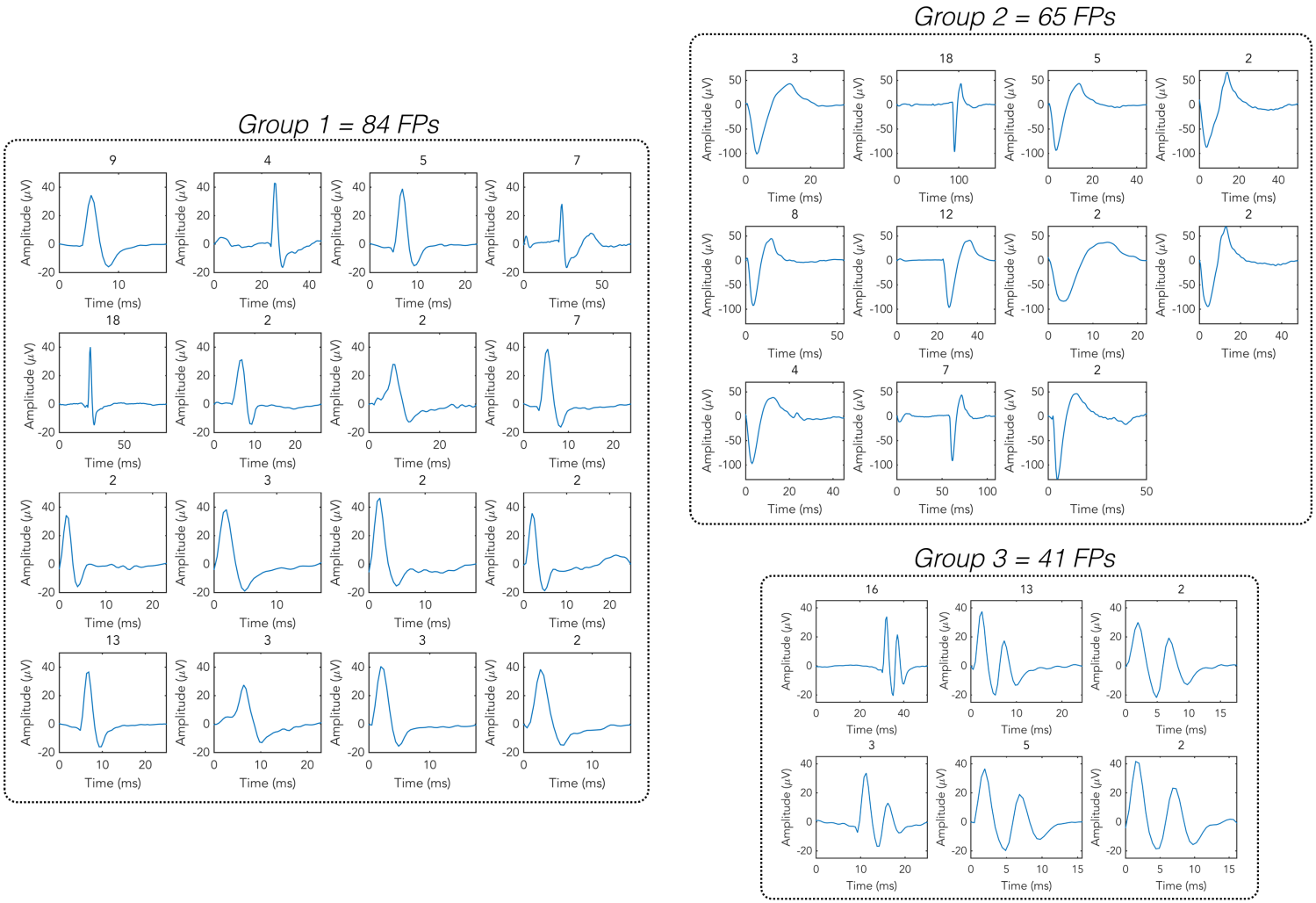


Figure 4-30: Matched filtering is used to group the averaged FP group templates. *This example demonstrates three groups of FP templates that have been grouped together using matched filtering. The number above each template indicates the total number of FP waveforms that were aligned and averaged to form the template. Strict threshold values are still employed to ensure the FP templates are not incorrectly grouped together, meaning that FP morphologies that should be grouped together end up separated into two more different subgroups.*

4.1.5.1 Manual grouping

Up until this stage, the grouping of FPs has been an automatic process, however in order to ensure the output of the pipeline has the highest *sensitivity (TPR)* and *selectivity (1-FPR)* as possible, human interaction is required. Lowering the matched filtering thresholds would result in more templates being automatically grouped together by matched filtering, but likely to result in incorrectly matched templates. This is *extremely undesirable* as any incorrect matching of templates at this stage will result in the incorrect grouping of the underlying FPs that have been used to form the templates.

The manual grouping of templates is a quick process, and may require the user to group just a few individual templates or several groups of templates together. As shown in Figure 4-31, templates have been added manually to Group 1 (see groups in Figure 4-30), increasing the number of FPs associated with this group of FPs from 84 to 96.

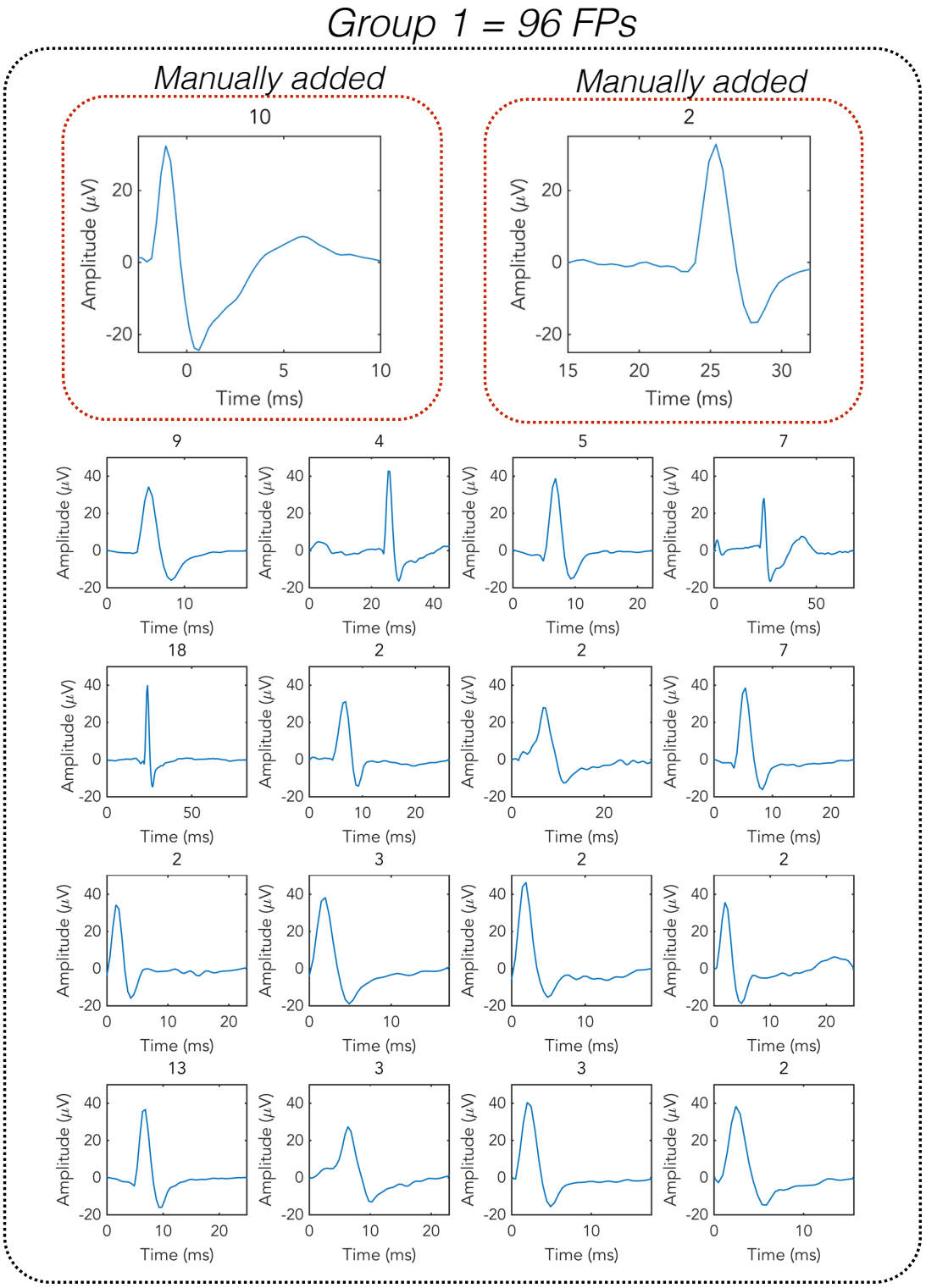


Figure 4-31: Manual grouping of templates. *Remaining unmatched templates are manually grouped together with existing template groups, based on the judgement of the user/operator. This sacrifices some automation but ensures the data-processing pipeline has high sensitivity and selectivity. In this example, two templates were added manually to this group, increasing the total number of FPs by 12.*

4.2 Pipeline Output

Thus far, the detected FPs have been sorted into larger groups of FPs by using a combination of matched filtering and manual grouping (see Figure 4-29). Figure 4-32 demonstrates a group of 74 FPs detected from an ALS patient recording taken at KCL; the recording has been conditioned, and FPs have been detected and grouped using the data-processing pipeline. The FPs have been aligned and superimposed to view the variability between the waveforms.

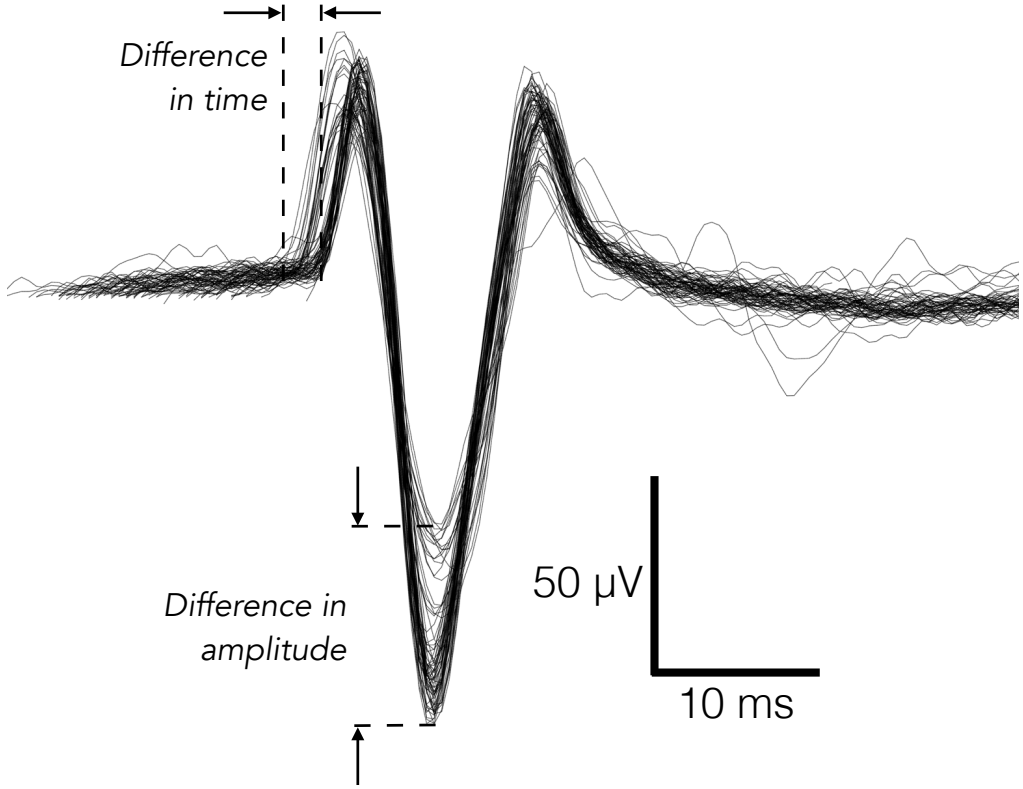


Figure 4-32: Group of FPs outputted by data processing pipeline. *A group of 74 matching FPs detected from an ALS patient recording taken at KCL. The FPs have been aligned using the matched filtering algorithm. Aligning the morphologies shows the differences and variability of morphology amongst fasciculations arising from the same motor unit.*

The FPs detected in the patient recordings have now been grouped based on their morphology, as seen in Figure 4-32, and thus the motor unit they originated from (see introduction of Chapter 3). With this tool, clinicians and investigators can output qualitative information that will help to build a picture about how the motor neuron associated with that MU is deteriorating, as well as the prognosis of the disease in general.

4.2.1 Parameterisation of waveform

There is an abundance of information that can be extracted from the morphology of FPs at this stage: amplitude, area, turns, duration, firing interval, and indices of waveform variability.[19] As well as this, the grouping of FPs will make it easier for clinicians/investigators to find evidence of axonal conduction block, axonal conduction variability and occurrence of doublets occurring in the MUs.[19] The matched filtering alignment described Section 4.1.4 is an effective way of aligning FPs; the algorithm iteratively finds the specific location of each FP that best aligns its morphology with the other FP waveforms it has been grouped with.

Aligning FPs in this way make it easy to identify variations in individual phases, such as amplitude, as well as identifying features caused by axonal conduction block (see top of Figure 4-33). The parameters ΔA_{P1}, and ΔA_{P2} are extracted from the first two phases of the waveforms; ΔA_{P1} = amplitude difference of phase 1, ΔA_{P2} = amplitude difference of phase 2. Aligning FPs by their start point, however, provides a more effective means of assessing the variability of FP duration and waveform speed (see bottom of Figure 4-33).

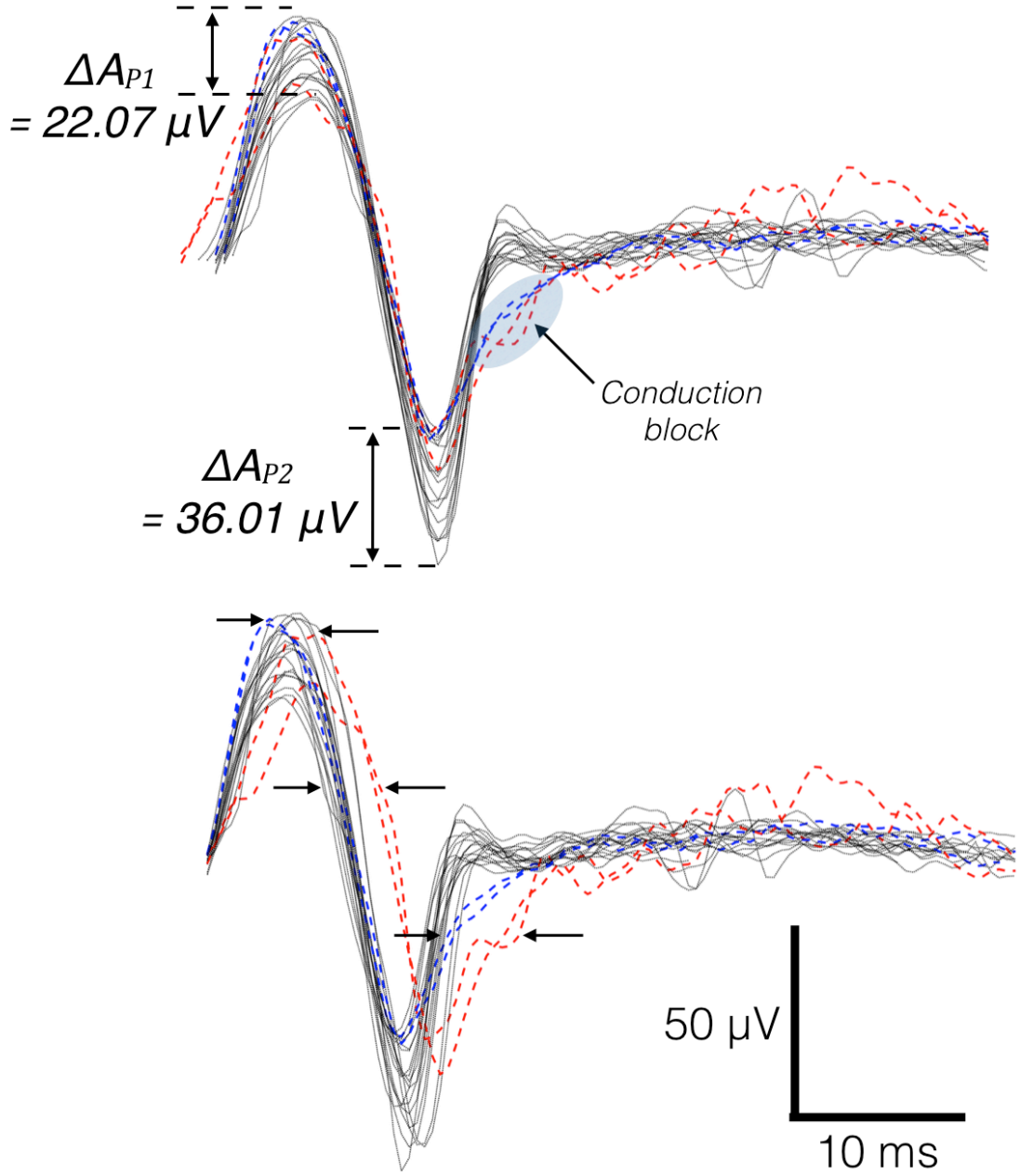


Figure 4-33: Grouped FPs aligned by matched filtering and by start point. *(Top) The FPs have been aligned using the matched filtering algorithm, giving insight into the variation of specific phases and the presence/absence of particular features. The absence of the third phase shown in the red and blue waveforms is suggestive of axonal conduction block. The parameters ΔA_{P1} and ΔA_{P2} refer to the amplitude difference in phase 1 and phase 2, respectively. (Bottom) The FPs have been aligned using the start point of the FPs, giving insight to the variation of duration amongst the grouped FPS. The blue and red waveforms, previously identified as being affected by axonal conduction block, show very different waveform speeds; red FPs are slower than the majority of FPs belonging to this group.*

The FPs shown in Figure 4-34 have also been aligned by the FP's start time, extracting the parameters Δt_{P1}, Δt_{X1}, and Δt_{P2}; Δt_{P1} = time difference of peak of phase 1, Δt_{X1} = time difference of zero crossing point 1, Δt_{P2} = time difference of peak of phase 2.

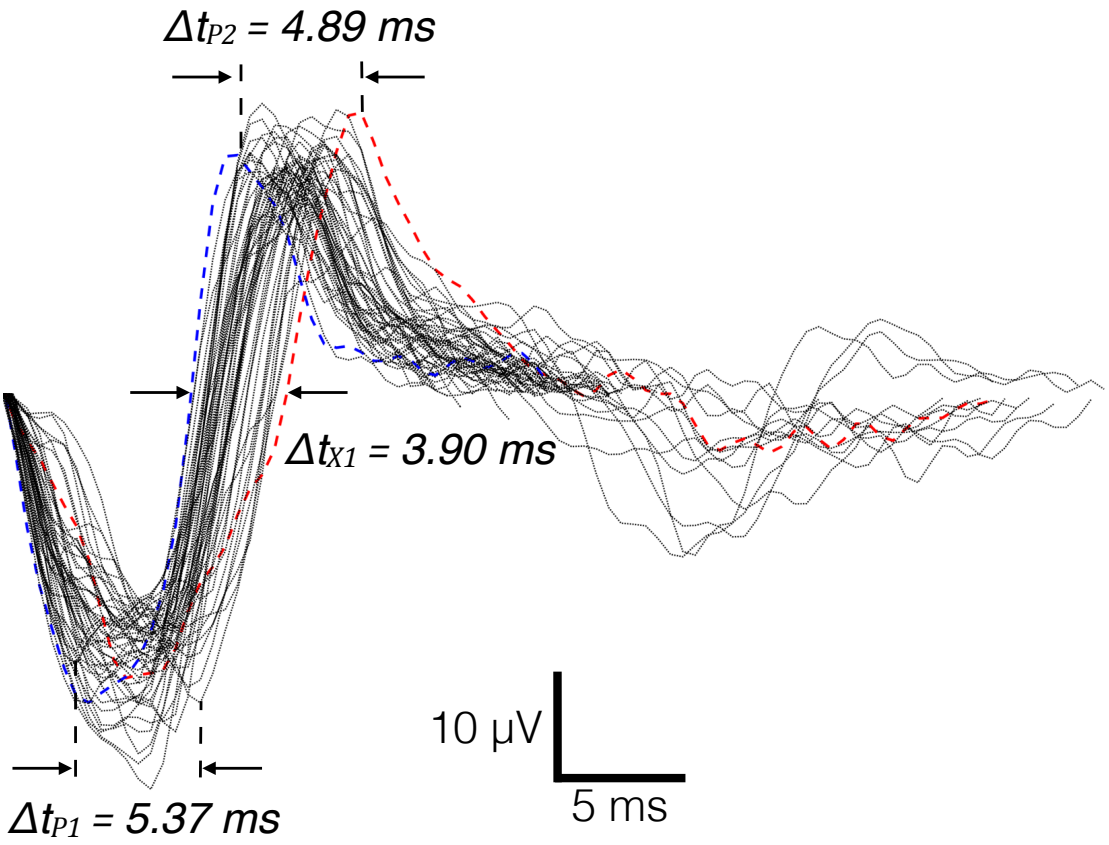


Figure 4-34: Grouped FPs aligned by start point showing time variations. *The FPs have been aligned by their start point, highlighting the temporal difference between waveforms. The red and blue FPs show the extremes in temporal difference for this group of FPs.*

Each group of FPs is aligned using matched filtering, and averaged to produce a template of the FP group (see top of Figure 4-35, as previously described in Section 4.1.4). These templates are representative of the FP morphology produced by each fasciculating motor unit, and are saved to be used as a reference when investigating future recording sessions from the patient.

There are more analytics and metrics that can also be outputted from the gathered data other than just waveform variability; for example, the neuronal firing rate of a motor unit can also be calculated be examining the inter-spike interval of the groups FPs as shown in Figure 4-35.

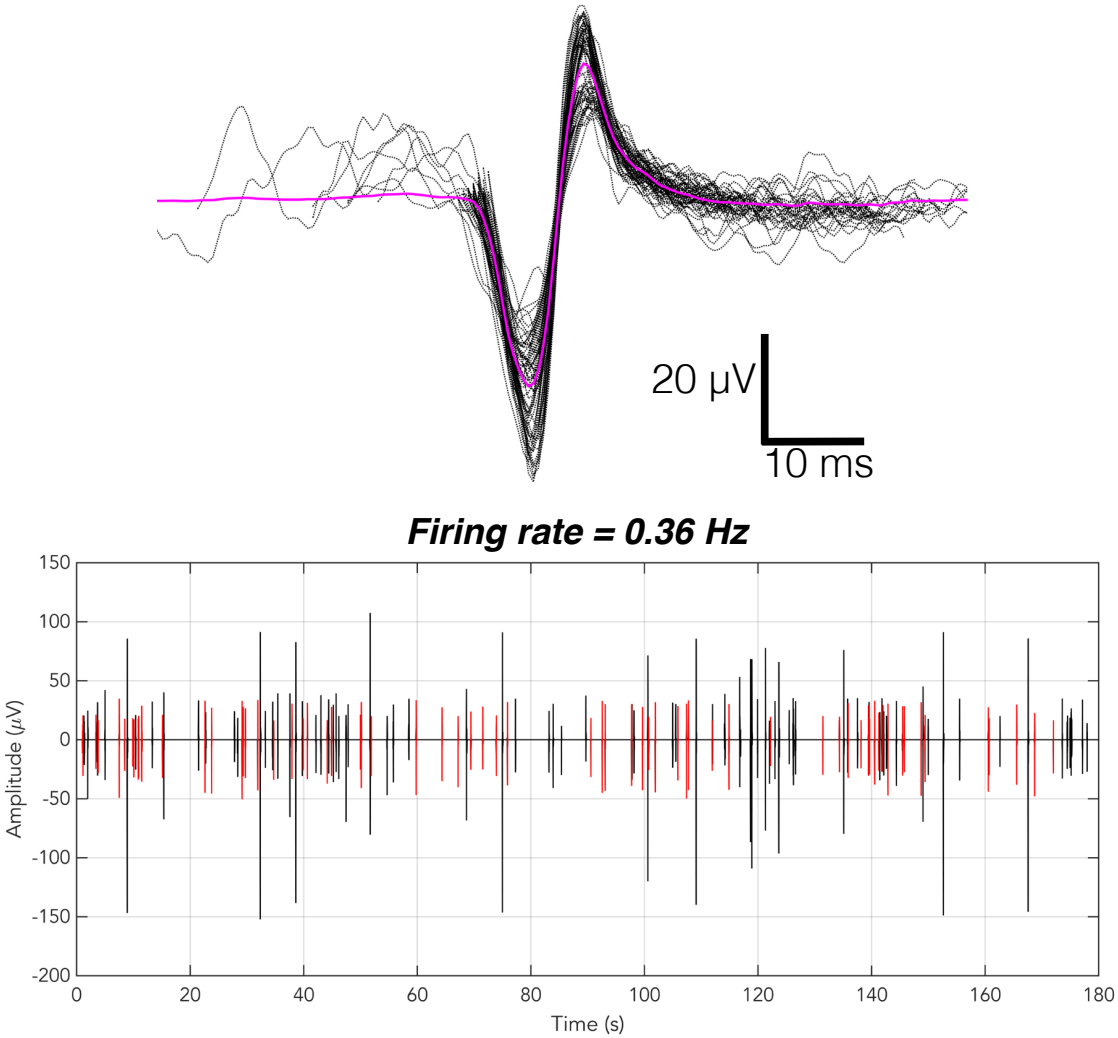


Figure 4-35: Template creation and firing rate calculation of individual motor unit. *Grouping of FPs allows for firing rate to be calculated for individual motor units. (Top) The grouped FPs are aligned and averaged to create a template (pink). (Bottom) The grouped FPs from the MU are highlighted in red and the 'super-channel' is shown in black. The firing rate for the individual MU is calculated to be 0.36 Hz (65 FPs/180 s).*

4.3 Conclusion of Part 2

In part 1 and part 2 of the data processing pipeline, I have taken raw EMG data and processed it to produce the outputs just discussed. Namely, that fasciculation potentials can be detected, grouped by morphology, and aligned so that parameters which describe a MU's variability can be extracted.

The matching of FPs by their morphology is a difficult task; attempting to match inherently different and variable waveforms by their similarity. The variability of FP morphology extends beyond just a variance in amplitude or

duration; entire components or phases of the waveform can be extremely exaggerated or missing entirely (possibly due to axonal conduction block), or present in one FP and not seen in other FPs from the same motor unit. This waveform variability, as well as the polyphasicity of the FP waveform, is increasingly seen as the disease progresses.[23] This variability, coupled with the reinnervation of motor units from neighbouring motor neurons, means that the morphology of ALS patient FPs can be quite unpredictable. For these reasons, common spike-sorting methods for detecting and sorting spikes/decomposing superimposed MUAPs that rely on a database of pre-existing waveform templates[36,66,72] would not be suitable for sorting FP waveforms from ALS patients.

The first method I have explored for grouping FP waveforms is the 'K-means' clustering method presented in Section 4.1.1; data clusters are formed based on the Euclidean distance of data-points to cluster centres. The data-points used for clustering are features extracted from the detected FP waveforms, such as amplitude, duration, and area-under-the-curve. These features are subjected to principle component analysis (PCA), a dimensionality reduction algorithm, which constructs new variables called 'principle components'; the first principle component expresses the majority of the variance in the dataset. Clustering the data in the principle component space, instead of the original feature space, helps remove any redundant information in the dataset and reveal any possible relationships/correlations between features. The clustering algorithm requires the user/operator to have *a priori* information on the number of cluster centres the dataset will have and there is no way of knowing this information without first inspecting the dataset, or reiterating the algorithm after reviewing the output. I have developed a framework termed *'cluster-and-seed'* to overcome this; datapoints that lie outside of a distance threshold from their assigned cluster centre are averaged to form a new cluster centre, which is then 'reseeded' into the clustering algorithm. The *cluster-and-seed* framework then iterates until every datapoint is within the distance threshold of its assigned cluster centre.

The second method I have developed for grouping FP morphologies is 'matched filtering'. The method works by convolving one waveform with the time-reverse of another and inspecting the resulting impulse response; the peak of the impulse response and its symmetry determine how well the two waveforms match. This can be potentially very computationally expensive, especially if there is a very high number of FPs to examine. The 'shuffling' framework I have developed alleviates the computational load of the matched-filtering algorithm, but it still remains more computationally expensive than clustering.

Clustering provides a fast, efficient method of grouping FP waveforms, however it can often make mistakes; FPs of different waveforms can express very similar features which can lead to the incorrect clustering of data. A high selectivity (low false positive ratio) is very desirable when grouping FP waveforms; wrongly grouped FPs will negatively affect the analytics outputted by the data processing pipeline.

The two grouping methods described here have been combined in order to successfully and efficiently group FPs; clustering is used to quickly group the data into smaller data clusters, and then matched filtering is applied to examine the morphologies of each data cluster, removing any wrongfully grouped FPs.

The peak of the impulse response produced by matched filtering also provides the location of the where the two convolved waveforms best match. This location is used to align the groups of FPs. The aligned FPs are overlaid and averaged to form a template representative of that group. Matched filtering is used again to group the templates of each FP group to reduce the number of templates and maximise the pipeline's efficiency. The groups of templates are manually inspected at this stage of the pipeline, and any wrongfully-unmatched templates are paired together. Finally, the underlying FPs that have formed each template are grouped together, which are then exported from the pipeline for analysis. The FPs can be aligned using matched filtering to highlight the presence/absence of particular waveform features (e.g. due to axonal conduction block) or aligned from the starting point of the FP to observe the variability in the waveform duration.

I have presented in this thesis a paradigm by which to detect and group fasciculation potentials by their morphology. The 'data-processing pipeline' that I have written takes signals recorded by HD-sEMG grid arrays, conditions them and checks them for quality. FPs are detected using a novel spike detection algorithm. Detected FPs are then grouped by their morphology, aligned, and a template is produced to be used as a reference when examining future recordings. Figure 4-36 summarises the work presented in Chapter 3 and Chapter 4. This work has provided clinicians with a means to producing analytics and metrics related to the morphology of FPs, and thus is a step towards quantifying the progression of ALS and obtaining a better understanding of its pathology.

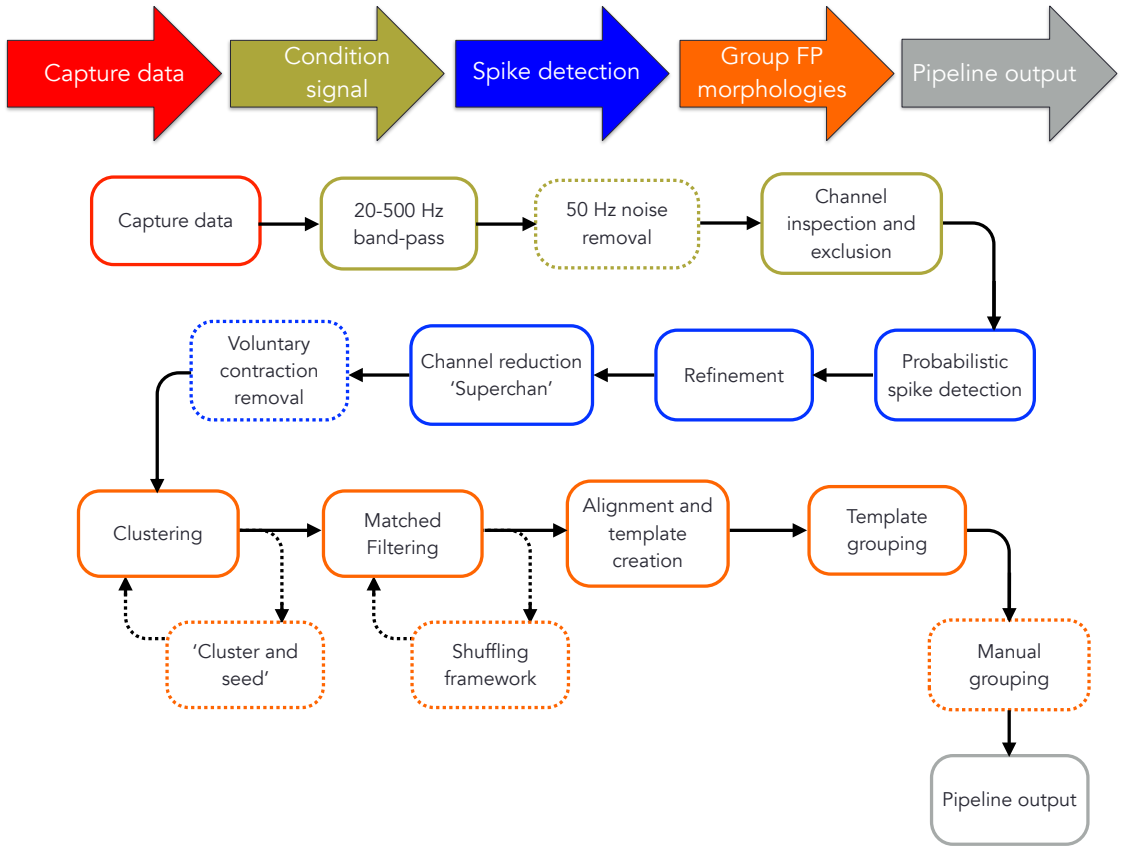


Figure 4-36: Full-data processing pipeline summary flowchart. *The flowchart shows the steps of each of the key stages in the data processing pipeline, from capturing the data to conditioning the signals, detection of FPs, grouping FP morphologies, and finally outputting the results.*

Chapter 5
Surface Potential Quantification Engine

The data-processing pipeline I have developed, presented in Chapter 3 and Chapter 4, takes electromyography (EMG) signals recorded from ALS patients, detects fasciculation potentials (FPs), and groups them by their morphology. This work has the potential to become a powerful tool for aiding clinicians and investigators in their research into ALS progression.

I have written the data-processing pipeline into a stand-alone application with a user-interface named '*Surface Potential Quantification Engine*' (SPiQE). SPiQE forms a collaboration between the data-processing pipeline I have developed for grouping FP waveforms and the collection of High-Density surface EMG (HD-sEMG) data by clinicians at King's College London (KCL). This chapter presents each panel of the application's user-interface (UI); each panel is responsible for each of the key steps of the data-processing pipeline (signal conditioning, FP detection, clustering, matched filtering and alignment, grouping of templates). Each panel of the UI consists of buttons that carry out specific functions of the data-processing pipeline, making it easy for users to operate and process EMG recordings. The UI includes common settings/parameters that are suitable for general use, but can also be tailored to the user's requirements. Users can navigate through each panel of the UI using *'next'* and *'back'* buttons at the top right-hand corner and top left-hand corner of each panel, respectively. The SPiQE application begins with the START panel shown in Figure 5-1.

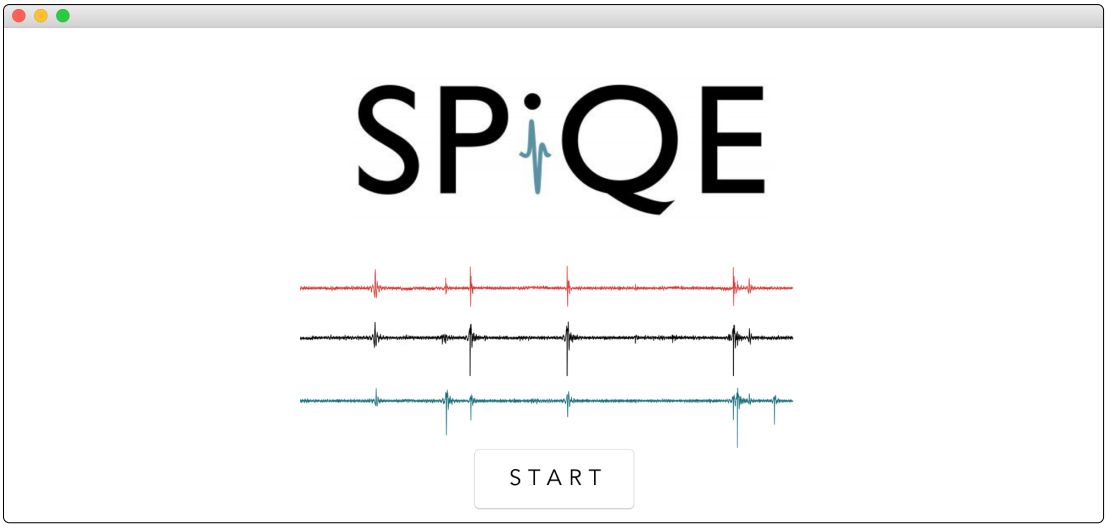


Figure 5-1: START panel of Surface Potential Quantification Engine (SPiQE). *The SPiQE application will begin when the user presses the START button.*

5.1 Channel Selection

The SPiQE application begins with the user being prompted to select the channels they wish to analyse from the HD-sEMG grid. The user can select to use the entire 8x8 grid (all 64 channels), the inner 6x6 grid, or the inner 4x4 grid using the radio buttons on the right-hand side of the panel (see Figure 5-2). The user can also choose a custom layout, by manually selecting the channels they wish to analyse from the grid of radio buttons in the centre of the UI, as shown in Figure 5-2. The list of selected channels will appear on the left hand side of the UI. Pressing the next button in the upper right hand corner will take the user to the next step in the application.

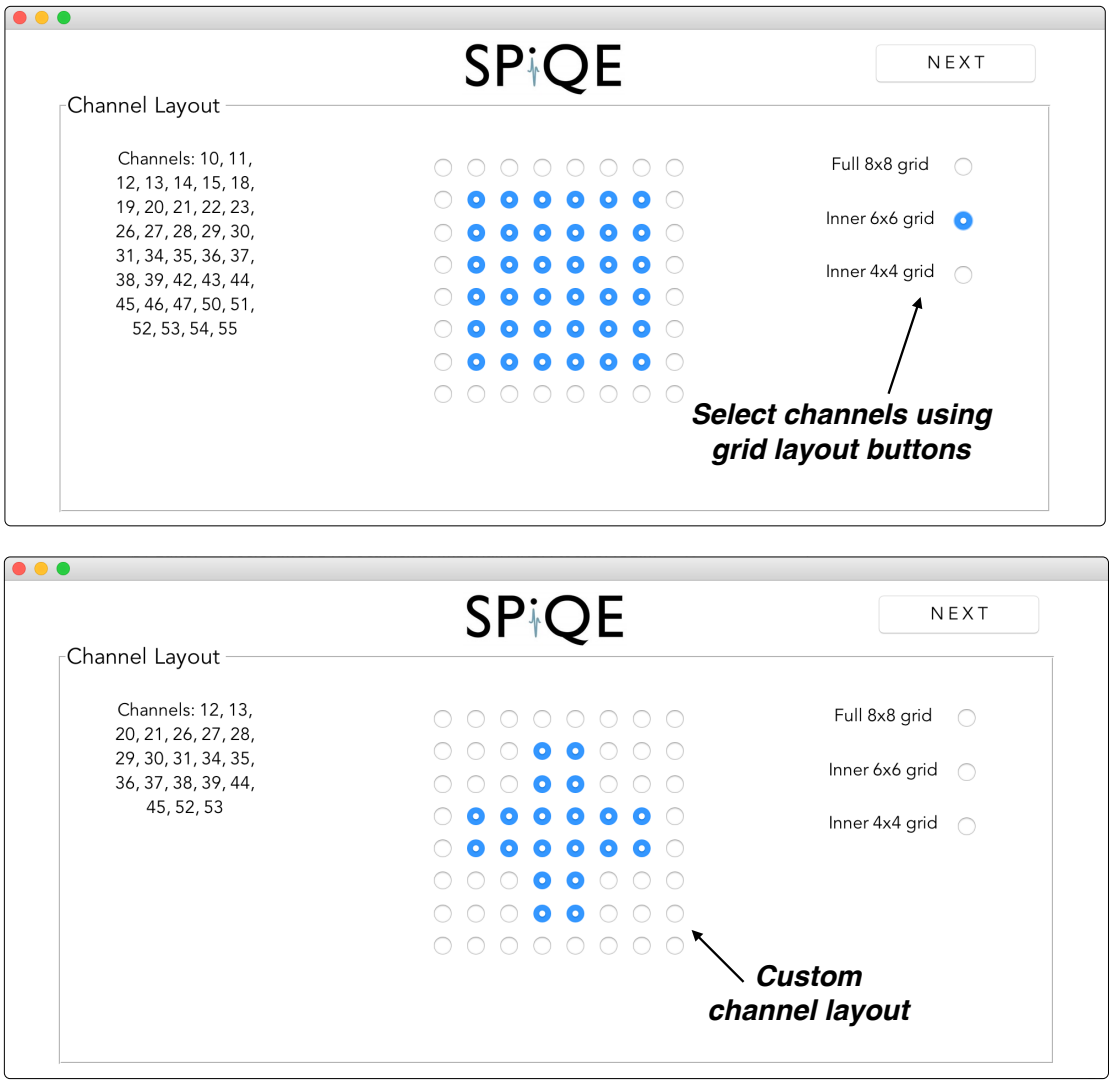


Figure 5-2: Channel select panel of SPiQE. *Users can either select pre-set channel layouts (e.g. 8x8, 6x6, 4x4 grids) or choose a custom layout by selecting the individual channels they wish to examine.*

5.2 Load Data and Condition Signals

The user can either load in the HD-sEMG data as a poly5 file (the format recorded by TMSi Refa, see Section 2.2.2) or as a .mat file (MATLAB file format). Once loaded, the user can filter the data to remove any unwanted frequencies (a blank or '0' in these fields will switch that particular field off). The user can also check the channels for abnormally high levels of 'activity' (see Section 3.2.3), which can then be removed using the *'Omit noisy channels'* button. If needed, the user can reset the data to the state it was in when it was originally loaded into

the UI, removing any filters that have been applied and re-including any channels that have been removed. Figure 5-3 shows the UI for these processes.

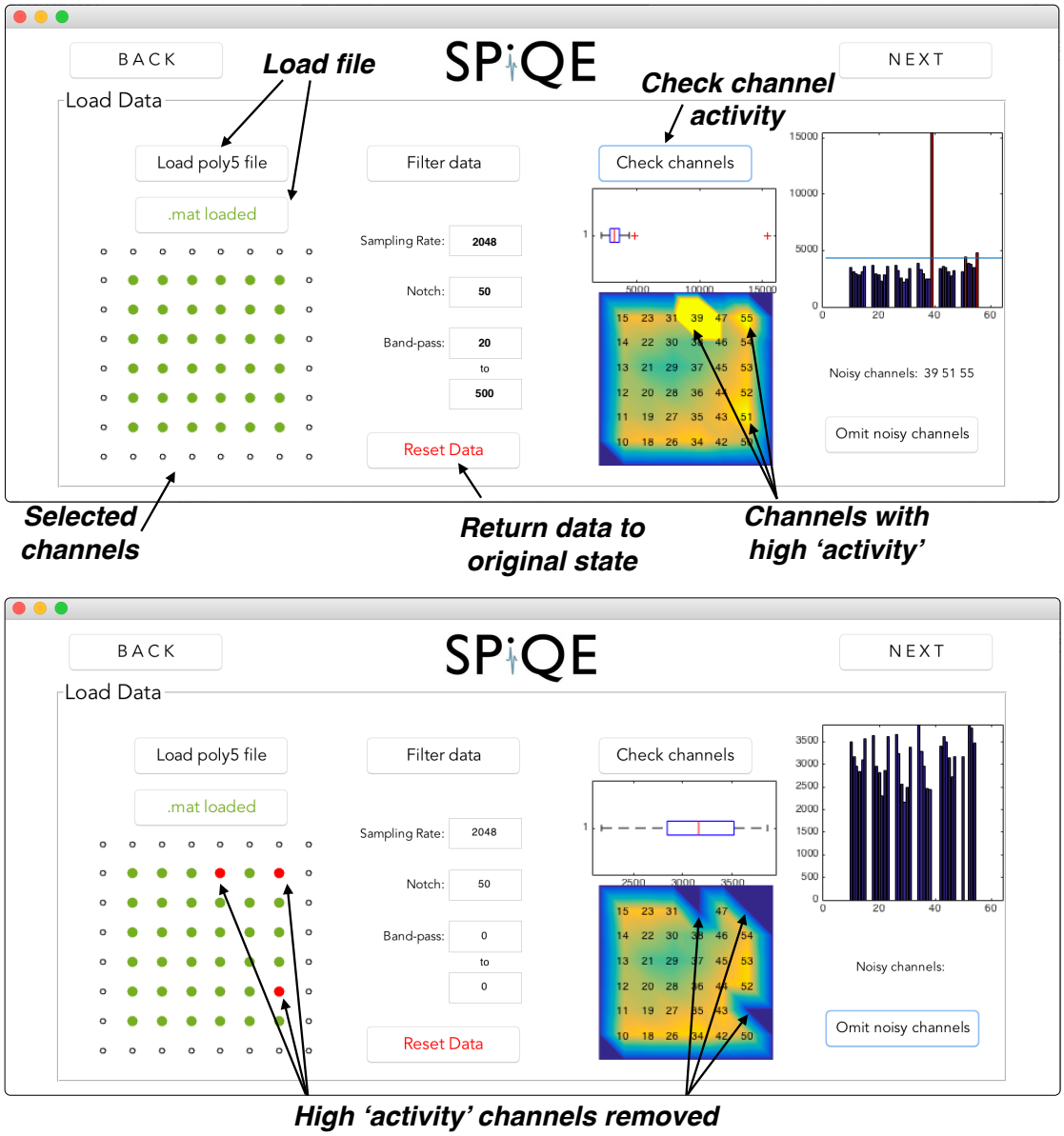


Figure 5-3: Load data and condition signals panel for SPiQE. *Data is loaded, filtered, and channels are inspected for quality. Noisy channels are omitted from further analysis.*

5.3 Fasciculation Potential Detection

The FPs are detected using the probabilistic spike detection algorithm described in Section 3.3.1. In this UI, the user can adjust certain variables such as the probability threshold used for detection and the amplitude threshold applied to the FPs that have been detected, as shown in Figure 5-4. The 'super-channel' is

constructed using the FPs that have the largest amplitude across the HD-sEMG grid (see Section 3.3.2). The occurrence of FPs detected in each channel is presented as a heat map and a bar chart. The UI also gives the user the option to listen to the super-channel as an audio file; it is common for electrophysiologists to listen to EMG recordings through a loudspeaker as a way of diagnosing different neuromuscular disorders.[32] The variable *'Fractime'* allows the user to only examine a certain percentage of the recording length. Figure 5-4 shows the user has set Fractime = 0.1, corresponding to the 180 seconds of data seen in the super-channel (full recordings are 30 minutes long). This can be beneficial to the user if they wish to investigate the effect that editable variables have on the detection of FPs before committing to processing the entire signal, which can be computationally expensive.

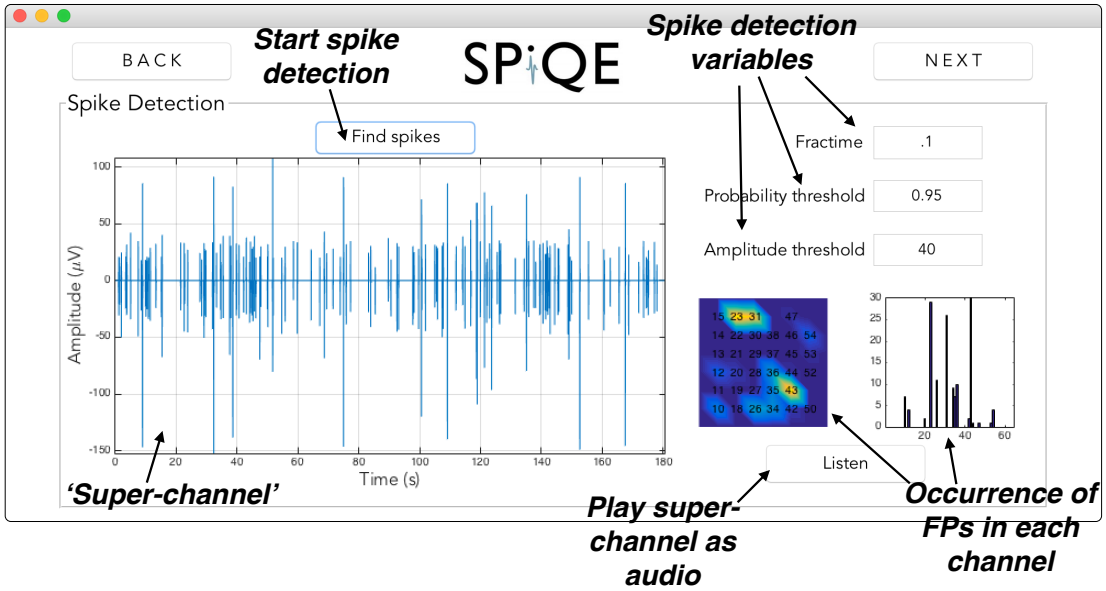


Figure 5-4: Spike detection panel for SPiQE. *FPs are detected across the selected channels, and the 'super-channel' is constructed from the FPs that have the largest amplitude across the grid. The user can input their own parameters for detecting FPs.*

5.4 Clustering

As shown in Figure 5-5, the user can select which features to extract from the FPs via a pop-up menu that will appear after pressing the *'Select features'* button. Upon pressing the *'Cluster data'* button, the FPs are split by polarity and principle component analysis (PCA) is performed on the extracted features. The first three

principle components (PCs) of the dataset are then clustered using K-means clustering and the 'cluster-and-seed' framework described in Section 4.1.1.4; the clustered data for the positive and negative FPs are displayed on separate axes, which the user can view from different viewpoints of the different PCs.

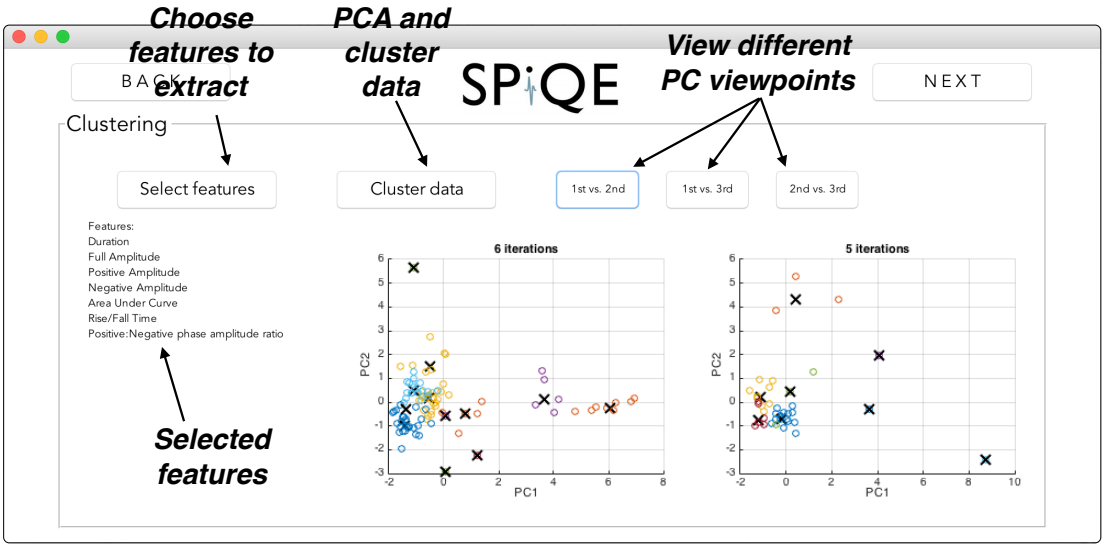


Figure 5-5: Clustering panel for SPiQE. *Features for extraction are selected via a pop-up-menu, which then undergo PCA and K-means clustering using the cluster-and-seed framework.*

5.5 Matched Filtering and Alignment

As shown in Figure 5-6, the morphology of each cluster group is then checked using the matched filtering algorithm and the shuffling framework described in Section 4.1.2. At this stage, incorrectly clustered FPs and solitary FPs (unmatched with any other waveform) are rejected; in the example in Figure 5-6, the number of positive FP groups is reduced from eight to five and the number of negative FP groups is reduced from 12 to 10 at this stage. Rejected FPs can be matched again using matched filtering in an attempt to re-include any incorrectly rejected FPs back into the data-processing pipeline; in the example in Figure 5-6, seven FP groups are formed from the rejected FPs, with twelve FPs being left unmatched.

