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

Εργαστήριο Ψηφιακών Συστημάτων Ελέγχου
Ακαδημαϊκό Έτος 2015-2016

Γιώργος Μπολάτογλου

PID-Έλεγχος Κινούμενου Ανεστραμμένου Εκκρεμούς

ΕΚΤΕΛΕΣΗ ΑΣΚΗΣΗΣ

Προσομοιώστε το γραμμικοποιημένο σύστημα του ανεστραμμένου κινούμενου εκκρεμούς χρησιμοποιώντας τις ακόλουθες τιμές των παραμέτρων M=1;m=1;l=1;B_l=0.3;B_r=0;g=10.

3.1 Υπολογισμός συνάρτησης μεταφοράς

Στον κώδικα που είχαμε κατασκευάσει στην προηγουμένη άσκηση καναμε τις καταλληλες μετατροπες ετσι ωστε να κατασκευασουμε την συναρτηση μεταφορας :

$$\frac{\Delta\theta}{\Delta F1}(s) = \frac{\frac{s}{Ml}}{s^3 + \frac{s^2 Bl}{Ml} + \frac{s(M+m)(-g)}{Ml} + \frac{Bl(-g)}{Ml}}$$

$$\boldsymbol{\frac{\Delta\theta}{\Delta F1}(s) = \frac{s}{(\ s\hat{}3\ +\ 0.3\,s\hat{}2\ -\ 20\,s\ -\ 3)}}$$

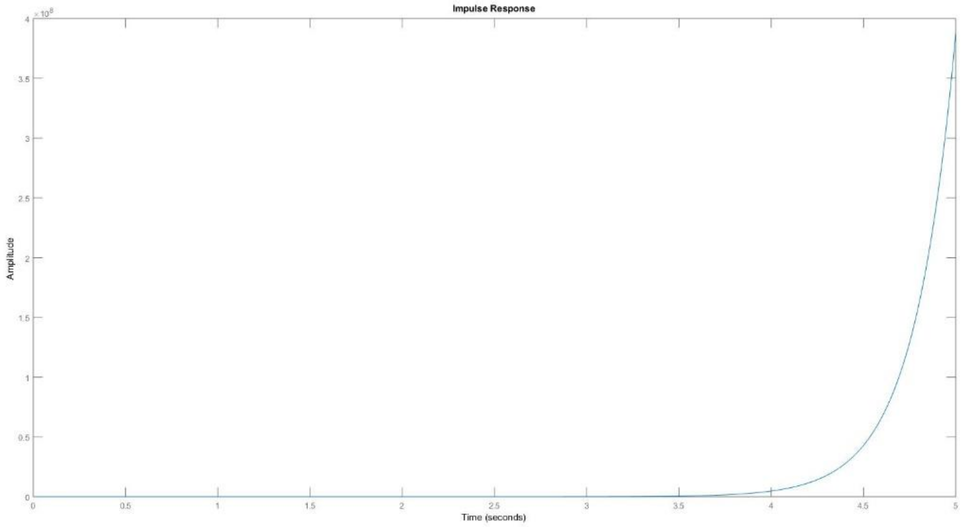
```
num=[1/M*l 0]
den=[1 (B_l/M) ((M+m)*(-g))/(M*l) (B_l*(-g))/M*l ]
T=tf(num,den);
```

Pole	Damping	Frequency (rad/seconds)	Time Constant (seconds)
-1.50e-01	1.00e+00	1.50e-01	6.67e+00
4.40e+00	-1.00e+00	4.40e+00	-2.27e-01
-4.55e+00	1.00e+00	4.55e+00	2.20e-01

3.2 Υπολογισμός κρουστικής απόκρισης

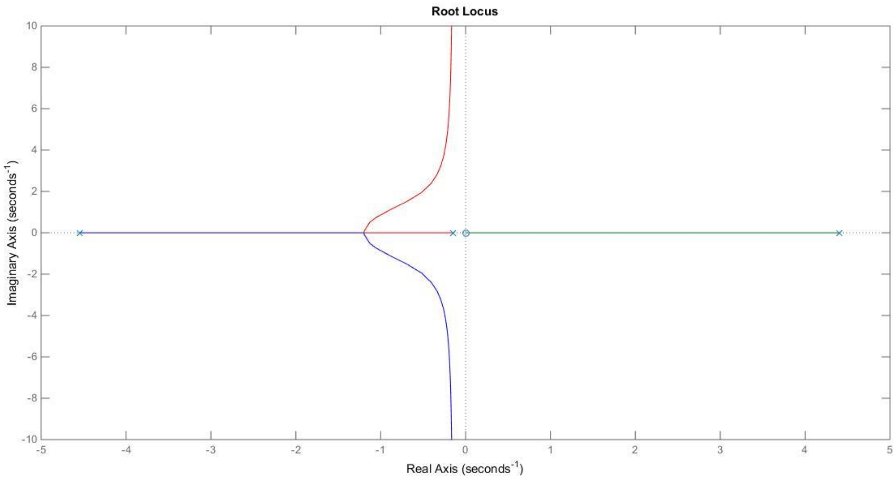
Ορίζοντας τον χρόνο από 0 μεχρι 5 δευτερολεπτα και με βημα 0,01 τρέχουμε τον κώδικα και μέσω της εντολής «impulse», παίρνουμε την παρακάτω γραφική παράσταση της απόκρισης του συστήματος:

```
t=0:0.01:5;
num=[1/M*l 0]
den=[1 (B_l/M) ((M+m)*(-g))/(M*l) (B_l*(-g))/M*l ]
T=tf(num,den);
impulse(T,t);
```



