Τμήμα Ηλεκτρολόγων Μηχανικών και Τεχνολογίας Υπολογιστών Πάτρας
Τομέας Συστημάτων και Αυτομάτου Ελέγχου

Εργαστήριο Ψηφιακών Συστημάτων Ελέγχου
Ακαδημαϊκό Έτος 2015-2016

ΣΑΕ-ΙΙ

6η Εργαστηριακή Άσκηση

Έλεγχος με χρήση Analog-to-Digital και Digital-to-Analog μετατροπέων και πεπερασμένης ακρίβειας αριθμητικής Κινούμενου Ανεστραμμένου Εκκρεμούς

Μπολάτογλου Γεώργιος

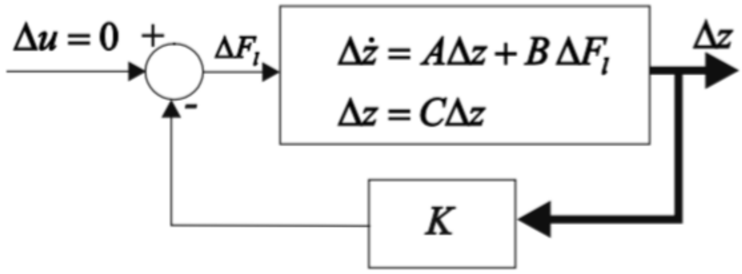
1. Δυναμικές εξισώσεις

Οι δυναμικές εξισώσεις του γραμμικοποιημένου συστήματος του κινούμενου εκκρεμούς γύρω από την κατακόρυφη θέση ισορροπίας είναι :

$$\begin{bmatrix} \Delta\dot{x} \\ \Delta\ddot{x} \\ \Delta\dot{\theta} \\ \Delta\ddot{\theta} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -\dfrac{B_l}{M} & \dfrac{mg}{M} & -\dfrac{B_r}{Ml} \\ 0 & 0 & 0 & 1 \\ 0 & -\dfrac{B_l}{Ml} & \dfrac{(m+M)g}{Ml} & -\dfrac{B_r}{ml^2} - \dfrac{B_r}{Ml^2} \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta\dot{x} \\ \Delta\theta \\ \Delta\dot{\theta} \end{bmatrix} + \begin{bmatrix} 0 \\ \dfrac{1}{M} \\ 0 \\ \dfrac{1}{Ml} \end{bmatrix} \Delta F_l$$

$$\begin{bmatrix} \Delta x \\ \Delta\dot{x} \\ \Delta\theta \\ \Delta\dot{\theta} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta\dot{x} \\ \Delta\theta \\ \Delta\dot{\theta} \end{bmatrix}.$$

Εφαρμόζουμε έναν νόμο ελέγχου ώστε οι πόλοι του κλειστού να πανε στο -1, -2, -3, -4



Εικόνα 1

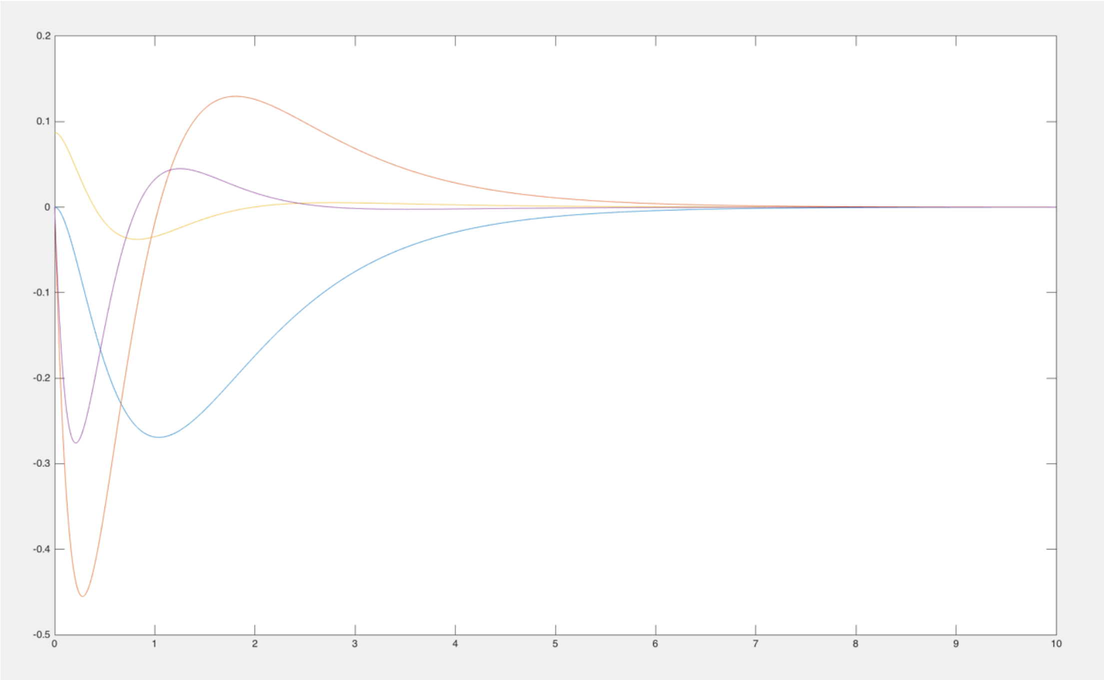
Α)

Απόκριση λόγω αρχικών συνθηκών $\Delta z(0) = [0\ 0\ \pi/36\ 0]$ του κλειστού συστήματος:

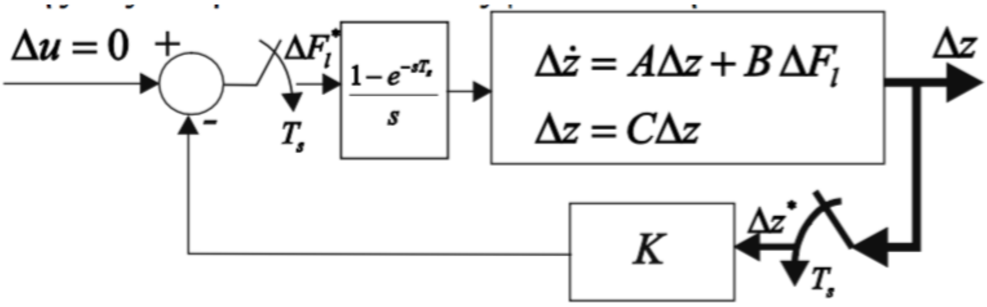
Υλοποιούμε τον εξής κώδικα:

```
clc;
clear;

M=1;m=1;l=1;B_l=0.3;B_r=0.3;g=10;

A_lin=zeros(4,4);B_lin=zeros(4,1);

A_lin(1,2)=1;
A_lin(2,2)=-B_l/M;
A_lin(2,3)=m*g/M;
A_lin(2,4)=-B_r/M/l;
A_lin(3,4)=1;
A_lin(4,2)=-B_l/M/l;
A_lin(4,3)=(m+M)*g/M/l;
A_lin(4,4)=-B_r/m/l^2-B_r/M/l^2

B_lin(2)=1/M;
B_lin(4)=1/M/l;

C_lin=eye(4);
D_lin=[0;0;0;0];
K_f_xdx=place(A_lin,B_lin,[-1;-2;-3;-4]);
A_lin_2=A_lin-B_lin*K_f_xdx;
t=[0:0.001:10];u=zeros(size(t));

z0=[0 0 pi/36 0]';
xdx_cl=lsim(A_lin_2,B_lin,C_lin,D_lin,u,t,z0);
plot(t,xdx_cl);
```

Και η απόκριση είναι:



B) Σχεδιάζουμε έναν ελεγκτή τέτοιον ώστε:



Καλούμαστε να υπολογίσουμε τις διακριτές συναρτήσεις μεταφοράς θεωρώντας ότι για διαφορετικές περιόδους δειγματοληψίας:

Προσθέτουμε στον παραπάνω κώδικα τα εξής:

```
Ts=1;
[A,B,C,D]= c2dm(A_lin_2,B_lin,C_lin,D_lin,Ts,'ZOH');
[n,d]=ss2tf(A,B,C,D);

dx=tf(n(1,:),d,Ts)
dxdot=tf(n(2,:),d,Ts)
dth=tf(n(3,:),d,Ts)
dthdot=tf(n(4,:),d,Ts)
```

όπου και μεταβάλλουμε το Ts.

Υπολογίζουμε τις διακριτές συναρτήσεις μεταφοράς θεωρώντας ότι για διαφορετικές περιόδους δειγματοληψίας:

Για Ts=0.1s:

```
dx =

  0.003582 z^3 - 0.004817 z^2 - 0.001901 z + 0.002521
  ---------------------------------------------------
     z^4 - 3.135 z^3 + 3.67 z^2 - 1.901 z + 0.3679

Sample time: 0.1 seconds
Discrete-time transfer function.


dxdot =

  0.05928 z^3 - 0.1829 z^2 + 0.1818 z - 0.0582
  ---------------------------------------------
  z^4 - 3.135 z^3 + 3.67 z^2 - 1.901 z + 0.3679

Sample time: 0.1 seconds
Discrete-time transfer function.
```

```
dth =

  0.003577 z^3 - 0.004591 z^2 - 0.00155 z + 0.002564
  ---------------------------------------------------
    z^4 - 3.135 z^3 + 3.67 z^2 - 1.901 z + 0.3679

Sample time: 0.1 seconds
Discrete-time transfer function.


dthdot =

  0.05951 z^3 - 0.1788 z^2 + 0.179 z - 0.05976
  ---------------------------------------------
  z^4 - 3.135 z^3 + 3.67 z^2 - 1.901 z + 0.3679

Sample time: 0.1 seconds
Discrete-time transfer function.
```

Για Ts=0.01

```
dx =

  4.84e-05 z^3 - 4.985e-05 z^2 - 4.538e-05 z + 4.672e-05
  ------------------------------------------------------
      z^4 - 3.901 z^3 + 5.708 z^2 - 3.711 z + 0.9048

Sample time: 0.01 seconds
Discrete-time transfer function.


dxdot =

  0.009524 z^3 - 0.02855 z^2 + 0.02852 z - 0.009495
  -------------------------------------------------
   z^4 - 3.901 z^3 + 5.708 z^2 - 3.711 z + 0.9048

Sample time: 0.01 seconds
Discrete-time transfer function.
```

```
dth =

  4.836e-05 z^3 - 4.995e-05 z^2 - 4.519e-05 z + 4.677e-05
  -------------------------------------------------------
      z^4 - 3.901 z^3 + 5.708 z^2 - 3.711 z + 0.9048

Sample time: 0.01 seconds
Discrete-time transfer function.


dthdot =

  0.009511 z^3 - 0.02853 z^2 + 0.02853 z - 0.009511
  -------------------------------------------------
   z^4 - 3.901 z^3 + 5.708 z^2 - 3.711 z + 0.9048

Sample time: 0.01 seconds
Discrete-time transfer function.
```

Για Ts=0.001

```
dx =

  4.984e-07 z^3 - 4.998e-07 z^2 - 4.952e-07 z + 4.966e-07
  -------------------------------------------------------
         z^4 - 3.99 z^3 + 5.97 z^2 - 3.97 z + 0.99

Sample time: 0.001 seconds
Discrete-time transfer function.


dxdot =

  0.0009952 z^3 - 0.002985 z^2 + 0.002985 z - 0.0009949
  -----------------------------------------------------
        z^4 - 3.99 z^3 + 5.97 z^2 - 3.97 z + 0.99

Sample time: 0.001 seconds
Discrete-time transfer function.
```

```
dth =

  4.983e-07 z^3 - 5e-07 z^2 - 4.95e-07 z + 4.967e-07
  --------------------------------------------------
      z^4 - 3.99 z^3 + 5.97 z^2 - 3.97 z + 0.99

Sample time: 0.001 seconds
Discrete-time transfer function.


dthdot =

  0.000995 z^3 - 0.002985 z^2 + 0.002985 z - 0.000995
  ---------------------------------------------------
       z^4 - 3.99 z^3 + 5.97 z^2 - 3.97 z + 0.99

Sample time: 0.001 seconds
Discrete-time transfer function.
```

Για Ts=0.0001

```
dx =

  4.998e-09 z^3 - 5e-09 z^2 - 4.995e-09 z + 4.997e-09
  ---------------------------------------------------
    z^4 - 3.999 z^3 + 5.997 z^2 - 3.997 z + 0.999

Sample time: 0.0001 seconds
Discrete-time transfer function.


dxdot =

  9.995e-05 z^3 - 0.0002999 z^2 + 0.0002998 z - 9.995e-05
  -------------------------------------------------------
       z^4 - 3.999 z^3 + 5.997 z^2 - 3.997 z + 0.999

Sample time: 0.0001 seconds
Discrete-time transfer function.
```

```
dth =

  4.998e-09 z^3 - 5e-09 z^2 - 4.995e-09 z + 4.997e-09
  ---------------------------------------------------
     z^4 - 3.999 z^3 + 5.997 z^2 - 3.997 z + 0.999

Sample time: 0.0001 seconds
Discrete-time transfer function.


dthdot =

  9.995e-05 z^3 - 0.0002999 z^2 + 0.0002999 z - 9.995e-05
  -------------------------------------------------------
       z^4 - 3.999 z^3 + 5.997 z^2 - 3.997 z + 0.999

Sample time: 0.0001 seconds
Discrete-time transfer function.
```