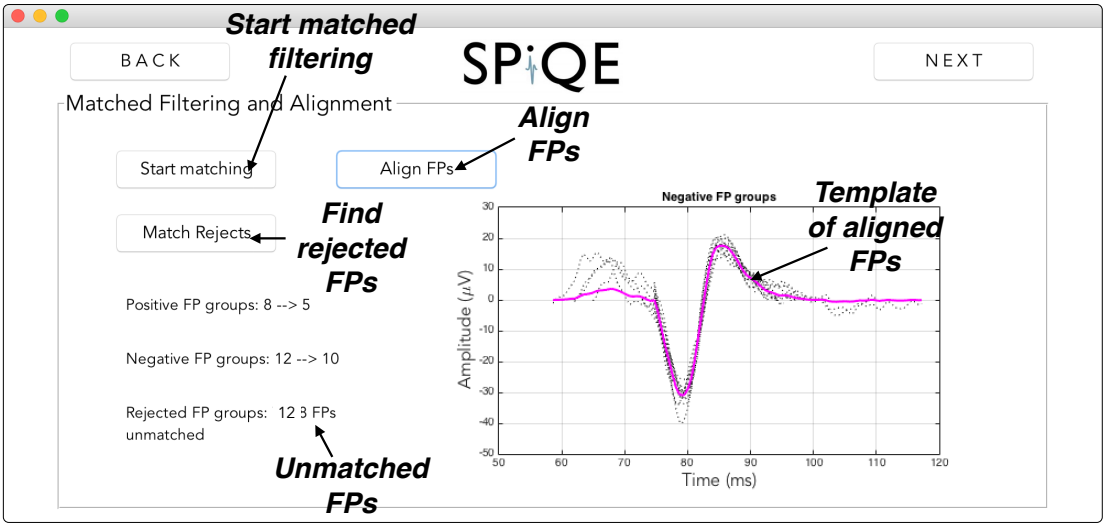


Figure 5-6: Matched filtering and alignment panel for SPiQE. *Cluster groups are checked using matched filtering, which rejects any incorrectly clustered or solitary FPs. Rejected FPs can then undergo matched filtering to re-group FPs that were originally grouped incorrectly. Groups of FPs are then aligned and the alignment of each group is displayed for the user in turn.*

5.6 Template Grouping

The templates formed from each FP group are then grouped together using matched filtering, in order to combine groups of FPs that may have been split into smaller subgroups (see Section 4.1.5). Any templates left unmatched, or groups of FPs that have been grouped into smaller subgroups, can then be manually grouped together by the user; pressing the *'Manual grouping'* button will open a pop-up-menu that the user can use to select template groups to group together. The user can view each template group by cycling through them using the arrow buttons indicated in Figure 5-7. The number above each template indicates how many FPs have been aligned to form that template. Once all templates have been grouped and the user is satisfied, pressing the *'Finish'* button will combine the groups of underlying FPs that have been used to form each template. These final groups of FPs are then realigned to form new templates, representative of the motor unit they have originated from. The output of SPiQE, containing the processed signals, detected FPs in each channel and their locations, the 'super-channel', the final groups of FPs, the alignment of the FP groups, and the averaged template for each FP group, can then by saved by pressing the *'Save output'* button.

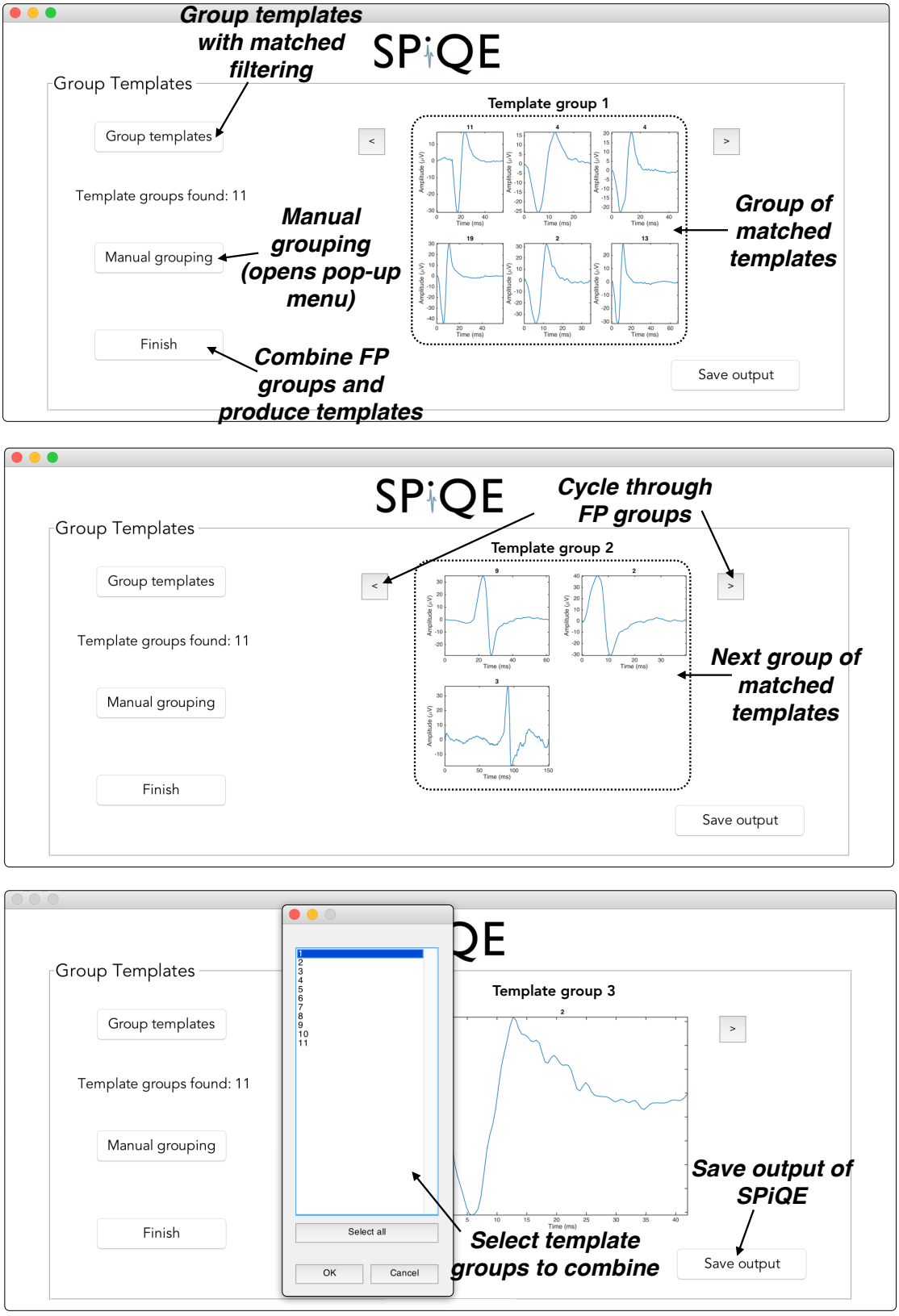


Figure 5-7: Template grouping panel for SPiQE. *Templates are grouped together using matched filtering and manual intervention from the user. When all templates have been successfully grouped, the 'Finish' button will combine the underlying groups of FPs and align them.*

5.7 Conclusion

The data processing pipeline described in Chapter 3 and Chapter 4 has been realised into a functional tool that can be used to condition EMG signals recorded by HD-sEMG grid arrays, detect FPs, and group them by their morphology. The SPiQE interface can be compiled into a stand-alone application without the need for the user to own or install MATLAB or be expert in programming. The user has control over different parameters that will affect the output of the pipeline, which they can adjust and tune should they wish to. The user can then save the output of the SPiQE software.

Variation in morphology is not the only metric that can be gained from the detection of FPs when investigating the progression of ALS. Collaborators at KCL are investigating the effect that progression has on the inter-spike-interval of the fasciculating motor units as well as on the occurrence of FPs across the entire bulk of the muscle using the probabilistic spike detection algorithm I have developed (see Section 3.3.1). Researchers at KCL have also developed a novel method for the exclusion of voluntary activity (see Section 3.3.3) from HD-sEMG recording, termed *AVID* (Active Voluntary Identification). This work will be incorporated into the SPiQE application as part of the continuing collaboration with KCL.

Chapter 6
Wireless Multiple Channel Device

In this chapter I describe the development of a wireless multiple-channel device for capturing EMG data from ALS patients. Knowledge gained from the patient studies at King's College London (KCL) and the development of the data-processing pipeline has informed the requirements for the design of an 8-channel wireless device for capturing EMG data. The high data throughput required by the device is made possible through the combination of Bluetooth transmission and local storage of data. Two versions of the device have been designed and tested; after assessing the performance of device design v1, changes were made to for device design v2, which are detailed in this chapter.

6.1 Device Design – Version 1

6.1.1 Requirements

Ideally, patient recording sessions would take place as often as possible to ensure that any changes in fasciculation occurrence or fasciculation potential (FP) morphology were being captured as continuously as possible. However, it is ambitious to expect a patient to travel for a recording session very frequently, particularly with their increasing disablement[iii] (even with the assistance of a

[iii] *As the disease progresses, the patient will become increasingly disabled as they lose the ability to perform motor tasks. In several cases, the clinician at KCL has had to make home visits to take EMG recordings from patients who have become unable to travel throughout the longitudinal study*

carer). It is also unrealistic to expect the clinician/operator to travel to different locations to perform the recordings at patient's homes especially if there is ambition to drastically increase the size of the cohort.

With the desire to record more frequently from a much larger patient cohort, there is a requirement for new instrumentation that can capture the FPs from ALS patients. The new device must be easy for the patient/carer to apply themselves, able to capture multiple-channel data (to ensure the same spatial resolution obtained by the current system), wireless (for ease of application), and have a resolution and sampling rate better or comparable to current industry standards.

6.1.2 Device overview

Figure 6-1 shows an overview of the device, demonstrating how the device will take a signal from an analogue input through to storing and transmitting the digitised data.

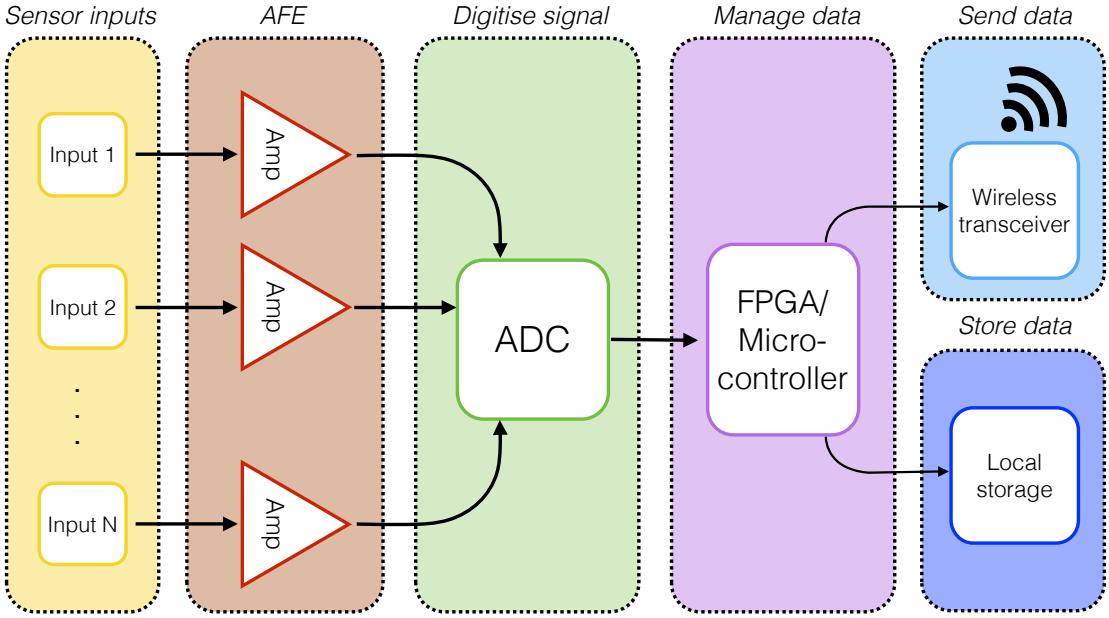


Figure 6-1: Device design overview. *The device will require sensor inputs, pre-amplification at the analogue front end (AFE), and digitisation of the analogue signal by means of an analogue-to-digital converter (ADC). The device will also require a means of controlling the device and managing the stream of digital data, such as a FPGA or microcontroller. The device will require a transceiver for wirelessly transmitting data, as well as storage for storing data locally to the device.*

The design of the device will require sensor inputs, pre-amplification of the analogue signal, an analogue-to-digital converter (ADC) to digitise the signals

being read, a field-programmable gate array (FPGA) or microcontroller to control the device, a power supply, a power management circuit, and a wireless transceiver. The device will also likely require some form of local storage, preferably removable such as a secure digital (SD) card. Not shown in Figure 6-1 is the device's power supply; each stage of the device will require a wireless power supply, such as a rechargeable battery.

I will now describe each stage in turn, detailing the components selected for each stage, how they are configured, and how they interact with each other.

6.1.3 Analogue front-end

6.1.3.1 Buffer

A unity gain buffer (voltage follower) is used at the sensor input, shown in the operational-amplifier (op-amp) configuration in Figure 6-2. The buffer op-amp is used to convert the high impedance of the input signal into a low-impedance output. Buffer op-amps allow the linking of different stages of circuitry without having to deal with mismatching impedances.

An ideal op-amp will have an infinite input impedance and a zero output impedance ($Z_{in} = \infty$, $Z_{out} = 0$). The high impedance input of the op-amp buffer means very little current will be drawn from the signal source. The op-amp's output is connected to its inverting input, forcing the op-amp's output to follow the voltage input but at the very low output impedance of the op-amp. The output signal from the op-amp therefore acts as a perfect voltage source. The buffer has 'unity gain' (gain = 1), as the op-amp configuration does not amplify the input signal.

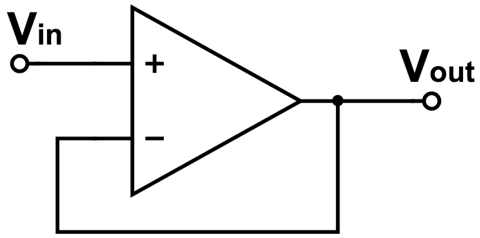


Figure 6-2: Op-amp configuration for a unity-gain buffer/voltage follower. *The inverted input of the op-amp is attached to the output, forcing the op-amp's output to follow the input voltage but at the low output impedance of the op-amp.*

6.1.3.2 Pre-amplification

The incoming analogue signal at the front end of the signal will require amplification before it is converted into a digital signal by the ADC. As seen in Chapter 3 and Chapter 4, the majority of FP waveforms recorded in the surface EMG signal will be in the sub-millivolt range. Without pre-amplification, the analogue signal will suffer from poor resolution when converted by the ADC. For example, an ADC with a 16-bit resolution (*two's complement*,[iv] see Appendix A) and a ±2.5 V reference will have a resolution of only 76.3 μV:

$$Resolution = \frac{2.5\ V}{(2^{15} - 1)} = 76.3\ \mu V \quad \textbf{(6-1)}$$

This resolution clearly isn't fine enough to record FPs with, let alone the fine intricacies of the waveform that would allow for the assessment of waveform variability. A non-inverting op-amp configuration is used to amplify the incoming analogue signal before it is digitised by the ADC, as shown in Figure 6-3. The non-inverting amplifier gain equation is shown in Equation (6-2). For example, using a gain of 100 (e.g. R_2 = 99 kΩ, R_1 = 1 kΩ) would significantly increase the resolution of the signal once digitised.

[iv] *When using two's complement, the first bit (most significant bit) is used to signify the signage (positive/negative) of the digitised value. For a 16-bit integer, this means only 15 bits are used to represent the signal value. This lowers the resolution of the ADC, but means that it can read both positive and negative values.*

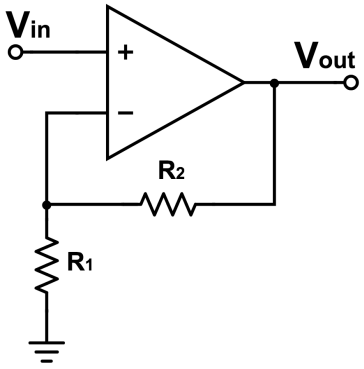


Figure 6-3: The op-amp configuration for a non-inverting amplifier.

$$A = \frac{V_{out}}{V_{in}} = (1 + \frac{R_2}{R_1}) \quad \textbf{(6-2)}$$

$$Resolution = \frac{2.5\ V}{100 * (2^{15} - 1)} = 0.763\ \mu V \quad \textbf{(6-3)}$$

6.1.3.3 Combined analogue front-end - LMP7702

The output of the buffer feeds directly into the input of a non-inverting amplifier, as shown in Figure 6-4. This configuration forms the analogue front-end (AFE) of the device for each sensor input.

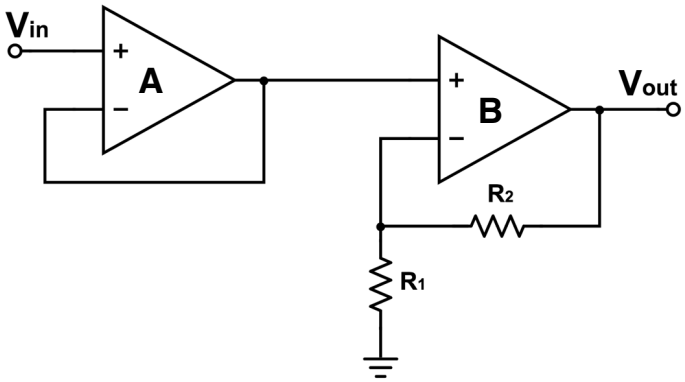


Figure 6-4: Analogue Front-End (AFE) of each sensor input. *Combining both the buffer input stage (A) with a non-inverting amplifier (B).*

The LMP7702 is a dual precision amplifier integrated circuit (IC), offering two op-amps within the same package. The LMP7702 op-amps have a high input impedance, low-offset voltage (max ±200 µV), and low input bias current (±200 fA), making them ideal for precision sensor interface applications. The LMP7702 op-amps are rail-to-rail input and output amplifiers, meaning they can amplify

the input signal to the maximum range of the IC's power supply. The buffer and non-inverting amplifier stages of each sensor input can be configured together using just one IC package, as seen in Figure 6-5.

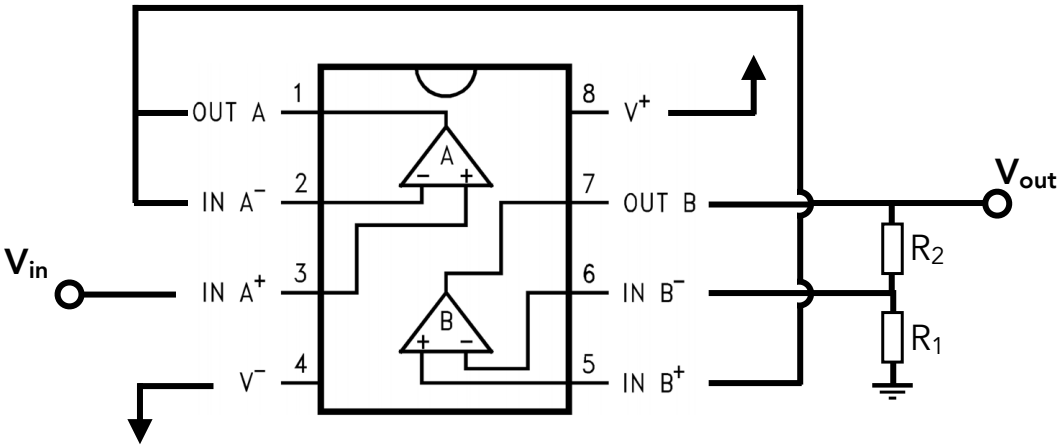


Figure 6-5: Dual precision amplifier LMP7702 used for the AFE of the device. *LMP7702 is used to combine both (A) the buffer and (B) the non-inverting amplifier op-amp configurations. A voltage divider network is used to provide the resistors R_1 and R_2. (Image adapted from the LMP770x datasheet, ©Texas Instruments).*[73]

The LMP7702 is powered by a ±2.5 V bipolar supply; the dual op-amp cannot amplify the analogue input signal outside of this range. Care should be taken when choosing the gain value to make sure the analogue signal is not 'clipped' by the rails of the amplifiers power supply. A high tolerance, high stability, voltage divider resistor network is used to provide R_1 = 1 kΩ and R_2 = 100 kΩ, giving the AFE a gain of 101 (see Equation (6-4)).

$$A = 1 + \frac{R_2}{R_1} = 1 + \frac{100\ k\Omega}{1\ k\Omega} = 101 \qquad \textbf{(6-4)}$$

The majority of FP amplitudes recorded by surface electromyography (sEMG) will be in the sub-millivolt range, and so a gain of 101 will not risk the waveform being 'clipped'; even very large FPs (i.e. 1-2 mV) will still be left with significant headroom within the power supply range, as shown in Figure 6-6. Amplifying the signal more than this is possible, and would achieve a finer resolution, but would reduce the bandwidth of the output signal.

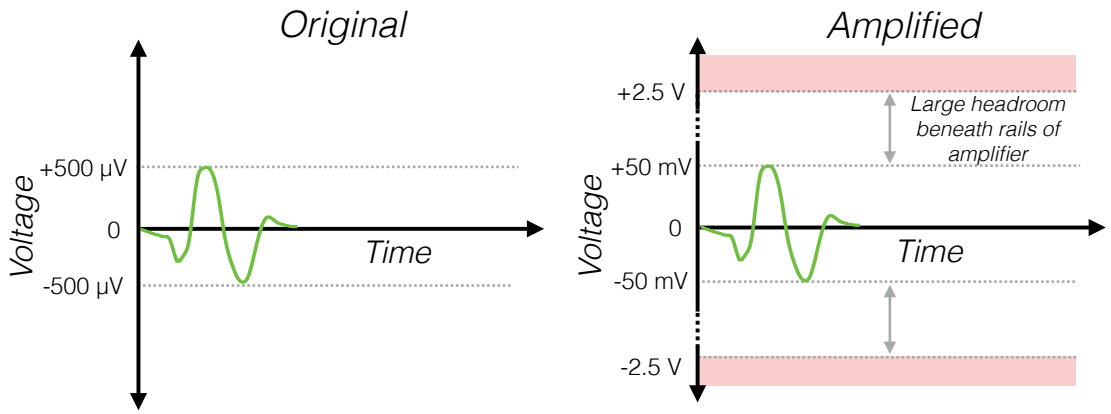


Figure 6-6: Diagram of amplification of a large FP. *The 1 mV (peak to peak) FP has been amplified to 101 mV, which still leaves a large amount of headroom beneath the power supply rails of the amplifier.*

Figure 6-7 shows testing of the AFE using 1 Hz sine waves at amplitudes 20 mV, 2 mV, and 200 µV. The signals have been measured using a PowerLab data acquisition system (see Section 2.4.2). The signal-to-noise-ratio (SNR) values for the input and output of the AFE are shown for each signal tested. The gain of the AFE has been calculated using the root-mean-square (RMS) value of the input and output signals.

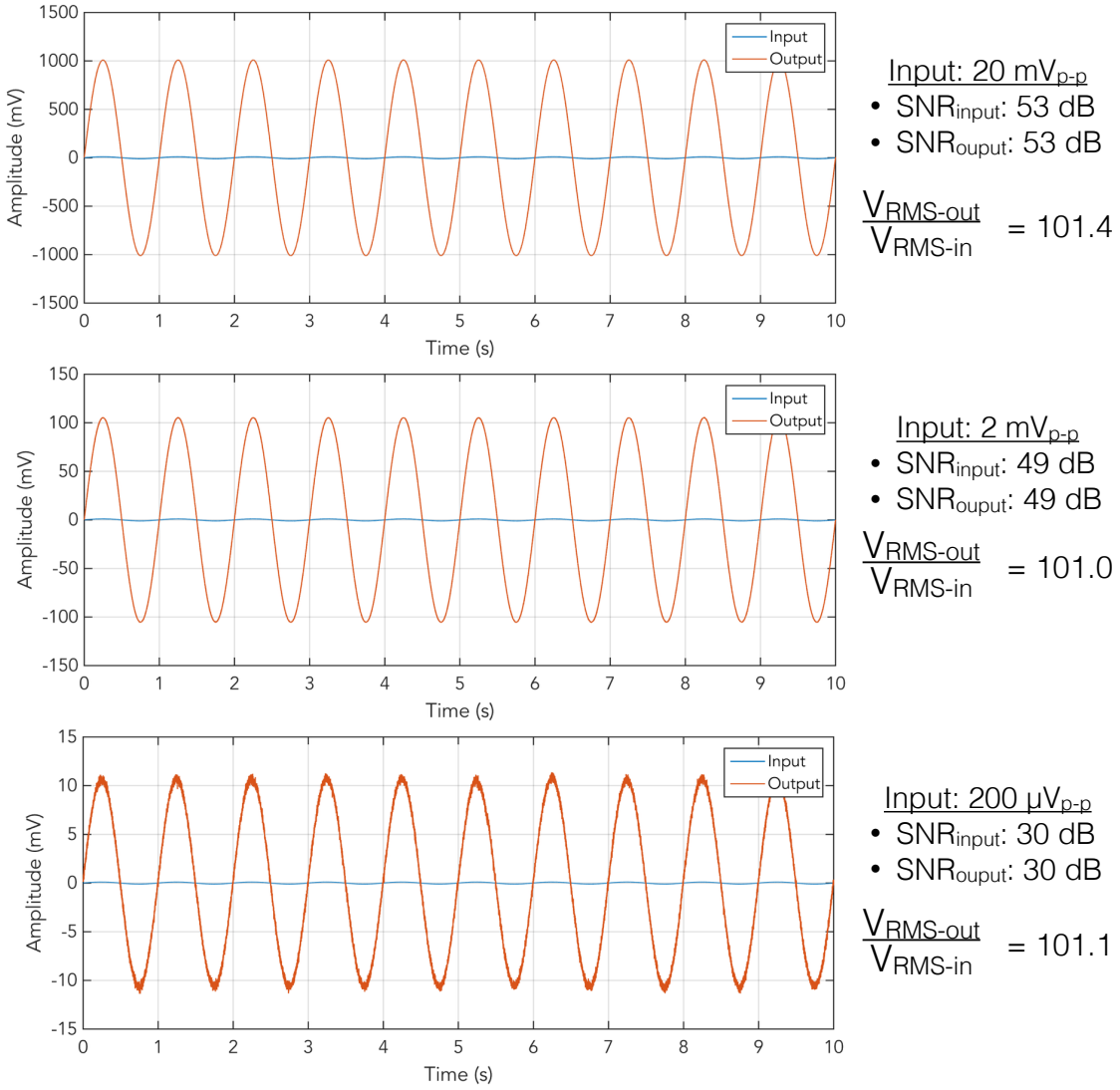


Figure 6-7: Output of signals amplified by the AFE. *1Hz sine waves of 20 mV, 2mV, and 200 µV are fed into the AFE. The AFE produces a gain of 101 as expected, and has a low-noise output (SNR of signal remains constant from input to output, any noise is likely to have come from the signal generator or from the data acquisition system).*

6.1.4 Analogue-to-digital converter selection – ADS1298

The ADC used in the device is the ADS1298 by Texas Instruments. The ADS1298 is an 8-channel, low power (0.75 mW/channel), low noise (4 μV_{p-p}) , 24-bit ADC with programmable gain amplifiers (PGAs). The sampling rate of the chip can be configured to be as low as 250 samples per second (SPS) in low-power mode and as high as 32 kSPS in high-resolution mode. The ADS1298 can be cascaded in a daisy-chain configuration to increase the number of channels of the system if necessary. The ADS1298 has an intended application for medical instrumentation such as ECG, EEG, and EMG. Each channel can be individually

configured to be enabled/disabled, have its own gain selection (1-12), or output internally generated test signals. The ADS1298 is a highly specified, configurable ADC for biopotential measurements and is a very suitable choice for this device.

The ADS1298 is configured (data rate, amplification, etc.) via its serial peripheral interface (SPI) port (see Appendix B) by an external device such as an FPGA or a microcontroller. Commands can also be issued to the ADS1298 over its SPI port to start/stop ADC conversion, power up/down the chip, and to reset the chip. Alternatively, the ADS1298 chip also has dedicated pins that can be used to issue these commands by a high or low voltage from the FPGA/microcontroller. The ADS1298 can also issue commands itself to external devices to let it know that the ADC has digitised the analogue signal and that it is ready to be transferred (DRDY = Data Ready). The pins used for communicating with the ADS1298 chip are shown in Figure 6-8).

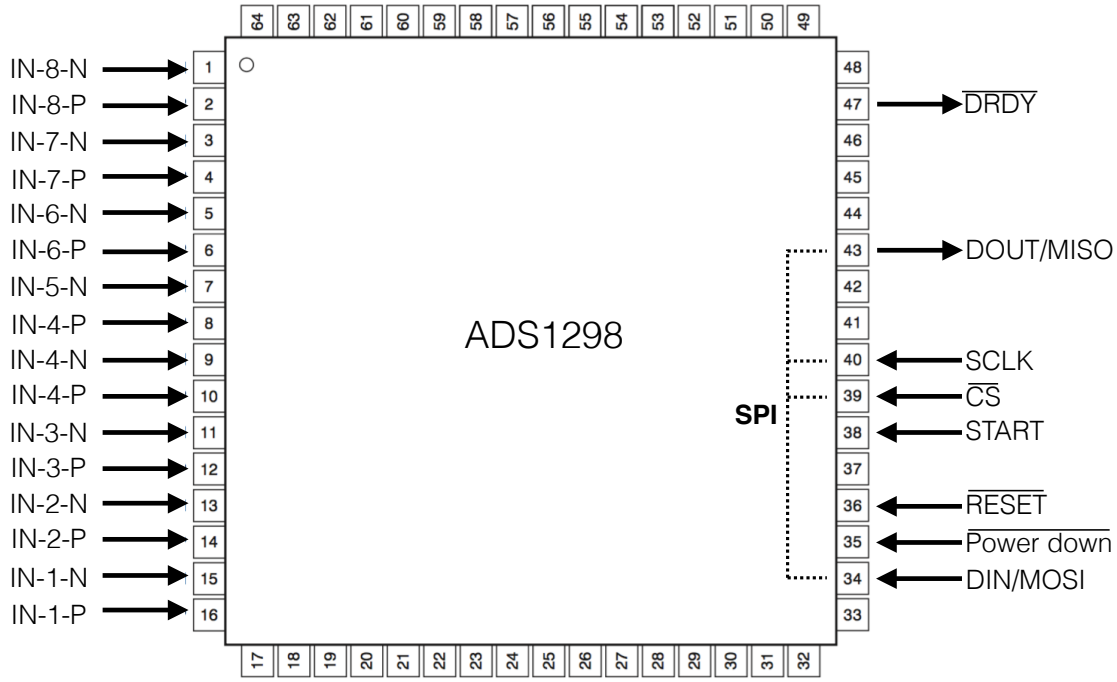


Figure 6-8: Key analogue input and digital input/output pins of the ADS1298. *Shown are the eight differential input channels (e.g. IN-1-P = Channel 1 positive input, IN-1-N = Channel 1 negative input). The SPI pins (DOUT = Data out, DIN = Data in) allow the ADS1298 to be configured and are also used to transfer digitised data. The DRDY pin (Data ready) indicates when digitised data are ready to be transferred. (Image adapted from ADS1298 datasheet, © Texas Instruments).*[74]

The ADS1298 can use either a unipolar or bipolar analogue power supply (AVDD). If running off a 3 V analogue supply, the internal reference of the

ADS1298 must be configured so that V_{Ref} = 2.4 V. If using a 5 V supply, the internal reference voltage can be configured to either V_{Ref} = 2.4 V or V_{Ref} = 4 V. The internal reference voltage of the ADS1298 has been configured to 2.4 V, which is discussed further in Section 6.1.4.2. The digital power supply (DVDD) must not exceed 3.6 V, as well as the voltage level of the logic inputs of the ADS1298. For example, if the SPI inputs from the FPGA/microcontroller exceed 3.6 V it will damage or 'burn' the ADS1298. Therefore, the FPGA/microcontroller's logic outputs need to either be configured or stepped-down to a safe voltage level.

6.1.4.1 Programmable-gain amplifier

The AFE of the ADS1298 consists of eight positive (IN-P) and eight negative (IN-N) inputs (see Figure 6-8). These inputs feed into the programmable gain amplifier (PGA), which is a differential input amplifier with programmable gain values of 1, 2, 4, 6, 8, and 12. The first stage of the PGA consists of two op-amps in a buffer configuration, one for the IN-P and one for the IN-N input of each channel. The output of each buffer op-amp feeds into the ADC, where the difference of the buffer outputs is digitised, as shown in Figure 6-9. The value of the gain resistor (R_1 in Figure 6-9) defines the gain of the PGA.

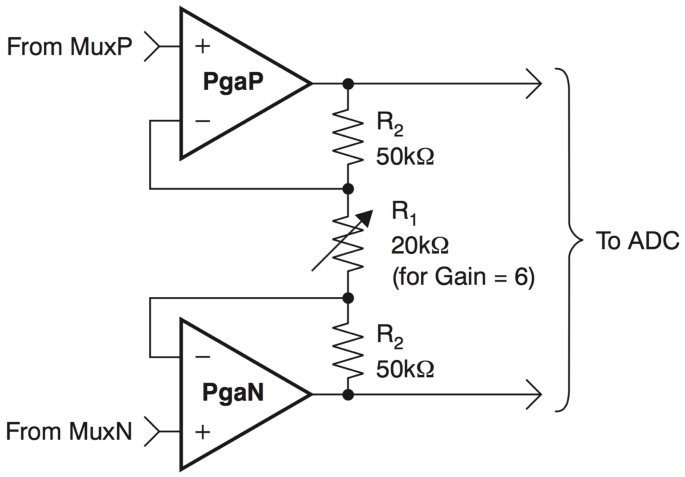


Figure 6-9: Programmable gain amplifier of ADS1298. *Differential input amplifier and variable resistor of ADS1298's PGA. (Image taken from ADS1298 datasheet, © Texas Instruments).*[74]

The circuit diagram in Figure 6-9 is actually just the first stage of a larger op-amp configuration known as an instrumentation amplifier, as shown in Figure 6-10. An instrumentation amplifier provides a very high common-mode rejection ratio

(CMRR) (ability of a device to reject signals common to both of its inputs) and so is an excellent choice when measuring signals in noisy environments. The CMRR of the ADS1298 is -115 dB (common-mode signals reduced by 1.78×10^{-6}), meaning its ability to reject common-mode signals that appear on both inputs, such as 50 Hz mains line noise, is extremely powerful.

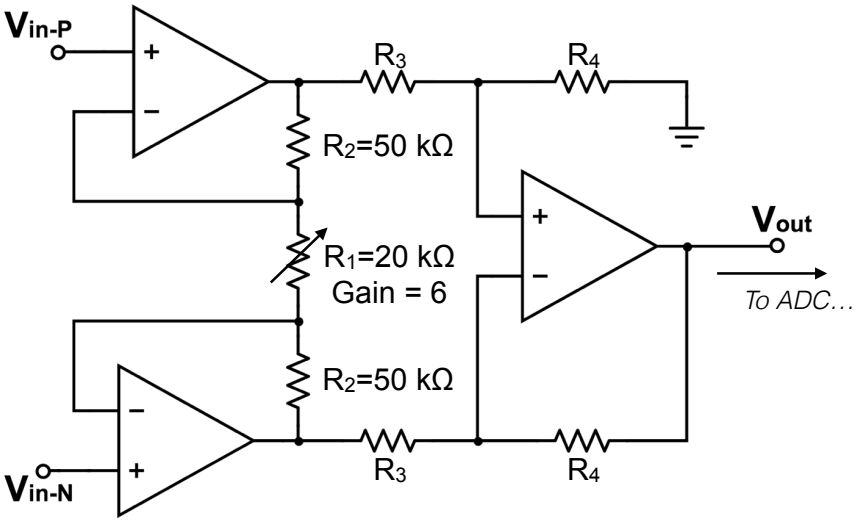


Figure 6-10: Instrumentation amplifier op-amp configuration of ADS1298. *Full op-amp configuration of instrumentation amplifier used to differentially amplify the positive (V_{in-P}) and negative (V_{in-N}) of each of the ADS1298's channels.*

The gain of the instrumentation amplifier can be calculated using Equation (6-5). If $R_3 = R_4$, then the transfer function of the system can be simplified to Equation (6-6). Hence, when the gain resistor is set to 20 kΩ in Figure 6-9, the gain of the system equals 6.

$$A = \frac{V_{out}}{V_{IN-P} - V_{IN-N}} = \frac{R_4}{R_3}\left(1 + \frac{2R_2}{R_1}\right) \quad \textbf{(6-5)}$$

$$A = \left(1 + \frac{2R_2}{R_1}\right) \quad \textbf{(6-6)}$$

The AFE op-amp configuration described in Section 6.1.3 has been combined with the PGA of the ADS1298 (see Figure 6-11) to demonstrate the overall AFE of the wireless device. The gain of the combined AFE can be calculated using Equation (6-7).

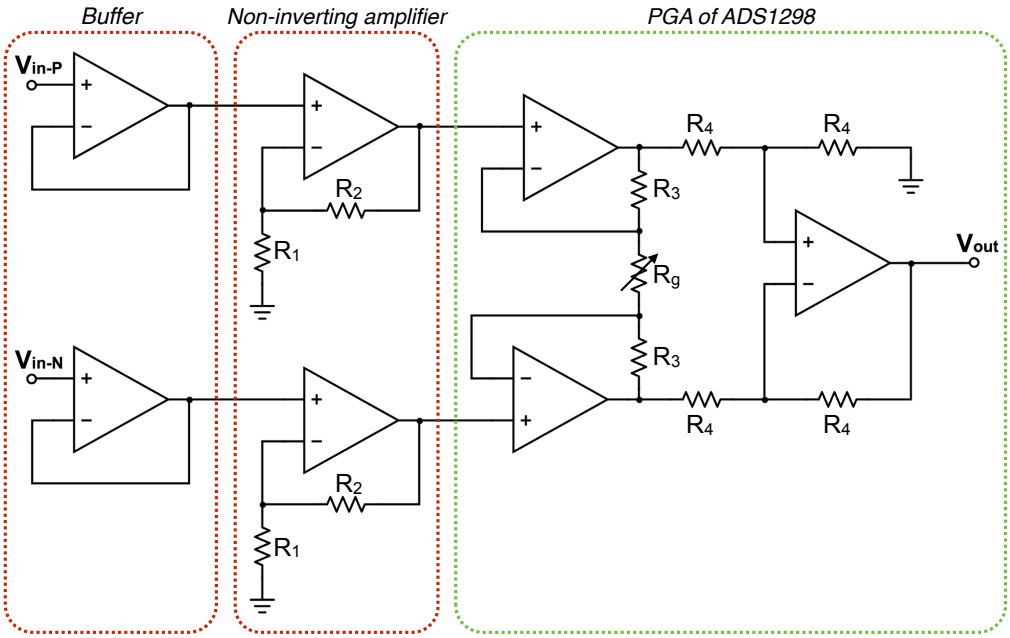


Figure 6-11: Complete AFE op-amp configuration of device. *Buffer and non-inverting amplifier configuration (red) used at the sensor input of the device combined with the instrumentation amplifier of the ADS1298's PGA (green).*

$$A = \frac{V_{out}}{V_{IN-P} - V_{IN-N}} = (1 + \frac{R_2}{R_1})(1 + \frac{2R_3}{R_g}) \qquad \textbf{(6-7)}$$

The combined amplification of the AFE and ADS1298 AFE needs to be large enough to ensure the resolution of the digitised signal can capture all the fine intricacies of the FP waveforms. However, if the gain is too large, it risks saturating the ADC, resulting in a 'clipped' signal as the input will have exceeded the reference voltage. In this thesis, the gain of the AFE has been set to 101 and the gain of the ADS1298 has been set to 12, using the full scale of the PGA, giving a total gain of 1212 (see Equation (6-8)). This was chosen in preference to A_{AFE} = 1212 and A_{PGA} = 1 for example, in order to maximise the bandwidth of the output signal. This combined gain will give the digitised signal a much finer resolution.

$$A_{total} = A_{AFE} A_{PGA} = 1212 \qquad \textbf{(6-8)}$$

6.1.4.2 Reference voltage

The analogue circuitry of the ADS1298 is powered by a bipolar 5V analogue supply (±2.5V), meaning the internal reference voltage of the ADC can be configured to either 2.4 V or 4 V. If the reference voltage is set to 4 V, the ADC has additional 'headroom' that it cannot use, as the output of the AFE cannot exceed

±2.5V (power supply of dual op-amp, described in Section 6.1.3.3). Therefore, the reference voltage of the ADS1298 is set to 2.4 V; this sacrifices some headroom of the input signal (as demonstrated by Figure 6-12), but means the resolution of the device is improved, as shown in Table 6-1. Table 6-1 shows the difference in resolutions of the converted data for both reference voltages of 2.4 V and 4 V, calculated using Equation (6-9) for two's complement data.

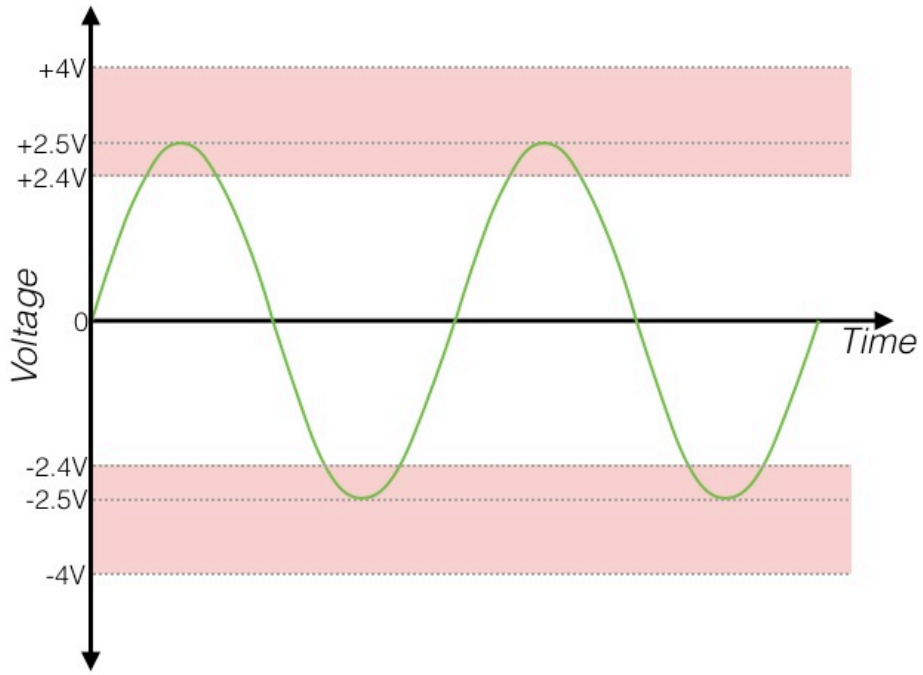


Figure 6-12: Effect of ADS1298 reference voltage. *The reference voltage of the ADC is set to 2.4 V, which sacrifices some headroom of the incoming signal (limited to ±2.5V) but improves the resolution of the converted data. (Red sections of plot are outside of reference voltage range).*

$$Resolution = \frac{V_{Ref}}{A_{total}(2^{bits-1}-1)} = \frac{2.4\,V}{1212\times(2^{bits-1}-1)} \qquad \textbf{(6-9)}$$

Table 6-1: ADC resolution of data for internal reference voltages 2.4 V and 4 V. *Amplified using A_{total} = 1212. Conversion of data is in two's complement.*

VRef	Resolution (bits)	Resolution (nV)
2.4	24	0.236
2.4	16	60.4
4	24	0.393
4	16	100.7

6.1.4.3 Data format

The ADS1298 can produce high-resolution data conversion at 24-bits. The data conversion is in binary *two's complement*, most significant bit (MSB) first (see Appendix A). This means the first bit of read converted data is used to determine the sign of the read value (e.g. if a positive value, the first bit of data will be a 0). A maximum positive output would therefore be read as 0x7FFFFF and the smallest positive value read, the least significant bit (LSB), would be read as 0x000001. The LSB relates to the value $V_{Ref}/(2^{23} - 1)$; if V_{Ref} = 2.4 V, then the value of the LSB ~= 286.1 nV. If the full-scale of the PGA (gain = 12) is used, it is possible to read input signal values with a resolution as small as 23.84 nV. Table 6-2 demonstrates examples of the output code (in hexadecimal and decimal) of the converted input signal at 24-bit resolution in two's complement format.

Table 6-2: Output code of converted data at 24-bit resolution (two's complement).

Voltage	Hexadecimal	Decimal	Notes
$+V_{ref}$	0x7FFFFF	8,388,607	*Positive full scale*
$+V_{ref}/(2^{23} - 1)$	0x000001	1	*Positive LSB*
0	0x000000	0	*Zero*
$-V_{ref}/(2^{23} - 1)$	0xFFFFFF	-1	*Negative LSB*
$-V_{ref}(2^{23}/2^{23} - 1)$	0x800000	-8,388,608	*Negative full scale*

6.1.4.4 ADS1298 configuration

As previously mentioned, the retrieval of the digitised data and configuration of the ADS1298 is done through its SPI port. The ADS1298 is the *slave* and the FPGA/microcontroller is the *master* of the SPI pair (see Appendix B); data are 'clocked out' of the ADS1298 via the DOUT/MISO line using the incoming SCLK signal from the FPGA/microcontroller. Similarly, configuration data to be written to the ADS1298 is 'clocked in' to the ADS1298 by the DIN/MOSI line.

The ADS1298 is configured by writing to different registers in its memory. The master device will issue a write register (WREG) command to the ADS1298 via the DIN/MOSI line of the SPI. The WREG operation code (opcode) is '010*r rrrr*', where '010' is the write command and '*r rrrr*' refers to the register address to start writing to. For example, the opcode '0100 0011' (0x43 in

hexadecimal) will start writing to the third register of the ADS1298. The next opcode issued to the ADS1298 determines how many registers from this address to write to, in the format '000*n nnnn*' where '*n nnnn*' is the number of registers to write to minus one. For example, '0000 0001' (0x01) tells the ADS1298 to write to two registers (starting from the register address issued in the previous opcode). Following the two opcodes, the data to be written to that register is clocked in using the DIN/MOSI line of the SPI as well; Figure 6-13 shows an SPI example of configuring registers of the ADS1298 using the WREG opcodes. Similarly, it is possible to read from registers using a read register (RREG) command should the user wish to check the current data written to a register; the first RREG opcode is '001*r rrrr*' and data are clocked out of the ADS1298 using the DOUT/MISO line. For both writing and reading to the ADS1298, the chip-select (CS) line must first be brought low, as shown in Figure 6-13.

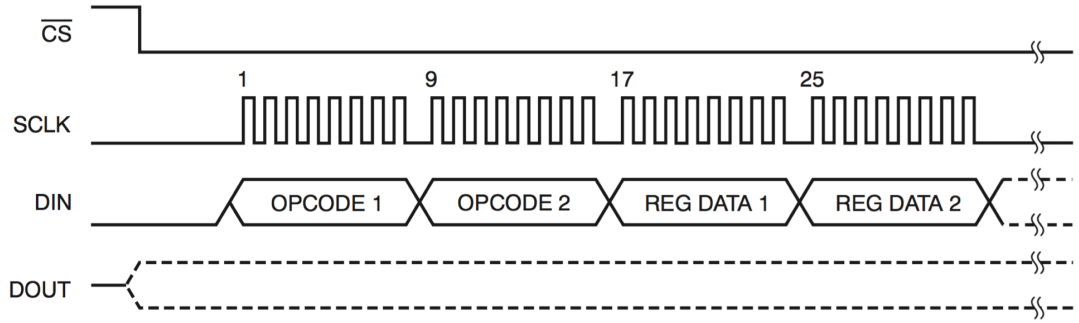


Figure 6-13: Write register (WREG) command example. *The second opcode in this example would be '0000 0001' (0x01) for writing to two registers. (Image taken from ADS1298 datasheet, © Texas Instruments).*[74]

Other commands can be issued to the ADS1298 without having to write to specific registers. System commands such as RESET, START, and STOP reset the device, start the conversion of data, and stop the conversion of data, respectively. These commands can be issued by pulling the RESET and START pins high/low (see pin layout in Figure 6-8) or by issuing byte-long commands over SPI (when CS is pulled low). Different data read commands, such as 'read data continuous' (RDATAC) and 'stop read data continuous' (SDATAC), can also be issued to the ADS1298; RDATAC mode will continuously convert data and make it available for retrieval when DRDY is low and SDATAC will stop this continuous conversion of data. The ADS1298 will be in RDATAC mode by default on power-up and will

need to be put into SDATAC mode in order to write or read to any of the device's configuration registers (WREG/RREG). Whilst the device is in SDATAC mode, the 'read data (RDATA)' command can be issued to retrieve a single piece of data; the RDATA command must be issued every time a piece of data is needed while the device is in SDATAC mode. The RDATAC, SDATAC, and RDATA commands are issued to the ADS1298 over SPI (when CS is pulled low).

There are eleven registers that require configuration for the ADS1298 to operate with the desired behaviour: eight channel configuration registers (gain, channel power-down etc.) and three device configuration registers (conversion rate, reference voltage etc.). The configuration data written to each register is shown in Table 6-3. The start-up and configuration procedure of the ADS1298 shown in Figure 6-14 is based on the recommended start-up procedure for data capture in the ADS1298 datasheet.

Table 6-3: Configuration data to be written to ADS1298 on power-up. *Eleven registers require configuration for the ADS1298 to operate as required. Included is the register data for turning off unused channels.*

Register	Address	Write data	Key functions
CONFIG1	01h	0x60 to 0x66	• *Low power mode* • *Data rate = 16 kSPS to 250 kSPS* • *0x60 = 16 kSPS*
CONFIG2	02h	0x00	• *Test signals off*
CONFIG3	03h	0xC1	• *Enable internal reference* • V_{ref} *= 2.4 V*
CHnSET	05h to 0Ch	0x00 to 0x60	• *Normal operation* • *Gain = 1 to 12* • *0x60 = gain 12*
CHnSET	05h to 0Ch	0x81	• *Channel power-down* • *Input shorted*

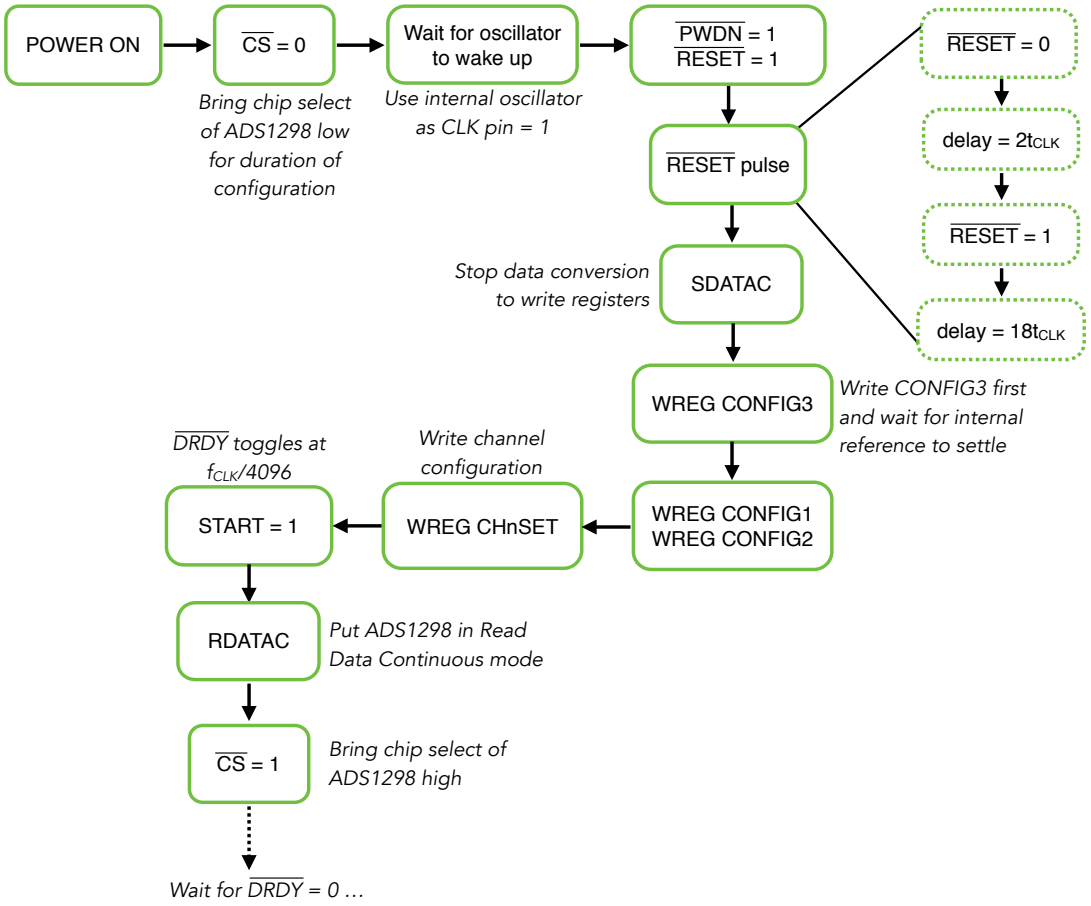


Figure 6-14: ADS1298 start-up configuration and procedure. *This protocol is to be executed at device start-up, before the conversion and retrieval of data.*

6.1.4.5 Data retrieval

Data are clocked out of the ADS1298 on the rising edge of the clock pulse. The number of bits being clocked out of the ADS1298 is 24 status bits followed by 24 bits for each of the 8 channels, totalling 216 bits; each clock pulse will retrieve 1 bit of data, thus requiring 216 clock pulses (see Figure 6-15). This is the number of bits required for each sampled piece of data, regardless of how many channels are being used (i.e. if channels are switched off in configuration, this 'empty' data will still need to be clocked out of the ADS1298).