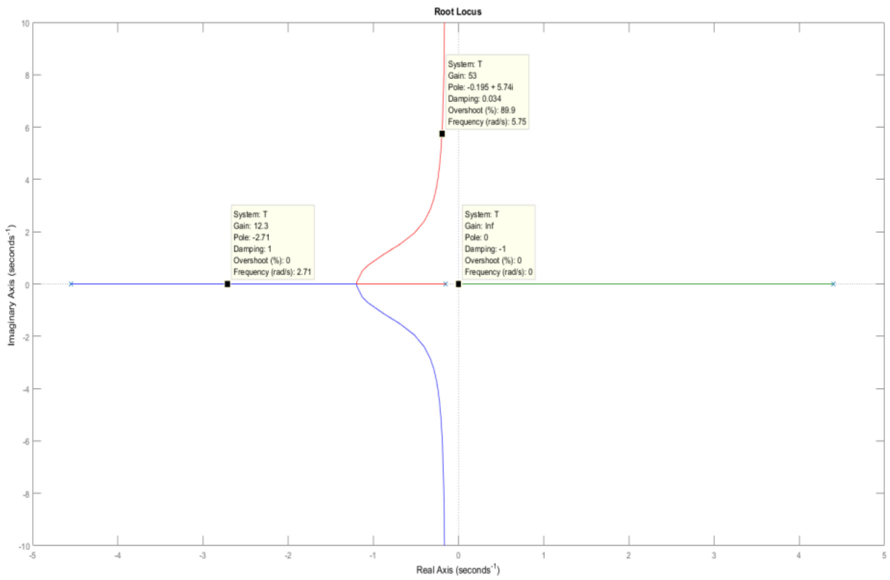
Βλέπουμε πως η κρουστική απόκριση της συνάρτησης μεταφορας μετα απο κάποιο χρόνο απειρίζειται. Κάτι τέτοιο είναι ένδειξη αστάθειας. Χρησιμοποιούμε τον γεωμετρικό τόπο ριζών για να το επιβεβαιώσουμε.

rlocus(T);

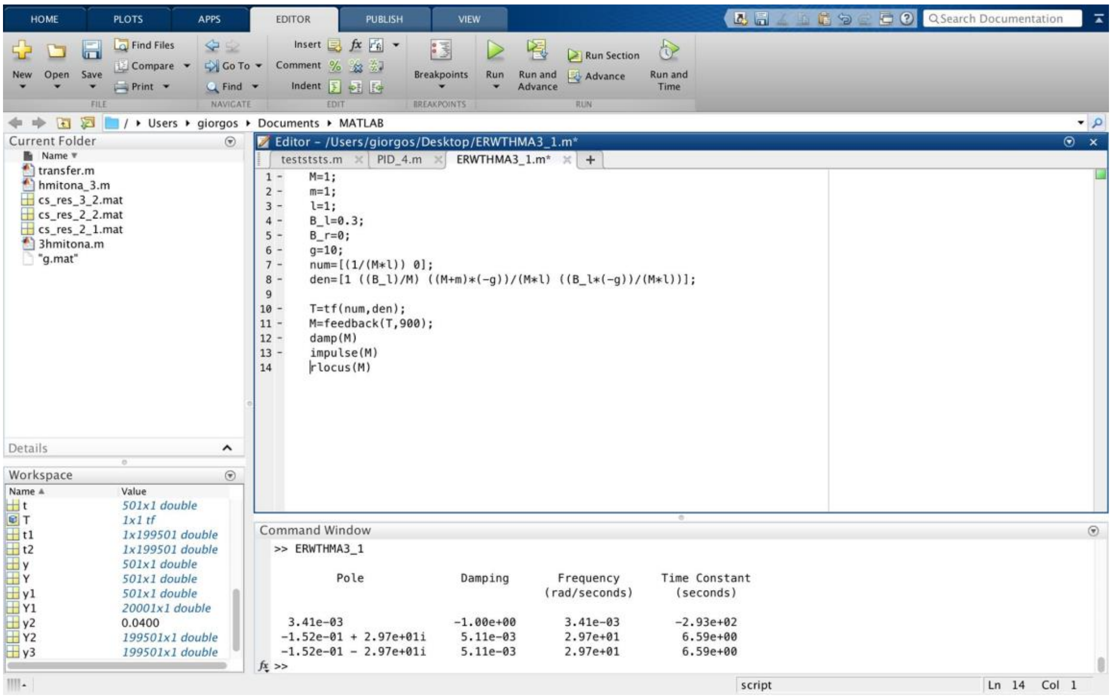


Από τον γεωμετρικό τόπο ριζών βλέπουμε πως ο ένας πόλος της συνάρτησης μεταφοράς βρίσκεται δεξιά του κατακόρυφου άξονα. Αυτό δηλώνει πως το σύστημά μας είναι ασταθές.

