



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΤΜΗΜΑ ΜΗΧΑΝΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

**Έλεγχος Εξωσκελετικής Πειραματικής Διάταξης Ενός Βαθμού
Ελευθερίας για Ενίσχυση του Άνω Άκρου με Χρήση
Ηλεκτρομυογραφικών Σημάτων**

Χαριτούδη Καλλιόπη

Επιβλέπων Καθηγητής:

Κωνσταντίνος Κυριακόπουλος

ΑΘΗΝΑ 2010

ΠΕΡΙΕΧΟΜΕΝΑ

Περιεχόμενα.....	1
Κεφάλαιο 1^ο – Εισαγωγή	
1.1 Αντικείμενο της Εργασίας.....	3
1.2 Βιβλιογραφική Ανασκόπηση.....	4
1.3 Διάρθρωση της Εργασίας.....	6
Κεφάλαιο 2^ο – Αρχιτεκτονική Συστήματος	
2.1 Υποσύστημα Ηλεκτρομυογραφικών Σημάτων.....	7
2.1.1 Λειτουργικό Σύστημα Linux.....	8
2.1.2 Βιοενισχυτής.....	8
2.1.3 Ενεργά Ηλεκτρόδια.....	8
2.2 Υποσύστημα Εξωσκελετικής Διάταξης.....	9
2.2.1 Λειτουργικό Σύστημα QNX.....	9
2.2.2 Κάρτα ServoToGo.....	9
2.2.3 Εξωσκελετική Διάταξη.....	10
2.2.4 Κλωβός μείωσης ηλεκτρομαγνητικής ακτινοβολίας.....	11
Κεφάλαιο 3^ο – Ανάλυση Σημάτων	
3.1 Δειγματοληψία.....	13
3.2 Επεξεργασία Σημάτων.....	13
3.2.1 Θεωρητική Προσέγγιση.....	14
3.2.2 Επιλογή Φίλτρων.....	16
3.3 Αναγνώριση Μυϊκής Έντασης.....	24
Κεφάλαιο 4^ο – Δυναμική Ανάλυση Συστήματος	
4.1 Μοντελοποίηση Ανθρώπινου Χεριού.....	26

4.2	Μοντελοποίηση Εξωσκελετικής Διάταξης.....	27
------------	--	-----------

Κεφάλαιο 5^ο – Αλγόριθμοι Ελέγχου

5.1	Ελεγκτής Υποχωρητικότητας.....	29
5.2	Ελεγκτής Ενίσχυσης.....	31
5.2.1	Επιλογή Κερδών με Προσομοίωση.....	32
5.2.2	Πειραματική υλοποίηση.....	34

Κεφάλαιο 6^ο – Αξιολόγηση Αποτελεσμάτων & Μελλοντικές Κατευθύνσεις

6.1	Αξιολόγηση Αποτελεσμάτων.....	37
6.2	Μελλοντικές Κατευθύνσεις.....	38

ΠΑΡΑΡΤΗΜΑ Α:	Οδηγίες Χρήσης.....	40
---------------------	----------------------------	-----------

ΠΑΡΑΡΤΗΜΑ Β:	Επεξήγηση Προγραμμάτων.....	46
---------------------	------------------------------------	-----------

Βιβλιογραφία.....	86
--------------------------	-----------

ΚΕΦΑΛΑΙΟ 1^ο – ΕΙΣΑΓΩΓΗ

1.1 Αντικείμενο της Εργασίας

Η ραγδαία εξέλιξη των ηλεκτρομηχανολογικών συστημάτων που παρατηρείται τις τελευταίες δεκαετίες έχει οδηγήσει στην εισχώρηση της τεχνολογίας σε τομείς που παλαιότερα δεν θα τολμούσαμε να φανταστούμε. Ένας τέτοιος τομέας, ο οποίος έχει προσφέρει δόκιμο πεδίο για πρωτοποριακές τεχνολογικές εφαρμογές είναι και η εμβιομηχανική. Λόγω της κρισιμότητας των προβλημάτων που έχουν να κάνουν με το ανθρώπινο σώμα και την ανθρώπινη υγεία, πλήθος ερευνητών έχει αφιερωθεί στην εξεύρεση καινοτόμων λύσεων στον τομέα αυτό.

Στα πλαίσια της φιλοσοφίας αυτής, η παρούσα μελέτη - διπλωματική εργασία πραγματεύεται την ανάπτυξη ενός αρχικού πειραματικού μοντέλου που θα χρησιμοποιηθεί για τη διερεύνηση των δυνατοτήτων ελέγχου μιας εξωσκελετικής διάταξης (στο εξής θα αναφέρεται και ως exoskeleton) με τη χρήση ηλεκτρομυογραφικών σημάτων (EMGs).

Η συγκεκριμένη διάταξη ενός βαθμού ελευθερίας έχει κατασκευαστεί να προσαρμόζεται στον αγκώνα του ανθρώπινου χεριού με σκοπό να ενισχύει το έργο που έχει τη δυνατότητα να παράγει ένα ανθρώπινο χέρι από τη φύση του. Για να λειτουργήσει μια τέτοια διάταξη σε αρμονία με τον χρήστη, εκτός από το αμιγώς μηχανικό σκέλος χρειάζεται και το αντίστοιχο υπολογιστικό κομμάτι το οποίο θα αναλάβει τον ηλεκτρονικό έλεγχο βασισμένο σε κατάλληλα σήματα. Τα σήματα αυτά πρέπει ανά πάσα στιγμή να είναι ικανά να δώσουν τόσο πληροφορίες για την κατάσταση του χεριού όσο και εντολές για την επιθυμία κίνησης. Στην περίπτωση μας τα σήματα που θα χρησιμοποιηθούν αντλούνται από τον μυ του δικεφάλου που είναι ο κύριος μυς ο οποίος ελέγχει την κίνηση της άρθρωσης του ανθρώπινου αγκώνα. Έτσι έχουμε στη διάθεση μας σε πραγματικό χρόνο μετρήσιμα στοιχεία της μυϊκής έντασης του χρήστη και μπορούμε να οδηγήσουμε κατάλληλα τη διάταξη μας ώστε να αναλάβει το μηχανικό της μέρος, ποσοστό από το φορτίο που ο χρήστης προσπαθεί να κινήσει.

Σε όλη αυτή τη διαδικασία είναι ζωτικής σημασίας για μελλοντικές πραγματικές εφαρμογές, ο χρήστης να αποφορτίζεται από το επιπλέον βάρος, χωρίς όμως να νοιώθει ότι η κίνηση του οδηγείται ή παρεμποδίζεται από το exoskeleton. Πρέπει δηλαδή ο μηχανισμός να είναι σε θέση να ακολουθεί πιστά την κίνηση του χεριού και να αναλαμβάνει το επιπλέον φορτίο. Εν συντομία θα πρέπει το σύστημα μας να παρέχει «υποχωρητικότητα» για να μην παρεμποδίζει τον χρήστη, προσφέροντας ταυτόχρονα «ενίσχυση» ώστε να αναλαμβάνει εξολοκλήρου το βάρος.

1.2 Βιβλιογραφική Ανασκόπηση

Όπως προαναφέραμε οι τεράστιες δυνατότητες που προσφέρουν οι εξωσκελετικές εφαρμογές έχουν προσελκύσει μεγάλη ερευνητική δραστηριότητα στο συγκεκριμένο πεδίο. Στο σημείο αυτό κρίνεται σκόπιμη μια ανασκόπηση της υπάρχουσας αρθρογραφίας έτσι ώστε να αποκτήσουμε μια εικόνα των όσων έχουν γίνει στο τομέα, και να αντλήσουμε χρήσιμη γνώση γύρω από τα θέματα που θα μας απασχολήσουν.

Σημαντικός αριθμός ερευνητών έχουν εστιάσει στην κατασκευή διατάξεων εξωσκελετικής υποστήριξης με στόχο την αποκατάσταση των βλαβών των άκρων. Η εργασία [1] αποτελεί ένα τέτοιο παράδειγμα. Ασχολείται με την κατασκευή μίας διάταξης η οποία με χρήση EMGs θα βοηθάει ασθενείς, με προοπτική αποκατάστασης των βλαβών που έχουν υποστεί, τότε υποβοηθώντας και τότε εμποδίζοντας τους στην κίνηση –ανάλογα με τις ανάγκες.

Σε κάποιες περιπτώσεις – όπως στην εργασία [4] – ο έλεγχος του εξωσκελετικού μηχανισμού γίνεται με το συνδυασμό χρήσης ηλεκτρομυογραφικών σημάτων και χρήση αισθητήρων δύναμης στην προκειμένη περίπτωση προσαρμοσμένους στον καρπό του χρήστη. Έχοντας αυτά σαν εισόδους του αλγορίθμου και με παράλληλη χρήση τεχνικών fuzzy control ελέγχει τον εξωσκελετικό μηχανισμό.

Μία άλλη εργασία που μελετάει έναν ελεγκτή για την υποστήριξη της άρθρωσης του αγκώνα είναι η εργασία [7]. Σε αυτή την περίπτωση χρησιμοποιούνται επίσης EMGs, τα οποία σε συνδυασμό με την κινηματική

ανάλυση της συγκεκριμένης άρθρωσης εισάγονται σε ένα μοντέλο του μυός που είναι υπεύθυνος για την κίνηση. Το μοντέλο αυτό προβλέπει την ροπή που δημιουργείται στην άρθρωση του αγκώνα, τη συγκρίνει με την πραγματική –μέσω αισθητήρα δύναμης-ροπής– και επενεργεί κατάλληλα στον εξωσκελετικό μηχανισμό.

Αξίζει να δούμε και μία εξωσκελετική κατασκευή για την ενίσχυση της άρθρωσής του ώμου, όπως αυτή της εργασίας [5]. Η δυσκολία στην περίπτωση αυτή αυξάνεται σημαντικά λόγω της πολυπλοκότητας της συγκεκριμένης άρθρωσης. Κατά την κίνηση της άρθρωσής αυτής, ακόμα και για πολύ απλές κινήσεις ενεργοποιούνται πολλοί μύες, ενώ παράλληλα είναι δύσκολο να λάβουμε το ίδιο ηλεκτρομυογραφικό σήμα για ένα είδος κίνησης, ακόμα και από τον ίδιο ασθενή. Στην εργασία αυτή περιγράφεται ένας ελεγκτής που χρησιμοποιώντας fuzzy-neuro control προσαρμόζεται ώστε να αντιμετωπιστούν τα πιο πάνω προβλήματα και μελετάται η δυνατότητα υποστήριξης του ώμου για διαφορετικές στάσεις του χεριού.

1.3 Διάρθρωση Εργασίας

Η παρούσα διπλωματική εργασία αποτελείται από τα ακόλουθα βασικά τμήματα – κεφάλαια:

1. Εισαγωγή : Στην εισαγωγή παρουσιάζεται συνοπτικά το γενικό πλαίσιο της εργασίας και ο τελικός στόχος της. Επιχειρείται μια ανασκόπηση των εργασιών που έχουν παγκοσμίως δημοσιευτεί σχετικά με τη χρήση EMGs σε εξωσκελετικές εφαρμογές, και τέλος αποτυπώνεται η δομή της συγκεκριμένης διπλωματικής εργασίας.

2. Αρχιτεκτονική Συστήματος : Στο συγκεκριμένο κεφάλαιο συνοψίζονται όλα τα μέρη που απαρτίζουν τη δομή του συστήματος μας έτσι ώστε να μπορέσει ο αναγνώστης να αποκτήσει μια σαφή εικόνα του τρόπου λειτουργίας του.

3. Ανάλυση Σημάτων : Το παρόν κεφάλαιο επικεντρώνεται στην περιγραφή του υποσυστήματος ανάκτησης και επεξεργασίας ηλεκτρομυογραφικών σημάτων. Παρουσιάζονται κάποιες βασικές θεωρητικές γνώσεις επεξεργασίας σήματος και περιγράφεται η μεθοδολογία που ακολουθήθηκε έτσι ώστε να

μπορέσει να εξαχθεί με αντικειμενικό και ποσοτικά μετρήσιμο τρόπο η ένταση της μυϊκής καταπόνησης.

4. Δυναμική Ανάλυση Συστήματος : Στη συνέχεια αναλύονται τα δυναμικά μοντέλα του χεριού καθώς και της εξωσκελετικής διάταξης με σκοπό να χρησιμοποιηθούν στην προσομοίωση του όλου συστήματος χέρι – exoskeleton. Η προσομοίωση αυτή κρίνεται απαραίτητη για να εξαχθούν κάποιες ικανοποιητικές αρχικές τιμές κερδών του ελεγκτή που θα χρησιμοποιηθούν στον τελικό έλεγχο του exoskeleton.

5. Αλγόριθμοι Ελέγχου : Στο συγκεκριμένο κεφάλαιο περιέχονται τα δυο επιμέρους τμήματα του ελεγκτή του exoskeleton. Το πρώτο τμήμα είναι υπεύθυνο για την υποχωρητικότητα της διάταξης ενώ το δεύτερο ελέγχει την αναγκαία ενίσχυση για να παραληφθεί το τυχόν επιπλέον βάρος. Στο κομμάτι αυτό της εργασίας παρουσιάζονται και τα τελικά αποτελέσματα που εξήχθησαν από τη διενέργεια της όλης εργασίας.

6. Αξιολόγηση Αποτελεσμάτων & Μελλοντικές Κατευθύνσεις : Στο τελευταίο κεφάλαιο επιχειρείται μια συνολική σύνοψη και αποτίμηση των αποτελεσμάτων που πήραμε. Επιπλέον με αφορμή τα αποτελέσματα αυτά, προτείνονται πιθανές μελλοντικές κατευθύνσεις για περαιτέρω μελέτη και ερευνητική προσπάθεια.

ΚΕΦΑΛΑΙΟ 2^ο – ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΣΥΣΤΗΜΑΤΟΣ

2.1 Υποσύστημα Ηλεκτρομυογραφικών Σημάτων

Λόγω των παράλληλων εργασιών που πρέπει να τρέχουν συνεχώς για τη λειτουργία και τον έλεγχο της όλης διάταξης που έχει στηθεί για την παρούσα διπλωματική, το σύστημα μας έχει χωριστεί σε δύο διακριτά μέρη που ομαδοποιούν ένα σύνολο παρεμφερών λειτουργιών. Έτσι έχουμε το υποσύστημα που είναι υπεύθυνο για την τέλεση όλων των υπολογιστικών διαδικασιών που έχουν να κάνουν με την λήψη και την επεξεργασία των ηλεκτρομυογραφικών σημάτων. Κεντρική συντονιστική μονάδα στο σύνολο αυτό, αποτελεί ένας επιτραπέζιος προσωπικός υπολογιστής στον οποίο καταλήγουν όλες οι σχετικές με τα σήματα καλωδιακές συνδέσεις και στον οποίο τρέχουν οι αντίστοιχες εφαρμογές.



Εικόνα 2.1 ^[3]

2.1.1 Λειτουργικό Σύστημα Linux

Το λειτουργικό σύστημα που έχει επιλεγεί για τον κεντρικό υπολογιστή του υποσυστήματος που εξετάζουμε είναι το Linux. Αποτελεί σύστημα πραγματικού χρόνου οπότε καθίσταται ως πλέον κατάλληλο για εφαρμογές σαν αυτή που εξετάζουμε όπου απαιτείται απόλυτος συντονισμός εφαρμογών και συνδεδεμένων εξοπλισμών. Επιπλέον κρίθηκε ως το πιο δόκιμο λειτουργικό σύστημα για την άμεση συνεργασία με την μονάδα λήψης σημάτων η οποία περιγράφεται στη συνέχεια. Στο συγκεκριμένο λειτουργικό σύστημα έχουν προγραμματιστεί και λειτουργούν όλοι οι αλγόριθμοι που απαιτούνται οι σχετικοί με την λήψη, επεξεργασία και αποστολή των EMGs.

2.1.2 Βιοενισχυτής

Αποτελεί το πιο νευραλγικό κομμάτι του συστήματος λήψης δεδομένων από το χρήστη. Πρόκειται για μια ηλεκτρονική κονσόλα που έχει τη δυνατότητα να πάρει σήμα από πολλαπλά ενεργά ηλεκτρόδια λήψης τα οποία στη συνέχεια αποστέλλει στον ηλεκτρονικό υπολογιστή που είναι συνδεδεμένος με τη προσθήκη ειδικής κάρτας ανάκτησης δεδομένων. Πιο συγκεκριμένα χρησιμοποιείται το βιοενισχυτή Bagnoli-16 Desktop System της εταιρίας Delsys με 16 εξόδους BNC και προσαρμογέα (adaptor) SCSI-68. Έχει δυνατότητα ενίσχυσης σήματος 100, 1000 και 10000 φορές και εύρος λήψης EMGs 20-450 Hz με αποκλιση 10%. Τέλος είναι πιστοποιημένο ως ιατρική συσκευή κατά το πρότυπο IEC601-1 (Εικόνα 2.1, [3]).

2.1.3 Ενεργά Ηλεκτρόδια

Για να μπορέσουν τα ηλεκτρικά σήματα της κίνησης του μυ να καταγραφούν από την κονσόλα πρέπει να συνδεθούν τα δύο κατάλληλα

ηλεκτρόδια πάνω στο χέρι του χρήστη. Το πρώτο τοποθετείται στο κέντρο του δικέφαλου και είναι αυτό που μεταφέρει τα σήματα κατά τη διάρκεια της κίνησης. Το δεύτερο ηλεκτρόδιο, το οποίο καλείται γείωση, συνδέεται συνήθως στο πάνω μέρος της παλάμης του άλλου χεριού του χρήστη και έχει ως στόχο τη δημιουργία μιας τάσης αναφοράς επί της οποίας θα υπολογιστεί η τιμή του ηλεκτρομυογραφικού σήματος. Συνιστάται ο καθαρισμός με οινόπνευμα της περιοχής του δέρματος όπου θα συνδεθούν τα δύο ηλεκτρόδια. (Εικόνα 2.1, [3])

2.2 Υποσύστημα Εξωσκελετικής Διάταξης

Το δεύτερο υποσύνολο που χρησιμοποιείται στην παρούσα εργασία είναι αυτό της εξωσκελετικής διάταξης. Το υποσύστημα αυτό είναι ουσιαστικά αφοσιωμένο στον έλεγχο κίνησης του exoskeleton. Λαμβάνει δεδομένα από το υποσύστημα ηλεκτρομυογραφικών σημάτων και βάση αυτών ασκεί έλεγχο κίνησης στον κινητήρα της εξωσκελετικής διάταξης.

2.2.1 Λειτουργικό Σύστημα QNX

Ο υπολογιστής ο οποίος αποτελεί την υπολογιστική και συντονιστική μονάδα του συγκεκριμένου υποσυστήματος, εργάζεται με λειτουργικό σύστημα QNX. Πρόκειται και αυτό για ένα σύστημα πραγματικού χρόνου το οποίο διαθέτει τα απαραίτητα χαρακτηριστικά αξιοπιστίας και ευελιξίας που το καθιστούν ικανό και για την συγκεκριμένη εφαρμογή ελέγχου.

