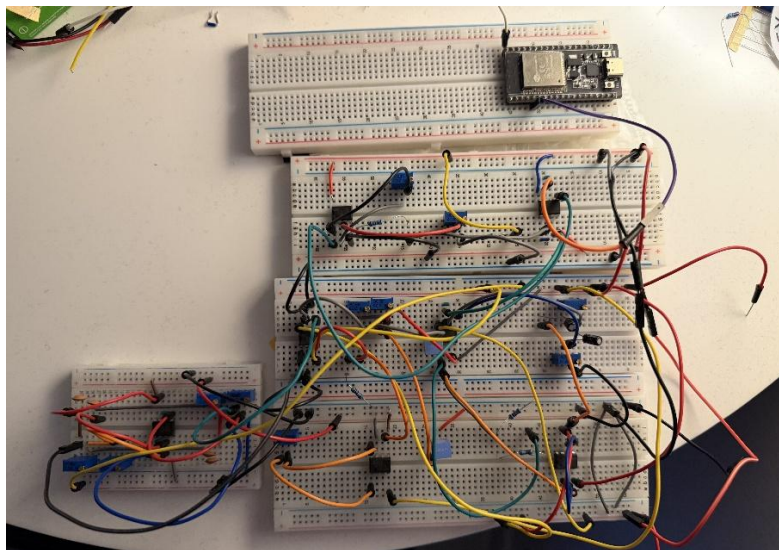


ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ
ΚΑΙ ΗΛΕΚΤΡΟΝΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

«Χειρισμός διάταξης με ηλεκτρομυογραφικά σήματα
χεριού»



Του φοιτητή
Νεκτάριου Τοπαλίδη
Αρ. Μητρώου: 2020253

Επιβλέπων
Δρ. Κιοσκερίδης Ιορδάνης
Βαθμίδα: Καθηγητής

Μάιος 2026

Χειρισμός διάταξης με ηλεκτρομυογραφικά σήματα χεριού

Κωδικός: 26106

Φοιτητής: Νεκτάριος Τοπαλίδης

Εισηγητής: Δρ. Ιορδάνης Κιοσκερίδης

Ημερομηνία ανάληψης: 16-01-2026

Ημερομηνία περάτωσης: 14-05-2026

Βεβαιώνω ότι είμαι ο συγγραφέας αυτής της εργασίας και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία της είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης, έχω καταγράψει τις όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών, εικόνων και κειμένου, είτε αυτές αναφέρονται ακριβώς είτε παραφρασμένες. Επιπλέον, βεβαιώνω ότι αυτή η εργασία προετοιμάστηκε από εμένα προσωπικά, ειδικά ως διπλωματική εργασία, στο Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του ΔΙ.Π.Α.Ε.

Η παρούσα εργασία αποτελεί πνευματική ιδιοκτησία του φοιτητή Νεκταρίου Τοπαλίδη που την εκπόνησε. Στο πλαίσιο της πολιτικής ανοικτής πρόσβασης, ο συγγραφέας/δημιουργός εκχωρεί στο Διεθνές Πανεπιστήμιο της Ελλάδος άδεια χρήσης του δικαιώματος αναπαραγωγής, δανεισμού, παρουσίασης στο κοινό και ψηφιακής διάχυσης της εργασίας διεθνώς, σε ηλεκτρονική μορφή και σε οποιοδήποτε μέσο, για διδακτικούς και ερευνητικούς σκοπούς, άνευ ανταλλάγματος. Η ανοικτή πρόσβαση στο πλήρες κείμενο της εργασίας, δεν σημαίνει καθ' οιονδήποτε τρόπο παραχώρηση δικαιωμάτων διανοητικής ιδιοκτησίας του συγγραφέα/δημιουργού, ούτε επιτρέπει την αναπαραγωγή, αναδημοσίευση, αντιγραφή, πώληση, εμπορική χρήση, διανομή, έκδοση, μεταφόρτωση (downloading), ανάρτηση (uploading), μετάφραση, τροποποίηση με οποιονδήποτε τρόπο, τμηματικά ή περιληπτικά της εργασίας, χωρίς τη ρητή προηγούμενη έγγραφη συναίνεση του συγγραφέα/δημιουργού.

Η έγκριση της διπλωματικής εργασίας από το Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του Διεθνούς Πανεπιστημίου της Ελλάδος, δεν υποδηλώνει απαραίτητως και αποδοχή των απόψεων του συγγραφέα, εκ μέρους του Τμήματος.

Στον πατέρα μου,
Άγαθὰ κόποις κτῶνται

Περίληψη

Ενώ αρκετές από τις προηγούμενες έρευνες έχουν αναλύσει μεμονωμένα στοιχεία του πεδίου της ηλεκτρομυογραφίας, η παρούσα εργασία αποσκοπεί στην διερεύνηση και υλοποίηση ενός πλήρους συστήματος για την καταγραφή και την κατηγοριοποίηση EMG σημάτων. Καθώς αυτό το επιστημονικό πεδίο γίνεται ολοένα και πιο γνωστό, με την εφαρμογή σε καταναλωτικές συσκευές, η σχεδίαση ενός λειτουργικού συστήματος με χαμηλό κόστος είναι απαραίτητη. Η συγκεκριμένη υλοποίηση αποτελεί μία ολοκληρωμένη προσέγγιση, και οι μέθοδοι που αξιοποιήθηκαν για την ανάπτυξή της είναι εξαιρετικά ενδιαφέρουσες. Αρχικά, η προσομοίωση τόσο του υλικού μέρους του συστήματος όσο και του μηχανισμού γέννησης ηλεκτρικών σημάτων από τους μύες έπαιξε καθοριστικό ρόλο στην μετέπειτα κατασκευή του. Τα φίλτρα που χρησιμοποιήθηκαν για την αναλογική επεξεργασία του σήματος ελέγχθηκαν εκτενώς και η ορθή λειτουργία τους επιβεβαιώθηκε με την χρήση φθηνού εργαστηριακού εξοπλισμού. Μέσω του συστήματος καταγράφηκαν ηλεκτρομυογραφικά σήματα τα οποία μετά την επεξεργασία τους χρησιμοποιήθηκαν για την εκπαίδευση ενός μοντέλου μηχανικής μάθησης. Τα χαρακτηριστικά του μοντέλου πέρα από μαθηματικές εξισώσεις που λαμβάνουν χώρα στο πεδίο του χρόνου, συμπεριλαμβάνουν και κάποια που βρίσκονται στο πεδίο της συχνότητας, κάτι που δεν παρατηρήθηκε σε αντίστοιχες έρευνες. Ακόμα, έγινε έλεγχος της συσχέτισης των χαρακτηριστικών για την επιλογή του πιο αποδοτικού συνδυασμού τους. Τέλος, το μοντέλο μεταγλωττίστηκε σε γλώσσα C για να μπορεί να μεταφορτωθεί σε έναν μικροελεγκτή ESP32, με σκοπό την κατηγοριοποίηση των σημάτων σε πραγματικό χρόνο, αξιοποιώντας έτσι τεχνολογίες Edge AI. Αν και η απόδοση του συστήματος χρήζει βελτίωσης, η κατασκευή του συγκεκριμένου πρωτοτύπου αποτελεί απόδειξη πως είναι δυνατή η υλοποίηση τέτοιων συστημάτων σε πολύ χαμηλό κόστος. Πιο συγκεκριμένα, η κατασκευή κόστισε συνολικά 38.5€-48€, η οποία είναι μια τιμή σχεδόν τρεις φορές μικρότερη από τα αντίστοιχα προϊόντα που βρίσκονται αυτή την στιγμή στην αγορά.

«Control of a device via hand electromyographic signals»

Nektarios Topalidis

Abstract

While several previous studies have analyzed individual elements of the electromyography field, the present thesis aims to investigate and implement a complete system for the recording and classification of EMG signals. As this scientific field becomes increasingly prominent through applications in consumer devices, the design of a functional, low-cost system is essential. This implementation represents an integrated approach, utilizing highly interesting developmental methods. Initially, the simulation of both the hardware components and the muscle electrical signal generation mechanism played a decisive role in the subsequent construction of the system. The filters used for analog signal processing were extensively tested, and their correct operation was verified using affordable laboratory equipment. Through this system, electromyographic signals were recorded and, following processing, were used to train a machine learning model. The model's features, beyond time-domain mathematical equations, include frequency-domain data, providing additional depth to the analysis. Furthermore, a feature correlation analysis was conducted to select the most efficient combination. Finally, the model was compiled into C code to be deployed on an ESP32 microcontroller, enabling real-time signal classification through Edge AI technologies. Although the system's performance requires further refinement, the construction of this prototype proves that implementing such systems at a very low cost is feasible. Specifically, the total construction cost ranged from 38.5€ to 48€, a price approximately three times lower than comparable products currently available on the market.

Ευχαριστίες

Θέλω να ευχαριστήσω τόσο την οικογένειά μου και τους φίλους όσο και τον καθηγητή μου κ. Κιοσκερίδη, αρχικά για την πολύτιμη καθοδήγηση που μου έδωσαν καθόλα τα χρόνια της φοίτησής μου, και εν συνεχεία που πίστεψαν σε εμένα και μου έδωσαν την δυνατότητα να τους κάνω περήφανους.

Περιεχόμενα

Περίληψη.....	iv
Abstract	v
Ευχαριστίες	vi
Περιεχόμενα	vii
Κατάλογος Σχημάτων	ix
Κατάλογος Πινάκων.....	ix
Συνομογραφίες.....	xi
Κεφάλαιο 1ο: Εισαγωγή	1
1.1 Εισαγωγή.....	1
1.2 Σύγχρονες Εφαρμογές και Τεχνολογικές Τάσεις	2
1.2.1 Ιατρικές και κλινικές εφαρμογές	2
1.2.2 Αλληλεπίδραση Ανθρώπου-Υπολογιστή (HCI) και Καταναλωτικά Προϊόντα	2
1.2.3 Αθλητική Επιστήμη και Εργονομία.....	3
1.2.4 Τεχνολογικές Τάσεις και Μηχανική Μάθηση.....	3
1.3 Σκοπός και Στόχοι της Εργασίας.....	4
1.4 Δομή της Διπλωματικής Εργασίας.....	6
1.5 Επίλογος.....	6
Κεφάλαιο 2ο: Ανάλυση Προσομοίωσης και Βιολογικού Σήματος	7
2.1 Εισαγωγή.....	7
2.2 Μηχανισμός Γέννησης Μυϊκών Σημάτων.....	7
2.2.1 Κινητικές Μονάδες (Motor Units).....	7
2.2.2 Δυναμικό Δράσης (Action Potential)	8
2.2.3 Δυναμικό Δράσης Κινητικής Μονάδας (Motor Unit Action Potential).....	9
2.2.4 Ανισοτροπία των μυϊκών ινών	11
2.3 Μαθηματικά Χαρακτηριστικά των Σημάτων EMG.....	11
2.4 Προκλήσεις Καταγραφής και Προεπεξεργασία Σήματος	13
2.5 Ανατομία και Φυσιολογία του Flexor Digitorum Superficialis	15
2.6 Επίλογος.....	16
Κεφάλαιο 3ο: Σχεδιασμός Συστήματος και Τεχνολογίες Υλοποίησης.....	17
3.1 Εισαγωγή.....	17
3.2 Γλώσσα Προγραμματισμού Python και Χρησιμοποιούμενες Βιβλιοθήκες	17
3.2.1 Βιβλιοθήκη NumPy	17

3.2.2	Βιβλιοθήκη Pandas	18
3.2.3	Βιβλιοθήκη Matplotlib	18
3.2.4	Βιβλιοθήκη PySpice	20
3.2.5	Βιβλιοθήκη Scikit-Learn	22
3.3	Εργαλεία Μέτρησης: Picoscope και το λογισμικό Picoscope 7	23
3.4	Ανάλυση Αναλογικών Φίλτρων και Ενίσχυση του Σήματος	24
3.4.1	Υψηλεπερατό Φίλτρο (Highpass Filter).....	25
3.4.2	Χαμηλοπερατό Φίλτρο (Lowpass Filter).....	27
3.4.3	Φίλτρο Εγκοπής (Notch filter).....	28
3.4.4	Βαθμίδες Ενίσχυσης	30
3.5	Μικροελεγκτής ESP32 και Στάδιο Προσαρμογής Σήματος (Level Shifting)	31
3.6	Επίλογος	33
Κεφάλαιο 4ο:	Μοντέλο Μηχανικής Μάθησης και Επεξεργασία Δεδομένων	34
4.1	Εισαγωγή	34
4.2	Εισαγωγή στη Μηχανική Μάθηση.....	34
4.2.1	Ιστορική Αναδρομή	34
4.2.2	Εξαγωγή Χαρακτηριστικών και Ανάλυση Συσχέτισης	36
4.2.3	Βασική θεωρία Μηχανικής Μάθησης	39
4.2.4	Εφαρμογές Μηχανικής Μάθησης σε Βιολογικά Δεδομένα	42
4.3	Συλλογή και Προεπεξεργασία Δεδομένων (Data Transformation).....	43
4.4	Αλγόριθμος Τυχαίων Δασών (Random Forests) και XGBoost.....	46
4.4.1	Μετατροπή Μοντέλου για Edge AI και Χρήση της Βιβλιοθήκης Treelite	48
4.5	Σύγκριση Αποτελεσμάτων και Αξιολόγηση Μοντέλου	50
4.6	Επίλογος	54
Κεφάλαιο 5ο:	Συμπεράσματα και Προτάσεις Βελτίωσης.....	55
5.1	Εισαγωγή	55
5.2	Σύνοψη Παρατηρήσεων	55
5.3	Προτάσεις για Μελλοντικές Βελτιώσεις	57
5.4	Εμπορικές Προοπτικές και Καινοτομία	59
5.5	Επίλογος	61
	ΒΙΒΛΙΟΓΡΑΦΙΑ.....	62
	ΠΑΡΑΡΤΗΜΑ Α : ΚΩΔΙΚΑΣ ΠΡΟΣΟΜΟΙΩΣΗΣ.....	66
	ΠΑΡΑΡΤΗΜΑ Β : ΚΩΔΙΚΑΣ ΕΠΕΞΕΡΓΑΣΙΑΣ ΔΕΔΟΜΕΝΩΝ	79
	ΠΑΡΑΡΤΗΜΑ C : ΚΩΔΙΚΑΣ ΜΗΧΑΝΙΚΗΣ ΜΑΘΗΣΗΣ	85

Κατάλογος Σχημάτων

Σχήμα 1.1 Γεωμετρία του μυός Flexor Digitorum Superficialis	5
Σχήμα 2.1 Σύσπαση μυϊκής ίνας	8
Σχήμα 2.2 Φυσιολογία κινητικής μονάδας.....	8
Σχήμα 2.3 Ριπή κατά την σύσπαση εκπόλωση και επαναπόλωση της μυϊκής μεμβράνης	9
Σχήμα 2.4 Ζώνη αποπόλωσης της μυϊκής μεμβράνης	9
Σχήμα 2.5 Υπέρθωση των σημάτων δυνακού δράσης	10
Σχήμα 2.6 Ενεργοποίηση των κινητικών μονάδων	10
Σχήμα 2.7 Διαφορά δυναμικού κατά μήκος των μυϊκών ινών	12
Σχήμα 2.8 Πυκνότητα ρεύματος κατά μήκος των μυϊκών ινών	12
Σχήμα 2.9 Τύποι ηλεκτροδίων	14
Σχήμα 2.10 Απόσβεση ανεπιθύμητων συχνοτήτων σε φίλτρα δευτέρου βαθμού.....	15
Σχήμα 2.11 Μήκος του FDS	15
Σχήμα 3.1 Παράδειγμα γραφήματος από Matplotlib	19
Σχήμα 3.2 Picoscope 2204A	23
Σχήμα 3.3 Λογισμικό Picoscope 7	24
Σχήμα 3.4 Τελικό σχηματικό του κυκλώματος.....	25
Σχήμα 3.5 Υψηπερατό φίλτρο δευτέρου βαθμού.....	25
Σχήμα 3.6 Απόσβεση σήματος σε φίλτρα δευτέρου βαθμού	26
Σχήμα 3.7 Καταγραφή EMG σήματος με bandpass και notch φίλτρα.....	27
Σχήμα 3.8 Χαμηλοπερατό φίλτρο δευτέρου βαθμού	27
Σχήμα 3.9 Φίλτρο εγκοπής με τοπολογία Twin-T	28
Σχήμα 3.10 Λειτουργία του υλοποιημένου φίλτρου εγκοπής	29
Σχήμα 3.11 Σχηματικό του INA128.....	30
Σχήμα 3.12 Μικροελεγκτής ESP32.....	31
Σχήμα 4.1 Γραφική αναπαράσταση ενός τεχνητού νευρώνα.....	34
Σχήμα 4.2 Γραφική αναπαράσταση του CART	35
Σχήμα 4.3 Μέθοδοι εξαγωγής χαρακτηριστικών	36
Σχήμα 4.4 Correlation heatmap των επιλεγμένων χαρακτηριστικών.....	39
Σχήμα 4.5 Διαδικασία ανάπτυξης εφαρμογών στην Μηχανική Μάθηση	40
Σχήμα 4.6 Γραφική αναπαράσταση του sliding window	41
Σχήμα 4.7 Αποτελέσματα δοκιμής του μοντέλου	51
Σχήμα 4.8 Κατηγοριοποίηση κινήσεων σε πραγματικό χρόνο	54
Σχήμα 5.1 Myo Armband.....	56
Σχήμα 5.2 Μικροελεγκτής nRF52840	58
Σχήμα 5.3 Εκπαίδευση κεντρικού μοντέλου μηχανικής μάθησης	60

Κατάλογος Πινάκων

Πίνακας 2.1 Γεωμετρικά χαρακτηριστικά του μυός Flexor Digitorum Superficialis	16
Πίνακας 4.1 Μαθηματικές εξισώσεις των επιλεγμένων χαρακτηριστικών	38
Πίνακας 4.2 Μορφή δεδομένων του πεδίου του χρόνου.....	43

Πίνακας 4.3 Μορφή δεδομένων του πεδίου της συχνότητας.....	43
Πίνακας 4.4 Μορφή δεδομένων μετά τον μετασχηματισμό	45
Πίνακας 4.5 Τιμές δεικτών του μοντέλου	53
Πίνακας 5.1 Ανάλυση κόστους του συστήματος	55

Συντομογραφίες

Δ.Ε.	Διπλωματική Εργασία
ΔΙΠΙΑΕ	Διεθνές Πανεπιστήμιο Ελλάδος
EMG	Electromyography
sEMG	Surface Electromyography
ADC	Analog to Digital Converter
AI	Artificial Intelligence
API	Application Programming Interface
HCI	Human-Computer Interaction
MU	Motor Unit
AP	Action Potential
MUAP	Motor Unit Action Potential
FFT	Fast Fourier Transform
CMRR	Common-Mode Rejection Ratio
PCB	Printed Circuit Board
SMD	Surface Mount Components
CAD	Computer-Aided Design
VR	Virtual Reality
AR	Augmented Reality
XR	Extended Reality
MAV	Mean Absolute Value
RMS	Root Mean Squared
ZC	Zero Crossing
WL	Waveform Length
SSC	Slope Sign Change
CNN	Convolutional Neural Networks
SVM	Support Vector Machines
LSTM	Long Short-Term Memory
CART	Classification And Regression Tree
XGBoost	eXTreme Gradient Boosting
TP	True Positives
TN	True Negatives

FP	False Positives
FN	False Negatives
LSB	Least Significant Bit
BW	Bandwidth
IR	Impulse Response
FDS	Flexor Digitorum Superficialis
PCSA	Physiological Cross-Sectional Area
MVC	Maximum Voluntary Contraction
fMRI	Functional Magnetic Resonance Imaging
B2B	Business to Business

Κεφάλαιο 1ο: Εισαγωγή

1.1 Εισαγωγή

Σήμερα η χρήση της ηλεκτρομυογραφίας (EMG) έχει εξαπλωθεί σε πολλούς κλάδους της τεχνολογίας και αποτελεί έναν από τους πιο περιζήτητους κλάδους με εταιρίες όπως η Meta, η Google και πολλούς ακόμα κολοσσούς να χρηματοδοτούν εκτεταμένες έρευνες. Αν και ο συγκεκριμένος επιστημονικός κλάδος, είδε τις πρώτες ραγδαίες εξελίξεις του κατά την Αναγέννηση – γνωστή και ως «η χρυσή εποχή της ανατομίας» - έχει απασχολήσει την επιστημονική κοινότητα ανά τους αιώνες, καθώς επιστήμονες από την αρχαία Αίγυπτο, Ελλάδα και την Ρωμαϊκή αυτοκρατορία προσπάθησαν να εξηγήσουν τα φαινόμενα που την διέπουν.

Κατά την διάρκεια της Αναγέννησης διακεκριμένοι επιστήμονες αφιέρωσαν αρκετό χρόνο στην ανάλυση των μηχανισμών που διέπουν τους μύες. Ιστορικές φιγούρες όπως ο Λεονάρντο Ντα Βίντσι (Leonardo da Vinci) και ο Ανδρέας Βεσάλιος (Andreas Vesalius) – γνωστός και ως «ο πατέρας της σύγχρονης ανατομίας» – ο οποίος πρόσφερε καινοτόμες ιδέες σε όλες τις πτυχές των λειτουργιών του ανθρώπινου σώματος και την διασύνδεσή του με το νευρικό σύστημα.

Ακόμα, προς το τέλος του 18^{ου} αιώνα ο Λουίτζι Γκαλβάνι (Luigi Galvani) μέσω της έρευνάς του στο πεδίο της ηλεκτροφυσιολογίας, απέδειξε πως οι μύες στα ζώα ενεργοποιούνται με την βοήθεια της ηλεκτρικής διέγερσης. Πιο συγκεκριμένα, ο Γκαλβάνι χρησιμοποίησε ένα γδαρμένο πτώμα βατράχου, όπου διέγειρε τους μύες των ποδιών του με την βοήθεια ηλεκτρικών σπινθήρων και παρατήρησε πως αυτοί συστάληκαν.

Αν και τα ευρήματα των παραπάνω επιστημόνων έθεσαν τα θεμέλια για τους επόμενους, στις έρευνές τους χρησιμοποιήθηκαν κυρίως νεκρά ή απομονωμένα βιολογικά δείγματα. Ο πρώτος επιστήμονας που πραγματοποίησε εκτεταμένες μελέτες πάνω σε ζωντανούς ανθρώπους ήταν ο Γάλλος νευρολόγος Γκιγιόμ-Μπενζαμίν-Αμάν Ντυσέν (Guillaume-Benjamin-Amand Duchenne), ο οποίος διερεύνησε σε βάθος την αγωγιμότητα των νεύρων. Με την χρήση ηλεκτρικών ερεθισμάτων ο Ντυσέν κατάφερε να χαρτογραφήσει τη λειτουργία των μυών, θέτοντας τις βάσεις για τη σύγχρονη κλινική EMG και τη λειτουργική ανατομία.

Μετάπειτα με την σειρά τους αρκετοί επιστήμονες χρησιμοποίησαν την EMG κυρίως για διαγνωστικούς σκοπούς, συνεχίζοντας την δουλειά των προηγούμενων. Προς το τέλος του 2^{ου} Παγκοσμίου Πολέμου οι Inman, Saunders και Abbott δημοσίευσαν τα ευρήματά τους σχετικά με την έρευνά των κινήσεων της ωμικής ζώνης. Η δουλειά τους έγινε ευρέως αποδεκτή από την ακαδημαϊκή και ιατρική κοινότητα, και πυροδότησε εκτενής έρευνα πάνω στην ανατομία με την χρήση EMG από επιστήμονες παγκοσμίως. Οι έρευνες αυτές έθεσαν τις βάσεις του κλάδου, εισάγοντας όρους όπως το Motor Unit, Action Potential και γενικότερα έδωσαν μια πιο ενδελεχής περιγραφή στους μηχανισμούς των μυών.

Πλέον, όπως προαναφέρθηκε, παρατηρούμε κολοσσούς του τεχνολογικού κλάδου να επενδύουν μεγάλα χρηματικά ποσά σε εσωτερικά ή εξωτερικά εργαστήρια, τα οποία ασχολούνται με σχετική έρευνα. Αυτό συμβαίνει γιατί με το πέρασμα του χρόνου οι εφαρμογές EMG έχουν πληθύνει, ξεφεύγοντας από το κλινικό περιβάλλον που οι επιστήμονες τις είχαν συνηθίσει.

1.2 Σύγχρονες Εφαρμογές και Τεχνολογικές Τάσεις

Στις ημέρες μας, η τεχνολογία EMG πέρα από την παραδοσιακή της εφαρμογή σε ιατρικά και κλινικά περιβάλλοντα, έχει επεκταθεί και σε προϊόντα ελέγχου ηλεκτρονικών συσκευών όπου οι χρήστες αλληλοεπιδρούν με αυτές χρησιμοποιώντας τη μυϊκή τους δραστηριότητα. Πιο συγκεκριμένα, τεχνολογίες όπως τα VR/AR γυαλιά, μέτρηση απόδοσης αθλητών καθώς και αμυντικές εφαρμογές έχουν αυξήσει την ζήτηση για έρευνα και την εμπορευματοποίησή της.

1.2.1 Ιατρικές και κλινικές εφαρμογές

Η EMG αποτελεί ένα από τα σημαντικότερα εργαλεία στην ιατρική και σε κλινικά περιβάλλοντα με σκοπό τη διάγνωση και την παρακολούθηση ασθενειών που επηρεάζουν το νευρομυϊκό σύστημα. Οι κυριότερες εφαρμογές της εντοπίζονται στους εξής τομείς:

- Διάγνωση Νευρομυϊκών Διαταραχών: Το EMG χρησιμοποιείται για τον εντοπισμό παθήσεων όπως η πλαγία μυατροφική σκλήρυνση (ALS), η μυασθένεια Gravis και οι μυϊκές δυστροφίες. Μέσω της ανάλυσης των δυναμικών δράσης των κινητικών μονάδων (MUAP), οι ιατροί μπορούν να διακρίνουν αν μια αδυναμία οφείλεται σε βλάβη του ίδιου του μυός ή του νεύρου που τον ελέγχει [1].
- Αξιολόγηση Νευρικών Παγιδεύσεων: Μία από τις πιο κοινές κλινικές χρήσεις είναι η διάγνωση του συνδρόμου καρπιαίου σωλήνα. Το EMG, σε συνδυασμό με μελέτες νευρικής αγωγιμότητας, βοηθά στον προσδιορισμό του ακριβούς σημείου και της σοβαρότητας της πίεσης του νεύρου [2].
- Νευροαποκατάσταση μετά από Εγκεφαλικό Επεισόδιο: Σε ασθενείς που έχουν υποστεί εγκεφαλικό επεισόδιο, το επιφανειακό EMG (sEMG) χρησιμοποιείται για την επανεκπαίδευση των μυών μέσω βιοανάδρασης (biofeedback). Ο ασθενής βλέπει σε πραγματικό χρόνο την ηλεκτρική δραστηριότητα των μυών του σε μια οθόνη, κάτι που τον βοηθά να ανακτήσει τον εκούσιο έλεγχο των κινήσεών του [3].
- Έλεγχος Μυοηλεκτρικών Προθέσεων: Στον τομέα της αποκατάστασης ακρωτηριασμένων, τα σήματα EMG από τους εναπομεινάντες μύες του μέλους χρησιμοποιούνται ως εντολές για την κίνηση τεχνητών μελών. Οι σύγχρονες προθέσεις χρησιμοποιούν εξελιγμένους αλγόριθμους αναγνώρισης προτύπων (pattern recognition) για να εκτελούν σύνθετες κινήσεις, όπως το κλείσιμο της γροθιάς ή η έκταση των δακτύλων [4].

1.2.2 Αλληλεπίδραση Ανθρώπου-Υπολογιστή (HCI) και Καταναλωτικά Προϊόντα

Η μετάβαση της τεχνολογίας sEMG από το κλινικό περιβάλλον στα καταναλωτικά προϊόντα έχει μεταμορφώσει τον τρόπο με τον οποίο οι χρήστες αλληλεπιδρούν με τα ψηφιακά συστήματα, προσφέροντας μια πιο φυσική και «αόρατη» διεπαφή χρήστη (Human-Computer Interface). Οι κυριότερες εφαρμογές περιλαμβάνουν:

- Νευρικές Διεπαφές Καρπού (Neural Wristbands): Η Meta (Reality Labs) αναπτύσσει περικάρπια που χρησιμοποιούν sEMG για να αποκωδικοποιούν τα νευρικά σήματα που κατευθύνονται προς τα δάκτυλα. Η τεχνολογία αυτή επιτρέπει τον έλεγχο συσκευών VR/AR με ελάχιστες κινήσεις (όπως το «surface typing» ή η ανίχνευση της στάσης του χεριού), μετατρέποντας τον καρπό σε μια καθολική διεπαφή ελέγχου [5], [6], [7].
- Αλληλεπίδραση σε Περιβάλλοντα Extended Reality (XR): Το sEMG επιλύει το πρόβλημα του «πάντα διαθέσιμου ελέγχου» (always-available input) σε περιβάλλοντα Εικονικής και Επαυξημένης Πραγματικότητας. Επιτρέπει την εκτέλεση εντολών χωρίς την ανάγκη οπτικής επαφής με κάμερες (hand tracking) ή τη χρήση ογκωδών χειριστηρίων, προσφέροντας υψηλή ταχύτητα και ακρίβεια στην απόκριση [8], [9].

- Έλεγχος μέσω Χειρονομιών (Gesture Control) για Μουσικά Όργανα και Πολυμέσα: Συστήματα που βασίζονται σε εμπορικά διαθέσιμα περικάρπια (όπως το Myo Armband) χρησιμοποιούνται για τον έλεγχο ψηφιακών μουσικών οργάνων και πολυμέσων, συνδυάζοντας δεδομένα EMG με γυροσκόπια για την ιχνηλάτηση της κίνησης του χεριού σε τρισδιάστατο χώρο [5].
- Βελτιστοποίηση μέσω Μηχανικής Μάθησης (Machine Learning): Η χρήση εξελιγμένων αρχιτεκτονικών, όπως τα Συνελκτικά Νευρωνικά Δίκτυα (CNN), επιτρέπει την αναγνώριση σύνθετων λειτουργικών κινήσεων και προτύπων δύναμης με ακρίβεια που ξεπερνά το 90%, καθιστώντας τα συστήματα αυτά αξιόπιστα για καθημερινή χρήση [6], [10].

1.2.3 Αθλητική Επιστήμη και Εργονομία

Η εφαρμογή της ηλεκτρομυογραφίας στην αθλητική επιστήμη και την εργονομία προσφέρει πολύτιμα δεδομένα για τη φυσική κατάσταση, την αποδοτικότητα και την ασφάλεια του ανθρώπου κατά την κίνηση. Οι βασικές εφαρμογές περιλαμβάνουν:

- Παρακολούθηση Μυϊκής Κόπωσης: Το sEMG επιτρέπει στους αθλητές και τους προπονητές να εντοπίζουν τα πρώιμα σημάδια μυϊκής κόπωσης μέσω της ανάλυσης της φασματικής μετατόπισης του σήματος προς χαμηλότερες συχνότητες. Η δυνατότητα αυτή συμβάλλει στη βελτιστοποίηση των προγραμμάτων προπόνησης και στην αποφυγή της υπερκόπωσης [11], [12].
- Βελτιστοποίηση Αθλητικής Τεχνικής: Μέσω της ανάλυσης του χρονισμού και της έντασης ενεργοποίησης των μυών (muscle activation timing), οι αθλητές μπορούν να διορθώσουν την τεχνική τους σε αθλήματα όπως η άρση βαρών, η κολύμβηση ή το τρέξιμο. Αυτό διασφαλίζει τη μέγιστη δυνατή απόδοση με τη μικρότερη δυνατή κατανάλωση ενέργειας [11], [13].
- Πρόληψη Τραυματισμών: Ο εντοπισμός μυϊκών ανισορροπιών ή λανθασμένων προτύπων ενεργοποίησης επιτρέπει την έγκαιρη παρέμβαση για την πρόληψη τραυματισμών, ειδικά σε αθλήματα που περιλαμβάνουν επαναλαμβανόμενες και εκρηκτικές κινήσεις [13].
- Εργονομική Αξιολόγηση Θέσεων Εργασίας: Στον τομέα της εργονομίας, το sEMG χρησιμοποιείται για την ποσοτική αξιολόγηση της μυϊκής καταπόνησης κατά τη διάρκεια επαγγελματικών δραστηριοτήτων. Μελετώντας τη στατική και δυναμική φόρτιση των μυών (π.χ. στην περιοχή του αυχένα ή της οσφυϊκής μοίρας), καθίσταται δυνατός ο σχεδιασμός εργαλείων και θέσεων εργασίας που μειώνουν τον κίνδυνο επαγγελματικών μυοσκελετικών παθήσεων [12], [14].

1.2.4 Τεχνολογικές Τάσεις και Μηχανική Μάθηση

Η σύγχρονη έρευνα στον τομέα του sEMG επικεντρώνεται στη μετάβαση από τα ογκώδη κλινικά συστήματα σε αυτόνομες, «έξυπνες» φορητές συσκευές. Οι κυριότερες τάσεις που διαμορφώνουν το μέλλον του κλάδου περιλαμβάνουν:

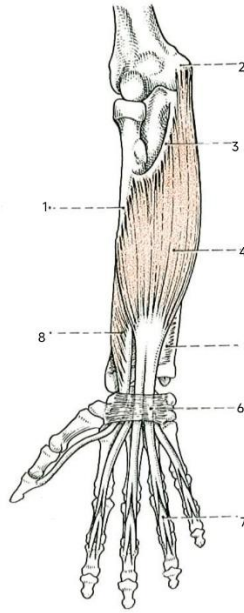
- Edge Computing και Επεξεργασία σε Πραγματικό Χρόνο: Η ανάγκη για χαμηλή καθυστέρηση (latency) στις διεπαφές ανθρώπου-υπολογιστή έχει οδηγήσει στην υιοθέτηση του Edge Computing. Αντί για τη μετάβαση των δεδομένων σε κεντρικούς διακομιστές, η επεξεργασία και η ταξινόμηση των σημάτων πραγματοποιούνται απευθείας στον μικροελεγκτή. Αυτή η προσέγγιση εξασφαλίζει ταχύτητα απόκρισης και ενισχύει την προστασία των προσωπικών δεδομένων του χρήστη [15].

- Αλγόριθμοι Μηχανικής Μάθησης (XGBoost): Ενώ οι παραδοσιακές μέθοδοι βασίζονταν σε απλά στατιστικά κατώφλια, οι σύγχρονες υλοποιήσεις χρησιμοποιούν προηγμένους αλγόριθμους για την αναγνώριση προτύπων. Το XGBoost αποτελεί μια κορυφαία επιλογή για φορητά συστήματα, καθώς προσφέρει εξαιρετική ακρίβεια στην ταξινόμηση χειρονομιών, διαχειριζόμενο αποτελεσματικά τη μη γραμμικότητα των σημάτων με χαμηλό υπολογιστικό κόστος [16].
- Εξαγωγή Χαρακτηριστικών (Feature Extraction): Η επιτυχία των μοντέλων μηχανικής μάθησης βασίζεται στον σωστό συνδυασμό χαρακτηριστικών από το πεδίο του χρόνου και της συχνότητας. Χαρακτηριστικά όπως η Ρίζα του Μέσου Τετραγώνου (RMS) και η Μέση Απόλυτη Τιμή (MAV) αποτελούν πλέον τη βάση για την εκπαίδευση μοντέλων που μπορούν να διακρίνουν σύνθετες κινήσεις των δακτύλων σε πραγματικό χρόνο [17].