Για Ts=1

```
dx =

   -0.05033 z^3 - 0.1572 z^2 - 0.005327 z + 0.0004034
  -----------------------------------------------------
  z^4 - 0.5713 z^3 + 0.08497 z^2 - 0.00385 z + 4.54e-05

Sample time: 1 seconds
Discrete-time transfer function.


dxdot =

     -0.1856 z^3 + 0.1396 z^2 + 0.04787 z - 0.001934
  -----------------------------------------------------
  z^4 - 0.5713 z^3 + 0.08497 z^2 - 0.00385 z + 4.54e-05

Sample time: 1 seconds
Discrete-time transfer function.
```

```
dth =

    0.01155 z^3 - 0.02589 z^2 + 0.01373 z + 0.0005996
  -----------------------------------------------------
  z^4 - 0.5713 z^3 + 0.08497 z^2 - 0.00385 z + 4.54e-05

Sample time: 1 seconds
Discrete-time transfer function.


dthdot =

    -0.03416 z^3 + 0.04986 z^2 - 0.01254 z - 0.003157
  -----------------------------------------------------
  z^4 - 0.5713 z^3 + 0.08497 z^2 - 0.00385 z + 4.54e-05

Sample time: 1 seconds
Discrete-time transfer function.
```

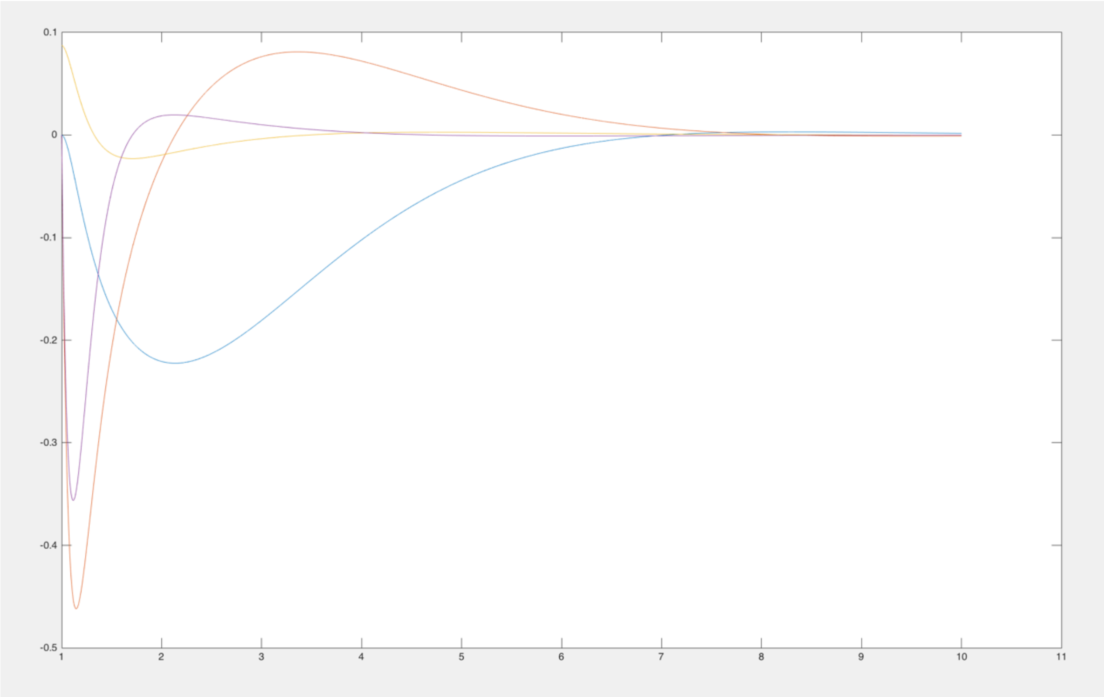
Γ)

Απόκριση $\Delta z^*(kTs)$ λόγω αρχικών συνθηκών $\Delta z(0) = [0\ 0\ \pi/36\ 0]$ του κλειστού συστήματος για διαφορετικές περιόδους δειγματοληψίας Ts:

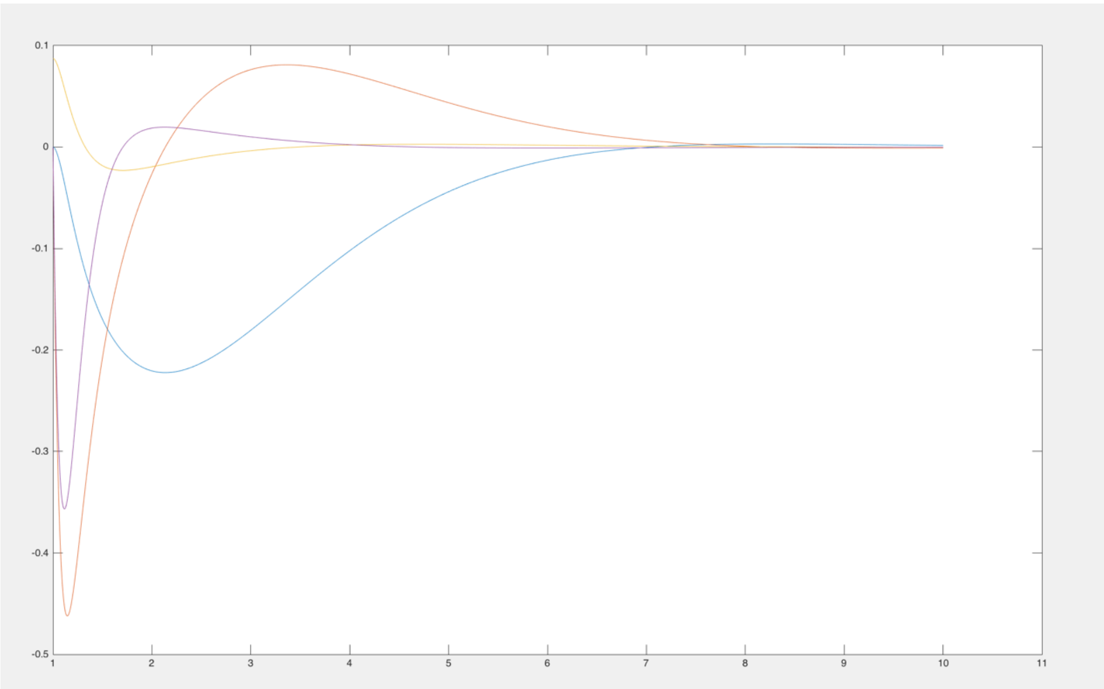
Στον παραπάνω κώδικα προσθέτουμε:

```
cnt=1;
z(:,1)=z0;
for t1=1:Ts:10
    u=-K_f_xdx*z(:,cnt);
    z(:,cnt+1)=A*z(:,cnt)+B*u;
    cnt=cnt+1;
end
t1=1:Ts:10+Ts;
plot(t1,z);
```