2.2.2 Κάρτα ServoToGo

Απευθείας συνδεδεμένη στον υπολογιστή καταρχάς αλλά και στον κινητήρα του exoskeleton, βρίσκεται η κάρτα ServoToGo. Χρησιμοποιείται σε

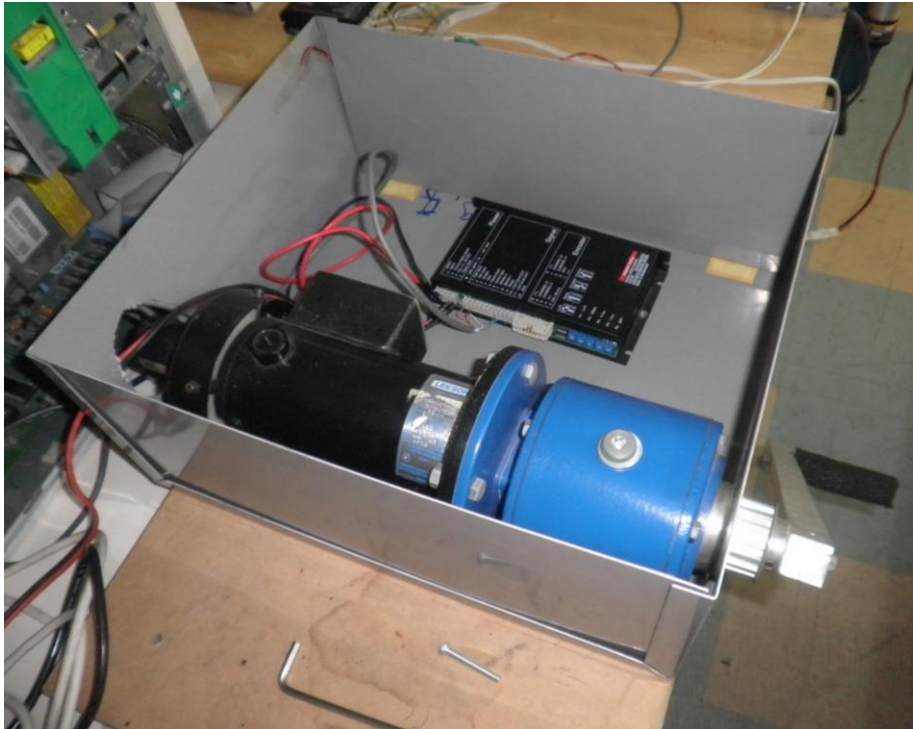
περιπτώσεις ελέγχου κινητήρων και αναλαμβάνει την μετατροπή που είναι απαραίτητη σε αντίστοιχες εφαρμογές. (Εικόνα 2.2, [8])



Εικόνα 2.2 ^[8]

2.2.3 Εξωσκελετική Διάταξη

Το μηχανικό κομμάτι του exoskeleton αποτελείται από μια πλατφόρμα, επάνω στην οποία εδράζεται ο κινητήρας συνεχούς ρεύματος (συμπεριλαμβανομένου του αντίστοιχου ενισχυτή) απαραίτητος για την κίνηση της διάταξης. Ο κινητήρας δίνει κίνηση σε κατάλληλο μειωτήρα στροφών για την αύξηση της εφαρμοζόμενης ροπής στην έξοδο του οποίου έχει προσαρτηθεί μια ράβδος αλουμινίου. Η ράβδος αυτή αποτελεί τον βραχίονα που θα συνδεθεί με τον αγκώνα του χρήστη και είναι αυτή που ουσιαστικά θα αναλάβει το επιπλέον φορτίο από το μυ. (Εικόνα 2.3, φωτογραφία)



Εικόνα 2.3 (φωτογραφία)

2.2.4 Κλωβός μείωσης ηλεκτρομαγνητικής ακτινοβολίας

Κατά τη λειτουργία της όλης διάταξης ο χρήστης βρίσκεται πολύ κοντά στον κινητήρα έτσι ώστε να μπορεί να συνδέσει άμεσα τον αγκώνα του με τη ράβδο αλουμινίου του exoskeleton. Επιπλέον στο ίδιο χέρι βρίσκονται προσκολλημένα τα ενεργά ηλεκτρόδια για τη λήψη των EMGs. Το γεγονός αυτό της παράλληλης χρήσης κινούμενων ηλεκτρομηχανολογικών μερών με ευαίσθητα ηλεκτρονικά στοιχεία δειγματοληψίας προσθέτει έντονο ηλεκτρονικό θόρυβο στο σύστημα επηρεάζοντας την λήψη των σημάτων. Αρχικά επιχειρήθηκε να τοποθετηθεί διηλεκτρικό υλικό στα στοιχεία που ο χρήστης ερχόταν σε επαφή με τα μεταλλικά μέρη της διάταξης αλλά χωρίς αποτέλεσμα. Έτσι για τις ανάγκες της διπλωματικής εργασίας κατασκευάστηκε σε μηχανουργείο αλουμινένιος κλωβός μείωσης ηλεκτρομαγνητικής ακτινοβολίας ο οποίος εσωκλείει τη διάταξη κίνησης του exoskeleton και επιτυγχάνει την απομόνωση των δύο υποσυστημάτων. Στα πειράματα που διεξήχθησαν μετά την τοποθέτηση του κλωβού διαπιστώθηκε

ότι ο θόρυβος που εμφανιζόταν στην αρχή είχε εξαλειφθεί. (Εικόνα 2.4, φωτογραφία)



Εικόνα 2.4 (φωτογραφία)

ΚΕΦΑΛΑΙΟ 3^ο – ΑΝΑΛΥΣΗ ΣΗΜΑΤΩΝ

3.1 Δειγματοληψία

Καθοριστικής σημασία παράγοντας σε κάθε διαδικασία ανάκτησης και επεξεργασίας σήματος είναι η οριστικοποίηση των παραμέτρων της δειγματοληψίας. Είναι σημαντικό να έχει καθοριστεί από την αρχή όλων των διαδικασιών, ποιες θα είναι οι τιμές εκείνες που ορίζουν τη συχνότητα και το μέγεθος των δεδομένων. Με τον τρόπο αυτό αποφεύγονται προβλήματα απώλειας δεδομένων ή και ελλείψεως επικοινωνίας μεταξύ συσκευών ανάκτησης.

Στην περίπτωση μας έχουν επιλεγεί οι ακόλουθες παράμετροι δειγματοληψίας:

- Συχνότητα = 1000Hz (βλ. Κεφ. 2.1.2)
- Σύνολο δείγματος = 30.000
- Ποσότητα παρτίδας = 100
- Σύνολο παρτίδων = 300

Οι ανωτέρω τιμές ρυθμίζονται σε όλες τις εμπλεκόμενες συσκευές του υποσυστήματος λήψης σημάτων.

3.2 Επεξεργασία Σημάτων

Το επόμενο στάδιο μετά την ανάκτηση των σημάτων από την κονσόλα είναι η επεξεργασία τους για την εξαγωγή ποσοτικών συμπερασμάτων. Απώτερος στόχος της όλης διαδικασίας διαχείρισης των σημάτων αποτελεί η εκτίμηση της λειτουργίας του χεριού του χρήστη έτσι ώστε από τα δεδομένα αυτά να δοθεί η εντολή οδήγησης του exoskeleton.

3.2.1 Θεωρητική Προσέγγιση

Η λειτουργία του ανθρώπινου μυ σίγουρα δεν αποτελεί μια μονοσήμαντη και εύκολη στην καταγραφή της διαδικασία. Το σήμα που λαμβάνεται από τα ηλεκτρόδια δεν είναι δυνατόν να χρησιμοποιηθεί άμεσα για την εξαγωγή ασφαλών συμπερασμάτων. Βεβαίως αυτή είναι η συνήθης κατάσταση σε περιπτώσεις ανάκτησης ηλεκτρομαγνητικών σημάτων πολλών εφαρμογών. Για το λόγο αυτό έχουν αναπτυχθεί ιδιαίτερα επιτυχημένες τακτικές επεξεργασίας σημάτων οι οποίες συνήθως προηγούνται της αξιολόγησης των ποιοτικών χαρακτηριστικών τους. Συχνά χρειάζεται να «περάσουμε» τα σήματα μας μέσα από φίλτρα έτσι ώστε να μπορέσουμε να διαχειριστούμε την πληροφορία που αυτά μεταφέρουν.

Στη συνέχεια θα επιχειρήσουμε να περιγράψουμε συνοπτικά τις πιο διαδεδομένες μεθόδους αρχικής επεξεργασίας σημάτων με την παράθεση των μαθηματικών σχέσεων με τις οποίες εκφράζονται.

➤ Integral of Absolute Value (IAV) :

Ως πρώτο φίλτρο παίρνουμε το ολοκλήρωμα της απόλυτης τιμής, το οποίο και υπολογίζεται από την ακόλουθη σχέση

$$IAV = \frac{1}{N} \sum_{i=1}^N x_i$$

➤ Variance (VAR) :

Το φίλτρο variance αποτελεί μέτρο της ισχύος του σήματος και ο τύπος υπολογισμού της είναι

$$VAR = \frac{1}{N-1} \sum_{i=1}^N x^2_i$$

όπου i είναι ο αύξων αριθμός της μέτρησης x και N ο αριθμός των δειγμάτων σε κάθε δειγματοληψία.

➤ Wilson Amplitude (WAMP) :

Με αυτό το φίλτρο μετράμε τις φορές που η διαφορά ανάμεσα σε δύο μετρήσεις είναι μεγαλύτερη από το όριο που έχουμε θέσει. Είναι ένα δείκτη της μυϊκής σύσπασης και υπολογίζεται ως εξής

$$WAMP = \sum_{i=1}^N f(|x_i - x_{i+q}|)$$

$$f(x) = \begin{cases} 1 \dots \text{if } \cdot x > \text{threshold} \\ 0 \dots \text{otherwise} \end{cases}$$

➤ Zero Crossing (ZC) :

Με αυτό το φίλτρο μετράμε ποσες φορές το σήμα περνά την μηδενική τιμή και για τον υπολογισμό του φίλτρου χρησιμοποιείται ο ακόλουθος τύπος

$$ZC = \sum_{i=1}^N \text{sgn}(-x_i x_{i+1})$$

$$\text{sgn}(x) = \begin{cases} 1 \dots \text{if } \cdot x > 0 \\ 0 \dots \text{otherwise} \end{cases}$$

➤ Wavelength (WL) :

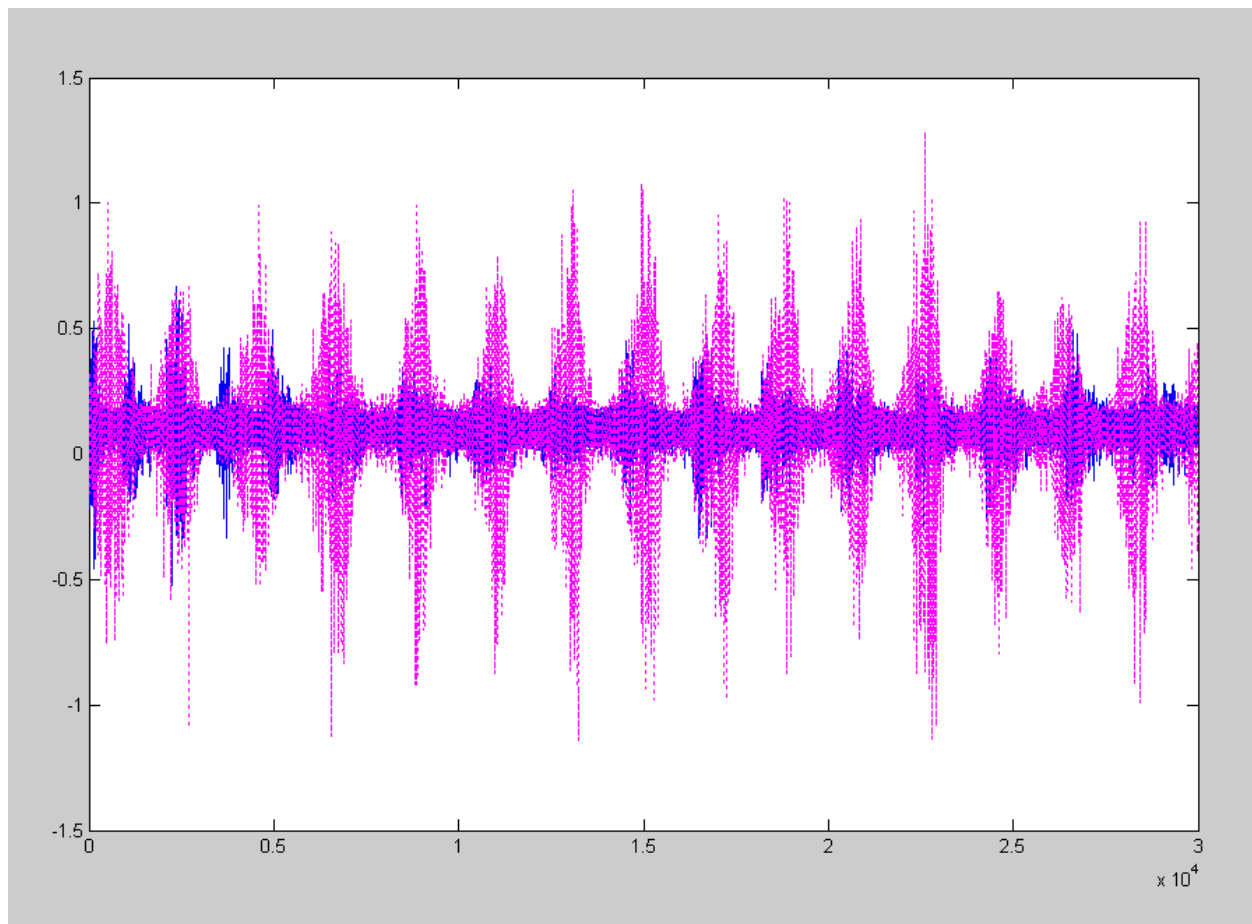
Το φίλτρο αυτό υπολογίζει την απόλυτη τιμή της διαφοράς ανάμεσα σε δύο γειτονικές τιμές, δηλαδή

$$W(n) = \sum_{i=n-N+1}^n |x_i - x_{i-1}|$$

- Number of turns (NT) : Το συγκεκριμένο φίλτρο εκφράζει τον αριθμό των εναλλαγών στο πρόσημο ενός σήματος σε ορισμένη χρονική διάρκεια.

3.2.2 Επιλογή Φίλτρων

Στην περίπτωση μας η γραφική παράσταση των EMGs, που λαμβάνουμε απ' ευθείας από το χρήστη προκύπτει όπως φαίνεται στο ακόλουθο σχήμα, όπου η μπλε γραμμή είναι το σήμα όταν το χέρι δε φέρει βάρος ενώ η μωβ όταν στο χέρι είναι τοποθετημένο βάρος. Όπως παρατηρούμε στα γραφήματα το σήμα είναι ιδιαίτερα πυκνό και δεν υπάρχει σαφής στάθμη διαχωρισμού του σήματος στο οποίο το χέρι σηκώνει βάρος από αυτό που το χέρι κινείται ελεύθερα. Συνεπώς είναι απαραίτητη η χρήση φίλτρων, έτσι ώστε να μπορούμε να εξάγουμε αντικειμενικά ποσοτικά συμπεράσματα.



Για την δημιουργία των φίλτρων και την επεξεργασία των EMGs χρησιμοποιούμε τα δεδομένα της θεωρίας που αναφέρθηκε στο προηγούμενο εδάφιο. Πιο συγκεκριμένα για την επεξεργασία του αρχικού σήματος που λαμβάνουμε ακολουθήθηκε η εξής διαδικασία:

- Θέτουμε ως βήμα δειγματοληψίας τις 100 τιμές EMGs
- Θέτουμε ως βήμα επεξεργασίας την τιμή 100

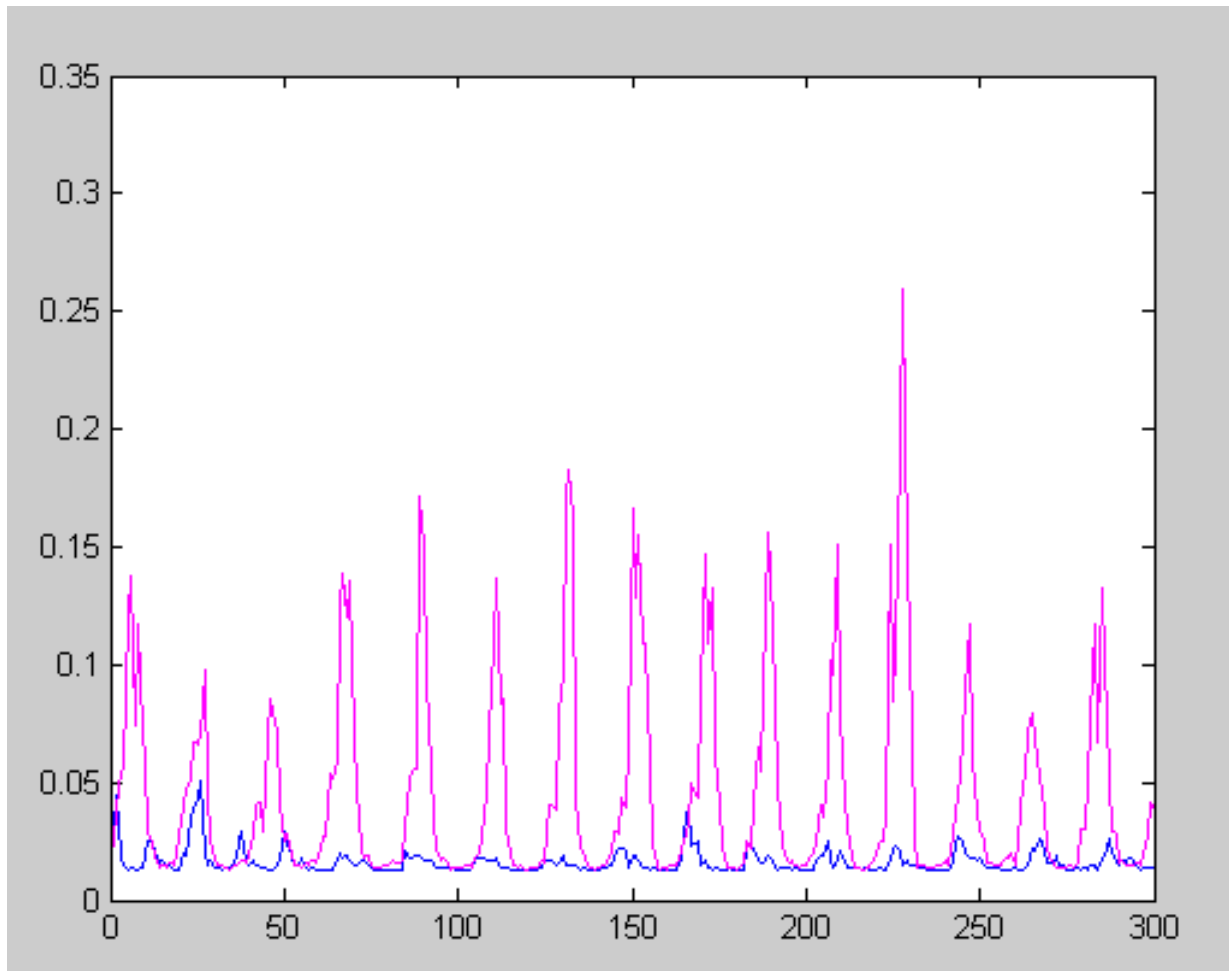
Έτσι η πρώτη τιμή του νέου γραφήματος θα προκύπτει από την επεξεργασία των εκατό πρώτων δειγμάτων (1-100), η δεύτερη από τη δεύτερη εκατοστάδα δειγμάτων (101-200) και συνεχίζει κατά αυτόν τον τρόπο. Προφανώς προκύπτουν 300 από τις 30000 τιμές που αρχικά είχαμε.