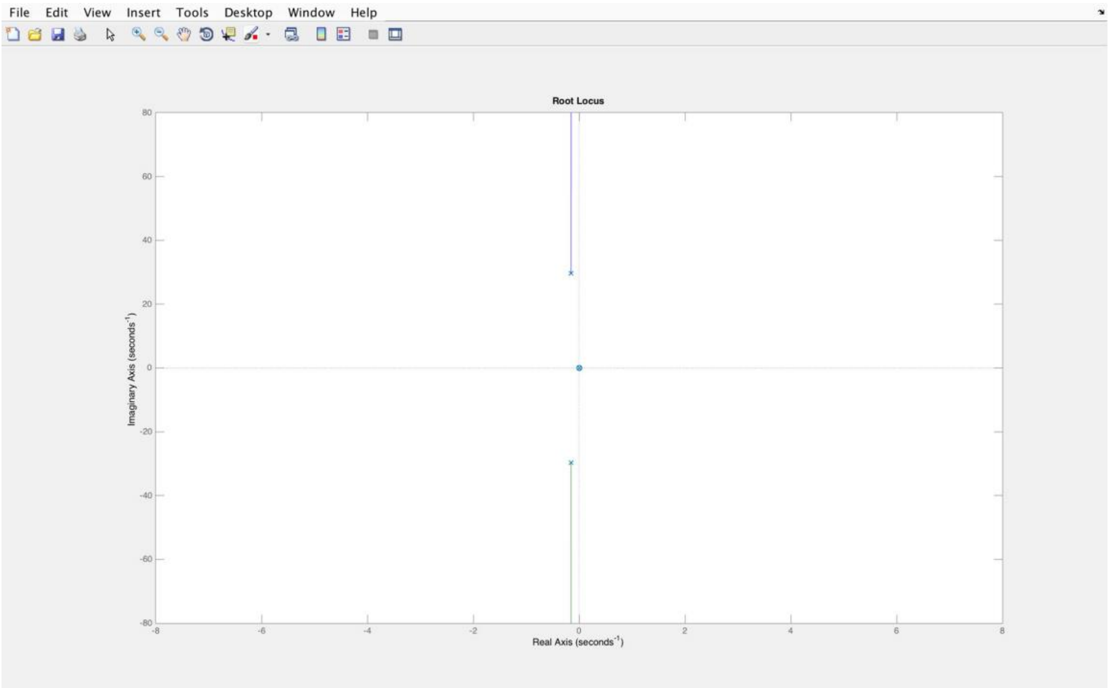
3.3 Ελεγκτής κέρδους



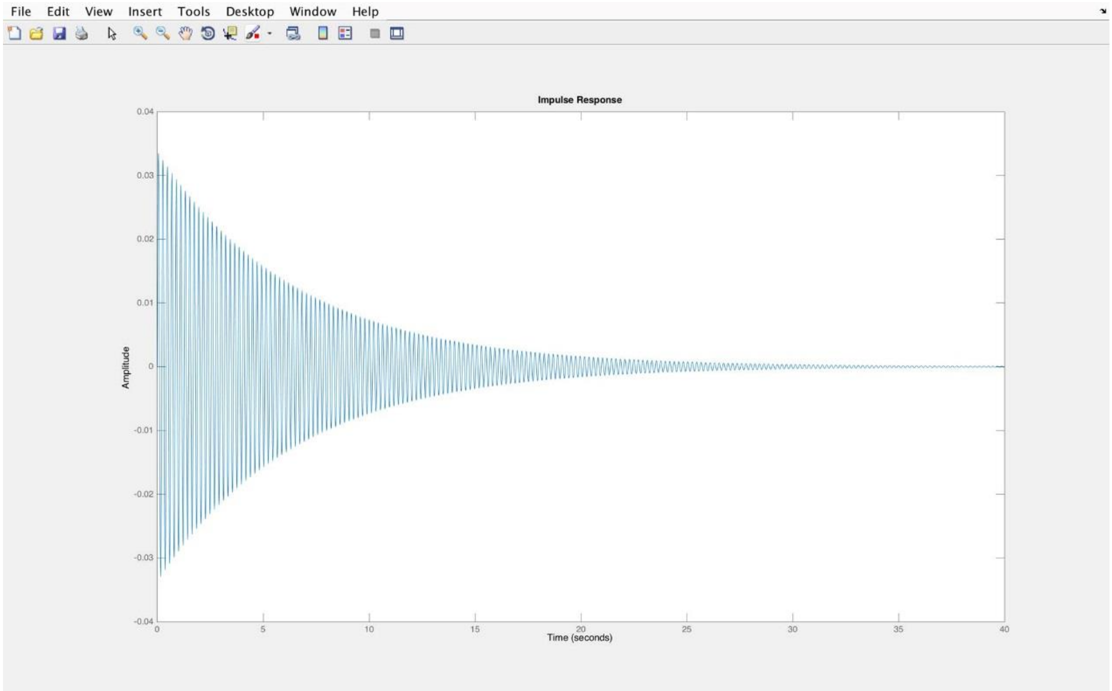
Παρατηρούμε ότι ο ένας πόλος βρίσκεται στον θετικό άξονα, άρα το σύστημα μας είναι ασταθές. Για να πάει στην ευστάθεια το σύστημα πρέπει όλοι οι πόλοι του συστήματος να βρίσκονται αριστερά του κάθετου άξονα. Κάτι τέτοιο δεν γίνεται να επιτευχθεί με έναν ελεγκτή κέρδους. Ακόμα και με άπειρο gain, πρακτικά, ο δεξιός πόλος θα τείνει στο 0 ερχόμενος από τα δεξιά, πράγμα το οποίο τον θέτει συνέχεια στο δεξί ημιεπίπεδο, άρα το σύστημά μας θα βρίσκεται συνέχεια στην αστάθεια.



Υλοποίηση ελεγκτή κέρδους K=900.



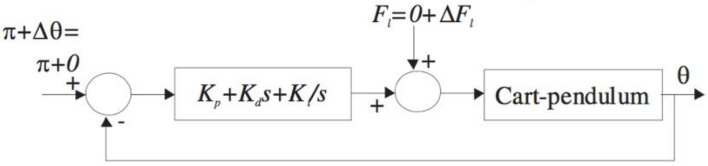
Παρατηρούμε πώς για ένα πολύ μεγάλο κέρδος (K=900) ο δεξιός πόλος θα ταυτιστεί με το 0 και θα έχουμε zero-pole cancelation.



Απ’ την impulse response βλέπουμε πως θεωρητικά το σύστημά μας είναι ευσταθές αλλά με πολλές ανεπιθύμητες ταλαντώσεις.