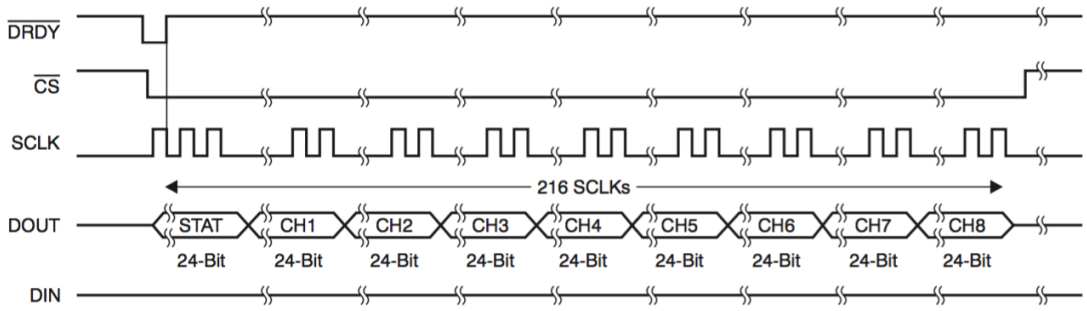


Figure 6-15: SPI data output for retrieving data. *When DRDY is low, data are ready for retrieval. The ADS1298 CS pin is pulled low. (Image taken from ADS1298 datasheet, © Texas Instruments).*[74]

As shown in Figure 6-15, when data are ready for retrieval the DRDY pin will go low, telling the FPGA/microcontroller to start clocking data over the SPI. The FPGA/microcontroller will then send a low signal to the ADS1298's CS pin to initiate the SPI transfer. DRDY will return to high on the falling edge of the first clock pulse in the SCLK signal. The CS pin is then pulled high after the 216 clock pulses.

6.1.5 Microcontroller

The ADS1298 can be controlled and configured by either an FPGA or a microcontroller. FPGAs are far more flexible that microcontrollers as they can be completely reprogrammed to perform any logic task that their number of gates will allow. However, the FPGA integrated circuit requires additional electronics and so the stand-alone FPGA system can have a large footprint. Embedding this into another device with the aim of eventually making it wearable may not be a feasible solution. Due to their flexibility, FPGAs also consume a considerable amount of power compared to a microcontroller, which can make them an unsuitable choice for compact, battery-operated devices.

Microcontrollers can be very small low-power integrated circuits (ICs), which makes them a better choice over FPGAs for this device. In addition, the requirements of the microcontroller in the device are relatively simple and do not require the programmable flexibility that an FPGA can provide; the role of the microcontroller in the device is to deliver instructions to the ADS1298 in the initial setup when power is turned on and to handle the incoming binary data from the ADS1298.

6.1.5.1 ATMEGA328p

The ATMEGA328p is a low-power 8-bit microcontroller with 32 kB of programmable flash memory and 2 kB of static read access memory (SRAM). It can communicate with peripheral devices using the SPI port, the universal synchronous/asynchronous receiver/transmitter (USART), or the two-wire serial interface (I^2C). The ATMEGA328p has 23 programmable in/out (I/O) pins, eight of which are 10-bit ADCs and two of which are interrupt pins that can be used to interrupt any processes being executed by the program the microcontroller is running. The communication ports and interrupt pins of the ATMEGA328p are shown in Figure 6-16.

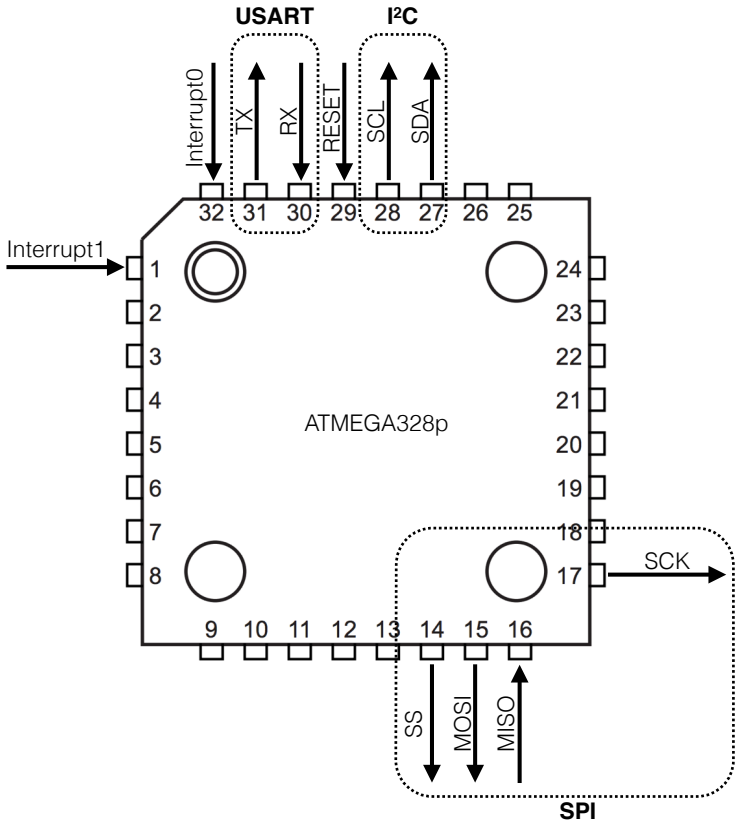


Figure 6-16: Communication ports (SPI, USART, I^2C) and interrupt pins of the ATMEGA328p. *(Image adapted from ATMEGA328p datasheet, © Atmel).*[75]

The pins of the ATMEGA328p are connected to peripheral devices as shown in Figure 6-17. Some of the programmable I/O pins are set to read/write mode in order to interact with peripheral devices. The SPI port is used to retrieve data from both the ADS1298 and the SD card, and the USART is used to send data as well as receive instructions issued to the microcontroller.

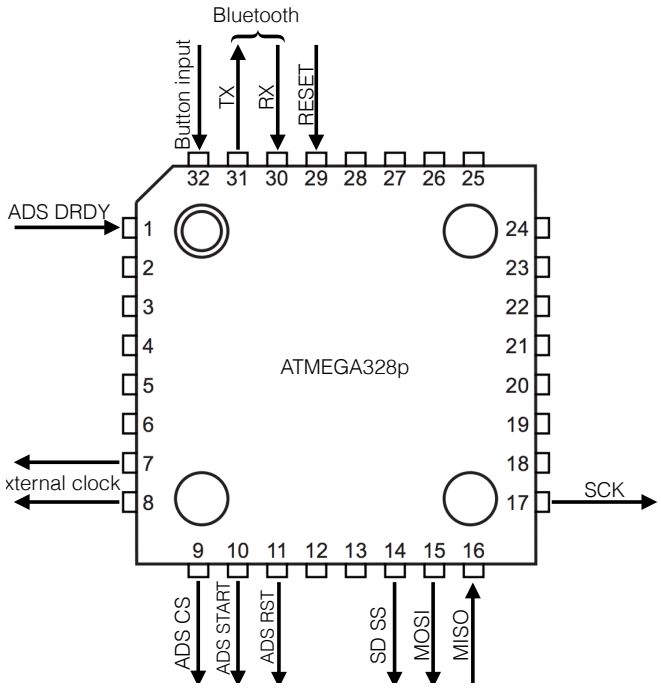


Figure 6-17: Key pin connections of ATMEGA328p to peripheral devices. *Peripheral devices include ADS1298, SD card, and Bluetooth module. (Image adapted from ATMEGA328p datasheet, © Atmel).*[75]

6.1.5.2 Arduino bootloader and clock rate

Arduino is an open-source hardware and software company, producing microcontroller devices for prototyping electronic devices. Arduino devices are can be programmed in C/C++ through the Arduino integrated development environment (IDE). The environment makes it simple for users to upload their programs to the microcontroller and provides many useful functions, common libraries, and even entire programs that the user can integrate into their own code to simplify and speed up programming. Burning the Arduino bootloader onto the microcontroller of the device means that it can be programmed using the Arduino IDE (see Section 2.5.1). The bootloader can be modified to allow the device to be run at different voltage levels as well as different clock frequencies. For example, if the microcontroller requires a logic output at a particular voltage level, then the bootloader can be amended to achieve this.

As the digital interface of the ADS1298 operates at 3.3 V, the logic level of the ATMEGA328p needs to be stepped down to match this so that is does not risk damaging the ADC. This can be done using a logic converter (a series of voltage regulators), however, this will take up space on the device layout and add to the

complexity of the device. Instead, the logic level of the ATMEGA328p is lowered by bootloading it to operate at V_{cc} = 3.3 V. However, by doing so, the clock rate of the device needs to be reduced to keep it within a safe operating range (see Figure 6-18). The frequency of the external oscillator driving the clock of the microcontroller is reduced to 8 MHz to ensure the device remains stable (if V_{cc} = 5V, clock = 16 MHz). The ATMEGA328p is bootloaded with a bootloader so that it can operate at V_{cc} = 3.3 V and a clock frequency of 8 MHz.

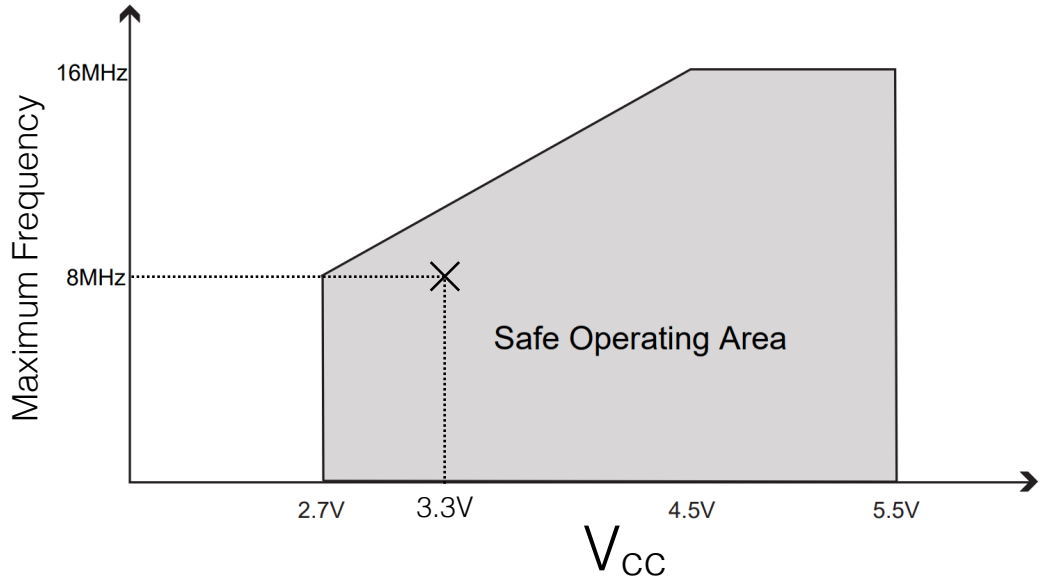


Figure 6-18: Maximum clock frequency vs. operating voltage of ATMEGA328p. *(Image adapted from ATMEGA328p datasheet, © Atmel)* [75]

6.1.6 Data transmission and storage

6.1.6.1 Wireless protocol – Bluetooth

It is desirable for the device to be made wireless, easing its application and making it more comfortable for the patient as they will not be constrained to a single location during a recording session.[v] Several wireless protocols were considered when designing the device: Wi-Fi, ZigBee, and Bluetooth. The wireless protocol will be used primarily for transmitting data handled by the microcontroller, but also for programming the device and issuing commands/instructions to the device.

[v] *Patients will need to lie still during a recording session, otherwise voluntary muscle contractions will interfere with the detection of FPs. As recording sessions can be very long, patients could feasibly move for short periods of time, with FP detection only taking place during 'still' periods.*

For this application, Bluetooth is chosen as the wireless protocol for transmitting data despite its relatively low bit rate compared to other wireless protocols (see Table 6-4). Bluetooth has low power consumption, making it ideal for a small battery-powered device that will need to potentially record data over a long period of time. It does, however, have a relatively short range, but it is unlikely the patient will move out of range of the target tablet/computer during the recording session (as they will need to remain still). Another benefit of Bluetooth is that most modern tablets/computers have a Bluetooth transceiver incorporated in their design therefore a peripheral transceiver attachment is not required.

Table 6-4: Comparison of wireless protocols ZigBee, Bluetooth (classic), and Wi-Fi. *(Data taken from www.sparkfun.com).*[76]

Protocol	Max bit rate	Typical bit rate
ZigBee	0.25 Mbps	0.2 Mbps
Bluetooth Classic	1-3 Mbps	0.7-2.1 Mbps
Wi-Fi	54 Mbps	24 Mbps

A disadvantage of Bluetooth is its relatively low bit rate, for example compared to Wi-Fi. It is desirable for the data rate of the device to be comparable to or greater than current industry standards (e.g. TMSi Refa sampling rate = 2048 SPS, see Section 2.2.2). At this sampling rate, the device should be able to transmit all eight channels at 16-bit resolution (see device resolution discussion in Section 6.1.8.1) using Bluetooth, as shown in Equation (6-10).

$$Bit\ rate = 16\ bit \times 2048\ SPS \times 8 \approx 262\ kbps \quad \textbf{(6-10)}$$

An external Bluetooth transceiver module (Bluefruit EZ-link, Adafruit) is used to give the device wireless connectivity; this Bluetooth module is designed for easy syncing and pairing with other Bluetooth devices.

The rate at which the device can wirelessly transmit data is limited by the rate at which data can be passed to the Bluetooth module via the USART port. As discussed in Section 6.1.5.2, the clock rate of the ATMEGA328p is set to 8 MHz to ensure the device is stable when operating at 3.3 V. At this clock rate, the ATMEGA328p datasheet states the USART port will start incurring large errors

at bit rates over 115.2 kbps; I have also tested higher bit rates than this and have not been successful at transferring data without errors or 'dropped bits'. The serial port is configured to '8N1', meaning that for every 8 bits of data there is 1 start bit, 1 stop bit, and 0 parity bits. This translates to a total of 10 bits of data being transferred for every 8 bits of converted data; for example, transferring 16 bits of data would require 20 bits to be transferred. Equation (6-11) demonstrates that only two full channels can be transmitted at this bit rate, resolution, and sampling rate:

$$Channels = \frac{115200\ bps}{20\ bits \times 2048\ SPS} = 2.8 \rightarrow 2 \qquad \textbf{(6-11)}$$

However, there is no actual requirement for all 8 channels of the device to be viewable at the same time. Therefore, to overcome any issues with the bit rate of Bluetooth, the device will instead transmit a single channel of data at a time so that the user can inspect the quality of the incoming signals as they are recorded. The user will be able to cycle through the channel being transmitted by pressing a button, that will be incorporated into the device design. Meanwhile, all 8 channels are logged locally at a high sampling rate to a removable storage card (see Section 6.1.6.2).

6.1.6.2 Removable storage card

Incorporating an SD card on the device provides a means of storing data locally to the device. All eight channels of data sampled by the ADC are stored to the SD card. As SD cards are removable, this stored data can be manually transferred by inserting the SD card into a secondary device (i.e. tablet/computer) or by uploading it via the device's Bluetooth connection after the recording session has finished. The SD card interfaces with the microcontroller over SPI, just as the ADS1298 does; the CS pins control which device the microcontroller's SPI interface is communicating with, as shown in Figure 6-19.

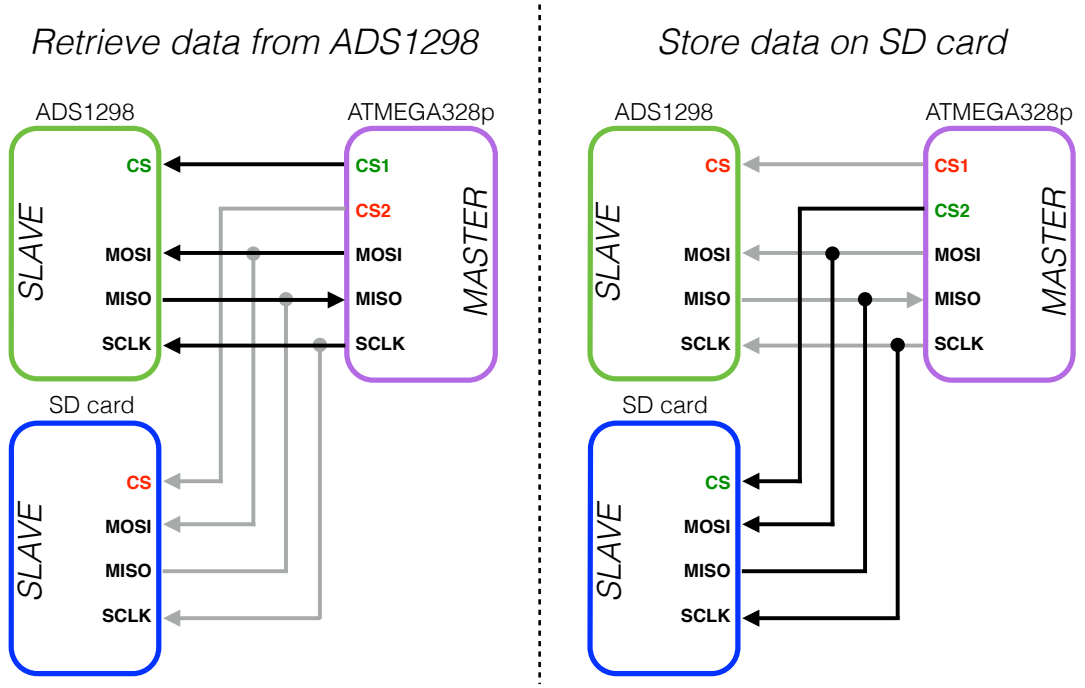


Figure 6-19: SPI connection between microcontroller and peripheral devices. *Using alternative chip-select pins (CS1 and CS2), the ATMEGA328p can retrieve data from the ADS1298 and then save it to the SD card using the same SPI.*

6.1.7 Power management

The device is powered by a rechargeable 3.7 V lithium-ion (Li-ion) battery. The power from the battery is delivered through a switch (for turning the device on and off) and resettable fuse. The resettable fuse protects the device from excessive currents by switching to a high resistance when the power is on; it automatically resets itself to a low resistance after power has been removed. The 3.7 V supply from the Li-ion battery is regulated to provide voltages at three different levels, as will be described in Section 6.1.7.2.

6.1.7.1 Battery charger

The device features an on-board battery charger (MCP73831) that takes a 5 V input, supplied by a micro USB port. The MCP73831 has a status pin that will turn on a connected LED when the battery is charging and turn it off when the battery is fully charged. As the MCP73831 charger outputs a regulated power supply, the system can still operate without the battery connected as long as the device is powered via USB (see schematic in Figure 6-20).

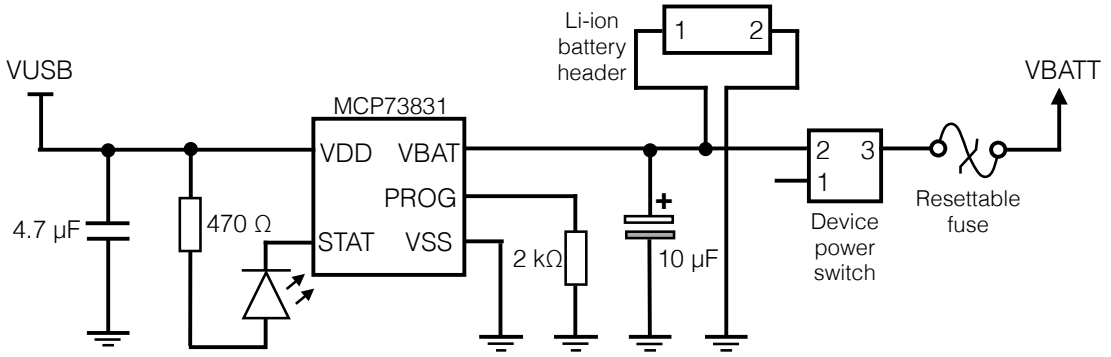


Figure 6-20: Schematic of the device power supply and battery charger. *The MCP73831 will charge a battery using USB as 5 V power source. If no battery is connected, the MCP73831 can still power the rest of the device, provided it is connected via USB. Power is delivered to the rest of the device via the power switch and a resettable fuse protects the rest of the circuitry from excessive currents.*

6.1.7.2 Voltage regulators

There are three voltage levels required by the device: 3.3 V for the digital supply of the ADS1298 and supply of the ATMEGA328p (bootloaded to 3.3 V, 8 MHz), and ±2.5 V for the bipolar analogue supply of the ADS1298 and the rails of the op-amp pre-amplification array. These are described below:

TPS7A7001

The TPS7A7001 by Texas Instruments is used to regulate the +3.3 V and the +2.5 V power supplies. The TPS7A7001 is a low-dropout regulator, meaning it can provide a regulated voltage supply similar to its input voltage. The regulator requires a voltage divider configuration in order to determine the output voltage, shown in the schematic in Figure 6-21. The formula is given in Equation (6-12). The V_{in} of both the regulators takes a 3.7 V supply from the Li-ion battery (4.5 V if powered directly by USB, without battery). Table 6-5 shows the values of R_1 and R_2 needed to achieve the +3.3 V and +2.5 V power supplies.

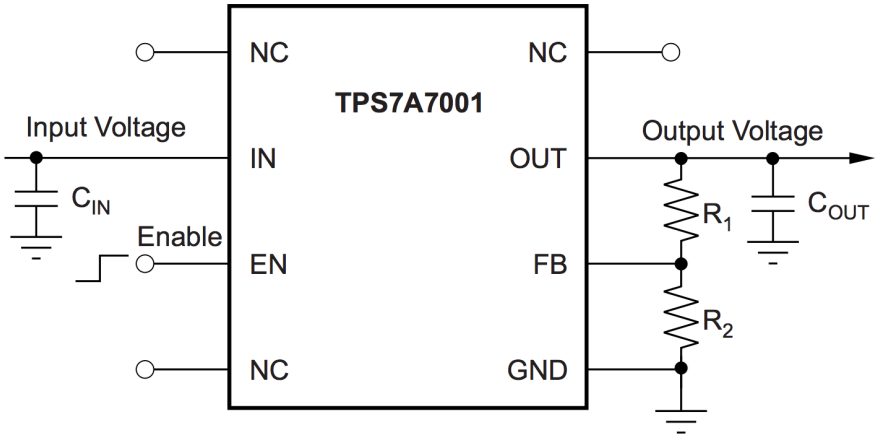


Figure 6-21: TPS7A7001 schematic for producing +3.3 V and +2.5 V. *Power supplies, configured by setting values of R_1 and R_2. (Image adapted from TPS7A7001 datasheet, © Texas Instruments).*[77]

$$V_{out} = 0.5 * (1 + \frac{R_1}{R_2}) \quad \textbf{(6-12)}$$

Table 6-5: Resistor values required for producing +3.3 V and +2.5V. *Values are as recommended in TPS7A7001 datasheet.*[78]

V_{out}	R_1	R_2
3.3 V	169 kΩ	30.1 kΩ
2.5 V	121 kΩ	30.1 kΩ

LT1611

The LT1611 by Linear Technology is an inverting regulator, meaning it can provide the negative power supply (-2.5 V) needed for the bipolar supply of the ADS1298 and LMP7702 dual op-amp. The LT1611 requires additional components for the voltage inversion (i.e. inductors, diodes). The value of the output inverted supply is also determined by the configuration, given by the formula in Equation (6-13). The recommended value for R_2 in the LT1611 datasheet is 10 kΩ. With R_1 = 10 kΩ, this will achieve the desired output voltage of -2.5 V.[78]

$$R_1 = \frac{|V_{out}| - 1.23}{\frac{1.23}{R_2} - 4.5\times10^{-6}} \quad \textbf{(6-13)}$$

6.1.8 Data rate and data management

This section describes how certain interactions between the selected device components have determined the specifications of the overall device.

6.1.8.1 Device resolution

The ADS1298 is capable of 24-bit data conversion, which provides a digitised signal with an impressively fine resolution. For example, the device will be able to digitise signals with a resolution of 236 pV at 24-bit conversion (two's complement, A_{total} = 1212, V_{ref} = 2.4 V), as shown in Equation (6-14). The usefulness of reading data at such a high-resolution is debatable; factors such as noise caused by a real-world environment or the device's input referred noise etc. would mean that very small voltages on the signal converted by the least significant bits of the ADC are unlikely to be useful. Even if the device were operated in an ideal/ultra-low-noise environment, this resolution is incredibly fine and beyond what is necessary for capturing the fine intricacies of the FP morphology. Instead, reading just the first 16 bits of the converted data still gives the device a very desirable resolution of 60 nV (two's complement, A_{total} = 1212, V_{ref} = 2.4 V), as shown in Equation (6-15), and also puts less stress on the delivery and storage of the converted data, i.e. Bluetooth transmission and memory. For this reason, the data are transmitted and stored as 16-bit; the last 8 bits of the converted data are discarded after the converted data are retrieved from the ADS1298.

$$Resolution = \frac{2.4\,V}{1212 * (2^{23} - 1)} = 236\,pV \quad \textbf{(6-14)}$$

$$Resolution = \frac{2.4\,V}{1212 * (2^{15} - 1)} = 60\,nV \quad \textbf{(6-15)}$$

6.1.8.2 Data rate

The main challenges with a multiple-channel wireless device are the handling of data and the rate at which it can be sampled and sent. In terms of sampling rate, the device should be able to reach at least the same sampling rates as current industry standards; for example, the off-the-shelf system used in the longitudinal study to collect patient data samples data at 2048 Hz (TMSi Refa). However, it is possible that this sampling rate is still too slow to capture vital variations in the

FP morphology as the changes in voltage during the depolarisation and repolarisation stages of an FP occur very quickly and are possibly too fast for a sampling rate of 2048 Hz to detect small, subtle changes. If it were possible, a higher sampling rate may be necessary to detect all variability in FP morphology.

The data-rate of the ADS1298 is limited by the speed of the SPI clock (SCLK), which is in turn controlled in the setup of the *master* device (ATMEGA328p). The maximum rate of the SPI clock is half the rate of the master's clock frequency, as discussed in Figure 6-18 and given by Equation (6-16).

$$MAX\ f_{SCLK} = 0.5 f_{masterCLK} = 4\ MHz \quad \textbf{(6-16)}$$

The maximum data-rate (DR) the ADS1298 can operate at is shown in Equation (6-17) (given in the ADS1298 datasheet). The time period t_{CLK} denotes the time period of the ADS1298's internal clock, where f_{CLK} = 2.048 MHz. As shown in Equations (6-18) to (6-20), the DR of the ADS1298 is limited to rates lower than 17.8 kHz and thus the DR of the ADS1298 is limited to 16 kSPS; the next highest DR of the ADS1298 is 32 kSPS, which is not achievable.

$$t_{SCLK} < \frac{(t_{DR} - 4t_{CLK})}{N_{bits} \times N_{channels} + 24} \quad \textbf{(6-17)}$$

$$0.25\ \mu s < \frac{(t_{DR} - 1.9531\ \mu s)}{216} \quad \textbf{(6-18)}$$

$$t_{DR} > 55.9531\ \mu s \quad \textbf{(6-19)}$$

$$f_{DR} < 17.8\ kHz \quad \textbf{(6-20)}$$

However, this is just the rate at which the ADS1298 can convert data and not the sampling rate of the overall device. The rate of the overall device is likely to be lower due to the other processes the microcontroller has to carry out in between retrieving each piece of data. The testing of the device's overall sampling rate is detailed in Section 6.1.10.

6.1.9 Printed circuit board layout – Device version 1

The schematic capture and printed circuit board (PCB) layout of device version 1 was designed in Altium (see Section 2.4.1). The resulting device is a four-layered PCB (see Figure 6-22). The layers consist of two component planes (top and

bottom), a ground plane (digital and analogue ground), and a power plane (+3.3 V, +2.5 V, -2.5 V). The assembled device is shown in Figure 6-23.

The device has nine sensor inputs: eight muscle sensing inputs and a patient reference to be placed over a neutral location (i.e. a bony prominence such as the olecranon or anklebone). Each sensor's signal passes through the buffer and pre-amplification stages of the AFE (Figure 6-22(A)) before entering the ADS1298 (Figure 6-22(B)). The patient reference signal is connected to each negative input of the ADS1298's eight inputs; any common mode signals that exist between the sensing inputs and the patient reference will be removed by the instrumentation amplifier configuration of the ADS1298's PGA.

The ATMEGA328 microcontroller (Figure 6-22(C)) is responsible for managing the stream of data converted by the ADS1298. Data are retrieved by the ATMEGA328p over SPI and then either transmitted by Bluetooth (Figure 6-22(D)) over a serial connection (TX/RX) or stored locally to an SD card over SPI (Figure 6-22(E)). Also included is a separate SPI header for burning the Arduino bootloader to the ATMEGA328p (Figure 6-22(F)).

Two push buttons are included on the device (Figure 6-22(G)). The first button is directly connected to the reset pin of the ATMEGA328p, returning the device to its original state upon booting up. The second button is connected to one of the ATMEGA328's interrupt pins. This is a useful feature to have on the device as it means that at any point during the device's program, pushing this button can execute whichever interrupt routine has been programmed (e.g. change sampling rate, gain of PGA etc.).

The Li-ion battery provides the system with power at 3.7 V; several different voltage regulators on the device provide voltages at +3.3 V, +2.5 V and -2.5 V. The battery is serviced by an on-board battery charger, which is powered by a 5 V supply via a micro USB port (Figure 6-22(I)).

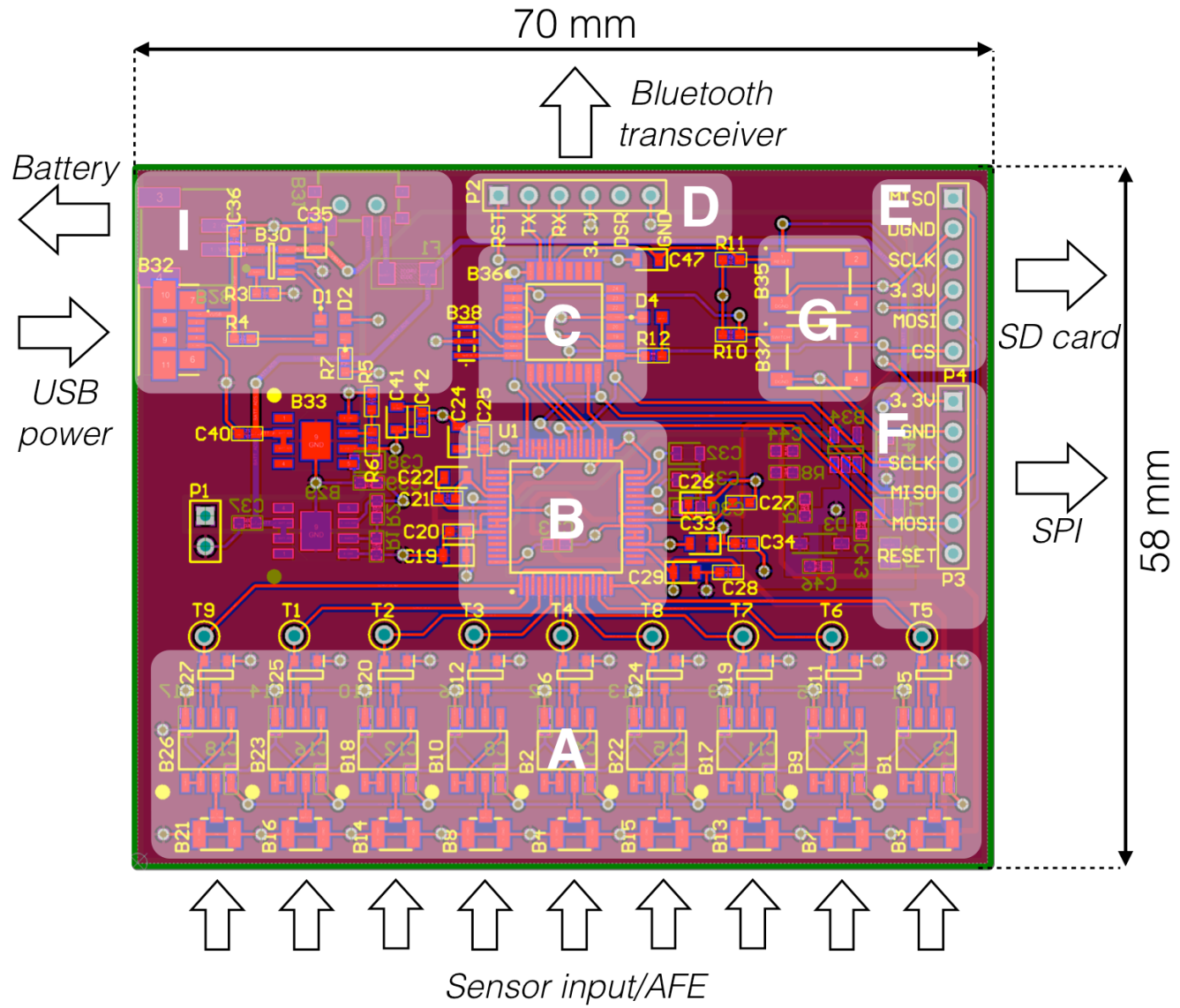


Figure 6-22: PCB layout of 8-channel wireless device. *(A) Device AFE (B) ADS1298 (C) Microcontroller ATMEGA328p (D) Header for Bluetooth module (E) Header for SD card module (F) header for SPI port (G) Push-button input (I) Li-ion battery charger and USB power supply.*

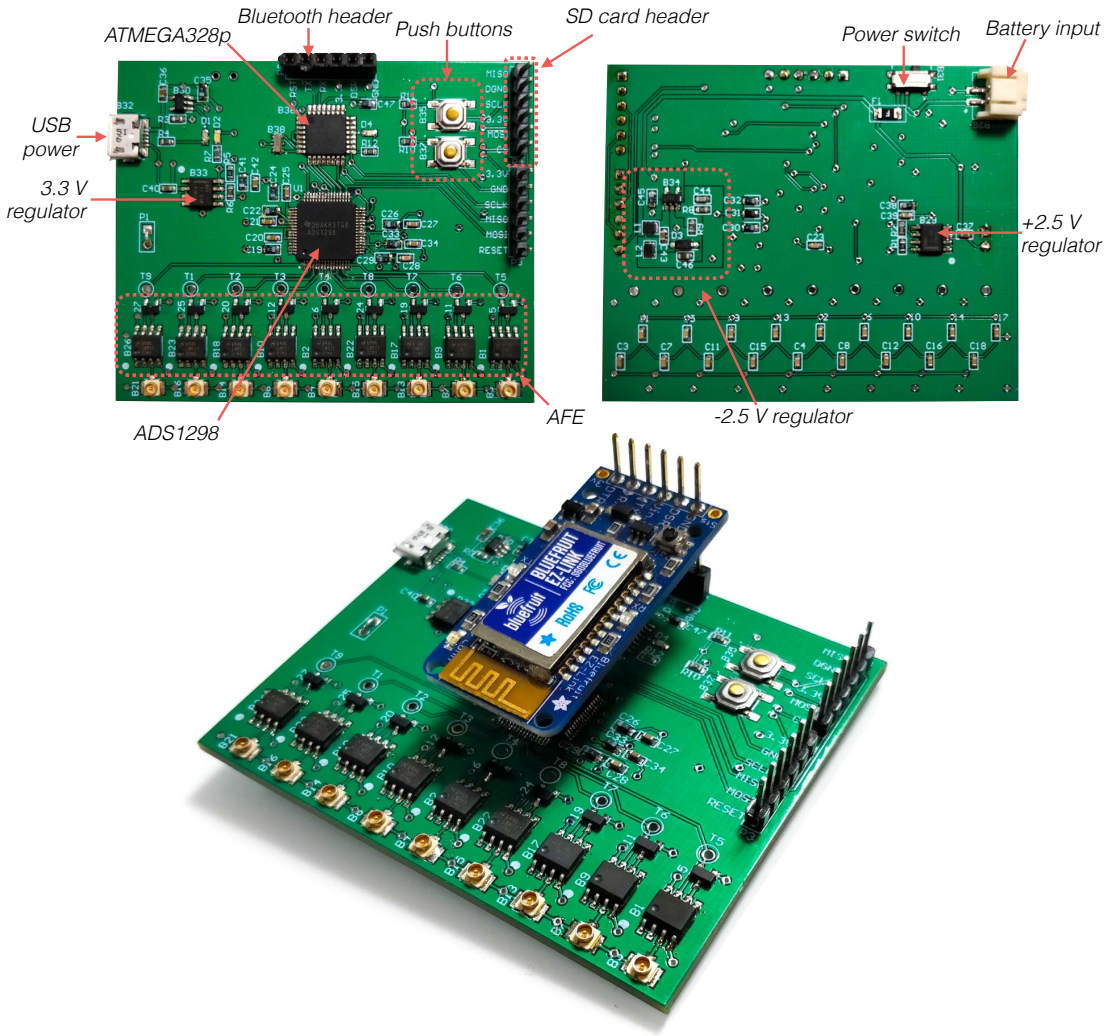


Figure 6-23: Assembled device version 1. *(Top) Labelled key components soldered to top and bottom layers of PCB. (Bottom) Assembled device with external Bluetooth module connected.*

6.1.10 Wireless device performance

The performance of the device has been tested in a standard laboratory setting. An Agilent signal generator was used to generate the waveforms used for testing the device (see Section 2.4.2). A desktop data acquisition (DAQ) system (PowerLab, see Section 2.4.2) was used to record the output of the signal generator at the same time as the signal was captured by the device.

6.1.10.1 Bluetooth streaming tests

As mentioned in Section 6.1.6.1, the Bluetooth transceiver of the device sends a single channel of data at a time as a means of visually checking the quality of the incoming EMG data. The full eight channels of data are saved locally to the device via an SD card. To test the device, a variety of signals were fed into the device via

its AFE; signals were amplified by the AFE and PGA of the ADS1298, digitised, and sent over Bluetooth to a nearby computer.

Figure 6-24 shows the results of testing the device with a range of 1 Hz sine waves of different amplitudes. The wireless device provides a much cleaner signal than the PowerLab DAQ system, as can be seen from Figure 6-24 and the SNR values of the signals in Table 6-6.

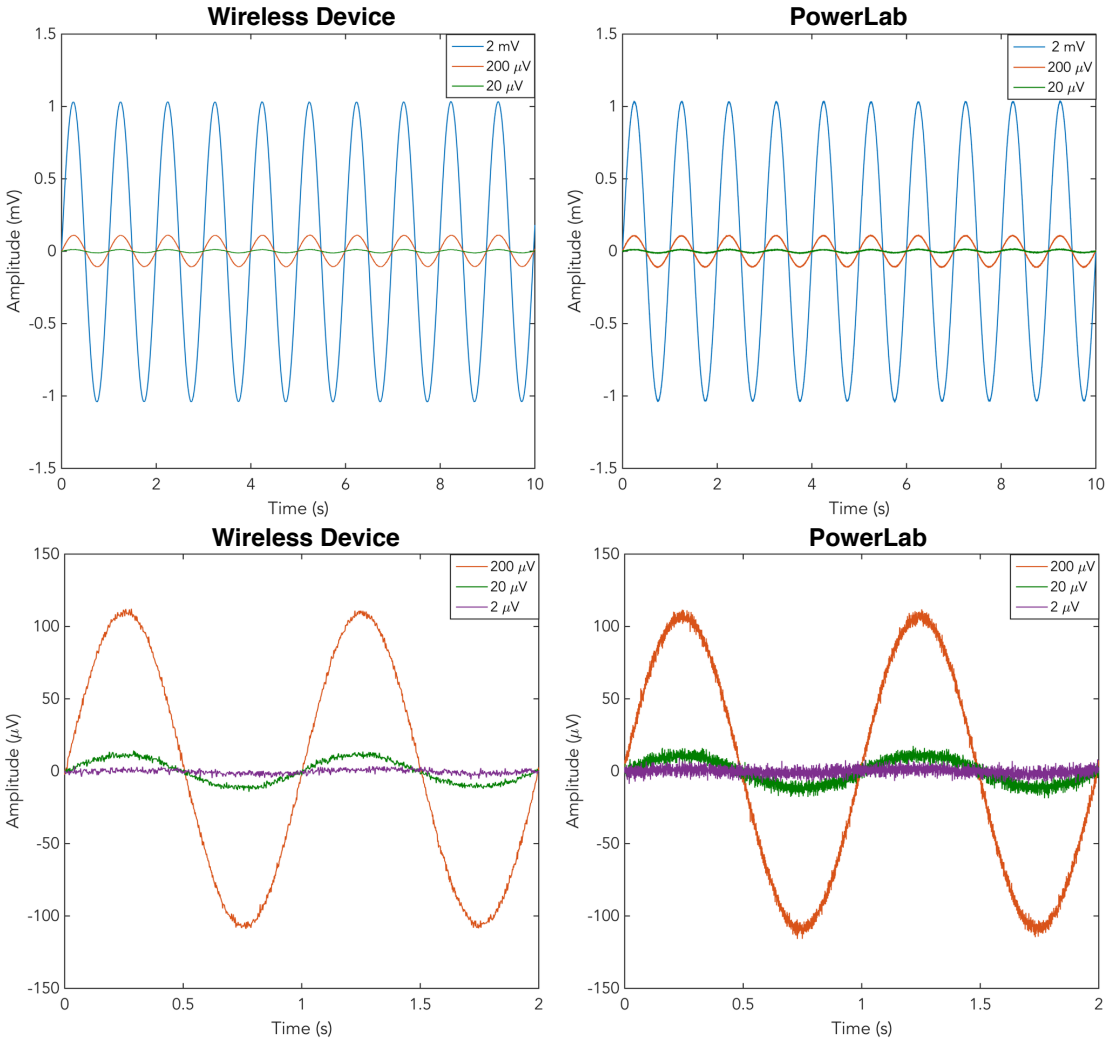


Figure 6-24: Wireless device Bluetooth test compared to PowerLab. *The wireless Bluetooth connection of the device was tested using sine waves (1 Hz) of amplitudes 2 mV_{p-p}, 200 μV_{p-p}, 20 μV_{p-p}, and 2 μV_{p-p}. The bottom plots show a zoomed in view of sine waves 200 μV_{p-p} - 2 μV_{p-p}. The wireless device has provided a cleaner signal than the one captured by the PowerLab DAQ. All data are raw and unfiltered.*

Figure 6-25 compares the capture of signals of amplitude 20 μV_{p-p} and 2 μV_{p-p} with the wireless device and the PowerLab. The signals recorded by both devices

are very noisy at these amplitudes, however, the wireless device still records at a better SNR than the PowerLab (as shown in Table 6-6).

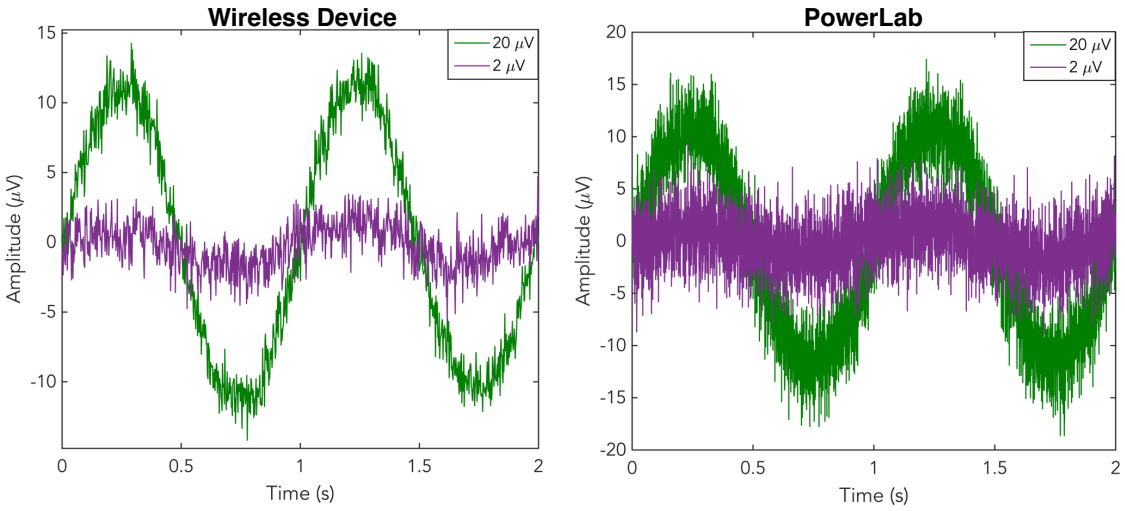


Figure 6-25: Device Bluetooth test of 20 μV_{p-p} and 2 μV_{p-p} sine waves compared to PowerLab. *For signals of very low amplitude, both devices are affected by noise, but the wireless device captures a cleaner recording of the signal.*

Table 6-6: Signal-to-noise ratio (SNR) values of signals recorded by wireless device and PowerLab.

Amplitude	Wireless Device (dB)	PowerLab (dB)
2 mV_{p-p}	48.0	47.7
200 μV_{p-p}	34.4	29.6
20 μV_{p-p}	15.9	9.5
2 μV_{p-p}	-3.3	-8.0

The time-stamp each piece of data was sampled at needs to be transmitted along with the data itself. Each time-stamp is recorded in microseconds and recorded as a 32-bit *long* data type. When testing the rate at which the wireless device can sample and transmit data, the device rate was not limited; as soon as the ADS1298 has data ready (DRDY goes low), data are retrieved by the ATMEGA328p and transmitted via Bluetooth. From testing, the average rate obtained for the device to sample the incoming signal and transmit both the data (16 bits) and time-stamp (32 bits) was approximately 500 Hz. This is quite a low rate to be sampling data at, especially for sampling higher frequency components of the EMG signal. Figure 6-26 demonstrates the effect this low sampling rate can have on signals of different frequencies. The sampling rate of the wireless device

is sufficient for sampling the 10 Hz signal (top trace in Figure 6-26), however at higher frequencies the sine wave starts to become under-sampled.

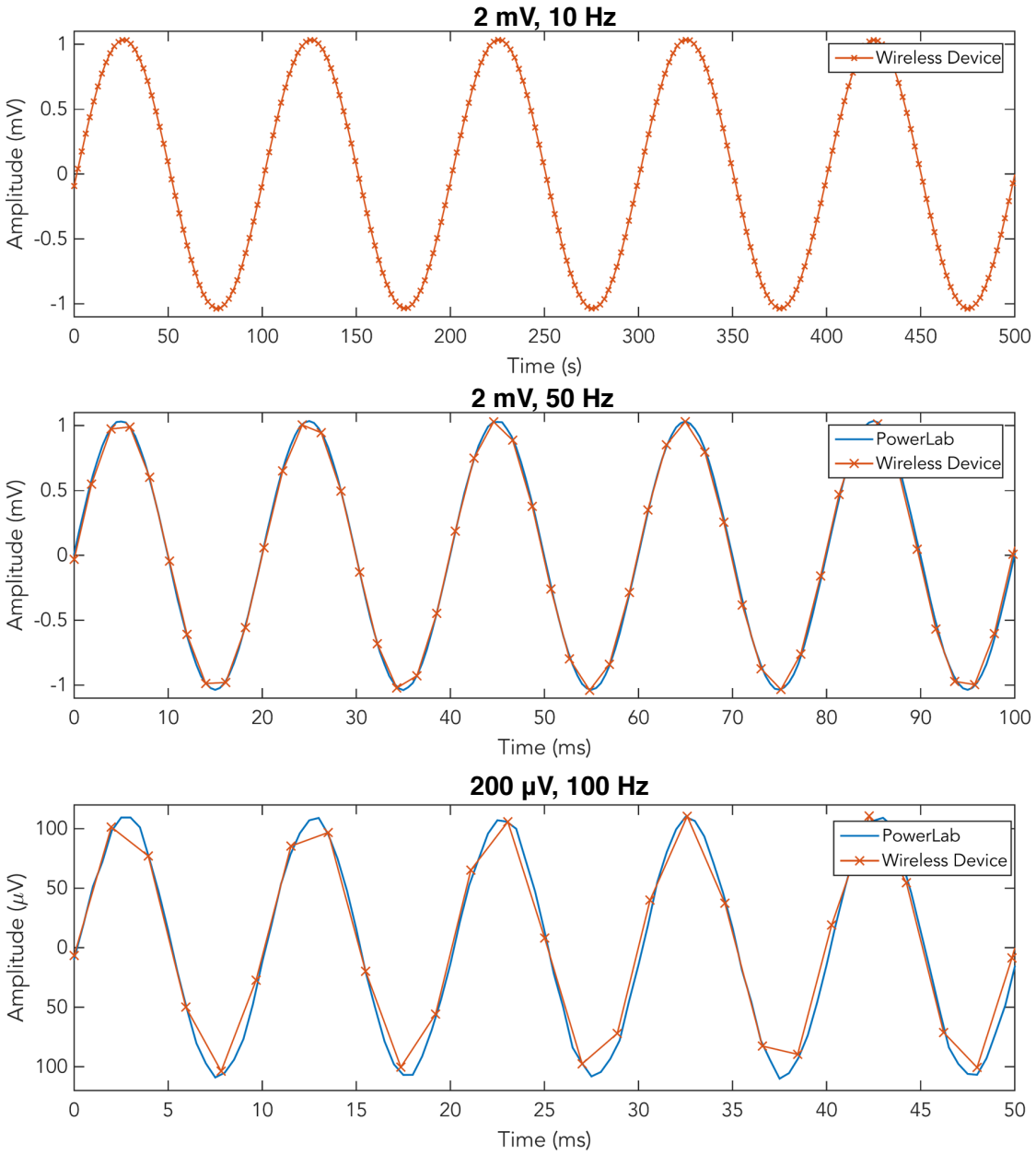


Figure 6-26: Loss of data as a result of low Bluetooth sampling rate. *The wireless device can only sample and transmit data at a rate of approximately 500 Hz, meaning that higher frequency signals are under-sampled. Note, the sampling rate of the Powerlab is set to 2 kHz.*

As previously described in Section 6.1.6.1, for every 8 bits transmitted over Bluetooth, 2 additional bits (start and stop bits) will also be transmitted. For each piece of 16-bit data and accompanying 32-bit time-stamp, this equates to a total of 60 bits being transmitted for each sample. This lowers the overall sampling rate of the device. It is possible to boost the rate of the device by

excluding the time-stamp that each data-point is sampled at; if the receiving device (i.e. tablet/computer) plots the data as soon as it is received, it is not necessary to transmit the sampling time of the data. From testing, I have found this increases the rate of the device's sampling and transmission to approximately 1 kHz. This is still a relatively low rate, however, as the data transmitted over Bluetooth is just for viewing and checking the quality of the signal, this is an acceptable rate for this purpose.

6.1.10.2 SD card problems

As previously discussed in Section 6.1.6, and seen through the testing in Section 6.1.10.1, due to the low bit rate of Bluetooth, the plan is to store all eight channels of data locally to an SD card while just streaming a single channel of data over Bluetooth for data visualisation. Data are to be stored locally to an SD card as a continuous stream and then uploaded to a tablet/computer after the recording session is finished by either manually removing the SD card or by streaming it at a lower bit rate.