Βάση αυτής της διαδικασίας και μετά από την πειραματική μελέτη των φίλτρων που εξηγήθηκαν στο κεφάλαιο 3.2.1 έγινε η επιλογή των κατάλληλων για τη συγκεκριμένη εφαρμογή. Τα φίλτρα αυτά είναι:

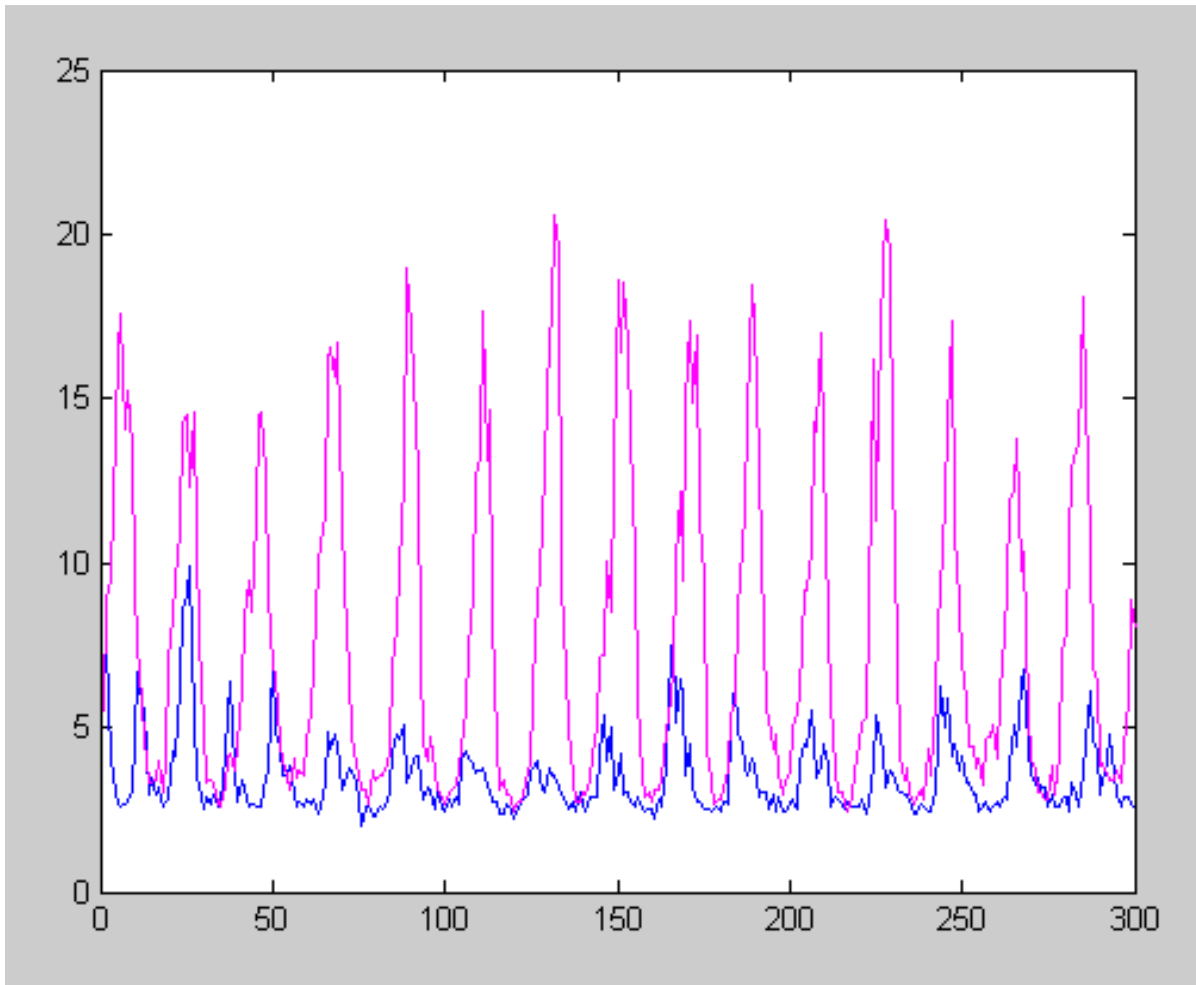
- 1) variance
- 2) wavelength

Στη συνέχεια φαίνεται το σήμα αφού το έχουμε επεξεργαστεί με αυτά τα φίλτρα. Στα γραφήματα που ακολουθούν η μπλε γραμμή είναι το σήμα όταν το χέρι δε φέρει βάρος ενώ η μωβ όταν στο χέρι είναι τοποθετημένο βάρος.

Variance



Wavelength

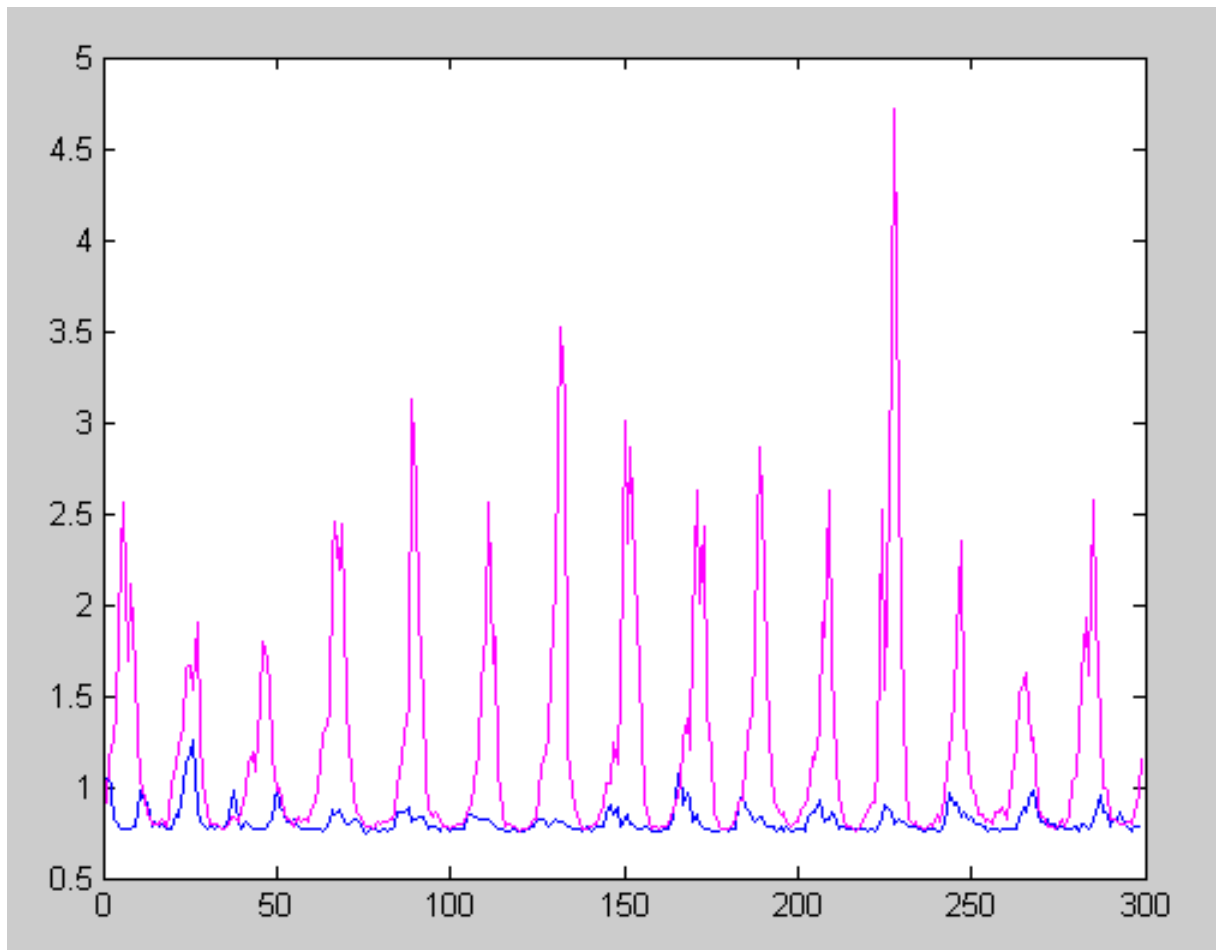


Σε μετέπειτα στάδιο της εργασίας, για λόγους που θα αναφερθούν σε επόμενο κεφάλαιο (βλ. Κεφ. 5.2.1) κρίνεται απαραίτητη η εν μέρη αλλαγή του τρόπου επεξεργασίας του σήματος. Γίνεται αλλαγή στο βήμα επεξεργασίας (όπως αυτό ορίστηκε πιο πάνω) από εκατό σε ένα, ενώ το βήμα δειγματοληψίας παραμένει το ίδιο με πριν (100). Δηλαδή μετά την αλλαγή της διαδικασίας φιλτραρίσματος των EMGs για την ίδια εκατοστάδα δειγμάτων γίνεται επεξεργασία με αφητηρία κάθε ένα από τα εκάστοτε δείγματα της εκατοστάδας (1-100, 2-101, 3-102,...). Ύστερα από έλεγχο που έγινε διαπιστώθηκε ότι παρά την αλλαγή του βήματος επεξεργασίας, τα φίλτρα που έχουν επιλεγεί εξακολουθούν να είναι κατάλληλα για την εφαρμογή μας, χωρίς να είναι απαραίτητη η προσθήκη κάποιου νέου φίλτρου.

3.3 Αναγνώριση Μυϊκής Έντασης

Σκοπός μας σε αυτό το στάδιο είναι να βρούμε κάποιο κριτήριο, το οποίο θα μας βοηθήσει να καθορίσουμε σε πραγματικό χρόνο πότε το χέρι φέρει βάρος και πότε όχι. Δηλαδή επιδιώκουμε κατά τη διάρκεια που λαμβάνουμε EMGs να μπορεί παράλληλα το σύστημα να αντιλαμβάνεται την ύπαρξη ή όχι μυϊκής καταπόνησης μεγαλύτερης από αυτή που έχει όταν είναι ελεύθερο.

Για να επιτευχθεί αυτός ο στόχος κρίθηκε απαραίτητη αρχικά η συγκέντρωση των αποτελεσμάτων των δύο φίλτρων σε μία γραφική παράσταση. Έτσι δημιουργήθηκε ένα πρόγραμμα κανονικοποίησης των δύο γραφικών παραστάσεων (μία για κάθε φίλτρο) και ακολούθησε η πρόσθεσή των γραφημάτων με βάρος μεταξύ τους και χωρίς βάρος μεταξύ τους επίσης. Η κανονικοποίηση έγινε με τη χρήση δύο παραμέτρων της WI_{max} και της $Varce_{max}$. Η παράμετρος WI_{max} αντιπροσωπεύει την μέγιστη τιμή του γραφήματος που προκύπτει από όλα τα EMGs μετά το φίλτρο wavelength και αντίστοιχα η $Varce_{max}$ είναι η μέγιστη τιμή μετά το φίλτρο variance. Οι δύο αυτές τιμές, μας είναι ιδιαίτερα χρήσιμες για να μπορέσουμε να κάνουμε την κανονικοποίηση και στην περίπτωση της λήψης EMGs σε πραγματικό χρόνο. Για να γίνει η κανονικοποίηση και στις δύο περιπτώσεις διαιρούμε τα αποτελέσματα του κάθε φίλτρου με τη μέγιστη τιμή που προκύπτει για κάθε μία από αυτές (δηλαδή τις WI_{max} και $Varce_{max}$). Η γραφική παράσταση που προέκυψε από την κανονικοποίηση είναι η ακόλουθη.



Από τα αποτελέσματα που προέκυψαν μπορέσαμε να καθορίσουμε μία τιμή κάτω από την οποία θεωρούμε ότι δεν υπάρχει βάρος, ενώ όσες τιμές βρίσκονται πάνω από αυτή δηλώνουν την ύπαρξη βάρους. Η τιμή αυτή μπορεί να θεωρηθεί με ασφάλεια η μέγιστη τιμή του γραφήματος που προέρχεται από την επεξεργασία (φιλτράρισμα και κανονικοποίηση) των EMGs στην περίπτωση μη ύπαρξης βάρους. Επελέγη αυτή η τιμή διότι δεν έχουμε κάποια άλλη ένδειξη ότι μπορεί να υπάρξει μεγαλύτερη τιμή για την περίπτωση στην οποία δεν καταπονείται ο μυς με πρόσθετο βάρος. Η τιμή αυτή του κριτηρίου είναι πολύ χρήσιμη και για τη δημιουργία της προσομοίωσης του συστήματος (βλ. Κεφ. 5.2.1).

ΚΕΦΑΛΑΙΟ 4^ο – ΔΥΝΑΜΙΚΗ ΑΝΑΛΥΣΗ ΣΥΣΤΗΜΑΤΟΣ

4.1 Μοντελοποίηση Ανθρώπινου Χεριού

Ύστερα από τις διαδικασίες που περιγράψαμε στο προηγούμενο κεφάλαιο (βλ. Κεφ. 3) και αφορούν στην υλοποίηση του υποσυστήματος σημάτων αυτό που μένει να γίνει είναι η εφαρμογή του ελέγχου στο exoskeleton. Προτού όμως μπορέσουμε να προχωρήσουμε στο υποσύστημα της εξωσκελετικής διάταξης θα πρέπει να στηθεί μια εφαρμογή προσομοίωσης έτσι ώστε να μπορέσουμε να έχουμε μια εικόνα για τα αρχικά κέρδη που θα τοποθετήσουμε στον PID. Η προσομοίωση αυτή λειτουργεί σε περιβάλλον Matlab και χρειάζεται το δυναμικό μοντέλο του χεριού του χρήστη. Πρέπει δηλαδή να είμαστε σε θέση να υπολογίσουμε σε περιβάλλον προσομοίωσης, πως θα αντιδράσει το χέρι κατά την κίνηση του με βάρος και υπό την ταυτόχρονη υποστήριξη από την εξωσκελετική διάταξη. Αυτό που μας ενδιαφέρει στην περίπτωση μας φυσικά δεν είναι οποιεσδήποτε κινηματικές ή δυναμικές φυσικές παράμετροι της κίνησης του χεριού. Το σημαντικό στην περίπτωση μας είναι να δημιουργηθεί ένα μοντέλο που να είναι σε θέση να μας δίνει ανά πάσα στιγμή την υπολογιζόμενη τιμή του επεξεργασμένου ηλεκτρομυογραφικού σήματος από το χέρι που θέλουμε να προσομοιώσουμε.

Στην ουσία, κατά την προσομοίωση χρησιμοποιούμε ένα μοντέλο ηλεκτρομυογραφικού σήματος το οποίο το έχουμε καταγράψει εκ των προτέρων και το χρησιμοποιούμε σαν πηγή τροφοδότησης της προσομοίωσης. Το σήμα αυτό αποτελεί την κυματομορφή που εξάγει το χέρι χωρίς την σύζευξη του με το exoskeleton. Για να μπορέσουμε να μελετήσουμε το όλο σύστημα σε ένα ικανό εύρος περιπτώσεων δημιουργούμε δυο διαφορετικές κυματομορφές, - μια με βάρος ενός κιλού, και μια με βάρος δύο κιλών. Επιδιώκουμε οι κινήσεις κατά τη διάρκεια της καταγραφής του σήματος να είναι κατά το δυνατόν σταθερές και ομοιόμορφες έτσι ώστε να είναι συγκρίσιμα τα αποτελέσματα. Εφαρμόζουμε τώρα την όλη φιλοσοφία επεξεργασίας σημάτων που περιγράψαμε στο προηγούμενο κεφάλαιο (βλ. Κεφ. 3) στις αρχικές κυματομορφές που έχουμε καταγράψει. Η εφαρμογή επιβεβαιώνει την ύπαρξη βάρους στο χέρι και τελικώς υπολογίζει την εντολή ενίσχυσης που κανονικά θα έδινε το υποσύστημα σημάτων στο υποσύστημα

του exoskeleton. Επιστρέφοντας τώρα ως ανάδραση την ροπή με την οποία συνεισφέρει η εξωσκελετική διάταξη την ενίσχυση στο χέρι, χρησιμοποιεί το δυναμικό μοντέλο του χεριού έτσι ώστε να μεταβάλλει κατάλληλα την κυματομορφή – πηγή που έχουμε καταγράψει αρχικά. Η τιμή αυτή που δείχνει πως αντιδρά το σήμα του χεριού στην εξωσκελετική ενίσχυση, αποτελεί τώρα την πηγή εισόδου στο επόμενο βήμα της προσομοίωσης κλείνοντας με τον τρόπο αυτό τον βρόγχο ελέγχου.

Η συνάρτηση που εκφράζει τον τρόπο με τον οποίο το χέρι του χρήστη αντιδρά στην ενίσχυση που προσφέρει η εξωσκελετική διάταξη παρουσιάζεται στη συνέχεια:

$$EMG_{Real} = EMG_{Graph(j)} * (1 + coef * (t_b - t_{exos})) / (1 + coef * t_b)$$

όπου $coef = (EMG_{max} / EMG_{ref} - 1) / t_b$ ο συντελεστής που εκφράζει την αναλογία με την οποία το σήμα του μυ μεταβάλλεται, συναρτήσει του βάρους που τοποθετούμε επί αυτού.

Επιπλέον οι παράμετροι που εμφανίζονται στην παραπάνω σχέση αναλύονται ως εξής:

$EMG_{Graph(j)}$: η “τρέχουσα” τιμή του γραφήματος του επεξεργασμένου ηλεκτρομυογραφικού σήματος όταν το χέρι φέρει βάρος

EMG_{ref} : η τιμή του κριτηρίου, δηλαδή το όριο πάνω από το οποίο θεωρώ ότι υπάρχει βάρος

t_b : ροπή βάρους

t_{exos} : ροπή εξωσκελετικού μηχανισμού

4.2 Μοντελοποίηση Εξωσκελετικής Διάταξης

Στο προηγούμενο υποκεφάλαιο θεωρήσαμε δεδομένη την ικανότητα να μπορούμε να υπολογίσουμε την ροπή ενίσχυσης που συνεισφέρει το

exoskeleton συναρτήσει της εντολής που δίδεται από το υποσύστημα σημάτων. Αυτό είναι εφικτό με τη βοήθεια του μοντέλου της εξωσκελετικής διάταξης που περιγράφεται στο παρόν υποκεφάλαιο.