3.4 PID-Ελεγκτής

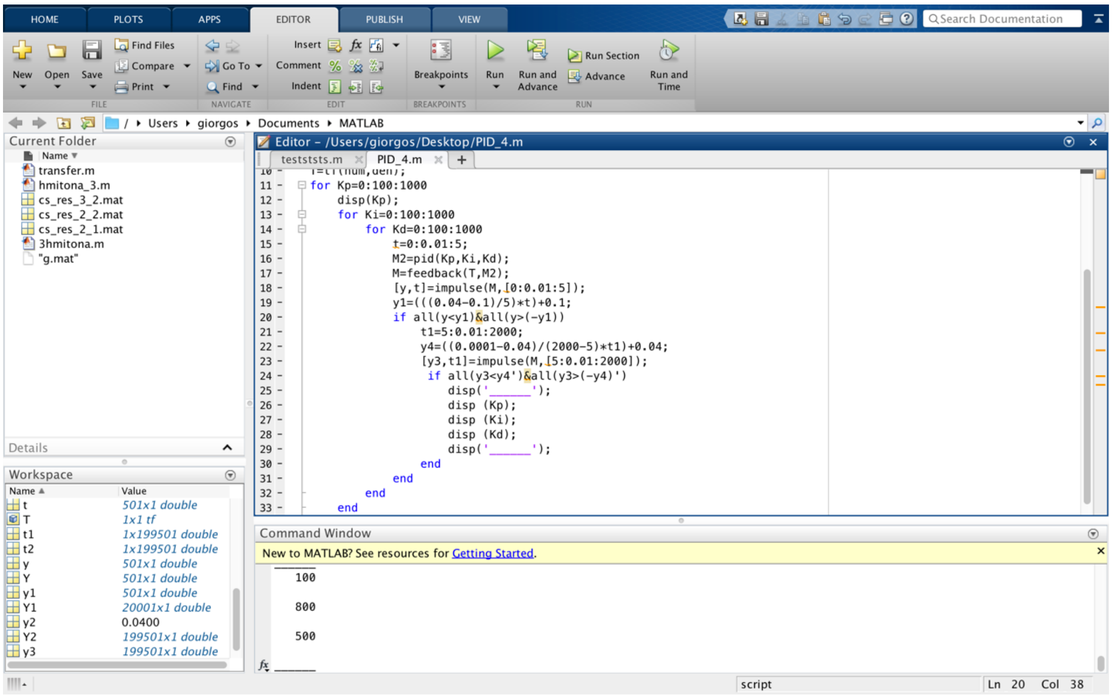
Να συντονιστεί ένας ελεγκτής τριών όρων $G_{PID}(s) = K_p + K_d s + \frac{K_i}{s}$ [Σχήμα 3]



Σχήμα 3: PID-έλεγχος κινούμενου ανεστραμμένου εκκρεμούς

έτσι ώστε η κρουστική απόκριση του κλειστού συστήματος να περιορίζεται εντός της περιοχής που φαίνεται στο Σχήμα 4, όπου $\Delta\theta^{\max} = -\Delta\theta^{\min} = 0.1$, $\Delta\theta^{+} = -\Delta\theta^{-} = 0.04$, t_1=5. Επίσης ζητείται η μόνιμη τιμή της γωνιακής απόκλισης να συγκλίνει στο μηδέν $\lim_{t\to\infty}\Delta\theta(t) = \Delta\theta_{ss} = 0$. Οι τιμές των παραμέτρων K_p, K_d και K_i θεωρούνται ότι δύνανται να μεταβάλλονται ανάμεσα από 0 έως 1000.

Κώδικας Matlab



Κώδικας Matlab-Σχολιασμός

```
M=1;m=1;l=1;B_l=0.3;B_r=0;g=10;

num=[(1/(M*l)) 0];

den=[1 ((B_l)/M) ((M+m)*(-g))/M*l ((B_l*(-g))/M*l)];

T=tf(num,den); %υλοποίηση συνάρτησης μεταφοράς

for Kp=0:20:1000 %3 LOOP για τις τιμες των K που κυμαίνονται από 0 εως 1000

    disp(Kp);

    for Ki=0:20:1000

        for Kd=0:20:1000

            t=0:0.01:5;   % διάνυσμα χρόνου μεχρι t=0

            M2=pid(Kp,Ki,Kd);

            M=feedback(T,M2); %Υλοποίηση feedback

            [y,t]=impulse(M,[0:0.01:5]);

            y1=(((0.04-0.1)/5)*t)+0.1;
```

```
%πρώτη ευθεία εως t=5
                if all(y<y1)&all(y>(-y1))
%έλεγχος διανυσμάτων(αν το impulse είναι ενδιάμεσα από της δύο ευθείες)
                    t1=5:0.01:2000;
                    y4=((0.0001-0.04)/(2000-5)*t1)+0.04;
%δεύτερη ευθεία για να ελεγξουμε ότι βρίσκεται κατω απο το 0.04 για τ=5 και ότι τείνει στο 0 στο +απειρο. Για να το πετύχουμε αυτό θα βάλουμε μια πολύ μικρή τιμή 0,0001(θεωρητικα 0 δηλαδη) και χρόνο 2000 second.
                    [y3,t1]=impulse(M,[5:0.01:2000]);
                     if all(y3<y4')&all(y3>(-y4)')
%έλεγχος και δεύτερης συνθήκης (αν το impulse είναι ενδιάμεσα από της δύο ευθείες που τείνουν ασυμπτωτικά στο 0).
                        disp('______');
                        disp (Kp);
                        disp (Ki);
                        disp (Kd);
                        disp('______');
                    end
                end
            end
        end
end
```