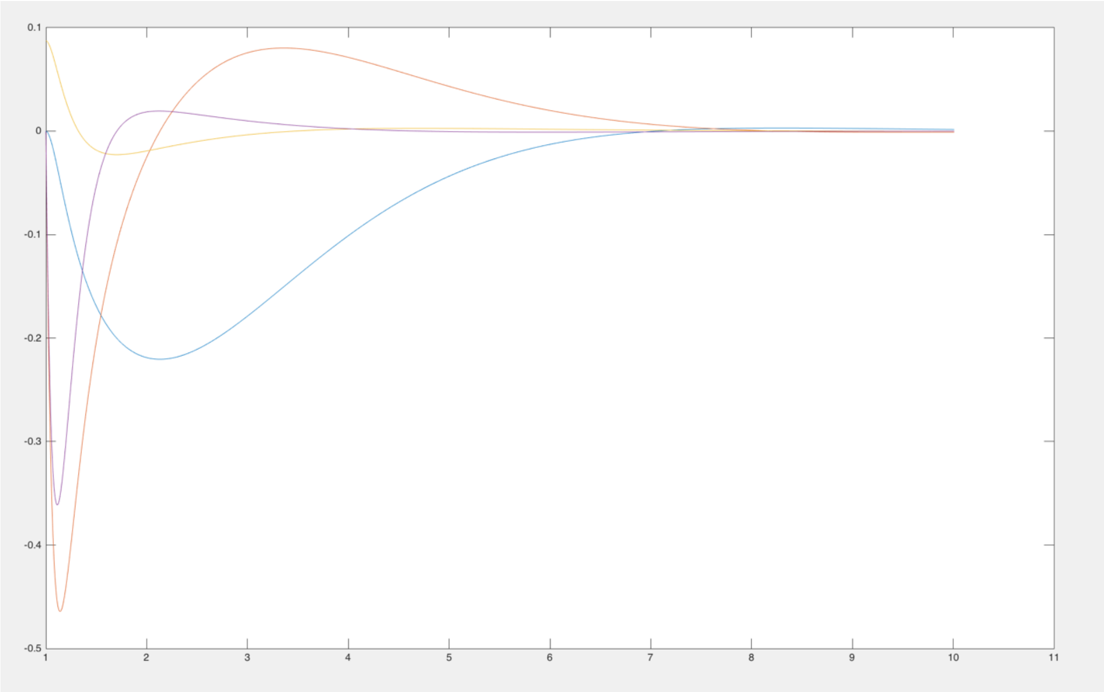
Για Ts=0.0001s



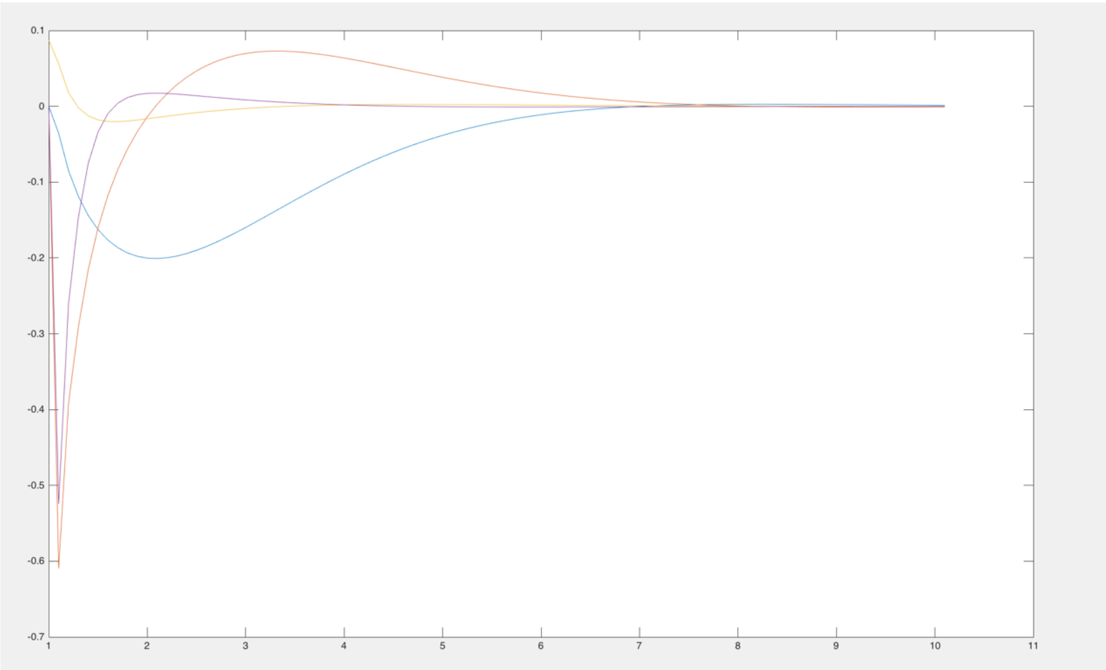
Για Ts=0.001s



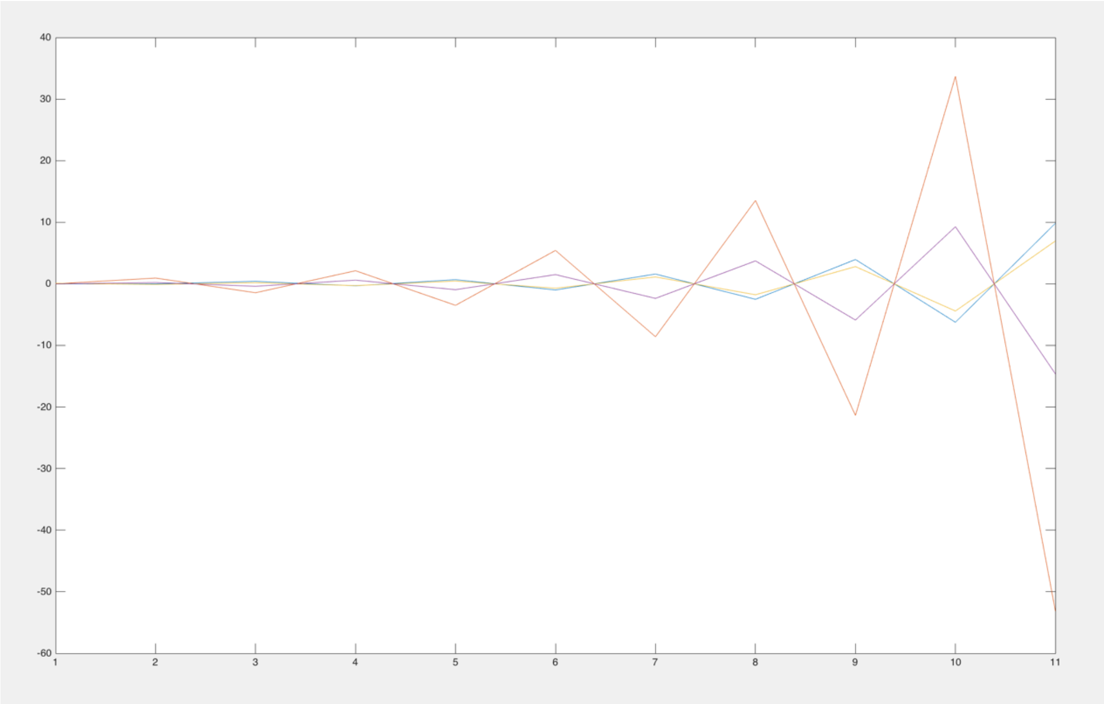
Για Ts=0.01s



Για Ts=0.1s



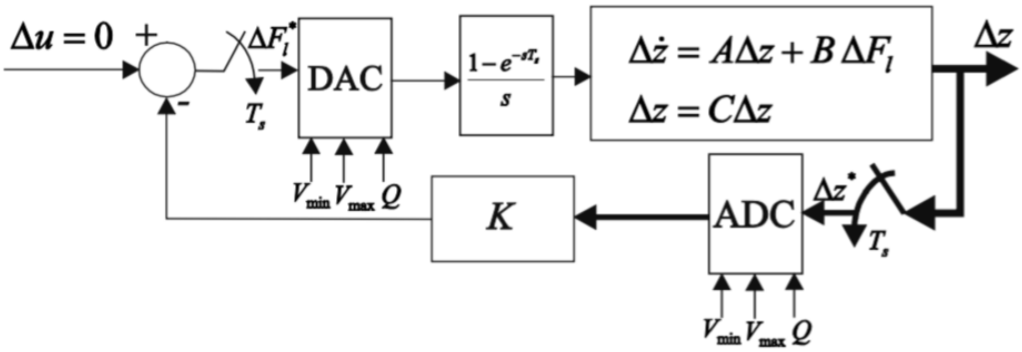
Για Ts=1s



Παρατηρούμε ότι για Ts=1 το σύστημα πάει στην αστάθεια, ενώ με τις υπόλοιπες έχουμε σχεδόν τέλειες αποκρίσεις.

Δ)

Καλούμαστε να φτιάξουμε έναν ADC και DAC μετατροπέα όπως φαίνεται παρακάτω:



Υλοποιούμε τον κώδικα:

```
clc;
clear;

M=1;m=1;l=1;B_l=0.3;B_r=0.3;g=10;

A_lin=zeros(4,4);B_lin=zeros(4,1);

A_lin(1,2)=1;
A_lin(2,2)=-B_l/M;
A_lin(2,3)=m*g/M;
A_lin(2,4)=-B_r/M/l;
A_lin(3,4)=1;
A_lin(4,2)=-B_l/M/l;
A_lin(4,3)=(m+M)*g/M/l;
A_lin(4,4)=-B_r/m/l^2-B_r/M/l^2

B_lin(2)=1/M;
B_lin(4)=1/M/l;

C_lin=eye(4);
D_lin=[0;0;0;0];
K_f_xdx=place(A_lin,B_lin,[-1;-2;-3;-4]);
A_lin_2=A_lin-B_lin*K_f_xdx;
t=[0:0.001:10];u=zeros(size(t));

z0=[0 0 pi/36 0]';
xdx_cl=lsim(A_lin_2,B_lin,C_lin,D_lin,u,t,z0);
plot(t,xdx_cl);
```

```
Q=6;
Vmin=-128;
Vmax=128;

cnt=1;
z(:,1)=z0;
ADC(:,1)=adc(z(:,1),Q,Vmin,Vmax);
for t1=1:Ts:10
    u=-K_f_xdx*ADC(:,cnt);
    z(:,cnt+1)=A*z(:,cnt)+B*u;
    ADC(:,cnt+1)=adc(z(:,cnt+1),Q,Vmin,Vmax);
    cnt=cnt+1;
end
t1=1:Ts:10+Ts;
plot(t1,ADC);
```

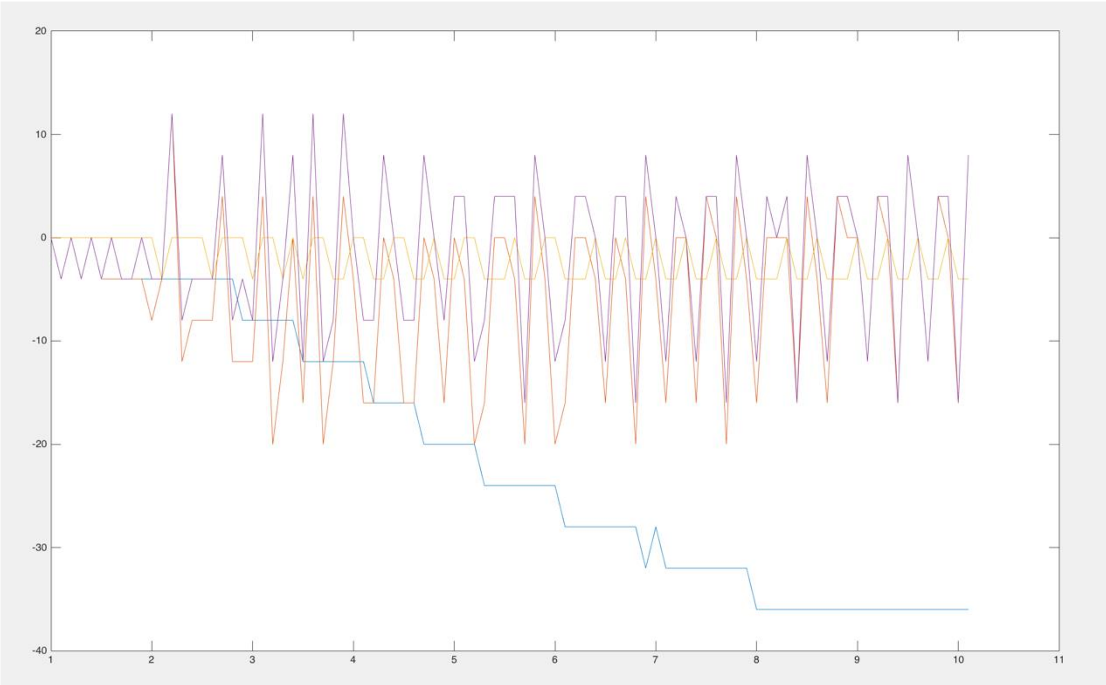
Και την συνάρτηση ADC

```
function [v_adc]=adc(v_in,q,v_min,v_max)
delta=(v_max-v_min)/(2^q);
if v_in<v_min,v_in=v_min;...
    elseif v_in>v_max-delta,v_in=v_max-delta;end
v_adc_b=floor((v_in-v_min)/delta);
v_adc=v_min+v_adc_b*delta;
v_adc_b=dec2bin(v_adc_b,q);
```

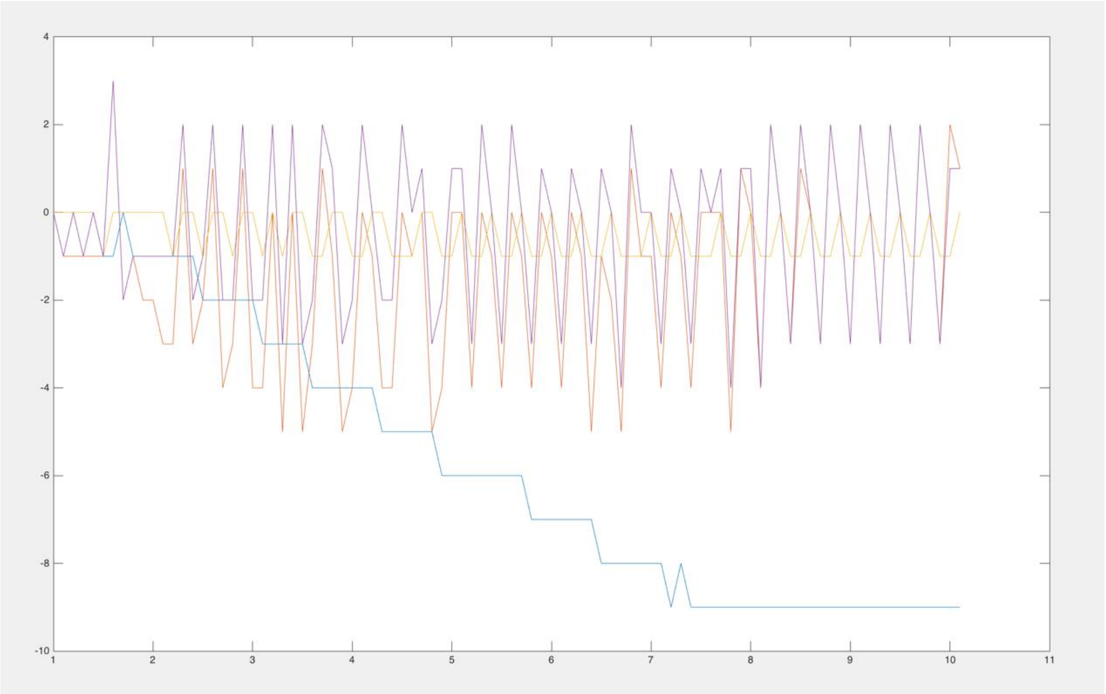
Για διαφορετικές τιμές δειγματοληψίας Ts και διαφορετικά bits έχουμε τα παρακάτω:

Για Ts=0.1

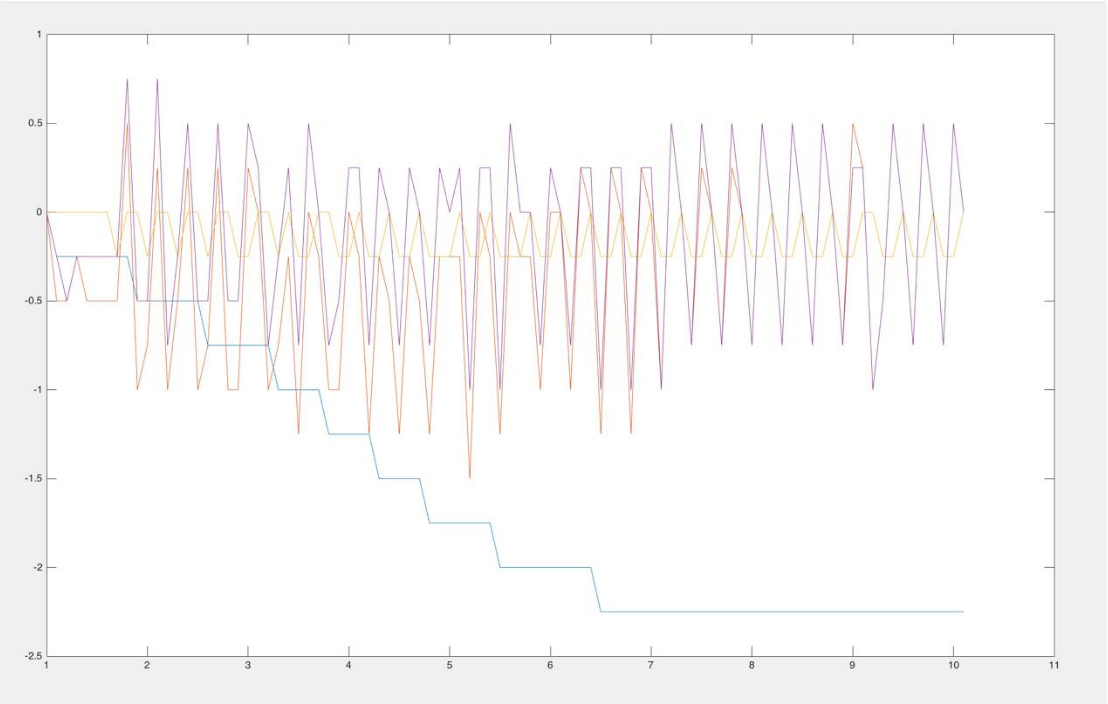
Q=6 bits



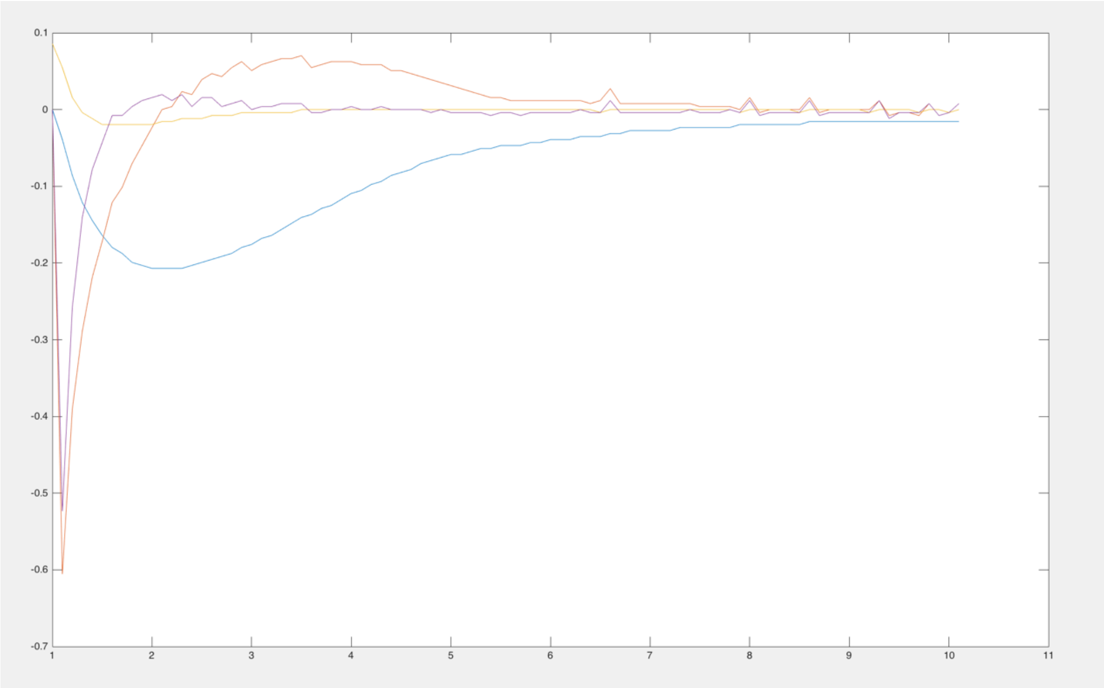
Q=8 bits



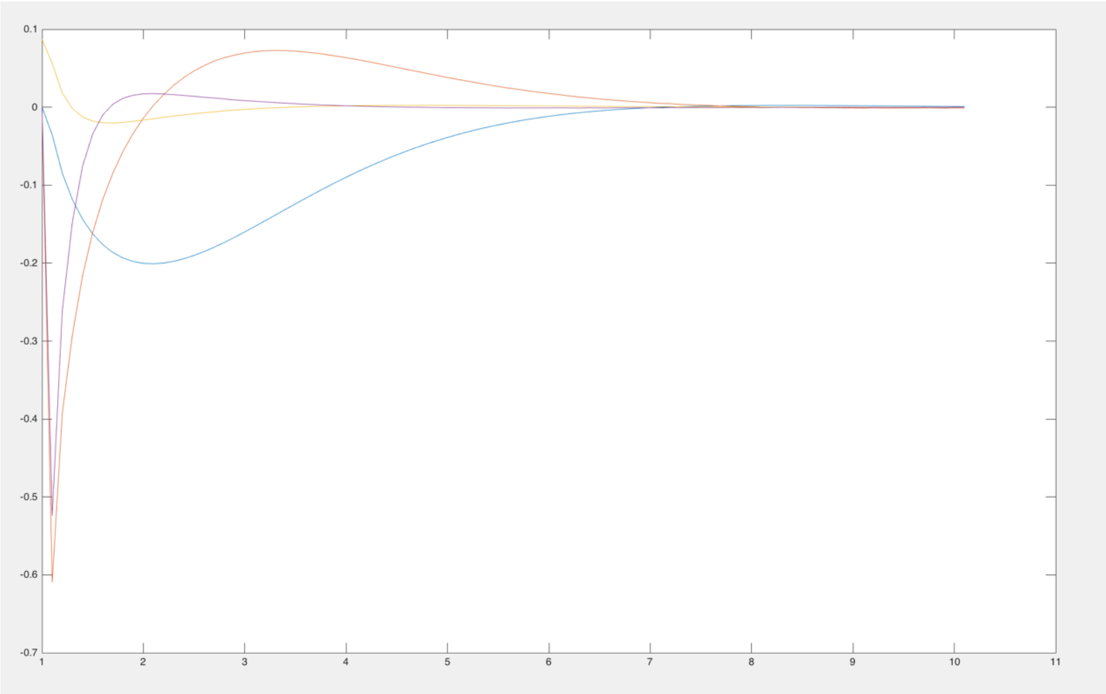
Q=10 bits



Q=16 bits

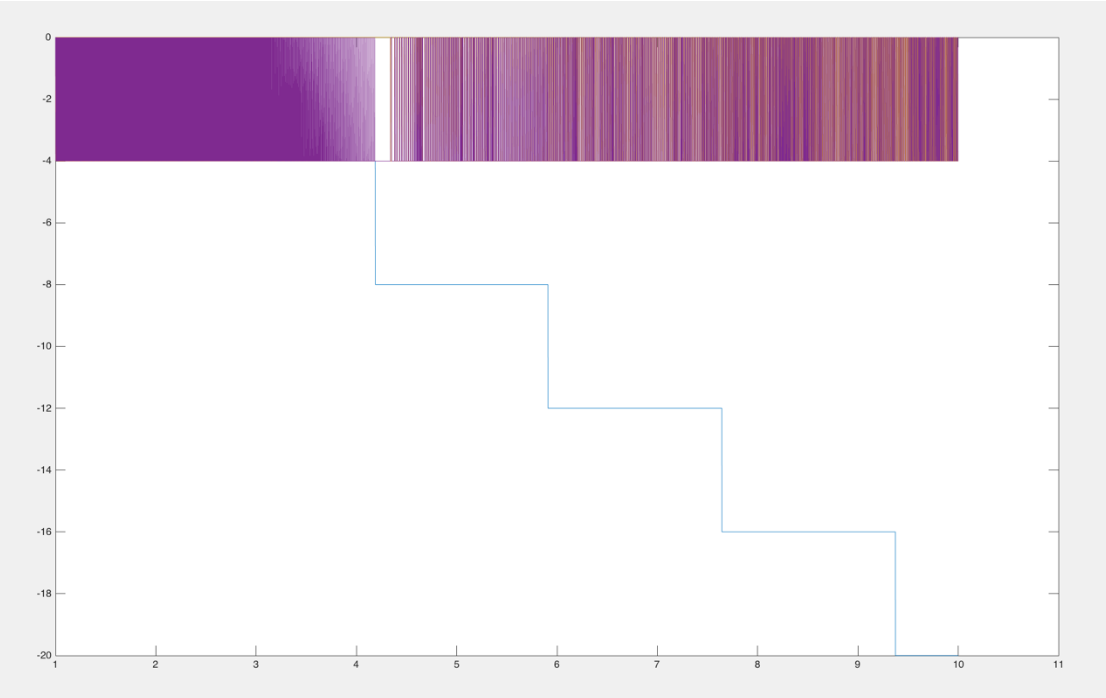


Q=22 bits

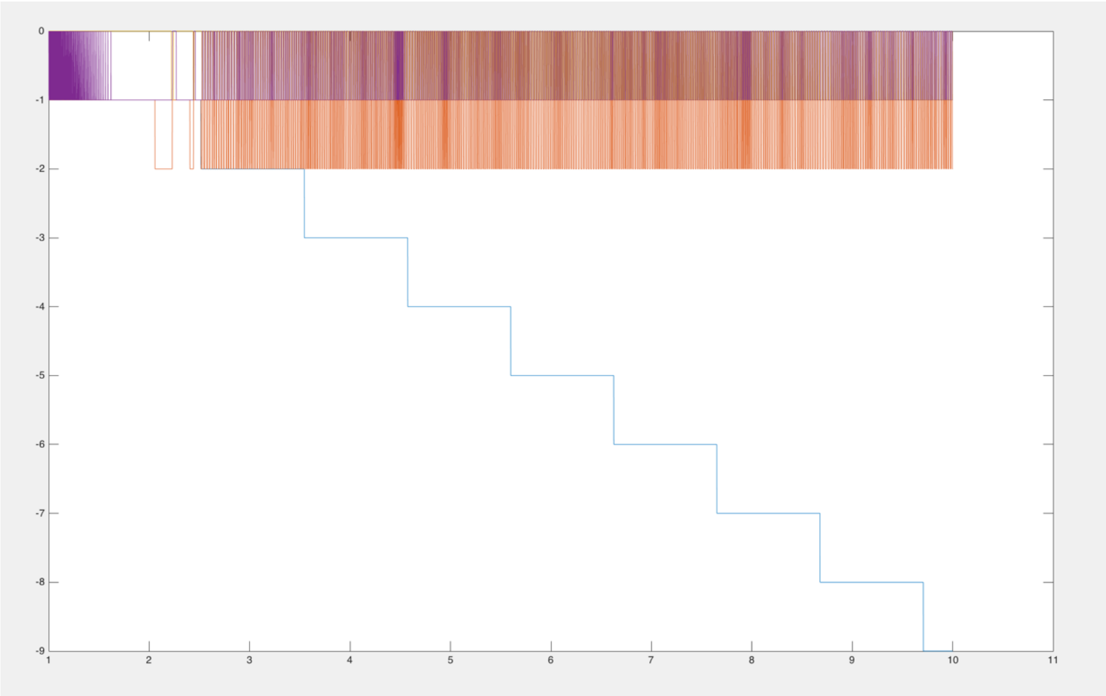


Για Ts=0.001

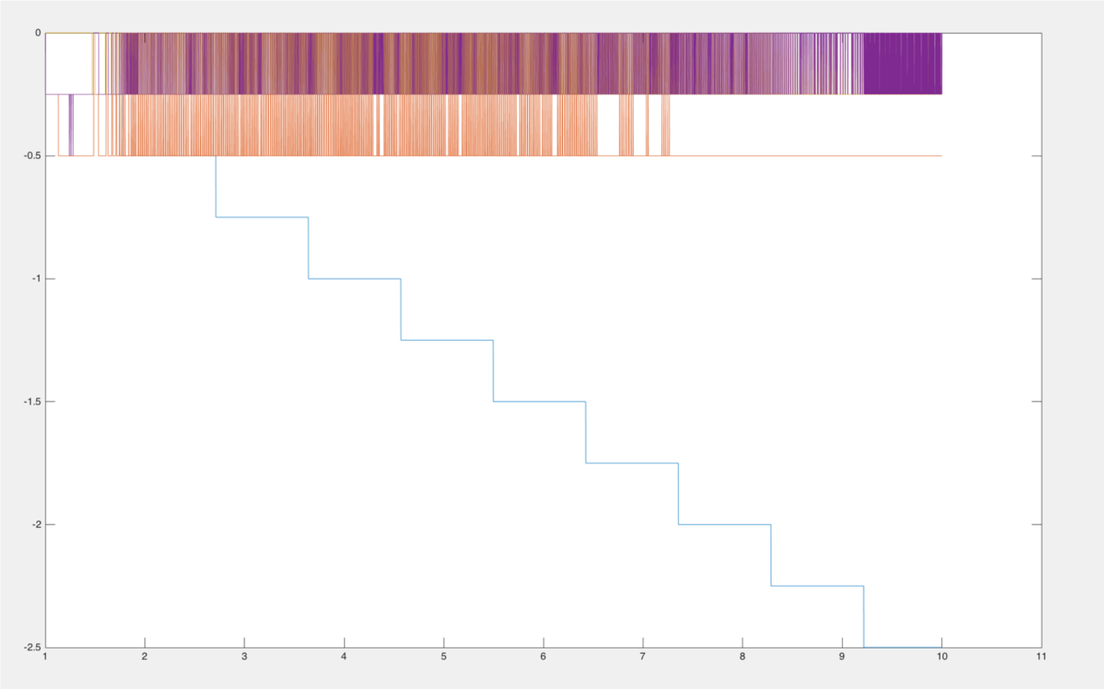
Q=6 bits



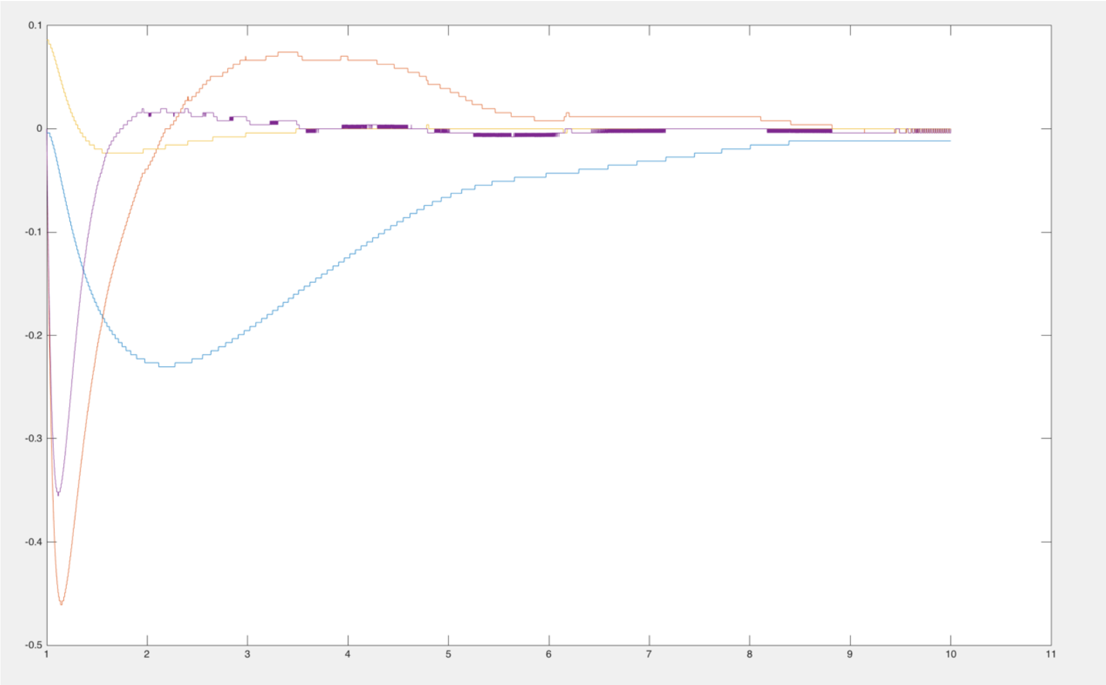
Q=8 bits



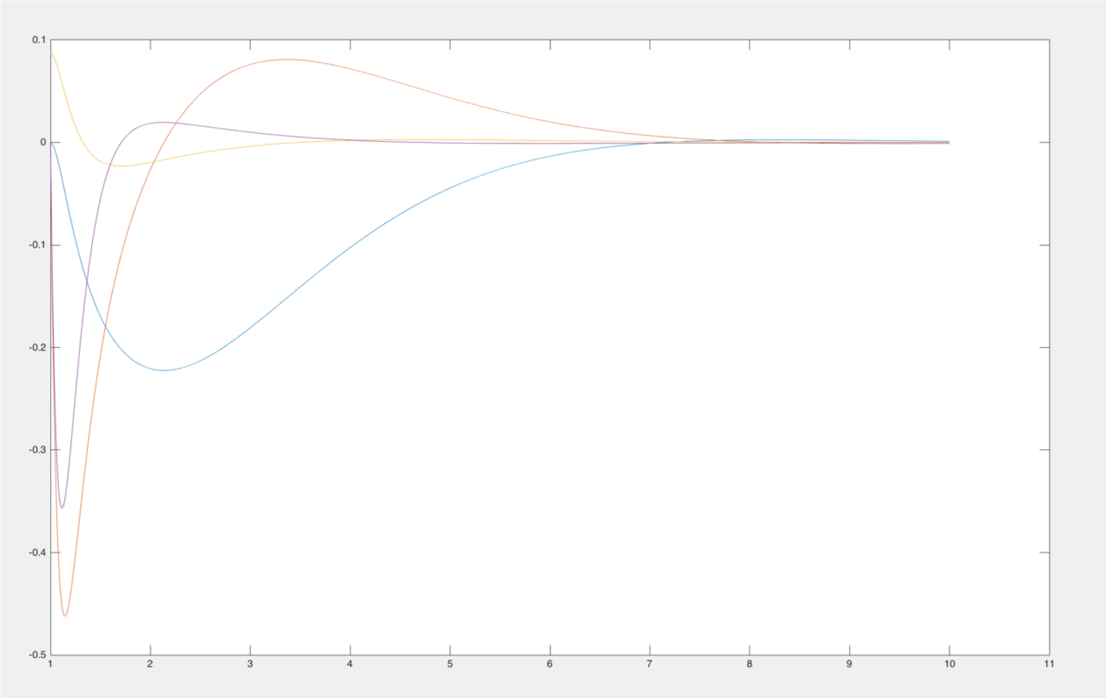
Q=10 bits



Q=16 bits

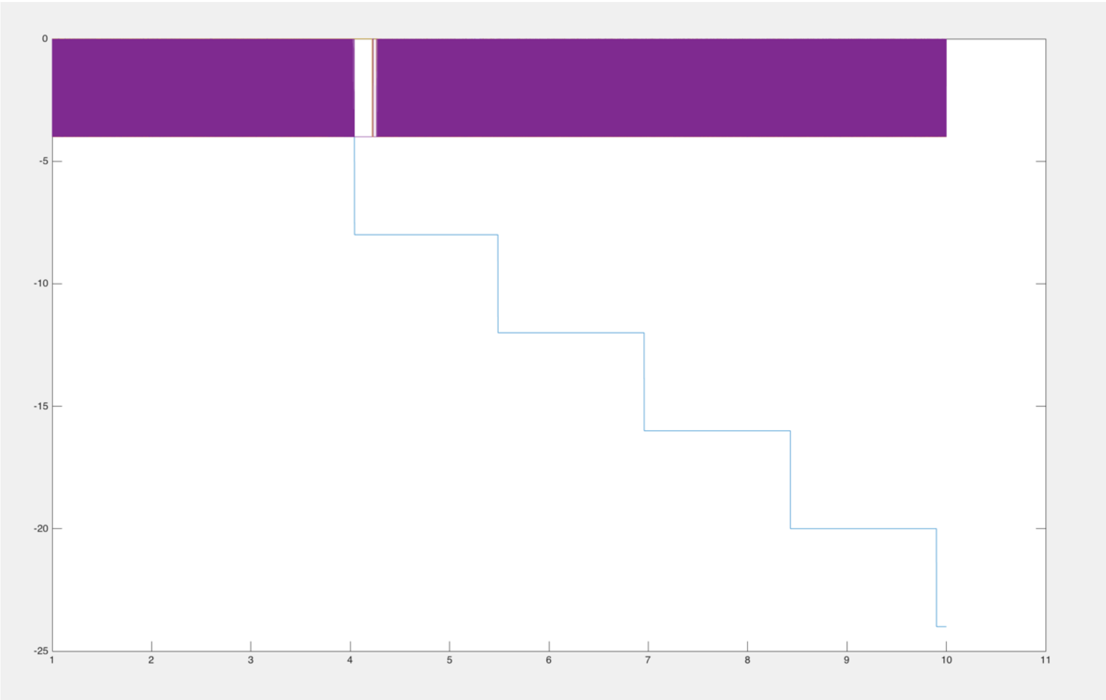


Q=22 bits

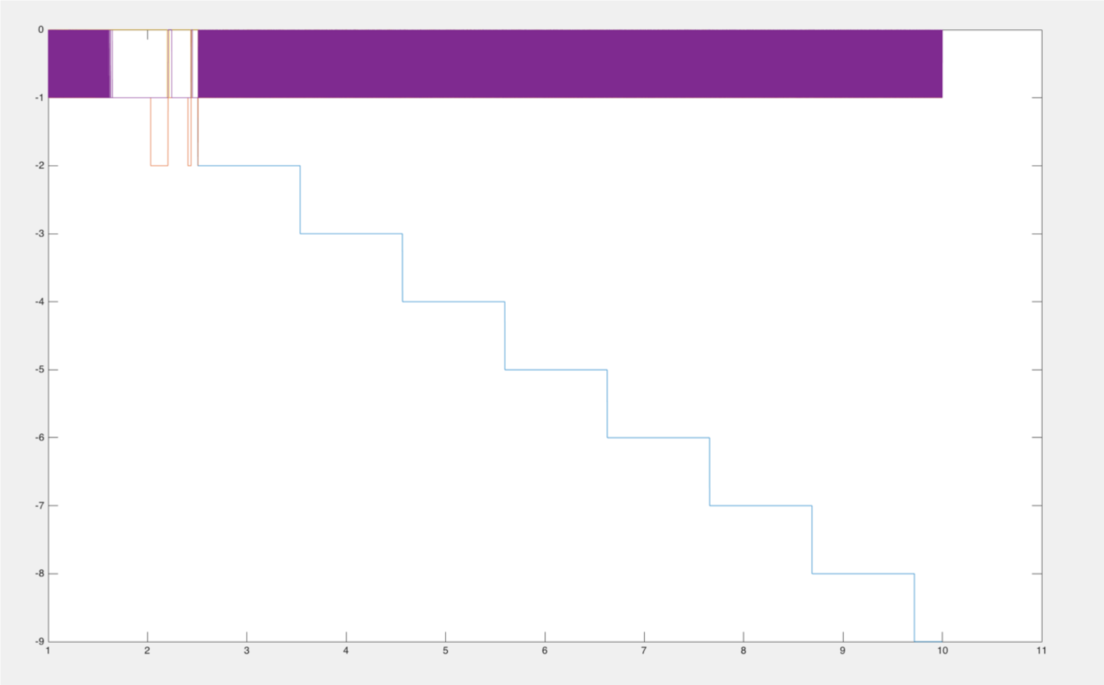


Για Ts=0.0001

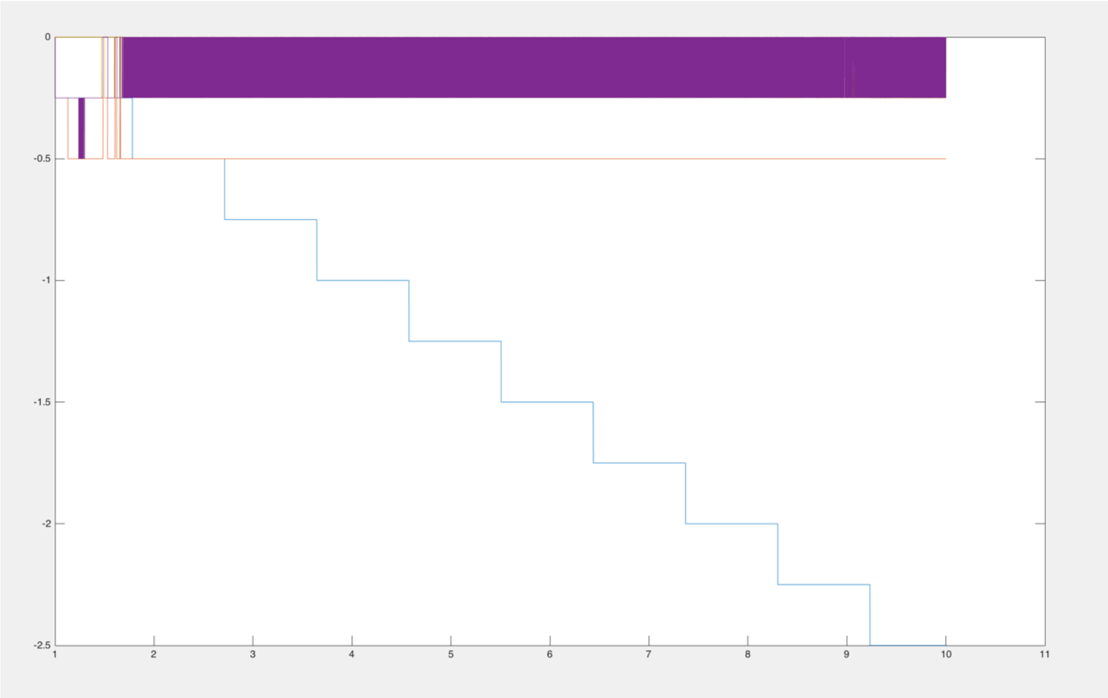
Q=6 bits



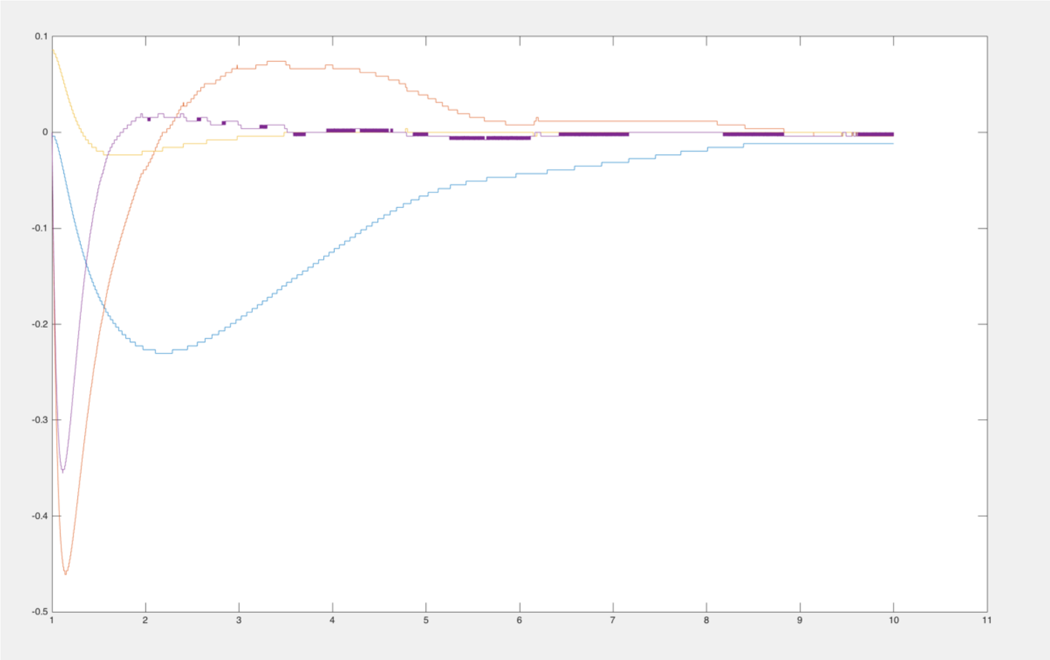
Q=8 bits



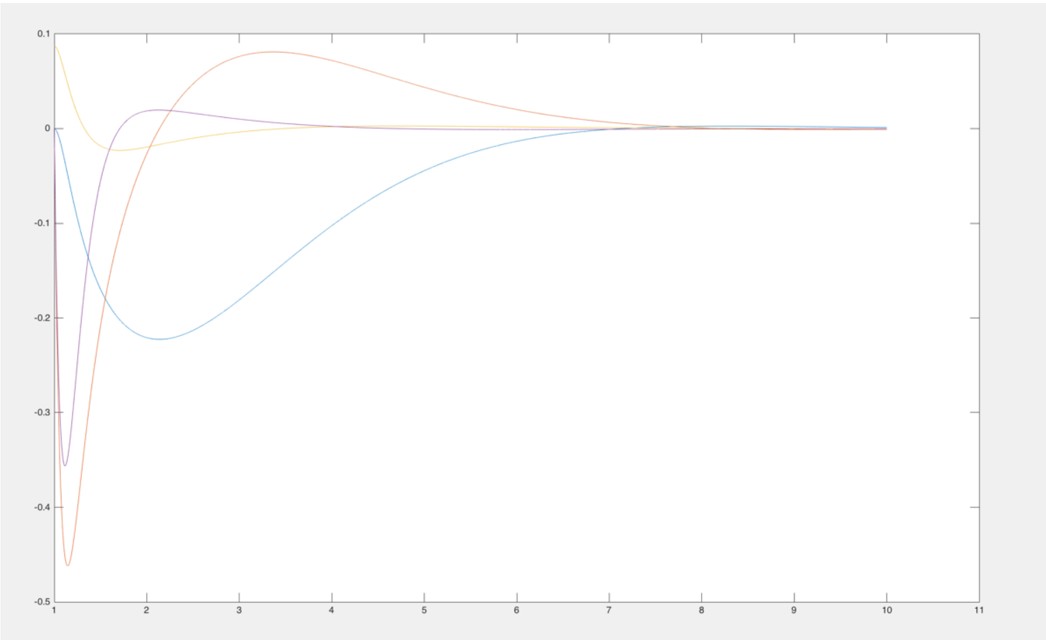
Q=10 bits



Q=16 bits



Q=22 bits

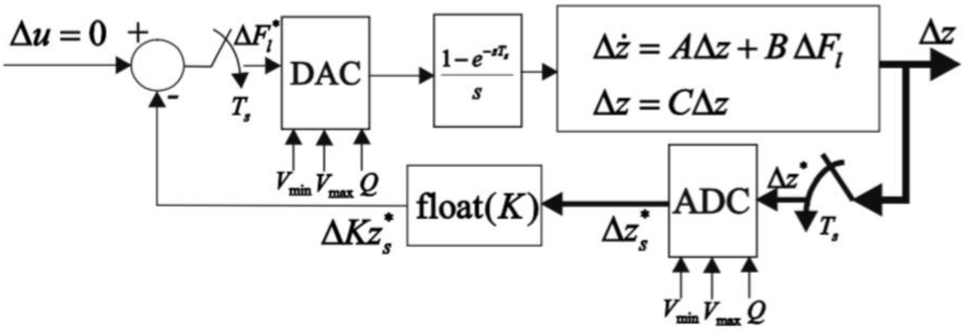