Η εξίσωση που περιγράφει την προσφερόμενη από το exoskeleton ροπή είναι:

$$t_{exos} = k_t k_r i_{exos}$$

Οι παράμετροι της πιο πάνω σχέσης ορίζονται ως εξής:

t_{exos} : ροπή εξωσκελετικού μηχανισμού

k_t : σταθερά μετατροπής ρεύματος κινητήρα σε ροπή εξόδου κινητήρα

k_r : σταθερά μείωσης του μειωτήρα στροφών

i_{exos} : ρεύμα κινητήρα εξωσκελετικού μηχανισμού

ΚΕΦΑΛΑΙΟ 5^ο – ΑΛΓΟΡΙΘΜΟΙ ΕΛΕΓΧΟΥ

5.1 Ελεγκτής Υποχωρητικότητας

Με τον ελεγκτή υποχωρητικότητας ο εξωσκελετικός μηχανισμός πρέπει να είναι υποχωρητικός στις κινήσεις του χρήστη. Αυτό σημαίνει ότι πρέπει ο exoskeleton να ακολουθεί την κίνηση του χεριού χωρίς ο χρήστης να αντιλαμβάνεται επιπλέον κόπωση από αυτή που νιώθει όταν το χέρι του κινείται ελεύθερο.

Για την υλοποίηση του ελεγκτή υποχωρητικότητας χρησιμοποιούμε το δυναμικό μοντέλο του exoskeleton, το οποίο υπολογίζεται από την ακόλουθη σχέση:

$$I_{exos} \cdot \ddot{q} + k_{rII}^2 \cdot I_{mII} \cdot \ddot{q} + b_{exos}(\dot{q}) + m_{exos} \cdot g \cdot l_{cgexos} \cdot \sin q = \tau_{exos}$$

Εάν στην παραπάνω εξίσωση αντικαταστήσουμε $\tau_{exos} = k_l k_r i_{exos}$ τότε προκύπτει ότι:

$$\frac{I_{exos} \cdot \ddot{q} + k_{rII}^2 \cdot I_{mII} \cdot \ddot{q}}{K_{III} \cdot k_{rII}} + \frac{b_{exos}(\dot{q})}{K_{III} \cdot k_{rII}} + \frac{m_{exos} \cdot g \cdot l_{cgexos} \cdot \sin q}{K_{III} \cdot k_{rII}} = i_{II}$$

εκφράζοντας την εξίσωση υποχωρητικότητας της εξωσκελετικής διάταξης.

Οι παράμετροι που αναφέρονται ορίζονται ως εξής:

\ddot{q} : Γωνιακή επιτάχυνση μηχανισμού

\dot{q} : Γωνιακή ταχύτητα μηχανισμού

I_{hand} : Ροπή αδρανείας της ράβδου που αντιστοιχεί στο αντιβράχιο

I_{ml} : Ροπή αδρανείας του δρομέα του κινητήρα που παίζει το ρόλο του αγκώνα

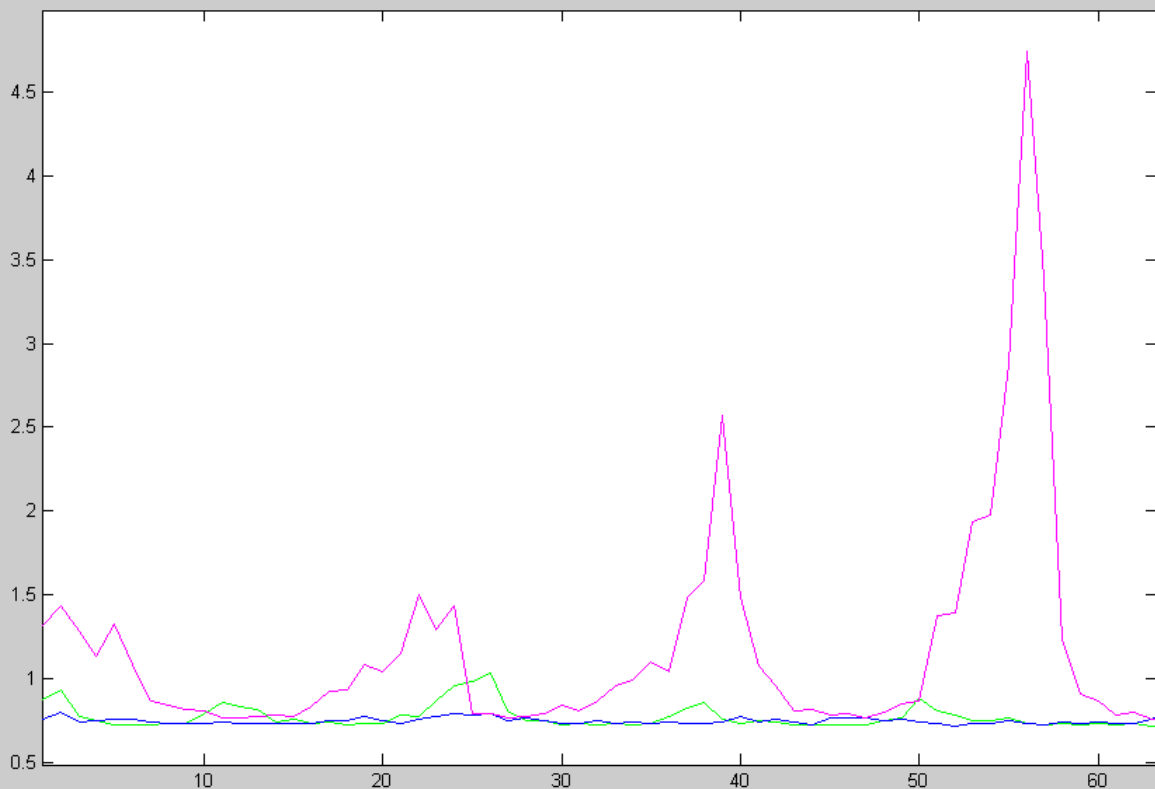
k_{rl} : λόγος μείωσης του μειωτήρα του υποσυστήματος του χεριού

I_{exos} : Ροπή αδρανείας της ράβδου που αντιστοιχεί στο exoskeleton

I_{mII} : Ροπή αδρανείας του δρομέα του κινητήρα του exoskeleton

k_{rII} : λόγος μείωσης του μειωτήρα του υποσυστήματος του exoskeleton

Με τη χρήση του ελεγκτή υποχωρητικότητας όπως αυτός υλοποιήθηκε τελικά προγραμματιστικά ο εξωσκελετικός μηχανισμός είναι υποχωρητικός στις κινήσεις του χρήστη. Ο χρήστης δεν αντιλαμβάνεται επιπλέον κόπωση είτε το χέρι του κινείται ελεύθερο είτε μαζί με την εξωσκελετική διάταξη. Την ύπαρξη ή όχι επιπλέον κόπωσης του χεριού την ποσοτικοποιούμε με τη λήψη και επεξεργασία EMGs για την περίπτωση κίνησης του ελεύθερου χεριού και του χεριού με το exoskeleton και τη σύγκριση των συνεπαγόμενων αποτελεσμάτων. Ελέγχουμε εάν οι τιμές που προκύπτουν κατά τη χρήση του exoskeleton είναι μεγαλύτερες από την τιμή του κριτηρίου που έχουμε θέσει, πάνω από το οποίο θεωρούμε ότι υπάρχει μυϊκή καταπόνηση. Παρατηρούμε ότι είναι ελάχιστες οι τιμές εκείνες οι οποίες είναι μεγαλύτερες από αυτή του κριτηρίου. Στο διάγραμμα που ακολουθεί φαίνονται τα αποτελέσματα της σύγκρισης κίνησης του ελεύθερου χεριού (μπλε γραμμή) και του χεριού με τη χρήση υποχωρητικού εξωσκελετικού μηχανισμού (πράσινη γραμμή). Στο διάγραμμα συμπεριλαμβάνεται και η γραφική παράσταση της κίνησης του χεριού με βάρος (μωβ γραμμή) ώστε να είναι διακριτή και γραφικά η καλή απόκριση του ελεγκτή υποχωρητικότητας.



Να σημειωθεί σε αυτό το σημείο ότι η όποια μικρή μυϊκή καταπόνηση θα αντιλαμβάνεται το χέρι του χρήστη από την παράλληλη κίνηση του χεριού του με το exoskeleton θα τη διορθώνει ο ελεγκτής ενίσχυσης.

5.2 Ελεγκτής Ενίσχυσης

Με την κατασκευή του ελεγκτή ενίσχυσης επιδιώκουμε εκτός από την υποχωρητικότητα του συστήματος την ενίσχυσή του σε περίπτωση επιπλέον μυϊκής καταπόνησης λόγω ανύψωσης βάρους. Έτσι ο χρήστης δεν πρέπει να αντιλαμβάνεται το πρόσθετο βάρος, δηλαδή το ηλεκτρομυογραφικό σήμα να μην αλλάζει είτε έχει βάρος είτε όχι.

5.2.1 Επιλογή Κερδών με Προσομοίωση

Για να γίνει η επιλογή των κερδών που θα χρησιμοποιηθούν για τον έλεγχο στο exoskeleton γίνεται προσομοίωση του συστήματος. Στο Κεφάλαιο 4 ήδη μελετήθηκε η μοντελοποίηση του ανθρώπινου χεριού και της εξωσκελετικής διάταξης, τα οποία μας είναι απαραίτητα για την προσομοίωση του συστήματος.

Η μαθηματική υλοποίηση του ελεγκτή έχει την πιο κάτω μορφή:

$$i_{exos} = K_p \cdot e + K_d \cdot \dot{e} + K_i \int e dt$$

όπου:

i_{exos} : ρεύμα που πρέπει να στείλουμε στον κινητήρα του exoskeleton

ώστε να παραχθεί η ροπή ενίσχυσης

K_p : αναλογικό κέρδος

K_d : διαφορικό κέρδος

K_i : ολοκληρωτικό κέρδος

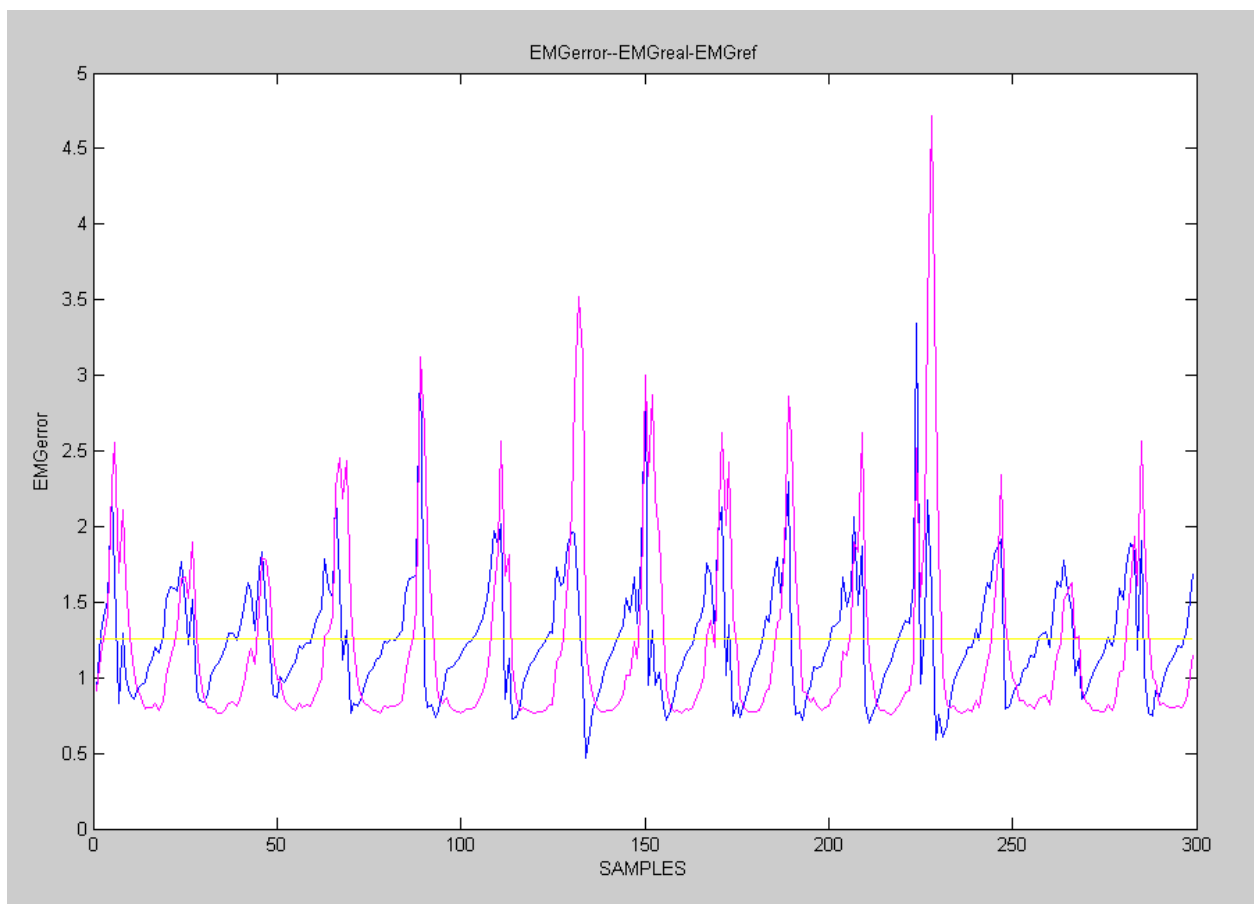
e : σφάλμα

Το σφάλμα e υπολογίζεται από τη σχέση:

$$e = \text{EMG}_{\text{real}} - \text{EMG}_{\text{ref}},$$

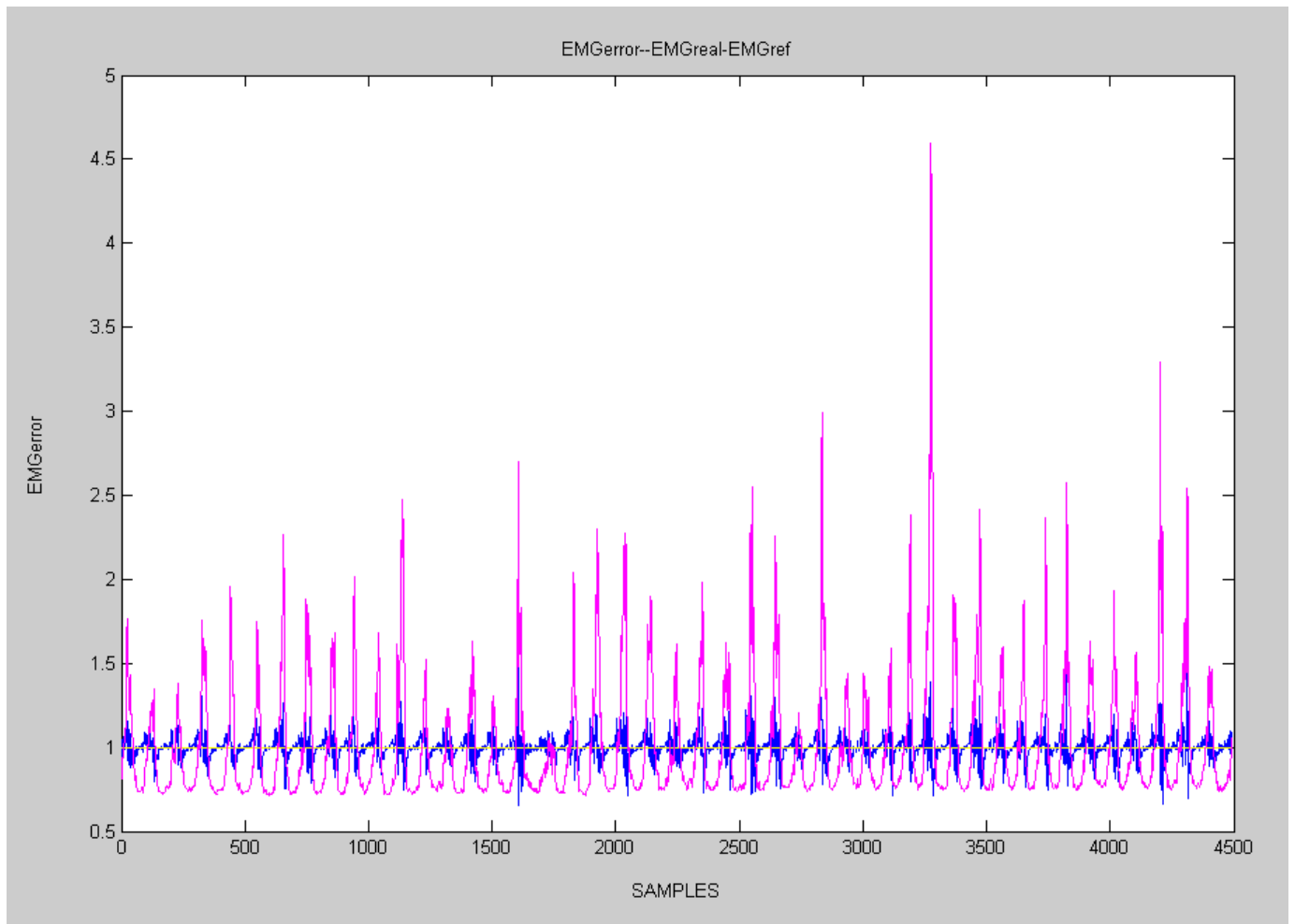
όπου EMG_{real} είναι το αποτέλεσμα της προσομοίωσης του χεριού και αντιπροσωπεύει το επεξεργασμένο σήμα από το χέρι που θα προέκυπτε από την πραγματική εφαρμογή. Το EMG_{ref} είναι σταθερά και είναι η τιμή του κριτηρίου. Έτσι βλέπουμε ότι το σφάλμα μηδενίζεται όταν η τιμή που προέρχεται από το μοντελοποιημένο χέρι είναι ίση με την τιμή του κριτηρίου που έχουμε θέσει, όταν δηλαδή δεν αντιλαμβάνεται το χέρι πρόσθετη μυϊκή καταπόνηση.

Τα αποτελέσματα της προσομοίωσης κρίθηκαν μη ικανοποιητικά και αυτό οφειλόταν στον τρόπο επεξεργασίας των EMGs. Όπως αναφέρθηκε στο Κεφάλαιο 3 αρχικά το βήμα επεξεργασίας ήταν 100, έτσι δεν ήταν αρκετά «πυκνά» τα αποτελέσματα ώστε ο ελεγκτής να μπορεί να ακολουθήσει πιστά τις μεταβολές των σημάτων και επί της ουσίας δε μπορούσε να επιτευχθεί έλεγχος. Στο πιο κάτω γράφημα -όπου η μωβ γραμμή είναι η γραφική παράσταση του επεξεργασμένου EMGs με βάρους και η μπλε γραμμή είναι η γραφική παράσταση της προσομοίωσης όταν θεωρούμε ότι έχουμε συνδεδεμένο το χέρι και το exoskeleton και ασκείται έλεγχος- φαίνεται η αδυναμία του συστήματος να επιτευχθεί ο έλεγχος.