The problem with this method is that writing to an SD card is a slow process; from testing, I have found that it typically takes 13 ms to write each piece of data to the SD card file. The reason for this is that the file on the SD card needs to be opened and then closed again each time data are written to it. Therefore, writing to the SD card as a continuous stream of data would result in a device sampling rate of 75 Hz, which is an unacceptable solution.

The maximum amount of data that can be written to the SD card in a single write function is 512 bytes; instead of writing data to the SD card one sample at a time, it may be faster to temporarily store data in a 512-byte buffer in the SRAM of the microcontroller and write the entire buffer to the SD card at once. If converting data at 16-bit resolution over 8 channels, the number of samples that can fit in the 512-byte buffer can be calculated:

$$N_{samples} = \frac{Buffer\ size}{Resolution * N_{channels}} = \frac{512\ bytes}{16\ bit * 8} = 32 \quad \textbf{(6-21)}$$

This calculation does not take into consideration that a value for each time-stamp needs to be stored as well; storing each time-stamp as a 32-bit long for

every sample taken reduces the number of samples that can fit into each buffer block:

$$N_{samples} = \frac{512\ bytes}{(16\ bits * 8 + 32\ bits)} = 25.6 \rightarrow 25 \quad \textbf{(6-22)}$$

Considering the device is required to sample data at a minimum of 2 kHz, a buffer of 25 samples would only be able to record 12.5 ms of data at a time followed by a delay whilst the data are written to the SD card. This quasi-continuous style of recording data may be acceptable granted the write-delay of the SD card isn't significant. However, upon testing, the write-time for a single block of data was found to be 28.9 ± 0.6 ms (mean average for n = 50 write cycles). This writing delay is more than double the duration of the data that is being recorded, and is clearly not an acceptable solution. Figure 6-27 demonstrates the amount of signal that would be recorded and the amount of signal that would be missed during the delay of writing to the SD card.

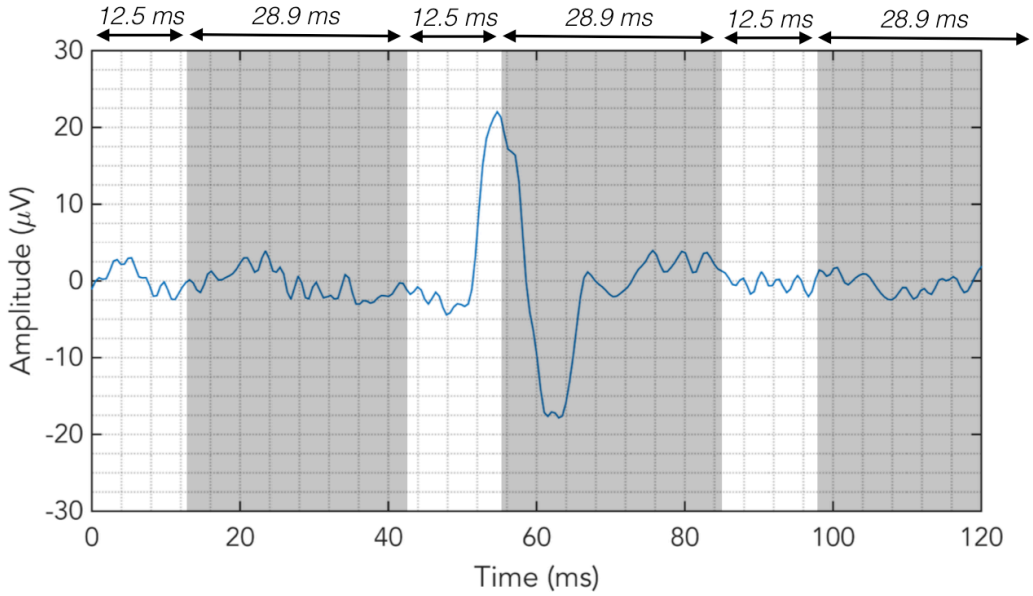


Figure 6-27: Writing delay for one 512-byte block. *A single 512-byte block of data has room for 25 samples of data, or 12.5 ms if sampling at 2 kHz. Grey areas indicate the writing delay incurred for writing a single 512-byte block of data to the SD card, which is on average 28.9 ± 0.6 ms (n = 50). This is too long compared to the amount of data being recorded; a significant portion of the signal will be missed during each writing delay.*

The ATMEGA328p has 2 kB of SRAM available, and so there should be scope to fill the microcontroller's SRAM with multiple 512-byte blocks in a buffer and thus record longer quasi-continuous periods of data. Once the SRAM is full, each

block of data can be written to the SD card at the same time; when the SD card file is opened for writing the first 512-byte block it can remain open so that the remaining blocks of data can be written, as shown in Figure 6-28.

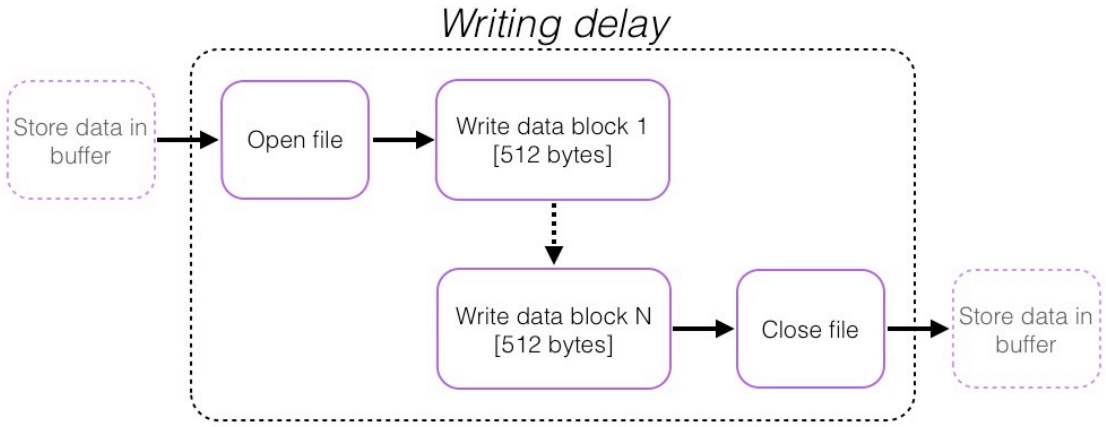


Figure 6-28: Plan for storing and writing multiple 512-byte blocks to an SD card. *When writing multiple 512-byte blocks of data from the buffer stored in SRAM, the file can stay open while each data block is written; the SD card file does not need to be closed and reopened for each data block.*

Using the 2 kB of SRAM the ATMEGA328p has, it should be theoretically possible to store four 512-byte blocks of data in a buffer to be saved to the SD card in a single write action. However, there are many other constant and dynamic variables that the program will need to store in SRAM as well, and so it cannot be used exclusively as a data buffer. In order to free as much space in SRAM as possible, constant variables used by the Arduino program are written to the program memory (flash) of the microcontroller instead. Even so, filling the SRAM of the microcontroller with too much data can risk the program crashing. In SRAM, the 'stack' is the location in memory where parameters and local variables are stored and the 'heap' is where dynamic data are stored (such as the buffer data being stored before being written to the SD card). The stack and heap require a space of free memory between them; if the size of the heap grows too much (i.e. multiple 512-byte blocks being stored), the heap and stack will likely crash, resulting in a program error and failure. From testing, I have found that the system can only run without crashing when using just two 512-byte blocks as a buffer to store data in. Using Equation (6-22) this equates to just 50 samples of data (including time-stamps), or 25 ms of data (sampled at 2 kHz). The delay

incurred by writing two 512-byte blocks at a time was found to be 34.1 ± 3.0 ms. As shown in Figure 6-29, this is an unacceptably small amount of time to be sampling data quasi-continuously, especially with such a long writing delay. Therefore, the device design must be revised in order to fix this problem.

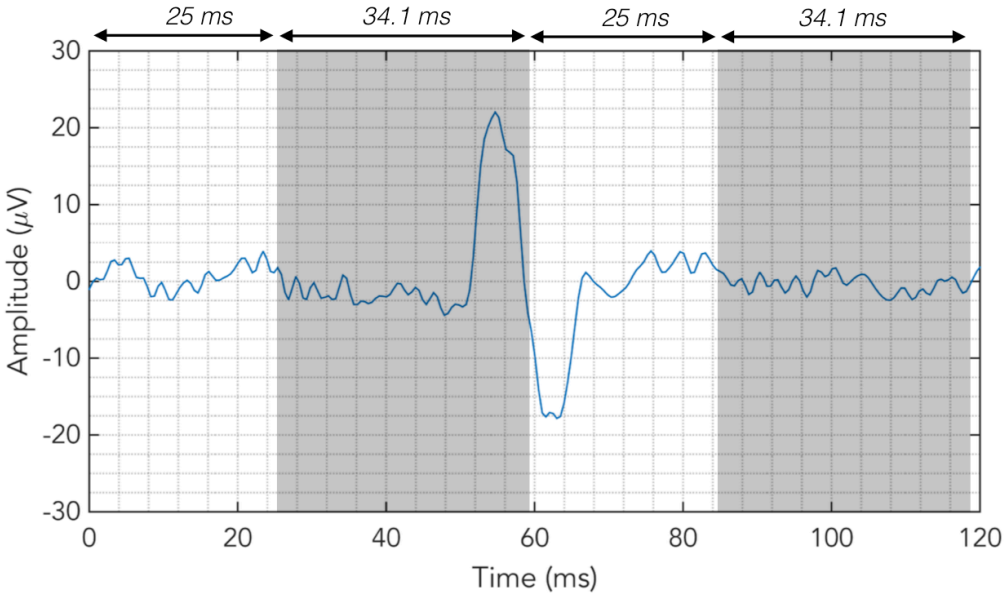


Figure 6-29: Writing delay for two 512-byte blocks. *Two 512-byte blocks of data have room for 25 ms of data (if sampling at 2 kHz). Grey areas indicate the writing delay incurred for writing two 512-byte blocks of data to the SD card, which is on average 34.1 ± 3.0 ms (n = 50). This is still too long compared to the amount of data being recorded; most of the FP waveform is likely to be missed during the writing delay.*

6.2 Device Design – Version 2

As discussed in the previous section, the current microcontroller does not possess enough SRAM to store large amounts of data in a buffer in dynamic memory. The device was therefore re-designed using a microcontroller with a larger amount of SRAM, meaning that the device can store larger amounts of data in a buffer array. The rest of the device remained the same as the first version of the device i.e. same AFE, ADC, data transmission and storage (Bluetooth and SD card), and power management.

6.2.1 Microcontroller – ATMEGA2560

6.2.1.1 ATMEGA2560

The ATMEGA2560 is a low-power, high-performance 8-bit microcontroller with 256 kB of programmable flash memory and 8 kB of SRAM. For communication it has an SPI port, four USART ports, and an I^2C interface. The ATMEGA2560 has 86 programmable in/out (I/O) pins, 16 of which are 10-bit ADCs and eight of which are interrupt pins. The communication ports and interrupt pins of the ATMEGA2560 that are used for communicating with peripheral devices (e.g. retrieving data from ADS1298 and writing to the SD card) are shown in Figure 6-30.

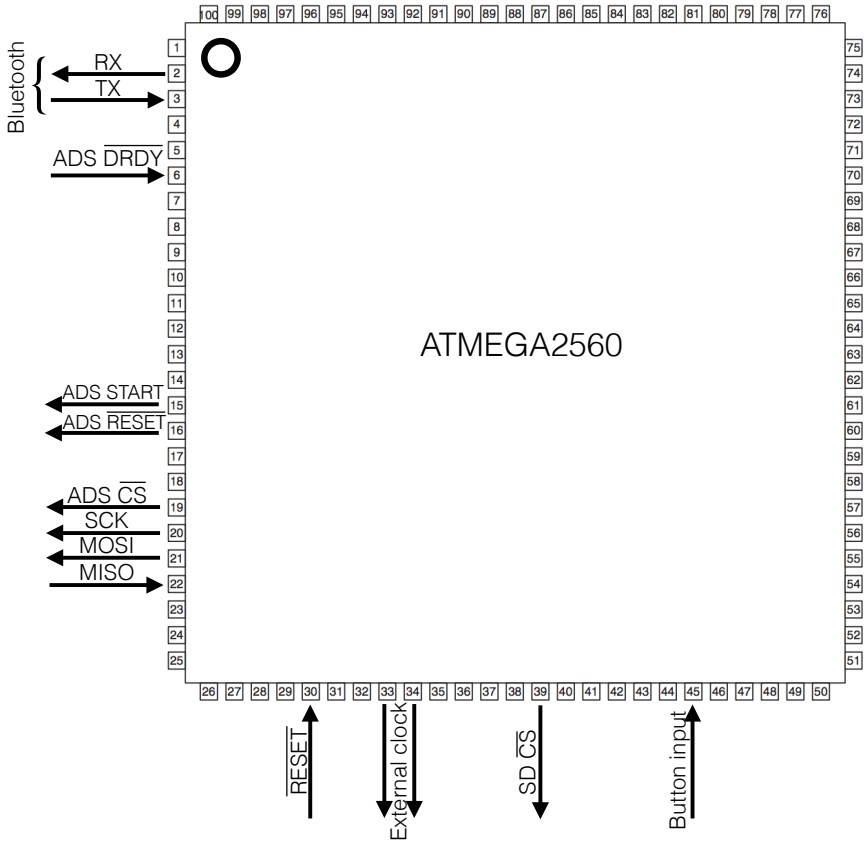


Figure 6-30: ATMEGA2560 pin connections to peripheral devices. *The power and ground connections for ATMEGA2560 are not shown here. (Image adapted from ADS2560 datasheet, © Atmel).*[79]

The ATMEGA2560 is an 8-bit microcontroller. It is first bootloaded with the Arduino bootloader, meaning it can be operated and programmed by the Arduino IDE just like the previous revision of the device. It is bootloaded to

operate at 3.3 V with an 8 MHz external clock. It has a slightly bigger footprint than the ATMEGA328p (see PCB layout of revised device in Section 6.2.2).

6.2.2 Printed circuit board layout – Device Version 2

The four-layer PCB layout of the revised device is shown in Figure 6-31. The device contains the same circuitry as used device version 1 for the AFE, ADC, and power management. The revised PCB design uses the ATMEGA2560 microcontroller (Figure 6-31(A)) to control the device; the larger footprint of the ATMEGA2560 has resulted in a slightly larger overall device (71 mm x 68 mm) compared to the previous version. The electronic schematics for device version 2 can be found in Appendix C.

This device version incorporates an on-board SD card slot (Figure 6-31(B)), rather than an external, detachable one as with the previous version of the device. The device's Bluetooth connection is still provided over an external Bluetooth module, which connects via the header at the top of the device. This version of the device includes an input for connecting the patient to the ground of the device (Figure 6-31(C)). This connection is not for measurement, but ensures that the device is in the same reference/voltage plane as the patient when data are converted by the ADC.

As with the previous design of the device, the SPI pins have been broken out so that the microcontroller can be bootloaded while on the PCB. The ATMEGA2560 has a generous number of general I/O pins available and so several pins have been broken out for use, should they be required. The assembled device is shown in Figure 6-31.

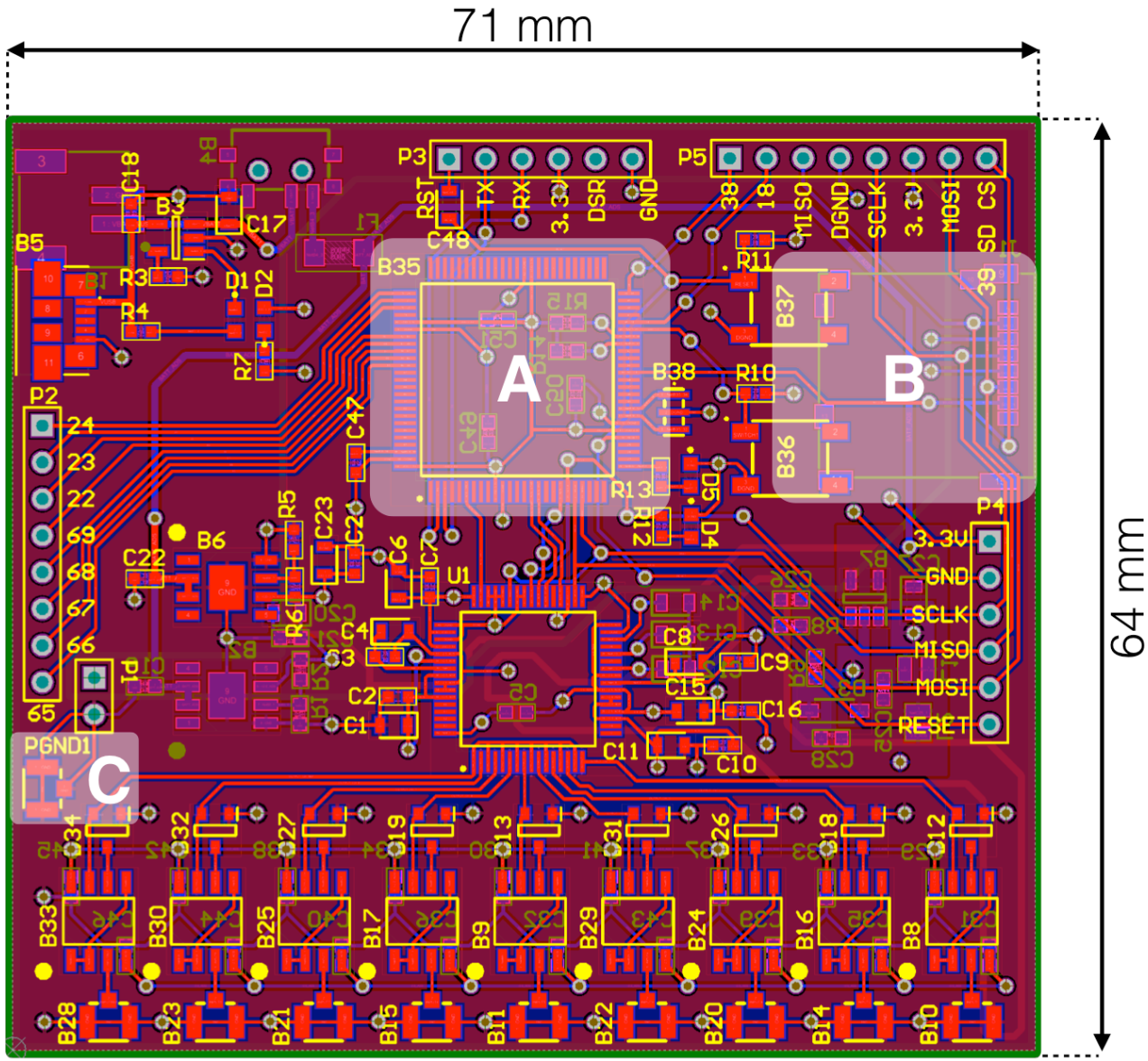


Figure 6-31: PCB layout of 8-channel wireless device version 2. *(A) New microcontroller ATMEGA2560 (B) On-board SD card slot (C) Patient ground input.*

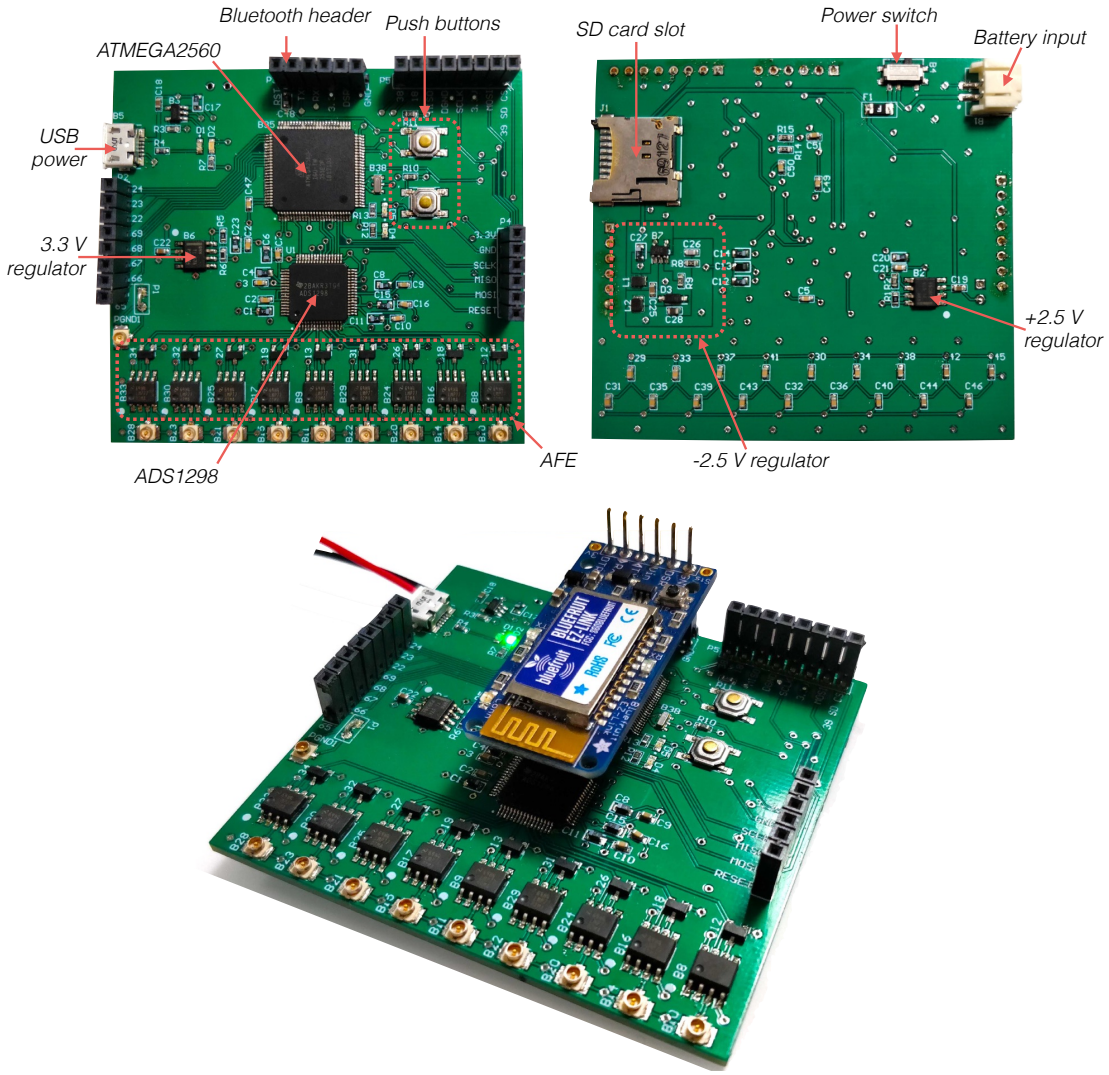


Figure 6-32: Assembled device version 2. *(Top) Labelled key components soldered to top and bottom layers of PCB. (Bottom) Assembled device with external Bluetooth module.*

6.2.3 Device version 2 characterisation

6.2.3.1 Buffer writing speed

The ATMEGA2560 possesses 8 kB of RAM (four times the amount of the ATMEGA328p) and so considerably more data can be stored in a buffer without risking the stack and heap crashing. From testing the new version of the device, I have found that the system can store a maximum of eleven 512-byte buffer blocks in SRAM before the system crashes due to stack/heap collision. I have decided to use no more than ten 512-byte blocks to reduce to the risk of stack/heap collision, and to make more free memory available for additional variables, should they be necessary. Ten 512-byte blocks of data would be

enough for 250 samples of data (including time-stamps), equating to 125 ms of data (if sampling at 2 kHz). Running this in a quasi-continuous mode would still result in a 77.3 ± 4.9 ms delay every time the buffer is written to the SD card, however, as shown in Table 6-7, this is a considerable improvement compared to the first version of the device.

Table 6-7: Comparison of quasi-continuous sampling for device versions 1 and 2. *Assuming data are sampled at 2 kHz, device versions 1 and 2 will be able to store 25 ms and 125 ms in a buffer in the microcontroller's SRAM, respectively.*

Device version	Blocks (512-byte)	Recording time (ms) *(sampled at 2 kHz)*	Writing time (ms)
1	2	25	34.1 ± 3.0
2	10	125	77.3 ± 4.9

Instead of attempting to save everything sampled by the ADS1298 to the SD card as a quasi-continuous EMG signal, it may be more desirable to only save the signal data that is of interest in the investigation into ALS progression, i.e. FPs. The size of the buffer stored in SRAM of the ATMEGA2560 is large enough to store an FP. The delay caused by the writing time of the SD card (approximately 77 ms) is short enough that the device will be ready to record the next FP when it occurs as fasciculations generally fire slowly, normally less than 1-2 Hz (> 0.5-1 s).[71]

6.2.3.2 Circular buffer array

In order to only save FPs from the incoming signal, a 'circular-buffer array' is used to store the data. A circular-buffer array is a data structure of fixed size that stores data as if it were connected at either end; once the structure is full, the oldest data in the structure will be overwritten. For example, the circular-buffer array in Figure 6-33 stores 'HELLOWORLD' in an 8-element array; once the buffer is full, the oldest elements in the array are overwritten first.

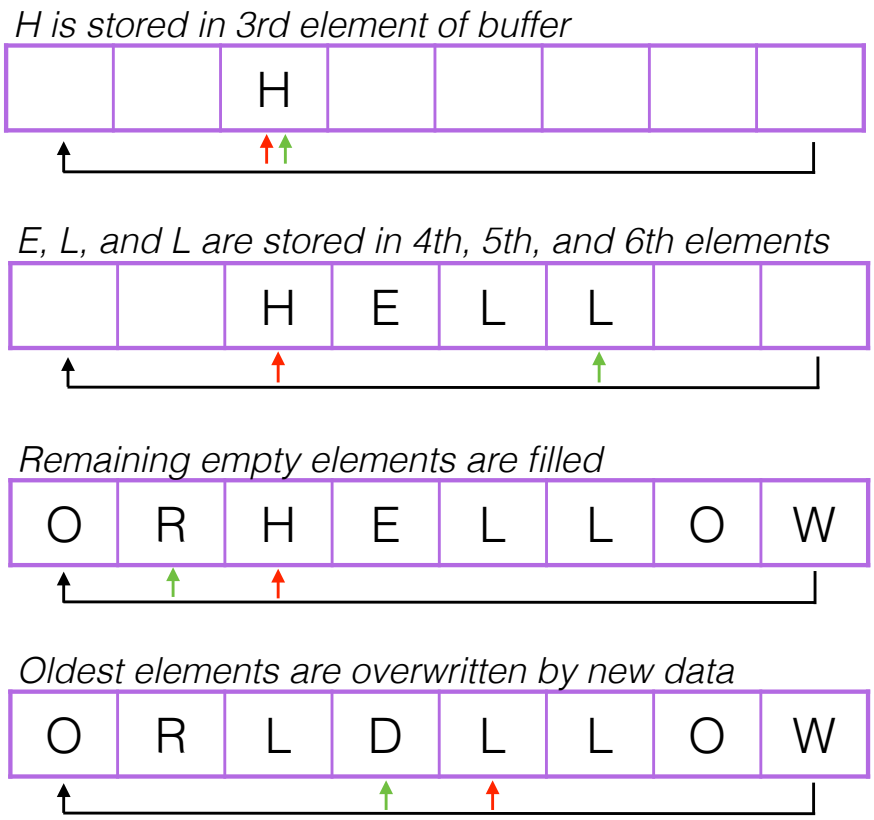


Figure 6-33: Demonstration of a circular-buffer array storing the phrase 'HELLOWORLD' in an 8-element array. *The red arrow indicates the oldest element in the array and the green arrow indicates the newest. Once the array is full, the oldest elements are overwritten to store new data. The location of the first element in a circular-buffer array does not matter; for example, the third element of the buffer stores the first character 'H'.*

A single 512-byte circular-buffer array is used to temporarily store the incoming EMG signal until an FP has been detected. Upon FP detection, the data in the circular-buffer array forms the first 512-byte block of data stored in the microcontroller's SRAM. After detection, the remaining nine 512-byte blocks that the SRAM can hold are filled with the incoming data that follows the detection of the FP; all ten 512-byte blocks in the buffer array are then written to the SD card. Using a circular buffer in this way means that the unnecessary 'empty' data (not containing an FP) does not need to be stored and saved, taking up valuable space in memory and time by writing to the SD card. This process is demonstrated in Figure 6-34 using an 8-element circular-buffer array; the circular-buffer stores ascending numbers, and when triggered by the value '12' the circular-buffer forms the first 512-byte block. The following numbers are then stored in the remaining blocks in SRAM.

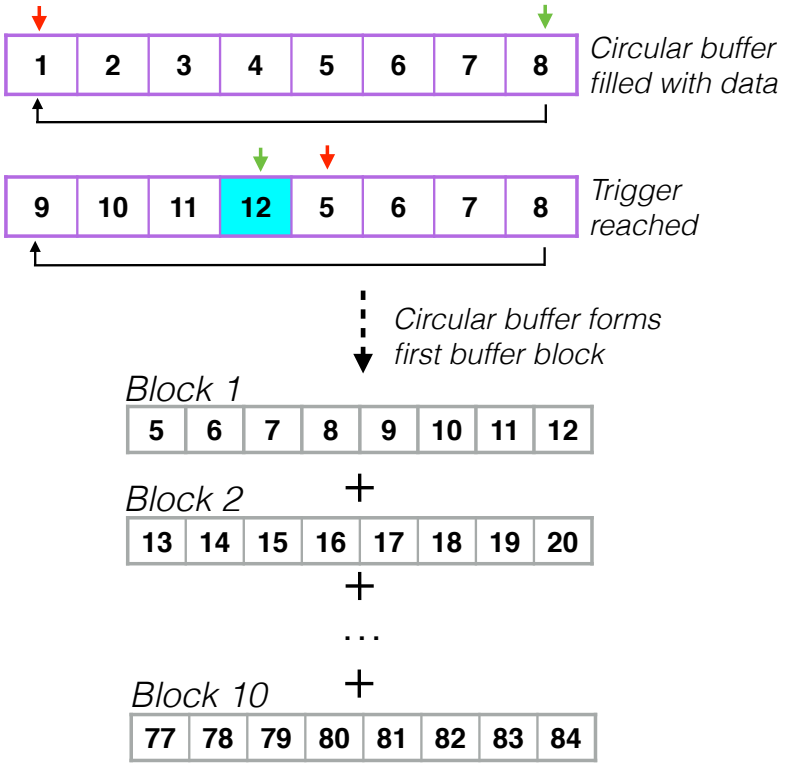


Figure 6-34: Schematic of circular-buffer array and buffer blocks stored in SRAM. *The red arrow indicates the oldest element in the array and the green arrow indicates the newest. Once the system is triggered by the value '12' (highlighted in blue), the circular buffer forms the first block of data that is stored in SRAM. The remaining buffer blocks are filled with the succeeding data values. The data stored in the SRAM buffer is then written to the SD card.*

6.2.3.3 SD card writing tests

The full device performance from signal input to writing to the SD card was tested using the same laboratory setup described in Section 6.1.10 and equipment described in Section 2.4.2. For simplicity, in order to detect an FP from the incoming signal, an amplitude threshold is set; when the device is triggered by a threshold value, it will write short periods of data to the SD card using the circular-buffer array just described in Section 6.2.3.2. During testing, the sampling rate was left 'open' and unlimited in order to test the maximum speed the device can log data at. The results were impressive; the device can write data to the SD card at a rate of 3.97 kHz (mean value), surpassing the sampling rate of the TMSi Refa (2.048 kHz) used for collecting HD-sEMG in the longitudinal studies.

Figure 6-35 shows how well the device can sample high-frequency signals such as 100 Hz and 500 Hz.

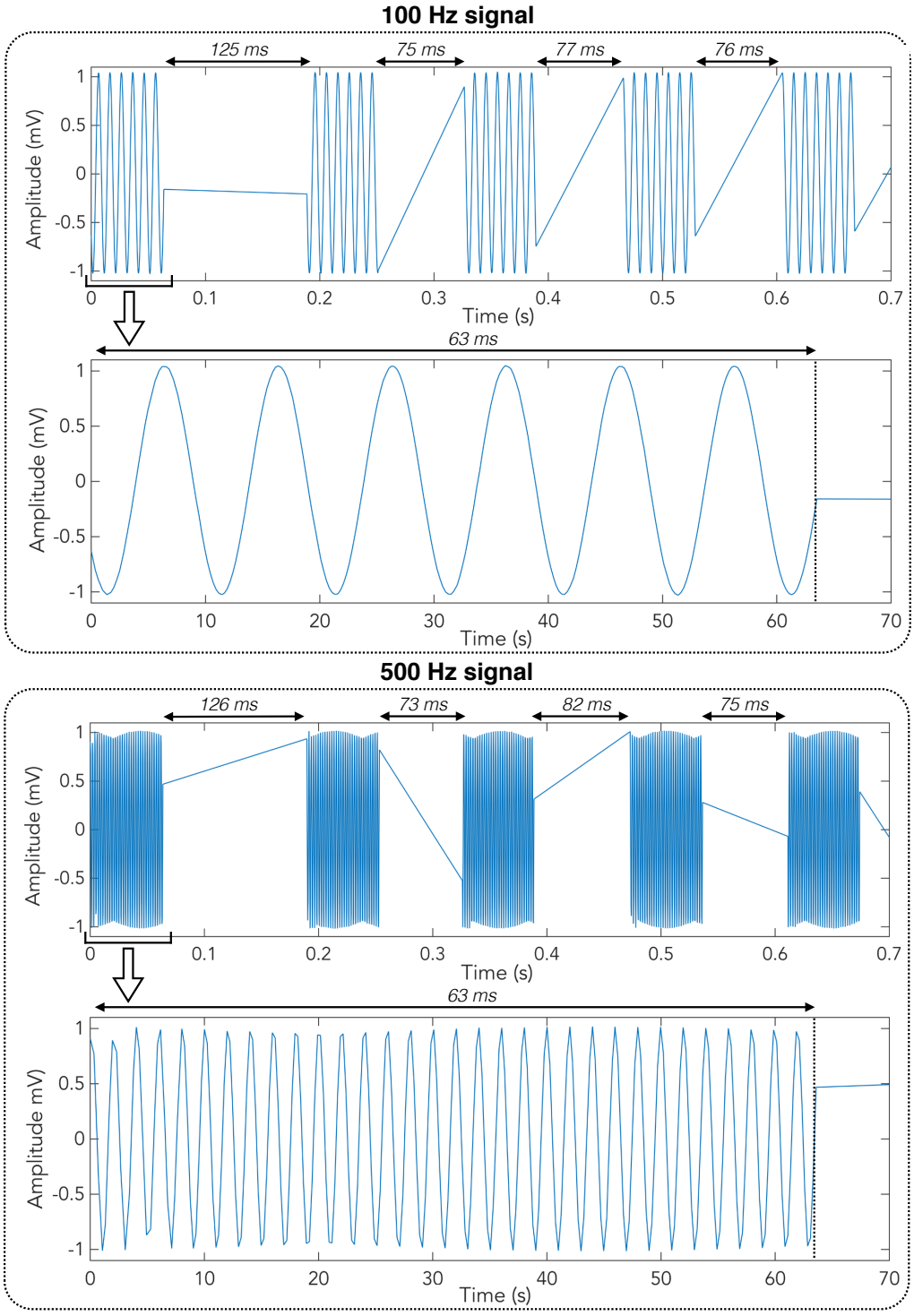


Figure 6-35: SD card writing test with 100 Hz and 500 Hz sine waves. *Higher sampling rate of SD card data storage using data buffers in SRAM has resulted in a high quality signal. Gaps in the signal are due to the write cycle of the SD card. There is some loss of data sampling the 500 Hz sine wave, however this is at the top-end of the frequency spectrum used for surface EMG. Each section of sampled signal is 63 ms long and sampled at 3.96 kHz. The first write cycle (first time writing to the SD card file) takes longer than each successive write cycle, taking approximately 126 ms for the first write compared to 76ms for each successive write. In each case, one channel of eight is shown.*

There is some loss of information when sampling at 500 Hz, however, as discussed in Section 3.2.1, this is at the very top end of the frequency spectrum when analysing surface EMG signals (signals are band-passed from 20-500 Hz) and therefore is not a problem for this application.

Sampling rate discussion

As discussed in Section 6.1.8.2, the data-rate of the ADS1298 is set to 16 kSPS, which is considerably faster than the sampling rate that has been achieved when storing data in a buffer array in the microcontroller's SRAM (3.97 kHz). Despite efforts at improving code efficiency in order to speed up the buffer storage process, the sampling rate of the device appears to be limited to 3.97 kHz. Several data rates of the ADS1298 have been experimented with, as seen in Table 6-8, however, it seems that the device cannot sample and store data to SRAM faster than 3.97 kHz, despite the high data-rate of the ADS1298.

Table 6-8: Effect of ADS1298 data rate on device SD card sampling rate. *At higher data rates, the device can still not store data in SRAM faster than 3.97 kHz.*

ADS1298 data rate	Sampling rate achieved
16 kSPS	3.97 kHz
8 kSPS	3.97 kHz
4 kSPS	3.74 kHz
2 kSPS	1.90 kHz

A higher sampling rate is desirable to ensure that the EMG signal is properly sampled, enabling the capture of all high frequency components of the signal. However, sampling at a higher rate will also mean shorter amounts of signal being written to the SD card; 250 samples stored in the buffer array equates to 125 ms if sampling at 2 kHz, but equates to just 63 ms if sampling at 3.97 kHz (as seen in Figure 6-35). Sampling at this higher rate would mean that the 'tail-end' (latter portion) of FP waveforms longer than 63 ms would not be recorded. I have investigated the distribution of the length of FPs using the EMG recordings from the longitudinal patient study conducted at King's College London (KCL); the histogram in Figure 6-36 shows the frequency of FP lengths from nearly 8430 FPs, detected using the data processing pipeline described in Chapter 3 and

Chapter 4. Out of the 8430 FPs detected, 85% of them have an FP length shorter than 63 ms and 98% of them are shorter than 125 ms; sampling at a higher rate will exclude the tail-end of only a small portion of FPs. Note, that signal noise or ringing caused by the application of a notch filter (see Section 3.2.2) can exaggerate the length of an FP.

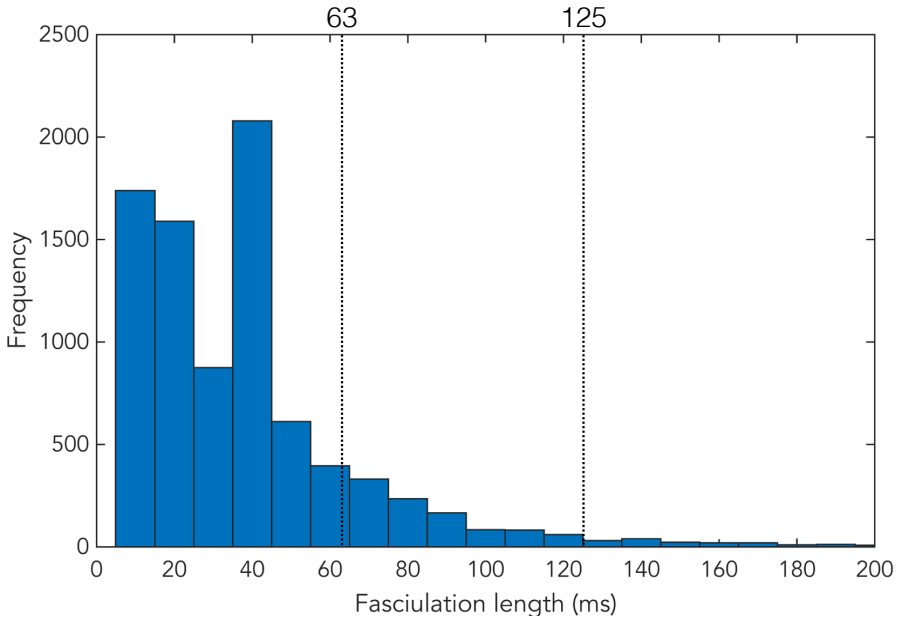


Figure 6-36: Histogram of FP length. *FPs were detected using the probabilistic spike detection method; 8430 FPs were detected using six hours of patient data from five different patients at different stages of ALS. Using the higher sampling rate, only FPs with lengths less than 63 ms will have their full waveform captured; the tail-end of longer FP waveforms will not be recorded.*

Based on these results, I have made the decision that the device should capture data at its maximum rate of 3.97 kHz (63 ms recording length). This will capture the full waveform of a large majority (85%) of detected FPs at a very high sampling rate and will result in only a small proportion of FPs not having their tail-end recorded. Figure 6-37 demonstrates the periods of reading and writing data for a typical FP in an EMG signal.

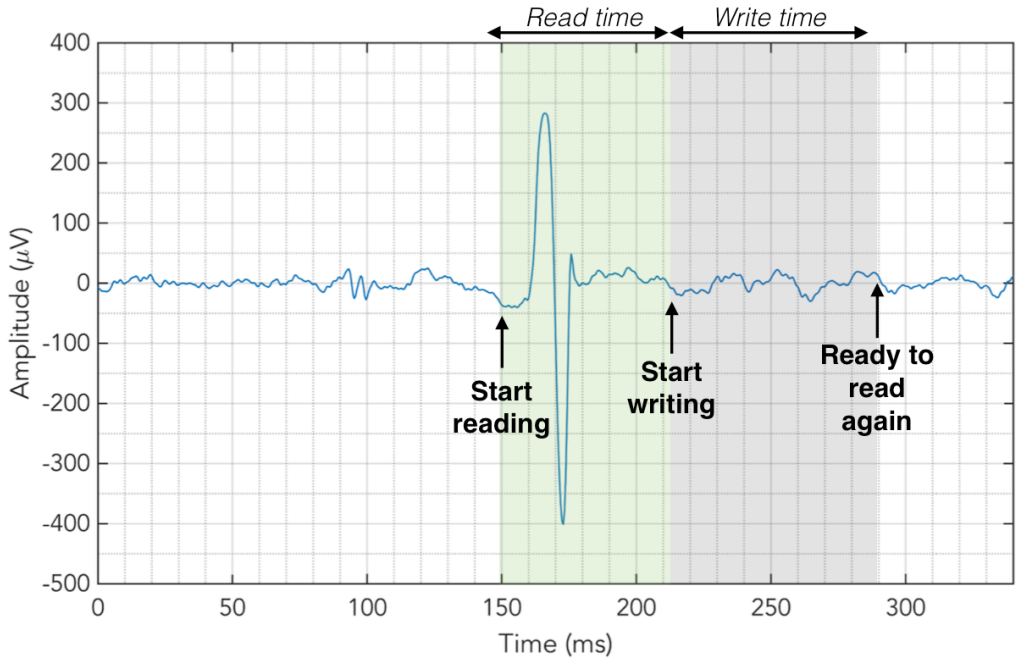


Figure 6-37: Read and write time of an FP to the SD card. *Using the highest sampling rate the device can output (3.97 kHz) to save data to the SD card means the device will be able to capture 63 ms of high temporal resolution data at a time. Once captured, the device must write the data to the SD card, during which the device cannot capture data.*

6.2.3.4 Bluetooth streaming and SD card testing

The device was tested using both the Bluetooth transceiver to send continuous data wirelessly to a nearby computer and to write short 63 ms bursts of data to an SD card (as just described in Section 6.2.3.3). Unfortunately, the slow Bluetooth transmission provides a 'bottleneck' for the entire system; the device must wait for each piece of data to be transmitted by Bluetooth before it can continue with the rest of its processes. This means the rate at which data are stored in the buffer array is slowed down to the rate at which the device can transmit data over Bluetooth; this is approximately 500 Hz for sampled data and a time-stamp (see Section 6.1.10.1), which is too slow a rate to sample EMG data at. Furthermore, the device is occupied during the SD card write process, meaning it cannot transmit data over Bluetooth. Therefore, it is recommended that the Bluetooth transceiver is only used for issuing commands and for checking the quality of each channel before starting the recording session; at this point only the buffer array and SD card should be used to record FPs in order to ensure the system records at its maximum sampling rate.

6.2.3.5 EMG signal test

A section of EMG recording from the longitudinal patient study at KCL (shown in Figure 6-38) was loaded onto the Agilent signal generator in order to test the device's efficacy at capturing FPs with real patient data. The signal has been band-passed filtered from 20-500 Hz and notch filtered at 50 Hz. The pre-recorded signal was 'played-back' through the input of the device. For the purposes of this test, the microcontroller was instructed to capture twenty FPs from the signal and then to stop recording. When the signal exceeds a threshold value, the buffer array is triggered to start capturing data (see Section 6.2.3.2). Figure 6-38 shows the EMG signal input into the device (blue trace) with the FPs that have been detected overlaid (red traces). Figure 6-39 provides a plot of each FP captured by the device (for one of the device's eight channels).

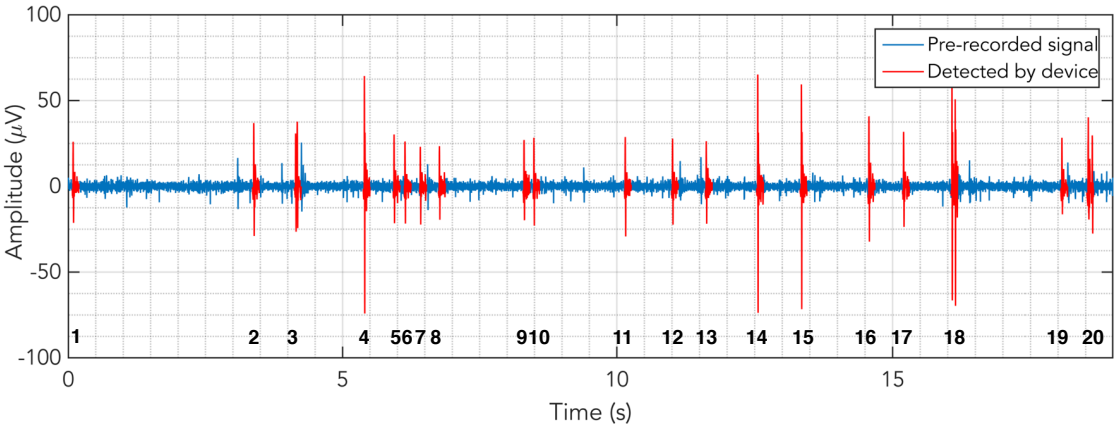


Figure 6-38: Pre-recorded signal and FPs detected by wireless device. *A signal recording from the longitudinal study at KCL (blue) was inputted into the device using a signal generator. The device captured 20 FPs (red) over a certain threshold. (Only the FPs captured from one of eight channels is shown).*

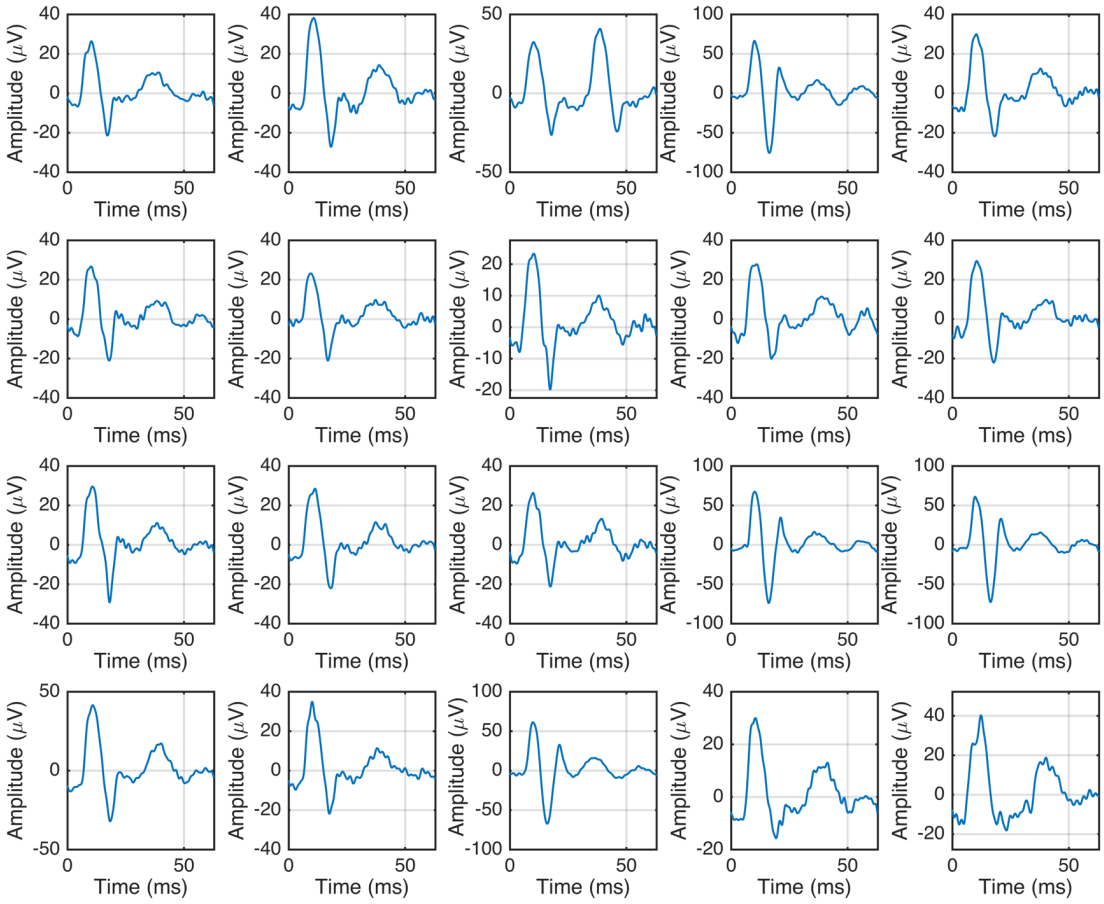


Figure 6-39: FPs captured by device from pre-recorded EMG recording. *When the device was triggered by the input signal, it captured and saved the FPs shown. Each FP recorded is 63 ms long as the device can save 250 samples of data at a rate of 3.97 kHz. Only the FPs captured from one of eight channels is shown.*

Figure 6-40 shows one of the FPs captured by the device overlaid on top of the signal recording. The recorded FP is very similar to the original signal inputted into the device; the device has captured the FPs from the pre-recorded signal with low noise, high resolution, and high sampling rate.

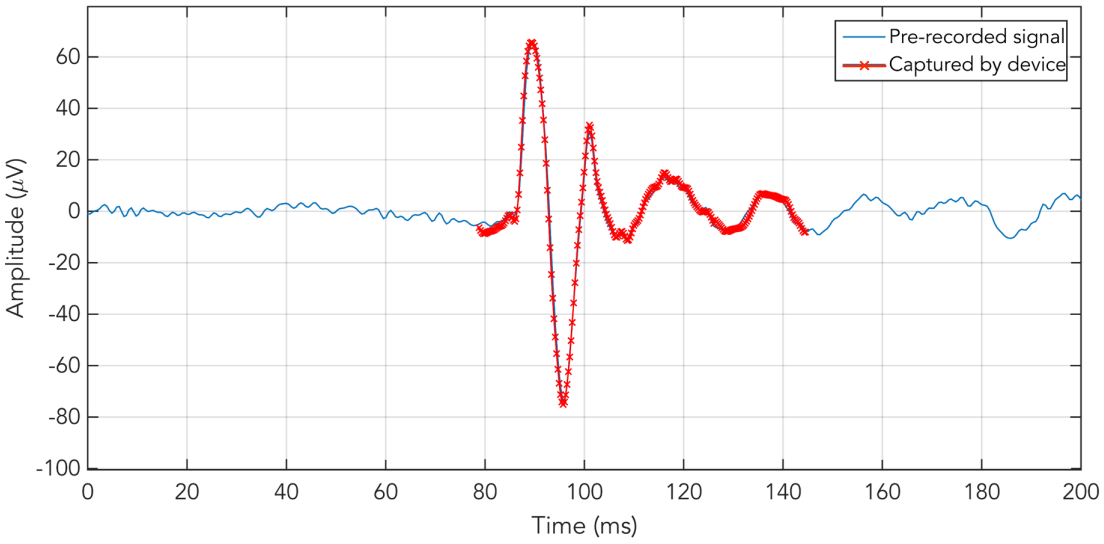


Figure 6-40: FP captured by device overlaid on input signal. *An FP captured by the device overlaid on top of the EMG recording from the signal generator. The data-points have been marked with an 'x' to highlight the high sampling rate of the device. The signal captured by the device has been band-passed from 20-500 Hz.*

6.3 Arduino Programming

As previously discussed, the device microcontroller is bootloaded with the Arduino bootloader (see Section 2.5.1), meaning that the device can be programmed through the Arduino IDE. Commands can be sent to the device over the serial port through the Arduino IDE in order to control the program that has been uploaded onto the device. The Bluetooth connection of the device means that it can also be programmed and commands can be issued to it via a wireless connection through the microcontroller's serial port. This section will cover the set up of the device, how the device operates after set up, and importantly, how data is handled and stored in different data structures in the microcontroller's SRAM.