1.3 Σκοπός και Στόχοι της Εργασίας

Μέχρι πρόσφατα, τα συστήματα ηλεκτρομυογραφίας περιορίζονταν αποκλειστικά σε κλινικά περιβάλλοντα και ήταν εξαιρετικά ακριβά και ογκώδη, θέτοντας δύσκολη την ενσωμάτωσή τους στην τεχνολογία που χρησιμοποιεί ο άνθρωπος καθημερινά. Πλέον, με την εξέλιξη της φορητής τεχνολογίας όπως έξυπνα ρολόγια, αξεσουάρ γυμναστικής ακόμα και εφαρμογές σε έξυπνα σπίτια έχει υπάρξει ένα μεγάλο κύμα μελετών με επίκεντρο τον εκδημοκρατισμό της συγκεκριμένης τεχνολογίας και την επέκταση του φάσματος των εφαρμογών της.

Ο σκοπός της παρούσας Διπλωματικής Εργασίας είναι η προσομοίωση, η σχεδίαση και η υλοποίηση ενός πλήρους συστήματος sEMG και η χρήση Μηχανικής Μάθησης για την αναγνώριση κινήσεων του χεριού και των δακτύλων. Το σύστημα αρχικά δέχεται σαν είσοδο - από ηλεκτρόδια - τα σήματα που παράγει ο μυς Flexor Digitorum Superficialis, ο οποίος βρίσκεται στο πήχη και ελέγχει τις κινήσεις του καρπού καθώς και τις κινήσεις των όλων των δακτύλων εκτός του αντίχειρα. Σε συνέχεια, γίνεται αναλογική επεξεργασία των συγκεκριμένων σημάτων με σκοπό να ενισχυθεί και να γίνει φιλτράρισμα συγκεκριμένων χρήσιμων συχνοτήτων. Με την έξοδο αυτού του αναλογικού κυκλώματος έγιναν 20 καταγραφές για κάθε επιθυμητή κίνηση με σκοπό την εκπαίδευση ενός μοντέλου Μηχανικής Μάθησης Random Forest. Τέλος, το μοντέλο αυτό ενσωματώνεται σε έναν μικροελεγκτή ESP32, ο οποίος δέχεται σαν είσοδο το επεξεργασμένο σήμα από το αναλογικό κύκλωμα και κάνοντας καταγραφές των 10ms και μπορεί να προβλέψει την κίνηση που έκανε ο χρήστης. Αν και το μέγεθος της συσκευής σε αυτό το επίπεδο της σχεδίασης είναι ακόμα μεγάλο, μπορεί να θεωρηθεί ως μία φορητή και φθηνή συσκευή ηλεκτρομυογραφίας [18].



Σχήμα 1.1 Γεωμετρία του μύος Flexor Digitorum Superficialis

Οι επιμέρους τεχνικοί στόχοι για την υλοποίηση αυτού του συστήματος είναι οι εξής:

- Προσομοίωση: Με χρήση του PSpice και της Python σχεδιάστηκε αρχικά το κύκλωμα σε κώδικα, καθώς και με την χρήση κατάλληλης βιβλιογραφίας και μαθηματικών έγινε προσομοίωση και των σημάτων που παράγουν οι μύες.
- Υλικό: Σχεδιασμός αναλογικού κυκλώματος με σκοπό την προενίσχυση του σήματος και φιλτράρισμα συγκεκριμένων συχνοτήτων.
- Δειγματοληψία: Με την χρήση του κυκλώματος έγινε κατάλληλη δειγματοληψία για τις επιθυμητές κινήσεις με σκοπό την εκπαίδευση μοντέλου μηχανικής μάθησης.
- Μηχανική Μάθηση: Εκπαίδευση ενός μοντέλου Random Forest και XGBoost για την ταξινόμηση των συγκεκριμένων χειρονομιών.
- Ενσωμάτωση: Χρήση ενός μικροελεγκτή ESP32 για την ενσωμάτωση του λογισμικού και την χρήση του αναλογικού κυκλώματος σαν είσοδο για την αναγνώριση χειρονομιών σε πραγματικό χρόνο.

Η ολοκλήρωση της παρούσας εργασίας αποσκοπεί στη δημιουργία ενός λειτουργικού πρωτοτύπου που αποδεικνύει την εφικτότητα της αναγνώρισης χειρονομιών σε πραγματικό χρόνο με χρήση υλικού χαμηλού κόστους. Η επιτυχής ενσωμάτωση του μοντέλου μηχανικής μάθησης απευθείας στον μικροελεγκτή (Edge AI) υπογραμμίζει τις δυνατότητες των σύγχρονων ενσωματωμένων συστημάτων στην επεξεργασία βιοσημάτων, χωρίς την ανάγκη διαρκούς σύνδεσης με εξωτερική υπολογιστική ισχύ. Τα αποτελέσματα της μελέτης αυτής δεν περιορίζονται μόνο στην τεχνική υλοποίηση, αλλά αποτελούν τη βάση για μελλοντικές εφαρμογές σε τομείς όπως η αποκατάσταση ασθενών, ο έλεγχος διεπαφών Εικονικής Πραγματικότητας (VR/AR) και η ανάπτυξη υποστηρικτικών τεχνολογιών για άτομα με κινητικές δυσκολίες, προσφέροντας μια οικονομικά προσιτή λύση στην αλληλεπίδραση ανθρώπου-μηχανής.

1.4 Δομή της Διπλωματικής Εργασίας

Η δομή της παρούσας Διπλωματικής Εργασίας (Δ.Ε.) οργανώνεται σε επιμέρους κεφάλαια, τα οποία ακολουθούν μια λογική αλληλουχία, ξεκινώντας από τη θεωρητική θεμελίωση και καταλήγοντας στην πρακτική εφαρμογή και υλοποίηση του συστήματος.

Στο δεύτερο κεφάλαιο πραγματοποιείται η ανάλυση της φυσιολογίας των μυών και του μηχανισμού παραγωγής των βιοηλεκτρικών σημάτων. Στο πλαίσιο αυτό, παρουσιάζονται οι μαθηματικές εξισώσεις και οι βιολογικές αρχές που διέπουν τη λειτουργία αυτή, ενώ παράλληλα αναλύεται ο κώδικας της προσομοίωσης που αναπτύχθηκε βάσει των προαναφερθέντων θεωρητικών στοιχείων.

Στο τρίτο κεφάλαιο αναλύεται ο σχεδιασμός του συστήματος και οι επιμέρους τεχνολογίες που χρησιμοποιήθηκαν, ενώ περιγράφονται τα εργαστηριακά εργαλεία που αξιοποιήθηκαν κατά τη διάρκεια της υλοποίησης. Συγκεκριμένα, γίνεται εκτενής αναφορά στα στάδια του αναλογικού κυκλώματος, καθώς και στις μεθόδους φιλτραρίσματος και ενίσχυσης του σήματος.

Στο τέταρτο κεφάλαιο περιγράφεται το μοντέλο μηχανικής μάθησης και η διαδικασία επεξεργασίας των δεδομένων εκπαίδευσης. Μετά από μια σύντομη εισαγωγή στις αρχές της Μηχανικής Μάθησης, αναλύεται η συλλογή και η προεπεξεργασία των δεδομένων, καθώς και η συσχέτιση των επιλεγμένων χαρακτηριστικών (features). Στη συνέχεια, παρουσιάζονται οι αλγόριθμοι Random Forest και XGBoost, ακολουθούμενοι από την αξιολόγηση του μοντέλου ως προς την ακρίβεια των αποτελεσμάτων του.

Στο πέμπτο και τελευταίο κεφάλαιο παρατίθεται η σύνοψη των αποτελεσμάτων της εργασίας. Παράλληλα, κατατίθενται προτάσεις για μελλοντικές βελτιώσεις του συστήματος, καθώς και μια ανάλυση των εμπορικών προοπτικών που παρουσιάζει μια αντίστοιχη ολοκληρωμένη συσκευή.

1.5 Επίλογος

Συνοψίζοντας, η ηλεκτρομυογραφία (EMG) απασχολεί επιστήμονες από τα αρχαία χρόνια μέχρι και σήμερα με τις εξελίξεις στον συγκεκριμένο τομέα να είναι ολοένα και πιο καινοτόμες καθιστώντας ευκολότερη την πρόσβαση σε αυτού του είδους τεχνολογία. Ειδικότερα, καθώς οι εφαρμογές τέτοιων συστημάτων ξεφεύγουν από την κλασική τους εφαρμογή σε κλινικά περιβάλλοντα και μεταβαίνουν στον κόσμο της φορητής τεχνολογίας, η ανάγκη για εκτεταμένη μελέτη και υλοποίηση συστημάτων που επιτυγχάνουν την ενσωμάτωσή τους με χαμηλό κόστος γίνεται ακόμα πιο επιτακτική. Προκειμένου να επιτευχθεί η υλοποίηση ενός τέτοιου συστήματος απαιτείται μια ενδελεχής κατανόηση των φυσικών και βιολογικών διεργασιών που λαμβάνουν μέρος τόσο στο ανθρώπινο σώμα όσο και στη σχεδίαση του υλικού που θα χρησιμοποιηθεί. Ως εκ τούτου, πριν γίνει εκτενής αναφορά στο κατασκευαστικό μέρος της μελέτης, πρέπει να γίνει μια εκτενής περιγραφή της φυσιολογίας των μυών και των αρχών της ηλεκτροφυσιολογίας. Στο συγκεκριμένο θέμα θα αφιερωθεί το επόμενο κεφάλαιο.

Κεφάλαιο 2ο: Ανάλυση Προσομοίωσης και Βιολογικού Σήματος

2.1 Εισαγωγή

Ενώ στο πρώτο κεφάλαιο αναλύθηκαν οι σύγχρονες εφαρμογές και οι τεχνολογικές τάσεις της ηλεκτρομυογραφίας, στο παρόν κεφάλαιο η έμφαση μετατοπίζεται στην κατανόηση της βιολογικής πηγής του σήματος. Τα σήματα ηλεκτρομυογραφίας δεν αποτελούν μια απλή ηλεκτρική μεταβλητή, αλλά είναι το αποτέλεσμα σύνθετων βιοηλεκτρικών διεργασιών που λαμβάνουν χώρα στο εσωτερικό του μυϊκού ιστού. Για την καλύτερη προσέγγιση του θέματος, κρίνεται απαραίτητη η ανάλυση των μηχανισμών αυτών πριν από οποιαδήποτε προσπάθεια τεχνικής υλοποίησης.

Η κατανόηση της φυσιολογίας της μυϊκής σύσπασης είναι καθοριστικής σημασίας για τον μηχανικό, καθώς οι βιολογικές ιδιότητες του σήματος καθορίζουν άμεσα τις τεχνικές προδιαγραφές του συστήματος. Παράμετροι όπως το εξαιρετικά χαμηλό εύρος τάσης (της τάξης των mV) και το συγκεκριμένο φασματικό περιεχόμενο των μυϊκών σημάτων αποτελούν τα δεδομένα βάσει των οποίων θα σχεδιαστούν τα αναλογικά φίλτρα και τα στάδια ενίσχυσης. Παράλληλα, η ποιότητα της ταξινόμησης από τους αλγόριθμους Μηχανικής Μάθησης εξαρτάται άμεσα από την ορθή επιλογή των χαρακτηριστικών που αντικατοπτρίζουν τη φύση της μυϊκής δραστηριότητας.

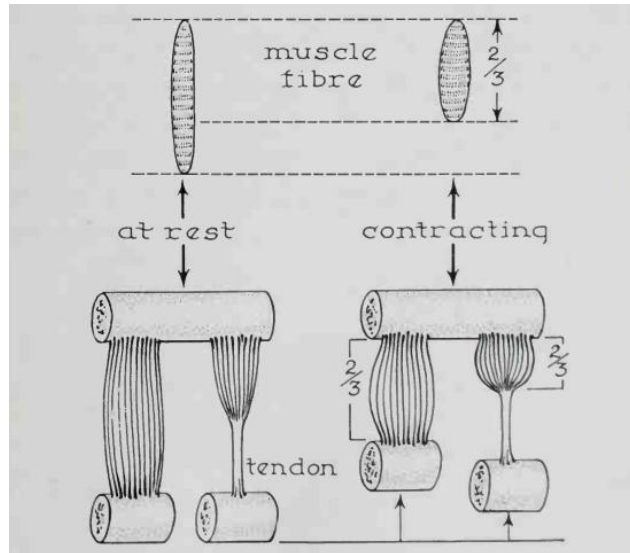
Στο πλαίσιο αυτό, το κεφάλαιο ξεκινά με την περιγραφή του δυναμικού δράσης (Action Potential) και της λειτουργίας των κινητικών μονάδων (Motor Units). Στη συνέχεια, εξετάζονται τα μαθηματικά χαρακτηριστικά που διέπουν τα σήματα EMG, καθώς και οι μέθοδοι επεξεργασίας και ερμηνείας τους. Το κεφάλαιο ολοκληρώνεται με την ανατομική περιγραφή του μυός Flexor Digitorum Superficialis, ο οποίος αποτελεί τον στόχο της παρούσας μελέτης για την αναγνώριση των χειρονομιών.

2.2 Μηχανισμός Γέννησης Μυϊκών Σημάτων

Η παραγωγή των μυϊκών σημάτων που ανιχνεύονται από τα ηλεκτρόδια βασίζεται σε έναν ιεραρχικό μηχανισμό βιοηλεκτρικής δραστηριότητας με κεντρικό ρόλο σε αυτόν τον μηχανισμό διαδραματίζουν οι Κινητικές Μονάδες (Motor Units - MUs), οι οποίες αποτελούν τη βασική λειτουργική μονάδα ελέγχου του νευρομυϊκού συστήματος κατά τη διάρκεια της σύσπασης. Κατά τη διέγερση ενός μυός, οι κινητικές μονάδες προκαλούν τη γέννηση στοιχειωδών ηλεκτρικών σημάτων, γνωστών ως Δυναμικά Δράσης (Action Potentials - AP). Η χωρική και χρονική υπέρθεση αυτών των σημάτων οδηγεί στη δημιουργία του Δυναμικού Δράσης της Κινητικής Μονάδας (Motor Unit Action Potential - MUAP), το οποίο αποτελεί τη βάση για το σύνθετο σήμα που καταλήγει στους αισθητήρες με σκοπό την καταγραφή και την ανάλυσή του [19].

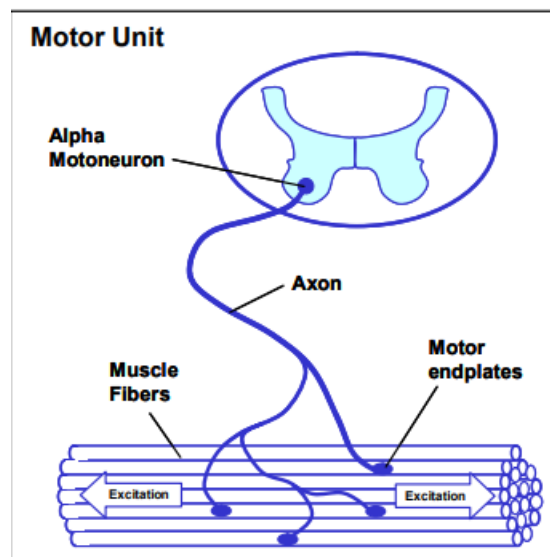
2.2.1 Κινητικές Μονάδες (Motor Units)

Για την καλύτερη επεξήγηση των κινητικών μονάδων θα πρέπει αρχικά να γίνει κατανοητή η δομή τους. Βασική συνιστώσα των κινητικών μονάδων είναι οι μυϊκές ίνες, η οποίες έχουν μήκος έως 30mm και πλάτος μικρότερο από 0.1mm. Κατά την σύσπαση, το μήκος τους μικραίνει κατά 57%, σε σχέση με το μήκος που έχουν όταν βρίσκονται σε αδράνεια [1].



Σχήμα 2.1 Σύσπαση μυϊκής ίνας [1]

Οι μυϊκές ίνες, όπως έχει παρατηρηθεί στα θηλαστικά, δεν συσπώνται ατομικά αλλά σε μικρά σύνολα. Το σώμα ενός κινητικού νευρώνα σε συνδυασμό με τις μυϊκές ίνες που καταλήγουν σε αυτόν αποτελούν μία κινητική μονάδα (fig. 3). Για τον λόγο αυτό χρησιμοποιείται και ο όρος μονάδα, καθώς κατά την νευρική τροφοδότηση των μυών όλες οι μυϊκές ίνες που υπάγονται στην εκάστοτε motor unit δρουν ως ομάδα [1].

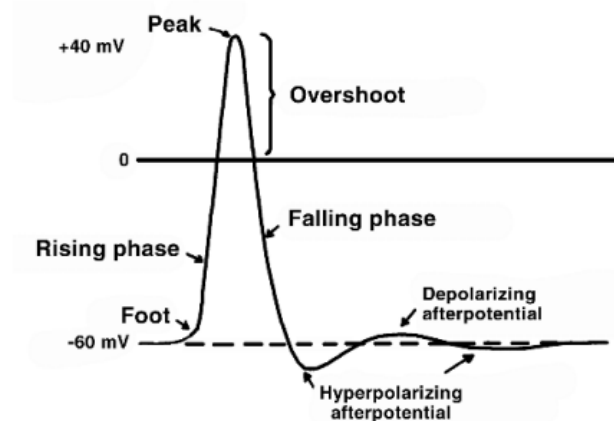


Σχήμα 2.2 Φυσιολογία κινητικής μονάδας [19]

2.2.2 Δυναμικό Δράσης (Action Potential)

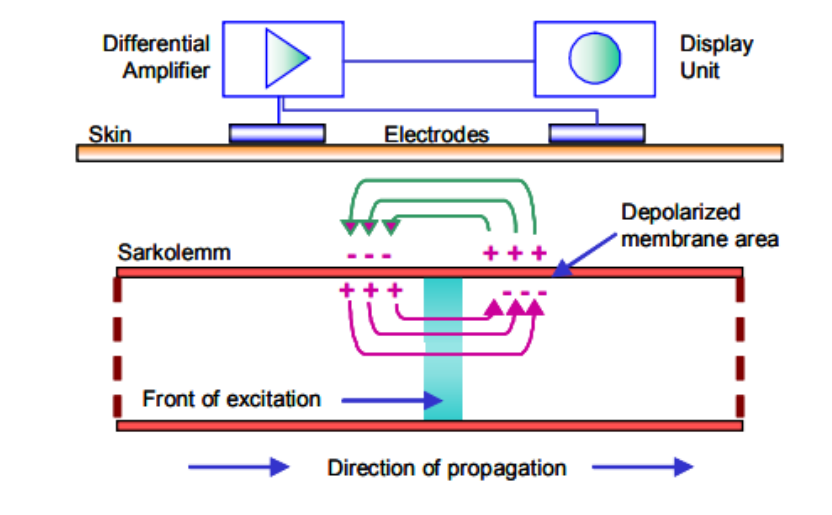
Τα μικρότερα σήματα που παράγονται από τους μύες είναι τα δυναμικά δράσης. Για την δημιουργία αυτών των σημάτων είναι αίτια η μεμβράνη της μυϊκής ίνας, η οποία χρησιμοποιεί την αποθηκευμένη ενέργεια για να επιτρέψει τη ροή ιόντων νατρίου για να αυξηθεί η διαφορά δυναμικού (εκπόλωση της μεμβράνης) και έπειτα τη ροή ιόντων καλίου με αποτέλεσμα την επαναπόλωση της μεμβράνης, δηλαδή

τη μείωση της τάσης. Το συγκεκριμένο σήμα είναι μια μονοπολική ριπή η οποία έχει πλάτος από -80mV έως $+30\text{mV}$ (fig. 3) [19], [20].



Σχήμα 2.3 Ριπή κατά την σύσπαση εκπόλωση και επαναπόλωση της μυϊκής μεμβράνης [20]

Το σήμα EMG που τελικά καταγράφεται από τα ηλεκτρόδια που είναι τοποθετημένα στην επιφάνεια του δέρματος, είναι αποτέλεσμα αυτών των μηχανισμών των μυών. Πιο συγκεκριμένα, η παραγόμενη διαφορά τάσης ταξιδεύει κατά μήκος της μυϊκής ίνας με ταχύτητα $2-6\text{m/s}$ και τελικά ανιχνεύεται από τους αισθητήρες. [19]

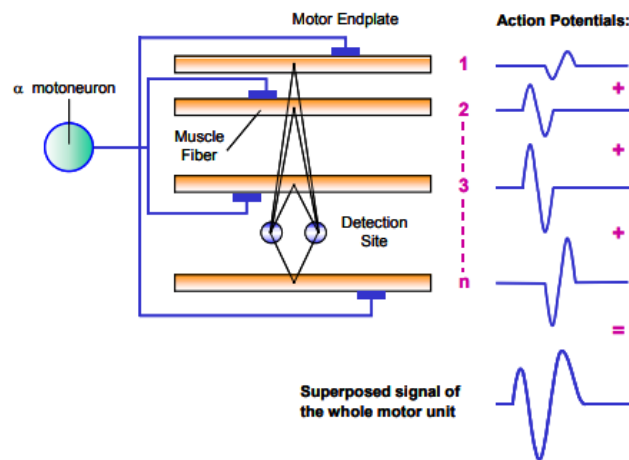


Σχήμα 2.4 Ζώνη αποπόλωσης της μυϊκής μεμβράνης [19]

2.2.3 Δυναμικό Δράσης Κινητικής Μονάδας (Motor Unit Action Potential)

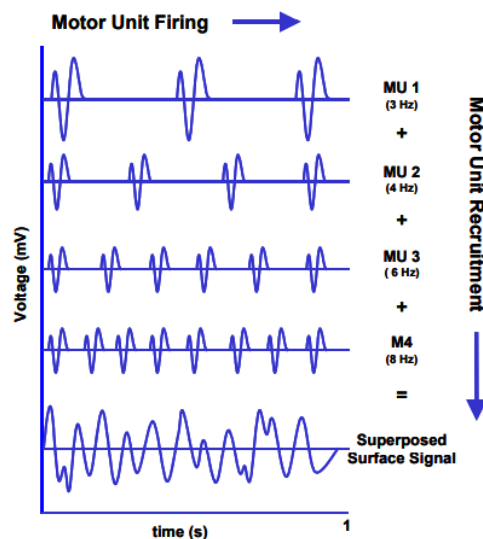
Καθώς οι κινητικές μονάδες αποτελούνται από πολλές μυϊκές ίνες, το σήμα που θα καταγραφεί από τον αναλογικό αισθητήρα θα είναι η υπέρθεση όλων των δυναμικών δράσης των μυϊκών ινών που υπάγονται στον συγκεκριμένο νευρώνα. Πιο συγκεκριμένα, από την επαλληλία των επιμέρους δυναμικών δράσης, παράγεται ένα τριφασικό σήμα που ονομάζεται δυναμικό δράσης κινητικής

μονάδας (Motor Unit Action Potential) και διαφέρει σε πλάτος ανάλογα τη τοποθέτηση των ηλεκτροδίων σε σχέση με γεωμετρία των μυϊκών ινών (Σχήμα 2.5) [19].



Σχήμα 2.5 Υπέρθεση των σημάτων δυνακού δράσης [19]

Για τη ρύθμιση της σύσπασης και της παραγόμενης ισχύς ενός μυ, υπάρχουν δύο βασικοί μηχανισμοί που λαμβάνουν μέρος. Αρχικά γίνεται η σταδιακή ενεργοποίηση των δυναμικών δράσης των κινητικών μονάδων κατά την οποία γίνεται η υπέρθεσή τους και έπειτα είναι η συχνότητα πυροδότησης, δηλαδή ο ρυθμός με τον οποίο μία κινητική μονάδα παράγει δυναμικά δράσης ανά μονάδα χρόνου. Το πλάτος του τελικού σήματος μπορεί να φτάσει τα 0.5mV σε άτομα όπως αθλητές, και το φασματικό του περιεχόμενο κυμαίνεται από 6 έως 500 Hz. [19]



Σχήμα 2.6 Ενεργοποίηση των κινητικών μονάδων [19]

Αρκετοί παράγοντες μπορούν να επηρεάσουν το τελικό EMG σήμα. Ειδικότερα, συνιστώσες όπως η αγωγιμότητα του δέρματος, παρεμβολές από γειτονικούς μυς, λάθος τοποθέτηση των ηλεκτροδίων σε σχέση με την τοποθεσία του μυ, εξωτερικός θόρυβος και η ποιότητα των υλικών που θα χρησιμοποιηθούν για την υλοποίηση του αναλογικού κυκλώματος μπορούν να επηρεάσουν το σήμα με τρόπο τέτοιο θέτοντας τη κατηγοριοποίησή του εξαιρετικά δύσκολη.

2.2.4 Ανισοτροπία των μυϊκών ινών

Πέρα από τους προαναφερθέντες παράγοντες που επηρεάζουν το τελικό EMG σήμα, μία ακόμη συνιστώσα είναι η αντίσταση που παρουσιάζουν οι μυϊκές ίνες. Αυτή μπορεί να διαχωριστεί στην αντίσταση κατά μήκος των μυϊκών ινών (σ_L) και εγκάρσια αυτών (σ_T). Ο λόγος αυτών των δύο μεταβλητών μας δίνει την ανισοτροπία του μυός ($\frac{\sigma_L}{\sigma_T}$) [21].

Οι πλέον αποδεκτές τυπικές τιμές για αυτές τις μεταβλητές είναι οι εξής:

$$0.33 < \sigma_L < 0.80 (\Omega)^{-1} \quad (2.1)$$

$$0.04 < \sigma_T < 0.15 (\Omega)^{-1} \quad (2.2)$$

$$2.04 < \frac{\sigma_L}{\sigma_T} < 15.33 \quad (2.3)$$

2.3 Μαθηματικά Χαρακτηριστικά των Σημάτων EMG

Σε αυτό το σημείο θα γίνει περιγραφή των πιο βασικών εξισώσεων που περιγράφουν τη γέννηση EMG σημάτων. Πιο συγκεκριμένα, θα γίνει ανάλυση των εξισώσεων που χρησιμοποιούνται για την εύρεση της τάσης που παράγει ένα δυναμικό δράσης, τη πυκνότητα του ρεύματος στις μεμβράνες των μυϊκών ινών και το εξωκυττάριο δυναμικό, το οποίο είναι και το τελικό σήμα που ανιχνεύουν τα ηλεκτρόδια.

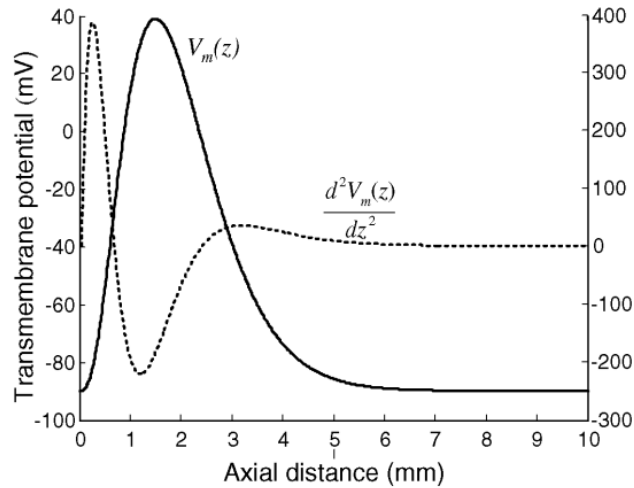
Η διαφορά δυναμικού που παράγει ένα δυναμικό δράσης κατά μήκος των μυϊκών ινών (V_m) μπορεί να αποτυπωθεί ως μια σειρά πηγών. Ειδικότερα, μια ευκολότερη μέθοδος για την περιγραφή του συγκεκριμένου μηχανισμού είναι η εξίσωση του δυναμικού δράσης κατά μήκος των μυών με την αξονική απόσταση z , όπως την διατύπωσε ο Ρόσενφאלκ (Rosenfalck) στο χωρικό πεδίο [12]:

$$V_m(z) = 96z^3 e^{-z} - 90 \quad (2.4)$$

και στο πεδίο του χρόνου:

$$V_m(t) = 96(u * t)^3 e^{-(u*t)} - 90 \quad (2.5)$$

Αρχικά η μεταβλητή της αξονικής απόστασης κατά μήκος της μυϊκής ίνας (z) περιγράφει τον φυσικό χώρο που ταξιδεύει το σήμα που παράγει ένα δυναμικό δράσης από τη κινητική τελική πλάκα. Περαιτέρω, η σταθερά -90 παρουσιάζει το ηλεκτρικό φορτίο (σε mV) της μυϊκής ίνας όταν αυτή βρίσκεται σε ανάπαυση. Το κομμάτι της εξίσωσης $96z^3 e^{-z}$ καθορίζει τη μαθηματική γεννήτρια παλμών που δημιουργεί την απότομη αύξηση και εκθετική μείωση της διαφοράς δυναμικού [12].

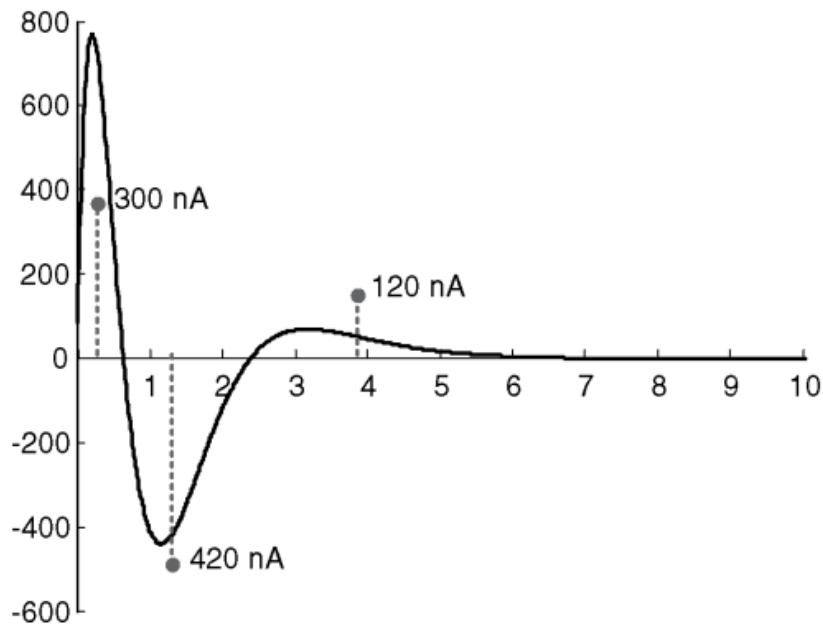


Σχήμα 2.7 Διαφορά δυναμικού κατά μήκος των μυϊκών ινών [12]

Η πυκνότητα του ρεύματος κατά μήκος των μυϊκών ινών είναι περίπου αντίστοιχη της δεύτερης παραγώγου της διαφοράς δυναμικού του δυναμικού δράσης:

$$i_m(z) = \frac{\sigma_{in}\pi d^2}{4} * \frac{d^2V_m(z)}{dz^2} \quad (2.6)$$

Για την σωστή αποτύπωση αυτού του μεγέθους γίνεται πολλαπλασιασμός της τάσης με τον λόγο της μυϊκής αγωγιμότητας σ_{in} , του π και της διαμέτρου της μυϊκής ίνας προς 4 [12].



Σχήμα 2.8 Πυκνότητα ρεύματος κατά μήκος των μυϊκών ινών [12]

Με την βοήθεια των δύο αυτών μεγεθών και την ανισοτροπία της μυϊκής ίνας μπορεί να προσδιοριστεί το τελικό σήμα που εν τέλει ανιχνεύεται από τα ηλεκτρόδια και ονομάζεται εξωκυττάριο δυναμικό [12]:

$$\varphi(z) = \frac{1}{4\pi\sigma_T} \int_{-\infty}^{\infty} \frac{i_m(z)}{\sqrt{\frac{\sigma_L}{\sigma_T} r^2 + (z - \zeta)^2}} d\zeta \quad (2.7)$$

Στον παραπάνω τύπο εξισώνονται αρκετές μεταβλητές όπως η r που περιγράφει την ακτινική απόσταση από την ίνα μέχρι το ηλεκτρόδιο, η ανισοτροπία $\frac{\sigma_L}{\sigma_T}$ και η πυκνότητα του ρεύματος $i_m(z)$ διαμέσου της μυϊκής ίνας. Μία πιο εύκολη και κατανοητή μέθοδος για τον υπολογισμό του εξωκυττάριου δυναμικού είναι η περιγραφή του με τη χρήση της πρώτης παραγώγου της διαφοράς δυναμικού V_m στο πεδίο του χρόνου:

$$\varphi(t, x_0, y_0, z_0) = \frac{\sigma_L}{\sigma_T} * \frac{dV_m(t)}{dt} * \frac{1}{v} * IR(t) \quad (2.8)$$

Όπου η κρουστική απόκριση $IR(t)$ είναι:

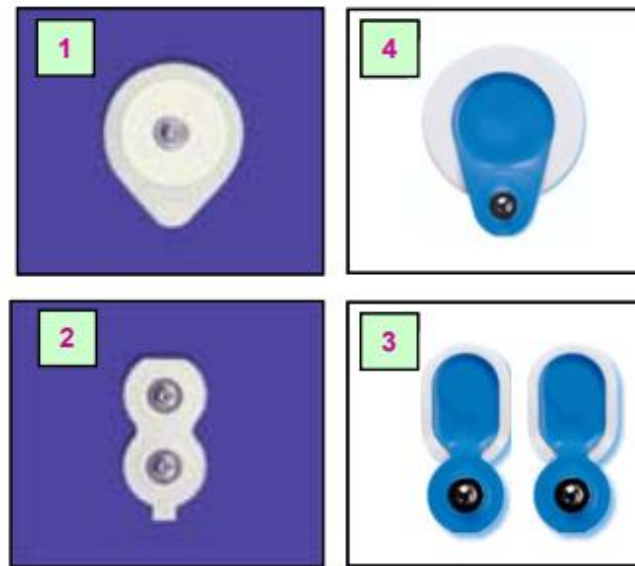
$$IR(t) = C1 \frac{d}{dt} \left\{ \frac{1}{[(x_0 - x_{ep} - vt)^2 + \frac{\sigma_L}{\sigma_T} (y_0 - y_i)^2 + \frac{\sigma_L}{\sigma_T} (z_0 - z_i)^2]^{\frac{1}{2}}} \right\} \quad (2.9)$$

$$+ C2 \frac{d}{dt} \left\{ \frac{1}{[(x_0 - x_{ep} + vt)^2 + \frac{\sigma_L}{\sigma_T} (y_0 - y_i)^2 + \frac{\sigma_L}{\sigma_T} (z_0 - z_i)^2]^{\frac{1}{2}}} \right\}$$

Οι πολλαπλασιαστές $C1$ και $C2$ που εμφανίζονται στην εξίσωση της κρουστικής απόκρισης, λειτουργούν σαν διακόπτες για την διάδοση του σήματος προς μία συγκεκριμένη κατεύθυνση. Πιο συγκεκριμένα, αν το σήμα που ξεκινάει από την τελική κινητική πλάκα βγει εκτός της ίνας τότε ο όρος $C1$ θα γίνει μηδέν και αντίστοιχα αν το κύμα ξεπεράσει το συνολικό μήκος της ίνας τότε ο όρος $C2$ θα μηδενιστεί, προσομοιώνοντας έτσι τον τερματισμό του σήματος στα άκρα της ίνας [12].