Σε αυτό το σημείο κρίθηκε αναγκαία η αλλαγή του βήματος επεξεργασίας από εκατό σε ένα, όπως αναφέρθηκε στο Κεφάλαιο 3. Με την αλλαγή αυτή προέκυψε νέα τιμή κριτηρίου. Ο ελεγκτής της εξωσκελετικής διάταξης

προσφέρει τώρα επιθυμητά αποτελέσματα και υπολογίστηκαν τα κέρδη που θα χρησιμοποιηθούν στο πραγματικό μοντέλο. Το γεγονός αυτό παρουσιάζεται γραφικά στο ακόλουθο σχήμα.



5.2.2 Πειραματική υλοποίηση

Στο προηγούμενο υποκεφάλαιο υπολογίσαμε με προσομοίωση τα κέρδη του ελεγκτή. Σε αυτό θα αξιοποιήσουμε τα αποτελέσματα αυτά ώστε να

βρούμε τα κέρδη που αντιστοιχούν στον έλεγχο της πραγματικής εξωσκελετικής διάταξης.

Συνδέουμε τον Η/Υ που λαμβάνει και επεξεργάζεται τα EMGs με τον Η/Υ του exoskeleton μέσω της σειριακής θύρας RS232. Ο Η/Υ που επεξεργάζεται τα EMGs αναγνωρίζει πότε η μυϊκή καταπόνηση είναι μεγαλύτερη από το όριο που έχουμε θέσει, για το συγκεκριμένο χρήστη, ως μέγιστο για τη μη ύπαρξη βάρους (τιμή κριτηρίου). Έτσι για κάθε τιμή πάνω από την τιμή του κριτηρίου στέλνει μια τιμή ελέγχου στον υπολογιστή που είναι συνδεδεμένο το exoskeleton, ενώ για κάθε τιμή μικρότερη του κριτηρίου στέλνει μηδενική τιμή. Επίσης, όπως έχει αναφερθεί και στο Κεφάλαιο 5.1, η εξωσκελετική διάταξη είναι υποχωρητική στις κινήσεις του χρήστη, με τη βοήθεια ενός ελεγκτή. Στόχος μας είναι όταν ο Η/Υ του exoskeleton λαμβάνει μηδενική τιμή ο ελεγκτής υποχωρητικότητας να λειτουργεί ως έχει. Όταν, όμως, λαμβάνει μια οποιαδήποτε άλλη τιμή ελέγχου από τον Η/Υ που επεξεργάζεται τα EMGs, τα κέρδη του ελεγκτή να αλλάζουν και να επιτυγχάνεται η ενίσχυση.

Πειραματικά διαπιστώθηκε ότι η μεταφορά της τιμής ελέγχου από τον έναν υπολογιστή στον άλλον καθυστερεί σημαντικά την παράλληλη επικοινωνία του Η/Υ του exoskeleton με την κάρτα ServoToGo. Το γεγονός αυτό έχει ως αποτέλεσμα να χαθεί ο συντονισμός με το χρονοστάθι της κάρτας και να εγείρονται πλήθος ηλεκτρονικών σφαλμάτων. Ουσιαστικά, η εφαρμογή λόγω των πόρων του συστήματος που αφομοιώνει η σειριακή επικοινωνία χάνει τον real-time χαρακτήρα της.

Για να αντιμετωπιστεί αυτό το πρόβλημα αρχικά έγινε προσπάθεια να μειωθεί ο όγκος των δεδομένων που μεταβιβάζονται από τον υπολογιστή επεξεργασίας των EMGs. Έτσι ρυθμίστηκε νέο βήμα επεξεργασίας μεγαλύτερο της μονάδας με σκοπό να μειωθούν κατά το δυνατόν οι τιμές που επιστρέφονται ανά εκατοστάδα σημάτων. Ακολούθησε πλήθος δοκιμών στο σύστημα της προσομοίωσης έτσι ώστε να βρεθεί το μέγιστο βήμα επεξεργασίας για το οποίο μπορεί να επιτευχθεί έλεγχος. Υπενθυμίζουμε σε αυτό το σημείο ότι εάν το βήμα είναι πολύ μεγάλο δεν υπάρχει ικανοποιητικός έλεγχος (Κεφ. 5.2.2). Κατάλληλο βήμα επεξεργασίας ώστε να επιτυγχάνεται έλεγχος μετά τις δοκιμές κρίθηκε το βήμα 20.

Εκτός του όγκου των δεδομένων, έγινε προσπάθεια να αλλάξουν οι παράμετροι χρονοισμού της κάρτας ServoToGo, ώστε να καθυστερήσει η συχνότητα δειγματοληψίας της κάρτας. Δοκιμάστηκαν και εδώ πλήθος τιμών, ακόμα και στο κατώτερο δυνατό όριο ταχύτητας που επιτρέπει η κάρτα.

Επιπλέον απενεργοποιήθηκαν τα ενσωματωμένα alarm που διαθέτει η κάρτα σε περίπτωση ετεροχρονισμού.

Σε όλες τις ανωτέρω περιπτώσεις το πρόγραμμα ελέγχου του εξωσκελετικού μηχανισμού έχανε την επικοινωνία με την κάρτα ServoToGo με αποτέλεσμα την ακαθόριστη κίνηση του βραχίονα.

Μετά από εξέταση όλων των παραπάνω λύσεων χωρίς κανένα αποτέλεσμα οδηγούμαστε στο συμπέρασμα ότι ο εξοπλισμός που χρησιμοποιείται για τη διασύνδεση της εξωσκελετικής διάταξης με τα ηλεκτρομυογραφικά σήματα δεν διαθέτει την απαιτούμενη υπολογιστική ισχύ ώστε να ανταποκριθεί κατάλληλα.

Για την ολοκλήρωση της παρούσας εργασίας και την τελική αξιολόγησή της, είναι δυνατή η αξιοποίηση του δυναμικού μοντέλου της εξωσκελετικής διάταξης έτσι ώστε σε περιβάλλον προσομοίωσης στον υπολογιστή να εξαχθούν τα απαραίτητα συμπεράσματα της όλης μεθοδολογίας που αναπτύχθηκε.

ΚΕΦΑΛΑΙΟ 6^ο – ΑΞΙΟΛΟΓΗΣΗ ΑΠΟΤΕΛΕΣΜΑΤΩΝ & ΜΕΛΛΟΝΤΙΚΕΣ ΚΑΤΕΥΘΥΝΣΕΙΣ

6.1 Αξιολόγηση Αποτελεσμάτων

Εξαρχής, βασικός στόχος της παρούσας διπλωματικής αποτέλεσε η διασύνδεση της εξωσκελετικής διάταξης με ένα σύστημα ανάκτησης και επεξεργασίας ηλεκτρομυογραφικών σημάτων έτσι ώστε να είναι δυνατός ο έλεγχος του exoskeleton απ' ευθείας από τον μυ του χρήστη. Έχοντας δεδομένη την υποχωρητικότητα του βραχίονα έπρεπε να δημιουργηθεί το υποσύστημα λήψης και επεξεργασίας σήματος καθώς και το κομμάτι της ενίσχυσης από το υποσύστημα εξωσκελετικής διάταξης.

Στήθηκε επιτυχώς το υποσύστημα σημάτων με την ανάπτυξη μιας ενιαίας εφαρμογής η οποία συγκεντρώνει τα EMGs σε πραγματικό χρόνο, εφαρμόζει τα απαραίτητα φίλτρα και στη συνέχεια κανονικοποιεί τα αποτελέσματα των δύο φίλτρων. Αποτέλεσμα όλων αυτών είναι η εξαγωγή του συμπεράσματος της ύπαρξης ή μη βάρους στο χέρι του χρήστη. Το επόμενο πιο σημαντικό στάδιο είναι το γεγονός ότι η εφαρμογή, με την επιβολή κριτηρίου προσαρμοσμένου στο χρήστη έχει την ικανότητα να ανιχνεύσει την μυϊκή ένταση. Κατά τη διάρκεια της κίνησης στέλνει στο υποσύστημα του exoskeleton εντολή ενίσχυσης ανάλογη της μυϊκής έντασης.

Στο σημείο αυτό έγκειται και η διαφοροποίηση της παρούσας εργασίας σε σχέση με αντίστοιχες. Το σύστημα δηλαδή έχει την ικανότητα, όχι μόνο να αναγνωρίζει την ύπαρξη βάρους ή μη, αλλά κυρίως αυξανόμενου του μυϊκού φόρτου να στέλνει εντολή για μεγαλύτερη ενίσχυση. Η αύξηση του μυϊκού φόρτου μπορεί να προέρχεται βεβαίως είτε από αύξηση του βάρους είτε και από την παροδική κόπωση του χρήστη. Σε κάθε περίπτωση το υποσύστημα σημάτων θα στείλει εντολή για αυξημένη ενίσχυση και αυτό τελικώς είναι και το ζητούμενο. Δηλαδή, ανεξαρτήτως εάν το φορτίο προέρχεται από αύξηση βάρους ή από μυϊκό κάματο, ο χρήστης πρέπει ορθώς να νοιώθει ότι αποφορτίζεται αναλόγως από την διάταξη.

Αναφορικά με το υποσύστημα της εξωσκελετικής διάταξης αναπτύχθηκε η εφαρμογή του ελεγκτή ενίσχυσης εντός της οποίας υπάρχει ενσωματωμένη η

υποχωρητικότητα και η διαδικασία ανάγνωσης από τη σειριακή θύρα της τιμής που στέλνει το υποσύστημα σημάτων σαν εντολή ενίσχυσης. Όταν όμως γίνεται ανάγνωση από τη σειριακή θύρα αναλώνονται υπολογιστικοί πόροι του συστήματος οι οποίοι «εκλείπουν» από την επικοινωνία με την κάρτα ServoToGo. Αποτέλεσμα αυτού είναι να εγείρονται σφάλματα από την κάρτα και να σταματάει η ροή της ίδιας της εφαρμογής ελέγχου, σταματώντας τον ίδιο το έλεγχο της διάταξης.

Αρχικά για να αντιμετωπιστεί το γεγονός αυτό μειώθηκε κατά πολύ το βήμα επεξεργασίας των σημάτων έτσι ώστε να μειωθεί ο όγκος δεδομένων που ανά μονάδα χρόνου διακομίζονται μέσω της σειριακής. Επιπλέον μειώθηκε η συχνότητα χρονικών παλμών της κάρτας καθιστώντας την πιο αργή έτσι ώστε να προλαβαίνει ο υπολογιστής του exoskeleton να διαβάσει και την θύρα αλλά και την κάρτα. Τέλος επιχειρήθηκε να διακοπεί η εκπομπή alarm από την κάρτα μέσα από επιλογή που προσφέρεται στον χρονιστή Watchdog timer. Σε όλες τις περιπτώσεις η ανάγνωση της σειριακής θύρας αποσυντονίζει την επικοινωνία με την κάρτα ServoToGo.

Λύση στο πρόβλημα αυτό πιθανότατα θα έδινε η εγκατάσταση υπολογιστικού συστήματος με αυξημένες υπολογιστικές δυνατότητες ώστε να ανταπεξέρχεται στον ταυτόχρονο έλεγχο της κίνησης και τη διακομιδή δεδομένων.

Τέλος, ως άλλη εναλλακτική θα μπορούσε να προταθεί η κατασκευή ολοκληρωμένου ηλεκτρονικού κυκλώματος για την εκτέλεση επιλεγμένων εργασιών που τώρα ζητούνται από προσωπικούς υπολογιστές γενικής χρήσης έτσι ώστε να μειωθούν κατά πολύ οι χρόνοι απόκρισης. Βεβαίως στην περίπτωση αυτή δε θα πρέπει κανείς να αγνοήσει την σημαντική αύξηση του κόστους στην κατασκευή τέτοιων εξειδικευμένων κυκλωμάτων.

6.2 Μελλοντικές Κατευθύνσεις

Όπως έχουμε αναφέρει από την αρχή της παρούσας εργασίας, ο τομέας της προσαρμογής ηλεκτρομηχανικών συστημάτων στο ανθρώπινο σώμα για εξυπηρέτηση είτε καθημερινών αναγκών είτε αποκατάσταση αναπηριών προσελκύει μεγάλο ερευνητικό ενδιαφέρον. Αυτό από μόνο του φαίνεται να

υπόσχεται νέες κατακτήσεις στο μέλλον. Σε άμεση συνάρτηση με την παρούσα εργασία μια σημαντική βελτίωση της μεθοδολογίας θα ήταν η δημιουργία ενός συστήματος το οποίο να μπορούσε αυτόματα να συνεργαστεί με οποιονδήποτε χρήστη. Στη παρούσα φάση πριν ξεκινήσει η διάταξη να δουλεύει θα πρέπει το υποσύστημα λήψης σημάτων να εκπαιδευτεί πάνω στο συγκεκριμένο χρήστη. Και αυτό γιατί η μυϊκή δραστηριότητα του κάθε ανθρώπου είναι διαφορετική ακόμα και για το ίδιο βάρος στην ίδια κίνηση. Συνεπώς το σύστημα στη φάση εκπαίδευσης θα πρέπει να αναγνωρίσει πιο είναι εκείνο το σήμα αναφοράς για το συγκεκριμένο χρήστη.

Ακόμα, μια σημαντική εξέλιξη στον τομέα θα μπορούσε να αποτελέσει η άντληση σημάτων απευθείας από τον εγκέφαλο για τον έλεγχο κίνησης εξωσκελετικών μηχανισμών. Κάτι τέτοιο θα έδινε σπουδαία ώθηση σε περιπτώσεις αποκατάστασης και υποβοήθησης ατόμων με σοβαρές ή και καθολικές σωματικές αναπηρίες. Βεβαίως τέτοιες ερευνητικές απόπειρες έχουν ήδη ξεκινήσει και φαίνονται να έχουν ισχυρή δυναμική. Μένει ακόμα να φτάσουν σε τέτοιο σημείο ωρίμανσης ώστε να μπορέσουν να αξιοποιηθούν περαιτέρω.

ΠΑΡΑΡΤΗΜΑ Α: Οδηγίες Χρήσης

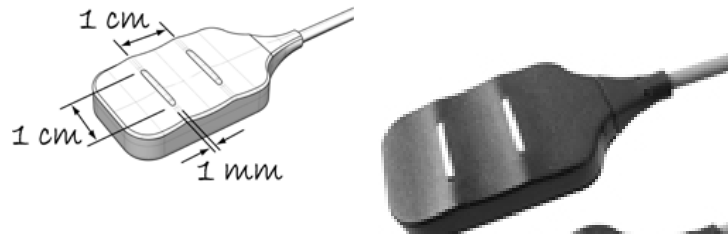
A.1 Υποσύστημα Ηλεκτρομυογραφικών Σημάτων

Για το υποσύστημα των EMGs χρησιμοποιείται ο ακόλουθος εξοπλισμός:

- ✓ Η/Υ λήψης ηλεκτρομυογραφικών σημάτων σε πραγματικό χρόνο
- ✓ Βιοενισχυτής για τη λήψη EMGs (Εικόνα Α.1, [3])
- ✓ Ενεργά ηλεκτρόδια λήψης EMGs (Εικόνα Α.2, [3])
- ✓ Module εισόδου οχτώ καναλιών (Εικόνα Α.3, [3])
- ✓ Ηλεκτρόδιο γείωσης



Εικόνα Α.1 ^[3]



Εικόνα A.2 ^[3]



Εικόνα A.3 ^[3]

Ο τρόπος σύνδεσης των πιο πάνω στοιχείων είναι αυτός που περιγράφεται στη συνέχεια.

Συνδέουμε το ηλεκτρόδιο λήψης και γείωσης με την κεντρική κονσόλα, μέσω του module εισόδου όπως αυτό φαίνεται σε λεπτομέρεια στην Εικόνα A.4. Συγκεκριμένα, το ηλεκτρόδιο λήψης θα πρέπει να συνδεθεί στην είσοδο 1 ενώ το ηλεκτρόδιο της γείωσης στην είσοδο ref. Στη συνέχεια συνδέουμε την κονσόλα με τον Η/Υ και με το ρεύμα μέσω μετασχηματιστή. Ρυθμίζουμε την κονσόλα λήψης στα 100Hz και τοποθετούμε τα ηλεκτρόδια στο χρήστη. Το ηλεκτρόδιο λήψης τοποθετείται στο κέντρο του δικέφαλου και το ηλεκτρόδιο για τη γείωσης σε κάποιο σημείο του σώματος του το οποίο θα είναι σταθερό κατά τη διάρκεια του πειράματος όπως είναι το πάνω μέρος του της παλάμης του άλλου χεριού του χρήστη.



Εικόνα A.4 ^[3]

A.2 Υποσύστημα Εξωσκελετικής Διάταξης

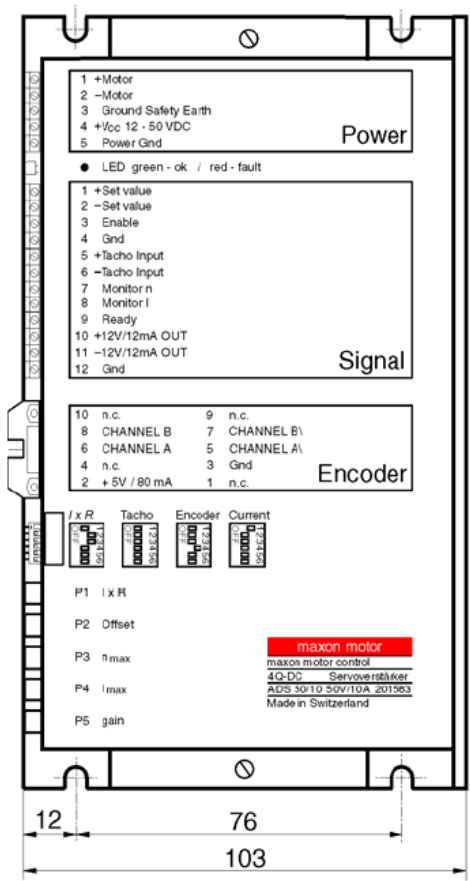
Το υποσύστημα του exoskeleton αποτελείται από τον ακόλουθο εξοπλισμό:

- ✓ Η/Υ για τον έλεγχο του εξωσκελετικού μηχανισμού
- ✓ Κινητήρας DC μόνιμου μαγνήτη με μειωτήρα στροφών και παλμογεννήτρια (Εικόνα 2.3, [3])
- ✓ Ράβδος εξωσκελετικής υποστήριξης (Εικόνα 2.3, φωτογραφία)
- ✓ Κάρτα ελέγχου κινητήρων (ServoToGo) (Εικόνα 2.2, [8])
- ✓ Ενισχυτής με δυνατότητα ελέγχου σε τάση και σε ρεύμα (Εικόνα 2.3, φωτογραφία)
- ✓ Ξύλινη βάση στήριξης της διάταξης
- ✓ Τροφοδοτικό συνεχούς τάσης
- ✓ Κλωβός μείωσης ηλεκτρομαγνητικής ακτινοβολίας (Εικόνα 2.4, φωτογραφία)

Η σύνδεση των πιο πάνω γίνεται με τον ακόλουθο τρόπο.