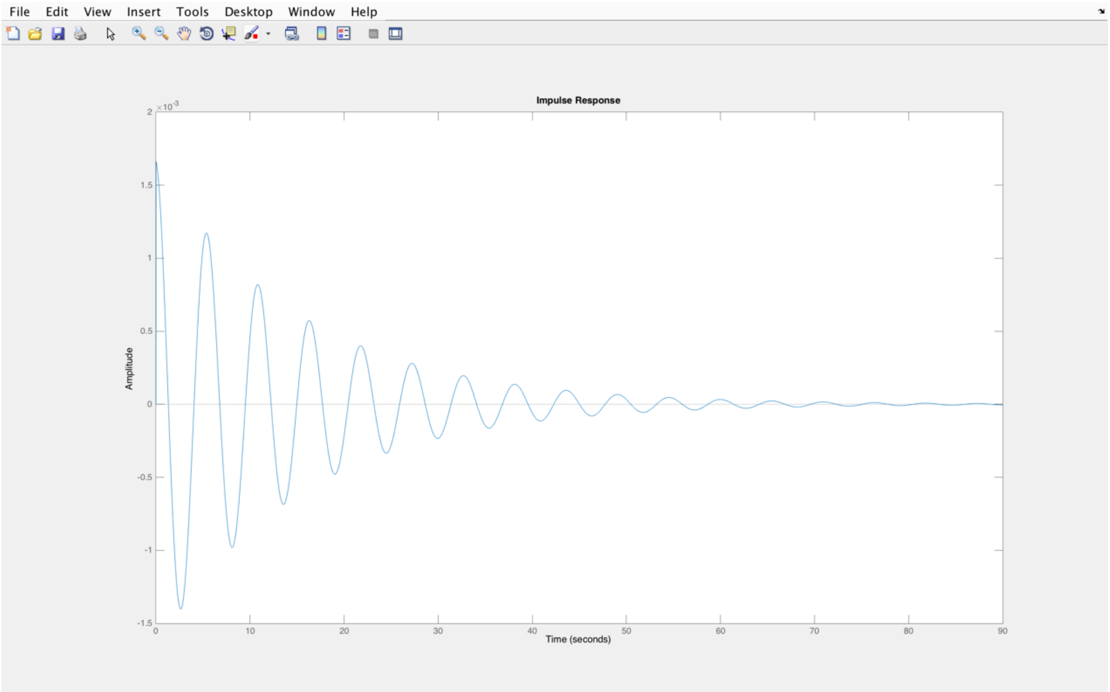
Απ΄ τον παραπάνω κώδικα παίρνουμε αρκετούς συνδυασμούς που ικανοποιούν τις συνθήκες μας για την δημιουργία του PID ελεγκτή.

Για βήμα 100 έχουμε σύνολο 1009 συνδυασμούς που ικανοποιούν τις συνθήκες.

Μερικές από αυτές τις κρουστικές αποκρίσεις είναι οι εξής:

Θα αναλυθουν μερικές εξ' αυτών .

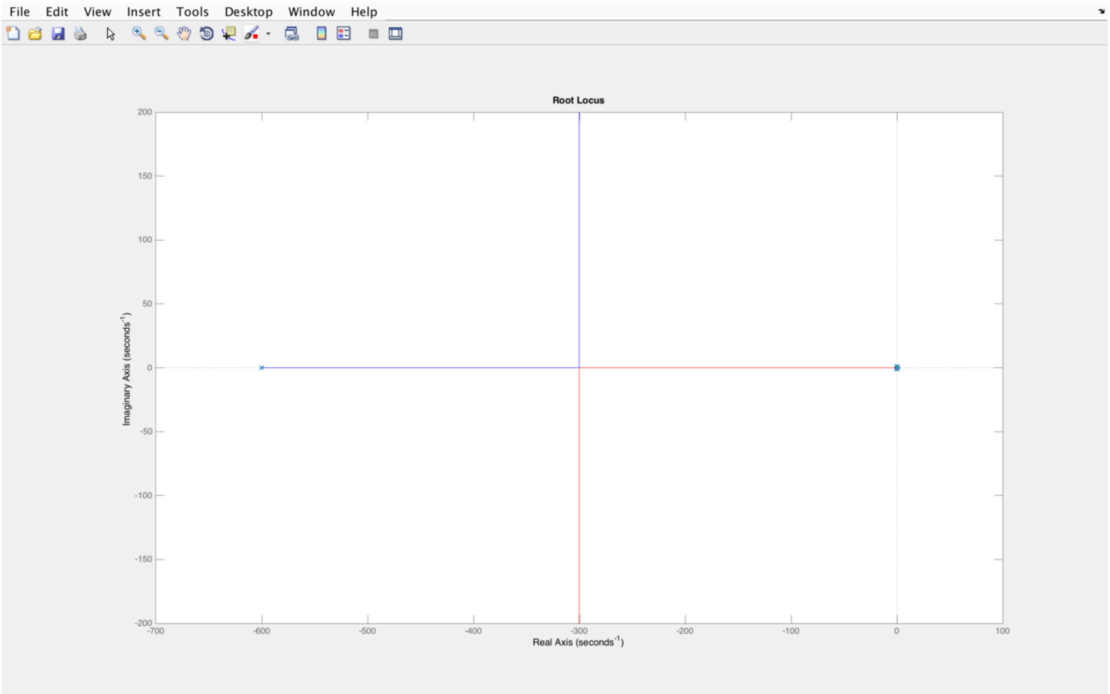
Για Kp=100, Ki=800,Kd=600:



Είναι μέσα στα όρια, αφού για την impulse response έχουμε 0.0017 για t=0,

0.0012 για t=0 και τείνει στο 0 για t=∞

Γεωμετρικός τόπος ριζών rlocus():



