

Εργασία στο Μάθημα της Βιοϊατρικής Τεχνολογίας

Θέμα: « Κατασκευή συστήματος καταγραφής επιφανειακού
μυογραφήματος μέσω Arduino και προβολή του στο matlab σε
πραγματικό χρόνο»



Διδάσκων:

Λεόντιος Χατζηλεοντιάδης

Οι φοιτητές:

Φίλιππος Σαμλίδης (7973)

Γεώργιος Σχινάς (7985)

Χρήστος Ταραμονλής (7986)

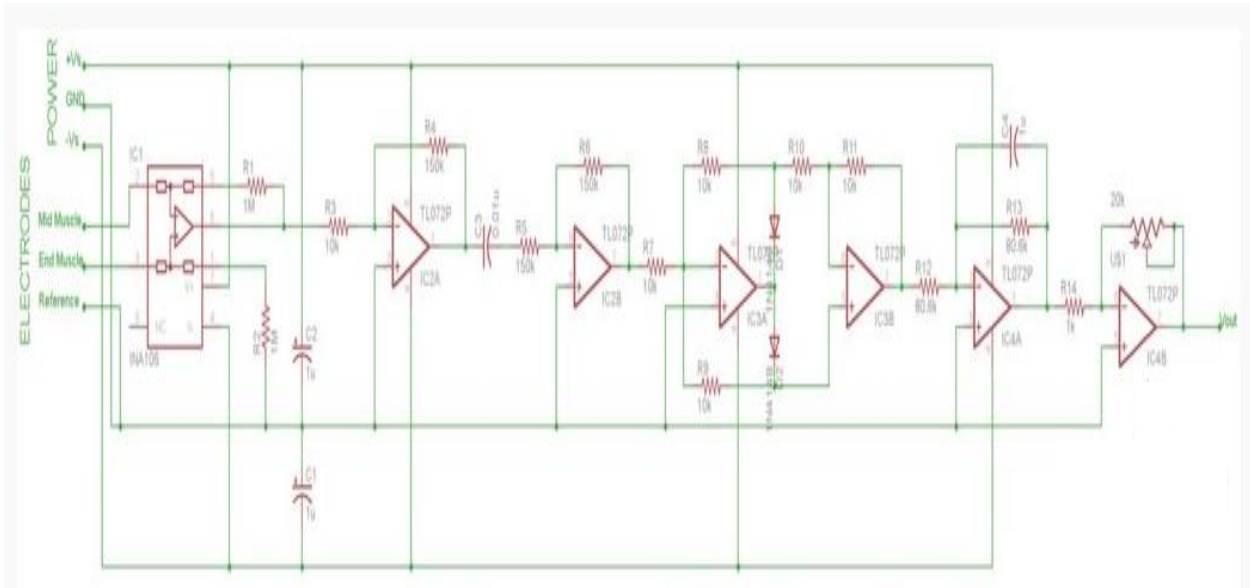
Ηλίας Χρυσοβέργης (8009)

1. Κατασκευή του κυκλώματος

Για την κατασκευή του κυκλώματος αρχικά χρησιμοποιήσαμε ένα έτοιμο κύκλωμα που βρήκαμε στην ιστοσελίδα:

<http://www.instructables.com/id/Muscle-EMG-Sensor-for-a-Microcontroller/>.

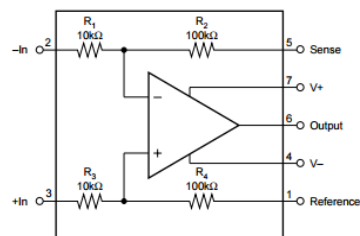
Το κύκλωμα δίνεται στην συνέχεια:



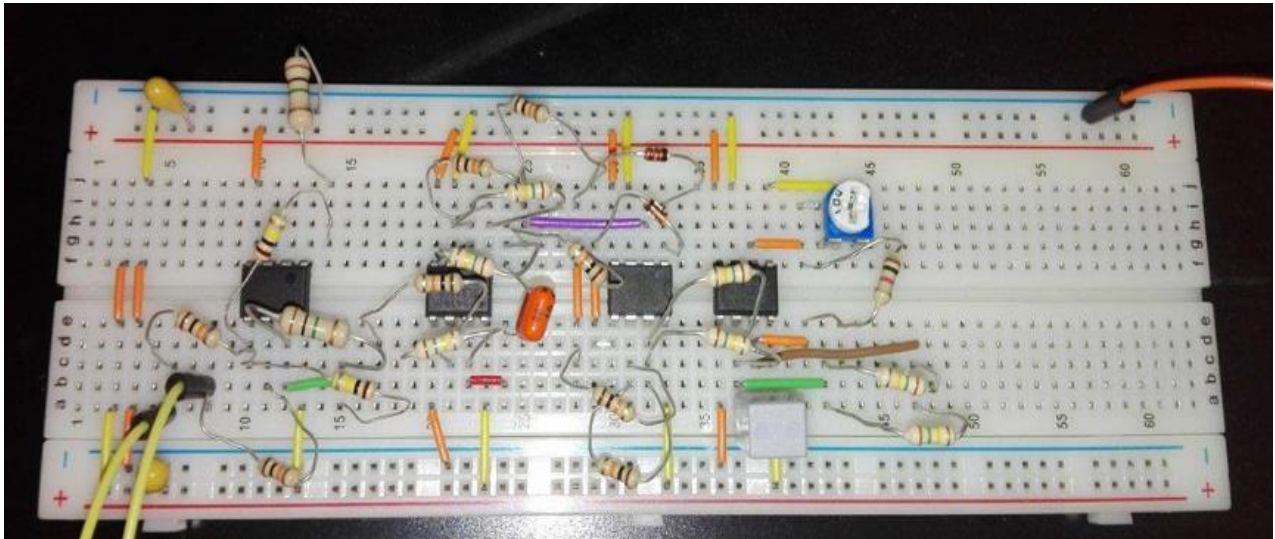
Δεν θα κάνουμε ανάλυση για αυτό το κύκλωμα εφόσον υπάρχει εκτενέστατη και πολύ λεπτομερής ανάλυση για κάθε βαθμίδα του στην ιστοσελίδα.

Το κύκλωμα αποτελείται από ενισχυτή ακριβείας, υψιπερατά και χαμηλοπερατά φίλτρα όπως και σταθεροποιητή και ανυψωτή τάσης.

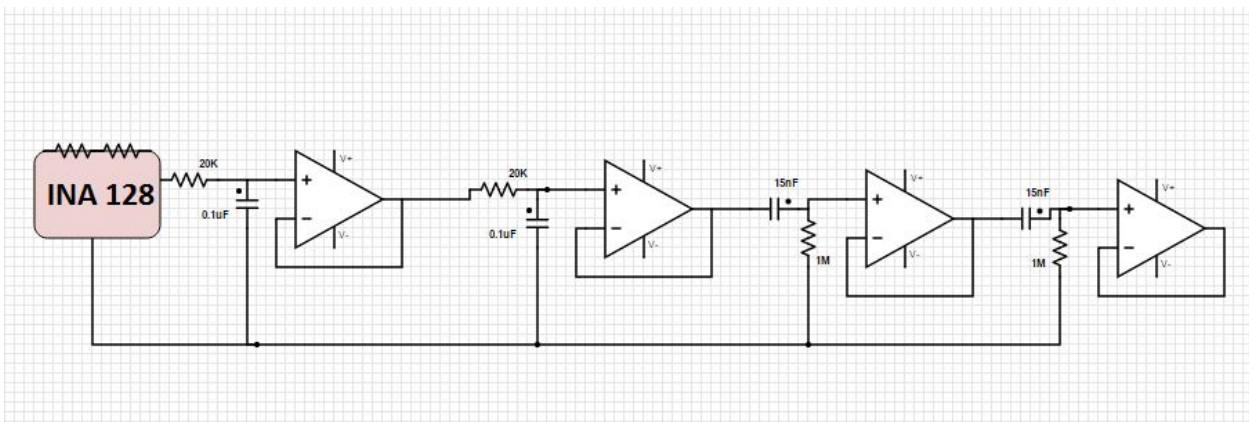
Δεν βρήκαμε πουθενά στην αγορά τον INA 106 οπότε αναγκαστήκαμε και τον φτιάξαμε μόνοι μας. Βλέποντας στο documentation του, αποφασίσαμε να χρησιμοποιήσουμε έναν απλό τελεστικό ενισχυτή με 4 αντιστάσεις όπως φαίνεται στην συνέχεια:



Το σχέδιο του κυκλώματος στο ράστερ δίνεται στην συνέχεια:

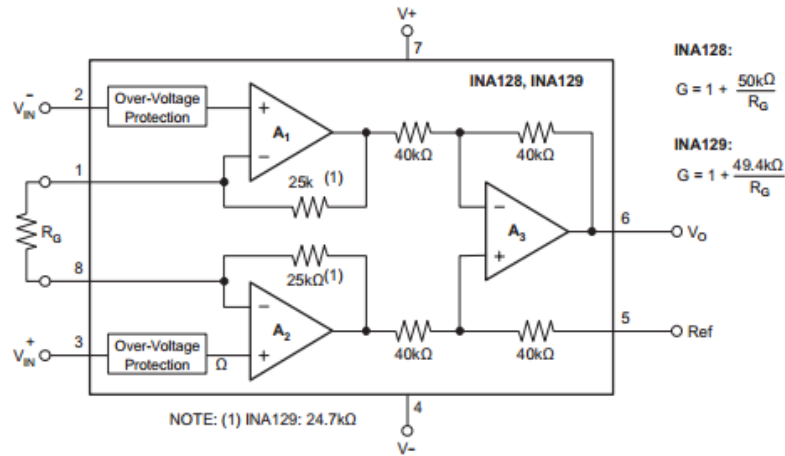


Εφόσον δουλέψαμε πάνω σε αυτό το κύκλωμα και βγάλαμε αποτελέσματα αποφασίσαμε να προσπαθήσουμε να κάνουμε και ένα δικό μας πιο απλοποιημένο κύκλωμα. Η σχεδίαση του κυκλώματος μας έδωσε αυτό το αποτέλεσμα:



Το κύκλωμα αυτό έχει στην αρχή έναν ενισχυτή ακριβείας (τον INA 128) ο οποίος παίρνει στην είσοδο του το σήμα από τον μν.

Στο σχήμα βλέπουμε το κύκλωμα του INA 128:

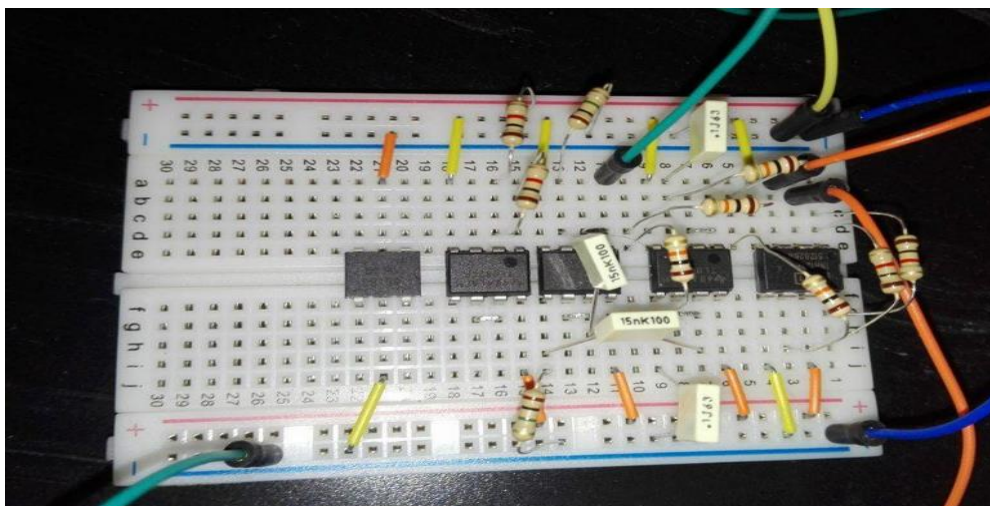


Η ενίσχυση του όπως βλέπουμε από το υπόμνημα εξαρτάται από την αντίσταση R_G η οποία συνδέεται στους ακροδέκτες 1 και 8. Μετά από διαδικασία trial and error βρήκαμε πως η τιμή $R_G = 10$ μας έδινε καλά αποτελέσματα στην έξοδο.

Οι 2 τελεστικοί στην συνέχεια λειτουργούν σαν ενεργά χαμηλοπερατά φίλτρα 2^{ης} τάξης ενώ οι επόμενοι 2 είναι υψιπερατά φίλτρα.

Η έλλειψη ανόρθωσης και σταθεροποίησης του σήματος θα δώσει χειρότερα αποτελέσματα σε σχέση με το πρώτο κύκλωμα. Θα το παρατηρήσουμε στη συνέχεια στην καταγραφή του σήματος.

Η σχεδίαση δίνεται στην συνέχεια:



2. Σύνδεση στο Arduino

Για την εκκίνηση του Arduino, την δειγματοληψία και την αποθήκευση των δεδομένων στο serial port του υπολογιστή χρησιμοποιήσαμε τον εξής κώδικα:

```
1  #include <math.h>
2  int analogPin = 3;
3  int val = 0;
4  float sample;
5  int i = 0;
6
7  void setup() {
8      Serial.begin(9600);
9      Serial.println('a');
10     char a = 'b';
11     while (a != 'a')
12     {
13         a = Serial.read();
14     }
15 }
16
17 void loop() {
18     while (Serial.available() == 0)
19     {
20     }
21
22     if (Serial.available() > 0)
23     {
24         val = Serial.read();
25         if (val == 'R')
26         {
27             sample = analogRead(analogPin);
28             //sample = cos(double(i)/10.);
29             sample *= 5./1024.;
30             Serial.println(sample);
31         }
32     }
33     i++;
34     delay(10);
35 }
```

3. Απεικόνιση στο Matlab

Για την real time απεικόνιση στο matlab χρησιμοποιήσαμε τους εξής κώδικες:

Αρχικά χρησιμοποιήθηκε το επόμενο script για την σύνδεση στο serial port:

```
1 function[obj,flag] = setupSerial(comPort)
2 % It accept as the entry value, the index of the serial port
3 % Arduino is connected to, and as output values it returns the serial
4 % element obj and a flag value used to check if when the script is compiled
5 % the serial element exists yet.
6 flag = 1;
7 % Initialize Serial object
8 obj = serial(comPort);
9 set(obj,'DataBits',8);
10 set(obj,'StopBits',1);
11 set(obj,'BaudRate',9600);
12 set(obj,'Parity','none');
13 fopen(obj);
14 a = 'b';
15 while (a~='a')
16     a=fread(obj,1,'uchar');
17 end
18 if (a=='a')
19     disp('Serial read');
20 end
21 fprintf(obj,'%c','a');
22 mbox = msgbox('Serial Communication setup');
23 uiwait(mbox);
24 fscanf(obj,'%u');
25 end
```

Στην συνέχεια, αυτό το script διαβάζει από το serial port:

```
1 function [output] = readData(s,command)
2 % Serial send read request to Arduino
3 fprintf(s,command);
4
5 % Read value returned via Serial communication
6 output = fscanf(s,'%f');
7
8 end
```