Αρχικά τοποθετούμε τον μειωτήρα στην ελεύθερη άτρακτο του κινητήρα με χρήση σφήνας η οποία είναι ήδη τοποθετημένη στο σφηναύλακα του κινητήρα. Στερεώνουμε με κοχλίες το σύστημα κινητήρα-μειωτήρα πάνω στην ξύλινη βάση μέσα στον κλωβό Faraday. Τοποθετούμε τη ράβδο του exoskeleton στο μειωτήρα κάνοντας χρήση κοχλιών, χωρίς όμως να τους σφίξουμε πολύ. Στη συνέχεια συνδέουμε τους ενισχυτές με την κάρτα ServoToGo και τους κινητήρες και ρυθμίζουμε τους ενισχυτές σύμφωνα με τις ανάγκες της εφαρμογής. Η κάρτα ServoToGo έχει τέσσερις connectors, κάθε ένας από του οποίους αποτελείται από πενήντα pins. Η ανάλυση όλων των pins δίνεται στις Εικόνες A.6 και A.7. Αντίστοιχα ο ενισχυτής (Εικόνα A.5) είναι χωρισμένος σε τρία μέρη :

- Power : στο τμήμα αυτό συνδέεται ο κινητήρας και το τροφοδοτικό
- Signal : στο τμήμα αυτό συνδέεται ο ενισχυτής με την κάρτα
- Encoder : στο τμήμα αυτό μπορούμε να συνδέσουμε την παλμογεννήτρια



Εικόνα Α.5 ^[12]

8 Channel Encoder Input and Analog Output Connectors

Connector P3, Motion I/O Axis 0-3				Connector P4, Motion I/O Axis 4-7			
Pin	Name	Pin	Name	Pin	Name	Pin	Name
1	Analog Gnd	2	DAC 0	1	Analog Gnd	2	DAC 4
3	Analog Gnd	4	Analog Gnd	3	Analog Gnd	4	Analog Gnd
5	DAC 2	6	Analog Gnd	5	DAC 6	6	Analog Gnd
7	Analog Gnd	8	DAC 1	7	Analog Gnd	8	DAC 5
9	Analog Gnd	10	Analog Gnd	9	Analog Gnd	10	Analog Gnd
11	DAC 3	12	Analog Gnd	11	DAC 7	12	Analog Gnd
13	Gnd	14	A 0 +	13	Gnd	14	A 4 +
15	A 0 -	16	Gnd	15	A 4 -	16	Gnd
17	B 0 +	18	B 0 -	17	B 4 +	18	B 4 -
19	Gnd	20	I 0 +	19	Gnd	20	I 4 +
21	I 0 -	22	Gnd	21	I 4 -	22	Gnd
23	A 1 +	24	A 1 -	23	A 5 +	24	A 5 -
25	Gnd	26	B 1 +	25	Gnd	26	B 5 +
27	B 1 -	28	Gnd	27	B 5 -	28	Gnd
29	I 1 +	30	I 1 -	29	I 5 +	30	I 5 -
31	Gnd	32	A 2 +	31	Gnd	32	A 6 +
33	A 2 -	34	Gnd	33	A 6 -	34	Gnd
35	B 2 +	36	B 2 -	35	B 6 +	36	B 6 -
37	Gnd	38	I 2 +	37	Gnd	38	I 6 +
39	I 2 -	40	Gnd	39	I 6 -	40	Gnd
41	A 3 +	42	A 3 -	41	A 7 +	42	A 7 -
43	Gnd	44	B 3 +	43	Gnd	44	B 7 +
45	B 3 -	46	Gnd	45	B 7 -	46	Gnd
47	I 3 +	48	I 3 -	47	I 7 +	48	I 7 -
49	+5	50	+5	49	+5	50	+5

Εικόνα Α.6 ^[8]

32 Bit Digital I/O, 8 Channel Analog Input and Sign Bit Output Connectors

Connector P1, Digital I/O				Connector P2, Analog & Digital I/O			
Pin	Name	Pin	Name	Pin	Name	Pin	Name
1	Opto-23, C7	2	Gnd	1	ADC Chan 0	2	Analog Gnd
3	Opto-22, C6	4	Gnd	3	ADC Chan 1	4	Analog Gnd
5	Opto-21, C5	6	Gnd	5	ADC Chan 2	6	Analog Gnd
7	Opto-20, C4	8	Gnd	7	ADC Chan 3	8	Analog Gnd
9	Opto-19, C3	10	Gnd	9	ADC Chan 4	10	Analog Gnd
11	Opto-18, C2	12	Gnd	11	ADC Chan 5	12	Analog Gnd
13	Opto-17, C1	14	Gnd	13	ADC Chan 6	14	Analog Gnd
15	Opto-16, C0	16	Gnd	15	ADC Chan 7	16	Analog Gnd
17	Opto-15, B7	18	Gnd	17	Opto-15, D7	18	Gnd
19	Opto-14, B6	20	Gnd	19	Opto-14, D6	20	Gnd
21	Opto-13, B5	22	Gnd	21	Opto-13, D5	22	Gnd
23	Opto-12, B4	24	Gnd	23	Opto-12, D4	24	Gnd
25	Opto-11, B3	26	Gnd	25	Opto-11, D3	26	Gnd
27	Opto-10, B2	28	Gnd	27	Opto-10, D2	28	Gnd
29	Opto-9, B1	30	Gnd	29	Opto-9, D1	30	Gnd
31	Opto-8, B0	32	Gnd	31	Opto-8, D0	32	Gnd
33	Opto-7, A7	34	Gnd	33	IN2	34	Gnd
35	Opto-6, A6	36	Gnd	35	IN1	36	Gnd
37	Opto-5, A5	38	Gnd	37	IN0	38	Gnd
39	Opto-4, A4	40	Gnd	39	EXLATCH	40	Gnd
41	Opto-3, A3	42	Gnd	41	T2GATE	42	Gnd
43	Opto-2, A2	44	Gnd	43	TA2	44	Gnd
45	Opto-1, A1	46	Gnd	45	/WATCHDOG	46	Gnd
47	Opto-0, A0	48	Gnd	47	NC	48	Gnd
49	+5V	50	Gnd	49	+5V	50	Gnd

Connector P1 can be directly connected to a 24 channel I/O card, such as made by OPTO-22, and other companies.

Εικόνα Α.7 ^[8]

Τέλος γίνεται η σύνδεση των δύο Η/Υ μέσω της σειριακής θύρας RS232 με τη χρήση του κατάλληλου καλωδίου σειριακής επικοινωνίας.

ΠΑΡΑΡΤΗΜΑ Β: Επεξήγηση Προγραμμάτων

Στην ενότητα αυτή θα αναλύσουμε τα προγράμματα που χρησιμοποιήθηκαν για την λήψη και επεξεργασία των EMGs και για την προσομοίωση και τον έλεγχο του συστήματος. Επιγραμματικά τα προγράμματα που χρησιμοποιούνται είναι τα εξής:

- I. Πρόγραμμα επεξεργασίας ηλεκτρομυογραφικού σήματος. Ο κώδικας είναι γραμμένος σε Matlab και η επεξεργασία γίνεται με χρήση αποθηκευμένων σημάτων.
- II. Πρόγραμμα προσομοίωσης σε περιβάλλον Matlab.
- III. Πρόγραμμα λήψης και επεξεργασίας σήματος σε πραγματικό χρόνο, γραμμένο σε κώδικα C++.
- IV. Πρόγραμμα ελεγκτή σε περιβάλλον C++.

Ακολουθεί ο κώδικας και η ανάλυση του κάθε προγράμματος.

I. Πρόγραμμα επεξεργασίας EMGs σε περιβάλλον Matlab.

Αυτό το πρόγραμμα είναι πολύ χρήσιμο διότι αποτελεί το στάδιο εκπαίδευσης του συστήματος για τον κάθε χρήστη. Εισάγοντας επαρκή αριθμό σημάτων για συγκεκριμένο χρήστη μπορούμε να εξάγουμε όλες τις απαραίτητες πληροφορίες τις οποίες θα εισάγουμε στα υπόλοιπα προγράμματα για τον κατάλληλο έλεγχο του εξωσκελετικού μηχανισμού.

```
clear all  
  
n = 3;  
  
batch = 100;  
  
step = 10;  
  
  
for i = 1:n
```

```

        D{i} = load(sprintf('data_%i.txt',i));
    end

    for i = 1:n
        E{i} = load(sprintf('data_%i_w.txt',i));
    end

    for i = 1:n
        F{i} = load(sprintf('data_%i_ww.txt',i));
    end

    A = cell2mat(D');
    B = cell2mat(E');
    C = cell2mat(F');
    [N M]= size(A);

```

Αρχικά καθαρίζει τη μνήμη από άλλες παραμέτρους που τυχόν να υπάρχουν και θέτει κάποιες βασικές παραμέτρους του προγράμματος. Κατά τη λήψη σημάτων σε πραγματικό χρόνο δημιουργείται ένα αρχείο με 30000 σήματα. Η παράμετρος n δηλώνει πόσες ομάδες των 30000 σημείων θα ληφθούν υπόψη κατά την επεξεργασία. Αυξάνοντας τα δείγματα αυξάνεται και η αντικειμενικότητα των αποτελεσμάτων. Ως `batch` ορίζουμε το βήμα δειγματοληψίας, ενώ ως `step` το βήμα επεξεργασίας, όπως αυτά έχουν ορισθεί στο κεφάλαιο 3.2.2. Εισάγουμε τα αρχεία με τις τιμές των σημάτων και δημιουργούμε τρεις πίνακες. Τα αρχεία που εισάγουμε για την περίπτωση που δεν υπάρχει μυϊκή καταπόνηση έχουν τη μορφή `data_i.txt`, όπου i ο αύξων αριθμός του αρχείου που καλούμε και αποθηκεύονται στον πίνακα `A`. Όταν υπάρχει μυϊκή καταπόνηση 1kg καλούμε τα αρχεία `data_i_w.txt` και για μυϊκή καταπόνηση 2kg τα αρχεία `data_i_ww.txt`. Τα δεδομένα των αρχείων εισάγονται στους πίνακες `B` και `C` αντίστοιχα. Τέλος ορίζουμε ως N των αριθμό των γραμμών του πίνακα `A` (είναι ο ίδιος αριθμός γραμμών στους πίνακες `A`, `B` και `C`).

```

loop=0;

for j=1: step:(N-batch)
loop= loop+1;
WLa(loop,1)= WLBRAKE (A(j:j+batch-1),batch);
WLb(loop,1)= WLBRAKE (B(j:j+batch-1),batch);
WLC(loop,1)= WLBRAKE (C(j:j+batch-1),batch);
Va(loop,1)= VARCEBRAKE (A(j:j+batch-1),batch);
Vb(loop,1)= VARCEBRAKE (B(j:j+batch-1),batch);
Vc(loop,1)= VARCEBRAKE (C(j:j+batch-1),batch);
end

```

Μηδενίζουμε την παράμετρο `loop`, η οποία καθορίζει τον αριθμό των κύκλων εργασιών που θα γίνει στον βρόγχο `for` που ακολουθεί. Ο βρόγχος `for` δημιουργεί έξι πίνακες δύο για κάθε έναν από τους πίνακες `A`, `B`, `C`. Οι συναρτήσεις `WLBRAKE` και `VARCEBRAKE` είναι τα φίλτρα που προκύπτουν από τη χρήση των εξισώσεων `Wavelength` και `Variance` αντίστοιχα. Έτσι οι πίνακες `WLa`, `WLb` και `WLC` έχουν τα αποτελέσματα των σημάτων των πινάκων `A`, `B` και `C` αντίστοιχα μετά το φιλτράρισμά τους από την παράμετρο `Wavelength`. Σε αντιστοιχία αυτού οι πίνακες `Va`, `Vb` και `Vc` περιέχουν τα αποτελέσματα των σημάτων των πινάκων `A`, `B` και `C` αντίστοιχα μετά το φιλτράρισμά τους από την παράμετρο `Variance`. Να σημειωθεί ότι το μέγεθος των πινάκων `A`, `B` και `C` είναι πάντα πολλαπλάσιο του 30000 ενώ το μέγεθος των πινάκων `WLa`, `WLb`, `WLC`, `Va`, `Vb` και `Vc` ορίζεται από το μέγεθος των πινάκων `A`, `B` και `C` διαιρούμενο δια το `step`. Δηλαδή αν ο πίνακας `A` έχει 90000 στοιχεία για `step = 10` οι πίνακες `WLa` και `Va` έχουν 9000 στοιχεία.

```

maxiWLa= max(abs(WLa));
maxiWLb= max(abs(WLb));
maxiWLC= max(abs(WLC));
maxiWL= max(maxiWLb,maxiWLC)

```

```

maxiVa= max(abs(Va));
maxiVb= max(abs(Vb));
maxiVc= max(abs(Vc));
maxiV= max(maxiVb,maxiVc)

nVa= Va/maxiV;
nWLa= WLa/maxiWL;

nVb= Vb/maxiV;
nWLB= WLb/maxiWL;

nVc= Vc/maxiV;
nWLC= WLC/maxiWL;

EMG= (nVa+nWLa);
EMG_w= (nVb+nWLB);
EMG_ww= (nVc+nWLC);

minus = ones(size(EMG))*max (EMG);
EMG_e= exp (EMG -minus);
EMG_e_w= exp (EMG_w -minus);
EMG_e_ww= exp (EMG_ww -minus);

maxEMG= max (EMG_e)
meanEMG= mean(EMG_e)
maxEMG_w= max (EMG_e_w)
meanEMG_w= mean(EMG_e_w)

```



```
maxEMG_ww= max (EMG_e_ww)
meanEMG_ww= mean(EMG_e_ww)

plot (EMG_e,'b-'), hold on
plot (EMG_e_w,'g-'), hold on
plot (EMG_e_ww,'m-')
```

Ακολουθεί το στάδιο της κανονικοποίησης. Υπολογίζουμε τη μέγιστη κατά απόλυτο τιμή των πινάκων WLa , WLb και WLC και διαιρούμε όλα τα στοιχεία των πινάκων με αυτή την τιμή. Ακολουθούμε την ίδια διαδικασία για τους πίνακες Va , Vb και Vc . Έτσι ο μέγιστος αριθμός στους πίνακες $nWLa$, $nWLb$, $nWLC$, nVa , nVb , nVc είναι η μονάδα και μπορούμε να προσθέσουμε τα αποτελέσματα των φίλτρων ώστε να έχουμε πάλι 3 πίνακες που ο κάθε ένας τους θα αντιπροσωπεύει τη μη ύπαρξη βάρους (EMG), την ύπαρξη βάρους 1kg (EMG_w) και την ύπαρξη βάρους 2kg (EMG_ww). Οι πίνακες περιέχουν τα φιλτραρισμένα ηλεκτρομυογραφικά σήματα. Για να είναι πιο σαφή τα αποτελέσματα και πιο εμφανής η διαφορά ανάμεσα στα σήματα με μυϊκή καταπόνηση και σε αυτά χωρίς, ακολουθεί περαιτέρω επεξεργασία. Αρχικά αφαιρούμε από τα περιεχόμενα των πινάκων την μέγιστη τιμή του πίνακα χωρίς μυϊκή καταπόνηση [$\max(EMG)$] και στη συνέχεια υψώνουμε τον αριθμό e στο κάθε ένα από τα αποτελέσματα. Έτσι δημιουργούνται οι τρεις νέοι πίνακες EMG_e , EMG_e_w και EMG_e_ww για την περίπτωση της κίνησης χωρίς βάρος, με βάρος 1kg και 2kg αντίστοιχα. Να σημειωθεί ότι από αυτό το πρόγραμμα εξάγουμε και την τιμή \maxEMG , πάνω από την οποία θεωρούμε ότι υπάρχει μυϊκή καταπόνηση. Παρατηρώντας είτε τις γραφικές παραστάσεις που προκύπτουν είτε τις μέγιστες και μέσες τιμές των πινάκων βλέπουμε ότι είναι τόσο μεγάλη η διαφορά της τιμής αυτής σε σύγκριση με αυτές κατά τις οποίες υπάρχει μυϊκή καταπόνηση που η χρήση αυτής ως τιμής κριτηρίου είναι ασφαλής.