2.4 Προκλήσεις Καταγραφής και Προεπεξεργασία Σήματος

Ως κύριο αισθητήρα για την ανίχνευση των σημάτων EMG χρησιμοποιούνται ηλεκτρόδια που προσκολλώνται στην επιφάνεια του δέρματος. Λόγω της μη επεμβατικής φύσης τους, αποτελούν τον πιο εύκολο και πρακτικό τρόπο για τη συγκεκριμένη εφαρμογή. Το μόνο τους μειονέκτημα είναι πως μπορούν να καταγράψουν σήματα αποκλειστικά από μύες που βρίσκονται κοντά στην επιφάνεια του δέρματος. Το μεταλλικό άκρο ενός ηλεκτροδίου είναι συνδεδεμένο σε ένα καλώδιο γεφύρωσης ώστε να μπορεί να χρησιμοποιηθεί σε μια δοκιμαστική πλακέτα.



Σχήμα 2.9 Τύποι ηλεκτροδίων [19]

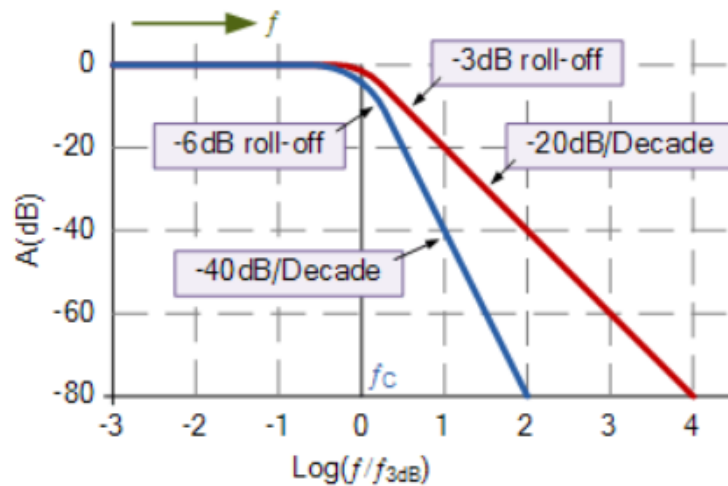
Η ορθή καταγραφή ηλεκτρομυογραφικών σημάτων γίνεται κατά κύριο τρόπο με την χρήση ηλεκτροδίων και προϋποθέτει κατάλληλη προετοιμασία τόσο ως προς τη τοποθέτησή τους, όσο και ως προς την επιφάνεια στην οποία αυτά θα προσκολληθούν.

Αρχικά, πρέπει να προσδιοριστεί το καλύτερο σημείο από το οποίο μπορούν τα ηλεκτρόδια να καταγράψουν τη δραστηριότητα του επιθυμητού μυ. Στη συγκεκριμένη επιφάνεια πρέπει να τοποθετηθούν δύο ηλεκτρόδια με μια μικρή απόσταση μεταξύ, με σκοπό να γίνει αξιοποίηση του CMRR του ενισχυτή οργανολογίας που βρίσκεται στο πρώτο στάδιο του αναλογικού κυκλώματος. Ακόμα, πρέπει να συνδεθεί και ένα τρίτο ηλεκτρόδιο σε μια οστική περιοχή, το οποίο δρα ως η κοινή γείωση του σώματος με το κύκλωμα [19].

Όταν προσδιοριστούν επιτυχώς τα παραπάνω, η επιφάνεια του δέρματος πρέπει να προετοιμαστεί για τη βέλτιστη χρήση των ηλεκτροδίων. Πιο συγκεκριμένα, προτείνεται να αφαιρεθούν οι τρίχες που καλύπτουν τα συγκεκριμένα σημεία και να γίνει καθαρισμός του δέρματος. Ο τελευταίος μπορεί να επιτευχθεί με τη χρήση ειδικού λειαντικού χαρτιού, ειδική αγωγίμη πάστα που αφαιρεί τα νεκρά κύτταρα από την επιφάνεια ή με την χρήση μιας μαλακής πετσέτας επικαλυμμένη με καθαρή αλκοόλη [19].

Πέρα από την προετοιμασία των επιφανειών που θα χρησιμοποιηθούν για την καταγραφή του σήματος υπάρχουν και άλλες προκλήσεις που αφορούν το σήμα καθαυτό. Ειδικότερα, πρέπει να αφαιρεθούν ορισμένες συχνότητες που παρουσιάζουν παράσιτα, θέτοντας την κατηγοριοποίηση του σήματος αρκετά δύσκολη. Τα σήματα ηλεκτρομυογραφίας εμφανίζουν την περισσότερη ισχύ στις συχνότητες από 10Hz-500Hz. Για τον λόγο αυτό πρέπει να κατασκευαστεί ένα ενεργό ζωνοπερατό φίλτρο (bandpass filter) που επιτρέπει την διέλευση μόνο αυτών των συχνοτήτων. Ακόμα, οι παρεμβολές του δικτύου ηλεκτροδότησης συντελούν στο θόρυβο που διέπει το ακατέργαστο σήμα. Για την αφαίρεσή του πρέπει να υλοποιηθεί ένα φίλτρο εγκοπής (notch filter) ρυθμισμένο στα 50Hz [19].

Για όλα τα φίλτρα που πρέπει να κατασκευαστούν, είναι προτεινόμενο να χρησιμοποιηθεί τοπολογία δεύτερης τάξης, λόγω του μεγαλύτερου ρυθμού αποκοπής (40dB/δεκάδα) σε σχέση με αυτού της πρώτης τάξεως (20dB/δεκάδα).



Σχήμα 2.10 Απόσβεση ανεπιθύμητων συχνοτήτων σε φίλτρα δευτέρου βαθμού [28]

2.5 Ανατομία και Φυσιολογία του Flexor Digitorum Superficialis

Ο μυς που επιλέχθηκε για καταγραφή στη παρούσα διπλωματική εργασία είναι ο Flexor Digitorum Superficialis (FDS). Πιο συγκεκριμένα, ο συγκεκριμένος μυς βρίσκεται στο μπροστά μέρος του πήχη και καταλήγει να διασπάται σε τέσσερις τένοντες, στο ύψος του καρπού, οι οποίοι καταλήγουν σε όλα τα δάχτυλα του χεριού εκτός του αντίχειρα.

Η επιλογή αυτή πηγάζει στη φύση της εργασίας και του γεγονότος πως αυτός είναι επιφανειακός μυς (Superficialis). Η κινήσεις που καταγράφηκαν είναι το κλείσιμο της παλάμης και η κάμψη του δείκτη, του μεσαίου, του παράμεσου και του μικρού δαχτύλου.



Σχήμα 2.11 Μήκος του FDS

Ο FDS αναγνωρίζεται ως ένας από τους πιο ογκώδεις μύες του πρόσθιου αντιβραχίου, με σημαντική φυσιολογική διατομή (PCSA) που δικαιολογεί τον κυρίαρχο ρόλο του στην παραγωγή δύναμης κατά τη λαβή. Ο συνδυασμός του μεγάλου μυϊκού όγκου και των συγκεκριμένων γωνιών πτέρωσης (pennation angles) εξασφαλίζει την απαραίτητη μηχανική ισχύ για σύνθετες κινήσεις των δακτύλων, καθιστώντας τον ιδανικό στόχο για την εξαγωγή χαρακτηριστικών μέσω ηλεκτρομυογραφήματος. [23]

Πίνακας 2.1 Γεωμετρικά χαρακτηριστικά του μυός Flexor Digitorum Superficialis [23]

Μυϊκή Μοίρα (Δάκτυλο)	Όγκος Μυός (cm ³)	Μήκος Ινών (cm)	Γωνία Πτέρωσης (μοίρες)	Φυσιολογική Διατομή - PCSA (cm ²)
Δείκτης (Index - 2nd digit)	14.8	5.2	5.0	2.5
Μέσος (Middle - 3rd digit)	11.2	5.5	6.0	1.8
Παράμεσος (Ring - 4th digit)	7.5	5.1	7.0	1.2
Μικρός (Little - 5th digit)	4.1	4.8	4.0	0.7
Σύνολο / Μέσος Όρος (FDS)	37.6	5.15	5.5	Συνολικό: 6.2

2.6 Επίλογος

Ανακεφαλαιώνοντας, η γέννηση ηλεκτρομυογραφικών σημάτων είναι ένας πολύπλοκος μηχανισμός που για να γίνει η περιγραφή του πρέπει να πρώτα να γίνει κατανοητή η φυσιολογία ενός μυ. Από τη πιο μικρή μετρήσιμη μονάδα, τη κινητική, μέχρι και το τελικό σήμα που καταγράφεται από τους αισθητήρες, το δυναμικό δράσης κινητικής μονάδας, μπορούν να αποτυπωθούν με μαθηματικές εξισώσεις, δίνοντας τη δυνατότητα προσομοίωσης και περαιτέρω κατανόησης του μηχανισμού αυτού. Ακόμα, υπάρχουν αρκετές δυσκολίες και στην υλοποίηση αναλογικών κυκλωμάτων που χρησιμοποιούνται για την καταγραφή αυτών των σημάτων. Η ορθή χρήση αυτού το υλικού προϋποθέτει ενδεδεγμένη προετοιμασία για την τοποθέτηση των ηλεκτροδίων. Η επιλογή του μυός προς καταγραφή είναι επίσης ύψιστης σημασίας, καθώς αυτός πρέπει να βρίσκεται κοντά στην επιφάνεια του δέρματος για να είναι ανιχνεύσιμος από επιφανειακά ηλεκτρόδια, όπως ο Flexor Digitorum Superficialis.

Κεφάλαιο 3ο: Σχεδιασμός Συστήματος και Τεχνολογίες Υλοποίησης

3.1 Εισαγωγή

Ενώ στο προηγούμενο κεφάλαιο έγινε ανάλυση της φυσιολογίας των μυών, των μαθηματικών εξισώσεων που τη περιγράφουν καθώς και των προκλήσεων που θα έρθει αντιμέτωπος ένας μηχανικός όταν επιχειρήσει να κατασκευάσει ένα σύστημα καταγραφή ηλεκτρομυογραφικών σημάτων, στο παρόν κεφάλαιο θα εξεταστεί η υλοποίηση ενός τέτοιου συστήματος τόσο από την σκοπιά του υλικού όσο και από του λογισμικού.

Αρχικά θα γίνει περιγραφή της γλώσσας προγραμματισμού που χρησιμοποιήθηκε, των βιβλιοθηκών καθώς και του εργαστηριακού εξοπλισμού που συνέβαλαν στη διαδικασία σχεδίασης και κατασκευής του συστήματος. Ακολούθως, θα διερευνηθεί λεπτομερώς το σχεδιασμένο υλικό με μεγάλη έμφαση στη λειτουργία των αναλογικών φίλτρων του και στη χρήση του τελικού σήματος ως είσοδο στον μικροελεγκτή ESP32.

3.2 Γλώσσα Προγραμματισμού Python και Χρησιμοποιούμενες Βιβλιοθήκες

Η υλοποίηση του λογισμικού για το συγκεκριμένο σύστημα απαιτεί τη χρήση αλγορίθμων και εφαρμογών πάνω στην Ανάλυση Δεδομένων, στη Μηχανική Μάθηση καθώς και στα προηγμένα Μαθηματικά. Για τον λόγο αυτό επιλέχθηκε η γλώσσα προγραμματισμού Python για την υλοποίηση του συστήματος. Η ευκολία χρήσης της και η πληθώρα διαθέσιμων βιβλιοθηκών την καθιστούν κατάλληλη για τέτοιου είδους εφαρμογές. Ακόμα, η επεκτασιμότητα του λογισμικού που έχει συγγραφεί σε Python είναι εξαιρετικά μεγάλη, καθώς είναι ευρέως γνωστή και υπάρχουν βιβλιοθήκες σχεδόν για όλες τις εφαρμογές, από χαμηλού έως υψηλού επιπέδου.

Το λογισμικό της κατασκευής αρχικά κάνει τους κατάλληλους μαθηματικούς υπολογισμούς με βάση τα καταγεγραμμένα δεδομένα και τα μορφοποιεί, με σκοπό να τροφοδοτήσουν αργότερα την εκπαίδευση του μοντέλου μηχανικής μάθησης.

3.2.1 Βιβλιοθήκη NumPy

Η βιβλιοθήκη NumPy είναι από τις πλέον διαδεδομένες βιβλιοθήκες τόσο για την επεξεργασία δεδομένων με τη χρήση πινάκων όσο και για τις μαθηματικές μεθόδους που αυτή εμπεριέχει. Πιο συγκεκριμένα, λόγω της δομής τους, οι πίνακες που δημιουργούνται μέσω αυτής της βιβλιοθήκης αξιοποιούν τη μνήμη και την επεξεργαστική ισχύ του υπολογιστή αποδοτικότερα από τις παραδοσιακές λίστες της Python. Ακόμα, έχουν ενσωματωμένες μεθόδους για προηγμένες μαθηματικές πράξεις, καθιστώντας τη συγκεκριμένη βιβλιοθήκη εξαιρετικά χρήσιμη.

Σε περαιτέρω ανάλυση οι πίνακες έχουν σταθερό μήκος κατά τη δημιουργία τους και πρέπει όλα τα δεδομένα που αποθηκεύουν να είναι ίδιου τύπου. Ακόμα, η ενσωμάτωση των μαθηματικών πράξεων μέσα στη βιβλιοθήκη βοηθάει στη συγγραφή μικρότερου και πιο ευανάγνωστου κώδικα [24].

Η εισαγωγή της βιβλιοθήκης σε κάποιο πρόγραμμα είναι αρκετά εύκολη και ακολουθεί την παρακάτω διαδικασία εφόσον έχει εγκατασταθεί στο σύστημα:

```
import numpy as np
a1D = np.array([1, 2, 3, 4])
```

```
a2D = np.array([[1, 2], [3, 4]])
a3D = np.array([[[1, 2], [3, 4]], [[5, 6], [7, 8]]])
```

3.2.2 Βιβλιοθήκη Pandas

Η βιβλιοθήκη Pandas είναι σχεδιασμένη για την εύκολη και γρήγορη χρήση σχεσιακών ή επισημασμένων δεδομένων. Πιο συγκεκριμένα, χρησιμοποιείται ευρέως στην ανάλυση δεδομένων επιδιώκοντας να διαμορφωθεί στο πιο διαδεδομένο εργαλείο στο οικοσύστημα της Python πάνω στην επεξεργασία δεδομένων [25].

Το πακέτο αυτό αξιοποιήθηκε για την υλοποίηση της παρούσας διπλωματικής εργασίας με αυτοσκοπό τον μετασχηματισμό των δεδομένων σε δομημένους πίνακες δεδομένων (dataframe) και τη χρήση των μεθόδων που προσφέρει.

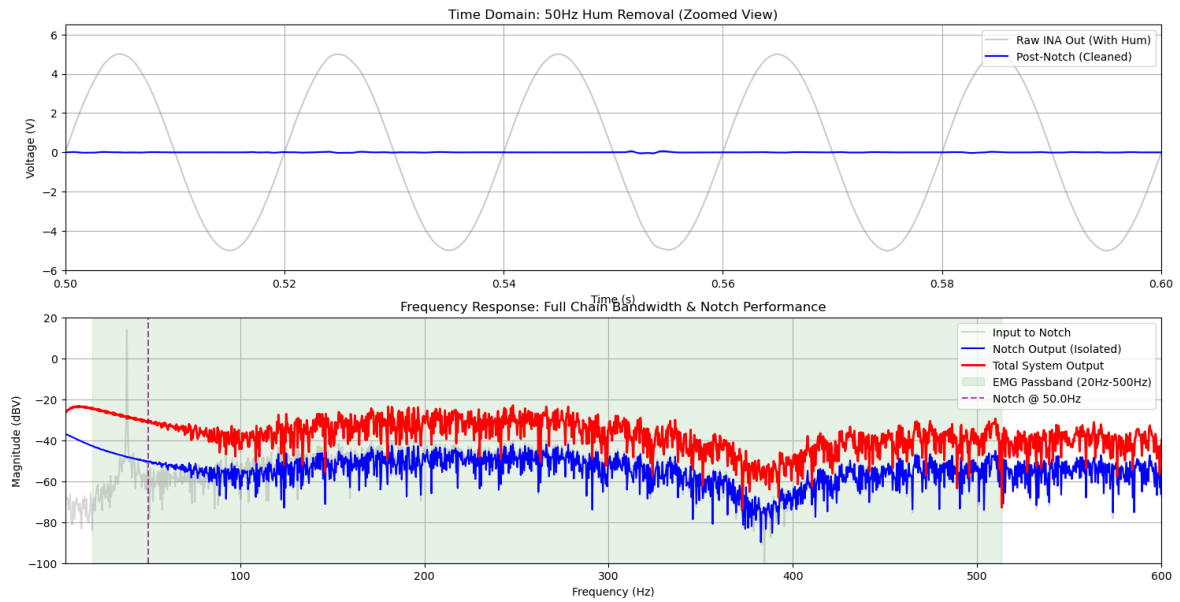
Η εισαγωγή και χρήση του πακέτου είναι εξαιρετικά εύκολη και περιγράφεται εν συντομία από το παρακάτω απόσπασμα κώδικα:

```
import pandas as pd
pd.DataFrame({'A': [1, 2, 3]})
```

3.2.3 Βιβλιοθήκη Matplotlib

Η Matplotlib αποτελεί το πλέον θεμελιώδες και καθιερωμένο εργαλείο της γλώσσας Python για την οπτικοποίηση δεδομένων. Πρόκειται για μια εκτενή βιβλιοθήκη που επιτρέπει τη δημιουργία στατικών, κινούμενων και διαδραστικών γραφημάτων υψηλής ακρίβειας και ακαδημαϊκού επιπέδου. Βασίζεται στενά στη δομή των πινάκων της NumPy, ενώ το υπο-πακέτο matplotlib.pyplot προσφέρει μια εύχρηστη διεπαφή προγραμματισμού, η οποία προσομοιώνει τη λογική σχεδίασης γραφημάτων γνωστών μαθηματικών πακέτων, όπως το MATLAB [26].

Στο πλαίσιο της παρούσας Διπλωματικής Εργασίας, η δυνατότητα απεικόνισης γραφικών για τα καταγεγραμμένα ηλεκτρομυογραφικά σήματα είναι βασικό κομμάτι τόσο της προσομοίωσης όσο και της τελικής υλοποίησης. Η άμεση παράθεση του θορυβώδους, ακατέργαστου σήματος ακριβώς πάνω από το καθαρισμένο, μάς δίνει μια ξεκάθαρη εικόνα για το αν έχει αφαιρεθεί επιτυχώς ο θόρυβος. Τα παραγόμενα γραφήματα λειτουργούν και σαν ένας ποιοτικός έλεγχος, πριν γίνει εκτεταμένη χρήση των δεδομένων.



Σχήμα 3.1 Παράδειγμα γραφήματος από Matplotlib

Το παρακάτω απόσπασμα κώδικα του παραρτήματος που προέρχεται από το λογισμικό της προσομοίωσης των ηλεκτρομυογραφικών σημάτων και του κυκλώματος για την επεξεργασία τους:

```
import matplotlib.pyplot as plt

...

# --- PLOTS ---

fig, (ax1, ax2) = plt.subplots(2, 1, figsize=(12, 10))

    ax1.plot(analysis.time, analysis['ina_out'], color='gray', alpha=0.4,
label='Raw INA Out (With Hum)')

    ax1.plot(analysis.time, analysis['notch_out'], color='blue', label='Post-
Notch (Cleaned)')

    ax1.set_xlim(0.5, 0.6)

    ax1.set_title('Time Domain: 50Hz Hum Removal (Zoomed View)')

    ax1.set_xlabel('Time (s)')

    ax1.set_ylabel('Voltage (V)')

    ax1.legend()

    ax1.grid(True)

    ax2.plot(f_pre, 20*np.log10(m_pre + 1e-9), color='gray', alpha=0.3,
label='Input to Notch')

    ax2.plot(f_post, 20*np.log10(m_post + 1e-9), color='blue', linewidth=1.5,
label='Notch Output (Isolated)')
```

```

ax2.plot(f_final, 20*np.log10(m_final + 1e-9), color='red', linewidth=2,
label='Total System Output')

f_low = 19.9
f_high = 513.4
ax2.axvspan(f_low, f_high, color='green', alpha=0.1, label='EMG Passband
(20Hz-500Hz)')

ax2.set_xlim(5, 600)
ax2.set_ylim(-100, 20) # Elevated ceiling to accommodate the final 11x gain
ax2.axvline(x=target_hz, color='purple', linestyle='--', alpha=0.7,
label=f'Notch @ {target_hz}Hz')
ax2.set_title('Frequency Response: Full Chain Bandwidth & Notch Performance')
ax2.set_xlabel('Frequency (Hz)')
ax2.set_ylabel('Magnitude (dBV)')
ax2.legend(loc='upper right')
ax2.grid(True, which='both')

plt.tight_layout()
plt.show()

```

3.2.4 Βιβλιοθήκη PySpice

Η βιβλιοθήκη PySpice καθιστά εφικτή τη γεφύρωση ανάμεσα στη προσομοίωση ηλεκτρονικών κυκλωμάτων και στον προγραμματισμό με τη χρήση της γλώσσας Python. Ειδικότερα, το συγκεκριμένο πακέτο προσφέρει μια προγραμματιστική διεπαφή, η οποία χρησιμοποιείται από τους πλέον καθιερωμένους προσομοιωτές στη βιομηχανία, όπως ο NgSpice και ο Xyce. Περαιτέρω, δίνει στον εκάστοτε μηχανικό τη δυνατότητα να σχεδιάζει, να παραμετροποιεί και να τρέχει αναλύσεις κυκλωμάτων χρησιμοποιώντας καθαρό αντικειμενοστραφή κώδικα Python. [27]

Το πακέτο αυτό στα πλαίσια της παρούσας εργασίας αξιοποιήθηκε εκτενώς κατά τον σχεδιασμό και την ανάλυση του αναλογικού κυκλώματος. Πριν την τελική υλοποίηση, έπρεπε να επιβεβαιωθεί η θεωρητική ανταπόκριση των φίλτρων κατά την επεξεργασία και ενίσχυση των EMG σημάτων. Μέσω αυτής της βιβλιοθήκης έγινε εκτέλεση προσομοιώσεων τόσο στο πεδίο του χρόνου (Transient Analysis) όσο και στο πεδίο της συχνότητας (AC Analysis) και στη συνέχεια με τη χρήση των δύο προαναφερθέντων βιβλιοθηκών έγινε η οπτικοποίηση των αποτελεσμάτων.

Το ακόλουθο απόσπασμα που ακολουθεί απεικονίζει ένα κομμάτι του παραρτήματος του κώδικα της προσομοίωσης με τη χρήση του πακέτου PySpice:

```

from PySpice.Spice.Netlist import Circuit

```

```

from PySpice.Unit import *

...
circuit = Circuit('EMG Circuit')

circuit.include('INA128.LIB')
circuit.include('TL072.LIB')

...
# circuit.X(Name, Model, In+, In-, V+, V-, Out, Ref, RG1, RG2)
circuit.X('U1', 'INA128', 'ina_in_pos', 'ina_in_neg', 'ina_vcc', 'ina_vee',
'ina_out', circuit.gnd, 'ina_gain_node_1', 'ina_gain_node_2')

# circuit.X(Name, Model, In+, In-, V+, V-, Out)
circuit.X('U2', 'TL072', 'tl_in_pos', 'tl_in_neg', 'tl_vcc', 'tl_vee', 'tl_out')
circuit.X('U3', 'TL072',
'notch_in_pos', 'notch_out', 'tl_vcc', 'tl_vee', 'notch_out')
circuit.X('U4', 'TL072',
'notch_feedback_in_pos', 'notch_feedback_out', 'tl_vcc', 'tl_vee', 'notch_feedback_o
ut')

# Grounding the subject
circuit.SinusoidalVoltageSource('V_BONE', 'electrode', circuit.gnd,
amplitude=0.05@u_V, frequency=50@u_Hz) # type: ignore

# First Stage INA128 instrumentation amplifier
# Gain Resistor
circuit.R('RG', 'ina_gain_node_1', 'ina_gain_node_2', 505@u_0hm) # type: ignore

# IC Supply
circuit.V('INA_VCC', 'ina_vcc', circuit.gnd, 10@u_V) # type: ignore
circuit.V('INA_VEE', 'ina_vee', circuit.gnd, -10@u_V) # type: ignore
circuit.V('TL_VCC', 'tl_vcc', circuit.gnd, 10@u_V) # type: ignore
circuit.V('TL_VEE', 'tl_vee', circuit.gnd, -10@u_V) # type: ignore

# IC Input
circuit.PiecewiseLinearVoltageSource('ina_vminus', 'ina_in_neg', circuit.gnd,
values=negative_input_pwl_values)

```

```

circuit.PieceWiseLinearVoltageSource('ina_vplus', 'ina_in_pos', 'electrode',
values=positive_input_pwl_values)

# Notch Filter
circuit.C('notch_c1','ina_out','c_middle',0.1@u_uF) # type: ignore
circuit.C('notch_c2','c_middle','notch_network_out',0.1@u_uF) # type: ignore
circuit.R('notch_r1','c_middle',circuit.gnd,15.91@u_kOhm) # type: ignore

circuit.R('notch_r2','ina_out','r_middle',31.83@u_kOhm) # type: ignore
circuit.R('notch_r3','r_middle','notch_network_out',31.83@u_kOhm) # type: ignore
circuit.C('notch_c3','r_middle',circuit.gnd,0.2@u_uF) # type: ignore

circuit.R('tie_notch_to_amp','notch_network_out','notch_in_pos',1@u_0hm) # type:
ignore
# Notch Feedback
circuit.R('notch_feedback_r1','notch_out','notch_feedback_middle',10@u_kOhm) #
type: ignore
circuit.R('notch_feedback_r2','notch_feedback_middle',circuit.gnd,90@u_kOhm) #
type: ignore
...

```

3.2.5 Βιβλιοθήκη Scikit-Learn

Η βιβλιοθήκη scikit-learn (γνωστή και ως sklearn) είναι από τις πιο διαδεδομένες βιβλιοθήκες για την υλοποίηση αλγορίθμων μηχανικής μάθησης με την γλώσσα προγραμματισμού Python. Χρησιμοποιεί τις βιβλιοθήκες NumPy, SciPy και Matplotlib, οι οποίες αναλύθηκαν προηγουμένως, θέτοντάς την εξαιρετικά χρήσιμη για μαθηματικούς υπολογισμούς και οπτικοποίηση δεδομένων.

Η συγκεκριμένη βιβλιοθήκη χρησιμοποιήθηκε για την συγγραφή του κώδικα του μοντέλου μηχανικής μάθησης που χρησιμοποιείτε στην υλοποίηση της εργασίας. Πιο συγκεκριμένα, δημιουργήθηκε ένα μοντέλο κατηγοριοποίησης, για την ανίχνευση της κίνησης που λαμβάνει χώρα μέσω των χαρακτηριστικών που θα αναλυθούν στο κεφάλαιο τέσσερα. Πέρα από την κατηγοριοποίηση ωστόσο, η βιβλιοθήκη αυτή περιλαμβάνει μια πληθώρα αλγορίθμων όπως Δέντρα Απόφασης, Τυχαία Δάση, k-Κοντινότεροι Γείτονες και πολλά ακόμα. Δίνει επίσης την δυνατότητα στον χρήστη να αξιολογήσει την αποδοτικότητα του μοντέλου που δημιούργησε μέσω της παραγωγής στατιστικών που θα αναλυθούν και αυτά στο τέταρτο κεφάλαιο.

Παρακάτω ακολουθεί ένα απλό παράδειγμα για την χρήση της βιβλιοθήκης:

```

from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split
from sklearn.neighbors import KNeighborsClassifier

```

```

from sklearn.metrics import accuracy_score

iris = load_iris()
X, y = iris.data, iris.target

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42)

clf = KNeighborsClassifier(n_neighbors=3)
clf.fit(X_train, y_train)

predictions = clf.predict(X_test)

print(f"Model Accuracy: {accuracy_score(y_test, predictions) * 100:.2f}%")

```

3.3 Εργαλεία Μέτρησης: Picoscope και το λογισμικό Picoscope 7

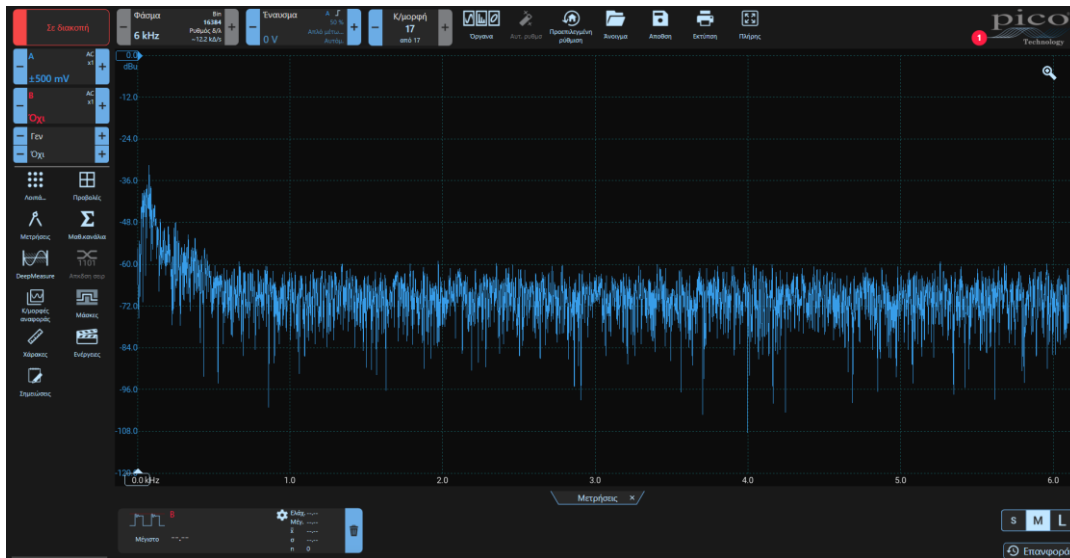
Η σειρά παλμογράφων Picoscope της Pico Technologies προσφέρει ένα μεγάλο φάσμα λειτουργιών σε ένα πολύ μικρού μεγέθους εργαστηριακό εργαλείο. Πιο συγκεκριμένα, έγινε χρήση του Picoscope 2204A, ο οποίος συνδέεται στον υπολογιστή με USB και έχει εύρος ζώνης 10MHz, έχει δύο αναλογικά κανάλια, μέγιστο ρυθμό δειγματοληψίας 100MS/s, καθώς και μια γεννήτρια συχνότητας. Το μικρό του μέγεθος και η προσιτή του τιμή τον καθιστούν κατάλληλο για ανάπτυξη πρωτοτύπων εκτός βιομηχανικών εγκαταστάσεων.



Σχήμα 3.2 Picoscope 2204A

Μαζί με τον παλμογράφο έρχεται και το λογισμικό που χρειάζεται για να λειτουργήσει, το Picoscope 7. Μέσω του συγκεκριμένου προγράμματος, μπορεί ο χρήστης πολύ εύκολα να παραμετροποιήσει τις λειτουργίες του με όσες επιλογές είναι διαθέσιμες και στους παραδοσιακούς παλμογράφους πάγκου εργασίας. Η κύρια διαφορά είναι πως μπορεί να γίνει πολύ εύκολα αποθήκευση των δεδομένων που

ανιχνεύει ο παλμογράφος στον υπολογιστή καθώς είναι συνδεδεμένος σε αυτόν. Ακόμα η διεπαφή χρήστη της εφαρμογής είναι πολύ πρακτική και γίνεται κατανοητή γρήγορα.



Σχήμα 3.3 Λογισμικό Picoscope 7

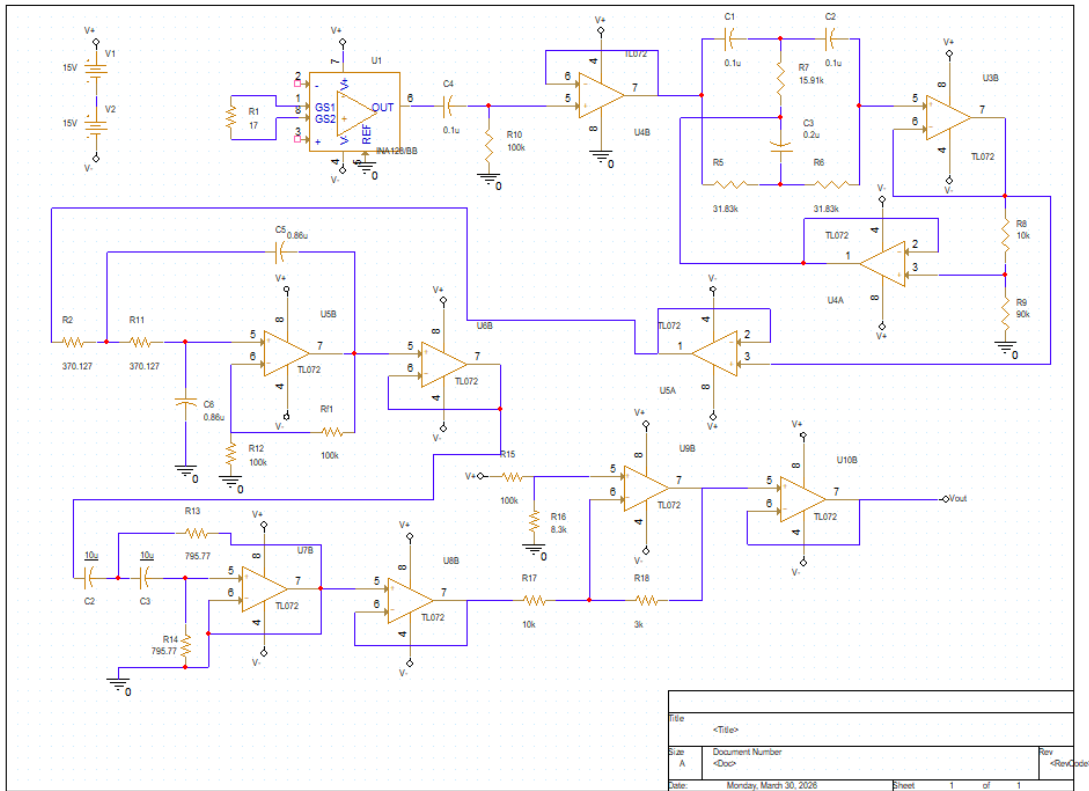
Ο ρόλος του συγκεκριμένου εργαλείου στα πλαίσια αυτής της Διπλωματικής Εργασίας ήταν καθοριστικής σημασίας. Αναλυτικότερα, αξιοποιήθηκε για την ανάλυση του κατασκευασμένου υλικού και την επιβεβαίωση των λειτουργιών του. Ακόμα, έγινε δειγματοληψία για συλλογή δεδομένων που έπειτα χρησιμοποιήθηκαν για την εκπαίδευση του μοντέλου μηχανικής μάθησης.

3.4 Ανάλυση Αναλογικών Φίλτρων και Ενίσχυση του Σήματος

Για την ακριβή ερμηνεία των ηλεκτρομυογραφικών σημάτων, όπως προαναφέρθηκε, πρέπει να λάβει μέρος το φιλτράρισμα κάποιων συχνοτήτων. Πιο συγκεκριμένα, τα EMG σήματα δεν φέρουν κάποια χρήσιμη πληροφορία κάτω των 10Hz και πάνω των 500Hz. Για τον λόγο αυτόν κατασκευάστηκαν ένα υπερπερατό και ένα χαμηλοπερατό φίλτρο για την απαλοιφή αυτών των συχνοτήτων. Ακόμα, οι παρεμβολές που υπάρχουν λόγω του δικτύου ηλεκτροδότησης, οι οποίες βρίσκονται στα 50Hz, πρέπει να αφαιρεθούν για την ορθή καταγραφή ηλεκτρομυογραφικών σημάτων.