6.3.1 Data handling

The captured data are to be initially stored in a circular-buffer array, previously described in Section 6.2.3.2. When triggered, the full capacity of the allocated memory buffer in SRAM will be used to store the incoming data (note that the memory already stored in the circular buffer will form the first block of data in the SRAM data buffer, see Figure 6-34). The trigger for initiating storage in the

buffer array used is an amplitude threshold; when the incoming data exceeds the threshold, the buffer array will start storing data.

Data is stored in a series of nested structures, as shown in Figure 6-41. The initial structure *'buffStruct'* stores the current time as a 32-bit long, and a single value recorded from each channel as eight 16-bit integers. These 20 bytes of data are nested within the next data structure *'buffStore'*. This structure is 512 bytes and has room to store 25 *buffStruct* structures; the remaining space in the structure is filled with empty/dummy data. Each *buffStore* is a 512-byte *block* of data.

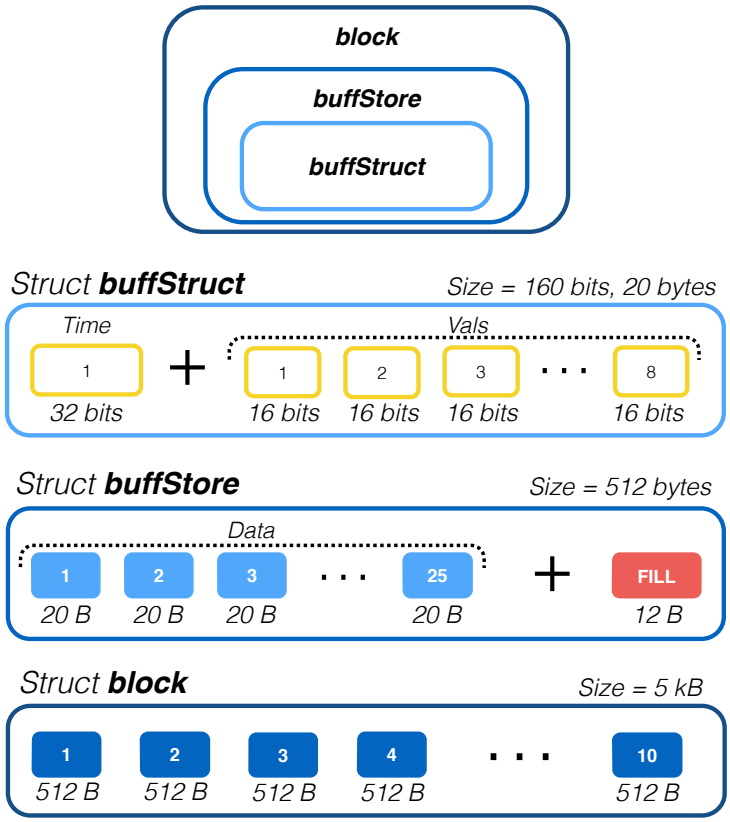


Figure 6-41: Data is stored in the SRAM using a series of nested structures. *The time-point and value for each sampled piece of data is saved to the structure 'buffStruct'. This is stored within the 512-byte structure 'buffStore', which has room for 25 buffStructs; the remaining space within the structure is 'filled' with empty/dummy data. The structure 'block' stores ten 512-byte buffStores as a buffer array.*

Individual locations can be read from/written to using these cascading structures. Note that when accessing locations in the buffer array, counting starts from zero '[0]'. For example, *'block[0].data[0].time'* will access the first timestamp recorded in the first 512-byte block of data, and

'*block[5].data[2].vals[7]*' will access the eighth channel's third datapoint in the sixth 512-byte block of data.

6.3.2 Device set up

6.3.2.1 SPI transfer

As the ATMEGA2560 is an 8-bit microcontroller, it will only be able to process 8 bits of data at a time. Issuing the command 'SPI.transfer' in Arduino code will send 8 clock pulses over the SCLK of the SPI port. This can be used to retrieve data from peripheral devices (i.e. *clocking out* data from the ADS1298), or issuing commands/instructions (i.e. *clocking in* configuration data to the ADS1298). This function is used frequently for controlling the data flowing in and out of the microcontroller.

6.3.2.2 Set up protocol

The pins of the ATMEGA2560 required for the device to operate (see Figure 6-30) are set to either input or output function and assigned a logic state of either LOW or HIGH to start in upon boot up. The SPI port is configured in the initial set up of the device, setting the bit order to 'most-significant-bit first', the SPI mode to 1 (see Appendix B), and the rate of the SPI clock to 4 MHz (the maximum rate is half the rate of the ATMEGA2560 clock, 8 MHz). The data structures described in Section 6.3.1 are created, and the serial port is opened to allow the transfer of data over Bluetooth. The configuration of the ADS1298, previously described in Section 6.1.4.4, is done by passing information over the SPI port. The chip-select of the ADS1298 is brought low, and commands are written to the ADS1298's registers using the SPI.transfer function.

6.3.3 Data capture loop

The Arduino program has a main loop, which will constantly cycle, carrying out the functions that are written within it (as demonstrated by the flowchart in Figure 6-42). The loop will initially wait for a keyboard input via the Bluetooth serial port before starting to capture data.

Once the program has received a keyboard input, it will check if the ADS1298 has converted data ready yet (DRDY = 0). If not, the loop will continue

to check this until it has. When data are ready, the data are retrieved from the ADS1298 over the SPI port using the SPI.transfer function. A single channel of the retrieved data is transmitted via Bluetooth to the tablet/computer. The channel of data that is to be streamed is determined by a button press; the button press is attached to an interrupt pin, so the channel being streamed can be changed at any time. The loop continues to check if the ADS1298 has converted data yet (DRDY = 0). If not, the program will wait until it has. The reason for streaming this data over Bluetooth is so the user can check for signal quality, and can check that all the sensors are properly connected to the patient. When the user is satisfied with the signal quality, a second keyboard input will start logging data to the SD card.

Upon the second keyboard input, the data will begin to be stored in the circular buffer. If the data have not yet reached the threshold value, each piece of newly sampled data that has been retrieved from the ADS1298 (when DRDY = 0) is continually stored in the circular buffer. If the data have reached the threshold value, the buffer array is first filled with the data currently in the circular buffer followed by the successive data captured by the device. When the buffer array is full, all ten 512-byte blocks of data stored in the microcontroller's SRAM are written to the SD card. These data are written as a binary file to speed up the writing process. After the buffer is written to the SD card, the buffer is cleared and the device continues to fill the circular buffer with new data, waiting for the data to exceed the threshold value again.

When the program receives a third keyboard input, the capture of data will end, and whatever data was written to the SD card file will be converted into Comma Separated Value (CSV) file format. This conversion can take some time, especially if the binary file is large.

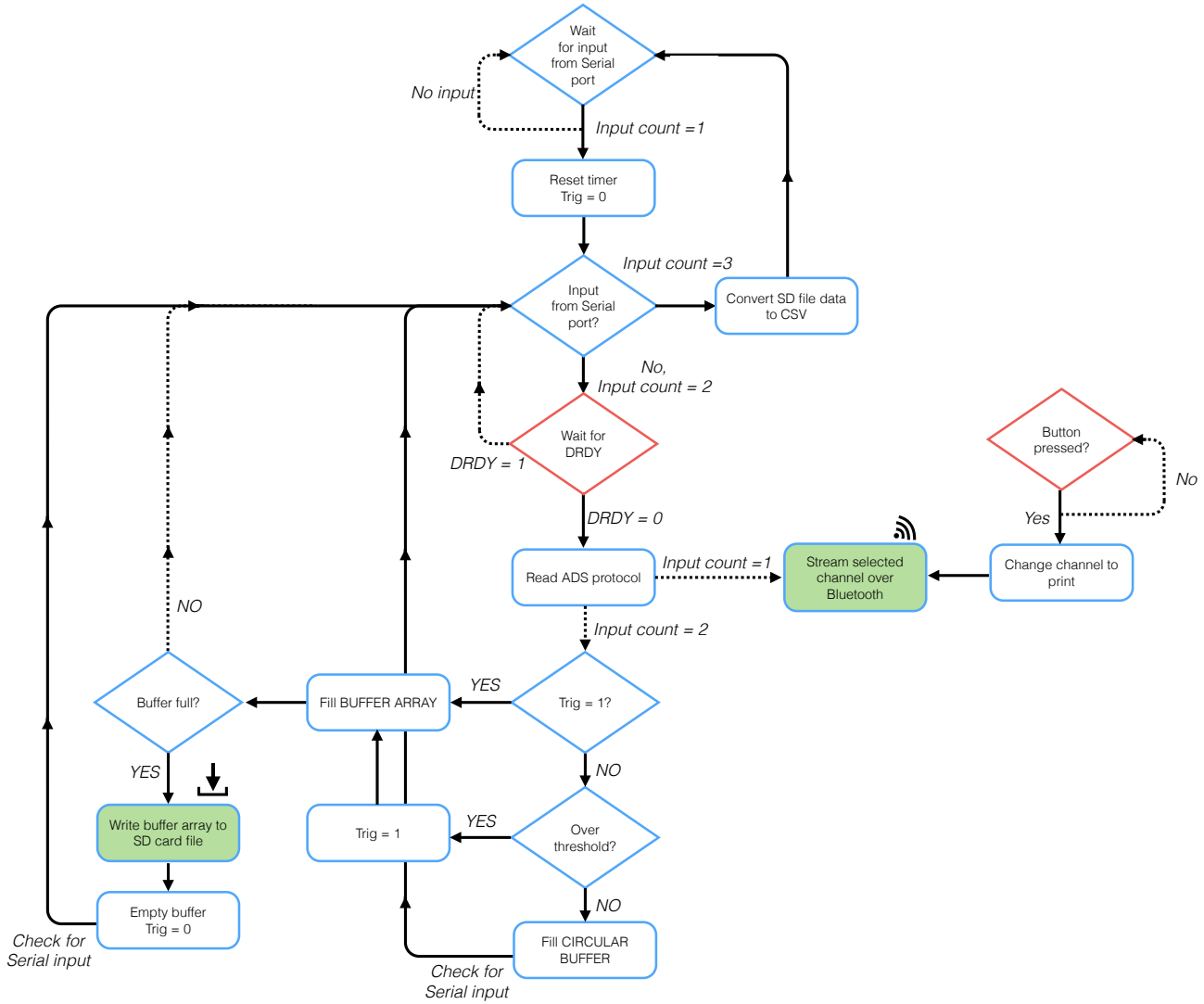


Figure 6-42: Flowchart of data capture loop. *Data are either streamed via Bluetooth or written to the SD card depending on the keyboard input from the user. The red cells indicate points in the data capture loop that are controlled by the interrupt pins of the microcontroller: DRDY input from ADS1298 and the push button included in the device design.*

6.4 Conclusion

A compact, wireless device for capturing high-resolution, multiple-channel data has been designed and built. The wireless device is capable of transmitting data over Bluetooth to a nearby tablet or computer and separately logging data to an SD card at a very high rate. The microcontroller of the device has been bootloaded with the Arduino bootloader and programmed through the Arduino IDE.

The Analogue Front End (AFE) of the device includes an array of buffer op-amps for interfacing with high-impedance sensors, such as surface EMG. An array of non-inverting op-amps then amplifies the signal by 101 times. Following this amplification, the signals then enter the ADS1298, a highly specified, 24-bit, low noise Analogue-to-Digital Convertor (ADC) and programmable gain amplifiers (PGAs). Before the signals are digitised, the PGA amplifies the signal by 12 times, making the total amplification of the device's front end equal to 1212 times. The 24-bit digital signal is transferred over SPI to the ATMEGA2560 microcontroller; only 16-bits of the digital signal are used as this will not only provide sufficient resolution, but will reduce the load imposed on Bluetooth transmission and data storage.

The Bluetooth transceiver is used for issuing commands to the device and for streaming a single channel of data. This Bluetooth stream is used to check individual channel quality and to ensure the electrodes are connected properly and have a good skin-electrode contact. The user can cycle through the channels using the push-button incorporated into the device design. Future development of this device would see this channel-checking process done automatically, using a measure such as 'signal activity' for checking channel quality, as previously described in Section 3.2.3. Channel quality could be assessed automatically, informing the user which electrode contacts need to be adjusted or reapplied.

In order to save the data at a high rate, the microcontroller's SRAM is utilised. The incoming signal is stored in a temporary circular-buffer array until the signal exceeds a threshold (indicating a possible FP), triggering the device to start storing incoming data into a data buffer in the microcontroller's SRAM. The

data buffer is formed of ten 512-byte blocks, which, when full, are written to the SD card in a single write procedure. This is a faster, more efficient way of logging the incoming data, rather than saving each piece of sampled data individually. The device stores and writes data at its maximum rate (approximately 4 kHz). This means the device can save short bursts of high temporal resolution data, making it ideal for capturing FPs. As discussed in Section 3.3, threshold detection using an arbitrary value is not necessarily the best way to detect FPs; future development of this device would see the device using a detection method such as the probabilistic spike detection method described in Section 3.3.1. If using this method of FP detection, the device would require a larger buffer to store data in, in order to be able to properly assess the incoming signal. Therefore, additional SRAM would be required, which could be provided via an external memory chip included into future device designs.

Chapter 7
Transdermal Electrodes

In this chapter I describe experiments to prototype a novel transdermal electrode using shrinkable resin technology, with the aim of producing lower noise EMG signals and transdermal sensors for measuring lactate.

A transdermal device should only penetrate several hundred micrometres through the top layers of skin (typically 200-300 μm, and up to 1000 μm); it should not cause any pain or draw any blood.[80–83] For this reason, the concept of transdermal devices is becoming increasingly attractive for use in electromyography (EMG) and electrophysiology, biological monitoring, and drug delivery. This is a rapidly emerging field with the current literature describing many different fabrication methods for producing transdermal devices and microneedle arrays (MNAs).[80–88] However, none of these methods offer a low-cost route without the need for specialist equipment or specialist training/personnel. This can represent a significant challenge for innovators who are new to this field undertaking preliminary investigations into transdermal devices. It also inhibits experimentation and optimisation of characteristics such as spike/microneedle height and geometry, and array layout.

This chapter describes the development, realisation, and experimental results of a new, innovative method for producing transdermal devices/MNAs. The method uses 'casting-shrinkage cycles', originally described by Sun *et al.* for

producing microfluidic devices.[49] I have adapted the casting-shrinkage method, which is presented in this chapter as a new route for prototyping transdermal biomedical measurement devices.

7.1 Skin-Electrode Impedance

Surface EMG electrodes (sEMG) have gained popularity in recent years in research investigations and for clinical use due to the fact that they are non-penetrative, painless, and do not require sterilisation, in contrast to penetrative needle EMG electrodes (as discussed in Section 1.5). The drawback of sEMG electrodes is their high susceptibility to noise due to the high impedance contact made between the electrode and the top layer of skin, known as the *stratum corneum* (see Figure 7-1). As covered previously in Section 3.2, removing noise from the recorded EMG signal without distorting the underlying signal is a difficult challenge. Using transdermal sensors to penetrate through the highly resistive *stratum corneum* could provide a painless, low impedance solution that benefits both the patient and clinicians.

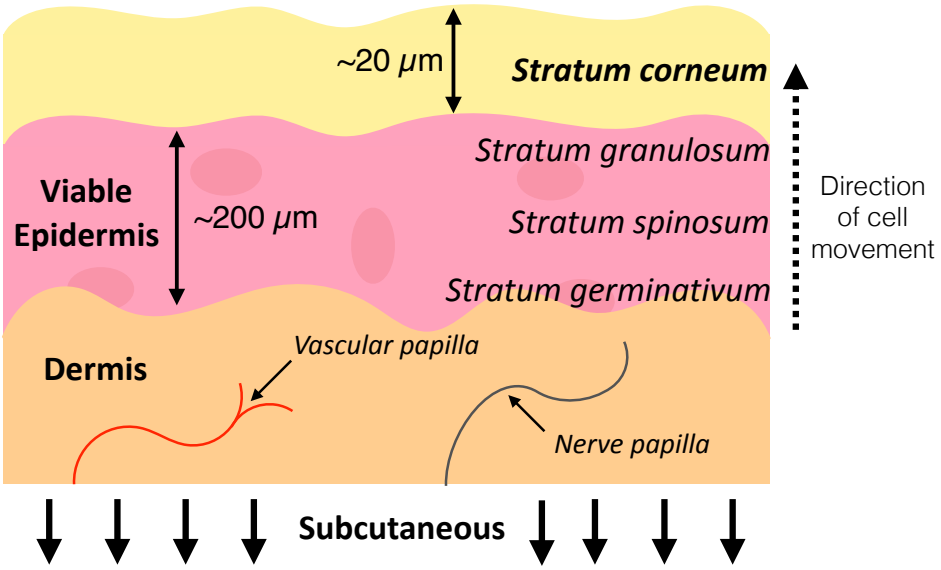


Figure 7-1: Diagram of skin layers. *The epidermis contains the stratum germinativum, the stratum spinosum, the stratum granulosum, and the outermost stratum corneum. Cells are produced in the stratum germinativum and grow outwards. The outermost stratum corneum consists of dead cells and oils, forming a highly resistive layer. The layers of live tissue underneath remain viable, and are much less resistive. The dermis is below the epidermis, and contains blood vessels and nerves.*

The skin is made up of three principle layers: the epidermis, the dermis, and the subcutaneous layer. The epidermis layer consists of four sub-layers: the innermost *stratum germinativum*, the *stratum spinosum*, the *stratum granulosum*, and the outermost *stratum corneum* (see Figure 7-1). New epidermis skin cells are produced in the *stratum germinativum* and are pushed outward in the *stratum spinosum* as new cells grow underneath them and take their place. By the time the skin cells have passed through the *stratum granulosum* and into the *stratum corneum* they have degenerated and died. Because the *stratum corneum* is made up of oils and dead skin cells, it has very different electrical characteristics compared to the layers of live tissue underneath.[39] The *stratum corneum* results in a skin-electrode contact with a very high electrical impedance and thus a noisy signal. This can be improved by preparing the patient's skin with cleaning and abrasion or by applying conductive gels, as previously discussed in Section 2.2.3. However, despite these precautionary steps, noise can still be found on the recorded signal. Penetrating through the *stratum corneum* into the more conductive layers of skin would improve the electrode-tissue contact, however, the dermis and deeper layers of tissue contain vascular and nerve papillae (see Figure 7-1); penetrating into these layers would be painful and draw blood. An electrode that can penetrate through the *stratum corneum* into the viable sub-layers of the epidermis but not as deep as the dermis layer, would improve the electrode-skin contact without causing pain or drawing blood. This is classed as a 'transdermal' or 'semi-penetrative' electrode and greatly improves the electrical contact, lowering the signal-to-noise-ratio (SNR).

7.1.1 Surface EMG electrode model

The resistive and reactive components of the skin-electrode contact of an sEMG electrode (using an electrolyte gel) can be modelled as a combination of resistor-capacitor (RC) equivalent circuits, as seen in Figure 7-2.

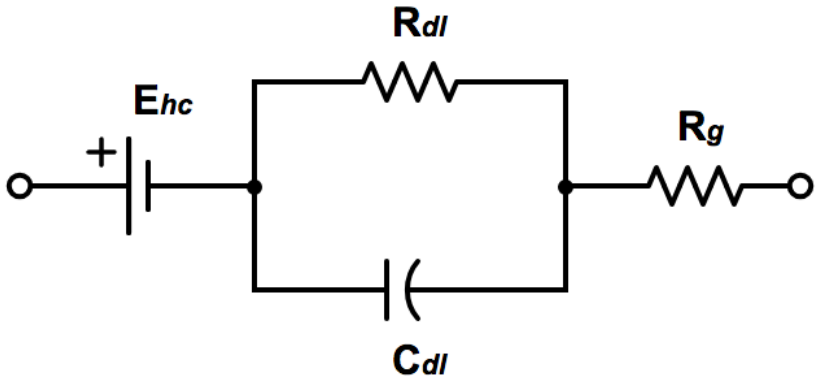


Figure 7-2: Electrical equivalent circuit of electrode-electrolyte contact. *The resistor-capacitor (RC) circuit describes the impedance at the interface of the sEMG electrode and electrolyte. Modelled are the half-cell potential of the electrode (E_{hc}), a parallel RC circuit representing the capacitance caused by the double layer of charge at the electrode-electrolyte interface and its leakage resistance (C_{dl} and R_{dl}, respectively), and finally the resistor in series representing the resistance of the electrolyte gel (R_g).*

A layer of surface charge forms at the electrode-electrolyte interface, caused by chemical interactions between the electrolyte and electrode surface. A secondary layer of charge forms in the electrolyte caused by oppositely charged ions in the electrolyte being attracted to the first layer of surface charge. In this electro-equivalent model, C_{dl} represents the capacitive component of the double layer of charge that forms at the electrode-electrolyte interface. The resistive component of the electrode-electrolyte interface R_{dl} is due to the leakage resistance of the capacitance across the double layer of charge.[39] The resistive and reactive components of this interface could be modelled as a series RC circuit, however as shown in Equations (7-1) to (7-3), using a series RC model would suggest an impedance that tends towards infinity for the lower signal frequencies close to direct current (DC):

$$Z_{cap} = \frac{1}{j\omega C} = 0 - j(\frac{1}{\omega C}) \quad \textbf{(7-1)}$$

$$\left|Z_{cap}\right| = \sqrt{Real^2 + Imaginary^2} = \sqrt{0^2 + (\frac{1}{\omega C})^2} = \frac{1}{\omega C} \quad \textbf{(7-2)}$$

$$\omega \to 0, \qquad \left|Z_{cap}\right| \to \infty \quad \textbf{(7-3)}$$

Instead, the electrode-electrolyte contact should be correctly modelled as a parallel RC circuit. As shown in Equations (7-4) to (7-7), a parallel RC circuit

gives a purely resistive impedance as frequencies approach DC. At higher signal frequencies the impedance will decrease.

$$\frac{1}{Z} = \frac{1}{Z_r} + \frac{1}{Z_{cap}} \tag{7-4}$$

$$\frac{1}{Z} = \frac{1}{R} + j\omega C = \frac{j\omega CR + 1}{R} \tag{7-5}$$

$$|Z| = \frac{R}{\sqrt{1+(\omega RC)^2}} \tag{7-6}$$

$$\omega \to 0, \quad |Z| \to R \qquad \omega \to \infty, \quad |Z| \to 0 \tag{7-7}$$

The electrode-electrolyte contact can be fully modelled as an electro-equivalent circuit consisting of a voltage source (half-cell potential), a parallel RC circuit (due to the double layer of charge), and an additional resistor R_g in series (resistance of electrolyte gel), as seen Figure 7-2. This electro-equivalent circuit can be developed further to model the impedance effect of the skin-electrode contact when placed on skin. Following the series resistance R_g, a potential difference E_{sc} is modelled due to the difference in ionic concentration across the semi-permeable *stratum corneum*. The highly resistive *stratum corneum* is modelled as the resistance R_{sc}. The epidermis also behaves like a parallel RC circuit; as Equation (7-6) describes, Rosell *et al.* have shown that impedance decreases at high frequencies and increases at low frequencies near DC,[89] with the dermis and subcutaneous layers underneath it behaving as a resistance, R_{sub}. The full electrical equivalent circuit is modelled in Figure 7-3 and the full impedance of the wet electrode-skin contact is described in Equation (7-8).

$$|Z_{wet}| = \frac{R_{dl}}{\sqrt{1+(\omega R_{dl} C_{dl})^2}} + R_g + R_{sc} + \frac{R_{epid}}{\sqrt{1+(\omega R_{epid} C_{epid})^2}} + R_{sub} \tag{7-8}$$

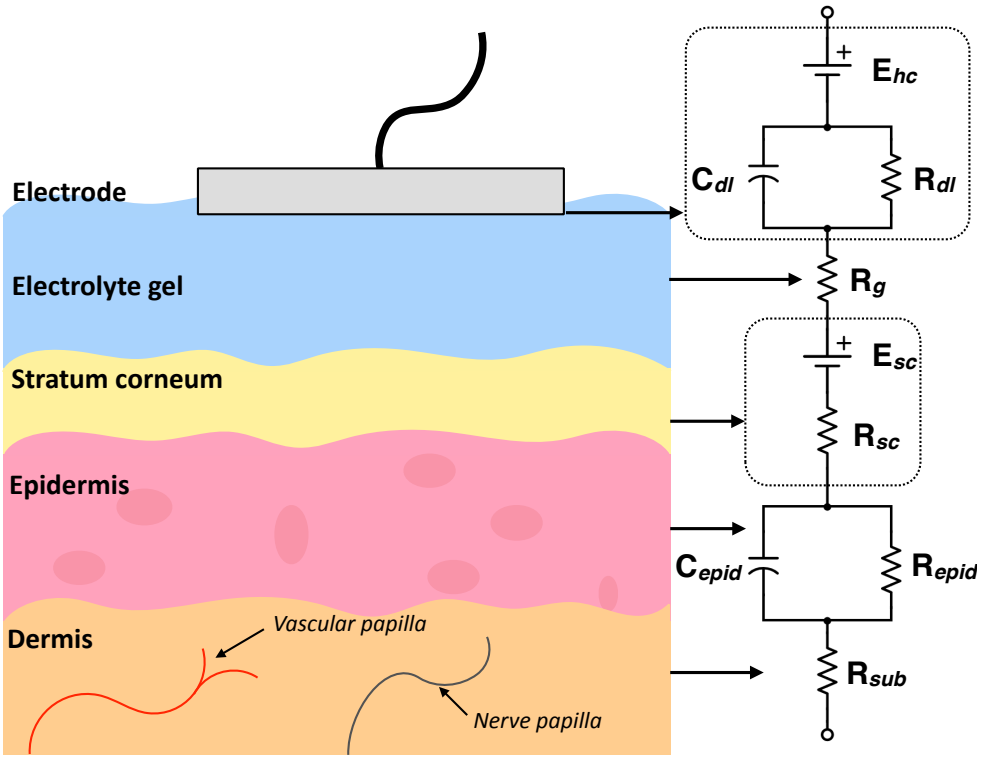


Figure 7-3: Full electrical equivalent model of wet sEMG skin-electrode contact electrode. *The potential difference caused by the difference in ionic concentration across the stratum corneum is modelled by E_{sc}, which is followed by the resistance of the stratum corneum R_{sc}. The epidermis layer can also be modelled as a parallel RC circuit, followed by the dermis and subcutaneous layers being modelled in series as a resistance R_{sub}.*

These resistances and capacitances accumulate to form the impedance modelled in Equation (7-8). If Z_{wet} is large, small currents caused by noise can manifest themselves as large voltages, hence why electrodes with high-impedance interfaces can be very susceptible to noise, such as sEMG. This noise can be removed by filtering, but as previously discussed in Section 3.2, this can distort the underlying waveform morphologies of the fasciculation potentials. Alternatively, removing the *stratum corneum* can reduce the skin-electrode contact impedance that affects sEMG electrodes. This can be done as previously described in Section 2.2.3, or as described by Webster, by *'vigorous rubbing with a pad soaked in acetone or abrading the stratum corneum with sandpaper to puncture it'*.[39] Neither of these methods of skin preparation would be particularly pleasant for the ALS patient, especially as ideally they should be monitored on a regular basis.

7.1.2 Transdermal/microneedle array electrodes

Using a transdermal/microneedle array (MNA) electrode, the electrode can penetrate through the *stratum corneum* layer without the need to abrasively remove it, interfacing directly with the conductive layers of the epidermis below (see Figure 7-4). As a result, the impedances typically associated with skin-electrode contact would be reduced, as shown in Equation (7-9). Ideally this type of electrode would not penetrate as deep as the dermis layer, which contains nerve and vascular papillae, so as not to cause pain or draw blood.

$$|Z_{MNA}| = \frac{R_{epid}}{\sqrt{1+(\omega R_{epid} C_{epid})^2}} + R_{sub} \quad \textbf{(7-9)}$$

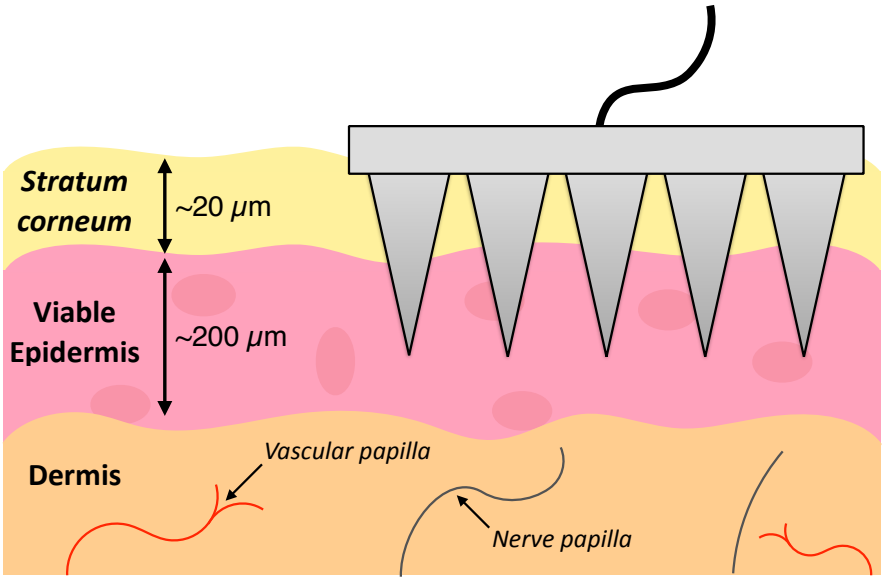


Figure 7-4: Microneedle array penetrating through the *stratum corneum*. *The transdermal device that has penetrated through the highly resistive stratum corneum into the conductive epidermis and is in contact with interstitial fluid, making it ideal for low noise electrophysiology, drug delivery, biosensing etc. Transdermal devices should not penetrate as deep as the dermis so that they remain painless and do not draw blood.*

Penetrating through this skin layer into the conductive layers of the epidermis would greatly improve the electrical contact for electrodes (i.e. for electrophysiology), improving the signal-to-noise ratio (SNR). Of course the best SNR would be achieved by penetrating further into the tissue, e.g. into the dermis/muscle like a true needle EMG electrode, however this would be painful and unlikely to be popular with patients who are being monitored on a regular basis.

Transdermal devices/MNAs have many uses that extend further than just electrophysiology. Penetrating through the first layer of skin gives the MNA access to interstitial fluid and thus would allow for the delivery of substances such as therapeutic drugs[90–92] or provide a minimally invasive means of continuous or discrete chemical monitoring.[93–100]

7.2 Current Methods of Microneedle Array Fabrication

There are many different ways of fabricating MNAs, with each method having their own advantages and disadvantages. Fabrication methods include typical microfabrication techniques such as deep reactive ion etching (DRIE),[84,101,102] wet etching,[85,103] photolithography,[83] and micromachining.[86] Other novel methods include injection moulding using an electric discharge milled master,[80,96–98] using acupuncture needles to create PDMS moulds,[81,88] or using state-of-the-art rapid prototyping systems.[87] Table 7-1 provides a qualitative assessment of some of these fabrication methods used for producing microneedle arrays; for example, 'Acupuncture PDMS moulds' has been scored a 'LOW' for 'Design flexibility' as the shape/profile of the microneedles is limited to that of the acupuncture needles.

Table 7-1: Qualitative assessment of various fabrication methods used for producing microneedle arrays.

Method	Skill level	Speed/ scalability	Design flexibility	Cost/ capital
DRIE	HIGH	LOW	HIGH	HIGH
Wet-etching	HIGH	LOW	HIGH	HIGH
Photolithography	HIGH	LOW	HIGH	HIGH
Injection moulding	LOW	HIGH	HIGH*	HIGH
Acupuncture PDMS moulds	LOW	HIGH	LOW	LOW
3D printing	LOW	LOW	HIGH	HIGH

**While the fabrication method of injection moulding is flexible to using different masters, the cost of producing new masters can be very expensive.*

These methods can all produce transdermal devices on the micro-scale, however, they all require some combination of capital for specialist machinery, a skilled operative, time (slow to fabricate), or a fixed, finalised design. These specialised fabrication methods are not always available to researchers new to the field of transdermal devices/MNAs and therefore limit the potential for new investigations, experiments, and preliminary studies. A new method of MNA fabrication is needed that does not require specialist machinery or skilled operatives, is reasonably fast and cost effective, and is flexible to MNA design changes, allowing for design experimentation and optimisation.

7.3 Iterative Casting-Shrinkage Cycles

The concept of the 'casting-shrinkage' cycles uses a combination of polyurethane (PU) polymer and water to form a cast of a mould. The cast will then shrink to a percentage of its original size as solvents evaporate away from it.

Using an iterative shrinking method, originally described by Sun *et al.,*[49] it is possible to shrink 3D printed templates to the micro-scale over several iterations, vastly improving the resolution of the original mould. Sun *et al.* describe the use of a polyurethane (PU) polymer (Hydrospan 400) that will gel and solidify when mixed with water within 1-2 minutes, forming a white solid (see Figure 7-5(A)). As the solvents from the solid evaporate away, the solid form shrinks isometrically to a percentage of its original size (shrinkage factor will depend on the PU-water ratio of the mixture); the resulting solid will be yellow in appearance (see Figure 7-5(B)).

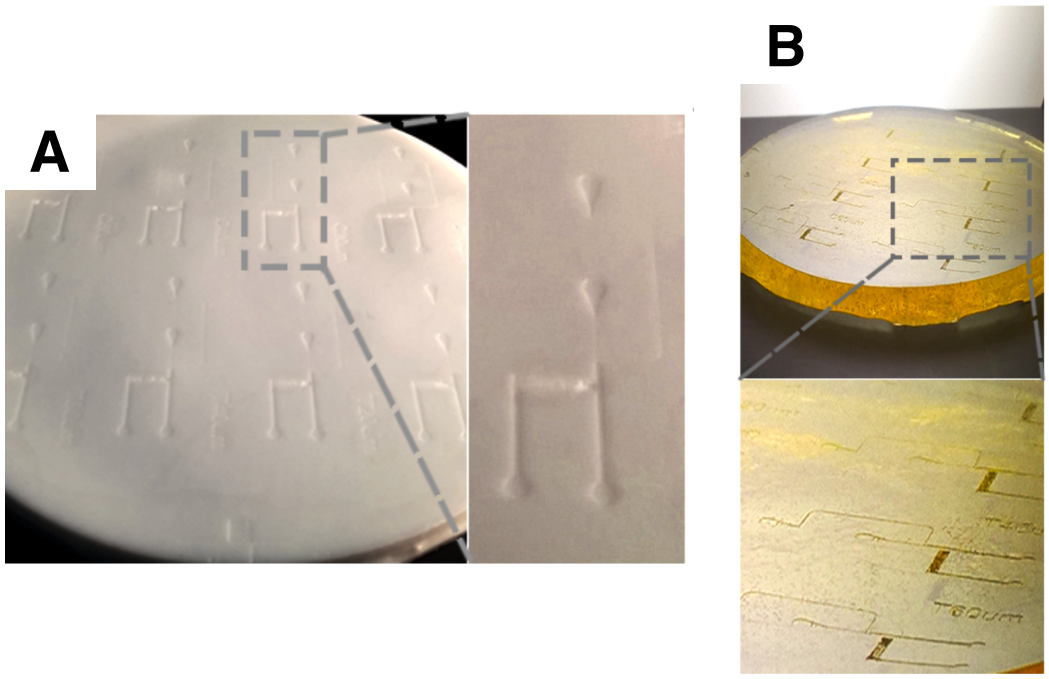


Figure 7-5: Casting-shrinkage method used to create microfluidic chips. *(A) White solid cast from solid PU-water mixture. (B) After the solvents have evaporated, the resulting solid will be yellow in appearance (Adapted with permission from ref* 49*. Copyright 2017 American Chemical Society).*

Sun *et al.* have proved the effectiveness of this 'casting-shrinkage' method by producing microfluidic chips from 3D printed moulds. If applied to fabrication of MNAs, this 'casting-shrinkage' method could provide a new route for early researchers who wish to investigate the use of MNAs and transdermal devices without the need for expensive equipment or skilled personnel.

7.3.1 Method

The process I describe here, I have adapted from the method developed by Sun *et al.*[49] The process begins with a 3D printed master of the transdermal design; the master was designed using AutoCAD and printed out of acrylic (polymethyl methacrylate) using a Stratasys Objet30 Pro (resolution: 600 dpi x 600 dpi x 900 dpi) in a glossy finish. A PDMS mould of the 3D printed master was taken, using the technique for PDMS mould creation described in Section 2.3.1. In initial work the 3D printed master was designed as a 7x7 array of microneedles printed with a 2000 μm microneedle height and 1200 μm base width (see Figure 7-6). In later experiments, I investigated several different designs for the microneedle geometry and layout of the MNAs were experimented with.

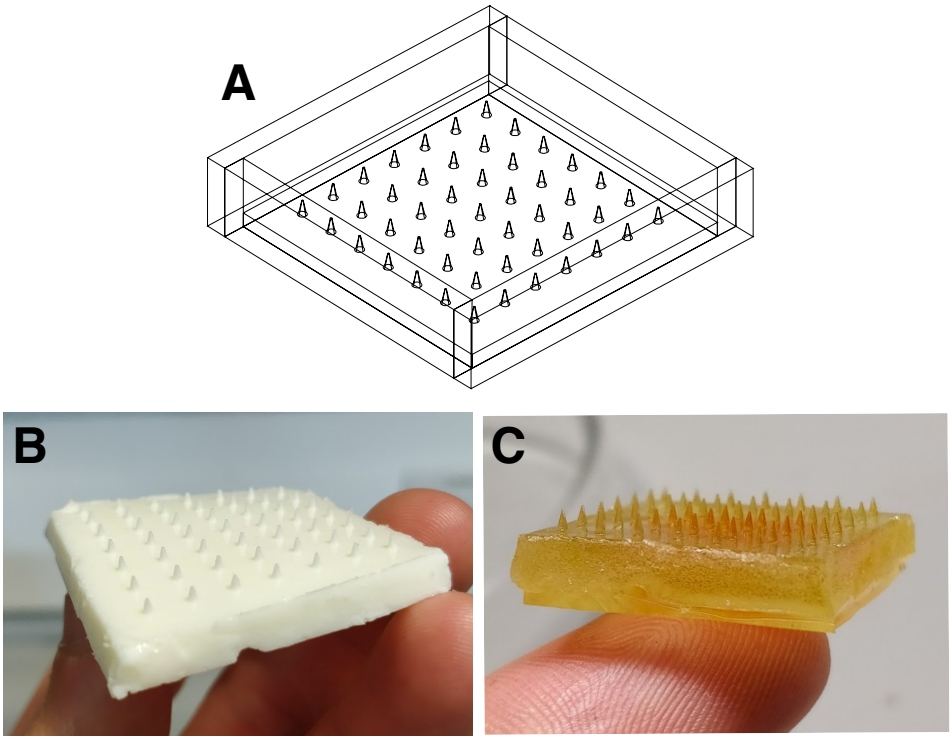


Figure 7-6: 3D printed master used for initial experiments into producing MNAs using the casting-shrinkage method *(A) Design of MNA master to be 3D printed (B) White PU-water solid replica of 3D printed master (C) Shrunken MNA after solvents have evaporated off.*

Following the process detailed by Sun *et al.,* a PU-water ratio of 1:3 was used in the casting-shrinkage cycles; I have briefly explored the use of different PU-water ratios, but have had unsatisfactory results. A higher ratio of water (e.g. PU-water ratios of 1:4 and higher) has the benefit of a greater shrinkage percentage per cycle, meaning that less cycles are required to achieve the desired spike height and there is less chance of compounded errors occurring in the final MNA cast. However, the higher the water content of the solid PU cast, the more brittle the resulting solid is, therefore making it more difficult to remove from the mould without the fine microstructures becoming damaged or breaking. Mixtures with a lower proportion of water (PU-water ratios of 1:2 and 1:1) maintain their structure as a solid well and are easier to remove from the mould but have a lower shrinkage percentage per cycle. The more cycles required to achieve the desired final spike height, the more risk there is of errors occurring and being compounded into the final cast.

After the mixture has been poured into the cast, it needs to be degassed to remove any air bubbles, otherwise the mixture will not fill the entirety of the mould and any bubbles sticking to the surface of the mould will result in distortions in the PU cast, as seen in Figure 7-7.

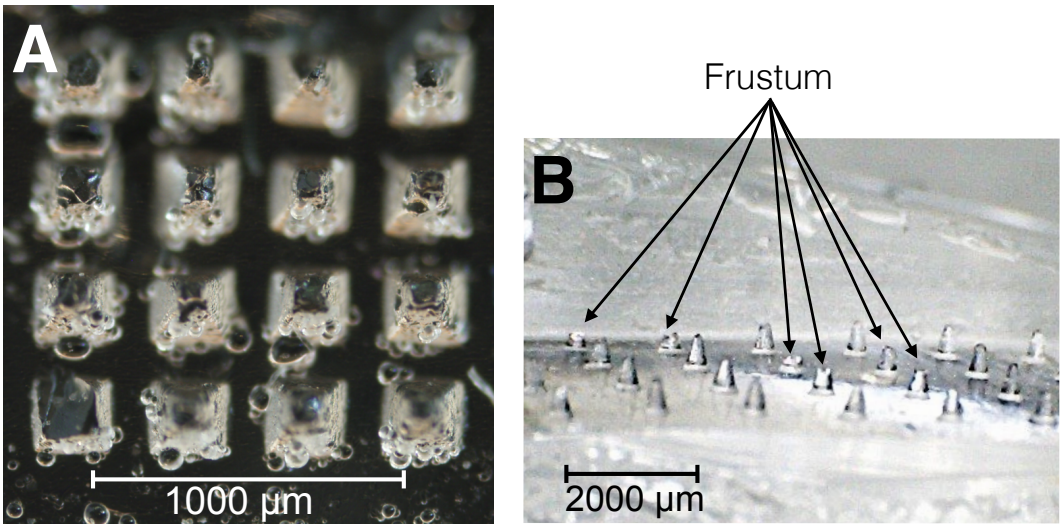


Figure 7-7: Shrunken PDMS replicas distorted by air bubbles. *(A) Air bubbles remaining in the mould have clung to the structure during the casting process, resulting in the distortions seen on the square-based pyramids. (B) Air bubbles have caused the PU mixture to not fill the mould properly, resulting in many incomplete spikes and a frustum type structure (incomplete cone) instead of the desired conical shape.*

Pouring the PU mixture into the mould and degassing it in a vacuum chamber to remove distorting air bubbles was found not to be a viable solution. When the PU and water are mixed together they form a white liquid that rapidly gels and solidifies. The gel time of the PU-mixture is so short that it solidifies during the degassing process, resulting in an extremely distorted cast; the PU mixture bubbles/foams while air is being drawn out and solidifies in this state, resulting in an unusable cast. Cold water (refrigerated at 3°C) is used to delay the setting time of the mixture, however even with cold water the mixture will still gel and solidify within 1-2 minutes of being combined. As such, there is not enough time to remove all the gas from the mould and mixture before it solidifies using a standard vacuum chamber. To ensure the PU mixture completely fills the mould, removing all gas and bubbles, I have designed and built a custom vacuum chamber (see Figure 7-8) based on the one used by Sun *et al*. in their original

paper. The mould was inserted into the vacuum chamber and sealed.[vi] Air is pumped out of the airtight chamber via the outlet tubing using a vacuum pump and sealed again using a clamp. The PU and water are mixed, and poured into the inlet container. When the clamp on the inlet tubing is opened, the negative pressure of the evacuated chamber pulls the PU-mixture into the chamber, filling the mould.

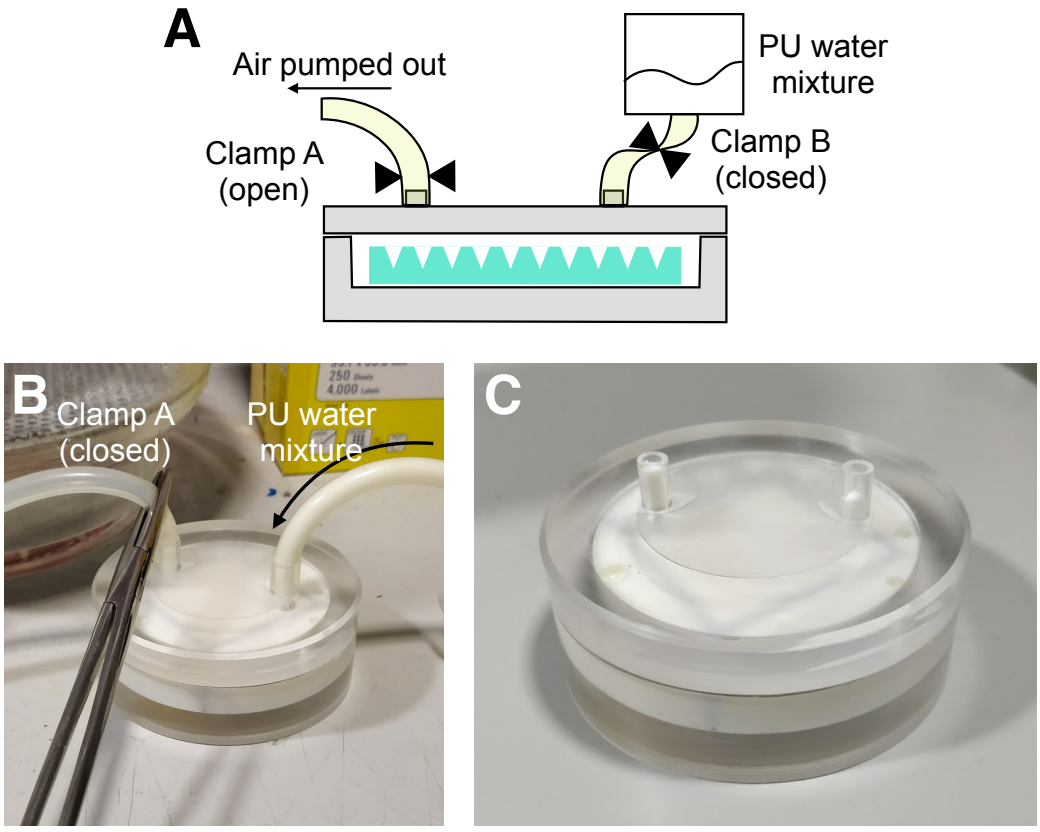


Figure 7-8: Custom vacuum chamber for degassing PU-water mixture and mould. *(A) Schematic of custom vacuum chamber: air is pumped out of the chamber containing the mould with clamp B closed. Following this, clamp A is closed and B is opened so the PU-water mixture can enter the chamber and fill the mould. (B) Photograph showing the vacuum chamber process; PU-water mixture has filled the mould in the chamber. (C) When the PU-water mixture has completely filled the chamber and has set, both clamps and tubing are removed so that the PU cast can be removed from the chamber.*

[vi] *The high tolerances used on the contact surfaces of the vacuum chamber and lid create a seal when closed. Tape is used to cover the join of the vacuum chamber to ensure an air-tight seal.*

Once set, the PU cast and mould were removed from the chamber. Taking great care not to damage the integrity of the new PU structures, the cast was released from the PDMS mould; this is possible as the white PU cast is a flexible elastomer. The white PU cast is then placed in the oven for 6 hours at 60°C to speed up the shrinking process; the process can be done at room temperature, but can take up to 14 days.[49] Paper towelling was placed underneath the fresh PU cast to prevent it from sticking to any surfaces that may cause it to distort during the shrinking process. When all the solvent has evaporated from the PU cast, it will have isometrically shrunk into a clear, yellow solid approximately 50% of its original dimensions (consistent with the 1:3 PU to water ratio used), as shown in Figure 1-7. The shrunken, yellow PU solid is a soft, pliable, elastic material and should be handled with care to ensure the integrity of the shrunken microstructures is retained.

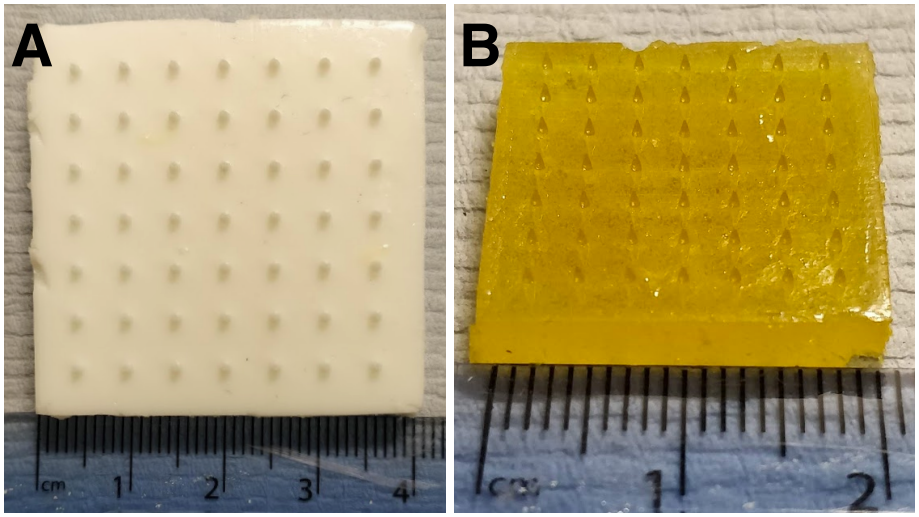


Figure 7-9: PU cast before and after shrinking process *(A) Fresh white PU cast of an MNA consisting of 7x7 conical spikes after it has been removed from the custom vacuum chamber and released from its PDMS mould. Excess material has been trimmed off, and the array measures approximately 40 mm in width. (B) The white PU cast is placed in the oven at 60°C for 6 hours to speed up the shrinking process. When all solvents have evaporated from the PU cast it shrinks isometrically into to a clear yellow replica measuring approximately 20 mm in width (~50% of the original mould's dimensions).*

The shrunken, yellow PU cast can then be used to create a new PDMS mould to initiate the next cycle of casting and shrinking. This completes the cast-shrinkage cycle, which can be repeated as many times as necessary, iteratively shrinking

the geometries of the master template with each turn. At this point in the cycle, the new PDMS mould can also be used to create a rigid cast for the microneedle array (MNA), ending the iterative process. A polyester casting resin (rigid when cured) is used to form a rigid cast of the shrunken spike array (see Section 2.3.2).

In this thesis, the casting-shrinkage method has been tested to a maximum of three cycles. A schematic of the casting-shrinkage cycle is detailed in Figure 7-10, following the format:

1. *3D print master design and produce PDMS mould.*
2. *Prepare PDMS mould for casting-shrinkage cycles.*
3. *Mix PU and water and fill mould with mixture (using vacuum chamber).*
4. *Remove solid PU cast from mould. Place in oven at 60°C for 6 hours.*
5. *Create new PDMS mould from shrunken PU cast. (Return to step 2 for next cycle).*
6. *Cast rigid form of shrunken MNA for use as a transdermal device.*

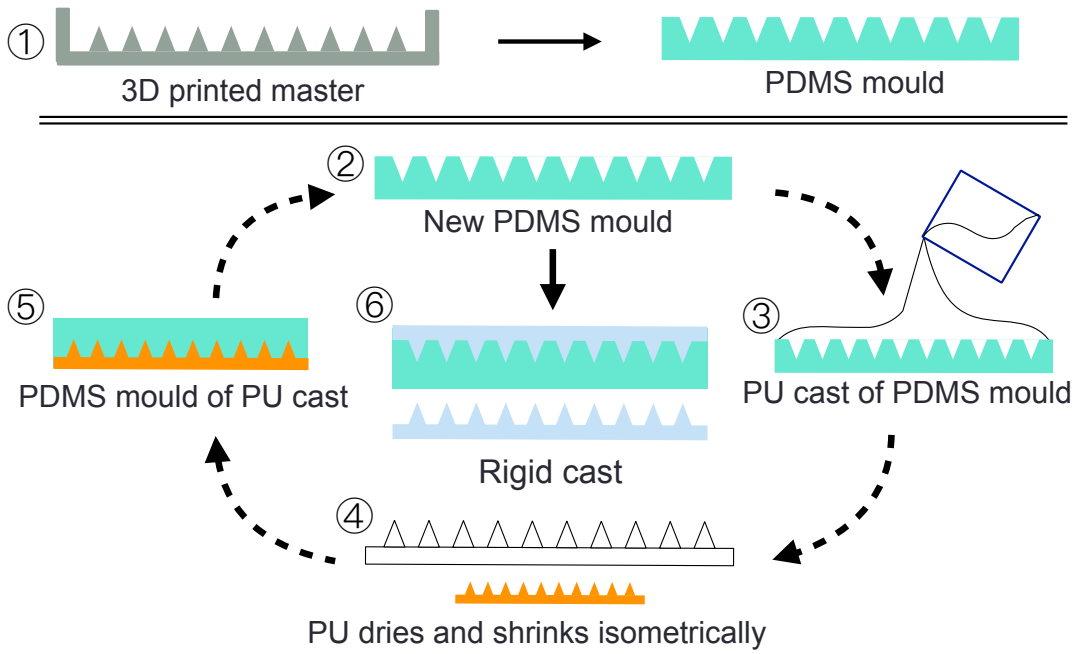


Figure 7-10: Casting-shrinkage cycle schematic *(1) PDMS mould is made from casting the 3D-printed master. (2) PDMS mould forms the starting point of casting-shrinkage cycles. (3) PU cast is taken of the PDMS mould using the custom vacuum chamber. (4) PU cast is put in the oven for 6 hours at 60°C where it shrinks isometrically. (5) A PDMS mould is taken of the shrunken PU cast, which then forms the starting point of the next iterative cycle. (6) Alternatively, a rigid cast of the PDMS mould can be taken using a polyester casting resin. Note, the PDMS mould can still be used for further cycles if used for a rigid cast.*

7.4 Results and Discussion

7.4.1 Microneedle characterisation

This method provides an inexpensive, rapid route of producing MNAs and transdermal devices, without the need for specialist training or expensive equipment. Using a mixing ratio of 1:3 PU-water, the cast will shrink isometrically to approximately 50% of its original size (see Figure 7-9). Using microscopic images (see Figure 7-11) and a laser-profilometer (see Figure 7-12), the microneedle dimensions can be accurately measured.