Το τελευταίο script καλεί τα προηγούμενα scripts και εμφανίζει σε πραγματικό χρόνο την δραστηριότητα του μυ:

```
1 % Save the serial port name in comPort variable.
2 comPort = 'COM4';
3 % It creates a serial element calling the function "stupSerial"
4 [arduino,serialFlag] = setupSerial(comPort);
5 % Time to create our plot window in order to visualize data collected
6 % from serial port readings
7 if (~exist('h','var') || ~ishandle(h))
8     h = figure(1);
9     set(h,'UserData',1);
10 end
11 if (~exist('button','var'))
12     button = uicontrol('Style','togglebutton','String','Stop',...
13         'Position',[0 0 50 25], 'parent',h);
14 end
15 % After creating a system of two axis, a line object through which the data
16 % will be plotted is also created
17 buf_len = 500;
18 index = 1:buf_len;
19 zeroIndex = zeros(size(index));
20 tcdata = zeroIndex;
21 limits = [-2 2];
22 myAxes = axes('Xlim',[0 buf_len],'Ylim',limits);
23 grid on;
24 l = plot(index,[tcdata;zeroIndex]);
25 drawnow;
26 % last part
27 button = uicontrol('Style','togglebutton','String','Stop');
28 mode = 'R';
29 while (get(button,'Value') == 0 )
30     tc = readData(arduino,mode);
31     tcdata = [tcdata(2:end),tc];
32     set(l,'Ydata',tcdata);
33     drawnow;
34 end
35 fclose(arduino)
```

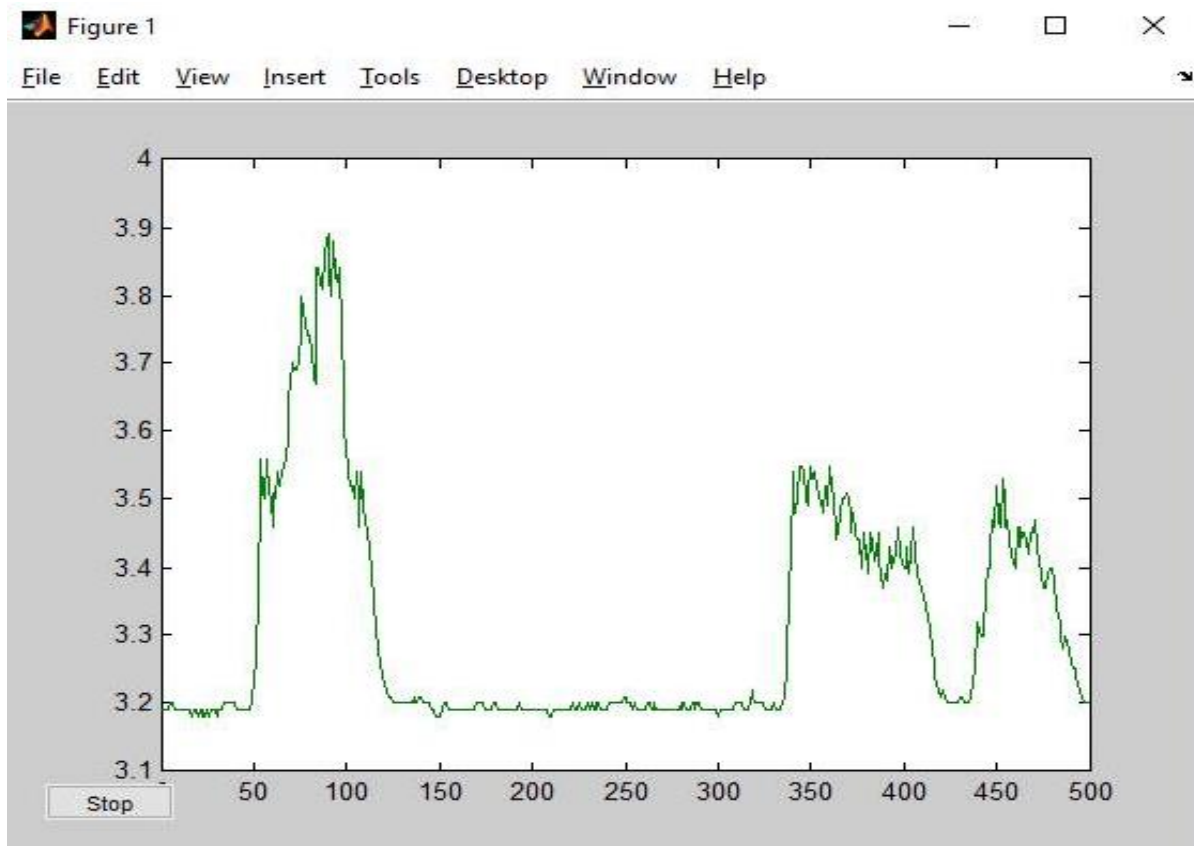
Η απεικόνιση έγινε στο matlab, διότι το matlab υποστηρίζεται σε όλες τις κατηγορίες smartphone (Android, Iphone, Microsoft) και λειτουργικών συστημάτων (Windows, Mac, Linux), οπότε το σύστημα καταγραφής μπορεί να χρησιμοποιηθεί σε οποιαδήποτε συσκευή. Επίσης, το matlab έχει πληθώρα συναρτήσεων και διαδικασιών για την ψηφιακή επεξεργασία σήματος, που θα μπορούσε να χρησιμοποιηθεί για την διερεύνηση και άλλων χαρακτηριστικών του σήματος.