Τα σήματα που παράγουν οι μύες από τη φύση έχουν πολύ μικρό πλάτος και για αυτό τον λόγο χρήζουν ενίσχυσης. Ειδικότερα, τα φίλτρα που αναφέρθηκαν είναι όλα ενεργά, δίνοντας ανά βαθμίδα μια μικρή ενίσχυση στο σήμα. Η μεγαλύτερη ενίσχυση όμως, συμβαίνει στη πρώτη βαθμίδα του κυκλώματος, με τη χρήση ενός ενισχυτή οργανολογίας INA128, για να γίνει αξιοποίηση του CMRR (Common Mode Rejection Ratio).

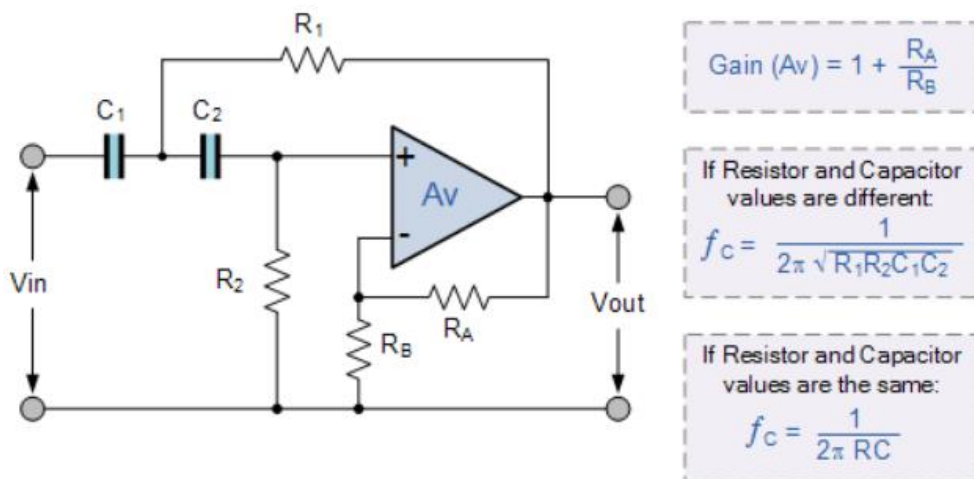
Όπως μπορεί να παρατηρηθεί στη παρακάτω εικόνα (Σχήμα 3.4), στη τελική του μορφή το αναλογικό κύκλωμα αποτελείται από τρία ενεργά φίλτρα και κάθε βαθμίδα είναι συνδεδεμένη με έναν ακολουθητή τάσης (buffer) για την απομόνωσή τους, εμποδίζοντας έτσι την αμέσως επόμενη βαθμίδα να αλλοιώσει τα χαρακτηριστικά της απόκρισης της προηγούμενης.



Σχήμα 3.4 Τελικό σχηματικό του κυκλώματος

3.4.1 Υψηλεπατό Φίλτρο (Highpass Filter)

Σε πρώτη φάση της σχεδίασης έπρεπε να γίνει σχεδίαση ενός υψηλεπατού φίλτρου με συχνότητα αποκοπής (f_c) τα 20Hz. Πιο συγκεκριμένα, για τη μεγιστοποίηση του ρυθμού αποκοπής αξιοποιήθηκε τοπολογία φίλτρου δεύτερης τάξεως όπου οι προς αφαίρεση συχνότητες χάνουν την ισχύ τους κατά 40dB/δεκάδα.



Σχήμα 3.5 Υψηλεπατό φίλτρο δεύτερου βαθμού [28]

Ο υπολογισμός των τιμών των στοιχείων της τοπολογίας έγινε με τη προϋπόθεση πως οι άγνωστες αντιστάσεις R1 και R2 έχουν την ίδια τιμή R, η οποία ισούται με 795.77Ω καθώς και οι πυκνωτές C1 και C2 έχουν επίσης την ίδια γνωστή τιμή των 10μF:

$$f_c = \frac{1}{2\pi RC} \quad (3.1)$$

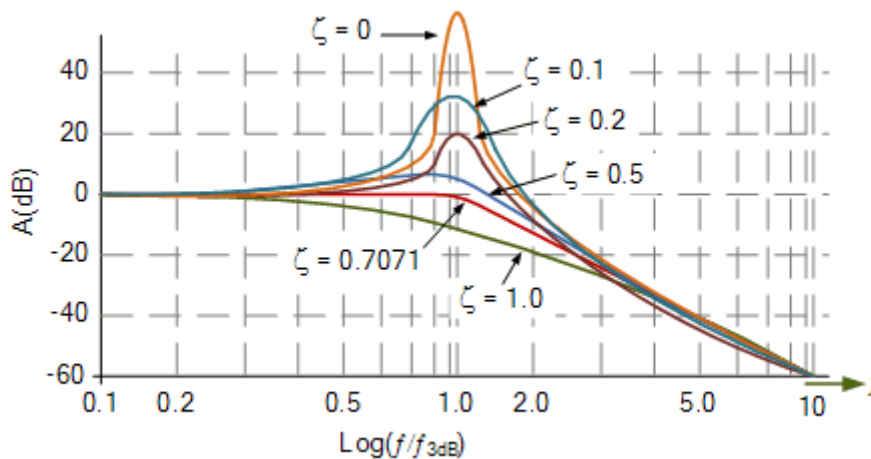
Η ενίσχυση (A_v) σε αυτή τη βαθμίδα είναι 1 και ο συντελεστή ποιότητας (Q) έχει τιμή 0.5:

$$A_v = 3 - \frac{1}{Q} \quad (3.2)$$

και με συντελεστή απόσβεσης (ζ) ίσο με 1:

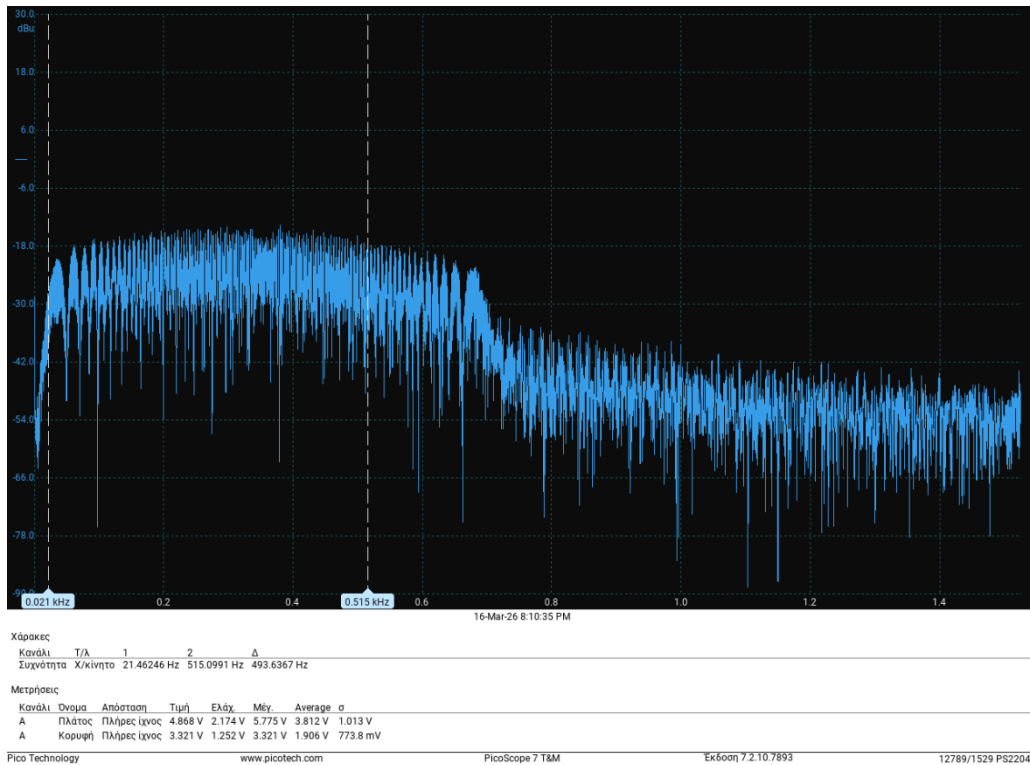
$$\zeta = \frac{1}{2Q} \quad (3.3)$$

Οι τιμές αυτές των παραπάνω συντελεστών περιγράφουν την απόσβεση του σήματος από την ζώνη διέλευσης προς την ζώνη αποκοπής. Για συντελεστή απόσβεσης ίσο με 1 το σήμα μεταβαίνει ομαλά μεταξύ αυτών των δύο ζωνών χωρίς να προκαλεί ταλαντώσεις ή υπερύψωση στο πεδίο του χρόνου [28].



Σχήμα 3.6 Απόσβεση σήματος σε φίλτρα δευτέρου βαθμού [28]

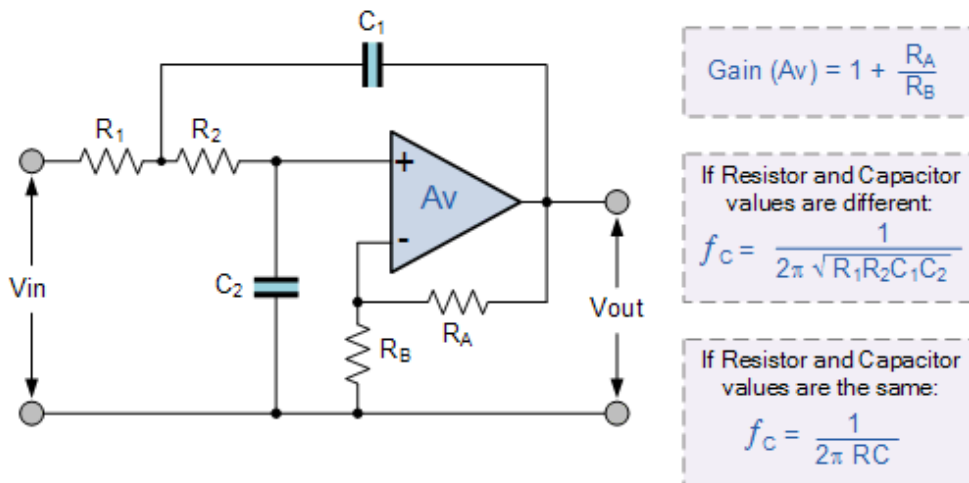
Όπως διακρίνεται στο παρακάτω σχήμα (Σχήμα 3.7), κατά την πρακτική του υλοποίησης, το υπερπαρατό φίλτρο αποκόπτει απότομα τις συχνότητες κάτω από 20Hz να εισέλθουν στο τελικό στάδιο της επεξεργασίας του σήματος.



Σχήμα 3.7 Καταγραφή EMG σήματος με bandpass και notch φίλτρα

3.4.2 Χαμηλοπερατό Φίλτρο (Lowpass Filter)

Εν συνεχεία κρίθηκε απαραίτητη η υλοποίηση χαμηλοπερατού φίλτρου με συχνότητα αποκοπής (f_c) τα 500Hz. Αναλυτικότερα, έγινε και εδώ χρήση τοπολογίας φίλτρου δεύτερης τάξης για την υλοποίησή του.



Σχήμα 3.8 Χαμηλοπερατό φίλτρο δευτέρου βαθμού [28]

Για τον υπολογισμό των τιμών των στοιχείων του συγκεκριμένου αναλογικού φίλτρου, έγινε η υπόθεση πως οι αντιστάσεις R_1 και R_2 έχουν την ίδια τιμή R η οποία ισούται με 370.12Ω . Ομοίως οι πυκνωτές C_1 και C_2 μοιράζονται την ίδια τιμή των $0.86\mu\text{F}$ βάσει της εξίσωσης (3.1).

Στη συγκεκριμένη βαθμίδα οι αντιστάσεις που ρυθμίζουν την ενίσχυση (R_A και R_B) έχουν επίσης την ίδια τιμή των $100k\Omega$ αποδίδοντας ενίσχυση (A_v) με τιμή 2:

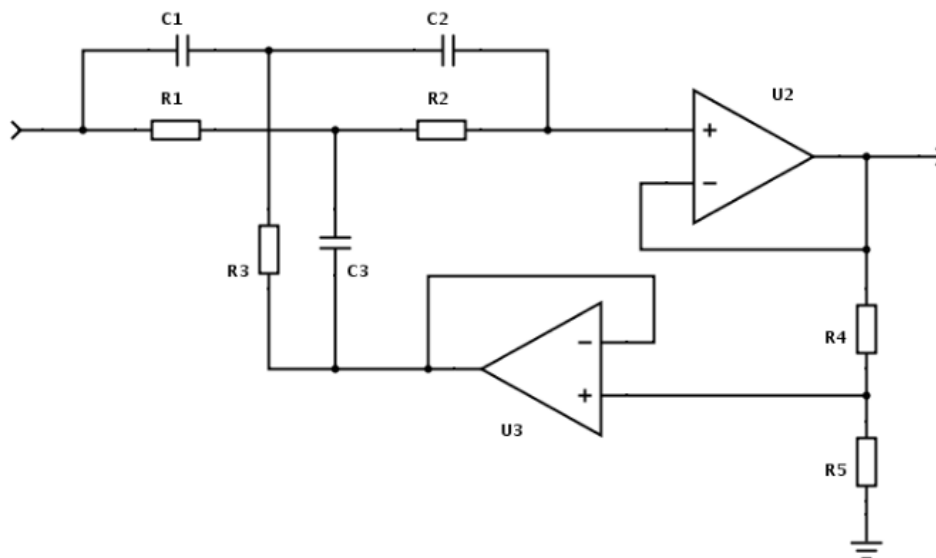
$$A_v = 1 + \frac{R_A}{R_B} \quad (3.4)$$

Οι τιμές αυτές δίνουν συντελεστή ποιότητας (Q) ίσο με 1 και συντελεστή απόσβεσης (ζ) ίσο με 0.5 βάσει των εξισώσεων (3.2) και (3.3). Λόγω της τιμής του συντελεστή απόσβεσης η συγκεκριμένη βαθμίδα παρουσιάζει κορυφώσεις γύρω από τη συχνότητα αποκοπής. Στα ενεργά φίλτρα για να διατηρηθεί η ευστάθειά τους προτείνεται να μην έχουν ενίσχυση πάνω από 3 [28].

3.4.3 Φίλτρο Εγκοπής (Notch filter)

Το τελευταίο φίλτρο που υλοποιήθηκε είναι ένα φίλτρο εγκοπής για την παρεμπόδιση της διάδοσης των παρεμβολών του δικτύου ηλεκτροδότησης στα 50Hz. Αναλυτικότερα, έγινε χρήση της τοπολογίας Twin-T με ανάδραση για την αποτελεσματική εξουδετέρωση στη συχνότητα αποκοπής και γύρω από αυτήν. Η συγκεκριμένη τοπολογία είναι ασταθής στην περίπτωση που τα στοιχεία έχουν ακόμα και μια πολύ μικρή απόκλιση από τις υπολογισμένες τιμές, αλλά είναι εξαιρετικά αποτελεσματική στον αποκλεισμό συγκεκριμένων συχνοτήτων.

Η ανάδραση του φίλτρου χρησιμεύει στην βελτίωση του συντελεστή ποιότητας, με αποτέλεσμα να δημιουργεί μία πιο στενή ζώνη αποκοπής. Με αυτό τον τρόπο χάνεται λιγότερη πληροφορία από τις συχνοτήτες γύρω από αυτή της αποκοπής.



Σχήμα 3.9 Φίλτρο εγκοπής με τοπολογία Twin-T [29]

Για τον υπολογισμό των τιμών των πυκνωτών της συγκεκριμένης βαθμίδας έγινε υπόθεση αρχικά πως οι C_1 και C_2 έχουν την ίδια τιμή C των $0.1\mu F$. Εφόσον, η τιμή του C είναι γνωστή τότε μπορεί να υπολογιστεί και η τιμή του C_3 , η οποία αναλογεί στη διπλάσια αυτής των άλλων δύο οπότε θα είναι $0.2\mu F$:

$$C_1 = C_2 = C \quad (3.5)$$

$$C_3 = 2 * C \quad (3.6)$$

Η κοινή τιμή R για τις αντιστάσεις R_1 και R_2 μπορεί εύκολα να βρεθεί με τη χρήση της εξίσωσης 3.1 και είναι 31.83kΩ και για τις δύο. Σε αντίθεση με τον υπολογισμό του τρίτου πυκνωτή που έχει διπλάσια τιμή σε σχέση με τους άλλους δύο, η τρίτη αντίσταση έχει τιμή ανάλογη του μισού της κοινής τιμής των προηγούμενων, οπότε 15.91kΩ:

$$R_1 = R_2 = R \quad (3.7)$$

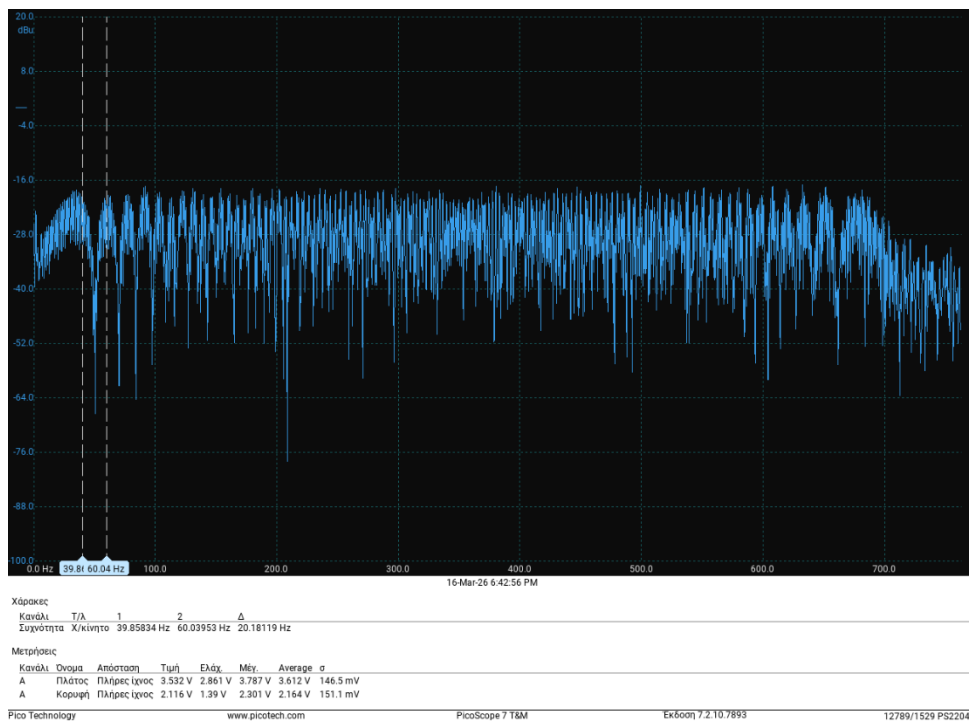
$$R_3 = \frac{R}{2} \quad (3.8)$$

Οι αντιστάσεις R_4 και R_5 χρησιμοποιούνται για τη ρύθμιση του συντελεστή ποιότητας του φίλτρου. Πιο συγκεκριμένα, ο λόγος $\frac{R_5}{R_4}$ είναι άμεσα συνδεδεμένος με τον υπολογισμό του Q και καθορίζει το εύρος ζώνης της αποκοπής. Στη συγκεκριμένη υλοποίηση οι αντιστάσεις αυτές έχουν τιμές 10kΩ και 90kΩ αντίστοιχα, οι οποίες αποδίδουν συντελεστή ποιότητας ίσο με 2.5 και εύρος ζώνης (BW) 20Hz. [30].

$$Q = \frac{1 + \frac{R_5}{R_4}}{4} \quad (3.9)$$

$$BW = \frac{f_c}{Q} \quad (3.10)$$

Η παρακάτω εικόνα (fig. 21) απεικονίζει το φίλτρο σε λειτουργία και γίνεται εύκολα αντιληπτή η αποτελεσματικότητά του. Πιο συγκεκριμένα, η απότομη αποκοπή από τα 40Hz μέχρι και τα 60Hz εξουδετερώνει επιτυχώς την συχνότητα των 50Hz χωρίς να χαθεί επιπρόσθετη πληροφορία από το σήμα EMG.



Σχήμα 3.10 Λειτουργία του υλοποιημένου φίλτρου εγκοπής

3.4.4 Βαθμίδες Ενίσχυσης

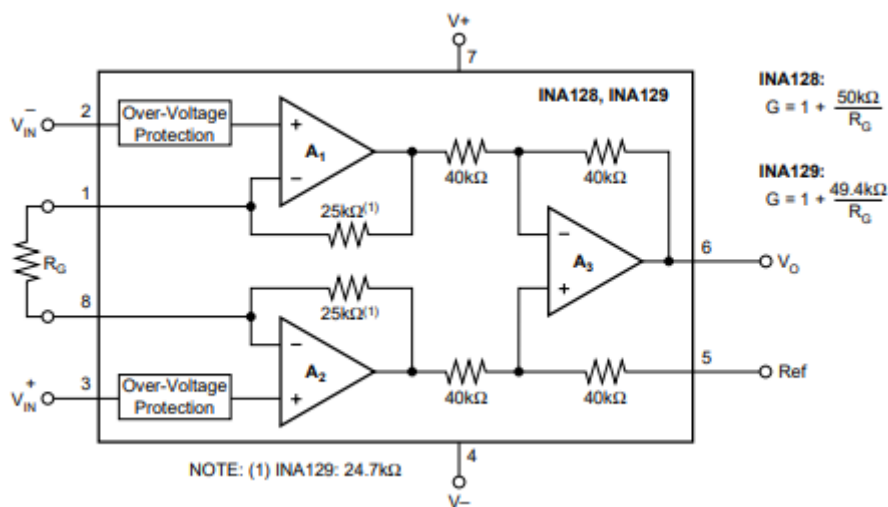
Η πρώτη και κύρια βαθμίδα ενίσχυσης γίνεται κατά την είσοδο του ηλεκτρομυογραφικού σήματος στο κύκλωμα. Ειδικότερα, ένας ενισχυτής οργανολογίας INA128 με μία αντίσταση κέρδους 17Ω (R_G) επιφέρει στο σήμα μία αρχική ενίσχυση κατά 2942.17 [31]:

$$A_v = 1 + \frac{50k\Omega}{R_G} \quad (3.11)$$

Αν και το νούμερο είναι σχετικά μεγάλο, τα EMG σήματα έχουν στην ακραία μορφή τους πλάτος $\pm 5mV$, κυρίως σε αθλητές, οπότε ο INA128 θα έχει σαν έξοδο $\pm 14.71V$ το οποίο καλύπτεται από την τροφοδοσία των $\pm 15V$, αποτρέποντας τη μετάβαση του ενισχυτή σε κατάσταση κόρου. [31].

$$V_{out} = A_v * V_{in} \quad (3.12)$$

Έγινε η επιλογή της μεγάλης ενίσχυσης κατά την είσοδο του σήματος στο κύκλωμα διότι λόγω των φίλτρων που υλοποιήθηκαν, και των παθητικών ηλεκτρονικών στοιχείων από τα οποία είναι φτιαγμένα, το σήμα χάνει αρκετά μεγάλο ποσοστό του πλάτους του. Περαιτέρω, η ενίσχυση γίνεται στη διαφορά δύο σχεδόν όμοιων ηλεκτρομυογραφικών σημάτων με την ειδοποιό διαφορά να είναι να προπορεύεται κατά κάποια ms το ένα από το άλλο, αξιοποιώντας έτσι το λόγο απόρριψης κοινού τρόπου (Common-Mode Rejection Ratio - CMRR). Βάσει του τεχνικού φυλλαδίου του INA128 για ενίσχυση μεγαλύτερη του 100 το CMRR έχει τιμή 120dB το οποίο είναι εξαιρετικά καλό για την απαλοιφή παρασίτων που υπάρχουν στα δύο σήματα εισόδου [31].



Σχήμα 3.11 Σχηματικό του INA128 [31]

Το δεύτερο στάδιο ενίσχυσης λαμβάνει χώρα στη βαθμίδα στην οποία υλοποιείται το ενεργό χαμηλοπερατό φίλτρο. Πιο συγκεκριμένα, οι αντιστάσεις για τη ρύθμιση του κέρδους σε αυτή τη βαθμίδα έχουν την ίδια τιμή των $100k\Omega$, με αποτέλεσμα να γίνεται μια μικρή ενίσχυση κατά 2, καθώς όπως αναλύθηκε και στο κεφάλαιο 3.4.2, με σκοπό τη διατήρηση της ευστάθειας του φίλτρου, προτείνεται μια μέγιστη ενίσχυση κατά τρία.

$$A_v = 1 + \frac{R_f}{R_{in}} \quad (3.13)$$

Έγινε επιλογή της συγκεκριμένης βαθμίδας για ενίσχυση, καθώς στη πρακτική υλοποίηση του κυκλώματος παρατηρήθηκαν σημαντικές απώλειες στο πλάτος του σήματος μετά το φίλτρο εγκοπής, το οποίο προηγείται του χαμηλοπερατού φίλτρου.

3.5 Μικροελεγκτής ESP32 και Στάδιο Προσαρμογής Σήματος (Level Shifting)

Ο μικροελεγκτής ESP32 αποτελεί μία από τις πιο διαδομένες πλατφόρμες ανάπτυξης προτύπων. Πιο συγκεκριμένα, η ενσωμάτωση ενός μικροεπεξεργαστή Tensilica Xtensa LX6 συντελεί στη χρησιμότητά του και το θέτει αρκετά πρακτικό για τη ταχεία ανάπτυξη ακόμα και πιο δύσκολων εργασιών. Ο συγκεκριμένος μικροεπεξεργαστής έχει δύο πυρήνες, χαμηλή κατανάλωση ενέργειας και αρκετά υψηλή ταχύτητα χρονισμού, στα 240MHz. Ακόμα, ο μικροελεγκτής αυτός έχει ενσωματωμένα συστήματα που επιτρέπουν τόσο την άμεση σύνδεση στο διαδίκτυο όσο και τη σύνδεση Bluetooth. Οι συγκεκριμένες λειτουργίες μπορούν να φανούν χρήσιμες σε μία μελλοντική έκδοση του συστήματος, για τη λήψη τηλεμετρίας [32].



Σχήμα 3.12 Μικροελεγκτής ESP32 [33]

Πέρα από τη τεχνική του υπεροχή σε σχέση με άλλους μικροελεγκτές της ίδια τάξης, ο ESP32 διαθέτει και 2 μονάδες μετατροπής αναλογικού σήματος σε ψηφιακό (Analog to Digital Converter - ADC), οι οποίες μοιράζονται αντίστοιχα σε 18 αναλογικά κανάλια εισόδου. Οι μετατροπείς αυτοί έχουν ανάλυση 12bit, γεγονός που τους επιτρέπει την αντιστοίχιση της αναλογικής τάσης εισόδου σε ψηφιακές τιμές από 0 έως 4095 και με διακριτική ικανότητα περίπου 0.8mV. Ακόμα, σαν είσοδο δέχονται σήματα από 0V έως 3.3V, αλλιώς σήματα που δεν βρίσκονται σε αυτά τα όρια πλάτους υπάρχει περίπτωση να οδηγήσουν στον ψαλιδισμό τους ή ακόμα και να δημιουργήσουν βλάβη στον ίδιο τον μικροελεγκτή.

$$LSB = \frac{V_{ref}}{2^N - 1} \quad (3.14)$$

Για τη σωστή είσοδο του ηλεκτρομυογραφικού σήματος στον μικροελεγκτή έγινε πόλωση ενός τελεστικού ενισχυτή απλής τροφοδοσίας στη τελική βαθμίδα του κυκλώματος (Σχήμα 3.4). Αρχικά, βάσει τις εργαστηριακές μετρήσεις που έλαβαν μέρος η μέγιστη έξοδος του κυκλώματος επεξεργασίας του σήματος EMG ήταν γύρω στα 10V. Δεδομένου του πλάτους του σήματος εισόδου σχεδιάστηκε ο κλάδος ανατροφοδότησης του τελεστικού ενισχυτή με στόχο ο τελευταίος να επιφέρει υπο-ενίσχυση του σήματος κατά 0.3. Για την επίτευξη της συγκεκριμένης τιμής έγινε χρήση δύο αντιστάσεων 10kΩ και 3kΩ αντίστοιχα. Ακόμα, για την υπερύψωση του σήματος σε επίπεδο ώστε να βρίσκεται στο όριο των 0V-3.3V, δημιουργήθηκε μια τάση αναφοράς στη θετική είσοδο του ενισχυτή στα 1.3V από τροφοδοσία των 15V.

$$A_{vcl(1)} = 1 + \frac{R_2}{R_1} \quad (3.15)$$

$$A_{vcl(2)} = -\frac{R_2}{R_1} \quad (3.16)$$

$$V_{ref} = \frac{R_4}{R_3 + R_4} V_{CC} \quad (3.17)$$

Λόγω της ανοχής των υλικών που χρησιμοποιήθηκαν για την υλοποίηση του κυκλώματος παρατηρήθηκε πως η τάση αναφοράς είχε τιμή γύρω στα 1.65V, γεγονός που δεν δημιούργησε πρόβλημα στη οριοθέτηση του σήματος. Ακόμα, λόγω του ότι ο τελεστικός ενισχυτής δέχεται το σήμα EMG στην αρνητική του είσοδο, οι καταγραφές που χρησιμοποιήθηκαν για την εκπαίδευση του μοντέλου μηχανικής μάθησης καθώς και η είσοδος στον μικροελεγκτή, αποτελούν την αναστροφή του αρχικού σήματος.

Για τον ορθό υπολογισμό των χαρακτηριστικών πάνω στα οποία έγινε η εκπαίδευση του μοντέλου μηχανικής μάθησης έγινε η αφαίρεση της σταθερής τάσης γύρω από την οποία έγινε η υπερύψωση του σήματος. Ακόμα, έγινε η μετατροπή των ψηφιακών τιμών που έχουν ως έξοδο τα κανάλια ADC του ESP32 σε τιμές αντίστοιχες του αναλογικού σήματος για τον ίδιο σκοπό. Περαιτέρω, τέθηκε σε χρήση ρουτίνα εξυπηρέτησης διακοπής κάθε 1ms για συχνότητα των 1000Hz.

```

...
void IRAM_ATTR onTimer() {
    if (!windowReady) {
        int rawValue = analogRead(EMG_PIN);

        // Convert 12-bit ADC directly to Voltage
        double voltage = (rawValue / 4095.0) * 3.3;

        emgBuffer[bufferIndex] = voltage;
        bufferIndex++;
    }
    ...
    timer = timerBegin(1000000); // 1MHz base clock (1 microsecond per tick)
    timerAttachInterrupt(timer, &onTimer);
    timerAlarm(timer, TIMER_INTERVAL_US, true, 0); // Trigger every 1000us, auto-
    reload
    ...

    // 2. Center the signal (Remove DC Offset dynamically)
    double sum = 0;
    for (int i = 0; i < WINDOW_SIZE; i++) sum += localWindow[i];
    double offset = sum / WINDOW_SIZE;

    for (int i = 0; i < WINDOW_SIZE; i++) {
        localWindow[i] = localWindow[i] - offset;
    }
}

```

```
}  
...  
}
```

3.6 Επίλογος

Συνοψίζοντας, τα ηλεκτρομυογραφικά σήματα απαιτούν εκτενή επεξεργασία, ώστε να θεωρηθούν έτοιμα προς καταγραφή. Στο παρόν κεφάλαιο αρχικά έγινε μια περιγραφή της γλώσσας προγραμματισμού Python και των βιβλιοθηκών που αξιοποιήθηκαν για τη συγγραφή τόσο του λογισμικού της αναλυτικής προσομοίωσης όσο και του λογισμικού της τελικής πρακτικής υλοποίησης του συστήματος. Ακόμα, έγινε παρουσίαση του παλμογράφου Picoscope και του λογισμικού Picoscope 7, τα οποία ήταν ζωτικής σημασίας στη γοργή ανάπτυξη αφενός του υλικού αφετέρου της δειγματοληψίας δεδομένων με σκοπό την εκπαίδευση του μοντέλου μηχανικής μάθησης, το οποίο θα αναλυθεί στο επόμενο κεφάλαιο. Περαιτέρω, έλαβε μέρος εκτενής εξέταση των αναλογικών φίλτρων του κυκλώματος και πως αυτά βοηθούν στην ορθή καταγραφή ηλεκτρομυογραφικών σημάτων. Πέρα από τα φίλτρα, διερευνήθηκαν και τα στάδια ενίσχυσης του σήματος. Τέλος, παρατέθηκε μία περιγραφή του μικροελεγκτή ESP32, καθώς και το πως αξιοποιήθηκε στο τελικό σύστημα.

Κεφάλαιο 4ο: Μοντέλο Μηχανικής Μάθησης και Επεξεργασία Δεδομένων

4.1 Εισαγωγή

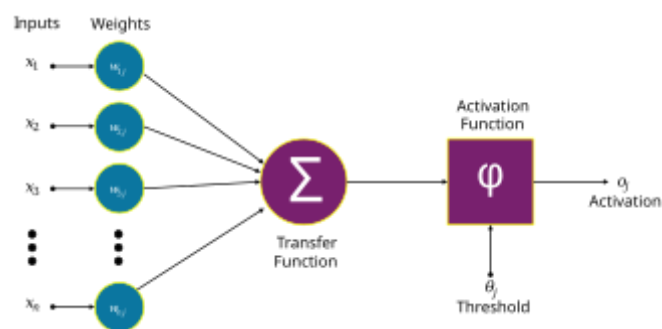
Παρότι στο προηγούμενο κεφάλαιο διερευνήθηκε εκτενώς όχι μόνο η αρχιτεκτονική του υλικού του συστήματος αλλά και του λογισμικού, στην παρούσα ενότητα θα γίνει ανάλυση αφενός του τρόπου δειγματοληψίας και επεξεργασίας αυτών των δεδομένων αφετέρου του μοντέλου μηχανικής μάθησης που υλοποιήθηκε. Αρχικά, θα γίνει μια εισαγωγή στη μηχανική μάθηση παραθέτοντας μία σύντομη ιστορική αναδρομή και τη βασική θεωρία του πεδίου. Εν συνεχεία, θα παρατεθεί ο τρόπος συλλογής των δεδομένων για την εκμάθηση του μοντέλου και προσδιοριστεί η συσχέτιση των χαρακτηριστικών. Τέλος θα αναλυθούν οι αλγόριθμοι που αξιοποιήθηκαν και θα γίνει η συνολική αξιολόγηση του μοντέλου.

4.2 Εισαγωγή στη Μηχανική Μάθηση

Σήμερα, οι εφαρμογές της μηχανικής μάθησης έχουν ενσωματωθεί στην καθημερινότητα ενός εκθετικά αυξανόμενου μέρους του παγκόσμιου πληθυσμού. Ωστόσο, πριν από την ευρεία διάδοση και τον εκδημοκρατισμό αυτής της τεχνολογίας, αρκετοί επιστήμονες και ερευνητικά κέντρα οικοδόμησαν σταδιακά το θεωρητικό και πρακτικό υπόβαθρο του πεδίου.