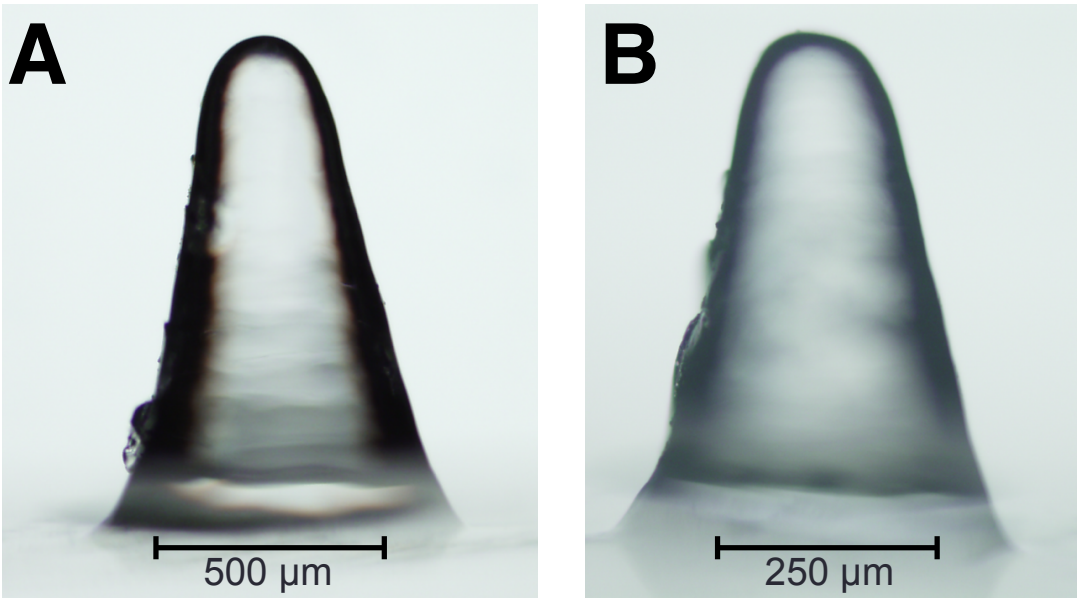


Figure 7-11: PDMS casts are taken of the shrunken casts. *PDMS casts were cut and prepared on slides for microscope imaging and measurement. Spike height and base width are measured using ImageJ. (A) First casting-shrinkage iteration, average spike height (7x7 array, n=49): 1068.7 µm ± 14.3 µm (53.4% of original). (B) Second casting-shrinkage iteration, average spike height (7x7 array, n=49): 574.6 µm ± 18.4 µm (53.7% of previous iteration and 28.7% of original).*

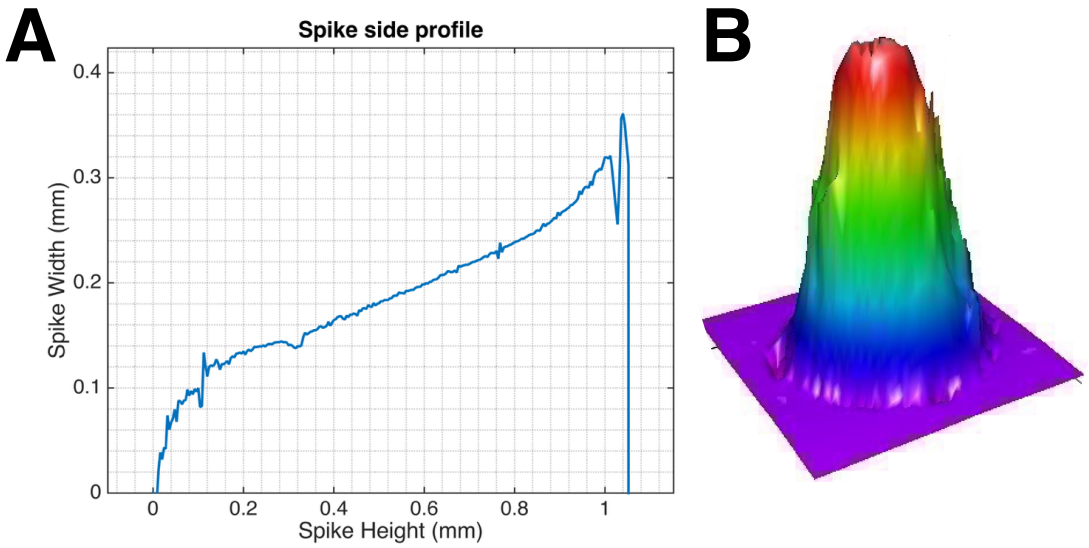


Figure 7-12: The shrunken microneedles after one cycle, measured using a laser profilometer. *(A) The MNA was turned on its side and the spike profile was scanned. The plot shows the profile of the microneedle, its half-base width measuring 0.36 mm (0.72 mm full width) and its height measuring 1.05 mm. (B) 3D image created of the microneedle using profiles scanned using the laser profilometer.*

PDMS casts of the MNAs were taken, cut, and prepared on slides so that microscopic images could be taken and measured against a scale bar. The MNAs were made from a 3D printed master of 49 conical spikes (arranged in a 7x7 grid), which was then shrunken twice to give moulds approximately 50% and 25% of the original size. The measurements were averaged over the entire array (n = 49) and the results are presented in Table 7-2. The average spike height after the first shrinkage cycle was 1068.7 ± 14.3 µm (53.4% of original) and after the second cycle was 574.6 ± 14.2 µm (53.7% of previous iteration, 28.7% of original). The shrinkage percentage of the microneedle base width was slightly higher than its height: 62.8% after first cycle, and 61.5% after the second cycle. Although there is a slight difference in shrinkage percentage between height and base width, the percentage of shrinkage between for the first and second cycles are consistent. This leads to the thought that perhaps the PU cast does not shrink as isometrically as previously believed. One possible explanation is that the interface where the microneedle structure attaches to the base of the array becomes distorted or stretched during the shrinking process; the base of the array is very large in comparison to the microneedles and therefore could shrink

at a different rate. This would lead to distortions, appearing as a wider microneedle base than expected. This would also explain why the profile of the microneedle is slightly bell-shaped, rather than conical, as seen in the plot in Figure 7-12.

Table 7-2: Measurements of microneedle height and base width after 1st and 2nd cycles of the cast-shrinking process. *Measurements for both cycles are averages over a 7x7 array (n = 49).*

	Original (µm)	1st cycle (µm)	Shrinkage (%)	σ (µm)	2nd cycle (µm)	Shrinkage (%)	σ (µm)
Height	2000	1068.7	53.4	14.3	574.6	53.7	14.2
Base width	1200	753.8	62.8	18.4	463.6	61.5	23.9

The iterative cycle was repeated a third time, producing a cast that was approximately 12.5% of the original 3D printed master (see Figure 7-13). Scanning electron microscope (SEM) images were taken of the casts following the third cycle to provide finer detail of the microneedle size and form (see Figure 7-14).

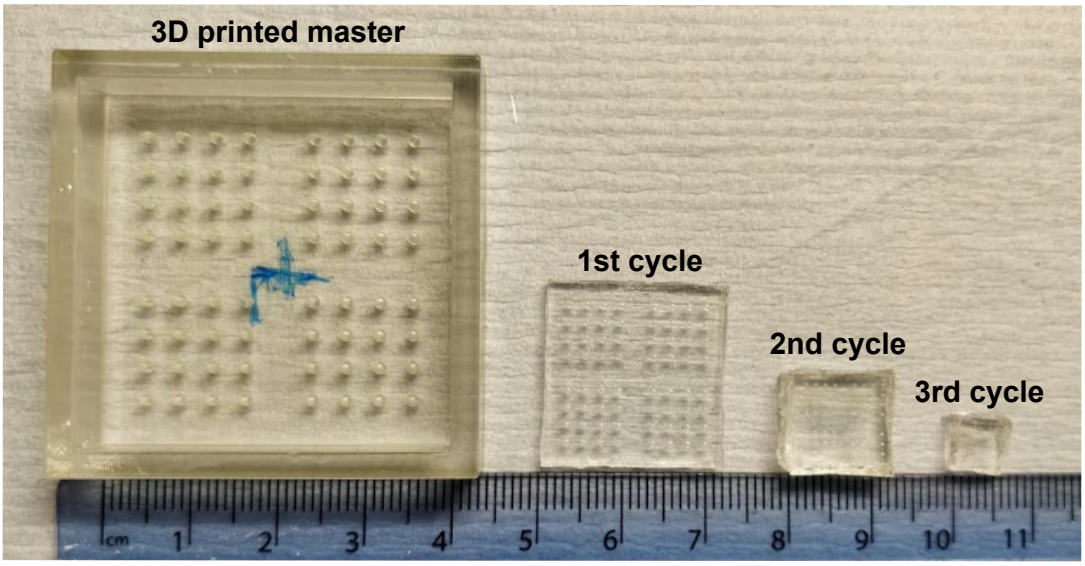


Figure 7-13: First, second, and third cycle casts of 2x2 array of 4x4 conical microneedles. *Casts are made from clear, rigid, polyester casting resin. The third cycle cast has been shrunk down to approximately 12.5% of the original 3D printed master's dimensions.*

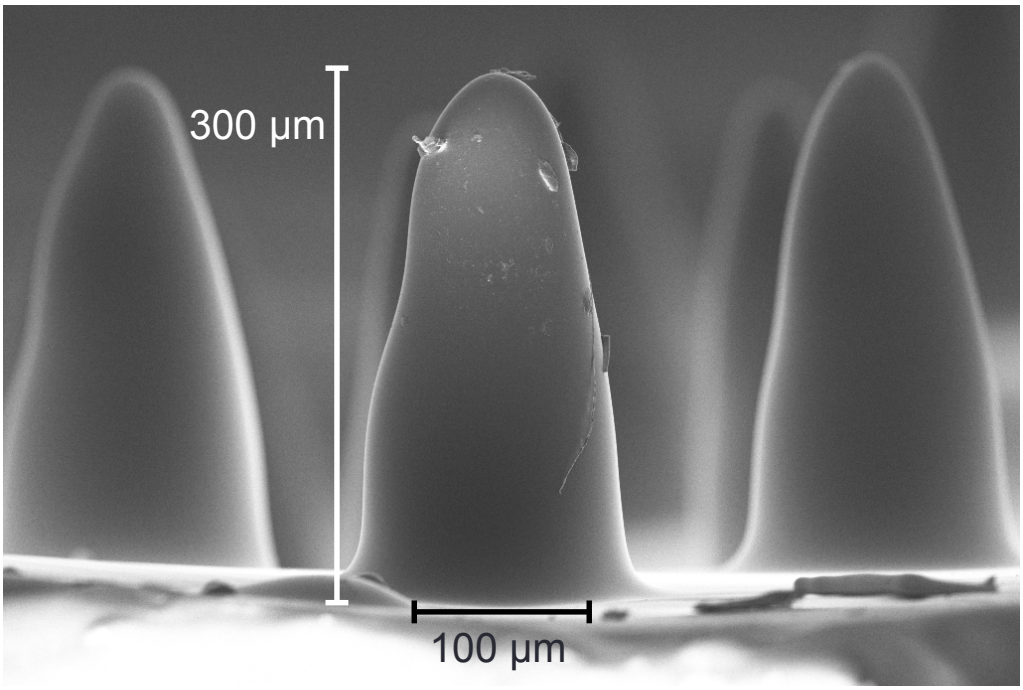


Figure 7-14: SEM image of third cycle cast of conical microneedles. *The example microneedle imaged here measures 297 µm high, 51.7% of the average microneedle height in the previous cycle and 14.9% of the original 3D printed master.*

The SEM image in Figure 7-14 demonstrates the ability of casting-shrinkage cycles to rapidly and inexpensively produce a prototype MNA at a desirable scale for transdermal monitoring. In just three iterations, the microneedle height has been shrunk from 2000 µm to 297 µm, approximately 15% of its original size. As shown in the literature,[80,104] not all of the measured spike height will penetrate into the skin when inserted and so 297 µm is believed to be an acceptable spike height for a MNA, as the microneedles will penetrate though the *stratum corneum* but not so deep that they draw blood or cause pain. It is possible to produce geometries smaller than this either by starting with a smaller 3D printed master (possibly requiring a higher-specification 3D printer) or by continuing the iterative process. As shown by Sun *et al.*, this technique can be used to produce geometries of impressive resolution (4 µm).[49] Starting with a smaller 3D printed master would also reduce the number of cycles required to achieve the scale shown in Figure 7-14. Other fabrication methods such as photolithography or state-of-the-art rapid prototyping systems may be able to produce MNAs with higher precision and resolution, however using casting-shrinkage cycles means one can quickly and inexpensively go from concept to

prototype in just a few iterations. This is ideal for researchers in the early stages of investigating transdermal devices, giving them the flexibility to try different designs without committing to expensive equipment, machinery, or specialist training.

To explore the flexibility of this process, I have experimented with many different microneedle layouts and microneedle geometries. For example, Figure 7-15(A) shows the second iteration of a 2x2 array of 4x4 square-based pyramid MNAs, similar to that used by Sharma *et al.,*[80,96] and Figure 7-15(B) shows the third iteration of a 13x13 array of conical microneedles, fitting 169 conical microneedles onto an approximately 25 mm^2 pad. Figure 7-15(C) shows the second iteration of a 7x7 array of 12-point-star-based conical microneedles. These examples demonstrate how fine structures and details are still preserved in successive iterations when using the casting-shrinkage method.

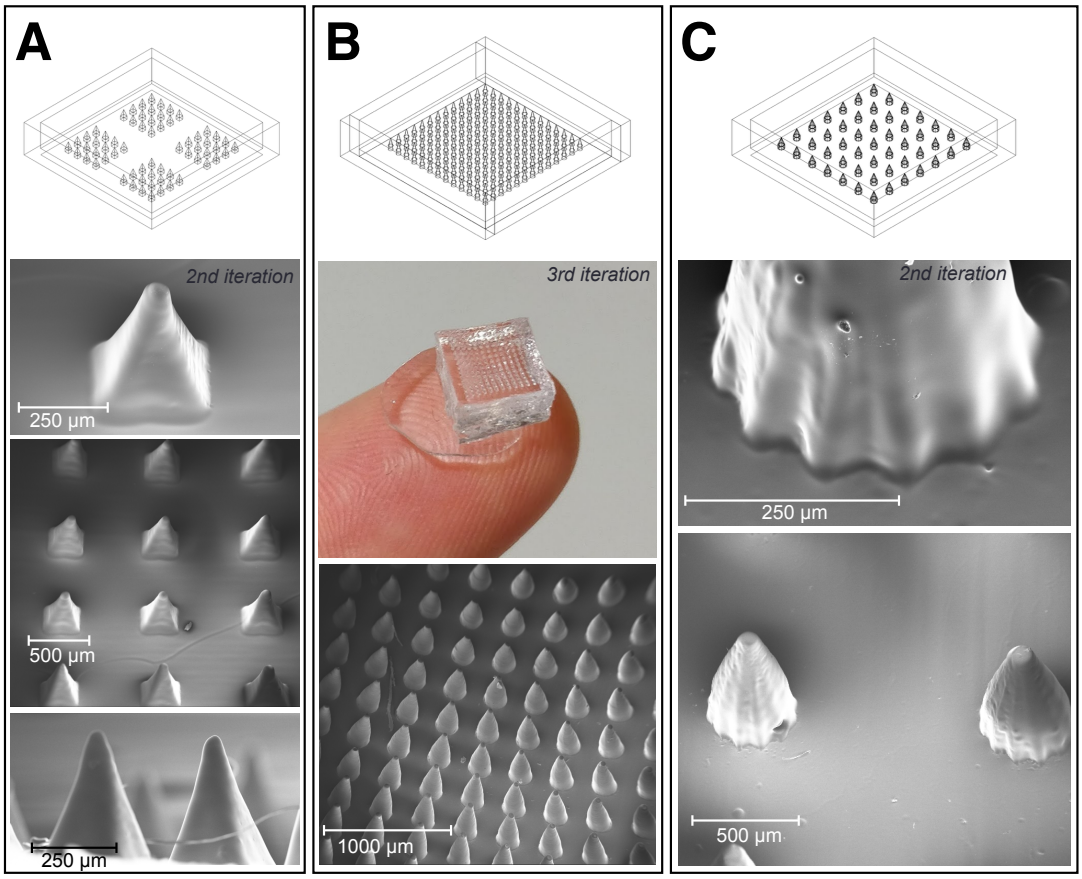


Figure 7-15: SEM images of different microneedle geometry and MNA layouts. *(A) Square-based pyramid design in 2x2 array of 4x4 MNA layout at 2nd iteration step of casting-shrinkage process, (B) 13x13 conical MNA at 3rd iteration step of casting-shrinkage process, (C) 7x7 array of 12-pointed-star-based conical MNA at 2nd step of casting-shrinkage process.*

7.4.2 Skin penetration tests

The microneedles need to be tested for skin penetration; the ideal transdermal EMG sensor, as mentioned previously in Section 7.1, should penetrate through the *stratum corneum* into the conductive epidermis layers in order to minimise the electrode's contact impedance, while remaining painless and not drawing blood. Skin penetration tests were performed on ethically obtained human skin biopsies.[vii] The MNAs tested were cast in a rigid polyester resin using the second- and third-cycle PDMS moulds of the cast-shrinking method. The microneedles of the 3D printed master were of conical design, 2000 μm height, 1200 μm base width, and 5000 μm spacing between the microneedles.

The frozen skin biopsies were allowed to thaw before the penetration tests began; it should be noted that the mechanical properties of the thawed skin samples will differ to fresh skin biopsies. The MNAs were then dipped into a dye (haemotoxylin) and pressed into the skin biopsies. The biopsies were inspected visually for puncture marks left behind by the dye to ensure the MNAs had penetrated the skin. The skin biopsies were suspended in optimal-cutting-temperature compound (OCT) and frozen using dry ice. The frozen samples were placed in a cryostat (a device to maintain the sample's low temperature), and sliced into 20 μm thick cross-sections using a microtome. The cross-sections were then transferred on to microscope slides for visualisation using a microscope. The method of producing the skin biopsy cross-sections is shown in Figure 7-16.

[vii] *Thanks to Shared Patel and Dr. Claire Higgins for their assistance in skin penetration experiments.*

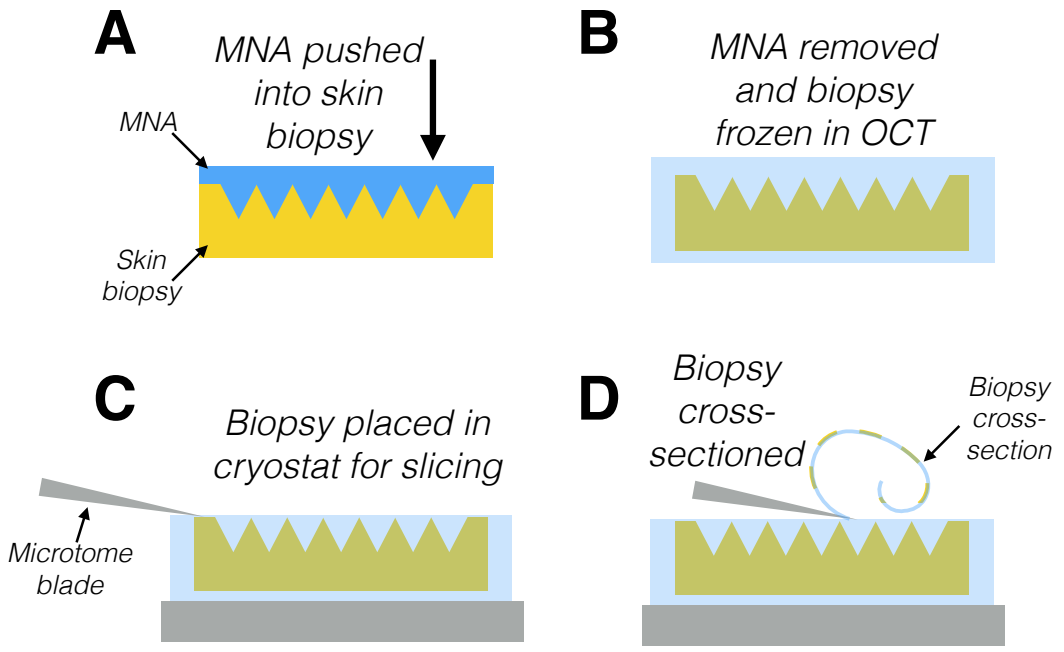


Figure 7-16: Skin biopsy cross-section method *(A) MNA is pushed into skin biopsy. (B) MNA is removed and the punctured skin biopsy is frozen in optimal cutting temperature (OCT) compound. (C) The frozen biopsy is placed into the cryostat to maintain its temperature. (D) Using a microtome, 20 µm thick cross-sections are produced. Cross-sections are placed on microscope slides for viewing.*

One difficulty with this method is that the MNAs have to be removed from the skin sample before it can be frozen and sliced, as the cryostat blade would not be able to cut through the MNAs. Due to the natural elasticity of skin it is very likely that the impressions/punctures caused by the microneedles will reduce in size after they are removed, making it difficult to determine what the puncture depth of the microneedle really is. For the second-cycle MNA, the puncture sites circled in Figure 7-17 show a maximum puncture depth of just 255 µm and are spaced 857 µm and 837 µm apart. This is less than expected; assuming 53.5% shrinkage per cycle, the second-cycle microneedles used in Figure 7-17 should be approximately 570 µm in height and have a spacing of 1430 µm. The *stratum corneum* can be seen in Figure 7-17 as the outer black line, measuring approximately 28 µm thick. The rest of the epidermal layers can be seen as a whitish-yellow band just below the *stratum corneum* that varies in thickness across the skin cross-section. Figure 7-18 shows an enhanced image of the central puncture site shown in Figure 7-17 at a higher magnification, detailing the effect the microneedle has had on the layers of the epidermis.

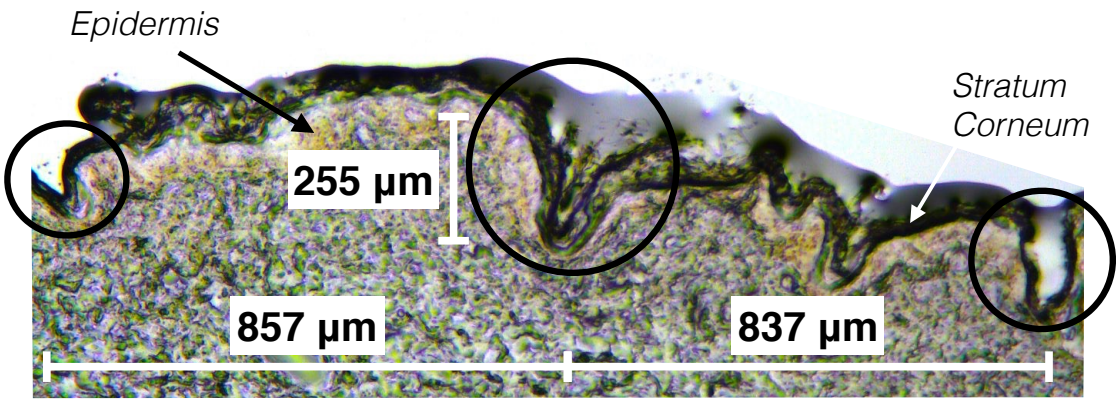


Figure 7-17: 20 µm skin slice taken from human cadaver skin biopsy penetrated by microneedles measuring approximately 572 µm in height (5X magnification). *Puncture sites from microneedles are circled in black. The stratum corneum can be seen as the outer black line of the skin cross-section, measuring approximately 28 µm thick. The rest of the epidermis can be seen as the yellowish band below the stratum corneum.*

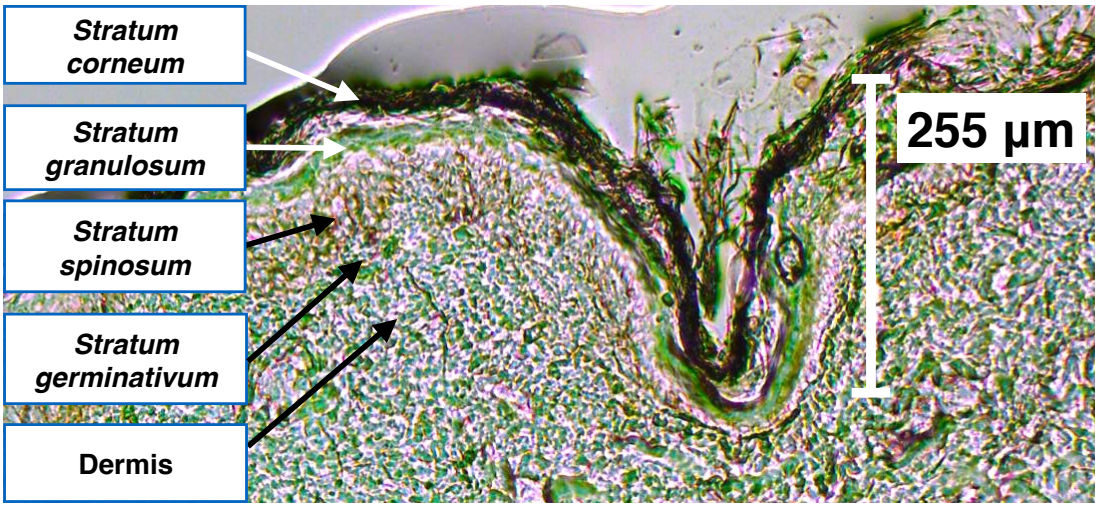


Figure 7-18: 20 µm skin slice taken from human cadaver skin biopsy penetrated by microneedles measuring approximately 572 µm in height (10X magnification). *Different layers of the epidermis are labelled; thickness of the epidermis varies across the cross-section. The epidermis has deformed around the microneedle site.*

This method has proved useful for detecting whether the microneedles penetrate the *stratum corneum*, however, it can be difficult to measure the exact depth the microneedle has penetrated the skin due to the skin elasticity. It is also difficult to tell exactly how far from the centre of the microneedle the cross-section was taken, thus making it problematic to determine the point at which the microneedle is at its maximum depth (see Figure 7-19). Figure 7-20

demonstrates how the depth of the puncture site can appear to change between each skin slice, and how some skin slices can reveal where the *stratum corneum* has been pierced by the microneedle.

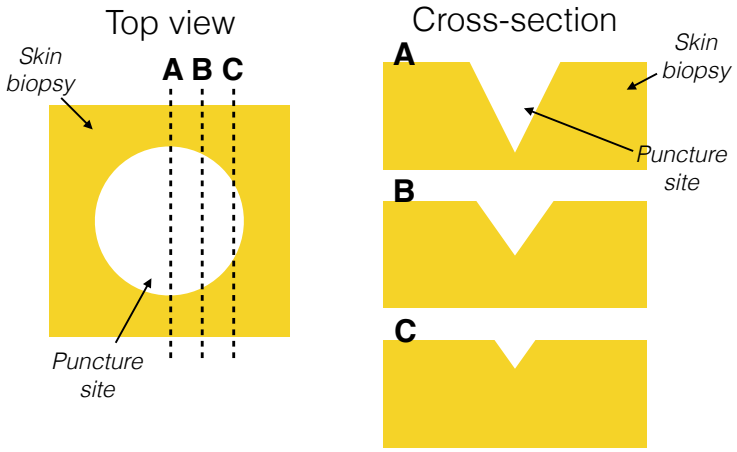


Figure 7-19: Schematic of puncture depth variation between skin biopsy cross-sections. *Cross-section A will show the puncture site at its maximum depth, whereas cross-section C will show a much smaller depth.*

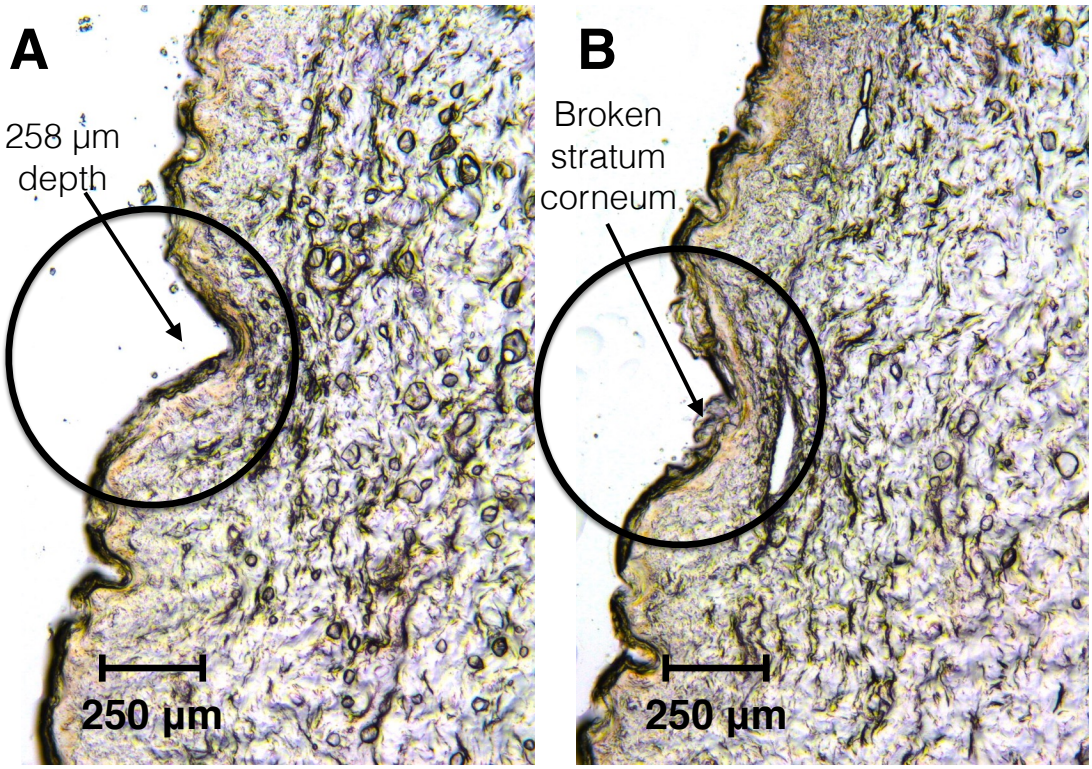


Figure 7-20: 20 µm skin slice taken from human cadaver skin biopsy penetrated by microneedles measuring approximately 572 µm in height showing broken *stratum corneum* (5X magnification). *(A) Microneedle puncture site measuring 258 µm in depth. (B) Same microneedle puncture site at different slice depth showing pierced stratum corneum, measuring 221 µm in depth.*

Rigid polyester casts from the third-cycle of the cast-shrinkage method were also used for skin penetration tests. Assuming 53.5% shrinkage per cycle, the resulting microneedle height should be around 15% the size of the 3D printed master. The third-cycle casts have a microneedle height of approximately 300 µm, as shown previously in Figure 7-14. When using the second-cycle MNA, a maximum penetration depth of 258 µm was found for a 572 µm microneedle, approximately 45% of the microneedle height (see Table 7-3). For the third-cycle MNA, 45% of the microneedle height would equate to 135 µm. This is not dissimilar in depth to the natural deformations in the *stratum corneum* and epidermis (e.g. pores etc.), as shown in the skin slice from control biopsy in Figure 7-21(A). Therefore, it can be difficult to interpret conclusive positive results from the penetration tests using such small microneedles. However, the skin slice from the third cycle MNA penetration tests in Figure 7-21(B) shows three puncture sites measuring 149 µm, 119 µm, and 148 µm where the *stratum corneum* has been broken. On average the depths of these penetration sites are approximately 46% of the third-cycle microneedle height, 300 µm, as expected (see Table 7-3).

Table 7-3: Puncture depths of microneedles from the 2nd and 3rd cycles of the casting-shrinkage method. *Note, that the measured depth for the third cycle is the average of the three puncture sites (149 µm, 119 µm, and 148 µm).*

Cycle	Height (µm)	Measured depth (µm)	Depth percentage (%)
2nd	572	258	45
3rd	300	139	46

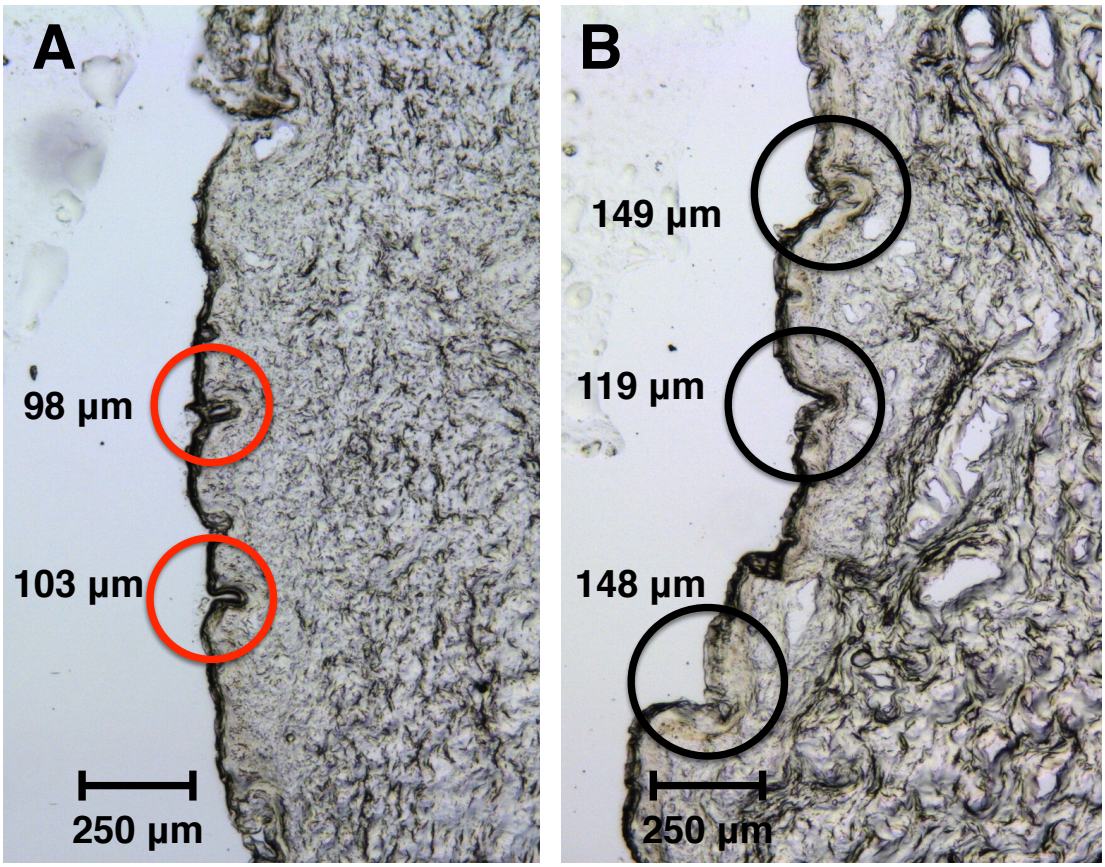


Figure 7-21: 20 µm skin slices taken from human cadaver skin biopsy - control test MNA penetration test from the third cycle of the cast-shrinking process (microneedles measuring approximately 300 µm in height). *(A) Control biopsy (no MNA penetration) skin slice showing the natural skin deformations that measure as deep at 103 µm. (B) Third cycle MNA penetration test skin slice, showing penetration sites measuring 149 µm, 119 µm, and 148 µm. The penetration sites show the deformation of the epidermis and breaking of the stratum corneum. Image taken using a microscope at 5X magnification.*

The skin penetration experiments have shown that the microneedles produced using the casting-shrinkage method can break/penetrate through the first, highly resistive, *stratum corneum* layer of skin. Penetrating through the *stratum corneum* into the more viable, conductive tissue underneath would immediately lower the impedance of the skin-electrode contact, to the benefit of a sensor used for electrophysiology measurement, such as EMG. Importantly, the scale of the microneedles produced has meant that the dermis has remained intact during the penetration testing, meaning the MNAs have not disrupted any blood vessels or nerve papilla that may cause a patient pain. The exact depth of the microneedles when inserted into the skin is difficult to determine; due to the

natural elasticity of skin, the depth of the puncture sites is likely to have shrunk when the MNA was removed to produce the skin cross-sections.

7.5 Gold Coating

For use *in vivo* the microneedles need to be conductive; the simplest way to do this was to coat the MNA with gold using a thermal evaporator. First, a 20 nm layer of chromium (Cr) was deposited to help with the adhesion of the gold. A 50 nm layer of gold (Au) was then deposited on top. A small hole was drilled in the corner of the microneedle array, through which a wire was threaded. The wire was conductively bonded to the surface of the Au using a two part silver epoxy. The silver epoxy bond was then insulated using araldite so that only the Au surface was exposed.

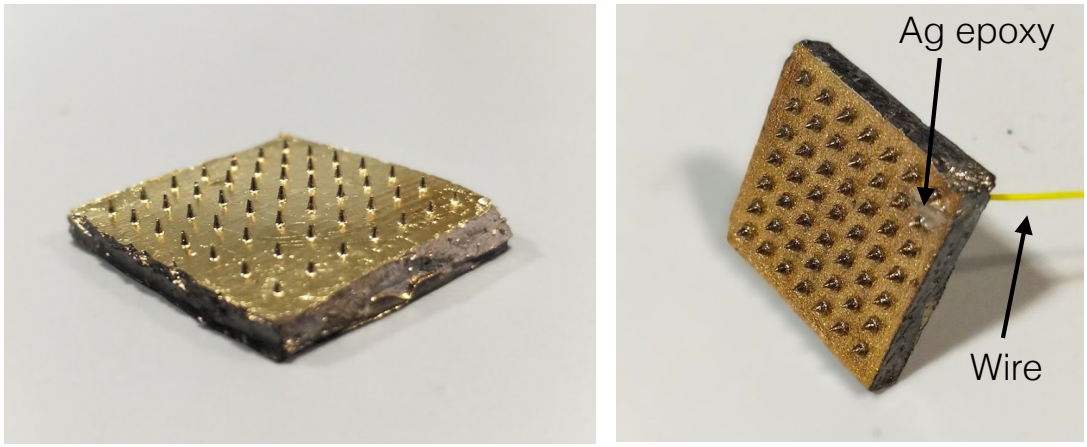


Figure 7-22: First cycle cast-shrinkage MNAs coated with chromium and gold. *MNAs were coated with 20 nm Cr (adhesion layer) and 50 nm Au using a thermal evaporator. A wire was then bonded to the Au surface using a conductive silver epoxy for connectivity.*

The Au-coated MNA was then cycled (in 1.5 mM ferrocene monocarboxylic acid) to check that all the connections were working and to assess the quality of the Au surface. The MNA was lowered into the solution using a micromanipulator (see Figure 7-23(A)) so that just the top surface of the MNA was in contact with the ferrocene (Fc) monocarboxylic acid solution, in case there was any additional Au sputtered on the sides of the MNA. A Ag|AgCl electrode was used as a reference electrode and a platinum mesh as a counter electrode. The oxidation peak of the cyclic voltammetry (CV) in Figure 7-23(B) is at the expected potential for Fc. The

current produced was large due to the large area of exposed Au on the MNA. The separation between the oxidation and reduction peaks (78 mV) is characteristic of a reversible couple at a clean Au surface.

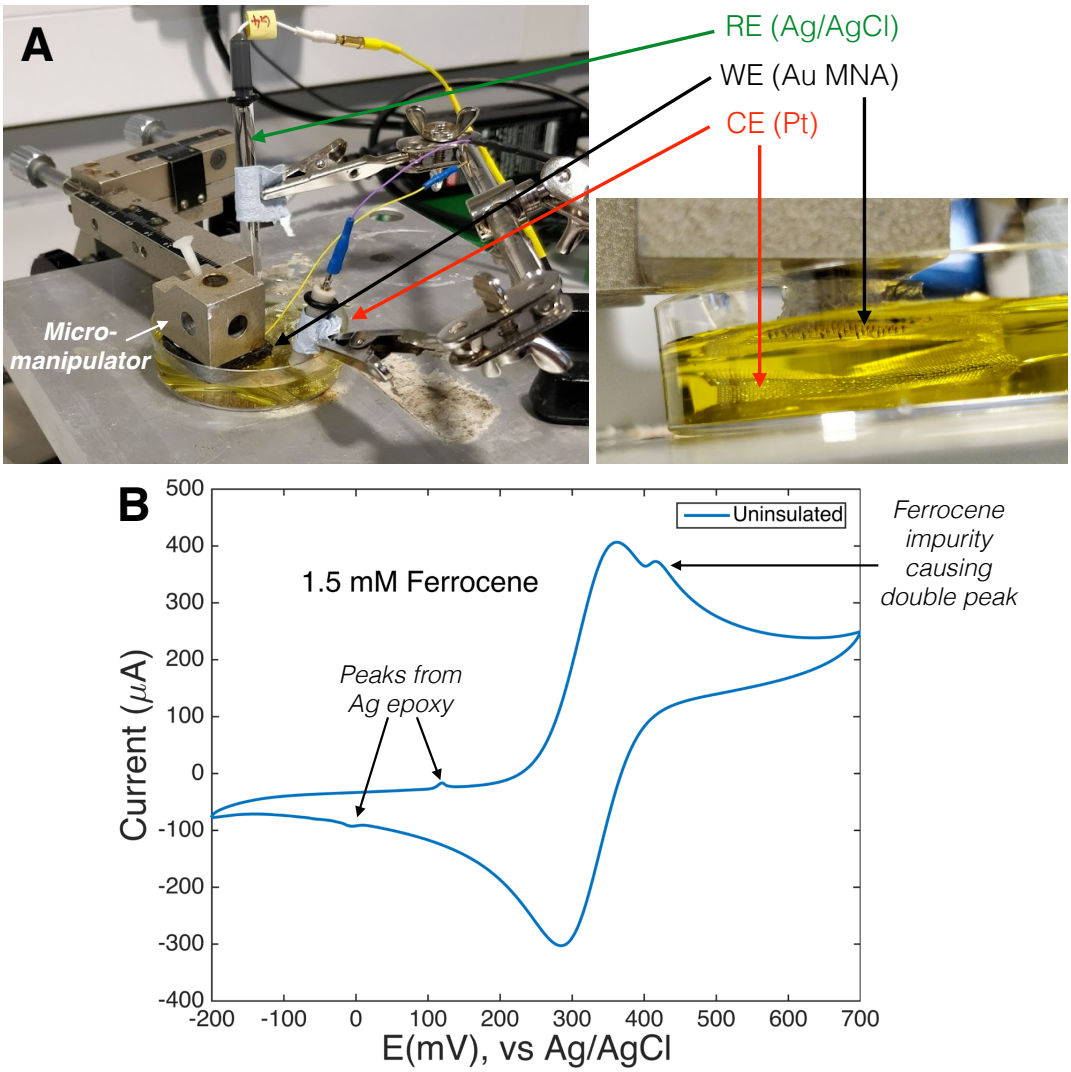


Figure 7-23: Cyclic Voltammetry (CV) setup and results of Au-coated MNA. *(A) CV setup up of Au-coated MNA working electrode (WE), silver|silver-chloride (Ag|AgCl) reference electrode (RE), and platinum mesh counter electrode (CE). A micro-manipulator was used to lower the Au WE into 1.5 mM ferrocene monocarboxylic acid so that just the top surface of the MNA was in contact with the solution. (B) CV of Au vs. Ag|AgCl in 1.5 mM ferrocene monocarboxylic acid (Ferrocene), 10 mV/s scan rate, (5 scans, 5th scan shown), demonstrating the large currents produced by the large surface area of the Au-coated MNA. As can be seen, the shape of the CV is predominantly that of a macroelectrode, but with some evidence of a microelectrode plateau.*

Despite the microneedles penetrating through the *stratum corneum*, there remains a large area of exposed Au between the microneedles that will be in

contact with the top layer of skin, causing sensor impedance and thus noise (see Figure 7-24(A)). To stop this effect, an insulating resin (RS components), used for electrically insulating contacts on electronic circuit boards, was deposited onto the base of the MNA using a pipette in order to insulate the base and hence reduce the impedance. As this insulating layer will take up some of the spike height of the MNA, a new design was made that incorporates a 'stilt' to raise the height of the spike above the insulating layer, shown in Figure 7-24(B). The cast-shrinkage method lends itself to quick re-design and production of new MNA configurations if problems, such as this, should arise.

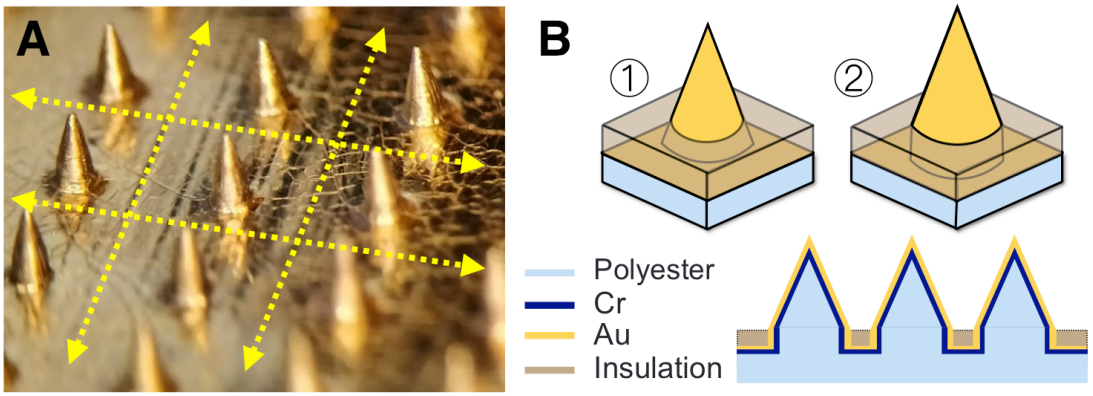


Figure 7-24: Insulating the base of the Au-coated MNA. *(A) Large uninsulated Au area between the microneedles will still cause a large contact impedance with stratum corneum, despite the microneedles penetrating through to the conductive layers of the epidermis. (B-1) A layer of insulating resin is deposited on to the base of the MNA to insulate it from the outer layer of skin. (B-2) A new MNA design that incorporates a 'stilt' to raise the height of the microneedle above the insulating resin layer.*

The insulated Au MNA was tested again by CV in 1.5 mM ferrocene monocarboxylic acid. The currents produced by CV for the insulated MNA were lower than before insulation, however, this was not proportional to the change in area of exposed Au. In fact, when normalised against the surface areas of exposed Au (see Table 7-4), the normalised oxidation peak of the insulated MNA was larger than that of the uninsulated MNA, as shown in Figure 7-25. This is due to the improved mass transport of an analyte to the curved surface of the microneedle, compared with that of a planar surface, combined with the hemispheric diffusion around the tip of the microneedle; despite the decrease in the surface area of exposed Au, the remaining exposed Au surface on the

insulated MNA provides better mass transport per mm^2 than the uninsulated MNA. The Mirkin group have shown that so long as a conical electrode is taller than its radius, it exhibits the steady-state current plateau found with microelectrodes:[105] using dimensions of 1st cycle microneedles, this is equivalent to a plateau current 2.2 times a microdisc of the same radius as the microneedle base, and approximately 80% of the current from a hemispheroid of the same dimensions.[105] We can then question whether these needles are small enough to behave as microelectrodes. In this scenario, the uninsulated Au MNA is behaving like a macroelectrode; the large exposed surface area of the Au MNA base dominates any mass transport benefit added by the conical planes of the microneedles. The oxidised ferrocene (Fc^+) stays localised and cannot diffuse away as it is limited by inefficient planar diffusion. The insulated MNA behaves like a microelectrode; the Fc^+ can diffuse away more easily due to the efficient hemispheric diffusion around the conical spikes. This enhanced Fc diffusion at the electrode surface explains the larger oxidation peak for the insulated MNA when normalised to surface area of exposed Au.

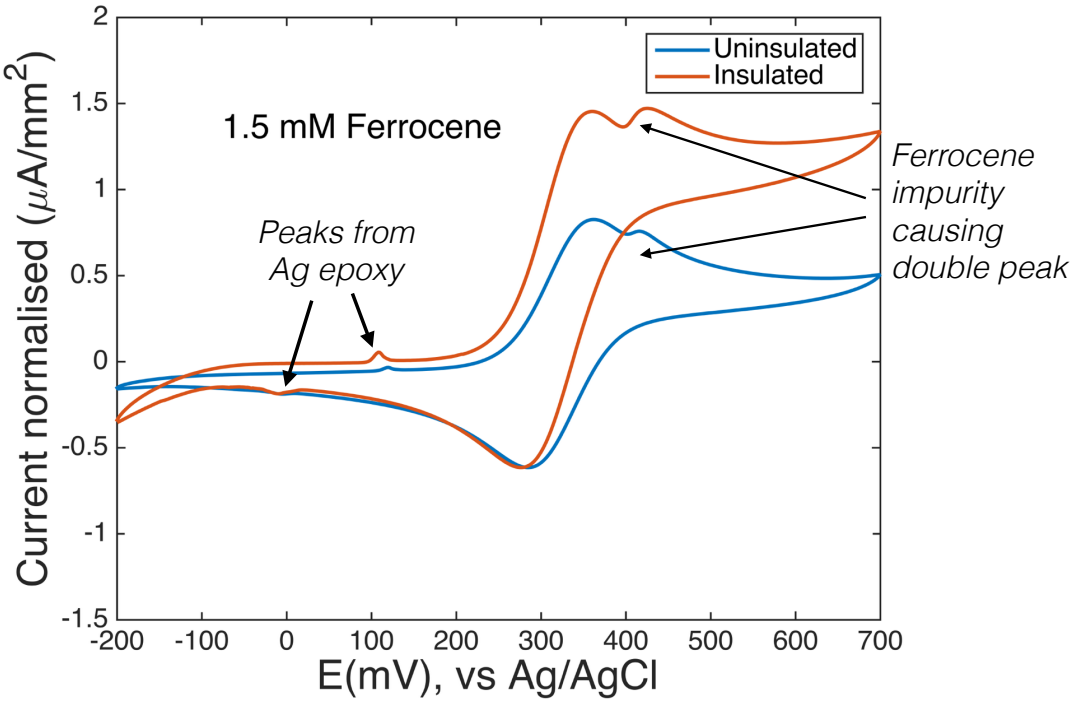


Figure 7-25: CVs of Au vs. Ag|AgCl in 1.5 mM ferrocene monocarboxylic acid for the insulated and uninsulated Au-coated MNAs. *The CVs are normalised using the approximate calculated surface area of the exposed Au on MNA (uninsulated = 492.2 mm^2, insulated = 55.2 mm^2). 10 mV/s scan rate (5 scans, 5th scan shown).*

Table 7-4: Approximate calculated surface area of exposed Au on MNA. *Exposed Au surface area before and after the deposition of insulating resin on the base of the first cycle MNA (7x7 grid of microneedles). Surface areas are calculated from the approximate dimensions of a first-cycle MNA.*

Surface Area (mm^2)	**Insulated MNA**	55.2
	Uninsulated MNA	492.2

7.6 Lactate Sensor Fabrication

To explore further possible uses of these electrodes, the MNA was used to fabricate a biosensor to detect different concentrations of lactate. To provide selectivity, the enzyme lactate oxidase (LOx) was used to catalyse lactate into hydrogen peroxide (H_2O_2) and pyruvate.

$$Lactate \xrightarrow{LOx} H_2O_2 + Pyruvate \quad \textbf{(7-10)}$$

The production of H_2O_2 is exploited as a means of detecting the concentration of lactate. The concentration of H_2O_2 is proportional to the concentration of lactate in the sample. The oxidation of H_2O_2 at the electrode surface produces a current by the generation of electrons:

$$H_2O_2 \rightarrow 2H^+ + 2e^- + O_2 \quad \textbf{(7-11)}$$

7.6.1 Platinum coating

The MNAs were originally coated with gold for conductivity, however, it is well known that gold is not the best surface for H_2O_2 oxidation at a usable rate at low potentials.[106] Oxidation of H_2O_2 on a platinum (Pt) surface occurs at a much faster rate than on gold for the same potential and displays much better sensitivity.[106] Rogers has reported the difference in sensitivity of H_2O_2 oxidation on platinum and gold electrodes (50 μm),[106] shown in the calibration plot in Figure 7-26.