Παρατηρούμε πως όσο αυξάνεται ο αριθμός των bits έχουμε μεγαλύτερη ακρίβεια και μικρότερο σφάλμα σε σχέση με τις τιμές στην είσοδο του μετατροπέα. Ανεξαρτήτως δειγματοληψίας μετά τα 16 bit αρχίζει και σχηματίζεται η μορφή της απόκρισης της διακριτής συνάρτησης μεταφοράς

Καθώς αυξάνεται η περίοδος δειγματοληψίας βλέπω πως έχω μεγαλύτερα σφάλματα στην απόκριση, μιας και μεγαλύτερη δειγματοληψία σημαίνει και μεγαλύτερα «χρονικά άλματα» από την μία τιμή στην επόμενη.

E)

Καλούμαστε να μετατρέψουμε τις τιμές από τον ADC σε αριθμούς κινητής υποδιαστολής m-bits για το mantissa και e-bits για το exponent.



Ο κώδικας που προσθέσαμε στον παραπάνω είναι ο εξής:

```
Q=16;
Vmin=-128;
Vmax=128;
m=5;
e=5;
cnt=1;
z(:,1)=z0;
ADC(:,1)=adc(z(:,1),Q,Vmin,Vmax);
for i=1:4
    ZOUN(i,1)=num2float(ADC(i,1),m,e);
end

for t1=1:Ts:10
    u=-K_f_xdx*ZOUN(:,cnt);
    z(:,cnt+1)=A*z(:,cnt)+B*u;
    ADC(:,cnt+1)=adc(z(:,cnt+1),Q,Vmin,Vmax);

    for i=1:4
         ZOUN(i,cnt+1)=num2float(ADC(i,cnt+1),m,e);
    end
    cnt=cnt+1;