4.2.1 Ιστορική Αναδρομή

Το 1943 οι Γουόρεν ΜακΚάλοκ (Warren McCulloch) και Γουόλτερ Πιτς (Walter Pitts) συνέλαβαν το βασικό μαθηματικό μοντέλο των τεχνητών νευρωνικών δικτύων (artificial neural networks), εμπνευσμένοι από τη δομή των βιολογικών νευρωνικών δικτύων. Πιο συγκεκριμένα, μοντελοποίησαν μαθηματικά τη λειτουργία ενός βιολογικού νευρώνα, με τις εισόδους του να είναι ανάλογες με δυναμικά διέγερσης των μυών και την έξοδό του να προσομοιάζει αυτή του δυναμικού δράσης. Ακόμα, αποδίδεται ένας συντελεστής βάρους σε κάθε είσοδο, ο οποίος αντικατοπτρίζει τη διεγερτική ή ανασταλτική επίδραση του νευρώνα [34], [35].



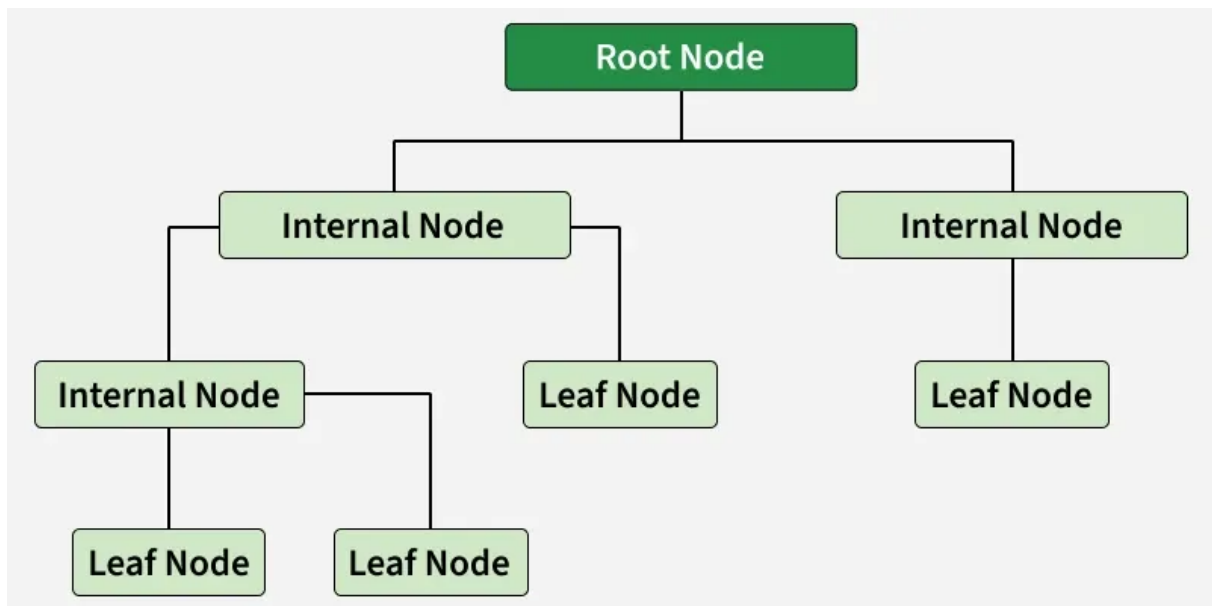
Σχήμα 4.1 Γραφική αναπαράσταση ενός τεχνητού νευρώνα [34]

Λίγα χρόνια αργότερα, το 1958, ο Αμερικανός ψυχολόγος Φρανκ Ρόζενμπλατ (Frank Rosenblatt) μετασημάτισε ριζικά το πεδίο μέσω της δημιουργίας του μοντέλου Perceptron. Γρήγορα η εφευρέσή του έγινε το πρότυπο των μοντέρνων τεχνητών νευρωνικών δικτύων, καθώς ο τρόπος με τον οποίο εκπαιδευόταν ήταν παρόμοιος με εκείνο που βρίσκεται στα ζώα και στους ανθρώπους βάσει της επιστήμης της ψυχολογίας. Πιο συγκεκριμένα, το μοντέλο Perceptron είναι μία εξελιγμένη μορφή του μοντέλου των ΜακΚάλοκ και Πιτς με ειδοποιό διαφορά τη μεταβλητότητα των συντελεστών των βαρών, θέτοντάς το πρώτο πραγματικό μοντέλο μηχανικής μάθησης [36].

Εν συνεχεία, στις αρχές του 1960 διεξάχθηκε εκτενής έρευνα πάνω στη σχεδίαση και την δοκιμή μοντέλων εκπαίδευσης. Ένα αξιοσημείωτο εύρημα στο πεδίο, δημοσιεύθηκε από τον πρωτοπόρο ερευνητή Χένρι Κέλεϊ, το οποίο περιέγραφε την οπισθοδιάδοση σφαλμάτων (Backpropagation) στα υπάρχον μοντέλα. Πιο συγκεκριμένα, ανέλυσε το πως σφάλματα μπορούν να διαδοθούν προς τα πίσω στο σύστημα επηρεάζοντας έτσι την απόδοσή του [35].

Ακόμα ο αλγόριθμος του κοντινότερου γείτονα (nearest neighbor algorithm), ο οποίος πρωτοπαρουσιάστηκε στο επιστημονικό κοινό το 1951 από την Έβελυν Φιξ (Evelyn Fix) και το Τζόσεφ Χόντιζ (Joseph Hodges), βρέθηκε ξανά στο παρασκήνιο με τους Τόμας Κόβερ (Thomas Cover) και Πίτερ Χαρτ (Peter E. Hart) να δημοσιεύουν την έρευνά τους, η οποία βασιζόταν σε αυτόν. Ο αλγόριθμός αυτός έχει εφαρμογές τόσο στη κατηγοριοποίηση (classification) όσο και στη παλινδρόμηση (regression). Πιο συγκεκριμένα, λειτουργεί μέσω της ανάκτησης των k πλησιέστερων δειγμάτων μεταξύ καινούριων δεδομένων και όλων των υπάρχον δειγμάτων εντός του συνόλου δεδομένων πάνω στα οποία εκπαιδεύτηκε [35].

Το 1984 δημοσιεύθηκε ο αλγόριθμος CART (Classification And Regression Tree) από τον Λίο Μπράινμαν (Leo Breiman). Πιο συγκεκριμένα, ο αλγόριθμος αυτός διασπάει το σύνολο των δεδομένων σε μικρότερες ομάδες, οι οποίες με τη σειρά τους αξιοποιούνται για τη δημιουργία δέντρων αποφάσεων (decision trees), τα οποία βοηθούν στη πρόβλεψη τόσο προβλημάτων κατηγοριοποίησης όσο και παλινδρόμησης [35], [37].



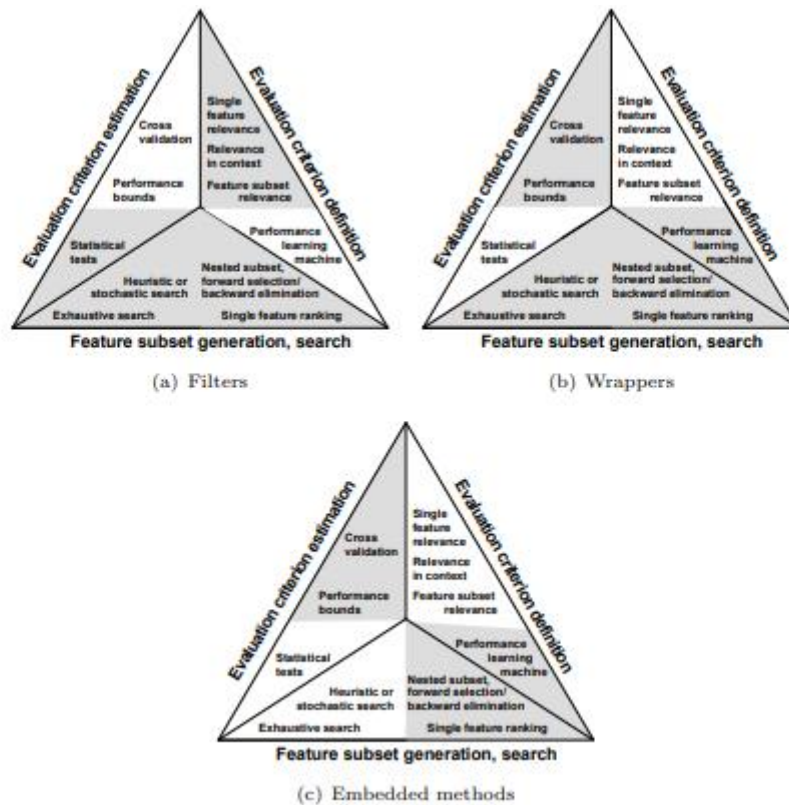
Σχήμα 4.2 Γραφική αναπαράσταση του CART [37]

Άλλοι αξιοσημείωτοι αλγόριθμοι και αρχιτεκτονικές που αναπτύχθηκαν προς τα τέλη του 20^{ου} αιώνα είναι ο C4.5 από τον Ρος Κουίνλαν (Ross Quinlan), η τεχνική ενίσχυσης (boosting) της συνδυαστικής μάθησης (ensemble learning) από τον Ρόμπερτ Σχαπίρε (Robert Schapire), ο αλγόριθμος μακροπρόθεσμης – βραχυπρόθεσμης μνήμης (long short-term memory - LSTM) και αυτός των μηχανών διανυσμάτων υποστήριξης (Support Vector Machines - SVMs) στα εργαστήρια της Bell από τον Βλάντιμιρ Βάπνικ (Vladimir Vapnik) και τους συναδέλφους του [35].

4.2.2 Εξαγωγή Χαρακτηριστικών και Ανάλυση Συσχέτισης

Προτού ξεκινήσει η ανάπτυξη ενός μοντέλου μηχανικής μάθησης, πρέπει να οριστεί ο στόχος για τον οποίο υλοποιείται. Πιο συγκεκριμένα, τα περισσότερα προβλήματα ενός μοντέλου εμφανίζονται όταν ο στόχος καθορίζεται από μία σειρά τυχαίων περιπτώσεων παρά από προκαθορισμένους κανόνες. Για την αποτελεσματική εκπαίδευση ενός μοντέλου μηχανικής μάθησης, πρέπει να οριστεί ένας αριθμός παραδειγμάτων που σχετίζεται με την επιθυμητή έξοδο. Για τον λόγο αυτό, η εύρεση δεδομένων που συνδέουν λογικά τα μοτίβα των παραδειγμάτων με την επιθυμητή έξοδο, είναι μία διαδικασία ζωτικής σημασίας για την εκπαίδευση τέτοιων μοντέλων [38].

Η εξαγωγή αυτών των μοτίβων, ή αλλιώς χαρακτηριστικών (Feature Extraction), διασπάται σε δύο βήματα, αυτό της διαμόρφωσης και αυτό της επιλογής. Αρχικά, κατά τη δημιουργία των χαρακτηριστικών πρέπει να βρεθεί η δομή από τα ακατέργαστα δεδομένα, η οποία περιγράφει καλύτερα τη διαδικασία που ακολουθείται για την επιθυμητή έξοδο. Έπειτα, κατά την επιλογή των χαρακτηριστικών τα δεδομένα αυτά φιλτράρονται, με σκοπό τη βελτιστοποίηση της αποδοτικότητας του μοντέλου και γενικότερα την αύξηση της συσχέτισής τους με το επιθυμητό αποτέλεσμα [38].



Σχήμα 4.3 Μέθοδοι εξαγωγής χαρακτηριστικών [38]

Τα χαρακτηριστικά αυτά μπορούν να διατυπωθούν μαθηματικά μέσω ενός διάνυσματος διάστασης n :

$$x = [x_1, x_2, \dots, x_n] \quad (4.1)$$

Ένας από τους τρόπους για το αποτελεσματικό φιλτράρισμα του διανύσματος που περιγράφουν τα διαμορφωμένα χαρακτηριστικά είναι η κατάταξή τους βάσει της μεμονωμένης συσχέτισής τους. Ειδικότερα, με αυτή τη μέθοδο γίνεται αντιληπτό ποια χαρακτηριστικά δεν προσφέρουν αρκετά στην εκπαίδευση του μοντέλου λόγω πλεονασμού ή ενώ κάποιο χαρακτηριστικό μπορεί να μην σχετίζεται

μεμονωμένα με το επιθυμητό αποτέλεσμα, να μπορεί να προσφέρει χρήσιμη πληροφορία σε συνοχή με άλλα χαρακτηριστικά [38].

Ένα από τα πλέον διαδεδομένα μαθηματικά μοντέλα για την εύρεση της συσχέτισης χαρακτηριστικών είναι ο συντελεστής συσχέτισης Pearson (Pearson Correlation Coefficient). Αναλυτικότερα, το συγκεκριμένο μοντέλο χρησιμοποιείται και στη παρούσα διπλωματική εργασία και υλοποιείται με τη χρήση δύο διανυσμάτων m διαστάσεων x και y , τα οποία περιγράφουν το σύνολο των τιμών των δεδομένων και των τιμών της επιθυμητής εξόδου. Η εξίσωση του μοντέλου είναι η εξής [38]:

$$C(j) = \frac{|\sum_{i=1}^m (x_{i,j} - \bar{x}_j)(y_i - \bar{y})|}{\sqrt{\sum_{i=1}^m (x_{i,j} - \bar{x}_j)^2 \sum_{i=1}^m (y_i - \bar{y})^2}} \quad (4.2)$$

Για την αξιοποίηση της συγκεκριμένης μαθηματικής μεθόδου στα πλαίσια της παρούσας διπλωματικής εργασίας, έγινε χρήση της βιβλιοθήκης Pandas, η οποία προσφέρει μία έτοιμη μέθοδο για τον υπολογισμό της συσχέτισης. Ο παρακάτω κώδικας αποτελεί την εφαρμογή της παραπάνω εξίσωσης:

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

df = pd.read_csv('final_dataframe_with_fft_second.csv')

features_df = df[['MAV', 'RMS', 'WL', 'ZC', 'SSC', 'Peak_FFT', 'Mean_FFT']]

corr_matrix = features_df.corr()

print(corr_matrix)

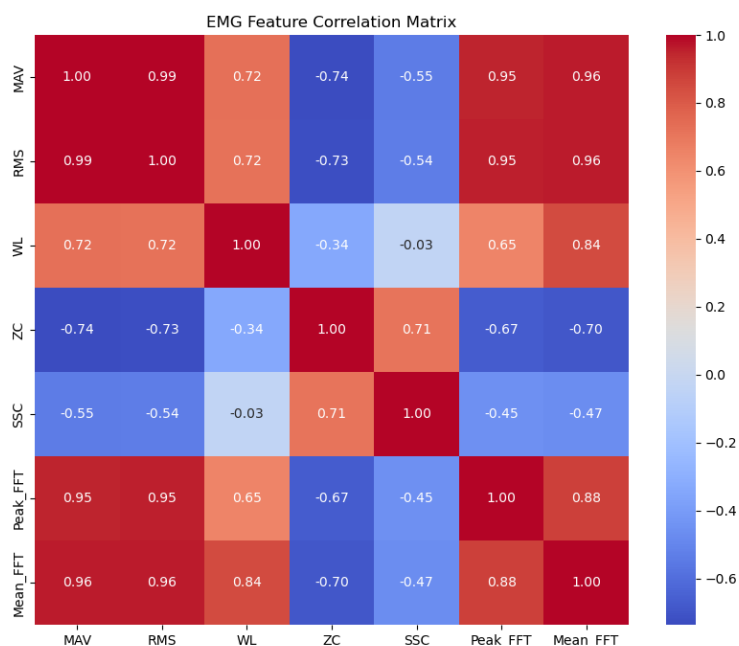
plt.figure(figsize=(10, 8))
sns.heatmap(corr_matrix, annot=True, cmap='coolwarm', fmt=".2f")
plt.title('EMG Feature Correlation Matrix')
plt.show()
```

Τα χαρακτηριστικά που φαίνονται στο απόσπασμα του κώδικα εξάχθηκαν βάσει της έρευνας των Cengiz Tere και Mehmet Can Demir. Πιο συγκεκριμένα, για την κατηγοριοποίηση των ηλεκτρομυογραφικών σημάτων χρησιμοποιούνται τα χαρακτηριστικά της τετραγωνικής μέσης τιμής (Root Mean Squared - RMS), της μέσης απόλυτης τιμής (Mean Absolute Value - MAV), της διέλευσης από το μηδέν (Zero Crossing - ZC), του μήκους κύματος (Waveform Length - WL), της αλλαγής προσήμου κλίσης (Slope Sign Change - SSC) και εφόσον οι καταγραφές που χρησιμοποιήθηκαν εμπεριείχαν και δεδομένα από το πεδίο της συχνότητας έγινε χρήση των κορυφών του μετασχηματισμού Fourier του σήματος (Peak_FFT) καθώς και της μέσης τιμής του [39].

Πίνακας 4.1 Μαθηματικές εξισώσεις των επιλεγμένων χαρακτηριστικών [39]

$WL = \sum_{n=1}^{N-1} x_{n-1} - x_n $	(4.3)
$MAV = \frac{1}{N} \sum_{n=1}^{N-1} x_n $	(4.4)
$ZC = \sum_{n=1}^{N-1} [sign(x_n * x_{n+1}) \cap (x_n - x_{n-1})],$ $sign(x) = \begin{cases} 1, & x \geq 0 \\ 0, & x < 0 \end{cases}$	(4.5)
$RMS = \sqrt{\frac{1}{N} \sum_{n=1}^N x_n^2}$	(4.6)
$SSC = \sum_{n=1}^{N-1} f[(x_n - x_{n-1}) * (x_n - x_{n+1})],$ $f(x) = \begin{cases} 1, & x \geq 0 \\ 0, & x < 0 \end{cases}$	(4.7)

Ακόμα η συσχέτιση αυτών των χαρακτηριστικών μπορεί να οπτικοποιηθεί με τη χρήση ενός θερμοκού χάρτη συσχέτισης. Πιο συγκεκριμένα, απεικονίζει τον συντελεστή συσχέτισης Pearson ο οποίος λαμβάνει τιμές από -1 έως +1, αντικατοπτρίζοντας τη σχέση μεταξύ των αυξομειώσεων των μεταβλητών. Όταν η τιμή είναι κοντά στο -1 τότε καθώς η μία αυξάνεται η άλλη μειώνεται και για τιμή κοντά στο +1 η αύξηση της μίας ακολουθεί και αύξηση της άλλης. Αν η τιμή είναι κοντά στο 0 τότε δεν υπάρχει κάποια γραμμική σχέση μεταξύ των μεταβλητών.

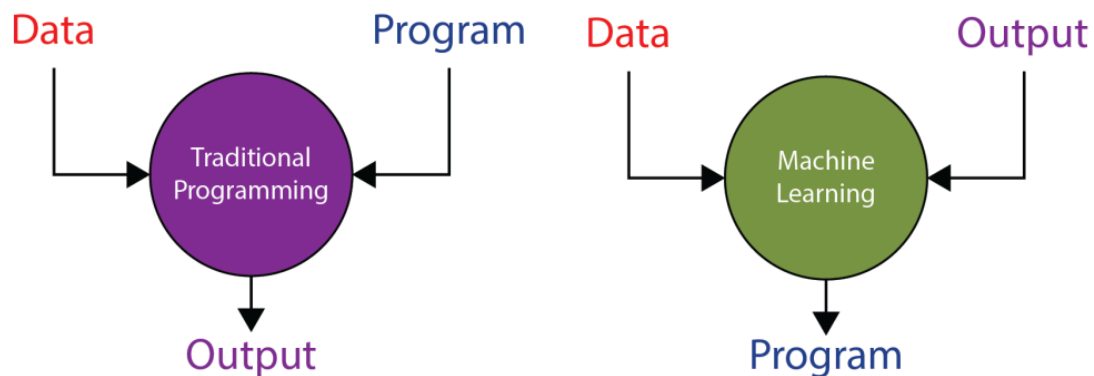


Σχήμα 4.4 Correlation heatmap των επιλεγμένων χαρακτηριστικών

Όπως μπορεί να παρατηρηθεί από το θερμικό χάρτη συσχέτισης υπάρχει πλεονασμός μεταξύ κάποιων χαρακτηριστικών. Πιο συγκεκριμένα, τα χαρακτηριστικά του πεδίου της συχνότητας παρουσιάζουν τιμή πολύ κοντά στο +1 τόσο αναμεταξύ τους όσο και τα χαρακτηριστικά MAV και RMS. Για τον λόγο αυτό μπορούν να αφαιρεθούν από τη διαδικασία της εκπαίδευσης, θέτοντας το τελικό μοντέλο πιο αποτελεσματικό. Ακόμα, θα μπορούσε να γίνει χρήση μόνο ενός χαρακτηριστικού από τα MAV και RMS, καθώς και αυτά παρουσιάζουν τιμή κοντά στο +1. Η συσχέτιση των υπόλοιπων χαρακτηριστικών είναι ικανοποιητική, καθώς φαίνεται ότι εξάγουν χρήσιμη πληροφορία από μόνα τους.

4.2.3 Βασική θεωρία Μηχανικής Μάθησης

Πριν γίνει περαιτέρω ανάλυση του συστήματος, είναι σημαντικό να παρατεθεί μια σύντομη εισαγωγή στην επιστήμη της Μηχανικής Μάθησης. Ο προγραμματισμός στην παραδοσιακή του μορφή αποσκοπεί στη καθοδήγηση των υπολογιστών για την εκτέλεση μιας ενέργειας, με τη συγγραφή κώδικα που περιγράφει αυστηρούς κανόνες σε γνωστά δεδομένα. Στη μηχανική μάθηση ωστόσο, η διαδικασία για να περιγράψει κάποιος σε έναν υπολογιστή την ενέργεια που θέλει να κάνει είναι διαφορετική. Πιο συγκεκριμένα, σε αυτή τη περίπτωση γίνεται χρήση των αρχικών δεδομένων και του επιθυμητού αποτελέσματος σαν είσοδο στο πρόγραμμα, με σκοπό ο υπολογιστής να θέσει τους κανόνες με τους οποίους θα διεκπεραιώσει την ενέργεια αυτή [40].



Σχήμα 4.5 Διαδικασία ανάπτυξης εφαρμογών στην Μηχανική Μάθηση [40]

Για την εκπαίδευση μοντέλων μηχανικής μάθησης υπάρχουν αρκετές προσεγγίσεις, εκ των οποίων μία είναι και η τεχνική της επιβλεπόμενης μάθησης (supervised learning), η οποία και χρησιμοποιήθηκε για την υλοποίηση της εργασίας. Πιο συγκεκριμένα, όπως αναλύθηκε και στη προηγούμενη παράγραφο, ο στόχος της μηχανικής μάθησης είναι η εξαγωγή κανόνων από τον υπολογιστή βάσει των δεδομένων εισόδου. Η συγκεκριμένη διαδικασία μπορεί να διατυπωθεί μαθηματικά ως εξής [40]:

$$y = g(X), \quad (4.8)$$

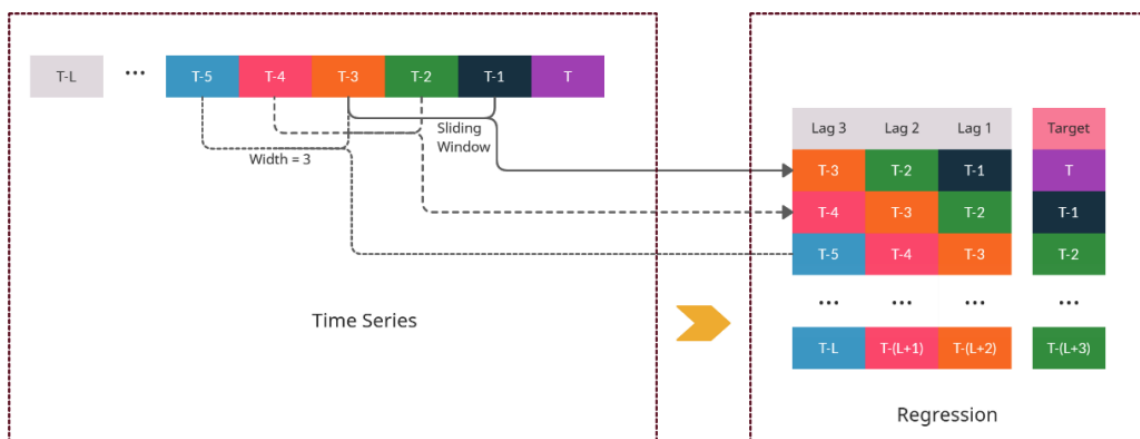
όπου το X αναπαριστά το σύνολο των δεδομένων των χαρακτηριστικών του μοντέλου και το g είναι η ιδανική συνάρτηση-στόχος (ideal target function), η οποία αντιστοιχίζει τα δεδομένα εισόδου X στην επιθυμητή έξοδο y [40].

Τα μοντέλα μηχανικής μάθησης μπορούν να χρησιμοποιηθούν για τη λύση προβλημάτων που σχετίζονται με τη παλινδρόμηση (regression), την κατηγοριοποίηση (classification) και τη σύσταση (recommendation) με τις πιο διαδεδομένες εφαρμογές να είναι οι πρώτες δύο. Στη παρούσα διπλωματική εργασία αποτελεί αντικείμενο εξέτασης η λύση ενός προβλήματος κατηγοριοποίησης ηλεκτρομυγραφικών σημάτων που καταγράφονται σε πραγματικό χρόνο. Πιο συγκεκριμένα, το EMG σήμα πρέπει να αντιστοιχηθεί στις κλάσεις FIST, INDEX, LITTLE, MIDDLE, RING, NOISE, που περιγράφουν την κίνηση του σχηματισμού γροθιάς, της κάμψης των δαχτύλων εκτός του αντίχειρα και της κατάστασης ηρεμίας του μυός αντίστοιχα [40].

Μία από τις κυριότερες προκλήσεις που αντιμετωπίζουν τα μοντέλα μηχανικής μάθησης είναι ότι αφού εκπαιδευτούν πρέπει να έχουν ικανοποιητικές επιδόσεις με είσοδο καινούρια σύνολα δεδομένων. Ανάλογα με την ικανότητα γενίκευσής τους (generalization capability) τα μοντέλα μπορούν να χωριστούν σε τρεις κατηγορίες. Αρχικά, άμα το σύνολο δεδομένων δεν ήταν αρκετό για να εκπαιδεύσει αποτελεσματικά το μοντέλο, τότε είναι προφανές πως αυτό δεν θα επιφέρει τα καλύτερα δυνατά αποτελέσματα. Σε αυτή την περίπτωση λέγεται πως έχει γίνει υποπροσαρμογή (underfitting). Στη δεύτερη κατηγορία, αντιθέτως με την πρώτη, αν τα δεδομένα που χρησιμοποιήθηκαν κατά την εκπαίδευση ήταν αρκετά συγκεκριμένα και μεγάλα σε όγκο τότε το μοντέλο δεν θα μπορεί εύκολα να κατηγοριοποιήσει τα καινούρια σύνολα δεδομένων που θα δεχτεί ως είσοδο. Σε αυτή τη περίπτωση λέγεται πως έχει γίνει υπερπροσαρμογή (overfitting). Η τελευταία κατηγορία συνιστά τον επιθυμητό στόχο, ο οποίος είναι η χρυσή τομή μεταξύ των προηγούμενων. Πιο συγκεκριμένα, κατά τη διαδικασία της εκπαίδευσης πρέπει το σύνολο των δεδομένων να είναι ικανοποιητικό σε όγκο και τα περιεχόμενά του να έχουν μικρές αποκλίσεις το ένα από το άλλο για να μπορέσει μετέπειτα το μοντέλο να καταλάβει τη μεταβλητότητα μίας εισόδου [40].

Για την επαλήθευση της ορθής λειτουργίας του μοντέλου, ένα μέρος του συνόλου δεδομένων δεν χρησιμοποιείται για την εκπαίδευσή του, το οποίο αποκαλείται σύνολο επαλήθευσης (validation set), αλλά δεσμεύεται για τον μετέπειτα έλεγχο που θα λάβει χώρα. Πιο συγκεκριμένα, αυτή η μέθοδος αξιοποιείται από διάφορους αλγορίθμους και αποτελεί τον τελευταίο έλεγχο που αποφασίζει αν η φάση της εκπαίδευσης θα τερματιστεί ή θα ξεκινήσει πάλι από την αρχή [40].

Η πρόβλεψη ή η κατηγοριοποίηση πάνω στα δεδομένα εισόδου, τα οποία είναι συνεχούς χρόνου, θα έπρεπε σε ένα ιδανικό σενάριο να γίνεται καθ' όλο το μήκος τους. Αυτή η τεχνική ωστόσο δεν είναι αρκετά πρακτική, καθώς τα δεδομένα μπορεί να έχουν ένα αυθαίρετο μέγεθος L . Για τον λόγο αυτό η πρόβλεψη και η κατηγοριοποίηση που έχει ως έξοδο ένα μοντέλο, περιορίζεται στα M πιο πρόσφατα δεδομένα που εισήλθαν στο σύστημα [40].



Σχήμα 4.6 Γραφική αναπαράσταση του sliding window [40]

Με σκοπό την εφαρμογή αυτής της τεχνικής, εφαρμόζεται ένα κινητό παράθυρο χρόνου (sliding window) για την εξαγωγή δεδομένων σταθερού μήκους. Πιο συγκεκριμένα, το μήκος των δεδομένων πάνω στα οποία γίνεται η πρόβλεψη είναι $M < L$. Η συγκεκριμένη τεχνική χρησιμοποιείται επίσης και για την υλοποίηση της παρούσας εργασίας ως κομμάτι του λογισμικού που μεταφορτώθηκε στον μικροελεγκτή ESP32:

```

...
// Interrupt & Buffer Variables
volatile double emgBuffer[WINDOW_SIZE];
volatile int bufferIndex = 0;
volatile bool windowReady = false;
hw_timer_t * timer = NULL;

double localWindow[WINDOW_SIZE];
...
// Interrupt Service Routine (Runs exactly at 1000Hz)
void IRAM_ATTR onTimer() {
    if (!windowReady) {

```

```

int rawValue = analogRead(EMG_PIN);

// Convert 12-bit ADC directly to Voltage
double voltage = (rawValue / 4095.0) * 3.3;

emgBuffer[bufferIndex] = voltage;
bufferIndex++;

if (bufferIndex >= WINDOW_SIZE) {
    windowReady = true; // Tell the main loop to process the data
}
}
}
...
void loop() {
    if (windowReady) {
...

```

4.2.4 Εφαρμογές Μηχανικής Μάθησης σε Βιολογικά Δεδομένα

Η εφαρμογή μοντέλων μηχανικής μάθησης με τη χρήση βιολογικών δεδομένων γίνεται ολοένα και πιο διαδεδομένη. Πιο συγκεκριμένα, η ανάλυση σημάτων όπως αυτά που εξάγονται μέσω της ηλεκτομυογραφίας, της ηλεκτροκαρδιογραφίας και της ηλεκτροεγκεφαλογραφίας, παρουσιάζει ιδιαίτερες προκλήσεις λόγω της μη γραμμικής, μη στάσιμης και θορυβώδους φύσης τους. Η μηχανική μάθηση βοηθάει ώστε η επεξεργασία αυτών των σημάτων να ξεφύγει από τις παραδοσιακές μεθόδους με σκοπό την ερμηνεία αυτών των σύνθετων δεδομένων, καθιστώντας τη ένα εργαλείο ζωτικής σημασίας για την εξαγωγή ουσιαστικής πληροφορίας.

Κάποιες από της κυριότερες εφαρμογές της μηχανικής μάθησης σε βιολογικά δεδομένα εστιάζουν στους τομείς του μυοηλεκτρικού ελέγχου προσθετικών μελών, την κλινική διάγνωση και παρακολούθηση ασθενών καθώς και τις διεπαφές ανθρώπου-υπολογιστή. Ειδικότερα, λόγω των εξελίξεων στο πεδίο της μηχανικής μάθησης, αρχικά ο έλεγχος προσθετικών μελών γίνεται όλο και πιο εξατομικευμένος στις ανάγκες του χρήστη. Ακόμα, μοντέλα μηχανικής μάθησης εκπαιδευμένα πάνω σε δεδομένα ηλεκτροκαρδιογραφημάτων μπορούν δίνουν την ικανότητα σε τμήματα επειγόντων περιστατικών να λάβουν πιο γρήγορες και ακριβής αποφάσεις για την αντιμετώπιση των προβλημάτων που φέρουν οι ασθενείς. Περαιτέρω, η χρήση τέτοιων συστημάτων έχουν μια πληθώρα εφαρμογών στη σχεδίαση και υλοποίηση διεπαφών ανθρώπου-υπολογιστή. [41], [42].

Μία ακόμα αξιοσημείωτη εφαρμογή της μηχανικής μάθησης που είναι εκπαιδευμένη πάνω σε βιολογικά δεδομένα, είναι το καινούριο μοντέλο της εταιρείας Meta που ονομάζεται Tribe V2. Πιο

συγκεκριμένα, το μοντέλο αυτό εκπαιδεύτηκε σε χιλιάδες ώρες εγκεφαλικής δραστηριότητας με τη χρήση απεικόνισης μαγνητικού συντονισμού (Functional Magnetic Resonance Imaging - fMRI), και προσφέρει προβλέψεις στο πως το ανθρώπινο μυαλό επεξεργάζεται διάφορα ερεθίσματα [43].

4.3 Συλλογή και Προεπεξεργασία Δεδομένων (Data Transformation)

Για τη συλλογή δεδομένων για τις επιθυμητές κινήσεις, όπως προαναφέρθηκε, έγινε χρήση του παλμογράφου Picoscope 2204A και του αντίστοιχου λογισμικού, το οποίο επιτρέπει την αποθήκευση καταγραφών σε μορφή csv. Πιο συγκεκριμένα, μέσω του λογισμικού Picoscope 7 υπάρχει η δυνατότητα αποθήκευσης των δεδομένων τόσο του πεδίου του χρόνου όσο και του πεδίου της συχνότητας που καταγράφει ο παλμογράφος. Για το πεδίο της συχνότητας έχει ως έξοδο δύο στήλες αυτή των kHz και αυτή των dBu, ενώ για το πεδίο του χρόνου έχει τις στήλες των δευτερολέπτων και της τάσης.

Πίνακας 4.2 Μορφή δεδομένων του πεδίου του χρόνου

	(s)	(V)
1	0	1.65685
2	0.0001	1.65685
3	0.0002	1.65685
4	0.0003	1.61214
5	0.0004	1.61214

Πίνακας 4.3 Μορφή δεδομένων του πεδίου της συχνότητας

	(kHz)	(dBu)
1	0	12.06475
2	0.000305	4.547506
3	0.00061	-11.3724
4	0.000916	-77.9619
5	0.001221	-67.9475

Έγιναν συνολικά 20 καταγραφές για κάθε κίνηση, όπου η κάθε μία περιέχει δεδομένα για 10 δευτερόλεπτα. Η συχνότητα δειγματοληψίας των καταγραφών είναι στα 10kHz, και επέφερε σε κάθε αρχείο csv περίπου 100 χιλιάδες γραμμές δεδομένων.

Είναι προφανές πως τα αποθηκευμένα δεδομένα δεν έχουν την επιθυμητή μορφή για την εκπαίδευση του μοντέλου. Ειδικότερα, όπως αναλύθηκαν στο κεφάλαιο 4.2.2 πρέπει αυτά τα δεδομένα να επεξεργαστούν για να αντικατοπτρίζουν τις τιμές των επιθυμητών χαρακτηριστικών. Για την μετατροπή αυτών των δεδομένων στη σωστή μορφή συγγράφηκε κώδικας, με τη χρήση της γλώσσας Python.

Αρχικά, το πρόγραμμα Python, θέτει ένα χρονικό παράθυρο των 100ms για την επεξεργασία των χιλιάδων γραμμών δεδομένων με βήμα 20ms. Πιο συγκεκριμένα, σε εφαρμογές διεπαφών ανθρώπου-υπολογιστή το αποδεκτό όριο καθυστέρησης είναι τα 100ms-200ms, καθώς ο άνθρωπος μπορεί να αντιληφθεί μια καθυστέρηση μεγαλύτερη από αυτή. Ακόμα με τη χρήση των τιμών αυτού του χρονικού παραθύρου και του βήματος, υπάρχει υψηλή ανάλυση δεδομένων καθώς λαμβάνει χώρα μεγάλη επικάλυψη στα τελικά δεδομένα, με αποτέλεσμα το μοντέλο μηχανικής μάθησης να μπορεί να ανιχνεύσει καλύτερα τις μεταβάσεις μεταξύ των χειρονομιών. Περαιτέρω, οι τιμές αυτές επιλέχθηκαν με σκοπό την βελτιστοποίηση της χρήσης των υπολογιστικών πόρων του ESP32, καθώς είναι πιο αποτελεσματική η επεξεργασία 1000 δειγμάτων ανά πρόβλεψη παρά 10 χιλιάδων. [44].

Εν συνεχεία, ο κώδικας φορτώνει τα αρχεία csv των καταγραφών με σκοπό την γρηγορότερη πρόσβασή του σε αυτά. Αξιοποιώντας τη μέθοδο `read_csv` που προσφέρει η βιβλιοθήκη Pandas δίνεται η δυνατότητα προσπέλασης των δεδομένων εντός αρχείων τέτοιου τύπου. Κατά την φόρτωσή τους, αρχειοθετούνται στο αντικείμενο `voltage_files` ανάλογα με την χειρονομία στην οποία αντιστοιχούν.