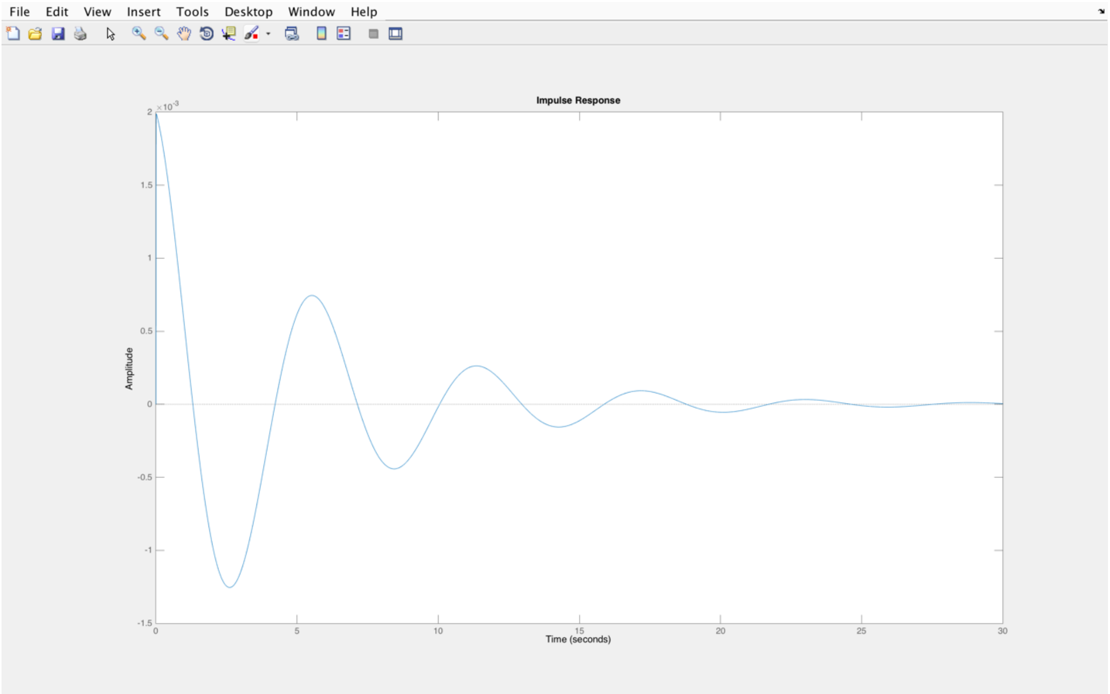
Σύστημα ευσταθές(Όλοι οι πόλοι στο αριστερό ημιεπίπεδο)

Damp():

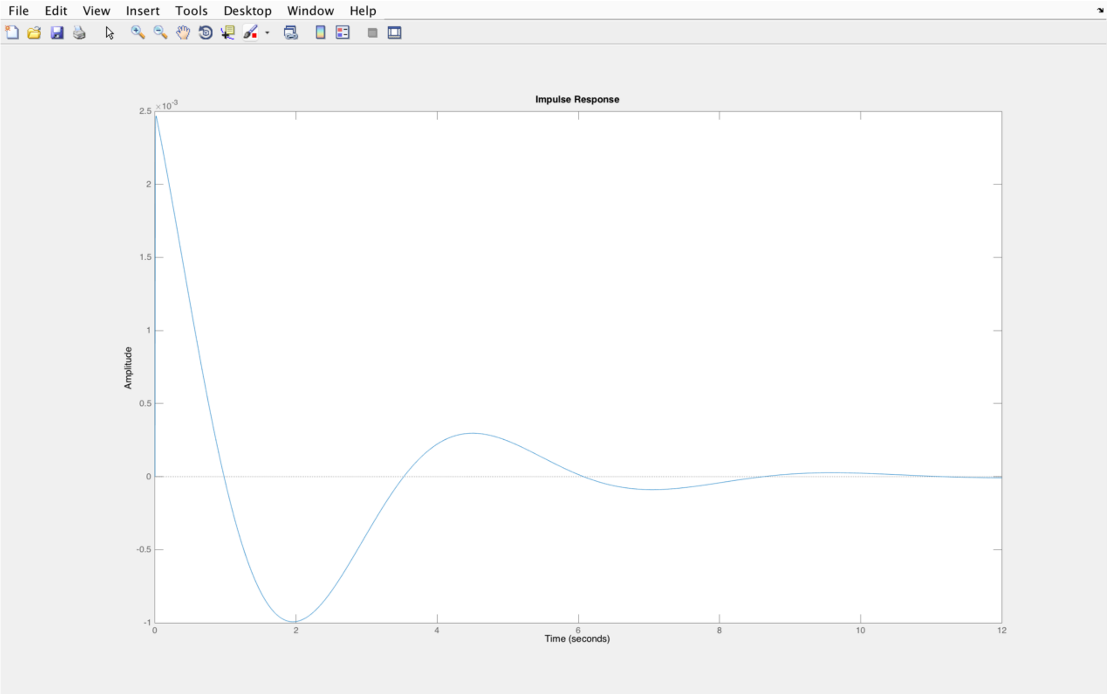
```
        Pole                  Damping       Frequency      Time Constant
                                           (rad/seconds)     (seconds)

  0.00e+00                   -1.00e+00       0.00e+00            Inf
 -6.55e-02 + 1.15e+00i        5.69e-02       1.15e+00         1.53e+01
 -6.55e-02 - 1.15e+00i        5.69e-02       1.15e+00         1.53e+01
 -6.00e+02                    1.00e+00       6.00e+02         1.67e-03
```