end
t1=1:Ts:10+Ts;
plot(t1,ZOUN);
```

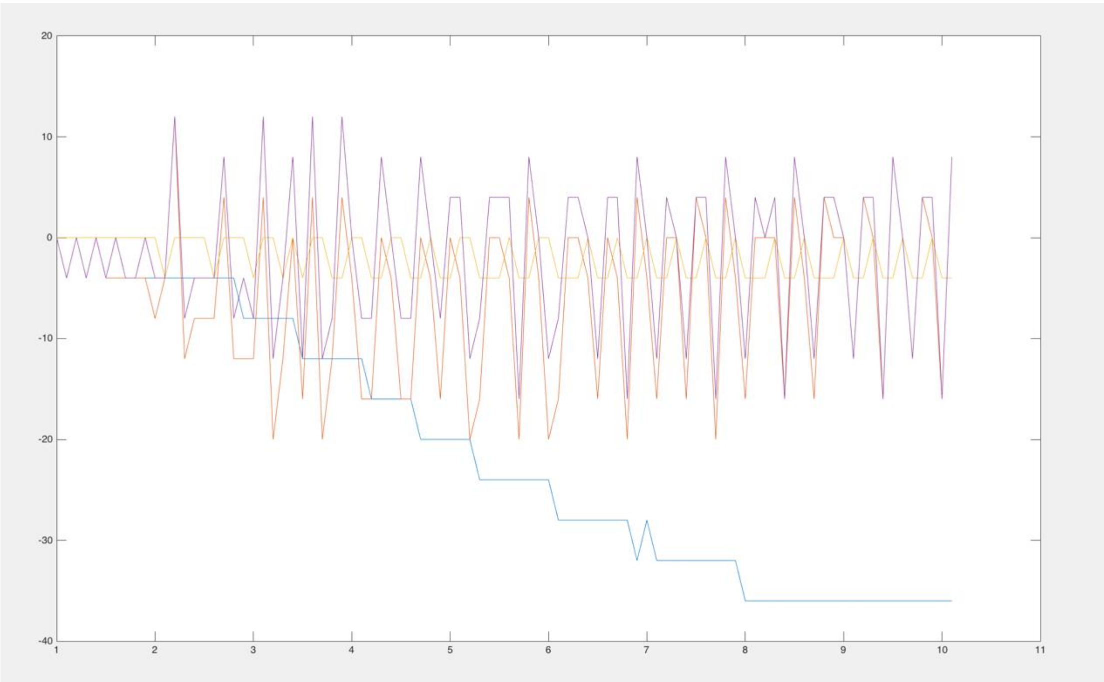
και η συνάρτηση numtofloat

```
function [x_float]=num2float(x,m,e)
e_number=0;
sm=0;for i=1:m-1;sm=sm+1/2^i;end
for j=-2^(e-1)-1:2^(e-1)-1;...
 if x==0,e_number=0;
 elseif x>sm*2^(e-1)-1,e_number=2^(e-1)-1;
 elseif and(0.5<=abs(x)/(2^j),abs(x)/(2^j)<1),e_number=j;end;...
end
if e_number>=0,e_number_b=strcat('0',dec2bin(e_number,e-1));end
if e_number<0, e_number_b=strcat('1',dec2bin(-e_number,e-1));end
m_number=round((abs(x)/(2^e_number))*2^(m-1));
if x>0,m_number_b=strcat('0',dec2bin(m_number,m-1));end
if x<0,m_number_b=strcat('1',dec2bin(m_number,m-1));end
if x==0,m_number_b=strcat('0',dec2bin(0,m-1));end
mantissa_exponent=strcat(m_number_b,e_number_b);
x_float=sign(x)*(m_number*2^(-m+1))*(2^e_number);
```

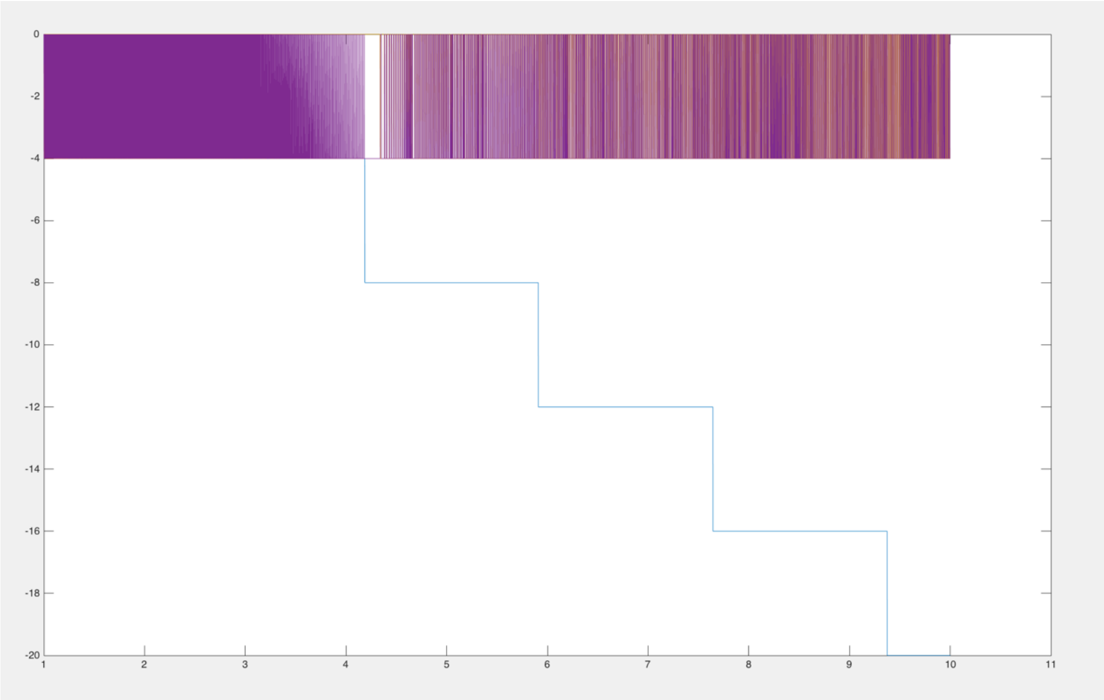
Για διάφορες τιμές Ts, Q και m,e-bits έχουμε:

Για (m,e)=(5,5)