II. Πρόγραμμα προσομοίωσης σε περιβάλλον Matlab.

Η σπουδαιότητα του συγκεκριμένου προγράμματος έγκειται στο γεγονός ότι μπορούμε να υπολογίσουμε θεωρητικά τα κατάλληλα κέρδη για τον έλεγχο και να έχουμε μία πρώτη εικόνα για τις τιμές αυτών πριν δοκιμαστούν σε έλεγχο σε πραγματικό χρόνο. Θα ήταν πολύ επικίνδυνο να δοκιμάσουμε τυχαία κέρδη απευθείας στον σύστημα εξωσκελετικός μηχανισμός – χέρι διότι θα είχε απρόβλεπτες επιπτώσεις και πιθανόν να προκαλούνταν κάποιος τραυματισμός ή καταστροφή κάποιου μηχανισμού. Γι' αυτό και το συγκεκριμένο πρόγραμμα χρησιμοποιείται για κάθε χρήστη ώστε να γίνει μία πρώτη εκτίμηση των τιμών των κερδών. Το πρόγραμμα αυτό αποτελείται από το βασικό πρόγραμμα και τις συναρτήσεις που καλούνται μέσα σε αυτό. Αρχικά θα εξετάσουμε όλες τις συναρτήσεις και στο τέλος το κεντρικό πρόγραμμα.

i. Δυναμικές παράμετροι του συστήματος

```
%dynamic_parameters
% This program is used to import the dynamic Parameters of the plant
% Giving values to the parameters....
%Variable      Value      Description

kr   =   67   ;      % exoskeleton's gear ratio
kt   =   0.11222;    % torque-current gradient
EMGref =   1.26;
EMGmax =   4.7;
Tb    =   6; %2*10*0.3 = 6 N*m
%Tb    =   3; %1*10*0.3 = 3 N*m
EMGgraph = load ('dokimi.txt');
```

Σε αυτή τη συνάρτηση εισάγονται οι δυναμικές παράμετροι του συστήματος. Όπου k_r είναι η σταθερά μετατροπής ρεύματος κινητήρα σε

ροπή εξόδου κινητήρα και kt η σταθερά μείωσης του μειωτήρα στροφών. Η τιμή EMG_{ref} είναι η τιμή του κριτηρίου, δηλαδή η τιμή $maxEMG$ του προγράμματος επεξεργασίας EMGs που περιγράψαμε πιο πάνω. Ο πίνακας EMG_{graph} περιέχει τις τιμές του επεξεργασμένου σήματος με μυϊκή καταπόνηση όπως αυτά προκύπτουν από το πρόγραμμα που εξετάσαμε πιο πάνω. Η τιμή EMG_{max} είναι η μέγιστη τιμή του πίνακα αυτού. Το βάρος το οποίο σηκώνει ο χρήστης εισάγεται μέσω της παραμέτρου T_b .

ii. Προσομοίωση χεριού

```
function EMGreal = Hand(Texos,j);  
  
%simulator_Hand/EMGs  
dynamic_parameters;  
coef=(EMGmax/EMGref - 1)/Tb;  
  
EMGreal = EMGgraph(j) * (1+coef*(Tb-Texos))/(1+coef*Tb);
```

Σε αυτή τη συνάρτηση γίνεται η υλοποίηση της προσομοίωσης του χεριού όπως αυτή περιγράφεται στο κεφάλαιο 4.

iii. Προσομοίωση εξωσκελετικής διάταξης

```
function T_exos = T_exoskeleton(iexos);  
  
%simulator_exoskeleton  
dynamic_parameters;  
T_exos=kt*kr*iexos;
```

Σε αυτή τη συνάρτηση γίνεται η υλοποίηση της προσομοίωσης της εξωσκελετικής διάταξης όπως επίσης περιγράφεται στο κεφάλαιο 4.

iv. *Ελεγκτής συστήματος*

```
function [iexos,err,sum1] = exoskeleton_controller(EMGreal,err,sum1);

% Current Controller Gains
dynamic_parameters;
Kp = 0.15;
Kd = 0.1/10000;
Ki = 400;

dt = 0.001;
Ts = 0.001;
error = EMGreal - EMGref;

% Exoskeleton Controller

sum1 = sum1 + (error + err)*0.5*0.001;
iexos = (Kp*error + Kd*(error - err)/dt + Ki*sum1);

err = error;
```

Η υπορουτίνα αυτή υλοποιεί προγραμματιστικά τον ελεγκτή του εξωσκελετικού μηχανισμού που αναφέρεται αναλυτικά στο κεφάλαιο 5. Παίρνει σαν ορίσματα εισόδου την τιμή του κριτηρίου και τις τιμές του πίνακα που προκύπτουν από την προσομοίωση του χεριού.

v. *Κεντρικό πρόγραμμα*

```
% main program of the simulator

% Clear working environment

clear all;
close all;
clc;

% Initial state of parameters and constants
Texos=0;
err=0;
sum1=0;
dynamic_parameters;

% Beginning Simulation

disp('Simulation has begun ! Please wait....');

% Control Loop

for j = 1 : length(EMGgraph)
    if j==1
        disp('1')
    end
    EMGreal=Hand(Texos,j);
    [iexos,err,sum1]=exoskeleton_controller(EMGreal,err,sum1);
    Texos=T_exoskeleton(iexos);
end
```

```

array1(j)=EMGreal;
arrayREF(j)=EMGref;
array2(j)=iexos+EMGref;
end

plot(array1,'b-');
hold on;
plot(arrayREF,'y');
hold on;
plot(EMGgraph,'m-');
%hold on;
%plot(array2,'r-');
title('EMGerror--EMGreal-EMGref');
xlabel('SAMPLES');
ylabel('EMGerror');

```

Αυτό είναι το βασικό πρόγραμμα της προσομοίωσης στο οποίο συνδέονται οι πιο πάνω υπορουτίνες προσομοιώνοντας τελικά την πειραματική μας διάταξη. Έτσι, δεδομένων του πραγματικού γραφήματος όταν το χέρι κινείται με βάρος, του βάρους που μεταφέρει και της τιμής που είναι το κριτήριο για το συγκεκριμένο χρήστη, υπολογίζεται το ρεύμα που πρέπει να δίνει ο κινητήρας του exoskeleton στο μηχανισμό για να ενισχύσει την κίνηση του χεριού ώστε ο χρήστης να μην αντιλαμβάνεται το πρόσθετο βάρος και το γράφημα το οποίο θα είχαμε μετά την ενίσχυση από το exoskeleton.

III. Πρόγραμμα λήψης και επεξεργασίας EMGs σε πραγματικό χρόνο.

Αυτό το πρόγραμμα χρησιμοποιείται αρχικά για τη λήψη και αποθήκευση των ηλεκτρομυογραφικών σημάτων σε αρχείο ώστε να μπορεί να γίνει η

επεξεργασία από τα δύο προγράμματα που αναφέρονται πιο πάνω. Στη συνέχεια και αφού έχουμε τις τιμές που χρειαζόμαστε από το πρόγραμμα επεξεργασίας EMGs σε περιβάλλον Matlab είναι εφικτή η επεξεργασία των EMGs σε πραγματικό χρόνο. Στο δεύτερο αυτό στάδιο γίνεται η λήψη EMGs, η επεξεργασία τους, η επιλογή του πότε έχουμε μυϊκή καταπόνηση του μυ και πότε όχι, η σύνδεση με τον Η/Υ του exoskeleton και τέλος η δυνατότητα αποστολής σε αυτόν μίας τιμής εφόσον υπάρχει μυϊκή καταπόνηση.

```
% EMG signal acquisition and data processing

#ifdef HAVE_CONFIG_H
#include <config.h>
#endif

#include <iostream>
#include <cstdlib>
#include <math.h>
#include "Isotrak.h"
#include "signal_acq.h"

//Serial port files
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
#include <errno.h> //xeirismos errors tou sistymatos
#include <termios.h> //terminal control

#include <fcntl.h> //orismos ton flags pou xrisimopoiountai sto 'open'
#include <sys/ioctl.h> //interface for passing requests to a device file
```

```

//Serial port variables
int fd = 0 ;      // File handler to the Serial Port
int BytesWritten = 0 ; // Bytesvoid Serial_Init(); Written to Serial Port

double dataAll[30000];
int iindex=0;
void Serial_Init();
void Serial_Write();
int Value;

using namespace std;

/* Functions*/
bool System_Initialization(int , float, int, speed_t, int isotrak_stations);
void Save2File(FILE*, double *, int, int);

double Criterion(int DataAllIndex);
double Varce(double *data);
double WaveLength(double *data);
void AppendData (double *newData);
FILE *fpww;

/* Global objects */
Signal_Acq signal_obj;

```

Στην αρχή του προγράμματος εισάγονται οι κατάλληλες βιβλιοθήκες της γλώσσας προγραμματισμού C++. Στη συνέχεια ορίζονται οι μεταβλητές της σειριακής σύνδεσης του ηλεκτρονικού υπολογιστή και το όρισμα των συναρτήσεων τοποθέτησης αρχικών τιμών της κάρτας και αποθήκευσης

των δεδομένων – ηλεκτρομυογραφικών σημάτων – που λαμβάνονται. Ακόμη ορίζονται οι συναρτήσεις επεξεργασίας των δεδομένων (Criterion, Varce, WaveLength, AppendData) και η μεταβλητή του αρχείου frww. Τέλος δίνεται και ο ορισμός του αντικειμένου που χρησιμοποιείται για την υλοποίηση της δειγματοληψίας.

```

int main(int argc, char *argv[])
{
    int global_counter = 0;
    bool init_flag = false;
    struct timeval start, end;

    /* Configuring EMG acquisition*/
    int n_chan = 1;
    float freq = 1000.0;
    int samples = 100;
    int n_times = 300; // (samples/freq)*n_times = ( ) sec
    int isotrak_stations = 2;
    double * data;
    data = (double *) malloc(n_chan*samples*sizeof(double));

    double ReturnValue;

    /* Configuring Isotrak - Selecting Baud Rate*/
    speed_t iso_baud_rate = B38400;

    /* System Initialization*/
    init_flag = System_Initialization(n_chan, freq, samples, iso_baud_rate,
isotrak_stations);
    if(!init_flag)
    {
        printf("Error at system initialization..Aborting!!\n");
        return -1;
    }
    FILE *fp;

```

```

char str[15];
fp = fopen("data.txt", "w");

FILE *fpw;
fpw = fopen("ReturnValues.txt", "w");
fpww = fopen("SumSampleNorm.txt", "w");

printf("hand success\n");

Serial_Init();

for ( int m = 0 ; m < n_times; m++ )
{
    gettimeofday(&end, NULL);
    end.tv_sec -= start.tv_sec;
    if (end.tv_usec < start.tv_usec) {
        end.tv_sec--;
        end.tv_usec+= 1000000;
    }

    end.tv_usec -= start.tv_usec;
    gettimeofday(&start, NULL);

    signal_obj.Signal_Acq_Get(data);
    Save2File(fp, data, samples, n_chan);

    AppendData (data);
    if (m != 0)
    {

```

```
        for (int mm = 0 ; mm < 100 ; mm+=50)
        {
            ReturnValue=Criterion((iindex-2*100)+mm);
            Value=int(ReturnValue);
            fprintf(fpw, "%f ", ReturnValue);
            fprintf(fpw, "\n");

            Serial_Write ();
        }
    }
}
fclose(fp);
fclose(fpw);
fclose(fpww);
printf("\n CLOSING Serial Port\n") ;
close(fd) ;

return EXIT_SUCCESS;
}
```

Αυτό είναι το κύριο μέρος του προγράμματος, όπου γίνεται η κλήση των συναρτήσεων από τη main. Σε αυτό το τμήμα του προγράμματος ορίζονται και οι συνθήκες δειγματοληψίας. Επιλέγουμε το κανάλι 1, επειδή όπως έχει ήδη αναφερθεί λαμβάνουμε EMGs από ένα μυ. Η συχνότητα δειγματοληψίας είναι 1000Hz, ο αριθμός των δειγμάτων που θα λαμβάνονται σε αυτή τη συχνότητα ορίζεται 100, ενώ η διαδικασία θα επαναληφθεί 300 φορές, έτσι ώστε σε με την ολοκλήρωση της να προκύπτουν 30000 δείγματα. Οι τιμές των σημάτων θα αποθηκεύονται στην μεταβλητή data οπότε στη συνέχεια γίνεται η ανάθεση μνήμης της μεταβλητής αυτής. Επιπλέον ορίζεται και η μεταβλητή ReturnValue στην οποία θα αποθηκεύεται η τελική τιμή που θα προκύπτει από την επεξεργασία και θα αποστέλλεται στον ελεγκτή του Exoskeleton.

Ακολούθως αρχικοποιείται το σύστημα δειγματοληψίας και ανιχνεύονται τυχόν σφάλματα που θα προκύψουν. Ορίζονται τα κατάλληλα αρχεία προσωρινής αποθήκευσης των τιμών που προκύπτουν ενδιάμεσα κατά την επεξεργασία και γίνεται η έναρξη της σειριακής σύνδεσης. Στο σημείο αυτό ξεκινάει σε πραγματικό χρόνο ο βρόγχος της ανάκτησης σημάτων από το μν, της επεξεργασίας τους, και της αποστολής της τελικής τιμής στη σειριακή θύρα

Εντός του βρόγχου αυτού, αρχικά λαμβάνονται τα EMGs από το μν μέσω των ενσωματωμένων συναρτήσεων που μας παρέχει η οντότητα `signal_obj`. Τα δεδομένα που ανακτώνται δηλώνονται κάθε φορά στη μεταβλητή `data` και αποθηκεύονται στο αρχείο με δείκτη `fp`. Επειδή θα ακολουθήσει η επεξεργασία των σημάτων αυτών, χρησιμοποιείται η μεταβλητή `AppendData` έτσι ώστε οι τελευταίες τιμές που λήφθηκαν από το μν να προσαρτηθούν στις τιμές που έχουν ληφθεί μέχρι τώρα κατά τη διάρκεια των προηγούμενων επαναλήψεων του συγκεκριμένου βρόγχου. Η επεξεργασία ξεκινάει στην δεύτερη επανάληψη `< if (m != 0) >`, και υλοποιείται ουσιαστικά από τη συνάρτηση `Criterion`, η ανάπτυξη της οποίας ακολουθεί στη συνέχεια. Το αποτέλεσμα της επεξεργασίας αποθηκεύεται στη μεταβλητή `Value` η οποία μέσω της υπορουτίνας `Serial_Write` αποστέλλεται στη σειριακή θύρα.

```
bool System_Initialization( int n_chan, float freq, int samples, speed_t
iso_baud_rate, int num)
{
    /* Signal Acquisition Initialization*/
    signal_obj.Signal_Acq_Init(n_chan, freq, samples);

    return true;
}

void Save2File(FILE* fp, double * data, int samples, int n_chan)
{
    for ( int rr = 0; rr < samples ; rr++ )
    {
```

```

    for ( int kk = 0; kk < n_chan; kk++ )
        fprintf(fp, "%f ", data[rr*n_chan+kk]);
    fprintf(fp, "\n");
}
}

```

Σε αυτό το κομμάτι αρχικοποιούμε την κάρτα δειγματοληψίας και αποθηκεύονται τα δεδομένα που λαμβάνει ο Η/Υ σε αρχείο .txt.

```

double Criterion(int DataAllIndex)
/*Varce & WaveLength*/
{
    double VSample;
    double WLSample;

    double VMax=0.37;
    double WLMax=33.00;

    double SumSampleNorm;
    double ControlValue_1;
    double ControlValue;

    double dblCriterion=50*1.26;
    double minus=0.4444;

    VSample = Varce(dataAll+DataAllIndex);
    WLSample = WaveLength(dataAll+DataAllIndex);
}

```

```

/*Criterion*/

SumSampleNorm=(VSample/VMax) +(WLSample/WLMax)-minus;

fprintf (fpww,"%f",SumSampleNorm);
fprintf(fpww, "\n");

ControlValue_1=50*exp(SumSampleNorm);

if (ControlValue_1 > dblCriterion)
ControlValue=ControlValue_1;
else
ControlValue=0;

if (ControlValue==0)
    printf("den exw varos\n");
else
    printf("exw varos\n");

return ControlValue;
}

```

Σε αυτό το σημείο είναι η συνάρτηση Criterion που καλείται από τη main. Αρχικά ορίζονται οι απαραίτητες παράμετροι, για κάποιες από τις οποίες οι τιμές προκύπτουν από το πρόγραμμα επεξεργασίας EMGs σε περιβάλλον Matlab που έχουμε περιγράψει πιο πάνω. Οι παράμετροι VMax και WLMax είναι οι μέγιστες τιμές των πινάκων maxV και maxWL αντίστοιχα του προγράμματος της Matlab που προαναφέρθηκε. Η παράμετρος dblCriterion είναι η τιμή του κριτηρίου πάνω από την οποία θεωρούμε ότι υπάρχει μυϊκή καταπόνηση. Καλούνται οι συναρτήσεις Varce και WaveLength –που περιγράφονται πιο κάτω– και ακολουθεί η

κανονικοποίηση όπως την έχουμε περιγράψει και πιο πάνω. Όταν η τιμή που προκύπτει μετά την επεξεργασία είναι μικρότερη από το κριτήριο η τιμή αυτή μηδενίζεται και τυπώνεται στην οθόνη η ένδειξη “den exw varos”. Σε αντίθετη περίπτωση, όταν δηλαδή η τιμή του κριτηρίου είναι μικρότερη της τιμής που προκύπτει, τυπώνεται στην οθόνη η ένδειξη “exw varos”.

```
/*Varce*/
double Varce(double *data)
{
    int j;
    int k;
    int N=100;
    double Data2[30000];
    double totData2;
    double dblVarce;

    for (j=0;j<=N-1;j++)
    {
        Data2[j]= *(data+j) * *(data+j));
    }
    totData2=0;
    for (k=0; k<=N-1; k++)
    {
        totData2 = totData2 + Data2[k];
    }
    dblVarce =totData2/(N-1);
    return dblVarce;
}
/*WaveLength*/
double WaveLength(double *data)
```



```

{
    double DiffData;
    double totDiffData;
    double dblWaveLength;
    int N=100;
    int k;

    totDiffData=0;
    DiffData=0;
    for (k=0; k<=N-1; k++)
    {
        if (k != 0)
            DiffData = fabs((*data+k)- (*data+k-1));
            totDiffData = totDiffData + DiffData;
    }
    dblWaveLength=totDiffData;
    return dblWaveLength;
}

```

Σε αυτό το τμήμα του προγράμματος τα δεδομένα που λαμβάνονται επεξεργάζονται σύμφωνα με το φίλτρο Variance και στη συνέχεια με το φίλτρο Wavelength, όπως τα έχουμε περιγράψει στη θεωρία επεξεργασίας ηλεκτρομυογραφικών σημάτων. Οι δύο αυτές συναρτήσεις καλούνται από την συνάρτηση Criterion.

```

void Serial_Init () /* Opens and sets Serial_Port */
{
    struct termios options ;

    /* Clear struct termios & input buffer */

```

```

memset(&options, 0, sizeof(options)) ;

/* Open Serial Port */
fd = open( "/dev/ttyS0", O_RDWR | O_NOCTTY | O_NDELAY) ;
if (fd < 0)
    printf("COULD NOT OPEN SERIAL PORT\n") ;
else
{
    fcntl( fd, F_SETFL, 0) ;
    printf("SERIAL PORT OPENED SUCCESSFULLY\n") ;
}

/* Get Settings of Serial Port */
tcgetattr( fd, &options) ;

/*Set the baud rates to 9600 */
cfsetispeed( &options, B9600) ;
cfsetospeed( &options, B9600) ;

/* Setting parity (8N1) */
options.c_cflag &= ~PARENB ; // No parity
options.c_cflag &= ~CSIZE ; // 8 bits
options.c_cflag |= CS8 ;

/* Setting Flow Control and Echoing */
options.c_iflag&=~IXON;
options.c_iflag&=~IXOFF;
options.c_iflag &= ~(ICANON | ECHO | ECHOE | ECHONL| ISIG ) ;
options.c_oflag&=~OPOST;

```

```

/* Set new Settings of Serial Port */
tcsetattr(fd,TCSAFLUSH,&options) ;

return;
}
void Serial_Write () /* Writing data to the Serial port*/
{
tcflush(fd,TCOFLUSH);
BytesWritten = write(fd, &Value, 1);
tcdrain(fd);
return;
}

```

Στο κομμάτι αυτό του προγράμματος είναι οι συναρτήσεις serial init και write οι οποίες όταν καλούνται συνδέουν προγραμματιστικά τους δύο υπολογιστές. Αποτελούν κοινή πρακτική σε εφαρμογές επικοινωνίας με σειριακή θύρα και για το λόγο αυτό δεν κρίνεται απαραίτητη η εκτενής ανάλυση της λειτουργίας τους. Στην ουσία αφού καθοριστούν οι παράμετροι με τις οποίες θα λειτουργήσει η σειριακή, δίνεται η εντολή για την έναρξη της θύρας και ενημερωνόμαστε ότι αυτό συνέβη με επιτυχία. Στη συνέχεια με την συνάρτηση Serial_Write αποθηκεύεται το περιεχόμενο της μεταβλητής value στο buffer εξόδου της σειριακής θύρας και στέλνονται τα επιθυμητά δεδομένα από τον Η/Υ που λαμβάνει και επεξεργάζεται τα EMGs στον Η/Υ που είναι συνδεδεμένος με τον εξωσκελετικό μηχανισμό.

```

void AppendData (double *newData)
{
    int i;
    int N=100;
    for (i=0;i<=N-1;i++)

```

```
{
    dataAll[iindex+i]=*(newData+i);
}
iindex=iindex+N;
return;
}
```

Όπως έχει ήδη αναφερθεί, για να γίνει η επεξεργασία των σημάτων θα πρέπει να είναι συνεχώς διαθέσιμες όλες οι τιμές που έχουν από προηγουμένως ληφθεί και αυτό επιτυγχάνεται με τη χρήση της συνάρτησης AppendData. Κατ' ουσίαν η ρουτίνα αυτή προσθέτει τις νέες τιμές που λήφθηκαν τελευταίες, στη μεταβλητή τύπου array με την ονομασία dataAll από την οποία μπορούν να ανακτηθούν όλα τα επιθυμητά αποτελέσματα.