4. Πείραμα

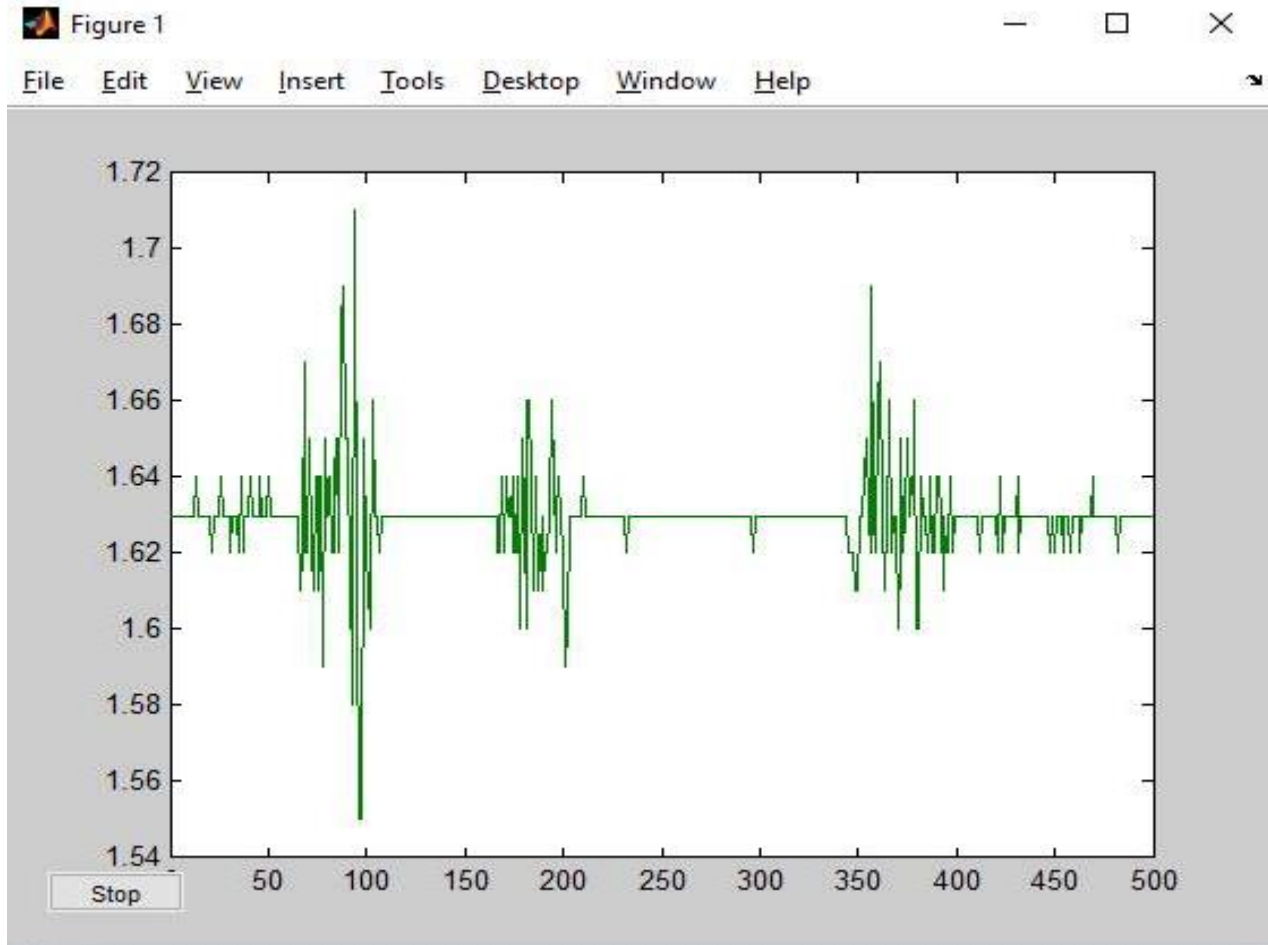
Για την διαδικασία του πειράματος τοποθετούμε 3 ηλεκτρόδια στο υποκείμενο. Ένα ηλεκτρόδιο αναφοράς (σε κάποιο κόκκαλο , ώστε να μην επηρεάζεται από τις συσφίξεις του μυ ή άλλων μυών), ένα ηλεκτρόδιο στο μέσο του μυ και ένα στο άκρο του. Στην είσοδο του ενισχυτή ακριβείας στέλνουμε το σήμα από τα ηλεκτρόδια στον μυ. Στο Arduino δίνουμε την έξοδο του κυκλώματος, που είναι το σήμα των ηλεκτροδίων ύστερα από ενίσχυση, αποκοπή συγκεκριμένων συχνοτήτων (και ανύψωση , σταθεροποίηση για το 1^ο κύκλωμα).