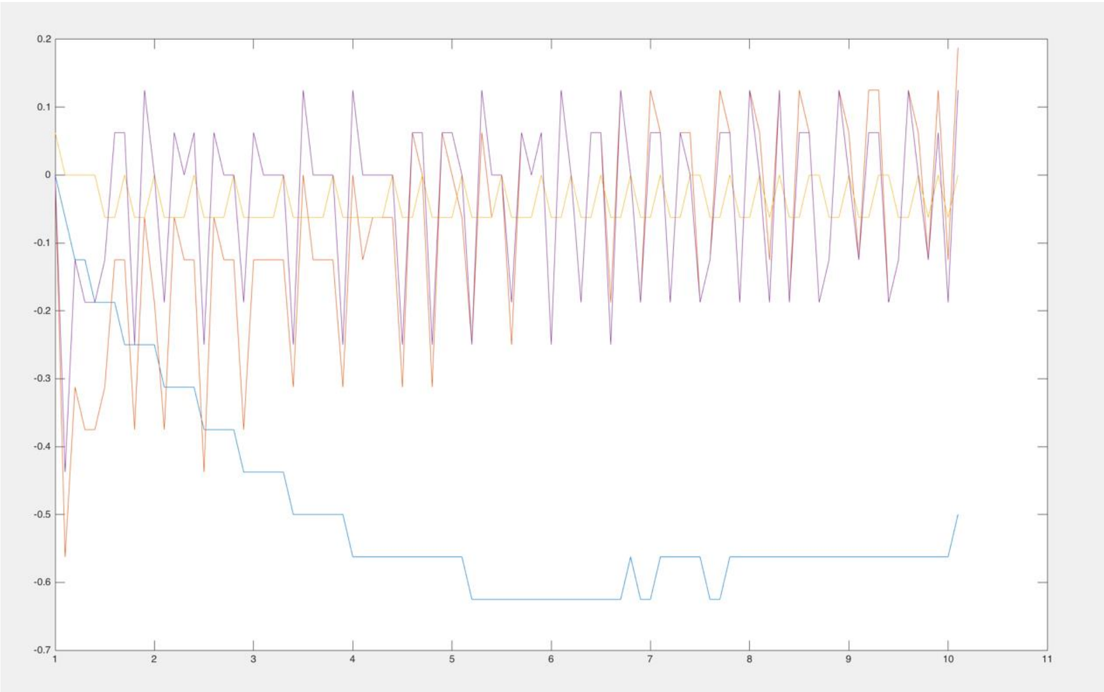
- **Ts=0.1 και Q=6**



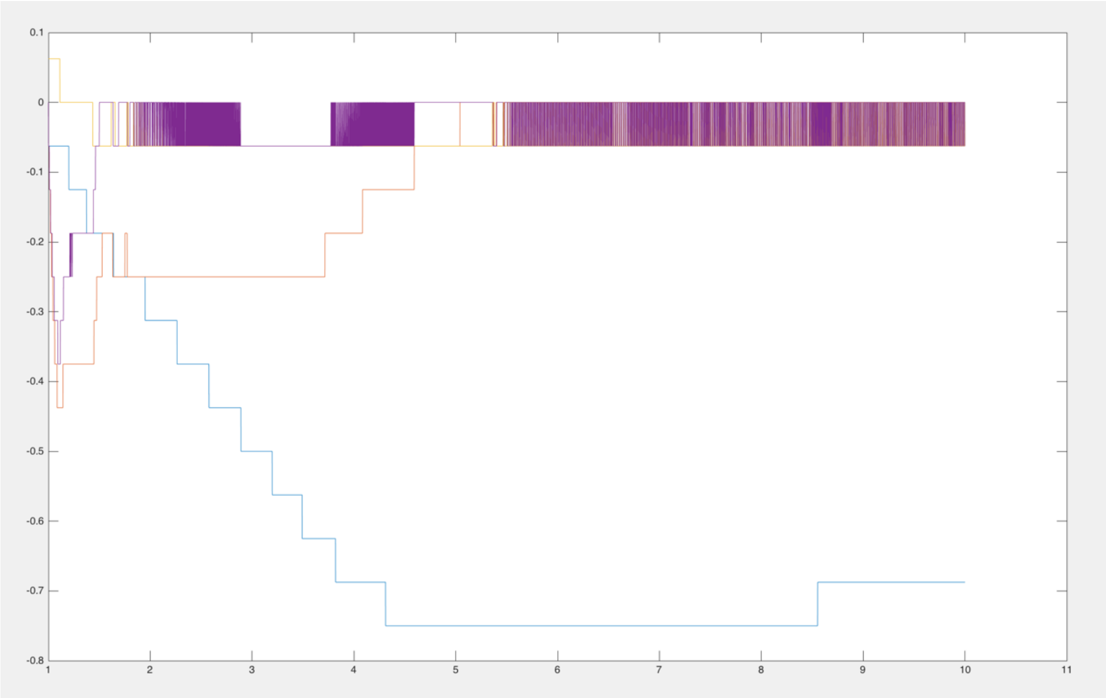
- **Ts=0.001 και Q=6**



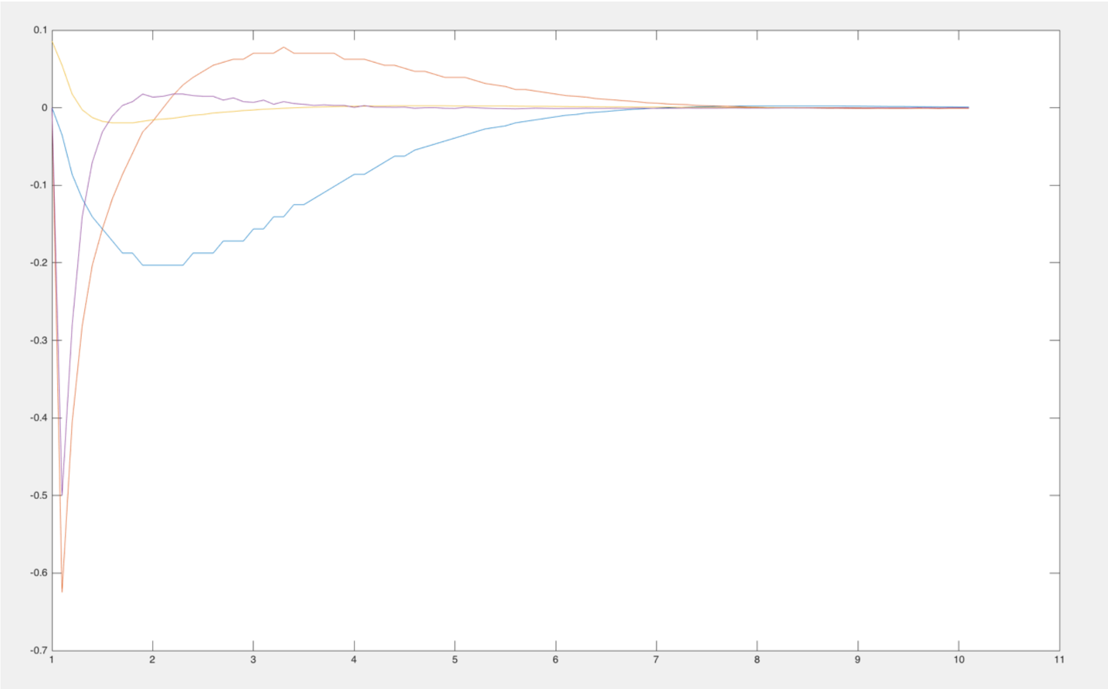
- **Ts=0.1 και Q=12**



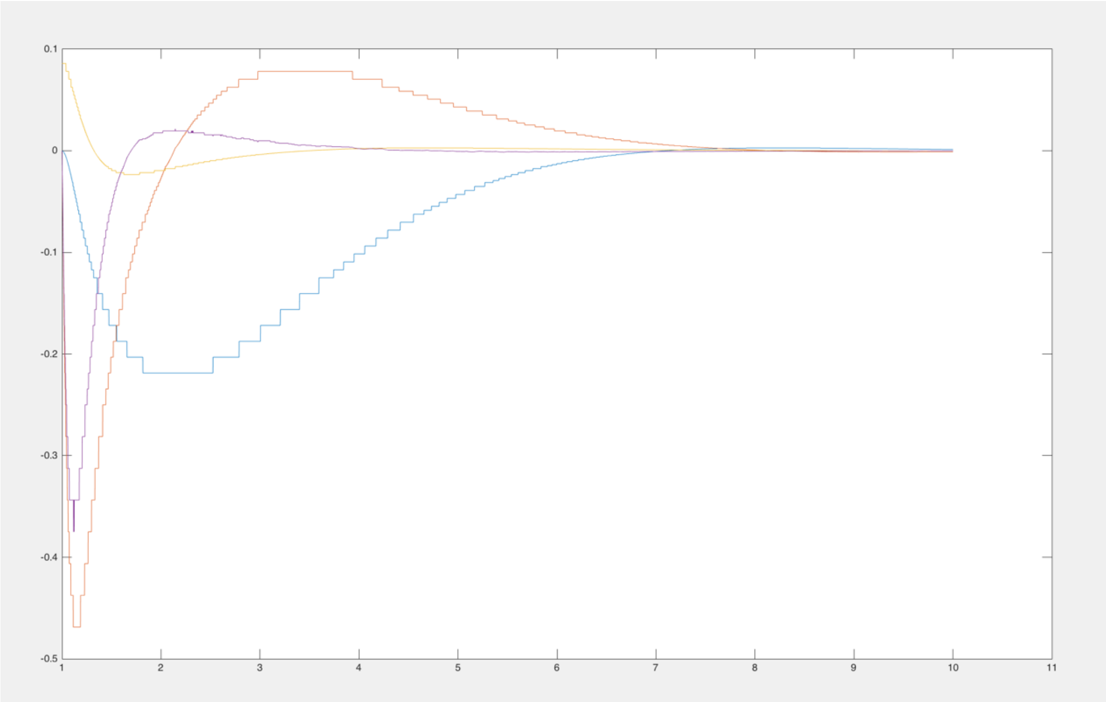
- **Ts=0.001 και Q=12**



- **Ts=0.1 και Q=22**

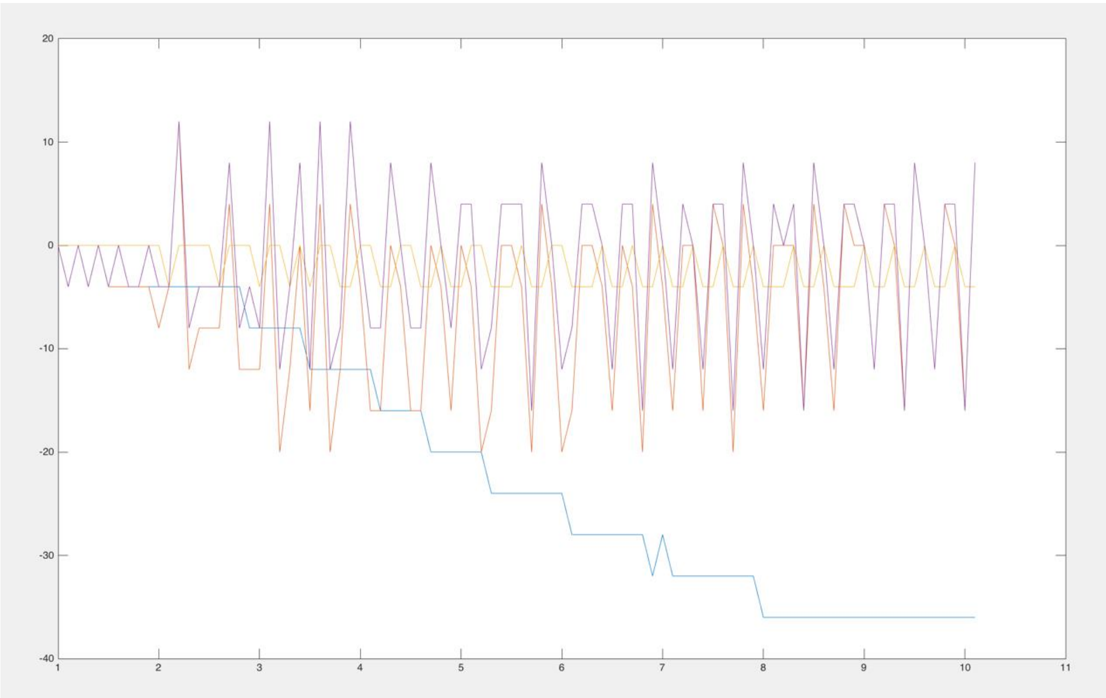


➢ **Ts=0.001 και Q=22**

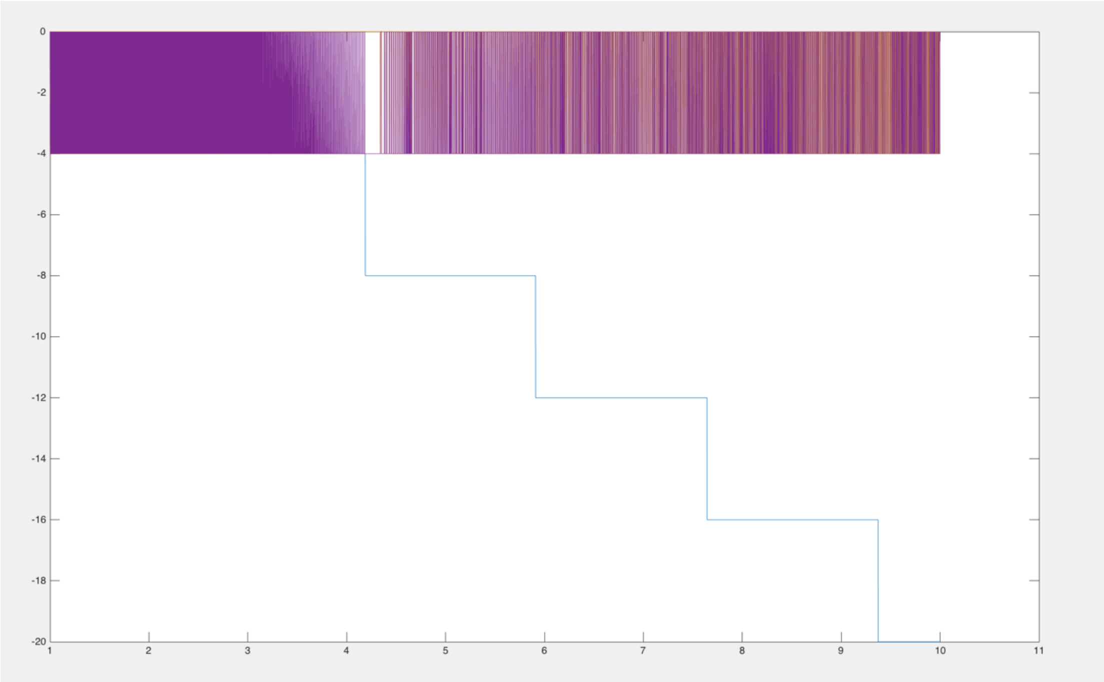


***Για (m,e)*=(8,8)**

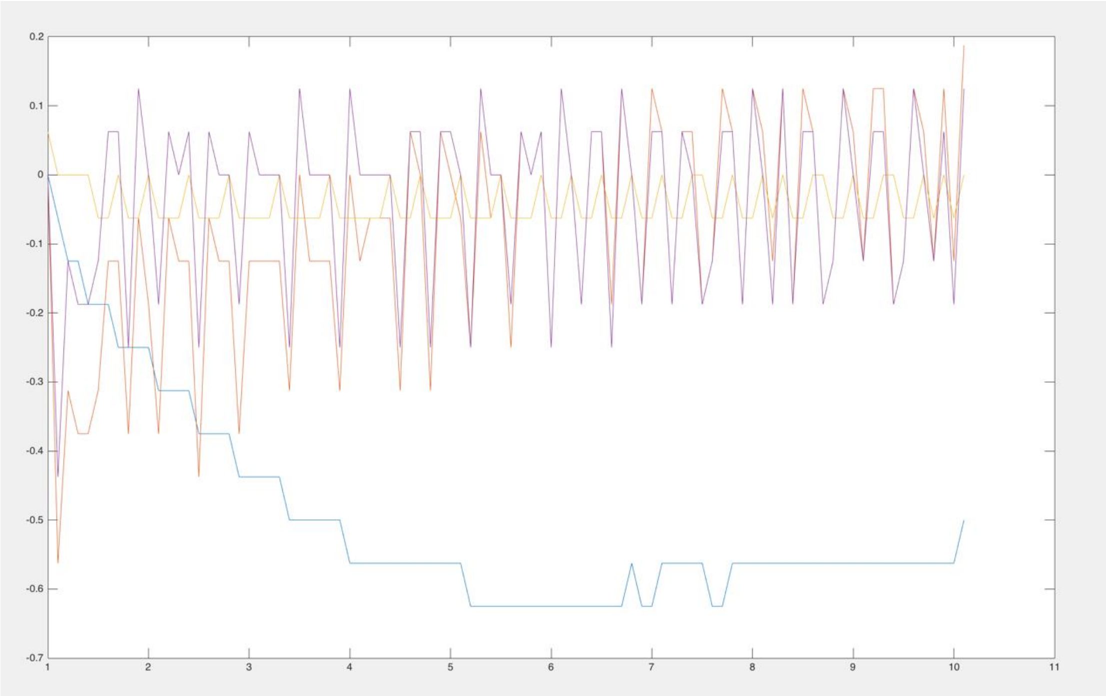
➢ **Ts=0.1 και Q=6**



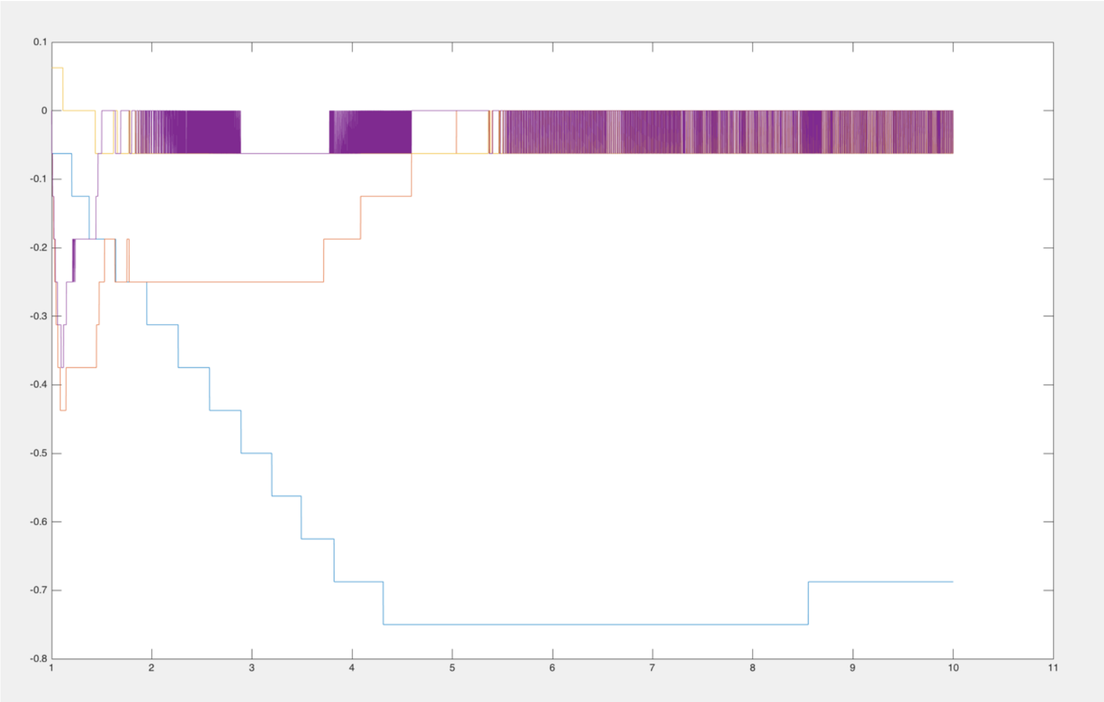