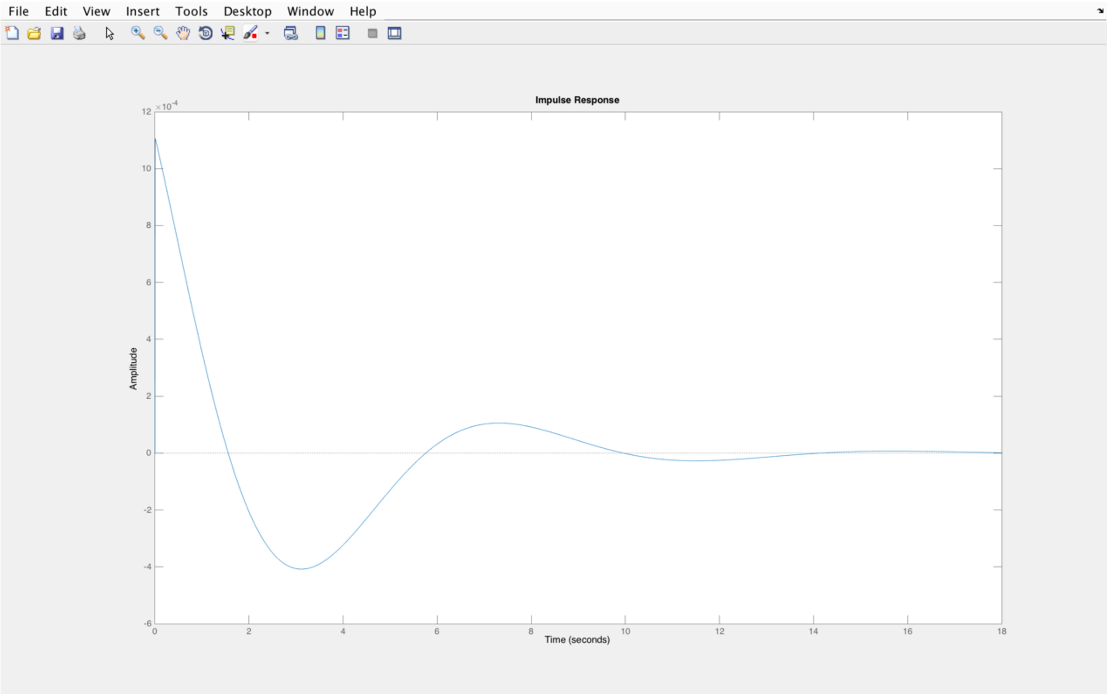
Για Kp=200, Ki=600,Kd=500:



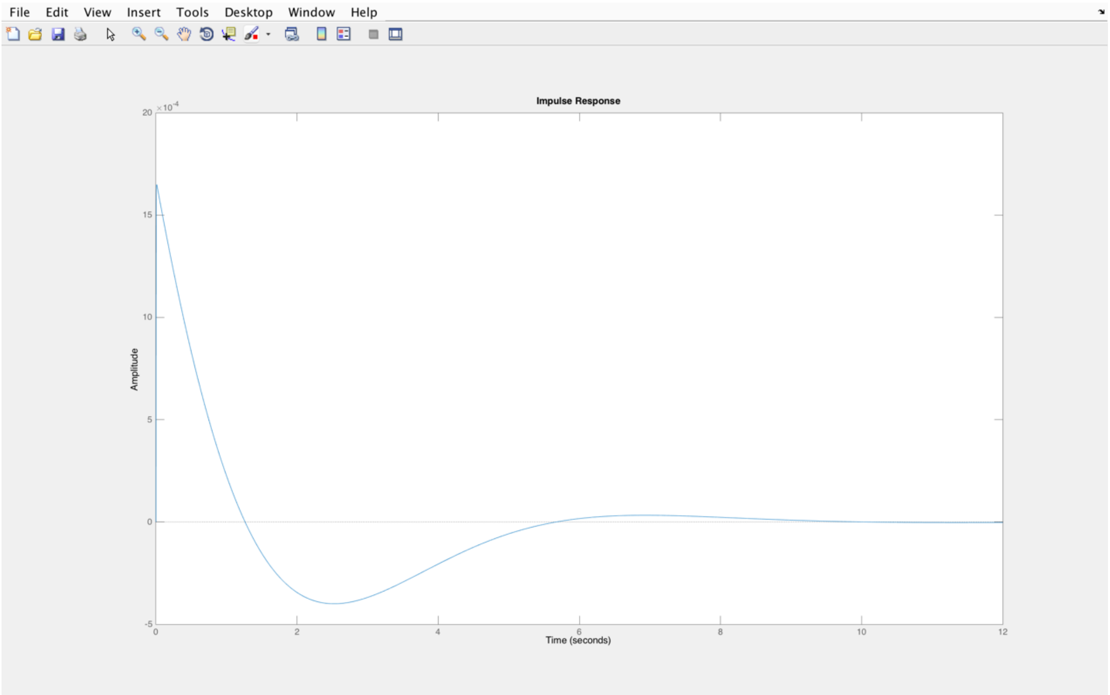
Για Kp=400, Ki=700,Kd=400:



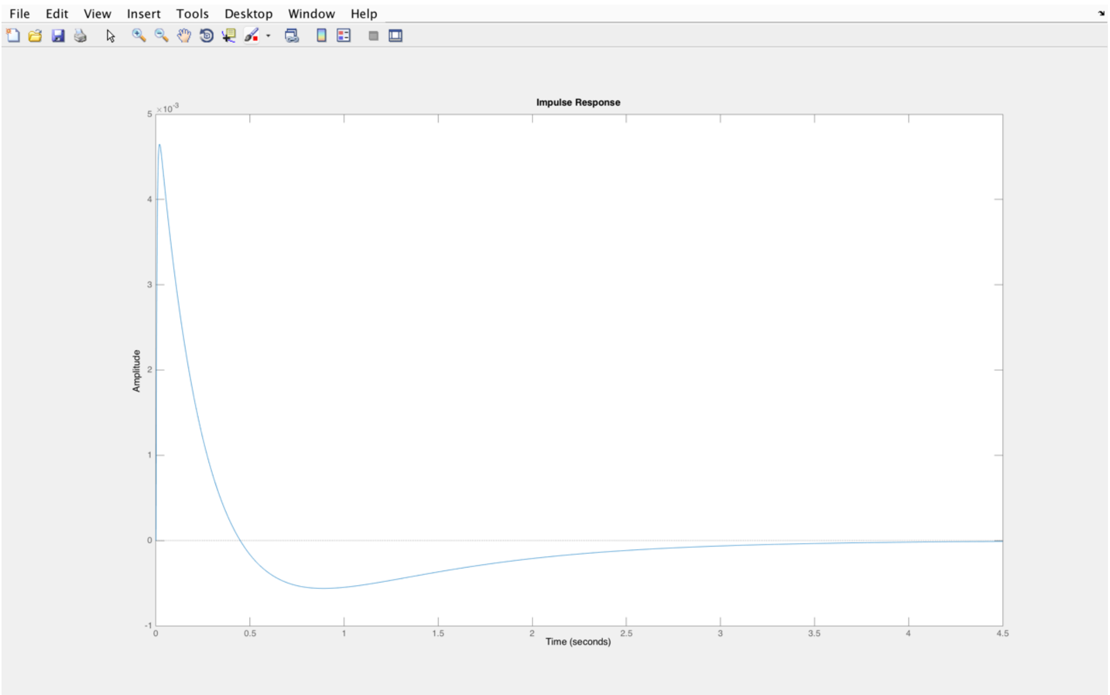
Για Kp=600, Ki=600,Kd=900:



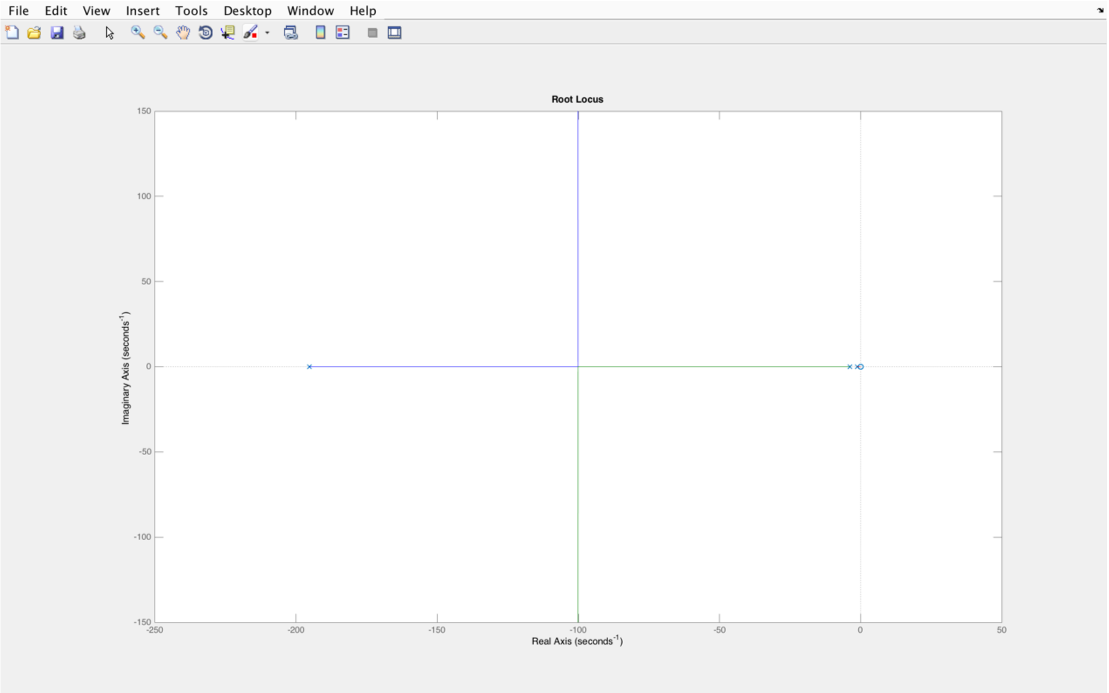
Για Kp=700, Ki=500,Kd=600:



Για Kp=1000, Ki=900,Kd=200:



rlocus():



damp():

Pole	Damping	Frequency (rad/seconds)	Time Constant (seconds)
0.00e+00	-1.00e+00	0.00e+00	Inf
-1.22e+00	1.00e+00	1.22e+00	8.23e-01
-3.78e+00	1.00e+00	3.78e+00	2.65e-01
-1.95e+02	1.00e+00	1.95e+02	5.12e-03

Παρατηρούμε τα εξης:

- Για Kp=0 όλα τα συστήματα είναι ασταθή
- Όσο μεγαλώνει το Kp τόσο μειώνονται οι ταλαντώσεις και μειώνεται το amplitude των.
- Η κρουστική απόκριση για μερικούς εκ΄ των συνδυασμών των Kp, Ki, Kd που ο συγκεκριμένος αλγόριθμος βρήκε ως πιθανούς συνδυασμούς ενδέχεται να πηγαίνουν στην αστάθεια μετά το πέρας της ευθείας που τείνει στο 0.0001. Για να αποτρέψουμε κάτι τέτοιο και να μειώσουμε τον αριθμό αυτών θα ήταν συνετό να χρησιμοποιήσουμε ακόμα μεγαλύτερο χρόνο(10.000 second π.χ.) για την ευθεία η οποία θα έχει π.χ. σαν τελικό σημείο το 0.000001 .

Θα επιλέξουμε τον βέλτιστο(όσο αφορά το amplitude και το πλήθος ταλαντώσεων και απόκριση συστήματος) από αυτούς που εμφανίζονται παραπάνω.

Για Kp=1000, Ki=900,Kd=200:

$$\boldsymbol{GPID(s) = 1000 + 200 * s + \frac{900}{s}}$$