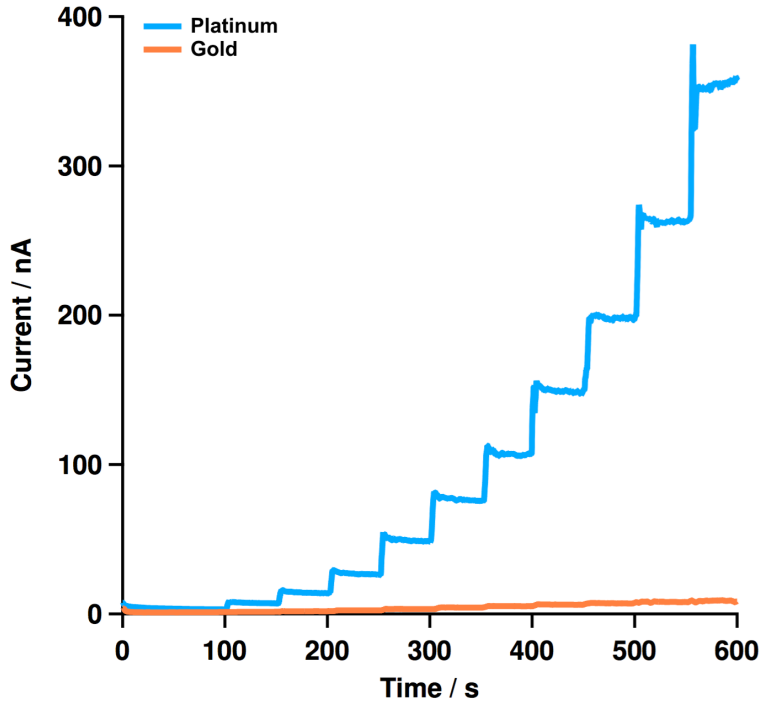


Figure 7-26: Hydrogen peroxide calibration on 50 µm Pt and Au electrodes. *H_2O_2 step changes every 50s of 0.5 mM, 1 mM, 2.5 mM, 5 mM, 7.5 mM, 10 mM, 15 mM, 20 mM, 30 mM, 50 mM (Taken from Rogers 2011, PhD thesis, Department of Bioengineering, Imperial College).*[106]

The reason for the difference in response is because the oxidation of H_2O_2 on Pt is a two-step reaction, seen in Equations (7-12) and (7-13). When the working electrode is held at +0.7 V, Pt at the electrode surface will react with water as seen in Equation (7-12) to form platinum dihydroxide. It is the platinum dihydroxide that then reacts with H_2O_2, regenerating Pt, in Equation (7-13). As the Pt produced in Equation (7-13) is immediately reconverted into platinum dihydroxide, the reaction gives a continuous supply of electrons to the Pt surface in the presence of H_2O_2.

$$Pt + 2H_2O \rightarrow \mathrm{Pt}(OH)_2 + 2H^+ + 2e^- \qquad E = 0.7V \qquad \textbf{(7-12)}$$

$$\mathrm{Pt}(OH)_2 + H_2O_2 \leftrightarrow Pt + H_2O + O_2 \qquad E = 0.7V \qquad \textbf{(7-13)}$$

The overall reaction is shown below in Equation (7-14).

$$Pt + H_2O_2 \rightarrow \mathrm{Pt} + 2H^+ + 2e^- + O_2 \qquad E = 0.7V \qquad \textbf{(7-14)}$$

7.6.1.1 Platinum black nanoparticle electrodeposition

Pt is an expensive metal and is technically much more difficult to use for coating than Au. Instead of coating the MNA with Pt using traditional methods (e.g. thermal evaporation), a thin layer of 'platinum black' nanoparticles was

electrodeposited onto the surface of the already Au-coated MNAs. The platinising solution used for the electrodeposition was made from 3 wt.% chloroplatanic acid hydrate and 0.005 wt.% lead acetate in deionised water.[107] A constant current was applied at -15 mA/cm^2 for 60 seconds using an Ag|AgCl reference electrode and a large mesh Pt counter electrode. The mesh counter electrode should be large enough as to not limit the flow of current between itself and the working electrode, to allow the growth of platinum black nanoparticles (Pt black NPs) on the working electrode's Au surface. The surface area of the base-insulated Au-coated MNA was calculated using the surface area of a cone (microneedle height approximated based on measurements of similar MNAs) multiplied by the number of microneedles in the MNA (see Equation (7-15), where h = cone height and r = cone radius). The MNA was held in the solution using a micromanipulator, as previously shown in Figure 7-23, and the mesh counter electrode was placed underneath the MNA. The solution was constantly stirred while the current was applied.

$$Surface\ area = \pi r\sqrt{h^2 + r^2} \times N_{spikes} \quad \textbf{(7-15)}$$

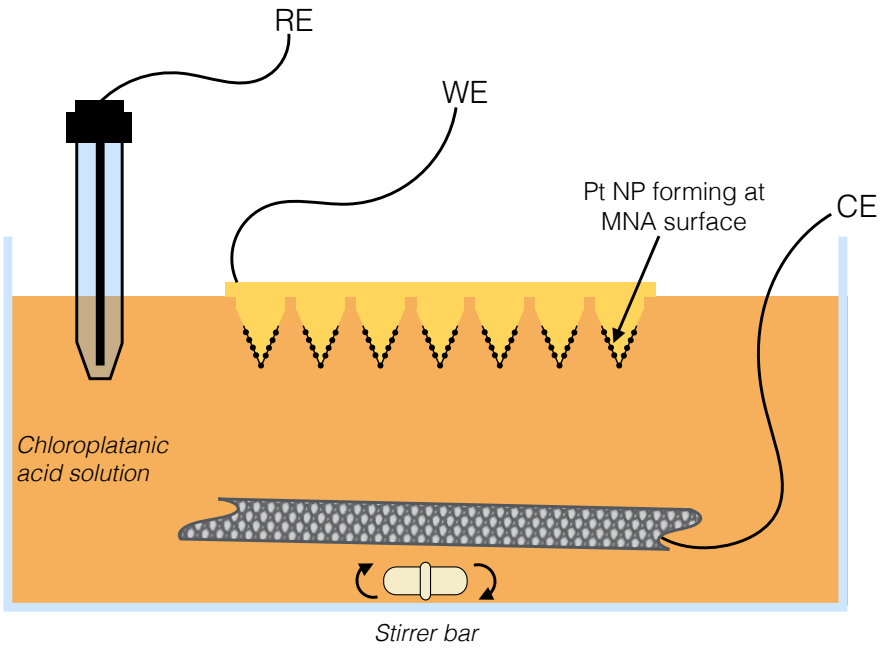


Figure 7-27: Platinum black nanoparticle deposition onto gold surface. *Platinum black nanoparticle (Pt NP) deposition onto the Au surface of the working electrode (WE) in a solution of 3 wt.% chloroplatanic acid and 0.005 wt.% lead acetate in deionised water. A large, Pt mesh counter electrode (CE) and Ag|AgCl reference electrode (RE) were used. The CE must be large enough as to not limit the flow of current between the working and counter electrodes.*

After the Pt black had been deposited onto the microneedles, the MNA was rinsed thoroughly with deionised water and then kept wet in deionised water to ensure the nanoparticles did not coagulate. The electrode was then cycled 10 times in 0.5 mM sulphuric acid to help stabilise them (-200 mV to 1300 mV, 50 mV/s).[107] Due to the insulation process described in Section 7.5, only the exposed gold tips of the microneedles were electrodeposited with platinum black (Figure 7-28). The electrodeposited Pt black nanoparticles will give the Pt black surface a larger surface area than a regular Pt surface, making it even better for oxidising H_2O_2.

Figure 7-28: Exposed gold of MNAs coated with Pt black nanoparticles. *The base of the MNA is uncoated in Pt black NPs as it has been electrically insulated and so remains gold in appearance.*

7.6.2 Lactate biosensor fabrication

The next step in biosensor fabrication was to electropolymerise an exclusion layer of m-phenylenediamine (poly-mPD) (*100 mM mPD in 10 mM phosphate buffered saline (PBS), pH = 7.4, 20 minutes at +0.7 V*) onto the surface of the Pt black MNA.[108] This stops any electroactive molecules that may be present in the sample from being oxidised on the electrode surface, reducing signal interference and improving the SNR of the sensor. The lactate oxidase (LOx) enzyme encapsulated in a hydrogel (*composed of 60 mg/mL lactate oxidase, 30 mg/mL bovine serum albumin (BSA), 2% glycerol and 45.6 mg/mL poly(ethylene*

glycol) diglycidyl ether (PEG-DE) in 10 mM PBS, pH = 7.4)[109,110] was then deposited onto the MNA spikes using a pipette, and cured in the oven for 2 hours at 55°C. The full layering of the lactate biosensor MNA is shown in Figure 7-29.

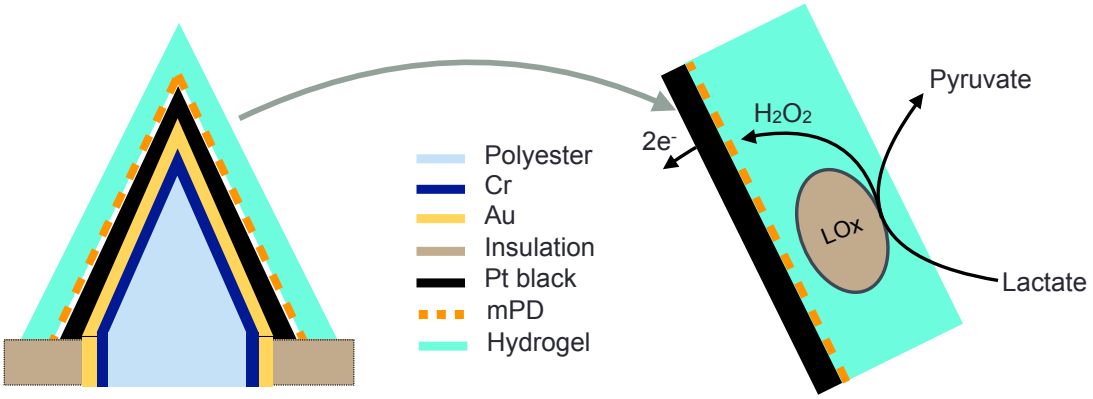


Figure 7-29: Lactate biosensor MNA layering. *An mPD exclusion layer was electropolymerised onto the Pt black surface of the fabricated MNA electrodes. On top of this, LOx encapsulated in a PEG-DE hydrogel was deposited onto MNA spikes and cured.*

The lactate biosensor-MNA was calibrated for characterisation. Known concentrations of sodium L-lactate were injected into a known volume of 10 mM phosphate buffer saline (PBS) solution and the corresponding current was measured, as shown in Figure 7-30. The PBS was constantly stirred during the calibration to provide good mass transport to the electrode. After each injection, the current was left to stabilise and an average was taken for each step change in lactate concentration. The mean current recorded for every step change in lactate concentration was then plotted against lactate concentration and fitted with a Michaelis-Menten curve, shown in Figure 7-30. The lactate biosensor-MNA had a K_m of 0.563 ± 0.080 mM (cf. K_m of free enzyme = 0.7 mM). The linear range of the lactate biosensor MNA goes up to approximately 1 mM lactate. Applying a diffusion-limiting layer of polyurethane can extend this linear range further, but at the expense of sensitivity.

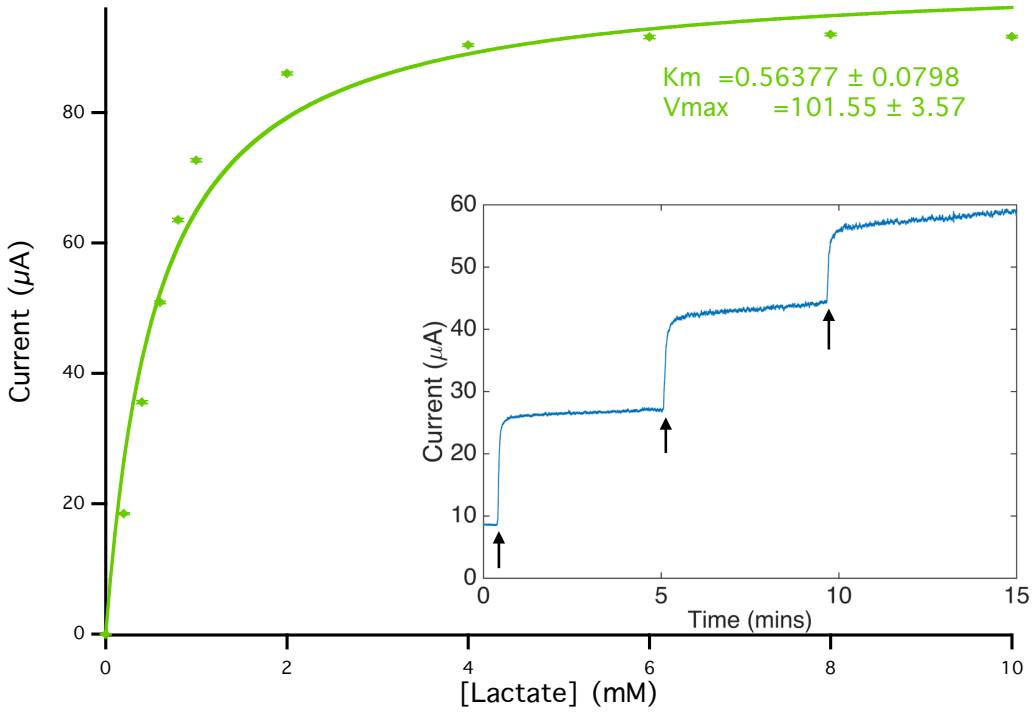


Figure 7-30: Current response for lactate biosensor MNA. *Step changes in lactate concentration: 0.2, 0.4, 0.6, 0.8, 1.0, 2.0, 4.0, 6.0, 8.0, and 10.0 mM. The K_m calculated for the lactate biosensor- MNA is 0.563 ± 0.080 mM. The first few steps of the raw, unfiltered calibration data are shown in the time series plot insert. Arrows indicate when each successive dose of sodium L-lactate was injected into the solution.*

The steps of the calibration data shown in the Figure 7-30-insert show the response times of the sensor for the first three doses of sodium L-lactate. The data is raw and unfiltered, and shows the LOx-MNA is a good, dynamic sensor with a fast reponse. The rise times ($t_r = t_{90} - t_{10}$) for the first steps in the calibration (linear range of sensor) are shown in Table 7-5.

Table 7-5: Rise time of the first five steps in the calibration of the Lactate biosensor MNA.

Dose	1	2	3	4	5	Mean
t_r (s)	6.12	10.32	9.24	7.20	9.00	8.37

7.6.2.1 Biocompatibility

There are several improvements and adjustments that need to be made to the sensing setup to make it biocompatible and safe for human use. The use of mPD is not desirable for human use due to the toxic properties of the monomer. However, removing this exclusion layer would mean that other molecules could

be oxidised and could interfere with the signal. One solution to this would be to use a second MNA electrode without the enzyme in the PEG-DE hydrogel layer as a sentinel electrode. The current response produced by this electrode could be subtracted from the biosensor response to give a measurement for H_2O_2 only without the response from common interference. For additional biocompatibility, a layer of parylene could be applied to the top surface of the electrode. Parylene is known for its biocompatibility and its suitability for human use, its high mechanical strength, and that it can be made nanoporous for conductivity.[83]

7.7 Individually Addressable Microneedle Arrays

The gold-coated MNAs shown in Figure 7-22 form a single sensor, made up of many microneedles connected together on a single pad. It would be desirable to create MNAs of which each microneedle acts as an individually addressable sensor. This would grant the MNAs the same advantages as an HD-sEMG array or would mean that each microneedle could also be functionalised separately to create a multifunctional sensing platform i.e. EMG, metabolite, or ionic sensing.

Masking off different regions of the MNA before metallic deposition is one method that could be used isolate individual microneedles. Sharma *et al.* use this method to isolate small arrays of microneedles.[80,96,98] Applying this to individual needles, however, would prove very difficult, especially when approaching the micro-scale. Connecting electrical wires to each individual microneedle in the way described in Section 7.5 would be very difficult without damaging the surrounding microneedle structures in the process, again especially when approaching the micro-scale. The proposed solution is to incorporate raised channels/routing into the MNA design, connecting the microneedle to raised pads that are easy to connect to, as shown in Figure 7-31 and Figure 7-32.

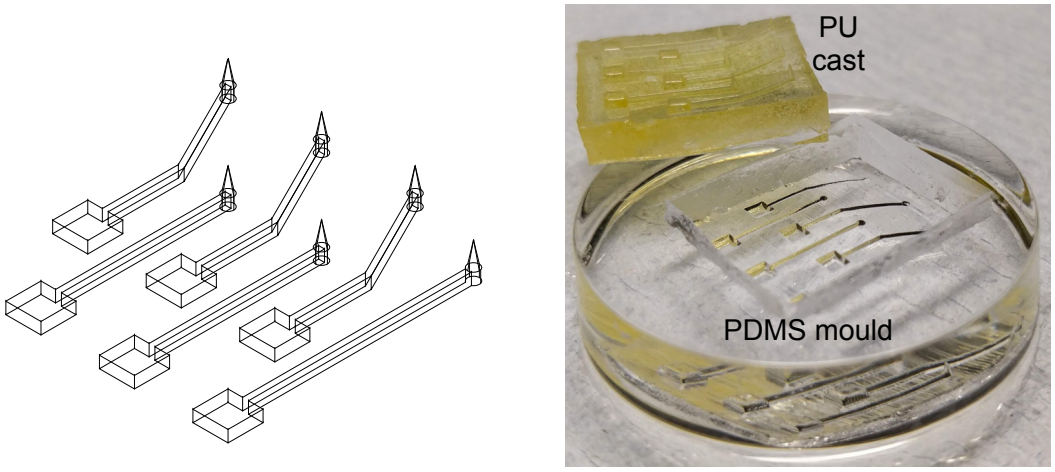


Figure 7-31: Model of 3D printed master used for the individually addressable MNA and the resulting PU cast and PDMS mould from the second cycle of the cast-shrinkage process. *The 3D printed master used had a 2000 µm microneedle height and 1200 µm base width, with a 1000 µm high stilt (total microneedle height 3000 µm). Channels/traces of the master were 500 µm in width and 1000 µm high, leading to 3000 µm x 3000 µm square pads. Electrical wire connections will be made to these square pads in the final cast.*

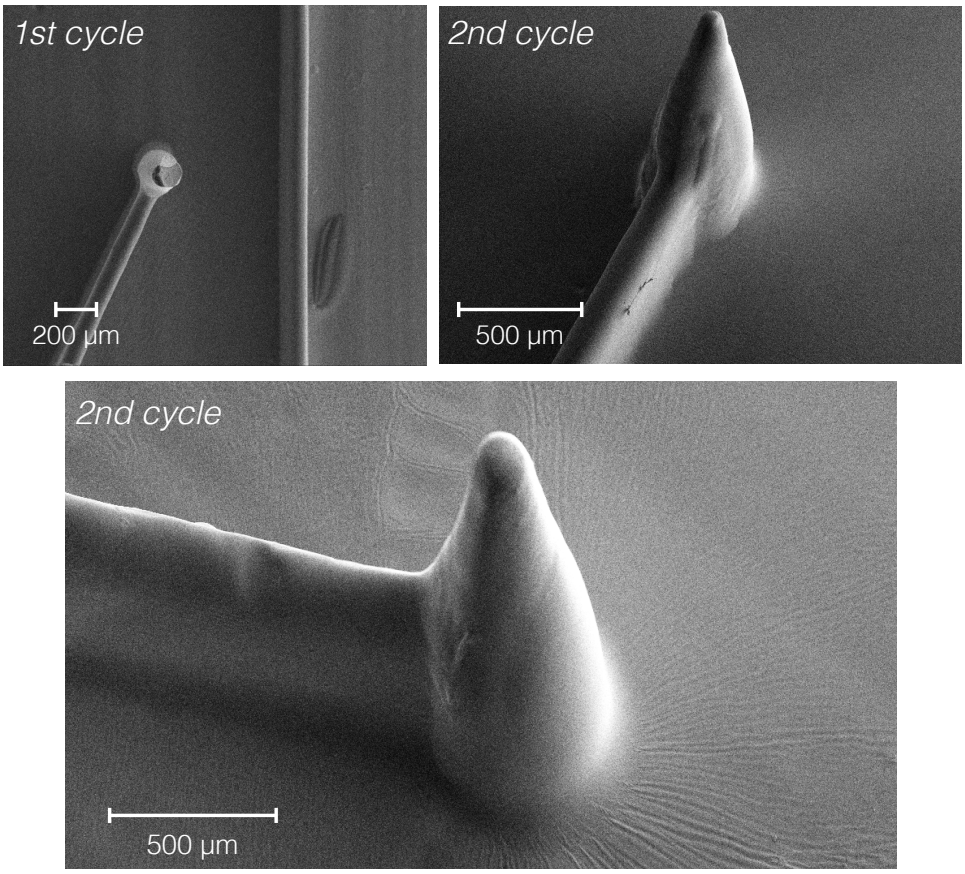


Figure 7-32: SEM images of the first and second-cycle casts of the individually addressable MNAs. *The geometries of the channels and the microneedles were preserved throughout the shrinking cycle process.*

The next step was to deposit a layer of PDMS in between the microneedles, channels, and pads to provide a sacrificial layer that can be removed after metal deposition. This was done by pipetting PDMS between the microneedles and channels. After the metallic deposition, the layer of sacrificial PDMS was removed leaving the individually addressable gold-coated microneedles, as shown in the schematic in Figure 7-33 and the photographs in Figure 7-34.

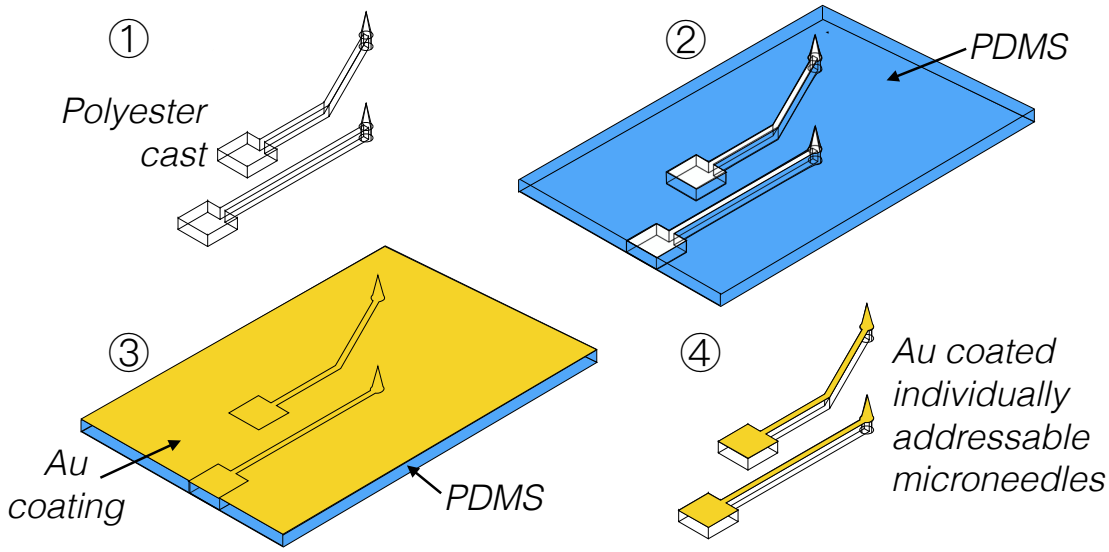


Figure 7-33: Steps 1-4 of coating individually addressable microneedles. *(1) Prepare polyester cast of shrunken MNA. (2) Fill gaps between microneedles and channels with PDMS. (3) Coat PDMS and polyester with conductive material (e.g. Au). (4) Peel off/remove sacrificial PDMS layer, leaving conductive traces to individually addressable microneedles.*

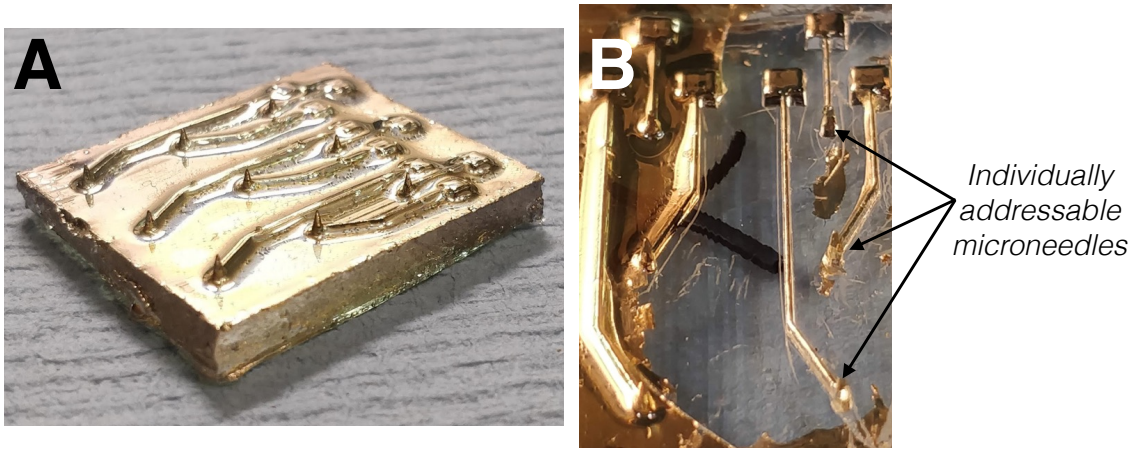


Figure 7-34: Fabricating Au coated individually addressable microneedles. *(A) PDMS and polyester surface are sputter coated with Au (50 nm). (B) Half-way through removal of sacrificial PDMS, revealing Au coated individually addressable microneedles.*

After the layer of sacrificial PDMS was removed, just the top surface of the raised channels, pads, and microneedles were left coated in Au. The continuity of the connection from the pad (where a wire would theoretically be connected to) to the tip of microneedle was tested with a multimeter and was successful. Figure 7-35 shows the Au coating of the channels connected to the Au coated microneedles.

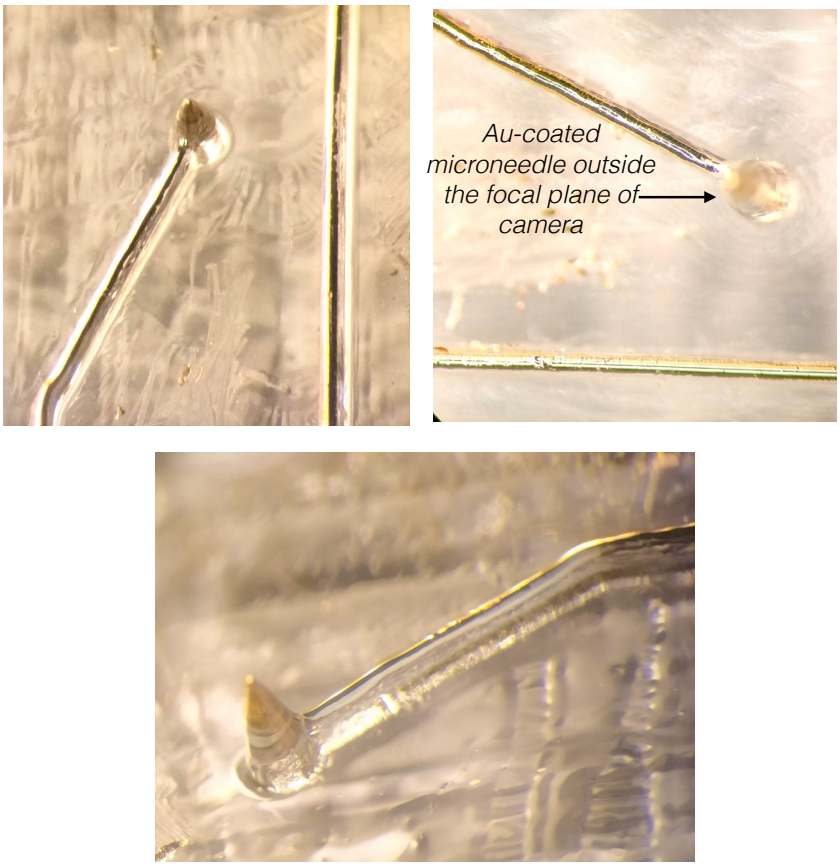


Figure 7-35: Au coated individually addressable microneedles. *The sacrificial layer of PDMS is removed, resulting in Au-coated individually addressable microneedles. The connection/continuity between the traces and microneedles was tested using a multimeter.*

Using conductive Ag epoxy (see Section 7.5), a wire was connected to the square pad at the end of the Au channel/trace. The trace and the Ag epoxy connection were then insulated using the insulating resin described in Section 7.5. The individually addressable Au microneedle was tested by CV in 1.5mM ferrocene, and compared with the previous CVs for the Au MNA (see Figure 7-36). The currents were normalised by the approximate surface area of exposed gold (1.13 mm^2, calculated from surface area of cone), as previously shown in Figure 7-25.

A slower scan rate of 1 mV/s was used to show the microelectrode effects of the individually addressable microneedle.

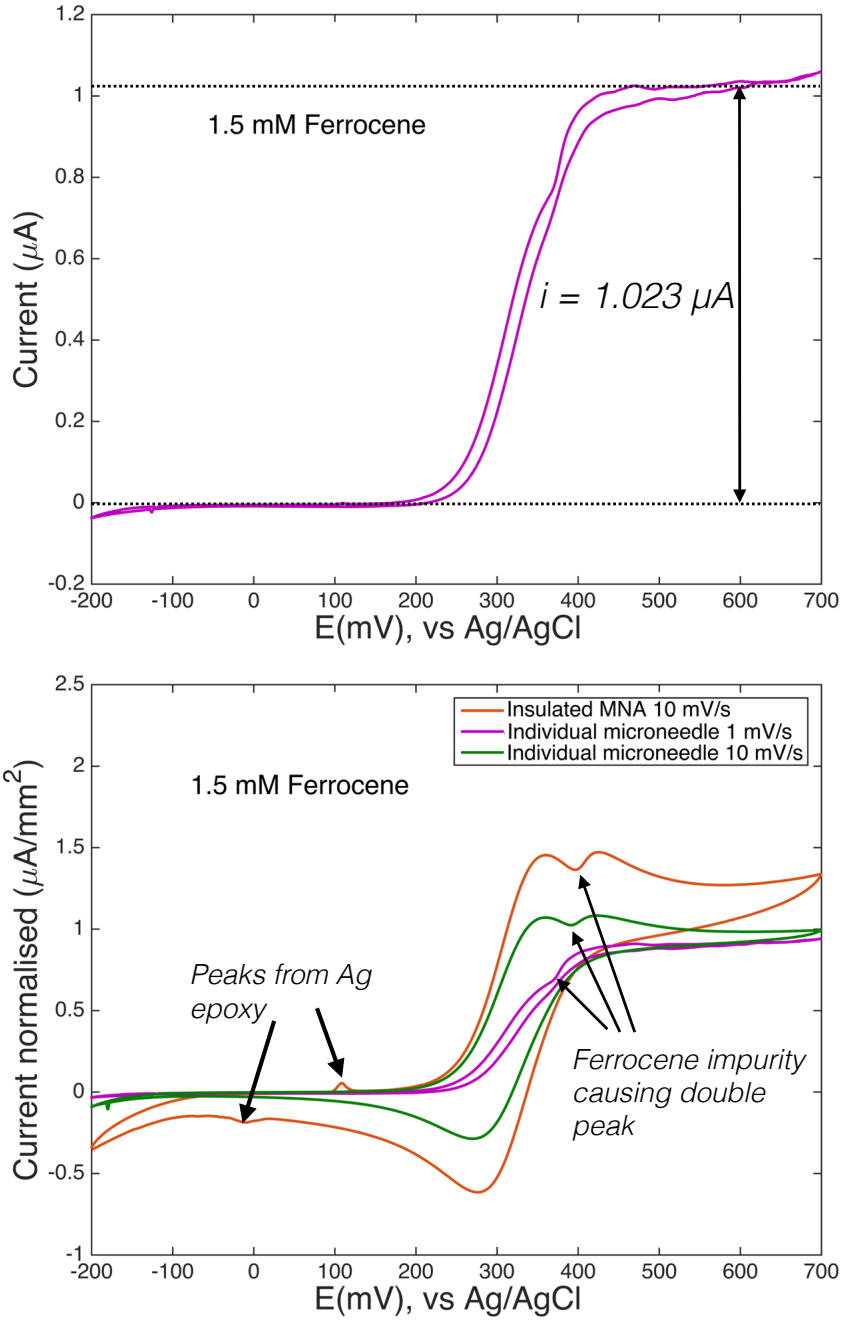


Figure 7-36: CVs of Au vs. Ag|AgCl in 1.5 mM ferrocene monocarboxylic acid of an individually addressable microneedle. *(Top) CV of individual addressable Au microneedle vs. Ag|AgCl in 1.5 mM ferrocene monocarboxylic acid (Ferrocene), 1 mV/s scan rate, showing a steady state current of 1.023 μA. (Bottom) The CVs of the insulated MNA and individually addressable microneedle are normalised using the approximate calculated surface area of the exposed gold on MNA (insulated = 55.2 mm², individually microneedle = 1.13 mm²). Insulated MNA scan rate at 10 mV/s. Slower scan rate of 1 mV/s of the individual microneedle demonstrates microelectrode effects.*

The Au microneedle is from the first cycle of the casting-shrinkage process, and has a base radius of and height of approximately 0.3 mm and 1 mm, respectively. If the microneedle was modelled as a disc electrode with these dimensions, the steady state current is calculated using Equation (7-16) as i_{disc} = 0.0997 μA, which is far smaller than the measured current of 1.023 μA. Modelling the microneedle a hemisphere (see Equation (7-17)) is a closer approximation to the shape of the microneedle, but still only gives a current of $i_{hemisphere}$ = 0.1563 μA.

$$i_{disc} = 4nFDcr \quad \textbf{(7-16)}$$

$$i_{hemisphere} = 2\pi nFDcr \quad \textbf{(7-17)}$$

Where n is the number of electrons, F is Faraday's constant, D is the diffusion coefficient, c is the concentration of ferrocene, and r is the radius of the electrode.

A closer approximation is to model the electrode as a cone, using the model for steady state current for a finite conical electrode developed by Zoski et al. (see Equation (7-18)).[105]

$$i_{cone} = 4nFDcr(1 + qH^p) \quad \textbf{(7-18)}$$

Where H is the ratio of cone height h to cone radius r (H = h/r), q = 0.30661, and p = 1.14466.

However, this still only results in a current of i_{cone} = 0.2206 μA. This smaller current is explained by the fact Zoski et al. model a conical electrode inlaid in insulative material (see Figure 7-37(A)). The individually addressable microelectrode behaves like a conical electrode on a stilt (see Figure 7-37(B)), and so the additional diffusion around its bottom edge will likely explain the larger measured steady state current of 1.023 μA.

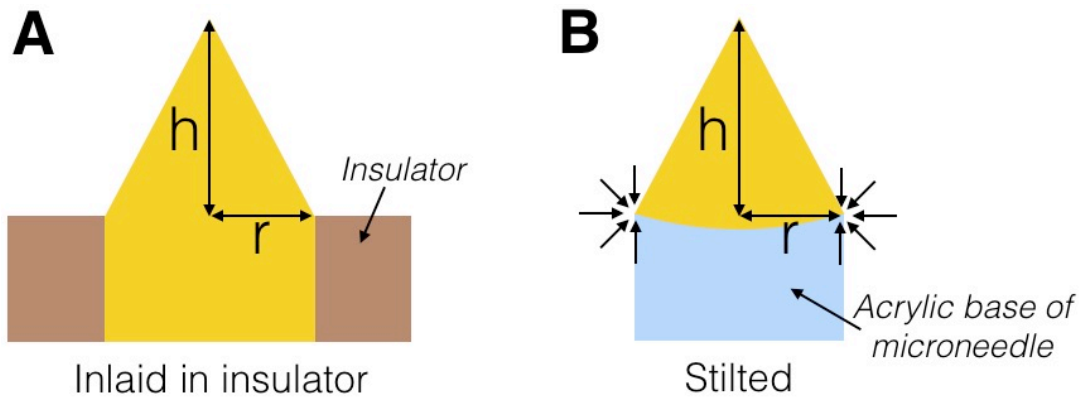


Figure 7-37: Conical electrode, inlaid vs. stilted. *(A) Conical electrode inlaid in insulator, as modelled by Zoski et al.*[105] *(B) Once the sacrificial PDMS layer as been removed, the resulting Au-coated microneedle should be modelled as a conical electrode upon a stilt. Additional diffusion around bottom edge of microneedle explains the larger steady state current measured.*

The current method for depositing the sacrificial PDMS layer is by pipetting. This has been successful using shrunken structures from the first cycle of the casting-shrinkage process (h ≈ 1 mm, r ≈ 0.3 mm), however, it is likely that this method will not suffice once the MNAs are shrunk further. Spin coating PDMS, or another easily removable material, could be a possible solution for creating a thin, removable layer when the individually addressable MNAs approach the micro-scale. However, if this is what is required to produce individually addressable MNAs by casting-shrinkage cycles, then the method has become expensive and requires skilled personnel, which was not the intention when developing this method.

7.8 Conclusion

This chapter has shown a new route for the prototyping of transdermal biomedical measurement devices using a 'cast-shrinkage' method, based on the method described by Sun *et al.*[49] This chapter has shown that iterative, cast-shrinkage cycles can be used to create microneedle arrays approximately 15% of the size of the original 3D printed master; in three cycles, the microneedles were shrunk from 2000 µm to 300 µm. It has been shown that the microneedles produced can be used to successfully penetrate through the top layer of skin, the highly resistive *stratum corneum*, without penetrating deep into the dermis that

would cause pain and draw blood. Doing so creates a low impedance skin-electrode contact,[85] and thus would allow for low-noise recordings. This would be an ideal replacement for taking electrophysiological recordings such as EMG, where the surface electrodes currently used are very susceptible to noise but are often a preferred option due to their painless application. Penetrating into the first layers of skin also gives the transdermal device access to interstitial fluid, which could potentially be monitored for analytes of interest if the MNA electrode is functionalised.

The MNAs produced using the cast-shrinkage cycles have been coated in gold for conductivity, and the benefits for insulating the exposed gold base of the MNA have been described and shown using cyclic voltammetry. The gold MNAs have been functionalised into a successful lactate biosensor. First, the gold surface was electrodeposited with platinum black nanoparticles, to provide the MNA with a superior surface for oxidising H_20_2. After applying an exclusion layer, a hydrogel containing the lactate oxidase enzyme was deposited onto the microneedles. The resulting lactate sensing MNA is a good, dynamic sensor, demonstrating fast response to different concentrations of lactate.

The use of producing individually addressable microneedles using casting-shrinkage cycles has also been investigated. It is possible to create such structures that could be made individually addressable, however the difficulty presented is how to electrically isolate each microneedle and its trace. Using a layer of sacrificial PDMS before coating the MNA in conductive material, to then be later removed, is a possible solution. This may work for the first casting-shrinkage cycle of MNAs, however when the structures are shrunk even further it is unlikely this is a viable method without the use of specialist machinery and/or skilled personnel.

Current fabrication methods used to produce transdermal devices and MNAs on the scale shown in this chapter require a combination of expensive equipment and specialist training, and can be inflexible to new designs or take a long time to produce. The casting-shrinkage method described in this chapter provides researchers, who are beginning to investigate the use of transdermal

devices and MNAs, a flexible, cheap, and simple route with which they can prototype a range of different designs on the micro-scale quickly.

Chapter 8
Conclusions and Future Work

8.1 Overview

The aim of this thesis was to improve the current instrumentation for monitoring the progression of Amyotrophic Lateral Sclerosis (ALS), by the detection of fasciculation potentials (FPs) recorded by electromyogram (EMG). This thesis has shown the development of a 'data-processing pipeline', a compact wireless device for capturing FPs, and a new prototyping route for producing transdermal devices using 'casting-shrinkage' cycles. The summarised work and outcomes of this thesis are as follows:

Signal conditioning and noise removal (Chapter 3)

The data processing pipeline includes the signal conditioning steps to remove high frequency components associated with white noise and low frequency components such as movement artefacts. Efforts to investigate a more suitable method of 50 Hz noise removal have been made; methods such as the empirical mode decomposition show promise, but are not yet tangible due to the computational load it imposes of current computing systems.

Automatic channel exclusion (Chapter 3)

When using HD-sEMG grid arrays that contain a high number of channels, there is a likelihood that some channels will have a poor connection resulting in either a very noisy recording or no recording at all (i.e. a dead-channel). These channels

need to be removed from the data-processing pipeline so that they do not interfere with the detection of FPs. A novel metric termed *'channel-activity'* has been developed and is used to exclude those channels that have recorded abnormally high levels of noise or those showing no recording at all.

Probabilistic spike detection (Chapter 3)

A novel method of detecting FPs has been developed. The method uses all the datapoints in the signal to form a probability function to provide a rational basis for selecting a threshold for FP detection; rather than using an arbitrary threshold value for detection, the probability threshold chosen is a variant of factors such as grid placement, electrode, skin, gel, patient, and operator. The detected 'epochs' of data are subjected to a refining algorithm, which combines epochs in close proximity to each other to form the FP. The FP detected at each time-stamp with the greatest amplitude across all 64 channels is extracted and used to form the 'super-channel'; reducing the number of channels that need to be examined in the latter stages of the data-processing pipeline, and making the data easier to examine visually for the clinician/investigator.

Fasciculation Potential grouping (Chapter 4)

Using a novel, iterative framework termed 'cluster-and-seed', the FP waveforms are grouped together using their principle components and K-means clustering; the number of cluster centres required is iteratively updated by examining the datapoints that fall outside of a distance threshold from the cluster centre they have been assigned to. The morphology of each cluster group is checked using 'matched filtering'; the peak and symmetry of the impulse responses outputted as a result convolving FPs together are examined to determine whether the morphologies of the FPs in the cluster truly do match. The FPs in each group are then aligned and averaged to produce a template for each group. The FP templates are then checked for matching morphologies using matched filtering, as FP subgroups may have been formed in the clustering stage. Finally, template groups are checked manually in order to maintain the high specificity of the data-processing pipeline. The FP waveforms have been grouped, and can now be parameterised and have measures of their waveform variability extracted for analysis.

Surface Potential Quantification Engine (Chapter 5)

A stand-alone application has been developed, named 'Surface Potential Quantification Engine' (SPiQE) after the collaboration with clinicians at King's College London (KCL). The application uses all the algorithms and frameworks described in Chapters 3 and 4. SPiQE takes the user through each stage of the data-processing pipeline; the application gives the user flexibility on how they wish to process and analyse their data. For example, the user can choose which channels they wish to examine or the frequency range of the band-pass filter (should they choose to apply it).

Wireless multiple-channel device (Chapter 6)

A compact, wireless device has been developed for the capture of fasciculation potentials. The device has 8 channels, is low-power, high resolution (16-bit), and capable of recording data at a high sampling rate (~4 kHz). The device has a Bluetooth transceiver for issuing commands and streaming data at a low data rate for the purpose of checking signal quality (i.e. to ensure there is a good skin-electrode contact). The device stores data to a buffer array in the microcontroller's SRAM and saves it all in a single write action to an SD card. The device is programmed to capture just the FP from the incoming signal. The device has been tested using EMG data recordings taken from ALS patients at KCL, capturing the FP in short segments of data across eight channels at high resolution and high sampling rate.

New prototyping method for transdermal devices (Chapter 7)

A new method for prototyping transdermal devices and microneedle arrays (MNAs) has been developed, using the 'casting-shrinkage' method originally described by Sun et al.[49] for producing microfluidic devices. The method has been adapted successfully, producing microneedles smaller than 300 μm in height, microneedles of various different profiles and geometries, and different MNA layouts. Skin penetration tests show the microneedles successfully penetrating through the *stratum corneum* into the conductive layers of the epidermis below. The MNAs have been coated with gold for conductivity, and also functionalised into a working biosensor for the measurement of lactate.

Finally, the MNAs have been made individually addressable by using sacrificial layers of PDMS to remove areas of gold between the microneedles.

8.2 Future Work

The algorithms and frameworks written for signal conditioning, detection of FPs, and the grouping of their morphology have been written in to a stand-alone application (SPiQE). The conditioning and FP sections of the data-processing pipeline have been validated.[70] Future work should include the full parameterisation of the FP waveform morphology to produce a composite measure of waveform variability and complexity. This can then be validated against current methods of assessing disease progression using a receiver operator characteristic (ROC) curve.

Morphology is not the only metric to be gained from detected FPs; as the disease progresses and more motor units in the muscle become diseased their occurrence is likely to increase. Collaborators at King's College London are investigating the effect progression has on the inter-spike-interval and amplitude of FPs detected across the entire muscle, using the probabilistic spike detection algorithm I have developed. As well as this, they have also developed a novel method for the exclusion of voluntary MUAPs from EMG recordings, termed *AVID* (Active Voluntary Identification). Both the analysis of FP amplitude and inter-spike interval, and the new method for removing voluntary MU activity are to be incorporated into the SPiQE application as part of the continuing collaboration with King's College London.

Increasing the amount of SRAM the microcontroller has available to it, possibly through an external memory chip incorporated in a future device's design, would allow for more data to be stored in the temporary buffer before saving to the SD card. This additional memory could be used for storing longer FPs (if saving at the current high sampling rate of ~4 KHz) or for storing more channels of data. The device currently has eight channels, far fewer than the 64 channel HD-sEMG system used in the longitudinal studies at King's College London. The number of channels can be increased by 'daisy-chaining' ADS1298 modules together, however this may impact the data rate of the device and the

length of the FPs that can be saved. The optimal number of electrodes and the inter-electrode spacing for HD-sEMG should be investigated so that the device does not have more channels than is necessary to successfully capture FPs.

At present, the Bluetooth transmission of data is intended only for checking the quality of the incoming signals being recorded; the Bluetooth transmission is slow, and creates a 'bottle-neck' if data is streamed and saved to the SD card concurrently. Faster methods of transceiving data should be investigated so that channel quality can be observed at the same time that FPs are saved locally to the device's on-board memory, however these should still adhere to the requirements that the device needs to be low-power, battery-operated, and compact. It may also be advantageous to engage with patient/relative groups to gain their input on the device form and functionality.

The casting-shrinkage cycles have proven to be a successful method for prototyping new MNA configurations, however the current method is not translational for large scale manufacture. The ideal microneedle geometry (i.e. profile, footprint etc.) and MNA layout for skin penetration, electrophysiological measurement, and biosensing is still to be investigated and characterised. The casting-shrinkage method lends itself to the rapid prototyping of different MNA designs and layouts. Once the ideal microneedle geometry and layout of the transdermal device has been found (according its intended application), it should be produced using a more sophisticated fabrication method, for example injection-moulding,[80] to quickly manufacture the MNAs on a large scale.

The concept of developing a multi-sensing MNA platform (i.e. both electrophysiology and biosensing on the same MNA) is one that should be investigated; it has been shown that the casting-shrinkage method can be used to produce both individually addressable microneedles and microneedles that can be functionalised into a working biosensor, and so the concept is not far away from realisation. ALS patients are found to have elevated levels of the neurotransmitter glutamate which is thought to mediate the excitotoxicity of motor neurons (the drug Riluzole is an anti-glutamate agent).[13,18,27] The PEG-DE hydrogel used to create the lactate-biosensor described in Section 7.6 can also be used for encapsulating the glutamate-oxidase enzyme.[111] A multi-sensing

platform combining an array of low-noise transdermal EMG electrodes with a glutamate biosensor could prove to be an effective tool in assessing the progression of ALS.

Bibliography

1. Hardiman, O., van den Berg, L. H. & Kiernan, M. C. Clinical diagnosis and management of amyotrophic lateral sclerosis. *Nat. Rev. Neurol.* **7,** 639–649 (2011).
2. Wolf, J. *et al.* Causes of death in amyotrophic lateral sclerosis: Results from the Rhineland-Palatinate ALS registry. *Nervenarzt* **88,** 911–918 (2017).
3. Bensimon, G., Lacomblez, L. & Meininger, V. A Controlled Trial of Riluzole in Amyotrophic Lateral Sclerosis. *N. Engl. J. Med.* **330,** 585–591 (1994).
4. Miller, R., Mitchell, J. & Moore, D. Riluzole for amyotrophic lateral sclerosis (ALS)/motor neuron disease (MND). *Cochrane Database Syst. Rev. 2012,* 1465–1858 (2012). doi:10.1002/14651858.CD001447.
5. Benatar, M. *et al.* ALS Biomarkers for Therapy Development: State of the Field & Future Directions. *Muscle Nerve* **53,** 169–182 (2016).
6. Leis, A. A. & Schenk, M. P. *Atlas of Nerve Conduction Studies and Electromyography*. (Oxford University Press, 2013).
7. Bokuda, K. *et al.* Quantitative analysis of the features of fasciculation potentials and their relation with muscle strength in amyotrophic lateral sclerosis. *Neurol. Sci.* **37,** 1939–1945 (2016).
8. Krarup, C. Lower motor neuron involvement examined by quantitative electromyography in amyotrophic lateral sclerosis. *Clin. Neurophysiol.* **122,** 414–422 (2011).
9. Cedarbaum, J. M. *et al.* The ALSFRS-R : a revised ALS functional rating scale that incorporates assessments of respiratory function. *J. Neurol. Sci.* **169,** 13–21 (1999).
10. Herndon, R. M. *Handbook of Neurologic Rating Scales*. (Demos Medical Publishing, 2006).
11. Shefner, J. M. *et al.* Multipoint incremental motor unit number estimation as an outcome measure in ALS. *Neurology* **77,** 235–241 (2011).
12. Neuwirth, C., Nandedkar, S., Stålberg, E. & Weber, M. Motor Unit Number Index (MUNIX): A novel neurophysiological technique to follow disease progression in amyotrophic lateral sclerosis. *Muscle and Nerve* **42,** 379–384 (2010).
13. Mills, K. R. & Nithi, K. A. Corticomotor threshold is reduced in early sporadic amyotrophic lateral sclerosis. *Muscle and Nerve* **20,** 1137–1141 (1997).
14. Kanai, K. *et al.* Altered axonal excitability properties in amyotrophic lateral sclerosis: Impaired potassium channel function related to disease stage. *Brain* **129,** 953–962 (2006).
15. Vucic, S. & Rutkove, S. B. Neurophysiological biomarkers in amyotrophic lateral sclerosis. *Curr. Opin. Neurol.* **31,** 640–647 (2018).
16. de Carvalho, M. & Swash, M. Origin of Fasciculations in Amyotrophic Lateral Sclerosis and Benign Fasciculation Syndrome. *JAMA Neurol.* **70,** 1562–1565 (2013).
17. Kandel, E. R., Schwartz, J. H. & Jessel, T. M. *Principles of Neural Science*. (New York; McGraw-Hill, 2000).
18. De Carvalho, M., Kiernan, M. C. & Swash, M. Fasciculation in amyotrophic lateral sclerosis: Origin and pathophysiological relevance. *J. Neurol. Neurosurg. Psychiatry* **88,** 773–779 (2017).
19. Mills, K. R. Characteristics of fasciculations in amyotrophic lateral sclerosis and the benign fasciculation syndrome. *Brain* **133,** 3458–3469 (2010).
20. Grapperon, A. M. & Attarian, S. Disorders of motor neurons manifested by hyperactivity. *Rev. Neurol. (Paris).* **173,** 345–351 (2017).
21. Stegeman, D. F., Kleine, B. U., Lapatki, B. G. & Van Dijk, J. P. High-density Surface EMG: Techniques and Applications at a Motor Unit Level. *Biocybern. Biomed. Eng.* **32,** 3–27 (2012).