Η απεικόνιση που μας έδωσαν τα 2 κυκλώματα (μέσω του matlab) φαίνεται στην συνέχεια:

Για το 1^ο κύκλωμα:



Για το 2^ο κύκλωμα:



Είναι φανερό η διαφορά ποιότητα στις 2 απεικονίσεις. Το πρώτο κύκλωμα, λόγω της σταθεροποίησης και της ανύψωσης δίνει πολύ πιο λείο σήμα από ότι το δεύτερο που δεν σταθεροποιεί ούτε ανυψώνει το σήμα. Επίσης, καλό θα ήταν να δώσουμε λίγο περισσότερη ενίσχυση στο 2^ο κύκλωμα.

5. Μελλοντικές εφαρμογές

Το συγκεκριμένο κύκλωμα όπως και ο κώδικας θα μπορούσε να επεκταθεί περισσότερο ώστε να προσθέσει κι άλλες λειτουργίες στο σύστημα, για να του δώσουν προστιθέμενη αξία. Κάποιες από αυτές είναι:

- a. Εύρεση κάποιας δυσλειτουργίας στα νεύρα ή στους μυς ή κάποιου προβλήματος στην αποστολή σήματος από το νεύρο στον μυ μέσα από ψηφιακή επεξεργασία σήματος,
- b. Κίνηση σε περιβάλλον εικονικής πραγματικότητας με την τοποθέτηση ηλεκτροδίων σε διάφορα χαρακτηριστικά μέρη του σώματος,
- c. Ασύρματος έλεγχος ρομπότ,
- d. Κίνηση πρόσθετου μέλους.