- **Ts=0.001 και Q=6**



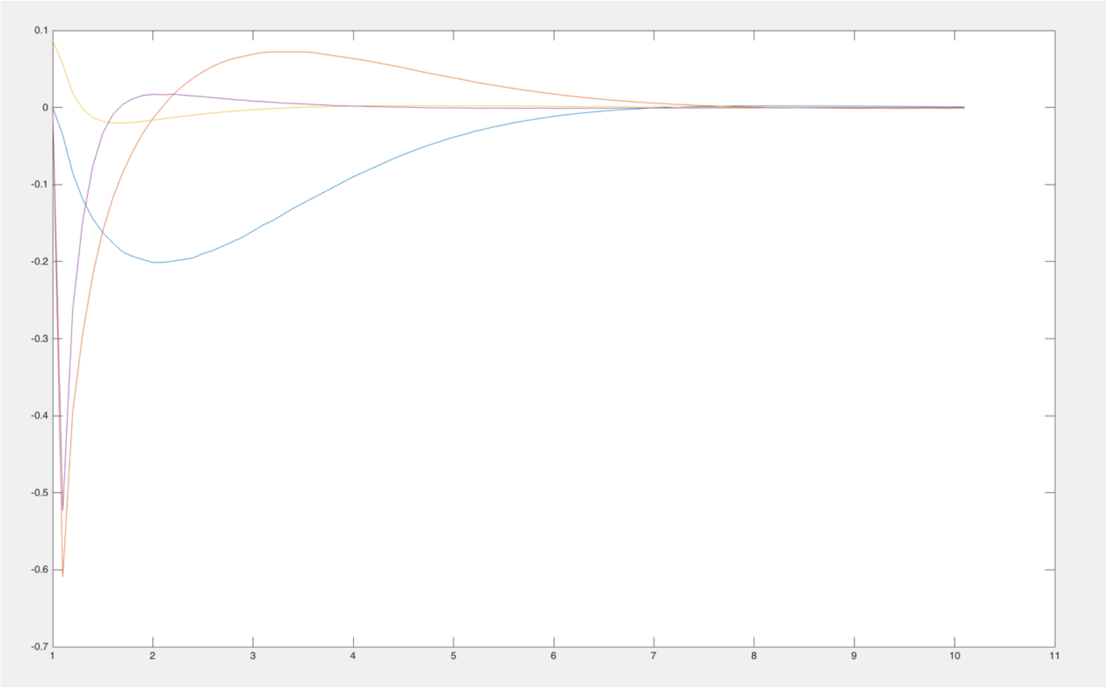
- **Ts=0.1 και Q=12**



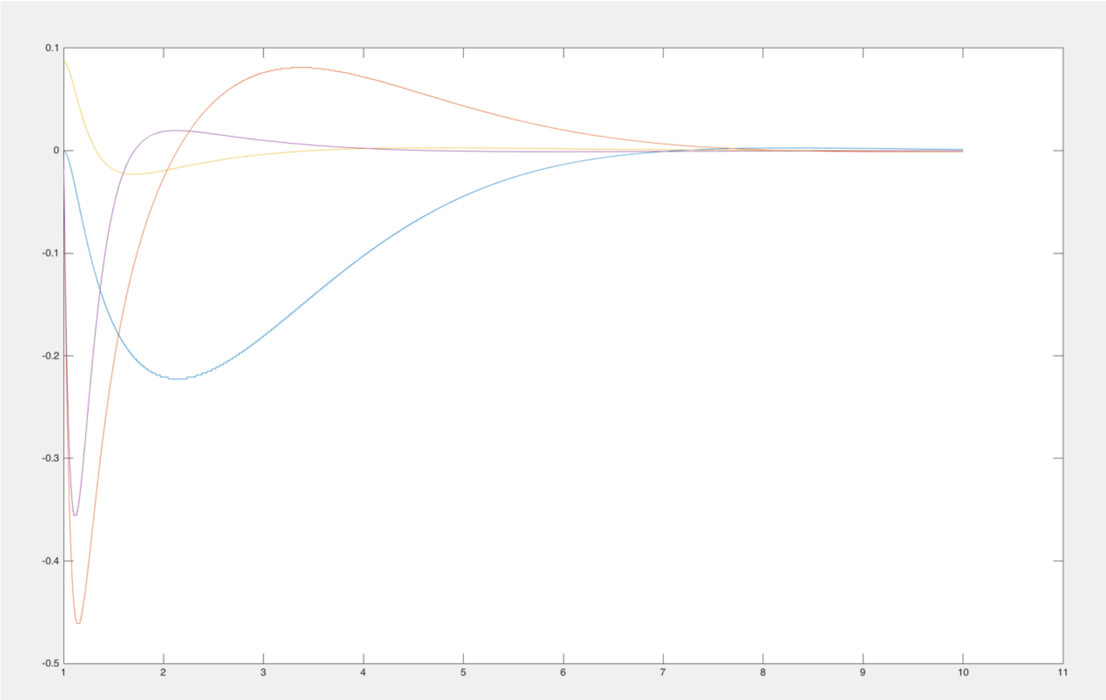
- **Ts=0.001 και Q=12**



- **Ts=0.1 και Q=22**

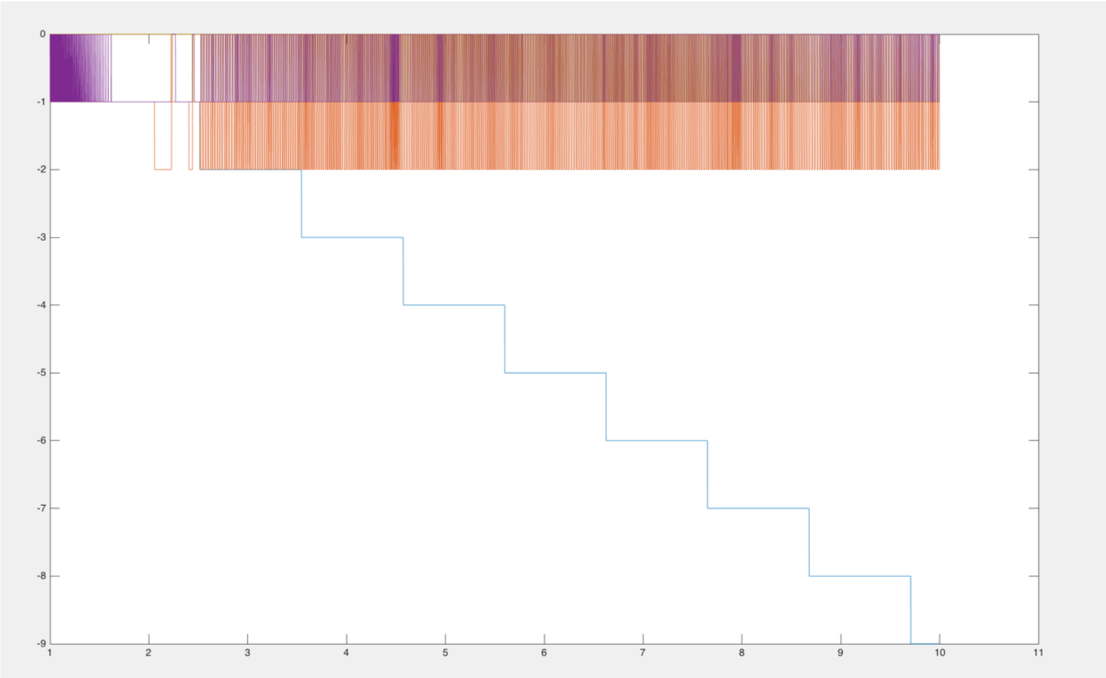


- **Ts=0.001 και Q=22**

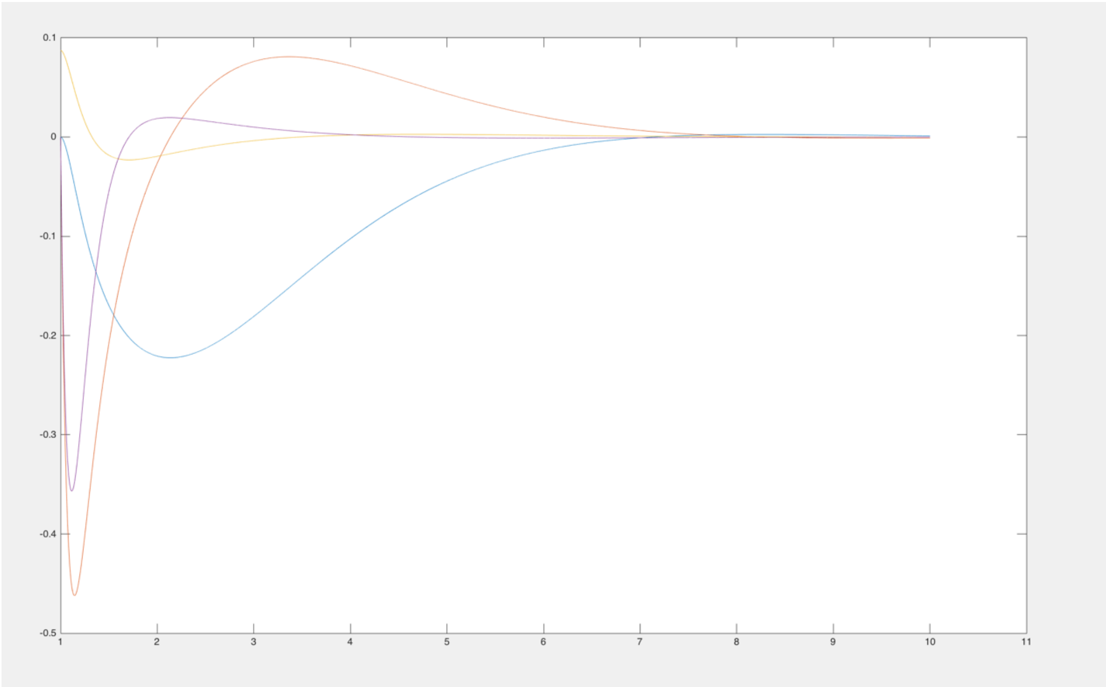


Για (m,e)=(12,12)

- **Ts=0.001 και Q=8**



- **Ts=0.001 και Q=22**



Παρατηρούμε πως όσα περισσότερα bit mantissas και exponent χρησιμοποιούμε θα έχουμε μικρότερο σφάλμα κβάντισης και άρα μεγαλύτερη ακρίβεια.