22. Drost, G., Kleine, B. U., Stegeman, D. F., van Engelen, B. G. M. & Zwarts, M. J. Fasciculation potentials in high-density surface EMG. *J. Clin. Neurophysiol.* **24,** 301–307 (2007).
23. de Carvalho, M. & Swash, M. Fasciculation potentials and earliest changes in motor unit physiology in ALS. *J. Neurol. Neurosurg. Psychiatry* **84,** 963–968 (2013).
24. de Carvalho, M. Pathophysiological significance of fasciculations in the early diagnosis of ALS. *Amyotroph Lateral Scler Other Mot. Neuron Disord.* **1,** S43–S46 (2000).
25. de Carvalho, M. *et al.* Electrodiagnostic criteria for diagnosis of ALS. *Clin. Neurophysiol.* **119,** 497–503 (2008).
26. Costa, J., Swash, M. & De Carvalho, M. Awaji criteria for the diagnosis of amyotrophic lateral sclerosis: A systematic review. *Arch. Neurol.* **69,** 1410–1416 (2012).
27. De Carvalho, M. & Swash, M. Fasciculation potentials: A study of amyotrophic lateral sclerosis and other neurogenic disorders. *Muscle and Nerve* **21,** 336–344 (1998).
28. Conradi, S., Grimby, L. & Lundemo, G. Pathophysiology of fasciculations in ALS as studied by electromyography of single motor units. *Muscle Nerve* **5,** 202–208 (1982).
29. Shimizu, T. & Bokuda, K. Complex Fasciculation Potentials in Amyotrophic Lateral Sclerosis. *US Neurol.* **11,** 53–56 (2015).
30. Eisen, A. & Vucic, S. Fasciculation potentials: A diagnsotic biomarker of early ALS? *J. Neurol. Neurosurg. Psychiatry* **84,** 948 (2013).
31. Trojaborg, W. & Buchthal, F. Malignant and Benign Fasciculations. *Acta Neurol. Scand.* **41,** 251–254 (1965).
32. Mills, K. R. The basics of electromyography. *J. Neurol. Neurosurg. Psychiatry* **76 Suppl 2,** ii32-i35 (2005).
33. Mills, K. R. Detecting fasciculations in amyotrophic lateral sclerosis: duration of observation required. *J. Neurol. Neurosurg. Psychiatry* **82,** 549–51 (2011).
34. Reaz, M. B. I., Hussain, M. S. & Mohd-Yasin, F. Techniques of EMG signal analysis: detection, processing, classification and applications. *Biol. Proced. Online* **8,** 11–35 (2006).
35. Kasi, P. K. *et al.* Characterization of motor unit behavior in patients with amyotrophic lateral sclerosis. *2009 4th Int. IEEE/EMBS Conf. Neural Eng.* 10–13 (2009). doi:10.1109/NER.2009.5109222
36. Ren, X., Mo, S., Huang, H. & Deng, L. Resolving superimposed action potentials for EMG signal decomposition. *2010 4th Int. Conf. Bioinforma. Biomed. Eng. iCBBE 2010* 1–4 (2010). doi:10.1109/ICBBE.2010.5514674
37. Fuglsang-Frederiksen, A. & Rønager, J. EMG power spectrum, turns-amplitude analysis and motor unit potential duration in neuromuscular disorders. *J. Neurol. Sci.* **97,** 81–91 (1990).
38. De Luca, C. J., Donald Gilmore, L., Kuznetsov, M. & Roy, S. H. Filtering the surface EMG signal: Movement artifact and baseline noise contamination. *J. Biomech.* **43,** 1573–1579 (2010).
39. Webster, J. G. *Medical Instrumentation: Application and Design*. (John Wiley & Sons, 2010).
40. Lapatki, B. G., Van Dijk, J. P., Jonas, I. E., Zwarts, M. J. & Stegeman, D. F. A thin, flexible multielectrode grid for high-density surface EMG. *J. Appl. Physiol.* **96,** 327–336 (2004).
41. Zhang, X. & Zhou, P. Analysis of surface EMG baseline for detection of hidden muscle activity. *J. Neural Eng.* **11,** 16011 (2014).
42. Kleine, B. U., Stegeman, D. F., Schelhaas, H. J. & Zwarts, M. J. Firing pattern of fasciculations in ALS. *Neurology* **70,** 353–359 (2008).

43. Jahanmiri-Nezhad, F., Barkhaus, P. E., Zev, W. & Zhou, P. Spike sorting paradigm for classification of multi-channel recorded fasciculation potentials. *Comput. Biol. Med.* **55,** 26–35 (2014).
44. Jahanmiri-Nezhad, F., Barkhaus, P. E., Rymer, W. Z. & Zhou, P. Sensitivity of fasciculation potential detection is dramatically reduced by spatial filtering of surface electromyography. *J. Chem. Inf. Model.* **53,** 1689–1699 (2013).
45. Jahanmiri-Nezhad, F., Li, X., Barkhaus, P. E., Rymer, W. Z. & Zhou, P. A Clinically Applicable Approach for Detecting Spontaneous Action Potential Spikes in Amyotrophic Lateral Sclerosis With a Linear Electrode Array. *J. Clin. Neurophysiol.* **31,** 35–40 (2014).
46. Zhou, P., Li, X., Jahanmiri-Nezhad, F., Rymer, W. Z. & Barkhaus, P. E. Duration of observation required in detecting fasciculation potentials in amyotrophic lateral sclerosis using high-density surface EMG. *J. Neuroeng. Rehabil.* **9,** 78 (2012).
47. Sia, S. K. & Whitesides, G. M. Microfluidic devices fabricated in poly(dimethylsiloxane) for biological studies. *Electrophoresis* **24,** 3563–3576 (2003).
48. Leong, C. L. Bioengineered microfluidic devices for the real-time clinical measurement of neurochemicals. (Imperial College London, 2013).
49. Sun, M., Xie, Y., Zhu, J., Li, J. & Eijkel, J. C. T. Improving the Resolution of 3D-Printed Molds for Microfluidics by Iterative Casting-Shrinkage Cycles. *Anal. Chem.* acs.analchem.6b05148 (2017). doi:10.1021/acs.analchem.6b05148
50. Martínez-Silva, M. de L. *et al.* Hypoexcitability precedes denervation in the large fast-contracting motor units in two unrelated mouse models of ALS. *Elife* **7,** 1–26 (2018).
51. Vázquez-Costa, J. F. *et al.* New insights into the pathophysiology of fasciculations in amyotrophic lateral sclerosis: An ultrasound study. *Clin. Neurophysiol.* **129,** 2650–2657 (2018).
52. Koch, V. M., McGill, K. C. & Loeliger, H. A. Resolution of superpositions in EMG signals using belief propagation: Results for the known constituent problem. *Annu. Int. Conf. IEEE Eng. Med. Biol. - Proc.* 1260–1263 (2006). doi:10.1109/IEMBS.2006.259341
53. Ren, X., Hu, X., Wang, Z. & Yan, Z. MUAP extraction and classification based on wavelet transform and ICA for EMG decomposition. *Med. Biol. Eng. Comput.* **44,** 371–382 (2006).
54. Zhang, X. & Zhou, P. Analysis of surface EMG baseline for detection of hidden muscle activity. *J. Neural Eng.* **11,** 16011 (2014).
55. Nakamura, H., Yoshida, M., Kotani, M., Akazawa, K. & Moritani, T. The application of independent component analysis to the multi-channel surface electromyographic signals for separation of motor unit action potential trains: Part I - measuring techniques. *J. Electromyogr. Kinesiol.* **14,** 433–441 (2004).
56. Mewett, D. T., Nazeran, H. & Reynolds, K. J. *Removing Power Line Noise From Recorded EMG*. *IEEE 23rd Annual conference proceedings* (2001).
57. Merletti, R., Botter, A., Troiano, A., Merlo, E. & Minetto, M. A. Technology and instrumentation for detection and conditioning of the surface electromyographic signal: State of the art. *Clin. Biomech.* **24,** 122–134 (2008).
58. Kasi, P. K. Characterization of Motor Unit Discharge Rate in Patients with Amytrophic Lateral Sclerosis (ALS) Thesis Committee : (2009).
59. De Luca, C. J., Kuznetsov, M., Gilmore, L. D. & Roy, S. H. Inter-electrode spacing of surface EMG sensors: Reduction of crosstalk contamination during voluntary contractions. *J. Biomech.* **45,** 555–561 (2012).
60. Wickham, A. Tracking the progression of Amyotrophic Lateral Sclerosis ALS using wireless sensors. (Imperial College London, 2015).

61. Baratta, R. V., Solomonow, M., Zhou, B. H. & Zhu, M. Methods to reduce the variability of EMG power spectrum estimates. *J. Electromyogr. Kinesiol.* **8,** 279–285 (1998).
62. Piskorowski, J. Time-efficient removal of power-line noise from EMG signals using IIR notch filters with non-zero initial conditions. *Biocybern. Biomed. Eng.* **33,** 171–178 (2013).
63. Mewett, D. T., Reynolds, K. J. & Nazeran, H. Reducing power line interference in digitised electromyogram recordings by spectrum interpolation. *Med. Biol. Eng. Comput.* **42,** 524–531 (2004).
64. Akwei-Sekyere, S. Powerline noise elimination in biomedical signals via blind source separation and wavelet analysis. *PeerJ* **3,** e1086 (2015).
65. Yacoub, S. & Raoof, K. Noise Removal from Surface Respiratory EMG Signal. *Int. J. Electron. Commun. Eng.* **2,** 266–273 (2008).
66. Zhang, X. & Zhou, P. Analysis of Surface EMG Baseline for Detection of Hidden Muscle Activity. *J. Neural Eng.* **11,** 1–19 (2014).
67. Hu, X. & Wang, Z. Detecting the motor unit action potential from surface EMG signals based on wavelet transform. *Biomed. Circuits Syst. 2004 IEEE ...* 15–18 (2004). at <http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1454166>
68. TMS International BV. High Density EMG Grids. (2015). at <http://www.tmsi.com/products/accessories/item/hd-emg-grids>
69. Rosenfalck, P. Electromyography: sensory and motor conduction. Findings in normal subjects. *Lab. Clin. Neurophysiol. Rigshospitalet* 1–49 (1975).
70. Bashford, J. *et al.* SPiQE: An automated analytical tool for detecting and characterising fasciculations in amyotrophic lateral sclerosis. *Clin. Neurophysiol.* (2019). doi:10.1016/j.clinph.2019.03.032
71. Preston, D. C. & Shapiro, B. E. in *Electromyography and Neuromuscular Disorders* 220–234 (W.B. Saunders, 2013). doi:https://doi.org/10.1016/B978-1-4557-2672-1.00014-3
72. Christodoulou, C. I. & Pattichis, C. S. Unsupervided pattern recognition for the classification of EMG signals. *IEEE Trans. Biomed. Eng.* **46,** 169–178 (1999).
73. Texas Instruments. LMP770x Precision, CMOS Input, RRIO, Wide Supply Range Amplifiers. (2015).
74. Texas Instruments. ADS129x Low-Power , 8-Channel , 24-Bit Analog Front-End for Biopotential Measurements. (2015).
75. Atmel. ATmega328P - 8-bit AVR Microcontroller with 32K Bytes In-System Programmable Flash. (2015).
76. Blom, J. Sparkfun Bluetooth Basics. (2013). at <https://learn.sparkfun.com/tutorials/bluetooth-basics/all>
77. Texas Instruments. TPS7A7001 Very Low Input, Very Low Dropout, 2-Amp Regulator With Enable. (2017).
78. Linear Technology. LT1611 Inverting 1.4MHz Switching Regulator in SOT-23. (1998).
79. Atmel. ATmega640/V-1280/V-1281/V-2560/V-2561/V - 8-bit Atmel Microcontroller with 16/32/64KB In-System Programmable Flash. (2014).
80. Sharma, S., Saeed, A., Johnson, C., Gadegaard, N. & Cass, A. E. G. Rapid, low cost prototyping of transdermal devices for personal healthcare monitoring. *Sens. Bio-Sensing Res.* (2016). doi:10.1016/j.sbsr.2016.10.004
81. Cha, K. J., Kim, T., Park, S. J. & Kim, D. S. Simple and cost-effective fabrication of solid biodegradable polymer microneedle arrays with adjustable aspect ratio for transdermal drug delivery using acupuncture microneedles. *J. Micromechanics Microengineering* **24,** 115015 (2014).
82. Lee, H. *et al.* Wearable/disposable sweat-based glucose monitoring device with

multistage transdermal drug delivery module. *Sci. Adv.* **3,** e1601314 (2017).
83. Nishinaka, Y., Jun, R., Prihandana, G. S. & Miki, N. Fabrication of Polymer Microneedle Electrodes Coated with Nanoporous Parylene. *Jpn. J. Appl. Phys.* **52,** 06GL10 (2013).
84. Byun, D., Cho, S. J. & Kim, S. Fabrication of a flexible penetrating microelectrode array for use on curved surfaces of neural tissues. *J. Micromechanics Microengineering* **23,** 125010 (2013).
85. Hsu, L. S., Tung, S. W., Kuo, C. H. & Yang, Y. J. Developing barbed microtip-based electrode arrays for biopotential measurement. *Sensors* **14,** 12370–12386 (2014).
86. Guvanasen, G. S. *et al.* Development of a stretchable, penetrating electrode array for measuring intramuscular electromyographic activity. *17th Int. Conf. Miniaturized Syst. Chem. Life Sci. MicroTAS 2013* **1,** 539–541 (2013).
87. Mishra, R. K., A. M., V. M., Soto, F., Chrostowski, R. & Wang, J. Microneedle Biosensor for Minimally-Invasive Transdermal Detection of Nerve Agents. *Analyst* **142,** 918–924 (2017).
88. Kim, M., Kim, T., Kim, D. S. & Chung, W. K. Curved microneedle array-based sEMG electrode for robust long-term measurements and high selectivity. *Sensors* **15,** 16265–16280 (2015).
89. Rosell, J., Colominas, J., Riu, P., Pallas-Areny, R. & Webster, J. G. Skin Impedance From 1 Hz to 1 MHz. *IEEE Trans. Biomed. Eng.* **35,** 649–652 (1988).
90. Kim, Y.-C., Park, J.-H. & Prausnitz, M. R. Microneedles for drug and vaccine delivery. *Adv. Drug Deliv. Rev.* **64,** 1547–1568 (2012).
91. Cleary, G. W. Microneedles for drug delivery. *Pharm. Res.* **28,** 1–6 (2011).
92. Lee, J. W., Park, J. H. & Prausnitz, M. R. Dissolving microneedles for transdermal drug delivery. *Biomaterials* **29,** 2113–2124 (2008).
93. Facchinetti, A., Sparacino, G. & Cobelli, C. Reconstruction of glucose in plasma from interstitial fluid continuous glucose monitoring data: Role of sensor calibration. *J. Diabetes Sci. Technol.* **1,** 617–623 (2007).
94. Woderer, S. *et al.* Continuous glucose monitoring in interstitial fluid using glucose oxidase-based sensor compared to established blood glucose measurement in rats. *Anal. Chim. Acta* **581,** 7–12 (2007).
95. Gowers, S. A. N. *et al.* 3D Printed Microfluidic Device with Integrated Biosensors for Online Analysis of Subcutaneous Human Microdialysate. *Anal. Chem.* **87,** 7763–7770 (2015).
96. Sharma, S., Huang, Z., Rogers, M., Boutelle, M. & Cass, A. E. G. Evaluation of a minimally invasive glucose biosensor for continuous tissue monitoring. *Anal. Bioanal. Chem.* **408,** 8427–8435 (2016).
97. Bollella, P., Sharma, S., Cass, A. E. G. & Antiochia, R. Minimally-invasive Microneedle-based Biosensor Array for Simultaneous Lactate and Glucose Monitoring in Artificial Interstitial Fluid. *Electroanalysis* **31,** 374–382 (2019).
98. Sharma, S. *et al.* A pilot study in humans of microneedle sensor arrays for continuous glucose monitoring. *Anal. Methods* **10,** 2088–2095 (2018).
99. Gowers, S. A. N. *et al.* Development of a Minimally Invasive Microneedle-Based Sensor for Continuous Monitoring of β-Lactam Antibiotic Concentrations in Vivo. *ACS Sensors* **4,** 1072–1080 (2019).
100. Mukerjee, E. V., Collins, S. D., Isseroff, R. R. & Smith, R. L. Microneedle array for transdermal biological fluid extraction and in situ analysis. *Sensors Actuators, A Phys.* **114,** 267–275 (2004).
101. Bodhale, D. W., Nisar, A. & Afzulpurkar, N. Design, Fabrication and Analysis of Silicon Microneedles for Transdermal Drug Delivery Applications. in *IFMBE Proceedings* **27,** 84–89 (2010).
102. Li, Y. *et al.* In-plane silicon microneedles with open capillary microfluidic networks

by deep reactive ion etching and sacrificial layer based sharpening. *Sensors Actuators, A Phys.* **292,** 149–157 (2019).
103. Wilke, N., Mulcahy, A., Ye, S. R. & Morrissey, A. Process optimization and characterization of silicon microneedles fabricated by wet etch technology. *Microelectronics J.* **36,** 650–656 (2005).
104. Donnelly, R. F. *et al.* Optical coherence tomography is a valuable tool in the study of the effects of microneedle geometry on skin penetration characteristics and in-skin dissolution. *J. Control. Release* **147,** 333–341 (2010).
105. Zoski, C. G. & Mirkin, M. V. Steady-state limiting currents at finite conical microelectrodes. *Anal. Chem.* **74,** 1986–1992 (2002).
106. Rogers, M. L. A Biosensor Based Digital Microfluidic System For Neuroanalysis In The Intensive Care Unit. (Imperial College London, 2011).
107. Findlay, A. & Levitt, B. P. *Findlay's Practical Physical Chemistry*. (Longman Group Limited, 1973).
108. Rogers, M. L. *et al.* Simultaneous monitoring of potassium, glucose and lactate during spreading depolarization in the injured human brain - Proof of principle of a novel real-time neurochemical analysis system, continuous online microdialysis. *J. Cereb. Blood Flow Metab.* **37,** 1883–1895 (2017).
109. Vasylieva, N. *et al.* Covalent enzyme immobilization by poly(ethylene glycol) diglycidyl ether (PEGDE) for microelectrode biosensor preparation. *Biosens. Bioelectron.* **26,** 3993–4000 (2011).
110. Viggiano, A., Marinesco, S., Pain, F., Meiller, A. & Gurden, H. Reconstruction of field excitatory post-synaptic potentials in the dentate gyrus from amperometric biosensor signals. **206,** 1–6 (1997).
111. Samper, I. C. *et al.* Lab on a Chip temporal resolution in human brain. 2038–2048 (2019). doi:10.1039/c9lc00044e

APPENDIX A

Bits and Bytes

In digital systems, a *bit* is a single unit of information that can possess the value of 0 or 1. The stream of bits in digital communication is represented as two different voltages, for example 0 V for a low/0 value and 5 V for a high/1 value. Note that the voltage level may differ depending on the logic level of the system (3.3 V is another common logic level). A string of bits, represented as a series of 0s and 1s is known as *binary* notation. Multiple bits can be combined together to represent values other than 0 and 1. For example, the binary notation of '10' is used to represent the value 2, and '11' is used to represent the value 3, as shown in Table A-1.

Table A-1: Binary conversion of two-bit values

Binary	Decimal value	Binary	Decimal value
00	**0**	10	**2**
01	**1**	11	**3**

Larger numbers can be represented by using more bits. For example, the binary notation '111' is used to represent the value 7. Equation (A-1) explains how to calculate the maximum decimal value that can be represented by *n* number of bits. For example, the maximum value that 8 bits can represent is 255, or for 16 bits it is 65,535. A stream of bits can also be used to communicate a message or instructions, or for example signify which register or space in memory data should be written to or read from.

$$Value = 2^n - 1 \quad \textbf{(A-1)}$$

When working with digital systems that use very long strings of binary digits, it can be useful to use an abbreviated notation to reduce the risk of errors. *Decimal* notation can be useful (as shown in Table A-1 and Table A-2), however for values greater than 9 (e.g. 10) there is a risk it may be confused for binary notation. Another notation, and commonly used for long strings of binary digits, is *hexadecimal*. Hexadecimal arranges strings of binary digits into groups of four, and represents values greater than 9 with letters A-F (see Table A-2) For

example, a large 16-bit binary string such as '1111|1001|1010|0011' is grouped into 4-bit blocks and is notated as 'F9A3' in hexadecimal. The notation '0x' is often used preceding a hexadecimal value to let the reader know the notation of the value they are reading.

The combination of 8 bits is known as a *byte*. Units of memory storage are often referred to in bytes (e.g. kilobyte, megabyte). An '8-bit' microcontroller will only be able to transfer data (e.g. via SPI) one byte at a time (in 8-bit blocks). Therefore, for these types of system, long binary sequences will have to be broken up into byte-size blocks when transferring data, and reassembled after the transfer is complete. For example, an analogue-to-digital converter (ADC) that converts data at 24-bit resolution will have to transfer each digitised value as three byte-sized blocks when transferring data to an 8-bit system.

Table A-2: Four-bit binary conversion to decimal and hexadecimal. *Hexadecimal stores values greater than 9 as letters to help reduce notation errors and confusion.*

Binary	Value		Binary	Value	
	Dec	Hex		Dec	Hex
0000	**0**	**0**	1000	**8**	**8**
0001	**1**	**1**	1001	**9**	**9**
0010	**2**	**2**	1010	**10**	**A**
0011	**3**	**3**	1011	**11**	**B**
0100	**4**	**4**	1100	**12**	**C**
0101	**5**	**5**	1101	**13**	**D**
0110	**6**	**6**	1110	**14**	**E**
0111	**7**	**7**	1111	**15**	**F**

Most Significant Bit/Least Significant Bit

In a binary value, the most significant bit (MSB) is the bit that holds the largest value, and is usually the left-most bit in the binary notation. The least significant bit (LSB) is the bit that holds the smallest value, and is often the right-most bit in binary notation. For example, in the 8-bit example '10011010' shown in Figure A-1, the MSB of the binary value is 1 and the LSB is 0. When data are transferred,

e.g. across SPI, the sequence of binary values can be transferred 'MSB-first' (binary sequence transferred from left to right) or transferred 'LSB-first' (binary sequence transferred from right to left). The order the binary sequence is transferred in is important to ensure data are read correctly. Data transfer is commonly conducted 'MSB-first'.

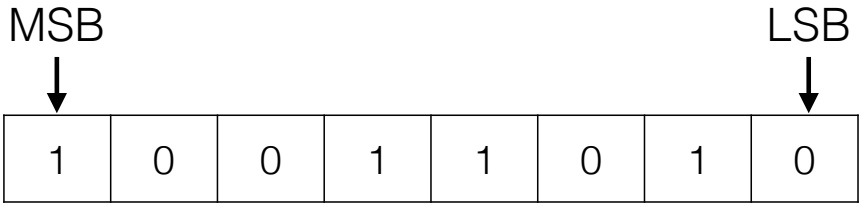


Figure A-1: MSB and LSB of an 8-bit value. *MSB in this example is the left-most bit, 1. LSB in this example is the right-most bit, 0.*

Two's complement

Two's complement is a form of binary notation used to represent signed values i.e. positive/negative values. This is often used when converting analogue signals where the analogue-to-digital converter (ADC) has a range that spans from positive to negative values. When using two's complement, the MSB is used to denote whether or not the value is positive or negative. This will mean that the greatest value that *n* number of bits can represent will be a factor of two smaller than if it were if it were unsigned binary notation. For example, a 16-bit ADC using two's complement will actually only be able to operate at 15-bit precision, but will be able to operate using a full 15-bit negative range as well. The maximum positive decimal value represented by *n* bits in two's complement is shown in Equation A-2.

$$Value = 2^{n-1} - 1 \quad \textbf{(A-2)}$$

As seen in the examples in Table A-3, the largest positive 4-bit binary value in two's complement is '0111', holding the decimal value of 7 and represented as 0x7 in hexadecimal. The largest negative value 4-bit binary value in two's complement is '1000', holding the decimal value of -8 and represented as 0x8 in hexadecimal. The smallest positive 4-bit binary value in two's complement is '0001', holding the decimal value of 1 and represented as 0x1 in hexadecimal. The smallest negative value 4-bit binary value in two's complement is '1111',

holding the decimal value of -1 and represented as 0xF in hexadecimal. Zero is represented as '0000' in 4-bit binary and represented as 0x0 in hexadecimal.

Table A-3: Two's complement demonstration. *Decimal and hexadecimal values of 4-bit values stored used two's complement notation.*

Binary	Value		Binary	Value	
	Dec	Hex		Dec	Hex
0000	**0**	**0**	1000	**-8**	**8**
0001	**1**	**1**	1001	**-7**	**9**
0010	**2**	**2**	1010	**-6**	**A**
0011	**3**	**3**	1011	**-5**	**B**
0100	**4**	**4**	1100	**-4**	**C**
0101	**5**	**5**	1101	**-3**	**D**
0110	**6**	**6**	1110	**-2**	**E**
0111	**7**	**7**	1111	**-1**	**F**

It is vital to check whether the devices within a system use two's complement binary format, and if there is any conflict, otherwise data will not be transferred as intended.

APPENDIX B

Serial Peripheral Interface

Serial Peripheral Interface (SPI) is a method of short distance communication between microcontrollers and other peripherals (e.g. analogue-to-digital converters (ADCs), sensors, memory units), typically within an embedded system. SPI uses a 'master-slave' architecture i.e. the microcontroller (master) controls the functions of the peripheral components (slaves). SPI is often used for the transfer and retrieval of data, and for the configuration or 'set-up' of peripheral devices within the embedded system. The interface utilises a communication bus consisting of four communication lines: master-out-slave-in (MOSI), master-in-slave-out (MISO), serial clock (SCLK), and slave/chip select (SS/CS) (see top of Figure B-1). Each bit of information being sent to or from the master is dependant on the clock signal in the SCLK line; each bit will be sent on each clock pulse from the master. For this reason, data transmission over SPI is often referred to as being 'clocked in' or 'clocked out'. SPI operates in 'full-duplex' mode, meaning data can be transmitted in both directions at the same time using the same carrier signal (SCLK signal).

In order to write to or retrieve data from the slave device, it must be selected by the master. To select the slave device, its CS line must be 'brought low' by the master (brought from a 'high' logic level to a 'low' logic level). This activates the slave device, opening its SPI port for communication. A master can take control of multiple slave devices by using multiple CS lines, activating each one in turn (see bottom of Figure B-1); if a slave's CS line is not selected, any SPI communication on the remaining three lines in the interface bus will be ignored by it.

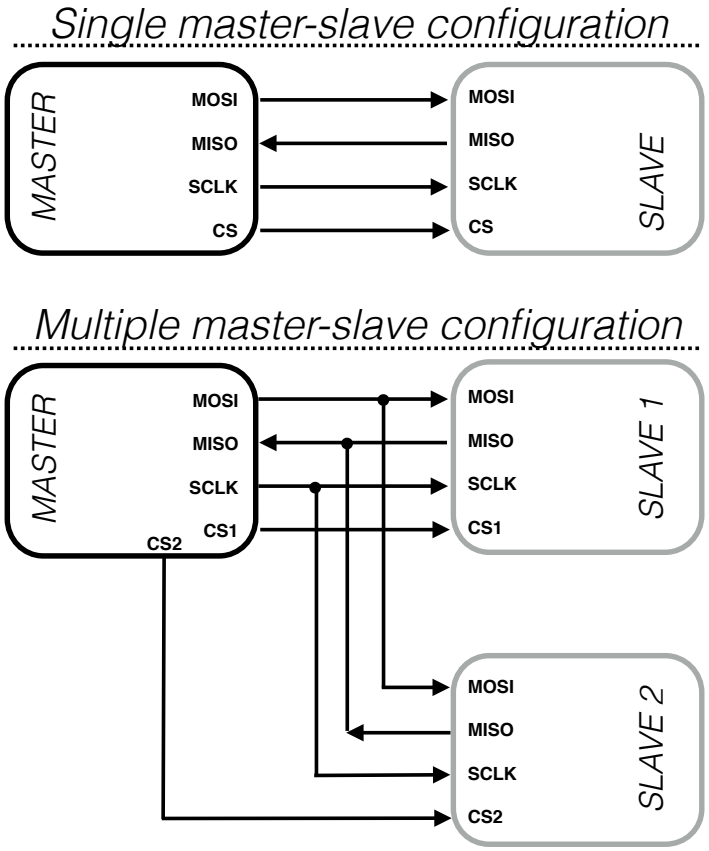


Figure B-1: The Serial Peripheral Interface (SPI) consists of a four-line communication bus: MOSI, MISO, SCLK, and CS. *Multiple CS lines can be utilised for the master device to control multiple slave peripheral devices; the remaining three lines in the SPI bus (MISO, MOSI, SCLK) are shared between salve devices. The slave device that the master has control of is determined by the CS line that is selected at that time. If the CS is not selected, the master device cannot communicate with that slave.*

Different systems will have different SPI modes which needs to be correctly configured. The SPI mode is determined by the polarity of the clock signal (if the clock signal moves from positive to negative or from negative to positive) and the clock phase (whether or not data is clocked on the leading edge or trailing edge of the clock pulse) (see Figure B-2 and Table B-1). If the master and slave devices are not configured to the same SPI mode, data will be read and written incorrectly.

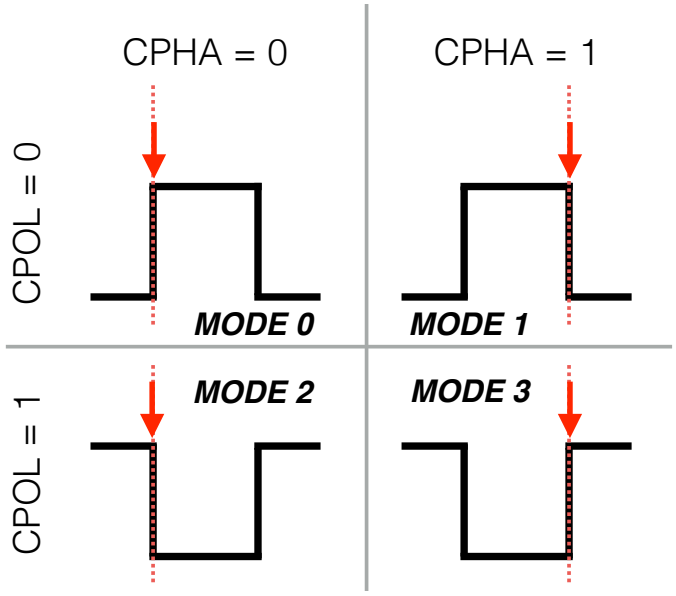


Figure B-2: Schematic of different SPI modes. *The clock polarity (CPOL) and the edge of the pulse where data is read (CPHA) determine the SPI mode of the system. The red arrows indicate the edge of the clock pulse where data is read.*

Table B-1: Different SPI modes. *The SPI mode of the system will depend on polarity of the serial clock (CPOL) and whether data is read on the leading or trailing edge of the clock pulse (CPHA).*

SPI mode	CPOL	CPHA
0	0	0
1	0	1
2	1	0
3	1	1

APPENDIX C

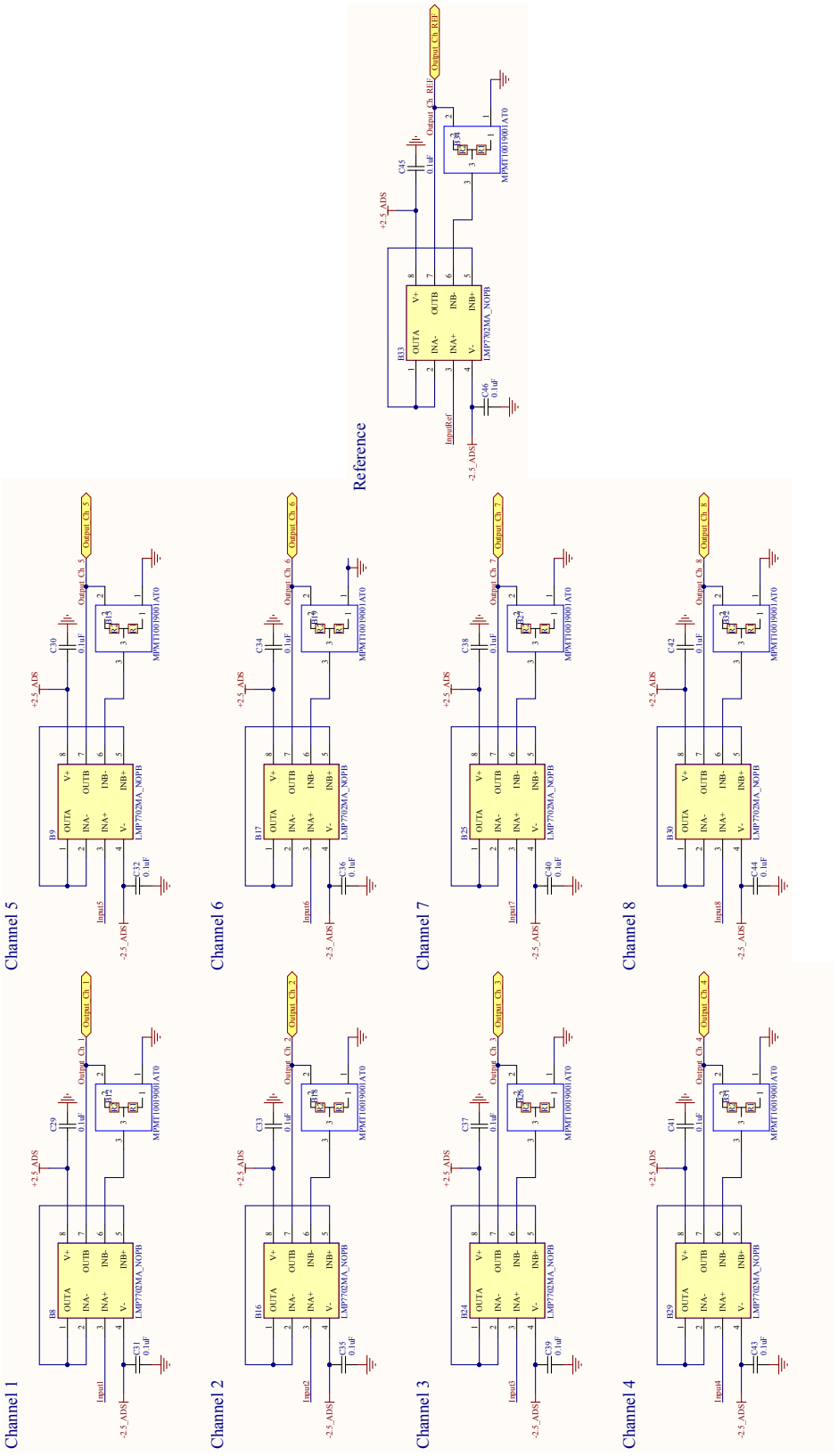


Figure C-1: Analogue front end (AFE) schematic.

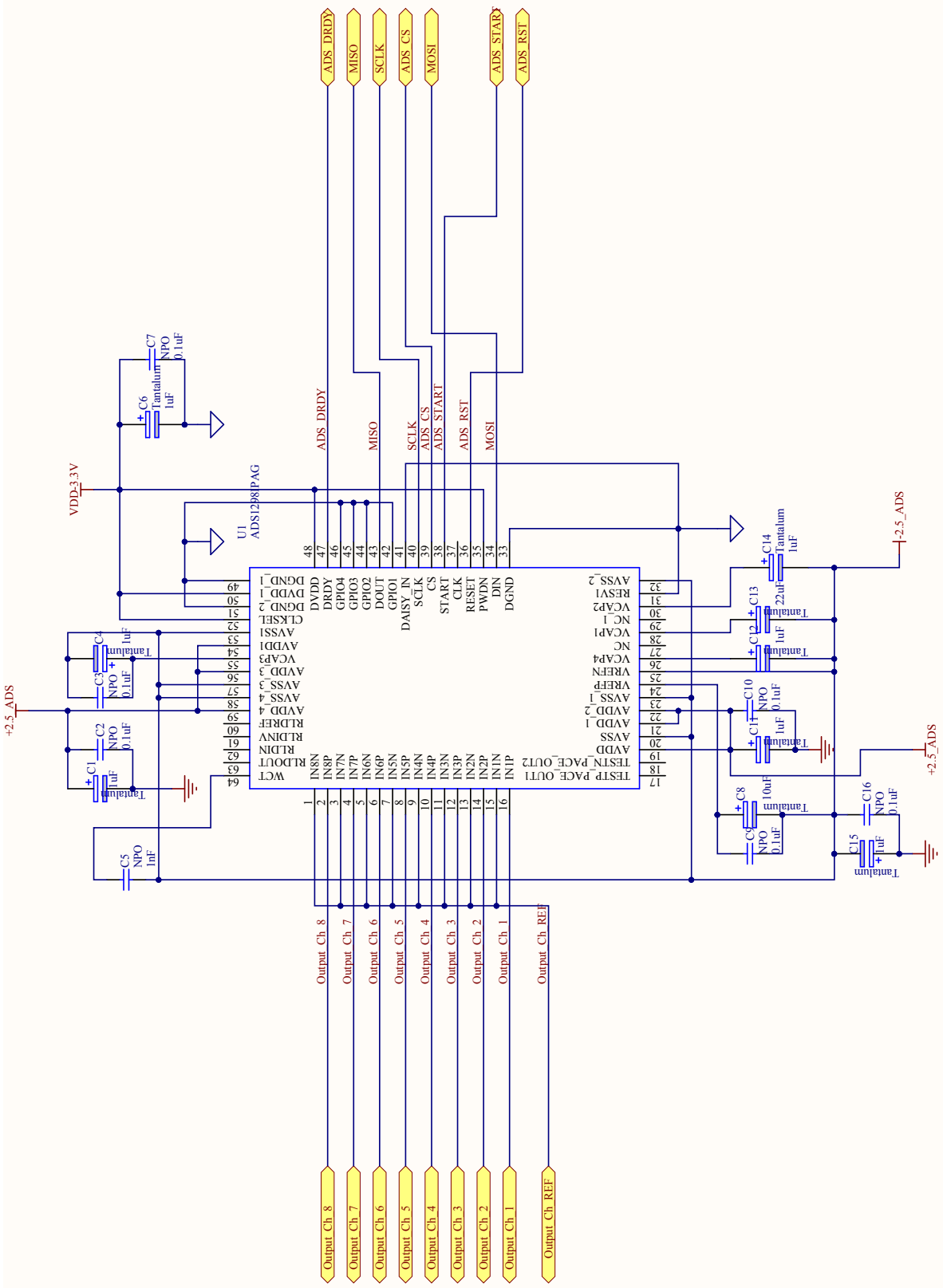


Figure C-2: ADS1298 breakout schematic.

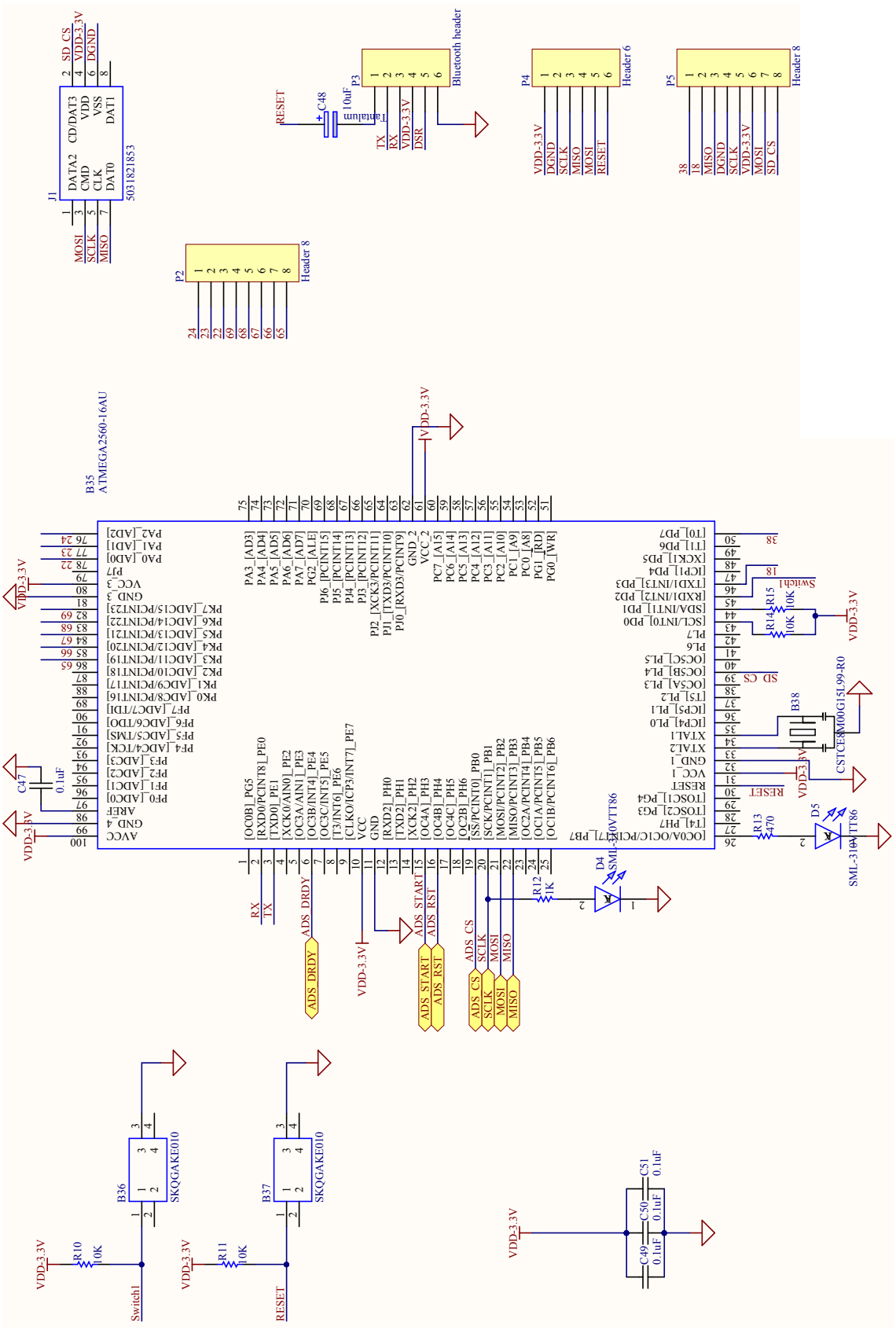


Figure C-3: ATMEGA2560 breakout schematic.

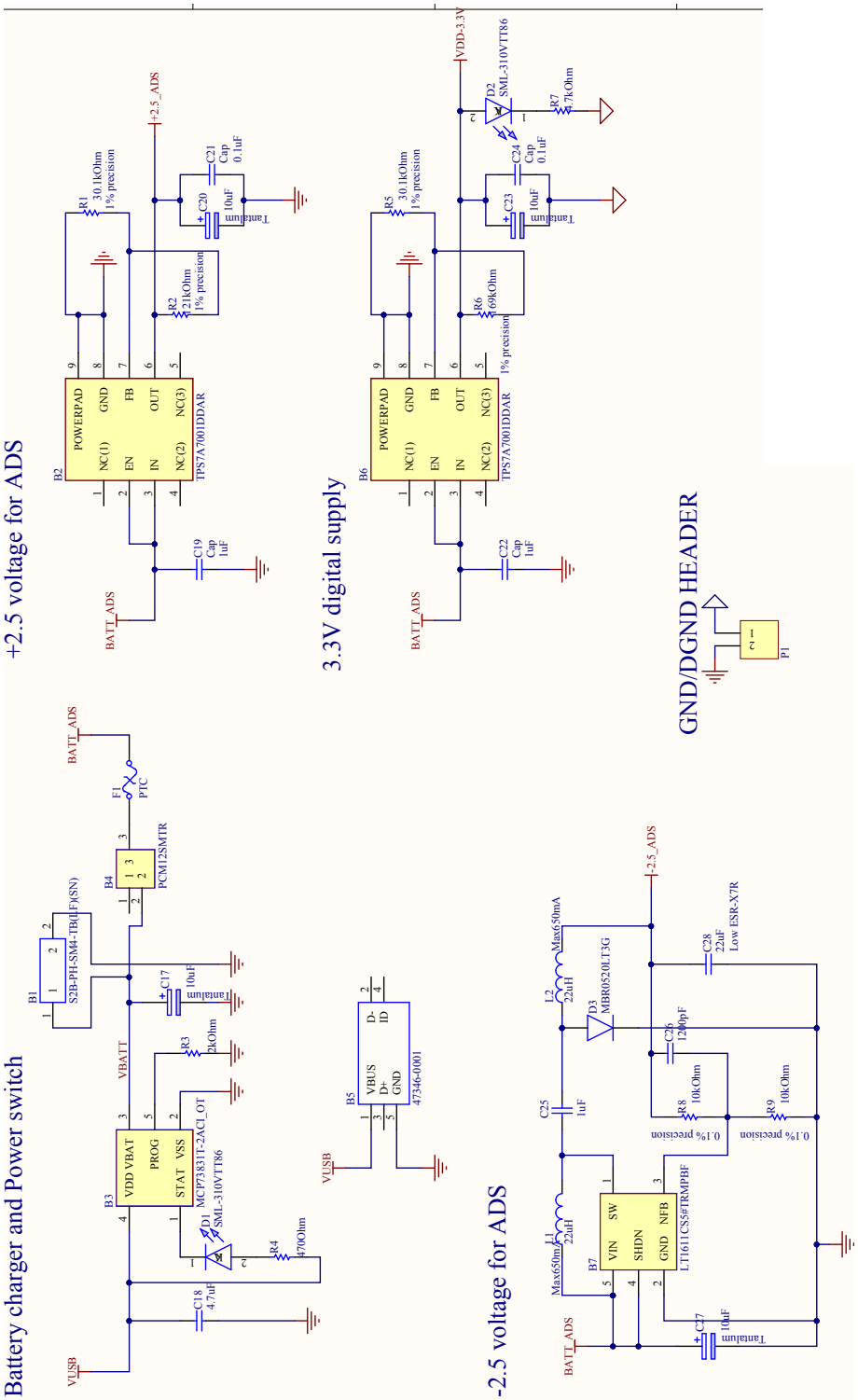


Figure C-4: Power source and voltage regulation schematic.

PERMISSIONS

Page No.	Type of work:	Name of work	Source of work	Copyright holder and contact	Permission requested on	I have permission yes /no	Permission note
37	Figure	Figure 1 Electromyography: Fasciculation potentials (FPs)…	Fasciculation potentials and earliest changes in motor unit physiology in ALS – J Neurol Neurosurg Psychiatry (2013), vol 84 (963-968)	© BMJ Publishing Group Limited	12.06.2019	Yes	Permission granted 12.06.19 via RightsLink. Order number: 4606440973638
39	Figure	Fig. 6. A motor unit potential fingerprint….	High-density Surface EMG: Techniques and Applications at a Motor Unit Level - Biocybernetics and Biomedical Engineering (2012), vol 32 (3-27)	Copyright © 2012 Nałęcz Institute of Biocybernetics and Biomedical Engineering. Published by Elsevier Urban & Partner Sp. z o.o.	12.06.2019	Yes	Permission granted 12.06.19 via RightsLink. Order number: 4606451221185
40	Figure	Figure 1 The electrode arrays used for FP recordings…	Duration of observation required in detecting fasciculation potentials in amyotrophic lateral sclerosis using high-density surface EMG – J Neuroeng Rehab 2012, vol 9, 78	© Zhou et al.; licensee BioMed Central Ltd. 2012	12.06.2019	Yes	Licensed with a Creative Commons Attribution-NonCommercial-NoDerivs 2.0 Generic (CC BY-NC-ND 2.0
50	Image (website)	Arduino1: Arduino Uno (Rev3)	Fritzing.org	Credit: Fritzing.org	12.06.2019	Yes	Licensed with a Creative Commons Attribution-ShareAlike 3.0 Unported (CC BY-SA 3.0)

Page No.	Type of work:	Name of work	Source of work	Copyright holder and contact	Permission requested on	I have permission yes /no	Permission note
170	Image	LMP7702 D or DGK Package 8-Pin SOIC or VSSOP Top View	LMP770x Precision, CMOS Input, RRIO, Wide Supply Range Amplifiers (*LMP7702, datasheet*)	Copyright © 2019, Texas Instruments Incorporated	30.07.2019	Yes	"TI grants you permission to use these resources only for development of an application that uses the TI products described in the resource".
173	Image (schematic)	Pag Package TQFP-64 (top view)	Low-Power, 8-Channel, 24-Bit Analog Front-End for Biopotential Measurements (*ADS1298 datasheet*)	Copyright © 2014, Texas Instruments Incorporated	30.07.2019	Yes	Permission granted if reproduction is without alteration
174	Figure	Figure 51. PGA Implementation	Low-Power, 8-Channel, 24-Bit Analog Front-End for Biopotential Measurements (*ADS1298 datasheet*)	Copyright © 2014, Texas Instruments Incorporated	30.07.2019	Yes	Permission granted if reproduction is without alteration
179	Figure	Figure 30. WREG Command Example: Write Two Registers Starting from 00h…	Low-Power, 8-Channel, 24-Bit Analog Front-End for Biopotential Measurements (*ADS1298 datasheet*)	Copyright © 2014, Texas Instruments Incorporated	30.07.2019	Yes	Permission granted if reproduction is without alteration
182	Figure	Figure 26. SPI Bus Data Output for the ADS1298 (Eight Channels)	Low-Power, 8-Channel, 24-Bit Analog Front-End for Biopotential Measurements (*ADS1298 datasheet*)	Copyright © 2014, Texas Instruments Incorporated	30.07.2019	Yes	Permission granted if reproduction is without alteration of information

Page No.	Type of work:	Name of work	Source of work	Copyright holder and contact	Permission requested on	I have permission yes /no	Permission note
183, 184	Figure	Figure 1-1. Pinout	8-bit AVR Microcontroller with 32K Bytes In-System Programmable Flash (*ATmega328p, datasheet)*	Copyright © 2015 Atmel Corporation	30.07.2019	Yes	"Fair-use" exception of the U.S. Copyright Act of 1976
185	Figure	Figure 28-1. Maximum Frequency	8-bit AVR Microcontroller with 32K Bytes In-System Programmable Flash (*ATmega328p, datasheet)*	Copyright © 2015 Atmel Corporation	30.07.2019	Yes	"Fair-use" exception of the U.S. Copyright Act of 1976
190	Image	Typical Application	Very Low Input, Very Low Dropout, 2-Amp Regulator With Enable (*TPS7A7001, datasheet*)	Copyright © 2019, Texas Instruments Incorporated	30.07.2019	Yes	"TI grants you permission to use these resources only for development of an application that uses the TI products described in the resource".
203	Figure	Figure 1-1. TQFP-pinout ATmega640/1280/2560	8-bit Atmel Microcontroller with 16/32/64KB In-System Programmable Flash (*ATmega2560, datasheet)*	Copyright © 2014 Atmel Corporation	30.07.2019	Yes	"Fair-use" exception of the U.S. Copyright Act of 1976
232	Figure	Figure 1. (a) Schematic picture of the fabrication process "shrink cycles"…	Improving the Resolution of 3D-Printed Molds for Microfluidics by Iterative Casting-Shrinkage Cycles – Anal Chem (2017), vol 89 (2227-2231)	Copyright © 2017, American Chemical Society	25.06.2019	Yes	Permission granted 25.06.19 given appropriate credit for requested material is provided

Page No.	Type of work:	Name of work	Source of work	Copyright holder and contact	Permission requested on	I have permission yes /no	Permission note
254	Figure	Figure 3.1: Hydrogen peroxide calibration on platinum and gold electrodes 0.5 mM, 1 mM, 2.5 mM, ...	A Biosensor Based Digital Microfluidic System For Neuroanalysis In The Intensive Care Unit (M. Rogers, 2011. *Thesis)*	Michelle Louise Rogers, Bioengineering, Imperial College London	30.07.2019	Yes	Written permission given 30/07/2019