IV. Πρόγραμμα ελεγκτή σε περιβάλλον C++.

Σε αυτό το πρόγραμμά πραγματοποιείται ο έλεγχος της εξωσκελετικής διάταξης. Γίνεται η υλοποίηση του ελεγκτή υποχωρητικότητας του exoskeleton και η σύνδεσή του με τον Η/Υ που λαμβάνει τα ηλεκτρομυογραφικά σήματα.

```
/*
* new5compliance.cpp
* =====
*
* DC Motor Control using SERVOTOGO card
*
*/

#ifdef HAVE_CONFIG_H
#include <config.h>
```

```
#endif

#include "ServoToGo.hpp"
#include <iostream.h>
#include <stdlib.h>
#include <stdio.h>
#include <unistd.h>
#include <iomanip.h>
#include <sys/sched.h>
#include <conio.h>
#include <cmath>
#include <string.h>
#include <errno.h>
#include <termios.h>
#include <fcntl.h>
#include <sys/ioctl.h>

#define PI 3.141592654
#define SAMPLES 5000
#define timesteps 20 //timesteps for which the desired joint angle remains
constant

float tf,dt;
float a2,a3;
int q[SAMPLES] ;
int qd[SAMPLES];
int qdummy[SAMPLES];
```

```

//      Serial port variables
int fd = 0 ;          /* File handler to the Serial Port */
int BytesRead=0;
int ReturnValue;
int input=0;
void Serial_Init();

int main ( void )
{
    unsigned int ticks ;
    unsigned int i ;
    float Angle, Ts, Frequency ;

    //      Define file handle

    FILE *pfile ;
    FILE *pfile2 ;

    //      Defining channel

    int exos = 1;

    //      Definition of parameters that will be used on PD control loop

    int PositionExoskeleton ;
    double CurrentExoskeleton[SAMPLES] ;
    float VoltageExoskeleton ;
    double PP = 0.0;

```

```

double PP2 = 0.0;
double vel = 0.0;
double acc = 0.0;

//      Definition of parameters that will be used on the Butterworth
filter

float dummyvel[2];
float dummyvelf[2];
float dummyacc[2];
float dummyaccf[2];

float b1 = 0.000944691843;
float b2 =0.001889383687;
float b3 =0.000944691843;
float a22 = -1.911197067426;
float a32 = 0.914975834801;

//      Initial state of parameters

PositionExoskeleton =0 ;
CurrentExoskeleton[0] = 0.0 ;
VoltageExoskeleton = 0.0 ;
q[0] = 0;
qd[0]=0;
qdummy[0]=0;
dummyvel[0] = 0.0;
dummyvel[1] = 0.0;

```

```

dummyvel[0] = 0.0;
dummyvel[1] = 0.0;
double velfiltered = 0.0;
dummyacc[0] = 0.0;
dummyacc[1] = 0.0;
dummyaccf[0] = 0.0;
dummyaccf[1] = 0.0;
double accfiltered = 0.0;

double dummy = 0.0;

ServoToGo *stg;

// Initialization of IRQ,BASE ADDRESS and ADC range
cout << " + Initilization of SERVOTOGO card" << endl ;
stg = new ServoToGo(5, 0, 10) ;
sleep(1) ;

// Giving high priority to the program

setprio(getpid(),29);

// Check the ServoToGo status (maybe the board isn't present,
etc)

cout << " + Checking STATUS of SERVOTOGO card" << endl ;
if (stg->d_status.isStatusError())
{
    cout << " \tERROR : Can't find board" << endl ;
}

```



```
        exit(-1) ;  
    }  
    cout << " \tSTATUS : OK" << endl ;  
    sleep(1) ;
```

Στην αρχή του προγράμματος εισάγονται τα απαραίτητα header files και ορίζονται οι μεταβλητές και συναρτήσεις που θα χρησιμοποιηθούν. Ορίζονται οι μεταβλητές της σειριακής θύρας, δείκτες επαναληπτικών βρόχων και οι μεταβλητές της τελικής γωνίας και της συχνότητας δειγματοληψίας. Καθορίζουμε ότι ο κινητήρας της εξωσκελετικής διάταξης είναι συνδεδεμένος στο κανάλι 1 της κάρτας και ορίζουμε τις μεταβλητές που χρησιμοποιούνται στον επαναληπτικό βρόχο, τις παραμέτρους του φίλτρου και τις μεταβλητές (dummy) όπου αποθηκεύουμε τις τιμές που έχουν προέλθει από προηγούμενες χρονικές στιγμές. Ακόμη ορίζονται οι αρχικές τιμές των παραπάνω μεγεθών και αρχικοποιούμε την κάρτα. Τέλος δίνουμε αυξημένη προτεραιότητα εκτέλεσης στο πρόγραμμα και γίνεται έλεγχος για τη σωστή τοποθέτηση και λειτουργία της κάρτας δημιουργώντας μία παύση για την αποκατάσταση πιθανών μεταβατικών φαινομένων.

```
// Input desired frequency  
  
do {  
    cout << " + Enter desired sample frequency (55-3000)Hz : " ;  
    cin >> Frequency ;  
}  
  
while ( (Frequency > 3000) || (Frequency < 50) ) ;  
  
sleep(1) ;
```

```

//
pfile2=fopen("P.txt","w") ;

//
//    Zero Output Voltage
stg->writeVoltage(0,0.0) ;
sleep(2) ;

//
//    Initialize & Start Timer
stg->startTimer0Periodic (Frequency);           //    Start    timer    0
periodic 2000Hz
stg->enableTimer0Interrupt() ;
stg->clearTimer0Interrupts() ;
cout<<Frequency<<endl;

//Initialize Encoders

stg->setEncoder(exos,0);
stg->enableIDLLatching(exos);
stg->resetIDL(exos);

```

Σε αυτό το στάδιο εισάγεται η τιμή της συχνότητας δειγματοληψίας από το χρήστη, η οποία θα πρέπει να είναι εντός των επιθυμητών ορίων. Στέλνουμε στον κινητήρα μηδενική τάση για ασφάλεια, ενεργοποιούμε το χρονόμετρο 0 σε περιοδική λειτουργία με συχνότητα ίση με αυτή που εισήχθη από το

χρήστη και του δίνουμε τη δυνατότητα να παράγει διακοπές στο σύστημα. Τέλος αρχικοποιούμε την παλμογεννήτρια του κινητήρα.

```
// CONTROL LOOP
//
for ( ticks = 1 ; ticks < SAMPLES ; ticks++ )
    {
        stg->resetWDT();

        // Wait for the timer to tick

        if (stg->waitForTimer0Interrupt() > 0)
            {
                cerr << " The timer overran " << endl;
                break;
            }

        if (stg->d_status.isStatusError())
            {
                cerr << stg->d_status.getMessageText();
                break;
            }

        // Reading the exoskeleton encoder

        if (stg->isIDLActive(exos))
```

```

{
    PositionExoskeleton = stg->readEncoder(exos);
    stg->resetIDL(exos);
}
else
{
    PositionExoskeleton = stg->readEncoder(exos);
}

stg->beginAdcConversion(exos);
    while(!stg->isAdcConversionFinished());
CurrentExoskeleton[ticks] = stg->getVoltage(exos);

```

Στην αρχή του επαναληπτικού βρόγχου μηδενίζουμε το χρονόμετρο Watchdog και στη συνέχεια περιμένουμε για τη δημιουργία διακοπής από το χρονόμετρο 0. Διαβάζουμε την τιμή της παλμογεννήτριας του κινητήρα και ξεκινάμε τη διαδικασία αναλογοψηφιακής μετατροπής για τη μέτρηση του ρεύματος του κινητήρα της εξωσκελετικής διάταξης. Όταν ολοκληρωθεί αποθηκεύουμε την τιμή που προέκυψε στην αντίστοιχη μεταβλητή.

```

PP = (double) PositionExoskeleton/67*360/8000/180*PI;
vel = (PP - dummy)*Frequency;
if (vel < 0.2 && vel > 0.0)
{
    vel = 0.2;
}

```

```

else if (vel > -0.2 && vel < 0.0)
{
    vel = -0.2;
}

velfiltered = b1*vel +b2*dummyvel[1] +b3*dummyvel[0] -
a22*dummyvel[1] - a32*dummyvel[0];

acc = (velfiltered - dummyvel[1]) * Frequency ;

accfiltered = b1*acc +b2*dummyacc[1] +b3*dummyacc[0] -
a22*dummyacc[1] - a32*dummyacc[0];

if (accfiltered > 0.6 && accfiltered > 0.0)
{
    accfiltered = 0.6;
}
else if (accfiltered < -0.6 && accfiltered < 0.0)
{
    accfiltered = -0.6;
}
if (velfiltered >= 0.0005)
{
    VoltageExoskeleton =(double) (0.2*accfiltered+0.31885*sin(PP) +
1.343*velfiltered +1.46);
}

```

```

else if (velfiltered <=-0.0005)
{
    VoltageExoskeleton =(double) (0.2*accfiltered+0.31885*sin(PP) +
1.456*velfiltered -2.102);
}
else
{
    VoltageExoskeleton = 0.0;
}

```

Ο ελεγκτής υποχωρητικότητας υλοποιείται σε αυτό το κομμάτι του προγράμματος. Αποθηκεύουμε την τιμή της θέσης της εξωσκελετικής διάταξης και υπολογίζουμε την γωνιακή ταχύτητα με αριθμητικό σχήμα πρώτης τάξης. Έπειτα ελέγχουμε αν η απόλυτη τιμή της ταχύτητας είναι μεγαλύτερη από την τιμή που έχουμε θέση ως όριο (0.2) και εάν δεν είναι θέτουμε την ταχύτητα του μηχανισμού ίση με το μηδέν, για να αποφευχθούν προβλήματα πιθανής ταλάντωσης του μηχανισμού σε μικρές ταχύτητες. Φιλτράρουμε το σήμα για την απομάκρυνση του θορύβου και παραγωγίζοντας και φιλτράροντας ξανά υπολογίζουμε την γωνιακή επιτάχυνση. Ελέγχουμε την απόλυτη τιμή της ταχύτητας και της επιτάχυνσης ώστε να αποφύγουμε φαινόμενα ταλαντώσεων για μικρές αποκλίσεις. Στη συνέχεια υπολογίζεται η τιμή του ρεύματος που πρέπει να “γράψουμε” στον κινητήρα του exoskeleton λαμβάνοντας υπόψη τις δυναμικές παραμέτρους του συστήματος, ώστε ο μηχανισμός να είναι υποχωρητικός.

```

//      Check for Minimum & Maximum allowable Output (exoskeleton)

        if ( VoltageExoskeleton > 10.0 )
        {
                stg->writeVoltage(exos,10.0) ;
        }

        else if ( VoltageExoskeleton < -10.0 )

```

```

        {
            stg->writeVoltage(exos,-10.0) ;
        }
    else if ( VoltageExoskeleton < 1.5 && VoltageExoskeleton > 0.0
)
    {
        stg->writeVoltage(exos,1.5) ;
    }
    else if ( VoltageExoskeleton < 0.0 && VoltageExoskeleton > -1.0
)
    {
        stg->writeVoltage(exos,-1.0) ;
    }
    else

    {
        stg->writeVoltage(exos,VoltageExoskeleton) ;
    }

    if ( stg->hasWDTTripped()
    {
        cout << " Watchdog Timer tripped " << endl;
    }

dummy = PP ;
dummyvel[0] = dummyvel[1];
dummyvel[1] = vel;
dummyvelf[0] = dummyvelf[1];
dummyvelf[1] = velfiltered;

```

```

dummyacc[0] = dummyacc[1];
dummyacc[1] = acc;
dummyaccf[0] = dummyaccf[1];
dummyaccf[1] = accfiltered;

CurrentExoskeleton[ticks] = VoltageExoskeleton;

PositionExoskeleton;

    //    Read data from Serial Port
BytesRead = read(fd, &ReturnValue, 1);
    printf("%d\n", ReturnValue) ;
    }

    //    Close port

printf("\n CLOSING Serial Port\n") ;
close(fd) ;

cout << "telos" << endl;
    stg->writeVoltage(exos,0) ;

    //    Wait for everything to settle
stg->disableTimer0Interrupt();
    sleep(1) ;

    //    Open file

pfile=fopen("poppy.txt","w") ;

```



```

//      Write file

cout << " + Starting writing of data to file" << endl ;
for ( i = 0 ; i < SAMPLES ; i++ )
{
        fprintf(pfile," %ld          %d          %f          %f          \n
",CurrentExoskeleton[i]/0.4) ;
}

cout << " + Ending writing of data to file" << endl ;

//      Close file

fclose(pfile) ;
fclose(pfile2) ;

//      Stop DC Motor
cout << " + Stopping DC Motor" << endl ;
cout << " Actual angle = " << PositionExoskeleton*360/8000/67
<<endl;
stg->writeVoltage(0,0.0) ;

delete stg;
return(0);
}

```

Γίνεται έλεγχος ώστε να διαπιστώσουμε αν η τιμή ρεύματος που υπολογίστηκε από τον ελεγκτή υποχωρητικότητας εμπίπτει στο εύρος

λειτουργίας της κάρτας και στη συνέχεια τη στέλνουμε στον κινητήρα της εξωσκελετικής διάταξης. Έπειτα ελέγχουμε αν εξέπνευσε ο Watchdog timer ώστε να ενημερωθεί ο χρήστης με ανάλογο μήνυμα που εκτυπώνεται στην οθόνη. Ακολουθεί ανανέωση των ψευδομεταβλητών και η ανάγνωση των δεδομένων από τη σειριακή θύρα. Κλείνουμε τη σειριακή θύρα και στέλνουμε μηδενικές τιμές τάσης και ρεύματος στον κινητήρα της εξωσκελετικής διάταξης. Στη συνέχεια απενεργοποιούμε τη λειτουργία διακοπών του χρονομέτρου 0, ανοίγουμε ένα αρχείο και αποθηκεύουμε σε αυτό τις τιμές των μεγεθών που μας ενδιαφέρουν. Τέλος κλείνουμε τα ενεργά αρχεία, εκτυπώνουμε στην οθόνη την τελική τιμή της γωνίας του exoskeleton και απενεργοποιούμε την κάρτα, ολοκληρώνοντας με τον τρόπο αυτό τον βρόγχο ελέγχου.

```
void Serial_Init () /* Opens and sets Serial_Port */
{
    struct termios options ;
    /* Clear struct termios & input buffer */
    memset(&options, 0, sizeof(options)) ;

    /* Open Serial Port */

    fd = open( "/dev/ser2", O_RDWR | O_NOCTTY | O_NDELAY) ;
    if (fd < 0)
        printf("COULD NOT OPEN SERIAL PORT\n") ;
    else
    {
        fcntl( fd, F_SETFL, 0) ;
        printf("SERIAL PORT OPENED SUCCESSFULLY\n") ;
    }
}
```

```
/* Get Settings of Serial Port */
tcgetattr( fd, &options) ;

/*Set the baud rates to 9600 */
cfsetispeed( &options, B9600) ;
cfsetospeed( &options, B9600) ;

/* Setting parity (8N1) */
options.c_cflag &= ~PARENB ; // No parity
options.c_cflag &= ~CSIZE ; // 8 bits
options.c_cflag |= CS8 ;

/* Setting Flow Control and Echoing */
options.c_iflag&=~IXON;
options.c_iflag&=~IXOFF;
options.c_lflag &= ~(ICANON | ECHO | ECHOE | ECHONL| ISIG ) ;
options.c_oflag&=~OPOST;

/* Set new Settings of Serial Port */
tcsetattr(fd,TCSAFLUSH,&options) ;

return;
}
```

Στο τελευταίο τμήμα του προγράμματος παρουσιάζονται οι συναρτήσεις που ελέγχουν τη σειριακή θύρα του υπολογιστή. Όταν καλούνται συνδέουν προγραμματιστικά τους δύο υπολογιστές “ανοίγοντας” τη θύρα σύνδεσης RS232 και στη συνέχεια λαμβάνονται τα επιθυμητά δεδομένα που έχουν σταλεί από τον Η/Υ που λαμβάνει και επεξεργάζεται τα EMGs.

Βιβλιογραφία

- [1] Andreasen D. S., Allen S. K., Backus D. A.”, Exoskeleton with EMG Based Active Assistance for Rehabilitation”, IEEE International Conference on Rehabilitation Robotics, 2005
- [2] Brian W. Kernighan, Dennis M. Ritchie, “The C programming Language”, second edition.
- [3] DELSYS website [online]. Διαθέσιμο στη διεύθυνση: http://www.delsys.com/Products/Bagnoli_Desktop.html
- [4] Kiguchi Kazuo, Kariya Shingo, Watanabe Keigo, Izumi Kiyotaka, Fukuda Toshio, “An Exoskeletal Robot for Human Elbow Motion Support-Sensor Fusion, Adaptation and Control”, IEEE Transactions on Systems, Man and Cybernetics, June 2001.
- [5] Kiguchi Kazuo, Iwami Koya, Yasuda Makoto, Kurata Hideaki, Watanabe Keigo, Fukuda Toshio, ”Intelligent Interface and Control of an Exoskeletal Robot for Human Shoulder Motion Support Considering Subject’s Arm Posture”, IEEE International Conference on Robotics and Automation, May 2002.
- [6] Reza Boostani, Mohammad Hassan Moradi “Evaluation of the forearm EMG signal features for the control of a prosthetic hand” March 2003
- [7] Rosen Jacob, Brand Moshe, Fuchs Moshe B., Arcan Mircea, “A Myosignal-Based Powered Exoskeleton System”, IEEE Transaction on Systems, Man and Cybernetics, May 2001.
- [8] SERVOTOGO website [online]. Διαθέσιμο στη διεύθυνση: <http://www.servotogo.com/>
- [9] Κρικέλης Ν. Ι., “Εισαγωγή στον Αυτόματο Έλεγχο: Θεωρία και Εφαρμογές”, Εκδόσεις Συμμετρία, Αθήνα, 2000.
- [10] Παπαδόπουλος Ε., Κυριακόπουλος Κ., “Σημειώσεις Ρομποτικής, Εθνικό Μετσόβιο Πολυτεχνείο”, Αθήνα 2004.
- [11] Αρτεμιάδης Παναγιώτης “ “
- [12] Φαγογένης Γεώργιος “Σχεδιασμός και έλεγχος εξωσκελετικής διάταξης ενός βαθμού ελευθερίας για την ενίσχυση του άνω ακρου”, Αθήνα 2007