```
...
voltage_files = {'noise': [], 'fist': [], 'index': [], 'middle': [], 'ring':
[], 'little': []}
...
def getFiles(self):
    for recording in self.recordings:
        print(f'Loading {recording} data...')
        for i in range(1,21):
            voltage_file_path = f"..../TRAINING
DATA/VOLTAGE_SECOND/{recording}/{i}.csv"
            self.voltage_files[recording].append(pd.read_csv(voltage_file_pa
th,header=1))
...
```

Για τον σωστό υπολογισμό των χαρακτηριστικών, γίνεται αφαίρεση της υπερύψωσης του σήματος, ώστε το κέντρο του να βρίσκεται στα 0V. Αναλυτικότερα, γίνεται φόρτωση ενός αρχείου που περιγράφει τη μυϊκή αδράνεια και υπολογίζεται ο μέσος όρος όλων των γραμμών του, με σκοπό την υπολογιστική εύρεση της τιμής κατά την οποία το σήμα υψώθηκε στη τελική βαθμίδα του κυκλώματος.

```
...
def getCenterOfSignal(self):
    baseline_file = pd.read_csv('../TRAINING
DATA/VOLTAGE_SECOND/noise/1.csv', header=1)
    offset = baseline_file['(V)'].mean()
    return offset
...
```

Έπειτα από τα δύο αυτά βήματα γίνεται ο υπολογισμός όλων των χαρακτηριστικών και η δημιουργία ενός καινούριου csv αρχείου που έχει την επιθυμητή μορφή δεδομένων. Αναλυτικότερα, όπως προαναφέρθηκε γίνεται ο υπολογισμός των MAV, RMS, WL, ZC, SSC και των χαρακτηριστικών του πεδίου της συχνότητας και τοποθετούνται σε ένα πλαίσιο δεδομένων με την πρόσθεση των πεδίων Label και Time_Range.

```

...
def __init__(self):
    self.offset = self.getCenterOfSignal()
    self.GetFiles()
    print('Caclulating MAV...')
    self.getMeanAbsoluteValue()
    print('Caclulating RMS...')
    self.getRootMeanSquared()
    print('Caclulating WL...')
    self.getWaveformLength()
    print('Caclulating ZC...')
    self.getZeroCrossings()
    print('Caclulating SSC...')
    self.getSlopeSignChange()
    print('Caclulating FFT features...')
    self.getDynamicFFTFeatures()

    self.master_df = pd.concat([
        self.mav_df,
        self.rms_df['RMS'],
        self.wl_df['WL'],
        self.zc_df['ZC'],
        self.ssc_df['SSC'],
        self.dynamic_fft_df[['Peak_FFT', 'Mean_FFT']]
    ], axis=1)
    self.master_df.to_csv('final_dataframe_with_fft_second.csv')
...

```

Πίνακας 4.4 Μορφή δεδομένων μετά τον μετασχηματισμό

	Label	Time_Range	MAV	RMS	WL	ZC	SSC	Peak_FFT	Mean_FFT
--	-------	------------	-----	-----	----	----	-----	----------	----------

0	noise	0.0ms-100.0ms	0.032	0.04	18.206	186	144	16.183	0.981
1	noise	20.0ms-120.0ms	0.031	0.041	16.595	174	139	15.555	0.874
2	noise	40.0ms-140.0ms	0.032	0.044	17.312	161	135	16.498	0.991
3	noise	60.0ms-160.0ms	0.032	0.046	18.162	164	143	16.721	1.051
4	noise	80.0ms-180.0ms	0.032	0.046	17.535	156	131	16.318	1.072
5	noise	100.0ms-200.0ms	0.032	0.045	17.446	166	137	15.871	1.053

4.4 Αλγόριθμος Τυχαίων Δασών (Random Forests) και XGBoost

Ο αλγόριθμος των Τυχαίων Δασών (Random Forests), μετασχηματίζει την τεχνική της συναθροιστικής επαναδειγματοληψίας (Bootstrap Aggregating - Bagging) δημιουργώντας ένα μεγάλο σύνολο δέντρων που δεν συσχετίζονται μεταξύ τους. Πιο συγκεκριμένα, η πρακτική Bagging είναι μία μορφή της συλλογικής μάθησης (ensemble learning) κατά την οποία τα δεδομένα διασπώνται σε υποσύνολα για την εκπαίδευση ασθενών μοντέλων (weak learners), και τα συνδυάζει είτε με βάση του μέσου όρου τους είτε με μια διαδικασία «ψηφοφορίας», ανάλογα με το αν η φύση του προβλήματος είναι παλινδρόμησης ή κατηγοριοποίησης αντίστοιχα [40].

Στον συγκεκριμένο αλγόριθμο μπορούμε να αποφασίσουμε τον τελικό αριθμό των δέντρων που θέλουμε να υλοποιηθούν. Πιο συγκεκριμένα για την κατασκευή M δέντρων πρέπει αρχικά να γίνει επιλογή ενός υποσυνόλου από τα δεδομένα εισόδου και σε συνέχεια να επιλεγούν τυχαία κάποια από τα χαρακτηριστικά που έχουμε ορίσει. Εν συνεχεία, λαμβάνει χώρα γίνεται διαχωρισμός με βάση τα χαρακτηριστικά που επιλέχθηκαν και ο κόμβος διασπάται σε δύο κόμβους παιδιά. Τέλος, επαναλαμβάνονται τα δύο ενδιάμεσα βήματα μέχρι να ικανοποιούνται τα κριτήρια τερματισμού του αλγορίθμου [40].

Η ειδοποιός διαφορά του συνόλου των M δέντρων που κατασκευάζονται από τον αλγόριθμο random forests σε σχέση με τα παραδοσιακά δέντρα, είναι η τυχαία δειγματοληψία των χαρακτηριστικών σε κάθε διάσπαση του εκάστοτε κόμβου. Σε προβλήματα κατηγοριοποίησης, λαμβάνεται υπόψη η πλειοψηφία των τελικών αποτελεσμάτων όλων των δέντρων, ενώ σε προβλήματα παλινδρόμησης υπολογίζεται ο μέσος όρος τους [40]:

$$\hat{y} = \frac{1}{M} \sum_{t=1}^M \tau_t(x) \quad (4.9)$$

Συνήθως ο συγκεκριμένος αλγόριθμος επιδεικνύει υψηλή απόδοση σε αρκετά σύνολα δεδομένων με λίγη παραμετροποίηση. Για τον λόγο αυτό, επιλέγεται αρκετά συχνά για την εκπαίδευση μοντέλων μηχανικής μάθησης, καθώς είναι δύσκολο να γίνει υπερπροσαρμογή του, ωστόσο εμφανίζει μία αδυναμία προεκβολής λόγω της χρήσης δέντρων επιλογής (decision trees) [40].

Ένας ακόμα τρόπος για την δημιουργία μοντέλων μηχανικής μάθησης είναι η ενίσχυση (Boosting). Η διαφορά του με την τεχνική bagging είναι ο τρόπος με τον οποίο συνδυάζονται οι weak learners. Πιο συγκεκριμένα, κατά την ενίσχυση οι τελευταίοι χρησιμοποιούνται με σειριακό τρόπο, εφαρμόζοντας κάθε φορά διάφορες παραλλαγές των δεδομένων. Η συγκεκριμένη τεχνική μπορεί να αποτυπωθεί μαθηματικά ως εξής:

$$F(x) = f_1(x) + f_2(x) + \dots + f_n(x), \quad (4.10)$$

όπου f είναι αντικατοπτρίζουν τους weak learners.

Ένας αλγόριθμος που εφαρμόζει την τεχνική της ενίσχυσης για την επίλυση προβλημάτων κατηγοριοποίησης είναι ο XGBoost (eXTreme Gradient Boosting). Δημοσιεύτηκε το 2016 από τους Γιάνκι Τσεν (Tianqi Chen) και Κάρλος Γκουέστριν (Carlos Guestrin) και διεύρυνε τα όρια της απόδοσης των μοντέλων που επικρατούσαν εκείνη την περίοδο. Ο συγκεκριμένος αλγόριθμος επιλέχθηκε επίσης και για την υλοποίηση του μοντέλου μηχανικής μάθησης που χρησιμοποιήθηκε στην παρούσα διπλωματική εργασία [40].

```
...
# 1. Load Data
data = pd.read_csv('final_dataframe_with_fft_second.csv')
data = data.dropna(subset=['Label'])

# 2. Define Features
X = data[['MAV', 'RMS', 'WL', 'ZC', 'SSC', 'Peak_FFT', 'Mean_FFT']]
y = data['Label']

# 3. Encode Labels
label_encoder = LabelEncoder()
y_encoded = label_encoder.fit_transform(y)

# 4. Train/Test Split
X_train, X_test, y_train, y_test = train_test_split(
    X, y_encoded, test_size=0.2, random_state=42, stratify=y_encoded
)

# 5. Define the XGBoost Model
best_model = xgb.XGBClassifier(
    max_depth=4,
    n_estimators=50,
    learning_rate=0.1,
)

print("Training the tiny model...")
```

```
best_model.fit(X_train, y_train)
```

```
...
```

Το παραπάνω απόσπασμα του κώδικα ακολουθεί μία λογική ροή για την εκπαίδευση του μοντέλου μηχανικής μάθησης. Αρχικά γίνεται εισαγωγή των επεξεργασμένων πλέον δεδομένων και φιλτράρονται για μία τελευταία φορά, σε περίπτωση που κάποια εγγραφή έχει τιμή NA στη στήλη των Labels. Στη συνέχεια κωδικοποιούνται οι ετικέτες με τη χρήση της κλάσης LabelEncoder() που προσφέρει η βιβλιοθήκη sklearn. Η συγκεκριμένη διαδικασία είναι απαραίτητη, καθώς οι αλγόριθμοι μηχανικής μάθησης εκτελούν μαθηματικές πράξεις και απαιτούν την έξοδό τους να είναι σε αριθμητική μορφή αντί για αλφαριθμητική. Έπειτα γίνεται ο διαχωρισμός των δεδομένων σε σύνολο εκπαίδευσης και σύνολο δοκιμής. Πιο συγκεκριμένα, το 20% των δεδομένων δεν χρησιμοποιείται στην εκπαίδευση του μοντέλου, αλλά για τον έλεγχο της απόδοσής του. Ακόμα, διασφαλίζεται πως ο τυχαίος διαχωρισμός θα είναι ίδιος κάθε φορά, κάτι εξαιρετικά χρήσιμο για την επαναληψιμότητα πειραμάτων σε ακαδημαϊκό πλαίσιο. Τέλος, λαμβάνει χώρα η εκπαίδευση του μοντέλου βάσει των παραπάνω παραμέτρων. Έγινε επιλογή εκπαίδευσης ενός μικρού μοντέλου μηχανικής μάθησης για την καλύτερη διαχείρισή του από τον μικροελεγκτή ESP32. Αναλυτικότερα, τα δέντρα απόφασης του μοντέλου έχουν μέγιστο βάθος 4, το σύνολο των δέντρων που δημιουργούνται είναι 50 και έχει ρυθμό εκπαίδευσης 0.1.

4.4.1 Μετατροπή Μοντέλου για Edge AI και Χρήση της Βιβλιοθήκης Treelite

Το Edge AI (Τοπική Τεχνητή Νοημοσύνη) αναφέρεται στην εκτέλεση αλγορίθμων μηχανικής μάθησης σε συσκευές όπως αισθητήρες, φορητά ηλεκτρονικά και μικροελεγκτές, χωρίς την χρήση απομακρυσμένων διακομιστών ή του νέφους (cloud). Η συγκεκριμένη εφαρμογή της τεχνητής νοημοσύνης, επιτρέπει αυτούς τους αλγορίθμους να επεξεργάζονται δεδομένα ακόμα και χωρίς σύνδεση στο διαδίκτυο. Ακόμα, εφόσον η επεξεργασία γίνεται τοπικά, η απόκριση των συστημάτων που την αξιοποιούν γίνεται σε πραγματικό χρόνο, χωρίς τις καθυστερήσεις που λαμβάνουν χώρα κατά την δρομολόγηση των δεδομένων μέσω του διαδικτύου [45].

Στα πλαίσια της παρούσας διπλωματικής εργασίας, έγινε αξιοποίηση αυτής της τεχνολογίας. Πιο συγκεκριμένα το μοντέλο που αναλύθηκε στην αρχή του κεφαλαίου 4.4 μεταγλωττίζεται σε γλώσσα C για να μπορεί να μεταφορτωθεί στον ESP32. Η διαδικασία αυτή γίνεται με την χρήση της βιβλιοθήκης Treelite, η οποία δέχεται σαν είσοδο ένα ήδη εκπαιδευμένο μοντέλο μηχανικής μάθησης σε Python και το μετατρέπει σε κώδικα C για χρήση σε μικροελεγκτές.

```
...
```

```
print("\nConverting XGBoost model to Treelite format...")
tl_model = treelite.frontend.from_xgboost(best_model.get_booster())

print("Compiling into C code...")
tl2cgen.generate_c_code(tl_model, dirpath="./gesture_model_c_treelite",
params={"parallel_comp": 5})
```

```
print("Treelite/TL2cgen Export Complete!")
```

```
...
```

Στον μεταγλωττισμένο κώδικα δημιουργείται η μέθοδος `predict`, η οποία χρησιμοποιείται για την πρόσβαση του μικροελεγκτή στο εκπαιδευμένο μοντέλο μηχανικής μάθησης. Πιο συγκεκριμένα, η βιβλιοθήκη Treelite διασπάει τον κώδικα σε δύο βασικά αρχεία και έπειτα δίνει την επιλογή περαιτέρω διάσπασής του για την αποδοτικότερη αξιοποίησή του από το υλικό της συσκευής στην οποία θα μεταφορτωθεί. Τα αρχεία αυτά είναι το `header.c` και το `model.c`, εκ των οποίων το πρώτο χρησιμοποιείται για την συμπερίληψη του μοντέλου στον πηγαίο κώδικα του προγράμματος που θα υλοποιηθεί και το δεύτερο είναι το μεταγλωττισμένο μοντέλο σε γλώσσα προγραμματισμού C.

```
// Αρχείο header.h
```

```
...
```

```
void predict(union Entry* data, int pred_margin, float* result);
```

```
void postprocess(float* result);
```

```
void predict_unit0(union Entry* data, float* result);
```

```
void predict_unit1(union Entry* data, float* result);
```

```
void predict_unit2(union Entry* data, float* result);
```

```
void predict_unit3(union Entry* data, float* result);
```

```
void predict_unit4(union Entry* data, float* result);
```

```
// Αρχείο model.c
```

```
#include "header.h"
```

```
...
```

```
void predict(union Entry* data, int pred_margin, float* result) {
```

```
    unsigned int tmp;
```

```
    predict_unit0(data, result);
```

```
    predict_unit1(data, result);
```

```
    predict_unit2(data, result);
```

```
    predict_unit3(data, result);
```

```
    predict_unit4(data, result);
```

```
    // Apply base_scores
```

```

result[0] += -6.29425048828125e-05;
result[1] += 6.306171417236328125e-05;
result[2] += 0.0003150701522827148438;
result[3] += -6.29425048828125e-05;
result[4] += 6.306171417236328125e-05;
result[5] += -0.0003150701522827148438;

// Apply postprocessor
if (!pred_margin) { postprocess(result); }
}

// Apply postprocessor for a single target
static void postprocess_impl(float* target_result, int num_class) {
    float max_margin = target_result[0];
    double norm_const = 0.0;
    float t;
    for (int k = 1; k < num_class; ++k) {
        if (target_result[k] > max_margin) {
            max_margin = target_result[k];
        }
    }
    for (int k = 0; k < num_class; ++k) {
        t = expf(target_result[k] - max_margin);
        norm_const += t;
        target_result[k] = t;
    }
    for (int k = 0; k < num_class; ++k) {
        target_result[k] /= (float)norm_const;
    }
}
...

```

4.5 Σύγκριση Αποτελεσμάτων και Αξιολόγηση Μοντέλου

Για την αξιολόγηση μοντέλων μηχανικής μάθησης υπάρχουν κάποιοι δείκτες, οι οποίοι χαρακτηρίζουν την αποδοτικότητά του. Αρχικά, οι προβλέψεις των μοντέλων μηχανικής μάθησης διαχωρίζονται σε

τέσσερις τύπους, αυτούς των σωστών προβλέψεων TP (True Positives) και TN (True Negatives), και αυτές των σφαλμάτων ταξινόμησης FP (False Positives) και FN (False Negatives). Μέσω αυτών, υπολογίζονται και οι βασικοί δείκτες που περιγράφουν την απόδοση του μοντέλου.

Η Ακρίβεια (Accuracy), περιγράφει την συνολική ποσοστιαία ικανότητα του μοντέλου, και ορίζεται ως ο λόγος των σωστών προβλέψεων προς το σύνολο των δειγμάτων:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (4.11)$$

Ωστόσο ο συγκεκριμένος δείκτης μεμονωμένα δεν αρκεί για την ανάλυση ενός μοντέλου με σκοπό την ανάλυση ηλεκτρομυογραφικών σημάτων. Πιο συγκεκριμένα, πέρα από την ακρίβεια, υπάρχουν και οι δείκτες Πιστότητας (Precision) και Ανάκλησης (Recall), οι οποίοι επιτρέπουν την κατανόηση της συμπεριφοράς του μοντέλου για κάθε κλάση ξεχωριστά:

$$Precision = \frac{TP}{TP + FP} \quad (4.12)$$

$$Recall = \frac{TP}{TP + FN} \quad (4.13)$$

Τέλος, ο υπολογισμός του μέσου όρου των δύο παραπάνω τιμών έχει ως αποτέλεσμα των δείκτη F1-Score, προσφέροντας μια ισορροπημένη εικόνα απόδοσης, ειδικά σε περιπτώσεις όπου τα δεδομένα μπορεί να παρουσιάζουν μικρές ανισοροπίες.

Για την αξιολόγηση των αποτελεσμάτων του μοντέλου, έγινε χρήση τυχαίων δειγμάτων από το σύνολο δεδομένων που χρησιμοποιήθηκε κατά την εκπαίδευσή του. Σε προηγούμενο κεφάλαιο αναφέρθηκε πως για την αξιολόγηση του μοντέλου χρησιμοποιούνται δύο υποσύνολα, αυτό της δοκιμής και αυτό της εκπαίδευσης. Ακόμα, έγινε αναφορά στο διαχωρισμό των αρχικών καταγραφών, με το δοκιμαστικό υποσύνολο να δεσμεύει το 20% αυτών.

```

Actual Gesture: FIST
Predicted Gesture: FIST ✓ CORRECT
Sensor Data: {'MAV': 0.2812, 'RMS': 0.3668, 'WL': 32.8037, 'ZC': 34.0, 'SSC': 96.0, 'Peak_FFT': 173.9624, 'Mean_FFT': 3.5198}

Actual Gesture: INDEX
Predicted Gesture: INDEX ✓ CORRECT
Sensor Data: {'MAV': 0.0643, 'RMS': 0.0853, 'WL': 18.7487, 'ZC': 81.0, 'SSC': 115.0, 'Peak_FFT': 20.4571, 'Mean_FFT': 1.4533}

Actual Gesture: LITTLE
Predicted Gesture: LITTLE ✓ CORRECT
Sensor Data: {'MAV': 0.0588, 'RMS': 0.0775, 'WL': 14.9875, 'ZC': 95.0, 'SSC': 87.0, 'Peak_FFT': 18.2593, 'Mean_FFT': 1.2078}

Actual Gesture: MIDDLE
Predicted Gesture: INDEX ✗ INCORRECT
Sensor Data: {'MAV': 0.0526, 'RMS': 0.0864, 'WL': 17.7622, 'ZC': 119.0, 'SSC': 133.0, 'Peak_FFT': 17.6877, 'Mean_FFT': 1.524}

Actual Gesture: NOISE
Predicted Gesture: RING ✗ INCORRECT
Sensor Data: {'MAV': 0.0249, 'RMS': 0.0322, 'WL': 15.0714, 'ZC': 233.0, 'SSC': 133.0, 'Peak_FFT': 8.8998, 'Mean_FFT': 0.8076}

Actual Gesture: RING
Predicted Gesture: RING ✓ CORRECT
Sensor Data: {'MAV': 0.0244, 'RMS': 0.0299, 'WL': 12.7918, 'ZC': 192.0, 'SSC': 121.0, 'Peak_FFT': 6.3945, 'Mean_FFT': 0.722}

```

Σχήμα 4.7 Αποτελέσματα δοκιμής του μοντέλου

Ακόμα, το παρακάτω απόσπασμα κώδικα δείχνει τον τρόπο με τον οποίο έγινε ο υπολογισμός όλων των παραπάνω δεικτών.

```

...
# 7. Evaluate the Model

```

```

y_pred = best_model.predict(X_test)

y_test_labels = label_encoder.inverse_transform(y_test)
y_pred_labels = label_encoder.inverse_transform(y_pred)

accuracy = accuracy_score(y_test_labels, y_pred_labels)
classification_rep = classification_report(y_test_labels, y_pred_labels)

print(f"\nOverall Accuracy: {accuracy*100:.2f}%")
print("\nClassification Report:\n", classification_rep)

# 8. Test One Sample from Each Class
print("\n" + "="*50)
print("TESTING ONE SAMPLE FROM EACH CLASS")
print("="*50)

unique_classes = label_encoder.classes_

for gesture in unique_classes:
    gesture_encoded = label_encoder.transform([gesture])[0]
    sample_index = (y_test == gesture_encoded).argmax()

    sample = X_test.iloc[[sample_index]]

    prediction_encoded = best_model.predict(sample)
    prediction_label = label_encoder.inverse_transform(prediction_encoded)[0]

    is_correct = "✅ CORRECT" if gesture.upper() == prediction_label.upper() else
    "❌ INCORRECT"

    print(f"\nActual Gesture:    {gesture.upper()}")
    print(f"Predicted Gesture: {prediction_label.upper()} {is_correct}")

# Print the sensor values rounded for readability

```

```

sample_dict = sample.iloc[0].round(4).to_dict()
print(f"Sensor Data:      {sample_dict}")...

```

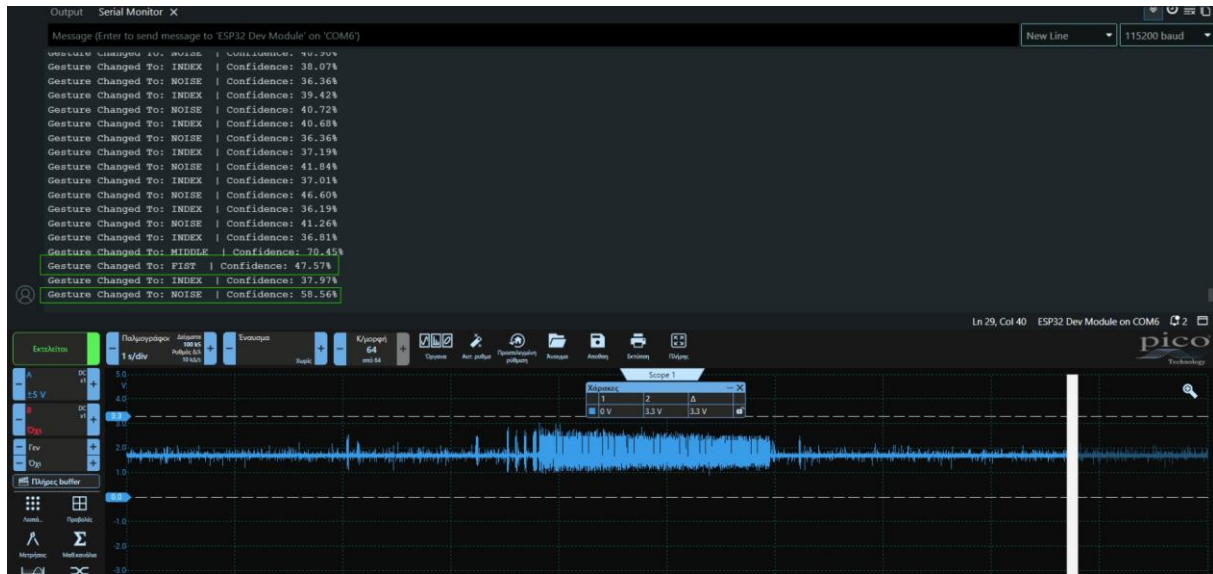
Στο στάδιο αυτό, τα αποτελέσματα των δεικτών αρχικά έδειξαν μια συνολική ακρίβεια της τάξης του 64.5%. Αν και ο συγκεκριμένος αριθμός μπορεί να φαίνεται μικρός, το σύνολο δεδομένων ήταν αρκετά μικρό για την πιο αποτελεσματική εκπαίδευση του μοντέλου. Η κλάση FIST και NOISE, ωστόσο παρουσιάζουν τα υψηλότερα ποσοστά επιτυχίας με υψηλή τιμή σε όλους τους δείκτες τους καθώς και τα ηλεκτρομυογραφικά τους πρότυπα είναι σαφώς διακριτά. Αντίθετα, παρουσιάζονται διακυμάνσεις στην απόδοση προβλεψιμότητας στις κλάσεις που αφορούν μεμονωμένες κινήσεις δαχτύλων. Πιο συγκεκριμένα, οι κλάσεις RING και LITTLE, συγχέονται μεταξύ τους καθώς ενεργοποιούν σχεδόν το ίδιο μέρος του μυ. Ο παρακάτω πίνακας απεικονίζει λεπτομερώς τις τιμές των δεικτών που αποδίδει το μοντέλο για κάθε κλάση.

Πίνακας 4.5 Τιμές δεικτών του μοντέλου

	Precision	Recall	F1-Score
FIST	0.72	0.79	0.75
INDEX	0.56	0.58	0.57
LITTLE	0.55	0.68	0.61
MIDDLE	0.53	0.23	0.32
NOISE	0.83	0.94	0.88
RING	0.61	0.65	0.63

Όπως μπορεί να παρατηρηθεί στην παρακάτω εικόνα (Σχήμα 4.8) το σύστημα κατηγοριοποιεί επιτυχώς ηλεκτρομυογραφικά σήματα σε πραγματικό χρόνο. Πιο συγκεκριμένα, η κίνηση FIST δημιουργεί ένα έντονο σήμα το οποίο αφότου περάσει από το απαραίτητο φιλτράρισμα και την προσαρμογή, λειτουργεί ως είσοδος στον ESP32, όπου το λογισμικό του σε συνεργασία με το μοντέλο μηχανικής μάθησης μπορεί να κατηγοριοποιήσει επιτυχώς το σήμα, δίνοντας και ένα ποσοστό αυτοπεποίθησης σε κάθε έξοδο. Η αδράνεια του μυ, ή αλλιώς NOISE, επίσης μπορεί να παρατηρηθεί πως ανιχνεύεται επιτυχώς

από το σύστημα, καθώς όπως αναλύθηκε, οι δύο αυτές κινήσεις έχουν τα πιο εύκολα ανιχνεύσιμα χαρακτηριστικά στο σήμα τους.



Σχήμα 4.8 Κατηγοριοποίηση κινήσεων σε πραγματικό χρόνο

4.6 Επίλογος

Ανακεφαλαιώνοντας, η εκπαίδευση μοντέλων μηχανικής μάθησης εξελίσσεται ραγδαία όπως και η εφαρμογή τους σε βιολογικά σήματα. Πιο συγκεκριμένα, η εξαγωγή χαρακτηριστικών για τέτοιου είδους σύνολα δεδομένων έχει μελετηθεί εκτενώς, δίνοντας πάτημα σε νέους επιστήμονες να δημιουργούν γρήγορα πρωτότυπες εφαρμογές. Ακόμα, η προεπεξεργασία των δεδομένων που καταγράφονται γίνεται ολοένα και πιο εύκολα με ένα ευρύ φάσμα βιβλιοθηκών της γλώσσας προγραμματισμού Python, που έχουν φτιαχτεί για αυτόν ακριβώς τον λόγο. Περαιτέρω, αλγόριθμοι όπως αυτός των Τυχαίων Δασών και του XGBoost γίνονται ακόμα πιο γνωστοί καθώς και η εφαρμογή τους ευκολότερη. Επίσης, η χρήση αυτών των αλγορίθμων μπορεί πλέον να ενσωματωθεί με μεγάλη ευκολία σε μικροελεγκτές καθιστώντας τη διαδικασία δημιουργίας πρωτοτύπων ταχύτερη με την χρήση του Edge AI. Τέλος, οι δείκτες για την αξιολόγηση ενός μοντέλου πρέπει πάντοτε να λαμβάνονται υπόψη και να λειτουργούν ως έναυσμα για να λάβουν μέρος οι κατάλληλες ενέργειες.

Κεφάλαιο 5ο: Συμπεράσματα και Προτάσεις Βελτίωσης

5.1 Εισαγωγή

Στα προηγούμενα κεφάλαια έγινε μια εκτενής ανάλυση του διεπιστημονικού θέματος της παρούσας διπλωματικής εργασίας. Πιο συγκεκριμένα, στο πρώτο κεφάλαιο έγινε η εισαγωγή τους θέματος με μια ιστορική αναδρομή, και μια σύντομη περιγραφή κάποιων από τις εφαρμογές της τεχνολογίας της ηλεκτρομυογραφίας. Ακόμα, ορίστηκε η σκοπιά και η δομή της εργασίας. Στο δεύτερο κεφάλαιο, αναλύθηκε εκτενώς η φυσιολογία του μυ Flexor Digitorum Superficialis και προσδιορίστηκαν οι μηχανισμοί γέννησης μυϊκών σημάτων στο ανθρώπινο σώμα μαζί με τις μαθηματικές εξισώσεις που τους περιγράφουν. Στο τρίτο κεφάλαιο αποτυπώθηκε ο τρόπος υλοποίησης του ηλεκτρονικού κυκλώματος του συστήματος καθώς και τα εργαλεία τα οποία αξιοποιήθηκαν κατά τη διαδικασία αυτή. Τέλος στο τέταρτο κεφάλαιο παρατέθηκε αρχικά η βασική θεωρία της μηχανικής μάθησης και η δημιουργία του μοντέλου του συστήματος που απασχολεί την εργασία.

Εν κατακλείδι, αναπτύχθηκε ένα πλήρες σύστημα ανίχνευσης ηλεκτρομυογραφικών σημάτων σε πραγματικό χρόνο. Η απόδοσή του και η ορθή του λειτουργία του επιβεβαιώνεται από τις τιμές που έχουν παρατεθεί στο μήκος όλων των προηγούμενων κεφαλαίων.

Το παρών και τελευταίο κεφάλαιο θα επικεντρωθεί στη σύνοψη όλων των παρατηρήσεων από τα προηγούμενα και θα προσφέρει αφενός προτάσεις για την μελλοντική βελτίωση του συστήματος, αφετέρου εμπορικές προοπτικές που θα μπορούσε να έχει.

5.2 Σύνοψη Παρατηρήσεων

Πριν γίνει η ανακεφαλαίωση όλων των παρατηρήσεων των προηγούμενων κεφαλαίων, είναι απαραίτητο να αναφερθεί το συνολικό κόστος του συστήματος. Ξεκινώντας από τους ενισχυτές του κυκλώματος χρησιμοποιήθηκαν ένας INA128 με τιμή περίπου στα 13€ και έξι τελεστικοί ενισχυτές TL072 με τιμή 1.50€ έκαστος. Ακόμα, έγινε χρήση έντεκα ποτενσιόμετρων με τιμή από 0.5€ έως 1€. Τέλος, οι πυκνωτές και οι αντιστάσεις που χρησιμοποιήθηκαν έχουν προσεγγιστικά μία τιμή γύρω στα 2€-5€ και ο μικροελεγκτής ESP32 αυτή τη στιγμή στη αγορά μπορεί να βρεθεί στα 9€-10€. Λαμβάνοντας την ακριβότερη τιμή στα δεδομένα περιθώρια κάθε στοιχείου έχουμε προσεγγιστικά μία συνολική τιμή των 48€.

Πίνακας 5.1 Ανάλυση κόστους του συστήματος

Στοιχεία	Ποσότητα	Τιμή
INA128	1	13€
TL072	6	1,5€
Ποτενσιόμετρα	11	0,5€-1€
Πυκνωτές και Αντιστάσεις	~20	2€-5€
ESP32	1	9€-10€
Σύνολο		38,5€-48€

Είναι επίσης σημαντικό να σημειωθεί πως εμπορικά συστήματα που πωλούνται αυτή τη στιγμή στη αγορά έχουν την τριπλάσια τιμή και παραπάνω, από αυτήν του συστήματος που υλοποιήθηκε στα πλαίσια αυτή της διπλωματικής εργασίας. Πιο συγκεκριμένα, το Myo Armband των Thalmic Labs κοστίζει \$149 καθώς και τα πιο εξειδικευμένα συστήματα για χρήση σε κλινικά περιβάλλοντα μπορούν να φτάσουν τα χιλιάδες δολάρια σε τιμή. Με βάση το παραπάνω, μια ολοκληρωμένη συσκευή ηλεκτρομυογραφίας με ενσωματωμένο ένα μοντέλο μηχανικής μάθησης για την κατηγοριοποίηση κινήσεων με τιμή 38,5€-48€, καθιστά μία πολύ οικονομική και προσιτή λύση σε σχέση με τα υπάρχοντα προϊόντα.



Σχήμα 5.1 Myo Armband [46]

Το χαμηλό κόστος υλοποίησης μαζί με την αποδοτικότητα του συστήματος το αναδεικνύουν σε μία εξαιρετική εναλλακτική. Αρχικά, τα αναλογικά φίλτρα που περιέχει το κύκλωμα και η προσαρμογή του σήματος για να μπορεί να λειτουργήσει σαν είσοδο σε μικροελεγκτή, προσδίδουν στο σύστημα ιδιαίτερη αξία, καθώς θέτουν την ανάπτυξη πρωτότυπων εφαρμογών με την χρήση ηλεκτρομυογραφικών σημάτων ακόμα πιο ταχεία. Ακόμα, η ενσωμάτωση ενός μοντέλου μηχανικής μάθησης στο τελικό σύστημα μπορεί να τυποποιηθεί και να αποδώσει ακόμα πιο γρήγορους χρόνους υλοποίησης εφαρμογών.

Ειδικότερα, όπως αναλύθηκε και σε προηγούμενο κεφάλαιο τα φίλτρα του αναλογικού κυκλώματος, απαλείφουν όλες τις ανεπιθύμητες συχνότητες από το τελικό σήμα. Πιο συγκεκριμένα, τα ηλεκτρομυογραφικά σήματα παρουσιάζουν χρήσιμη πληροφορία σε συχνότητες από 10Hz έως 500Hz, βάσει των οποίων σχεδιάστηκαν τα κατάλληλα φίλτρα δευτέρου βαθμού, ώστε να επιτρέπονται μόνο οι ενδιάμεσες συχνότητες του σήματος. Περαιτέρω, έπρεπε να αφαιρεθεί η συχνότητα των 50Hz καθώς αποτελεί θόρυβο που προέρχεται από το ηλεκτρικό δίκτυο. Για τον λόγο αυτό κατασκευάστηκε ένα φίλτρο εγκοπής με τοπολογία Twin-T για να είναι πιο αποδοτική η απόσβεση του σήματος γύρω από αυτή τη συχνότητα. Οι παραπάνω τοπολογίες σε συνδυασμό με το CMRR του ενισχυτή οργανολογίας INA128 απέδωσαν ένα καθαρό σήμα στην έξοδο του κυκλώματος.

Καθώς τα ηλεκτρομυογραφικά σήματα είναι πολύ μικρά σε πλάτος, χρήζουν σημαντική ενίσχυση για να αναλυθούν αποτελεσματικά. Για τον λόγο αυτό, σχεδιάστηκαν δύο βαθμίδες ενίσχυσης, αρχικά κατά

την είσοδο του σήματος στο κύκλωμα μέσω του INA128 και έπειτα στο υπερυπεραυτό φίλτρο. Η αποτελεσματική ενίσχυση του σήματος βοήθησε στην μεταφορά του δια μέσω των παθητικών στοιχείων του κυκλώματος χωρίς απώλειες ως προς το πλάτος του.

Ο μικροελεγκτής ESP32 απαιτεί το σήμα εισόδου στον ADC του, να έχει πλάτος από 0V έως 3.3V. Για την προσαρμογή του σήματος EMG σε αυτά τα όρια, υλοποιήθηκε μια βαθμίδα υπο-ενίσχυσης καθώς το καταγραφόμενο σήμα είχε ως μέγιστο πλάτος περίπου 10V από κορυφή σε κορυφή. Σε αυτή τη βαθμίδα έγινε και υπερύψωση του σήματος, ώστε να βρίσκεται γύρω από τα αποδεκτά όρια. Με αυτόν τον τρόπο επιτεύχθηκε η ασφαλής είσοδος του επεξεργασμένου σήματος στον μικροελεγκτή, χωρίς να προκαλέσει βλάβες στο υλικό του.

Τέλος η δημιουργία ενός μοντέλου μηχανικής μάθησης και η ενσωμάτωσή του στον ESP32 αποτελεί το τελευταίο στάδιο της υλοποίησης. Πιο συγκεκριμένα, τα καταγεγραμμένα δεδομένα κάθε κλάσης μετασχηματίστηκαν ώστε να αντικατοπτρίζουν τα χαρακτηριστικά που επιλέχθηκαν και χρησιμοποιήθηκαν για την εκπαίδευση και επιβεβαίωση ενός αλγορίθμου XGBoost. Το τελικό μοντέλο είχε ακρίβεια 64.5%, εμφανίζοντας δυσκολίες στην κατηγοριοποίηση συγκεκριμένων κλάσεων.

5.3 Προτάσεις για Μελλοντικές Βελτιώσεις

Σαν αρχικός στόχος βελτίωσης είναι η μεταφορά του συστήματος από την πλακέτα ανάπτυξης σε μία πλακέτα τυπωμένου κυκλώματος. Έτσι, το μέγεθος του υλικού θα μικραίνει κατά πολύ, θέτοντάς το φορητό. Ειδικότερα, τα μεγάλα στοιχεία του κυκλώματος θα αντικατασταθούν με τα αντίστοιχα επιφανειακά στηριζόμενα εξαρτήματα (Surface Mount Components - SMDs), τα οποία έχουν πολύ μικρότερο μέγεθος από τα υπάρχοντα. Ο συγκεκριμένος στόχος μπορεί να επιτευχθεί με την σχεδίαση ενός PCB μέσω κάποιου προγράμματος σχεδίασης (Computer-Aided Design - CAD) και αναθέτοντας την κατασκευή του σε κάποια εταιρία που εξειδικεύεται σε αυτόν τον τομέα.

Η σμίκρυνση του συστήματος καθαυτή, δεν το καθιστά αυτόματα φορητό αλλά πρέπει να σχεδιαστεί και ένα περίβλημα τύπου περικάρπιου. Πιο συγκεκριμένα, όπως όλα τα αντίστοιχα προϊόντα στην αγορά, για να γίνει το σύστημα φορητό χρειάζεται ένα περίβλημα που να επιτρέπει στον χρήστη να το περιδέσει επάνω του. Για τον σκοπό αυτό μπορεί να σχεδιαστεί ένα περικάρπιο περίβλημα, ώστε το σύστημα να μπορεί να φορεθεί σαν ρολόι.

Αυτή τη στιγμή το κύκλωμα, όπως προαναφέρθηκε, τροφοδοτείται από δύο πηγές τάσης στα $\pm 15V$, γεγονός που αποτρέπει την φορητότητά του. Για την λύση του συγκεκριμένου ζητήματος, πρέπει να γίνει επανασχεδιασμός του συστήματος με τροφοδοσία μπαταρίας. Ακόμα, πρέπει να γίνουν ξανά οι υπολογισμοί ενίσχυσης και προσαρμογή του σήματος για να βρίσκεται στα όρια της τάσης της νέας πηγής τροφοδοσίας.

Για την καλύτερη κατηγοριοποίηση των κινήσεων σε πραγματικό χρόνο, πρέπει να ληφθούν περισσότερες καταγραφές δεδομένων για την εκπαίδευση του μοντέλου μηχανικής μάθησης του συστήματος. Ειδικότερα, όπως αναλύθηκε σε προηγούμενο κεφάλαιο, έγιναν 20 ξεχωριστές καταγραφές για κάθε μία από τις 6 κινήσεις, με το μοντέλο μηχανικής μάθησης να αποδίδει 64.5% ακρίβεια. Ακόμα, από τις 120 αυτές καταγραφές, τα δεδομένα συμπίεστηκαν ακόμα παραπάνω για την ενσωμάτωση του sliding window. Μελλοντικά, θα μπορούσε να ελεγχθεί το αν με την αύξηση των καταγραφών, το μοντέλο μηχανικής μάθησης εκπαιδευτεί αποδοτικότερα και αποδώσει καλύτερα στατιστικά.

Το σύστημα έχει τη δυνατότητα ανίχνευσης μίας μόνο κίνησης κάθε δεδομένη στιγμή, γεγονός που καθιστά περιορισμένη την χρήση του. Πιο συγκεκριμένα, οι κινήσεις που μπορούν να ανιχνευθούν από

το σύστημα είναι μετρημένες, καθώς έτσι είναι δομημένη η φυσιολογία του μυός Flexor Digitorum Superficialis. Με την προσθήκη περισσότερων καναλιών γίνεται εφικτή η ανίχνευση περισσότερων κινήσεων ταυτόχρονα, δίνοντας τη δυνατότητα καταγραφής συνδυασμών αυτών, με αποτέλεσμα να γίνει μεγαλύτερο το διαθέσιμο εύρος τους.

Η λειτουργία αυτή μπορεί να υλοποιηθεί με την ενσωμάτωση περισσότερων ηλεκτροδίων στην είσοδο του κυκλώματος. Αναλυτικότερα, στην περίπτωση που υλοποιηθεί το περίβλημα του συστήματος, θα ήταν δυνατό να προσαρτηθούν μικρές μεταλλικές επαφές, οι οποίες θα χρησιμοποιούνταν για την καταγραφή των ηλεκτρομυογραφικών σημάτων. Με αυτόν τον τρόπο και την κατάλληλη διαχείρισή τους μέσω του κώδικα και του ηλεκτρονικού κυκλώματος, γίνεται εφικτή η καταγραφή πολλαπλών σημάτων ταυτόχρονα, χωρίς μεγάλη σχεδιαστική επέμβαση στο υπάρχοντα σύστημα.

Για τον πιο επαγγελματικό και ορθό σχεδιασμό του συστήματος, φεύγοντας από την φάση ανάπτυξης πρωτοτύπου, πρέπει να γίνει χρήση ενός πιο ισχυρού μικροελεγκτή. Πιο συγκεκριμένα, αν και ο ESP32 είναι αρκετά ισχυρός, ο nRF52840 της εταιρείας Nordic Semiconductor αποτελεί μία καλύτερη εναλλακτική για την κατασκευή ενός πιο ολοκληρωμένου συστήματος. Η χαμηλή του κατανάλωση και η 32-bit ARM αρχιτεκτονική που ακολουθεί, τον καθιστούν κατάλληλο τόσο για την ταχεία επεξεργασία των σύνθετων σημάτων ηλεκτρομυογραφίας, όσο και για την διαχείριση του μοντέλου μηχανικής μάθησης που θα μεταφορτωθεί σε αυτόν.



Σχήμα 5.2 Μικροελεγκτής nRF52840 [47]

Ο συγκεκριμένος μικροελεγκτής έχει ενσωματωμένα διάφορα πρωτόκολλα επικοινωνίας, εκ των οποίων ένα είναι το Zigbee. Το συγκεκριμένο πρωτόκολλο χρησιμοποιείται κυρίως για δημιουργία μικρών προσωπικών δικτύων και έχει χαμηλή κατανάλωση ισχύος, κάτι που είναι εξαιρετικά χρήσιμο για ένα σύστημα που τροφοδοτείται από μπαταρίες. Το εύρος των ορίων της απόστασης που επιτρέπει το πρωτόκολλο την μετάδοση πληροφοριών είναι 10-100 μέτρα με δυνατότητα επέκτασής του, κατασκευάζοντας ένα πλεγματο δίκτυο (mesh network) από ενδιάμεσους κόμβους. Με την χρήση του συγκεκριμένου πρωτοκόλλου γίνεται αποφυγή των καθυστερήσεων που εμφανίζονται κατά την δρομολόγηση πακέτων εντός του διαδικτύου, και δίνεται η δυνατότητα κατασκευής ενός πλήρους τοπικού συστήματος [48].

Ένα παράδειγμα εφαρμογής αυτής της αρχιτεκτονικής θα μπορούσε να είναι η ενσωμάτωσή του σε ένα εργοστάσιο. Πιο συγκεκριμένα, οι εργάτες θα μπορούσαν να χρησιμοποιούν το σύστημα για τον

χειρισμό μηχανημάτων εντός του εργοστασίου που μπορεί να έθετε την φυσική τους ακεραιότητα σε κίνδυνο. Ακόμη, με την χρήση του παραπάνω μικροελεγκτή και του πρωτοκόλλου Zigbee, η απόκριση του συστήματος θα ήταν πολύ μικρή, με αποτέλεσμα την καλή εμπειρία του χρήστη και τον ορθό χειρισμό των μηχανημάτων γραμμής παραγωγής, ανύψωσης ή οποιοδήποτε άλλο μηχάνημα μπορεί να κριθεί επικίνδυνο για την φυσική ακεραιότητα του εργάτη.

Ακόμα, θα ήταν χρήσιμο να γίνει διαχωρισμός του συστήματος σε δύο υποσυστήματα, αυτό της ανάπτυξης και αυτό της παραγωγής. Ειδικότερα, η πλατφόρμα της ανάπτυξης θα χρησιμοποιούταν για την καταγραφή δεδομένων των επιθυμητών κινήσεων και σε συνέχεια θα γινόταν εκπαίδευση του μοντέλου μηχανικής μάθησης πάνω σε αυτά τα δεδομένα. Μετά, το ήδη εκπαιδευμένο μοντέλο θα μεταφορτωνόταν στον μικροελεγκτή που βρίσκεται στο υποσύστημα της παραγωγής, θέτοντας έτσι εφικτή την κατασκευή δυναμικών συστημάτων και όχι μεμονωμένων λύσεων.

Για την ευκολότερη χρήση του υποσυστήματος ανάπτυξης, θα ήταν χρήσιμο να αναπτυχθεί μία εφαρμογή διαχείρισης των καταγραφών. Αναλυτικότερα, αυτό θα βοηθούσε στην αρχειοθέτηση των καταγραφών και στην ευκολότερη εκπαίδευση και μεταφόρτωση του μοντέλου μηχανικής μάθησης, καθώς τα δεδομένα θα βρίσκονται σε μία κεντρική πλατφόρμα.

5.4 Εμπορικές Προοπτικές και Καινοτομία

Το σύστημα για να λειτουργήσει, όπως έχει αναλυθεί εκτενώς στα προηγούμενα κεφάλαια, χρειάζεται έναν ικανοποιητικό όγκο δεδομένων για την ορθή και αποδοτική εκπαίδευση του μοντέλου μηχανικής μάθησης που ενσωματώνει. Στα πλαίσια μίας εμπορικής εφαρμογής του συστήματος, θα ήταν δύσκολο να εκπαιδεύει η εταιρία το μοντέλο πάνω στις κινήσεις που χρειάζεται ο πελάτης. Η λύση σε αυτό το πρόβλημα είναι η χρήση των υποσυστημάτων ανάπτυξης όπως αναλύθηκαν και στο προηγούμενο κεφάλαιο.

Θα λαμβάνει αρχικά ο πελάτης ένα περικάρπιο το οποίο περιβάλλει το υποσύστημα ανάπτυξης. Με την χρήση αυτού και της εφαρμογής που θα αναπτυχθεί, θα μπορεί ο πελάτης να κάνει την καταγραφή των δεδομένων που επιθυμεί και με ένα κουμπί να εκπαιδεύει το μοντέλο. Σε συνέχεια, η εταιρία θα έχει εμπιστευτική πρόσβαση στον κώδικα του μοντέλου που θα παραχθεί με σκοπό την μεταφόρτωσή του στα περικάρπια που ενσωματώνουν το υποσύστημα παραγωγής.

Με αυτή τη ροή δημιουργίας ξεχωριστών λύσεων για κάθε πελάτη δίνεται η δυνατότητα αφενός στην εταιρία να μην αφιερώνει χρόνο στην καταγραφή δεδομένων και να επικεντρωθεί στην περαιτέρω εξέλιξη του συστήματος, αφετέρου στον πελάτη να έχει πλήρη έλεγχο των δεδομένων που θα χρησιμοποιηθούν για το προϊόν που θα λάβει.

Ακόμα η χρήση του συστήματος θα μπορούσε να επεκταθεί και σε άλλους μύες εκτός του Flexor Digitorum Superficialis. Πιο συγκεκριμένα, ο πελάτης θα μπορούσε να παραγγείλει το ίδιο σύστημα με την εφαρμογή για παράδειγμα σε κάποιο σημείο του ποδιού. Η δουλειά της εταιρείας θα ήταν να βρει τον κατάλληλο μυ για την εκάστοτε εφαρμογή και να προσαρμόσει το περίβλημα για να μπορεί να φορεθεί στο συγκεκριμένο σημείο του σώματος. Με αυτόν τον τρόπο η εταιρία θα πρόσφερε την καλύτερη δυνατή λύση στον πελάτη, ενισχύοντας τόσο το κέρδος της, όσο και την εμπιστοσύνη την οποία επιφέρει.

Το μοντέλο πωλήσεων που ταιριάζει περισσότερο στο συγκεκριμένο προϊόν είναι η πώληση από εταιρία σε εταιρία (Business to Business – B2B). Αναλυτικότερα, η δυναμική φύση της αρχιτεκτονικής του συστήματος το χρήζει σαν μια πλατφόρμα ανάπτυξης παρά ένα καταναλωτικό προϊόν. Οι εφαρμογές

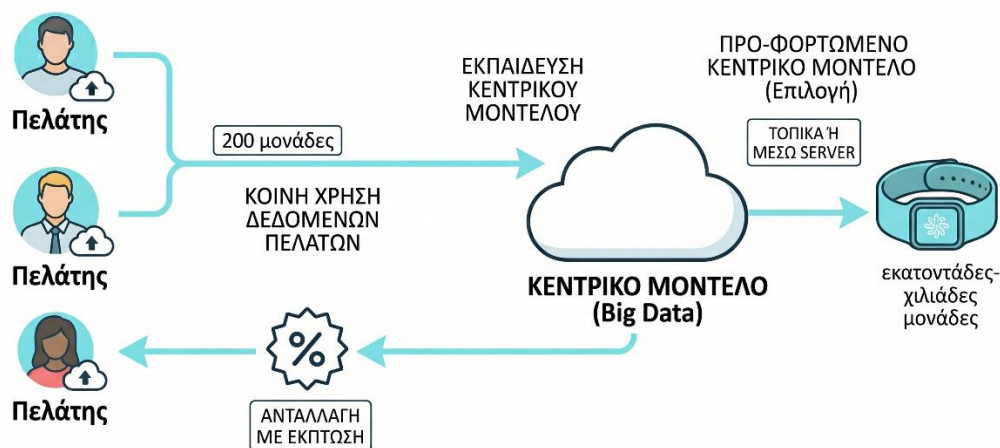
ποικίλουν από κλινικά μηχανήματα, σε χειρισμό της γραμμής παραγωγής εργοστασίων και πολλές ακόμα, οι οποίες είναι χρήσιμες κυρίως για άλλες εταιρίες και όχι για μεμονωμένους χρήστες – πελάτες.

Ακολουθώντας το μοντέλο B2B η εταιρία θα μπορούσε να μεγιστοποιήσει τα κέρδη της. Πέρα από την αρχιτεκτονική του συστήματος, το μοντέλο αυτό αποτελεί μία πολύ καλή ευκαιρία για μεγαλύτερα κέρδη. Πιο συγκεκριμένα, μια άλλη εταιρία, θα χρειαστεί εκατοντάδες αν όχι χιλιάδες μονάδες του συγκεκριμένου προϊόντος ενώ αν στόχος ήταν η πώληση σε μεμονωμένους πελάτες, η ζήτηση δεν θα έφτανε με τίποτα τόσο υψηλά. Ακόμα, θα μπορούσαν η εταιρίες – συνεργάτες να τιμολογούνται και για τις περαιτέρω υπηρεσίες που θα αναπτυχθούν, όπως η κεντρική πλατφόρμα και η έρευνα για την καλύτερη δυνατή λύση ως προς τις προδιαγραφές που ζητούν για το τελικό προϊόν.

Άλλη μία ροή εσόδων για την εταιρία θα μπορούσε να είναι η ανάπτυξη ενός κεντρικού μοντέλου μηχανικής μάθησης εκπαιδευμένο πάνω σε ηλεκτρομυογραφικά σήματα. Αναλυτικότερα, θα μπορούσε να υπάρχει η επιλογή για τον πελάτη να μοιραστεί τα δεδομένα που θα καταγράψει για την εφαρμογή του με την εταιρία με σκοπό της δημιουργίας ενός μεγάλου συνόλου δεδομένων που θα χρησιμοποιηθεί για την εκπαίδευση ενός μεγάλου κεντρικού μοντέλου μηχανικής μάθησης. Ο πελάτης αν επέλεγε να προχωρήσει με την κοινή χρήση των δεδομένων του θα λάμβανε ως αντάλλαγμα μία έκπτωση στην τελική του τιμολόγηση.

Όταν το κεντρικό αυτό μοντέλο μηχανικής μάθησης ήταν πλήρως εκπαιδευμένο, θα μπορούσε να δοθεί η επιλογή στους πελάτες να είναι ήδη μεταφορτωμένο στο τελικό προϊόν χωρίς να χρειάζεται η διαδικασία καταγραφής δεδομένων. Ειδικότερα, οι πελάτες θα μπορούσαν να ελέγξουν τις κινήσεις τις οποίες συμπεριλαμβάνει το κεντρικό μοντέλο μηχανικής μάθησης, και αν αυτό κάλυπτε τις ανάγκες τους θα μπορούσαν να το επιλέξουν.

DATA ECONOMY / SHARED CENTRAL MODEL



Σχήμα 5.3 Εκπαίδευση κεντρικού μοντέλου μηχανικής μάθησης

Ακόμα, η χρήση του θα μπορούσε είτε να μεταφερθεί τοπικά στον χώρο τους με την κατασκευή των απαραίτητων εγκαταστάσεων, είτε να γίνει χρήση του μοντέλου μέσω ενός διακομιστή της εταιρείας. Και στις δύο περιπτώσεις θα μπορούσε να γίνεται τιμολόγηση για τις περαιτέρω υπηρεσίες που προσφέρονται, αφενός για την κατασκευή των απαραίτητων εγκαταστάσεων καθώς αυτό προϋποθέτει να καταναλωθούν πόροι και να υπάρχει η κατάλληλη τεχνογνωσία από μεριά της εταιρείας αφετέρου η διαχείριση και συντήρηση του διακομιστή κοστίζει στην εταιρία και πρέπει με κάποιο τρόπο να καλύπτονται τα συγκεκριμένα έξοδα.

Για την διευκόλυνση ανάπτυξης εξατομικευμένων εφαρμογών είτε από τους πελάτες είτε από την ίδια την εταιρία, θα ήταν χρήσιμη η ανάπτυξη μίας διεπαφής προγραμματισμού εφαρμογών (Application Programming Interface - API). Όπως κάθε πλατφόρμα προσφέρει ένα API για την ανάπτυξη λύσεων μέσω αυτών, έτσι τίθεται απαραίτητο να υπάρχει και ένα API που να δίνει την δυνατότητα στους μηχανικούς τόσο εντός της εταιρείας όσο και της εταιρείας – πελάτη να μπορούν εύκολα να αναπτύξουν εφαρμογές με το τελικό προϊόν. Η χρήση του API θα γίνεται χωρίς χρέωση, ωστόσο θα μπορεί η εταιρία να προσφέρει υπηρεσίες ανάπτυξης των εξατομικευμένων εφαρμογών που ζητάνε οι πελάτες και να τους τιμολογεί και την έρευνα που θα λάβει χώρα και για την υλοποίηση.

Τέλος, είναι σημαντικό να σημειωθεί πως ενώ υπάρχουν αντίστοιχα προϊόντα στην αγορά, κανένα δεν πλησιάζει την χαμηλή τιμή της υλοποίησης του συστήματος που απασχόλησε την παρούσα διπλωματική εργασία. Αν και υπάρχουν προϊόντα με χαμηλή τιμή δεν έχουν ενσωματωμένες όλες τις λειτουργίες που μελετήθηκαν εκτενώς στα προηγούμενα κεφάλαια.

5.5 Επίλογος

Στο παρόν και τελευταίο κεφάλαιο έγινε η σύνοψη των παρατηρήσεων της υλοποίησης της κατασκευής καθώς και αναλύθηκαν τόσο προτάσεις για την μελλοντική βελτίωση του συστήματος όσο και οι εμπορικές προοπτικές του. Αρχικά, έγινε η ανάλυση κόστους του και έγινε σύγκριση με τα προϊόντα που είναι διαθέσιμα αυτή τη στιγμή στη αγορά. Σε συνέχεια, έλαβε χώρα σχολιασμός της αποτελεσματικότητας αφενός του υλικού μέρους του συστήματος αφετέρου του μοντέλου μηχανικής μάθησης που κατασκευάστηκε. Ακολούθησαν προτάσεις για την μελλοντική του βελτίωση, εκ των οποίων είναι η σμίκρυνσή του με σκοπό την φορητότητα, η πρόσθεση καναλιών για παράλληλη ανίχνευση κινήσεων σε πραγματικό χρόνο, η αλλαγή του μικροελεγκτή στον nRF52840, χρήση του πρωτοκόλλου Zigbee, ο διαχωρισμός του συστήματος σε αυτό της ανάπτυξης και αυτό της παραγωγής και η δημιουργία μιας κεντρικής πλατφόρμας για την διαχείριση των δεδομένων που καταγράφονται. Τέλος, προτάθηκε ένα επιχειρησιακό μοντέλο για την εμπορευματοποίηση του συστήματος. Το μοντέλο αυτό είναι B2B και αναλύθηκαν αρκετοί τρόποι για την δημιουργία ροών εσόδων.

Μέσα από την υλοποίηση του πρωτοτύπου που παρουσιάστηκε, γίνεται εύκολα κατανοητή η αξία του εκδημοκρατισμού της συγκεκριμένης τεχνολογίας. Καθώς και η μηχανική μάθηση και η τεχνητή νοημοσύνη εξελίσσονται ολοένα και πιο ραγδαία, η παρούσα διπλωματική εργασία αποδεικνύει πως η εφαρμογή τους στην ανίχνευση βιολογικών σημάτων με εξαιρετικά χαμηλό κόστος είναι εφικτή. Αυτό το συμπέρασμα είναι ιδιαίτερα σημαντικό για την ανάπτυξη καινοτόμων εφαρμογών που θα βοηθήσουν στην περαιτέρω έρευνα αυτού του τομέα και πιθανότατα θα διαταράξουν την αγορά με πολύ προηγμένα συστήματα.

ΒΙΒΛΙΟΓΡΑΦΙΑ

- [1] J. V. Basmajian and C. J. De Luca, *Muscles Alive: Their Functions Revealed by Electromyography*, 5th ed. Baltimore, MD: Williams & Wilkins, 1985.
- [2] A. J. Preston and B. E. Shapiro, *Electromyography and Neuromuscular Disorders: Clinical-Electrophysiologic Correlations*, 3rd ed. London, UK: Elsevier, 2012.
- [3] D. Farina, R. Merletti, and R. M. Enoka, "The extraction of neural strategies from the surface EMG: an update," *Journal of Applied Physiology*, vol. 96, no. 4, pp. 1486-1495, 2004.
- [4] K. Englehart and B. Hudgins, "A robust, real-time control scheme for multifunction myoelectric control," *IEEE Transactions on Biomedical Engineering*, vol. 50, no. 7, pp. 848-854, July 2003.
- [5] Meta Reality Labs, "Human-Computer Input via a Wrist-Based sEMG Wearable," *Meta Quest Blog*, 2024. [Online]. Available: <https://www.meta.com/blog/surface-emg-wrist-white-paper-reality-labs/>
- [6] T. R. Mendez Guerra et al., "A generic noninvasive neuromotor interface for human-computer interaction," *bioRxiv*, 2024. [Online]. Available: <https://www.biorxiv.org/content/10.1101/2024.02.23.581779v1>
- [7] "Electromyography and Meta Neural Wristbands: Combining Neural Interfaces for Better Human-Computer Interaction," *International Journal for Multidisciplinary Research (IJFMR)*, vol. 6, no. 4, 2024.

- [8] S. Naikar et al., "Embracing differences in virtual reality: inclusive user-centered design of bimanual interaction techniques," *Frontiers in Virtual Reality*, 2025.
- [9] A. Graf-Barthet, "Combining Vision and EMG-Based Hand Tracking for Extended Reality Musical Instruments," *Semantics Scholar*, 2024.
- [10] "EMG Based Gesture Recognition by Visual Method for Human-Machine Interaction and System Health Management," *ResearchGate*, 2023.
- [11] C. J. De Luca, "The use of surface electromyography in biomechanics," *Journal of Applied Biomechanics*, vol. 13, no. 2, pp. 135-163, 1997.
- [12] R. Merletti and P. A. Parker, *Electromyography: Physiology, Engineering, and Non-Invasive Applications*. Hoboken, NJ: IEEE Press, 2004.
- [13] G. L. Soderberg and T. M. Cook, "Electromyography in Biomechanics," *Physical Therapy*, vol. 64, no. 12, pp. 1813-1820, Dec. 1984.
- [14] E. Criswell, *Cram's Introduction to Surface Electromyography*. Sudbury, MA: Jones & Bartlett Publishers, 2010.
- [15] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge Computing: Vision and Challenges," *IEEE Internet of Things Journal*, vol. 3, no. 5, pp. 637-646, Oct. 2016.
- [16] T. Chen and C. Guestrin, "XGBoost: A Scalable Tree Boosting System," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016, pp. 785–794.
- [17] A. Phinyomark and E. Scheme, "EMG Pattern Recognition in the Era of Big Data and Deep Learning," *Big Data and Cognitive Computing*, vol. 2, no. 3, p. 21, 2018.
- [18] "Flexor digitorum superficialis muscle," in *Wikipedia*, May 10, 2026. [Online]. Available: https://en.wikipedia.org/wiki/Flexor_digitorum_superficialis_muscle.
- [19] P. Konrad, *The ABC of EMG: A Practical Introduction to Kinesiological Electromyography*, ver. 1.0. USA: Noraxon INC., Apr. 2005.
- [20] R. Plonsey and R. C. Barr, *Bioelectricity: A Quantitative Approach*, 3rd ed. New York, NY, USA: Springer, 2007.
- [21] F. L. H. Gielen, W. Wallinga-de Jonge, and K. L. Boon, "Electrical conductivity of skeletal muscle tissue: Experimental results from different muscles in vivo," *Med. Biol. Eng. Comput.*, vol. 22, no. 6, pp. 569–577, Nov. 1984. [Online]. Available: <https://scispace.com/pdf/electrical-conductivity-of-skeletal-muscle-tissue-562e6scc4w.pdf>.
- [22] "Flexor Digitorum Superficialis," *TeachMeAnatomy*. [Online]. Available: <https://teachmeanatomy.info/encyclopaedia/f/flexor-digitorum-superficialis/>.
- [23] F. Kerckhof, T. Van Leeuwen, and E. Vereecke, "The digital human forearm and hand," *J. Anat.*, vol. 233, no. 5, pp. 557–566, Nov. 2018, doi: 10.1111/joa.12877.
- [24] NumPy Developers, "NumPy," 2024. [Online]. Available: <https://numpy.org/>.
- [25] pandas Development Team, "pandas: Python Data Analysis Library," 2024. [Online]. Available: <https://pandas.pydata.org/>.

- [26] Matplotlib Development Team, "Matplotlib: Visualization with Python," 2024. [Online]. Available: <https://matplotlib.org/>.
- [27] F. Salvaire, "PySpice Documentation," 2024. [Online]. Available: <https://pyspice.fabrice-salvaire.fr>.
- [28] Electronics Tutorials, "Second Order Filters," *Electronics Tutorials*. [Online]. Available: <https://www.electronics-tutorials.ws/filter/second-order-filters.html>.
- [29] Changpuak.ch, "Active Notch Filter," *Changpuak.ch*. [Online]. Available: https://www.changpuak.ch/electronics/Active_Notch_Filter.php.
- [30] H. Zumbahlen, "Mini Tutorial MT-225: Twin-T Notch Filter," Analog Devices, Inc., Norwood, MA, USA, 2012. [Online]. Available: <https://www.analog.com/media/en/training-seminars/tutorials/MT-225.pdf>.
- [31] Texas Instruments, "INA128, INA129 Precision, Low Power Instrumentation Amplifiers," SBOS051B, Oct. 1995, revised Feb. 2005. [Online]. Available: <https://www.sigmaelectronica.net/manuals/ina128.pdf>.
- [32] Espressif Systems, "ESP32 Series Datasheet," v4.1, 2023. [Online]. Available: https://documentation.espressif.com/esp32_datasheet_en.pdf.
- [33] ElectronicWings, "Introduction to ESP32," *ElectronicWings*. [Online]. Available: <https://www.electronicwings.com/esp32/introduction-to-esp32>.
- [34] "Artificial neuron," in *Wikipedia*, 2026. [Online]. Available: https://en.wikipedia.org/wiki/Artificial_neuron.
- [35] B. Auffarth, *Machine Learning for Time-Series with Python*. Birmingham, UK: Packt Publishing, 2021.
- [36] A. L. Fradkov, "Early History of Machine Learning," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 1385–1390, 2020.
- [37] GeeksforGeeks, "CART (Classification and Regression Tree) in Machine Learning," *GeeksforGeeks*. [Online]. Available: <https://www.geeksforgeeks.org/machine-learning/cart-classification-and-regression-tree-in-machine-learning/>.
- [38] I. Guyon and A. Elisseeff, "An Introduction to Feature Extraction," in *Feature Extraction: Foundations and Applications*, I. Guyon, S. Gunn, M. Nikravesh, and L. A. Zadeh, Eds. Berlin, Germany: Springer, 2006, pp. 1–25.
- [39] C. Tepe and M. C. Demir, "Detection and Classification of Muscle Activation in EMG Data Acquired by Myo Armband," *Avrupa Bilim ve Teknoloji Dergisi*, no. Special Issue, pp. 178–183, Aug. 2020, doi: 10.31590/ejosat.779660.
- [40] M. Joseph, *Modern Time Series Forecasting with Python*. Birmingham, UK: Packt Publishing, 2022.
- [41] M. Martínez-Sellés and M. Marina-Breyse, "Current and Future Use of Artificial Intelligence in Electrocardiography," *J. Cardiovasc. Dev. Dis.*, vol. 10, no. 4, Art. no. 175, Apr. 2023, doi: 10.3390/jcdd10040175.

- [42] I. Fagioli, A. Mazzarini, F. Gennari, and S. Crea, "The Role of Artificial Intelligence and Machine Learning in Personalizing the Control of Robotic Lower Limb Prostheses," in *Enabling and Safeguarding Personalized Medicine* (Data Science, Machine Intelligence, and Law, vol. 7), F. Casarosa, F. Gennari, and A. Rossi, Eds. Cham, Switzerland: Springer Nature, 2025, pp. 243–266, doi: 10.1007/978-3-031-99709-9.
- [43] S. d'Ascoli, J. Rapin, Y. Benchetrit, T. Brooks, K. Begany, J. Raugel, H. Banville, and J.-R. King, "A foundation model of vision, audition, and language for in-silico neuroscience," *Meta AI*, Mar. 25, 2026. [Online]. Available: <https://ai.meta.com/research/publications/a-foundation-model-of-vision-audition-and-language-for-in-silico-neuroscience/>.
- [44] C. Attig, N. Rauh, T. Franke, and J. F. Krems, "System Latency Guidelines Then and Now — Is Zero Latency Really Considered Necessary?," in *Proceedings of the 9th Conference on Professional Knowledge Management (Wissensmanagement)*, 2017, pp. 242–245.
- [45] IBM, "What is Edge AI?," *IBM*. [Online]. Available: <https://www.ibm.com/think/topics/edge-ai>.
- [46] "Myo," *Wearables*. [Online]. Available: <https://wearables.com/products/myo>.
- [47] Nordic Semiconductor, "nRF52840," *Nordic Semiconductor*. [Online]. Available: <https://www.nordicsemi.com/Products/nRF52840>.
- [48] "Zigbee," in *Wikipedia*, May 5, 2026. [Online]. Available: <https://en.wikipedia.org/wiki/Zigbee>.

ΠΑΡΑΡΤΗΜΑ Α : ΚΩΔΙΚΑΣ ΠΡΟΣΟΜΟΙΩΣΗΣ

Σε αυτό το παράρτημα παρατίθεται ο κώδικας των αρχείων που χρησιμοποιήθηκαν για την υλοποίηση της αρχικής προσομοίωσης του συστήματος.

Αρχείο actionPotential.py:

```
import numpy as np

class ActionPotential:
    phi_array = np.array([])

    def __init__(self,t_vector,x0,y0,          yi,
zi,u,mu_sigma_L,mu_sigma_T,mu_anisotropy,z0=0,total_length=96,
firing_delay_s=0.002):
        self.t_vector = t_vector
        self.dt = self.t_vector[1] - self.t_vector[0]
        self.x0 = x0
        self.y0 = y0
        self.z0= z0

        # Fiber Properties
        self.yi = yi
        self.zi = zi
        self.u = u
        self.mu_anisotropy = mu_anisotropy
        self.sigma_T = mu_sigma_T
        self.sigma_L = mu_sigma_L
        self.total_length = total_length
        self.firing_delay_s = firing_delay_s

        self.x_ep = (total_length / 2) + np.random.uniform(-5, 5)
        self.L1 = self.x_ep
        self.L2 = total_length - self.x_ep

        self.phi_array = np.zeros_like(t_vector)
```

```

def vm(self,t):
    z = self.u * t * 1000
    return 96 * (z**3) * np.exp(-z) - 90

def impulse_response(self,t):
    vt = self.u * t * 1000

    pos1 = self.x_ep - vt
    pos2 = self.x_ep + vt

    c1 = (pos1 >= 0).astype(float)
    c2 = (pos2 <= (self.L1 + self.L2)).astype(float)

    radial_gap_sq = (self.y0 - self.yi)**2 + (self.z0 - self.zi)**2
    radial_term = self.mu_anisotropy * radial_gap_sq

    denom1 = np.sqrt((self.x0 - pos1)**2 + radial_term)
    denom2 = np.sqrt((self.x0 - pos2)**2 + radial_term)

    ir = (1 / (4 * np.pi * self.sigma_T)) * ((c1 / denom1) + (c2 / denom2))
    return np.nan_to_num(ir)

def phi(self):
    t_active = np.maximum(0, self.t_vector - self.firing_delay_s)

    vm_values = self.vm(t_active)
    dVm_dt = np.gradient(vm_values, self.dt)
    d2Vm_dt2 = np.gradient(dVm_dt, self.dt)

    d = 0.05
    cable_scale = (self.sigma_T * np.pi * (d**2)) / 4
    transmembrane_current = cable_scale * d2Vm_dt2

    ir_values = self.impulse_response(self.t_vector)

```

```

                                full_convolution      =
np.convolve(transmembrane_current,ir_values,mode='full') * self.dt
    convolution_values = full_convolution[:len(self.t_vector)]

    self.phi_array = (self.mu_anisotropy / self.u) * convolution_values

    return self.phi_array

```

Αρχείο motorUnitActionPotentials.py:

```

import numpy as np
import matplotlib.pyplot as plt
from actionPotential import ActionPotential

class MUAP:
    def __init__(self, t_vector, u_fiber_mean, x0, y0, z0, num_fibers = 400):
        self.t_vector = t_vector
        self.num_fibers = num_fibers
        self.muap_signal = np.zeros_like(t_vector)
        self.fibers = []

        mu_center_y = 10.0
        mu_center_z = 0.0
        mu_radius = 5.0

        for _ in range(num_fibers):
            u_fiber = np.random.normal(u_fiber_mean, 0.2)
            sigma_L_median_value = (0.33 + 0.80) / 2
            sigma_L_spread = (0.80 - sigma_L_median_value) / 3
            sigma_L = np.random.normal(sigma_L_median_value, sigma_L_spread)
            sigma_T_median_value = (0.04 + 0.15) / 2
            sigma_T_spread = (0.15 - sigma_T_median_value) / 3
            sigma_T = np.random.normal(sigma_T_median_value, sigma_T_spread)
            anisotropy = sigma_L/sigma_T

```

```

angle = np.random.uniform(0, 2 * np.pi)
r = mu_radius * np.sqrt(np.random.uniform(0, 1))
yi_pos = mu_center_y + r * np.cos(angle)
zi_pos = mu_center_z + r * np.sin(angle)

delay_s = np.random.normal(0.002, 0.0002)

fiber_length = np.random.uniform(40, 96)

fiber = ActionPotential(
    t_vector=t_vector,
    x0=x0, y0=y0, u=u_fiber,
    yi=yi_pos, zi=zi_pos,
    mu_sigma_L=sigma_L,
    mu_sigma_T=sigma_T,
    mu_anisotropy=anisotropy,
    z0=z0,
    total_length=fiber_length,
    firing_delay_s=delay_s
)
self.fibers.append(fiber)

def generate_signal(self):
    print(f"Calculating superposition for {self.num_fibers} fibers...")
    for fiber in self.fibers:
        fiber_signal = fiber.phi()
        self.muap_signal += fiber_signal

    return self.muap_signal

```

Αρχείο emgSignalSim.py

```

import numpy as np
import matplotlib.pyplot as plt

```

```

from motorUnitActionPotentials import MUAP

def create_sequence(t, fs):
    mvc_seq = np.zeros_like(t)

    def add_movement(start_t, duration, peak=50, steepness=100):
        start_idx = int(start_t * fs)
        end_idx = int((start_t + duration) * fs)
        if start_idx >= len(t): return

        chunk_t = t[start_idx:end_idx]

        onset = peak / (1 + np.exp(-steepness * (chunk_t - (start_t + 0.05))))
        release = (1 - 1 / (1 + np.exp(-steepness * (chunk_t - (start_t + duration
- 0.05)))))
        mvc_seq[start_idx:end_idx] = onset * release

    # --- DEFINE YOUR CHAIN HERE ---
    add_movement(start_t=0.4, duration=0.1, peak=70)
    add_movement(start_t=0.6, duration=0.1, peak=80)
    add_movement(start_t=0.9, duration=0.5, peak=60, steepness=5)
    add_movement(start_t=1.5, duration=0.4, peak=85)

    baseline_noise = np.random.normal(0, 0.2, len(t))
    mvc_seq += baseline_noise
    mvc_seq += np.random.normal(0, 1.2, len(t))
    return np.clip(mvc_seq, 0, 100)

def save_for_pyspice(t, signal, filename):
    """Exports time and voltage for PySpice (scales uV to Volts)."""
    signal_volts = signal * 1e-6
    data = np.column_stack((t, signal_volts))
    np.savetxt(filename, data, fmt='%1.6e %1.6e')
    print(f"File saved: {filename}")

```

```

def emg_signal_sim():
    fs = 10000
    duration = 2.0
    t = np.arange(0, duration, 1/fs)

    emg_engine = EMGSignal(t_vector=t, x0=10, y0=0, z0=0, num_units=200)

    mvc_profile = create_sequence(t, fs)

    biological_signal = emg_engine.simulate(mvc_profile)

    hum_50hz = 5.0 * np.sin(2 * np.pi * 50 * t)
    random_noise = np.random.normal(0, 1.5, len(t))

    dirty_pos = biological_signal + hum_50hz + random_noise
    dirty_neg = (np.roll(biological_signal, 20) * 0.95) + (hum_50hz * 0.85) +
random_noise

    save_for_pyspice(t, dirty_pos, "emg_positive_input.pwl")
    save_for_pyspice(t, dirty_neg, "emg_negative_input.pwl")

    plt.figure(figsize=(12, 6))
    plt.subplot(2, 1, 1)
    plt.plot(t, mvc_profile)
    plt.title("Squeeze Effort (% MVC)")
    plt.ylabel("Intensity")

    plt.subplot(2, 1, 2)
    plt.plot(t, dirty_pos, color='purple')
    plt.title("Synthetic FDS EMG Signal (Interference Pattern)")
    plt.xlabel("Time (s)")
    plt.ylabel("Potential (uV)")
    plt.tight_layout()
    plt.show()

```

```

class EMGSignal:
    def __init__(self, t_vector, x0=10, y0=0, z0=0, num_units=20):
        self.t_vector = t_vector
        self.dt = t_vector[1] - t_vector[0]
        self.num_units = num_units
        self.x0, self.y0, self.z0 = x0, y0, z0

        indices = np.arange(num_units)
        self.fiber_counts = (50 * np.exp(np.log(800/50) * indices / (num_units -
1))).astype(int)

        self.thresholds = 90 * (indices / (num_units - 1))**1.5

        self.muap_pool = []
        self._build_pool()

    def _build_pool(self):
        for i in range(self.num_units):
            print(f"Iteration: {i+1}/{self.num_units}")

            u_velocity = 3.0 + (2.0 * (i / self.num_units))

            unit = MUAP(
                t_vector=self.t_vector,
                u_fiber_mean=u_velocity,
                x0=self.x0, y0=self.y0, z0=self.z0,
                num_fibers=self.fiber_counts[i]
            )
            self.muap_pool.append(unit.generate_signal())

    def simulate(self, mvc_profile):
        full_signal = np.zeros_like(self.t_vector)
        num_samples = len(self.t_vector)

        for i, muap_signature in enumerate(self.muap_pool):

```

```

        threshold = self.thresholds[i]

        t_idx = 0
        while t_idx < num_samples:
            current_mvc = mvc_profile[t_idx]

            if current_mvc > threshold:
                hz = 8 + 0.4 * (current_mvc - threshold)
                hz = min(hz, 35)

                start_idx = t_idx
                end_idx = min(num_samples, start_idx + len(muap_signature))
                full_signal[start_idx:end_idx] += muap_signature[::(end_idx -
start_idx)]

                interval = int((1 / hz) / self.dt)
                jitter = int(np.random.normal(0, 0.1 * interval))
                t_idx += max(1, interval + jitter) # Move to next pulse
            else:
                t_idx += 1

        return full_signal

emg_signal_sim()

```

Αρχείο emgCircuitSim.py:

```

import matplotlib.pyplot as plt
import numpy as np
from PySpice.Spice.Netlist import Circuit
from PySpice.Unit import *
import math
from scipy.integrate import quad

circuit = Circuit('EMG Circuit')

```

```

circuit.include('INA128.LIB')
circuit.include('TL072.LIB')

positive_input_pwl_data = np.loadtxt('emg_positive_input.pwl')
positive_input_pwl_values = [tuple(row) for row in positive_input_pwl_data]

negative_input_pwl_data = np.loadtxt('emg_negative_input.pwl')
negative_input_pwl_values = [tuple(row) for row in negative_input_pwl_data]

# circuit.X(Name, Model, In+, In-, V+, V-, Out, Ref, RG1, RG2)
circuit.X('U1', 'INA128', 'ina_in_pos', 'ina_in_neg', 'ina_vcc', 'ina_vee',
'ina_out', circuit.gnd, 'ina_gain_node_1', 'ina_gain_node_2')

# circuit.X(Name, Model, In+, In-, V+, V-, Out)
circuit.X('U2', 'TL072', 'tl_in_pos', 'tl_in_neg', 'tl_vcc', 'tl_vee', 'tl_out')
circuit.X('U3', 'TL072',
'notch_in_pos', 'notch_out', 'tl_vcc', 'tl_vee', 'notch_out')
circuit.X('U4', 'TL072',
'notch_feedback_in_pos', 'notch_feedback_out', 'tl_vcc', 'tl_vee', 'notch_feedback_out')

# Grounding the subject
circuit.SinusoidalVoltageSource('V_BONE', 'electrode', circuit.gnd,
amplitude=0.05@u_V, frequency=50@u_Hz) # type: ignore

# First Stage INA128 instrumentation amplifier

# Gain Resistor
circuit.R('RG', 'ina_gain_node_1', 'ina_gain_node_2', 505@u_0hm) # type: ignore

# IC Supply
circuit.V('INA_VCC', 'ina_vcc', circuit.gnd, 10@u_V) # type: ignore
circuit.V('INA_VEE', 'ina_vee', circuit.gnd, -10@u_V) # type: ignore
circuit.V('TL_VCC', 'tl_vcc', circuit.gnd, 10@u_V) # type: ignore
circuit.V('TL_VEE', 'tl_vee', circuit.gnd, -10@u_V) # type: ignore

```

```

# IC Input
circuit.PieceWiseLinearVoltageSource('ina_vminus', 'ina_in_neg', circuit.gnd,
values=negative_input_pwl_values)
circuit.PieceWiseLinearVoltageSource('ina_vplus', 'ina_in_pos', 'electrode',
values=positive_input_pwl_values)

# Notch Filter
circuit.C('notch_c1','ina_out','c_middle',0.1@u_uF) # type: ignore
circuit.C('notch_c2','c_middle','notch_network_out',0.1@u_uF) # type: ignore
circuit.R('notch_r1','c_middle',circuit.gnd,15.91@u_kOhm) # type: ignore

circuit.R('notch_r2','ina_out','r_middle',31.83@u_kOhm) # type: ignore
circuit.R('notch_r3','r_middle','notch_network_out',31.83@u_kOhm) # type: ignore
circuit.C('notch_c3','r_middle',circuit.gnd,0.2@u_uF) # type: ignore

circuit.R('tie_notch_to_amp','notch_network_out','notch_in_pos',1@u_0hm) # type:
ignore

# Notch Feedback
circuit.R('notch_feedback_r1','notch_out','notch_feedback_middle',10@u_kOhm) #
type: ignore
circuit.R('notch_feedback_r2','notch_feedback_middle',circuit.gnd,90@u_kOhm) #
type: ignore

# High Pass Filter
circuit.C('high_pass_c', 'notch_out', 'tl_in_pos', 0.8@u_uF) # type: ignore
circuit.R('high_pass_r','tl_in_pos',circuit.gnd,10@u_kOhm) # type: ignore
circuit.R('tl_rminus','tl_in_neg',circuit.gnd, 1@u_kOhm)# type: ignore

# TL072 Output
circuit.R('RF','tl_out','tl_in_neg', 10@u_kOhm)# type: ignore

# Low Pass Filter
circuit.R('low_pass_r','tl_out','circuit_out',10@u_kOhm) # type: ignore

```

```

circuit.C('low_pass_c','circuit_out',circuit.gnd,0.031@u_uF) # type: ignore

simulator = circuit.simulator(temperature=25, nominal_temperature=25)
simulator.options(
    reltol=0.01,
    gmin=1e-10,
    chgtol=1e-13,
    itl1=1000,
    itl2=1000
)

try:
    analysis = simulator.transient(step_time=10@u_us, end_time=2@u_s) # type:
ignore

# 2. Updated DFT Function for Isolation
def get_dft(node_name):
    voltages = np.array(analysis[node_name])
    N = len(voltages)
    fft_values = np.fft.rfft(voltages)
    freqs = np.fft.rfftfreq(N, d=10e-6)
    mags = np.abs(fft_values) * (2 / N)
    return freqs, mags

f_pre, m_pre = get_dft('ina_out') # Input to Notch
f_post, m_post = get_dft('notch_out') # Output of Notch Buffer
f_final, m_final = get_dft('circuit_out') # Final System Output

target_hz = 50.0
idx_50 = np.argmin(np.abs(f_pre - target_hz))

hum_before = 20 * np.log10(m_pre[idx_50] + 1e-9)
hum_after = 20 * np.log10(m_post[idx_50] + 1e-9)
rejection = hum_before - hum_after

```

```

print(f"\n--- TWIN-T NOTCH VERIFICATION (STAGE ISOLATION) ---")
print(f"Target Frequency:      {target_hz} Hz")
print(f"Magnitude Pre-Notch:    {hum_before:.2f} dBV")
print(f"Magnitude Post-Notch:    {hum_after:.2f} dBV")
print(f"Isolated Notch Depth:     {rejection:.2f} dB")

# --- PLOTS ---
fig, (ax1, ax2) = plt.subplots(2, 1, figsize=(12, 10))

    ax1.plot(analysis.time, analysis['ina_out'], color='gray', alpha=0.4,
label='Raw INA Out (With Hum)')

    ax1.plot(analysis.time, analysis['notch_out'], color='blue', label='Post-
Notch (Cleaned)')

ax1.set_xlim(0.5, 0.6)
ax1.set_title('Time Domain: 50Hz Hum Removal (Zoomed View)')
ax1.set_xlabel('Time (s)')
ax1.set_ylabel('Voltage (V)')
ax1.legend()
ax1.grid(True)

    ax2.plot(f_pre, 20*np.log10(m_pre + 1e-9), color='gray', alpha=0.3,
label='Input to Notch')

    ax2.plot(f_post, 20*np.log10(m_post + 1e-9), color='blue', linewidth=1.5,
label='Notch Output (Isolated)')

    ax2.plot(f_final, 20*np.log10(m_final + 1e-9), color='red', linewidth=2,
label='Total System Output')

f_low = 19.9
f_high = 513.4
    ax2.axvspan(f_low, f_high, color='green', alpha=0.1, label='EMG Passband
(20Hz-500Hz)')

ax2.set_xlim(5, 600)
ax2.set_ylim(-100, 20) # Elevated ceiling to accommodate the final 11x gain
    ax2.axvline(x=target_hz, color='purple', linestyle='--', alpha=0.7,
label=f'Notch @ {target_hz}Hz')

```

```
ax2.set_title('Frequency Response: Full Chain Bandwidth & Notch Performance')
ax2.set_xlabel('Frequency (Hz)')
ax2.set_ylabel('Magnitude (dBV)')
ax2.legend(loc='upper right')
ax2.grid(True, which='both')

plt.tight_layout()
plt.show()

except Exception as e:
    print(f"Simulation failed: {e}")
```

ΠΑΡΑΡΤΗΜΑ Β : ΚΩΔΙΚΑΣ ΕΠΕΞΕΡΓΑΣΙΑΣ ΔΕΔΟΜΕΝΩΝ

Σε αυτό το παράρτημα παρατίθεται ο κώδικας που χρησιμοποιήθηκε για την επεξεργασία των δεδομένων και την αξιολόγηση των χαρακτηριστικών.

Αρχείο features.py:

```
import pandas as pd
import numpy as np

class Features:
    voltage_files = {'noise': [], 'fist': [], 'index': [], 'middle': [], 'ring':
[], 'little': []}
    offset = 0
    recordings = np.array(['noise', 'fist', 'index', 'middle', 'ring', 'little'])
    window_size = 1000 # in ms
    step_size = 200 # in ms

    mav_df = pd.DataFrame()
    rms_df = pd.DataFrame()
    wl_df = pd.DataFrame()
    zc_df = pd.DataFrame()
    ssc_df = pd.DataFrame()
    dynamic_fft_df = pd.DataFrame()

    master_df = pd.DataFrame()

    def __init__(self):
        self.offset = self.getCenterOfSignal()
        self.GetFiles()
        print('Caclulating MAV...')
        self.getMeanAbsoluteValue()
        print('Caclulating RMS...')
        self.getRootMeanSquared()
        print('Caclulating WL...')
        self.getWaveformLength()
        print('Caclulating ZC...')
```

```

self.getZeroCrossings()
print('Caclulating SSC...')
self.getSlopeSignChange()
print('Caclulating FFT features...')
self.getDynamicFFTFeatures()

self.master_df = pd.concat([
    self.mav_df,
    self.rms_df['RMS'],
    self.wl_df['WL'],
    self.zc_df['ZC'],
    self.ssc_df['SSC'],
    self.dynamic_fft_df[['Peak_FFT', 'Mean_FFT']]
], axis=1)
self.master_df.to_csv('final_dataframe_with_fft_second.csv')

def getCenterOfSignal(self):
    baseline_file = pd.read_csv('../TRAINING DATA/VOLTAGE_SECOND/noise/1.csv',
header=1)
    offset = baseline_file['(V)'].mean()
    return offset

def getFiles(self):
    for recording in self.recordings:
        print(f'Loading /{recording} data....')
        for i in range(1,21):
            voltage_file_path = f"../TRAINING
DATA/VOLTAGE_SECOND/{recording}/{i}.csv"
            self.voltage_files[recording].append(pd.read_csv(voltage_file_pa
th,header=1))

def getMeanAbsoluteValue(self):
    rows = []
    for label, file_list in self.voltage_files.items():

```

```

        for file in file_list:
            centeredSignal = file['(V)'] - self.offset
            for i in range(0, len(centeredSignal) - self.window_size,
self.step_size):
                window = centeredSignal[i : i + self.window_size].values
                mav = np.mean(np.abs(window))
                start_ms = i / 10
                end_ms = (i + self.window_size) / 10
                time_label = f"{start_ms}ms-{end_ms}ms"
                rows.append({'Label': label, 'Time_Range': time_label, 'MAV':
mav})
            self.mav_df = pd.DataFrame(rows)

    def getRootMeanSquared(self):
        rows = []
        for label, file_list in self.voltage_files.items():
            for file in file_list:
                centeredSignal = file['(V)'] - self.offset
                for i in range(0, len(centeredSignal) - self.window_size,
self.step_size):
                    window = centeredSignal[i : i + self.window_size].values
                    rms = np.sqrt(np.mean(window**2))
                    start_ms = i / 10
                    end_ms = (i + self.window_size) / 10
                    time_label = f"{start_ms}ms-{end_ms}ms"
                    rows.append({'Label': label, 'Time_Range': time_label, 'RMS':
rms})
            self.rms_df = pd.DataFrame(rows)

    def getWaveformLength(self):
        rows = []
        for label, file_list in self.voltage_files.items():
            for file in file_list:
                centeredSignal = file['(V)'] - self.offset
                for i in range(0, len(centeredSignal) - self.window_size,
self.step_size):

```

```

        window = centeredSignal[i : i + self.window_size].values
        wl = np.sum(np.abs(np.diff(window)))
        start_ms = i / 10
        end_ms = (i + self.window_size) / 10
        time_label = f"{start_ms}ms-{end_ms}ms"
        rows.append({'Label': label, 'Time_Range': time_label, 'WL':
wl})

    self.wl_df = pd.DataFrame(rows)

def getZeroCrossings(self):
    rows = []
    for label, file_list in self.voltage_files.items():
        for file in file_list:
            centeredSignal = file['(V)'] - self.offset
            for i in range(0, len(centeredSignal) - self.window_size,
self.step_size):
                window = centeredSignal[i : i + self.window_size].values
                zc = ((window[:-1] * window[1:]) < 0).sum()
                start_ms = i / 10
                end_ms = (i + self.window_size) / 10
                time_label = f"{start_ms}ms-{end_ms}ms"
                rows.append({'Label': label, 'Time_Range': time_label, 'ZC':
zc})

    self.zc_df = pd.DataFrame(rows)

def getSlopeSignChange(self):
    rows = []
    for label, file_list in self.voltage_files.items():
        for file in file_list:
            centeredSignal = file['(V)'] - self.offset
            for i in range(0, len(centeredSignal) - self.window_size,
self.step_size):
                window = centeredSignal[i : i + self.window_size].values
                slopes = np.diff(window)
                ssc = ((slopes[:-1] * slopes[1:]) < 0).sum()
                start_ms = i / 10

```

```

        end_ms = (i + self.window_size) / 10
        time_label = f"{start_ms}ms-{end_ms}ms"
        rows.append({'Label': label, 'Time_Range': time_label, 'SSC':
ssc})

    self.ssc_df = pd.DataFrame(rows)

def getDynamicFFTFeatures(self):
    rows = []
    for label, file_list in self.voltage_files.items():
        for file in file_list:
            centeredSignal = file['(V)'] - self.offset
            for i in range(0, len(centeredSignal) - self.window_size,
self.step_size):
                window = centeredSignal[i : i + self.window_size].values

                # 1. Compute FFT dynamically for this exact window
                fft_values = np.abs(np.fft.rfft(window))

                # 2. Extract frequency features (Peak and Mean)
                peak_fft = np.max(fft_values)
                mean_fft = np.mean(fft_values)

                # 3. Save matching time labels
                start_ms = i / 10
                end_ms = (i + self.window_size) / 10
                time_label = f"{start_ms}ms-{end_ms}ms"
                rows.append({'Label': label, 'Time_Range': time_label,
'Peak_FFT': peak_fft, 'Mean_FFT': mean_fft})

    self.dynamic_fft_df = pd.DataFrame(rows)

features = Features()

```

Αρχείο correlation.py:

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

df = pd.read_csv('final_dataframe_with_fft_second.csv')

features_df = df[['MAV', 'RMS', 'WL', 'ZC', 'SSC', 'Peak_FFT', 'Mean_FFT']]

corr_matrix = features_df.corr()

print(corr_matrix)

plt.figure(figsize=(10, 8))
sns.heatmap(corr_matrix, annot=True, cmap='coolwarm', fmt=".2f")
plt.title('EMG Feature Correlation Matrix')
plt.show()
```

ΠΑΡΑΡΤΗΜΑ C : ΚΩΔΙΚΑΣ ΜΗΧΑΝΙΚΗΣ ΜΑΘΗΣΗΣ

Σε αυτό το παράρτημα παρατίθεται ο κώδικας των αρχείων που χρησιμοποιήθηκαν για την εκπαίδευση του μοντέλου μηχανικής μάθησης και την εφαρμογή στον μικροελεγκτή ESP32.

Αρχείο ml_model.py:

```
import pandas as pd
import xgboost as xgb
import treelite
import t12cgen
from sklearn.model_selection import train_test_split, RandomizedSearchCV
from sklearn.preprocessing import LabelEncoder
from sklearn.metrics import accuracy_score, classification_report
import warnings

warnings.filterwarnings('ignore')

# 1. Load Data
data = pd.read_csv('final_dataframe_with_fft_second.csv')
data = data.dropna(subset=['Label'])

# 2. Define Features
X = data[['MAV', 'RMS', 'WL', 'ZC', 'SSC', 'Peak_FFT', 'Mean_FFT']]
y = data['Label']

# 3. Encode Labels
label_encoder = LabelEncoder()
y_encoded = label_encoder.fit_transform(y)

# 4. Train/Test Split
X_train, X_test, y_train, y_test = train_test_split(
    X, y_encoded, test_size=0.2, random_state=42, stratify=y_encoded
)

# 5. Define the XGBoost Model
best_model = xgb.XGBClassifier()
```

```

    max_depth=4,
    n_estimators=50,
    learning_rate=0.1,
)

print("Training the tiny model...")
best_model.fit(X_train, y_train)

# 6. Evaluate the Model
y_pred = best_model.predict(X_test)

y_test_labels = label_encoder.inverse_transform(y_test)
y_pred_labels = label_encoder.inverse_transform(y_pred)

accuracy = accuracy_score(y_test_labels, y_pred_labels)
classification_rep = classification_report(y_test_labels, y_pred_labels)

print(f"\nOverall Accuracy: {accuracy*100:.2f}%")
print("\nClassification Report:\n", classification_rep)

# 7. Test One Sample from Each Class
print("\n" + "="*50)
print("TESTING ONE SAMPLE FROM EACH CLASS")
print("="*50)

unique_classes = label_encoder.classes_

for gesture in unique_classes:
    gesture_encoded = label_encoder.transform([gesture])[0]
    sample_index = (y_test == gesture_encoded).argmax()

    sample = X_test.iloc[[sample_index]]

    prediction_encoded = best_model.predict(sample)

```

```

prediction_label = label_encoder.inverse_transform(prediction_encoded)[0]

is_correct = "✅ CORRECT" if gesture.upper() == prediction_label.upper() else
"❌ INCORRECT"

print(f"\nActual Gesture:    {gesture.upper()}")
print(f"Predicted Gesture: {prediction_label.upper()} {is_correct}")

# Print the sensor values rounded for readability
sample_dict = sample.iloc[0].round(4).to_dict()
print(f"Sensor Data:        {sample_dict}")

print("\nConverting XGBoost model to Treelite format...")
tl_model = treelite.frontend.from_xgboost(best_model.get_booster())

print("Compiling into C code...")
tl2cgen.generate_c_code(tl_model,          dirpath="./gesture_model_c_treelite",
params={"parallel_comp": 5})

print("Treelite/TL2cgen Export Complete!")

```

Αρχείο model.c:

```

#include "header.h"

static const int32_t num_class[] = { 6, };

int32_t get_num_target(void) {
    return N_TARGET;
}

void get_num_class(int32_t* out) {
    for (int i = 0; i < N_TARGET; ++i) {
        out[i] = num_class[i];
    }
}

```

```

int32_t get_num_feature(void) {
    return 7;
}
const char* get_threshold_type(void) {
    return "float32";
}
const char* get_leaf_output_type(void) {
    return "float32";
}

void predict(union Entry* data, int pred_margin, float* result) {
    unsigned int tmp;
    predict_unit0(data, result);
    predict_unit1(data, result);
    predict_unit2(data, result);
    predict_unit3(data, result);
    predict_unit4(data, result);

    // Apply base_scores
    result[0] += -6.29425048828125e-05;
    result[1] += 6.306171417236328125e-05;
    result[2] += 0.0003150701522827148438;
    result[3] += -6.29425048828125e-05;
    result[4] += 6.306171417236328125e-05;
    result[5] += -0.0003150701522827148438;

    // Apply postprocessor
    if (!pred_margin) { postprocess(result); }
}

// Apply postprocessor for a single target
static void postprocess_impl(float* target_result, int num_class) {
    float max_margin = target_result[0];
    double norm_const = 0.0;

```

```

float t;
for (int k = 1; k < num_class; ++k) {
    if (target_result[k] > max_margin) {
        max_margin = target_result[k];
    }
}
for (int k = 0; k < num_class; ++k) {
    t = expf(target_result[k] - max_margin);
    norm_const += t;
    target_result[k] = t;
}
for (int k = 0; k < num_class; ++k) {
    target_result[k] /= (float)norm_const;
}
}

void postprocess(float* result) {
    // softmax
    postprocess_impl(&result[0], 6);
}

```

Αρχείο header.h:

```

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <float.h>
#include <math.h>
#include <stdint.h>

#if defined(__clang__) || defined(__GNUC__)
#define LIKELY(x)    __builtin_expect(!!(x), 1)
#define UNLIKELY(x) __builtin_expect(!!(x), 0)
#else

```

```

#define LIKELY(x)    (x)
#define UNLIKELY(x) (x)
#endif

#define N_TARGET 1
#define MAX_N_CLASS 6

union Entry {
    int missing;
    float fvalue;
    int qvalue;
};

int32_t get_num_target(void);
void get_num_class(int32_t* out);
int32_t get_num_feature(void);
const char* get_threshold_type(void);
const char* get_leaf_output_type(void);
void predict(union Entry* data, int pred_margin, float* result);
void postprocess(float* result);
void predict_unit0(union Entry* data, float* result);
void predict_unit1(union Entry* data, float* result);
void predict_unit2(union Entry* data, float* result);
void predict_unit3(union Entry* data, float* result);
void predict_unit4(union Entry* data, float* result);

```

Αρχείο esp_emg.ino:

```

#include "arduinoFFT.h"

#define EMG_PIN 34
#define SAMPLE_RATE_HZ 1000
#define TIMER_INTERVAL_US (1000000 / SAMPLE_RATE_HZ)

#define WINDOW_SIZE 1000

```

```

#define STEP_SIZE 200

#define FFT_SAMPLES 1024

extern "C" {
    #include "header.h"
}

const char* gesture_names[] = {"FIST", "INDEX", "LITTLE", "MIDDLE", "NOISE",
"RING"};

volatile double emgBuffer[WINDOW_SIZE];
volatile int bufferIndex = 0;
volatile bool windowReady = false;
hw_timer_t * timer = NULL;

double localWindow[WINDOW_SIZE];

double vReal[FFT_SAMPLES];
double vImag[FFT_SAMPLES];
ArduinoFFT<double> FFT = ArduinoFFT<double>(vReal, vImag, FFT_SAMPLES,
SAMPLE_RATE_HZ);

int last_predicted_class = -1;

void IRAM_ATTR onTimer() {
    if (!windowReady) {
        int rawValue = analogRead(EMG_PIN);

        double voltage = (rawValue / 4095.0) * 3.3;

        emgBuffer[bufferIndex] = voltage;
        bufferIndex++;

        if (bufferIndex >= WINDOW_SIZE) {
            windowReady = true;
        }
    }
}

```

```

    }
  }
}

void setup() {
  Serial.begin(115200);
  analogSetAttenuation(ADC_11db);
  analogReadResolution(12);

  Serial.println("Starting EMG ML Pipeline...");
  Serial.println("Waiting for first gesture change...");
  Serial.println("-----");

  timer = timerBegin(1000000);
  timerAttachInterrupt(timer, &onTimer);
  timerAlarm(timer, TIMER_INTERVAL_US, true, 0);
}

void loop() {
  if (windowReady) {
    noInterrupts();
    memcpy(localWindow, (void*)emgBuffer, sizeof(localWindow));

    int remaining = WINDOW_SIZE - STEP_SIZE;
    for(int i = 0; i < remaining; i++) {
      emgBuffer[i] = emgBuffer[i + STEP_SIZE];
    }
    bufferIndex = remaining;
    windowReady = false;
    interrupts();

    double sum = 0;
    for (int i = 0; i < WINDOW_SIZE; i++) sum += localWindow[i];
    double offset = sum / WINDOW_SIZE;

```

```

for (int i = 0; i < WINDOW_SIZE; i++) {
    localWindow[i] = localWindow[i] - offset;
}

double MAV = 0, RMS = 0, WL = 0;
int ZC = 0, SSC = 0;
double sum_sq = 0;

for (int i = 0; i < WINDOW_SIZE; i++) {
    double val = localWindow[i];
    MAV += abs(val);
    sum_sq += (val * val);

    if (i > 0) {
        WL += abs(val - localWindow[i - 1]);

        if ((localWindow[i - 1] * val) < 0) {
            ZC++;
        }
    }

    if (i > 1) {
        double slope1 = localWindow[i - 1] - localWindow[i - 2];
        double slope2 = val - localWindow[i - 1];
        if ((slope1 * slope2) < 0) {
            SSC++;
        }
    }
}

MAV = MAV / WINDOW_SIZE;
RMS = sqrt(sum_sq / WINDOW_SIZE);

```

```

for (int i = 0; i < FFT_SAMPLES; i++) {
    if (i < WINDOW_SIZE) {
        vReal[i] = localWindow[i];
    } else {
        vReal[i] = 0.0;
    }
    vImag[i] = 0.0;
}

FFT.windowing(FFTWindow::Hamming, FFTDirection::Forward);
FFT.compute(FFTDirection::Forward);
FFT.complexToMagnitude();

double Peak_FFT = 0;
double sum_fft = 0;
int usable_bins = FFT_SAMPLES / 2;

for (int i = 0; i < usable_bins; i++) {
    sum_fft += vReal[i];
    if (vReal[i] > Peak_FFT) {
        Peak_FFT = vReal[i];
    }
}

double Mean_FFT = sum_fft / usable_bins;

union Entry t1_features[7];

t1_features[0].fvalue = (float)MAV;
t1_features[1].fvalue = (float)RMS;
t1_features[2].fvalue = (float)WL;
t1_features[3].fvalue = (float)ZC;
t1_features[4].fvalue = (float)SSC;
t1_features[5].fvalue = (float)Peak_FFT;
t1_features[6].fvalue = (float)Mean_FFT;

```

```
float output_probs[6] = {0.0};

predict(tl_features, 0, output_probs);

int predicted_class = 0;
float max_prob = output_probs[0];
for(int i = 1; i < 6; i++) {
    if(output_probs[i] > max_prob) {
        max_prob = output_probs[i];
        predicted_class = i;
    }
}

if (predicted_class != last_predicted_class) {
    Serial.print("Gesture Changed To: ");
    Serial.print(gesture_names[predicted_class]);
    Serial.print(" | Confidence: ");
    Serial.print(max_prob * 100);
    Serial.println("%");

    last_predicted_class = predicted_class;
}
}
```