

Άσκηση 10

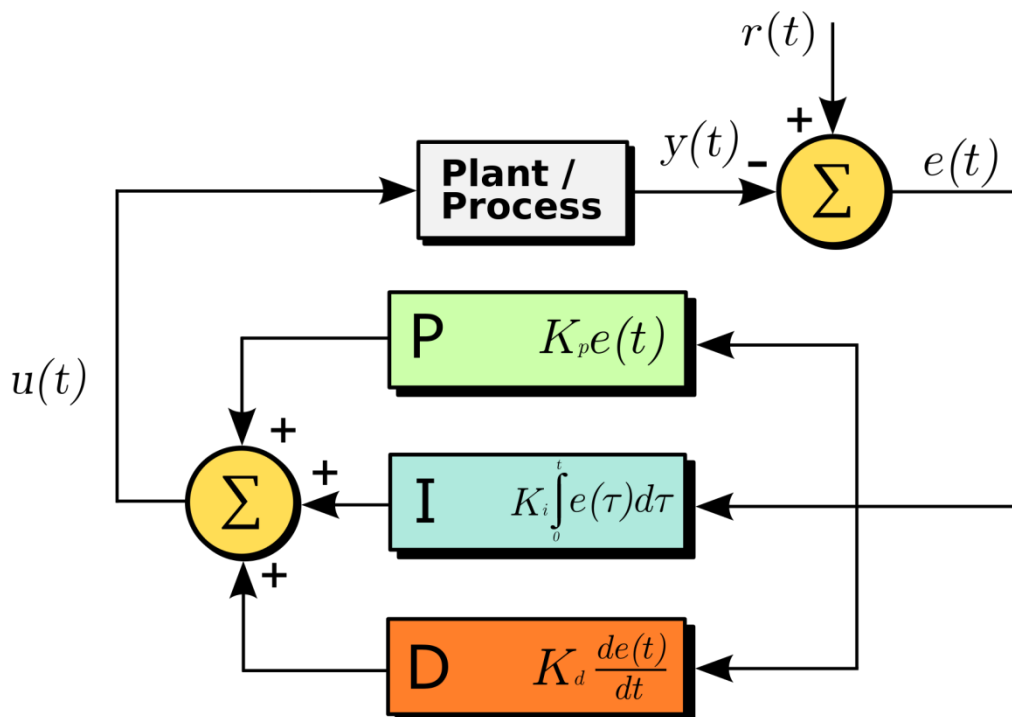
Η θεωρία των PID Controller στο Matlab

Ένας αναλογικός-ολοκληρωτικός-διαφορικός ελεγκτής (ελεγκτής PID) είναι ένας βρόχος ελέγχου μηχανισμού ανάδρασης (controller), που χρησιμοποιείται ευρέως σε βιομηχανικά συστήματα ελέγχου (Programmable Logic Controllers , SCADA συστήματα, μονάδες Remote Terminal κλπ). Ένας ελεγκτής PID υπολογίζει μία τιμή το "σφάλμα" ως την διαφορά ανάμεσα σε μια μετρημένη μεταβλητή της διαδικασίας και ένα επιθυμητό σημείο ρύθμισης. Ο ελεγκτής προσπαθεί να ελαχιστοποιήσει το σφάλμα εξόδου ρυθμίζοντας τις εισόδους ελέγχου της διαδικασίας.

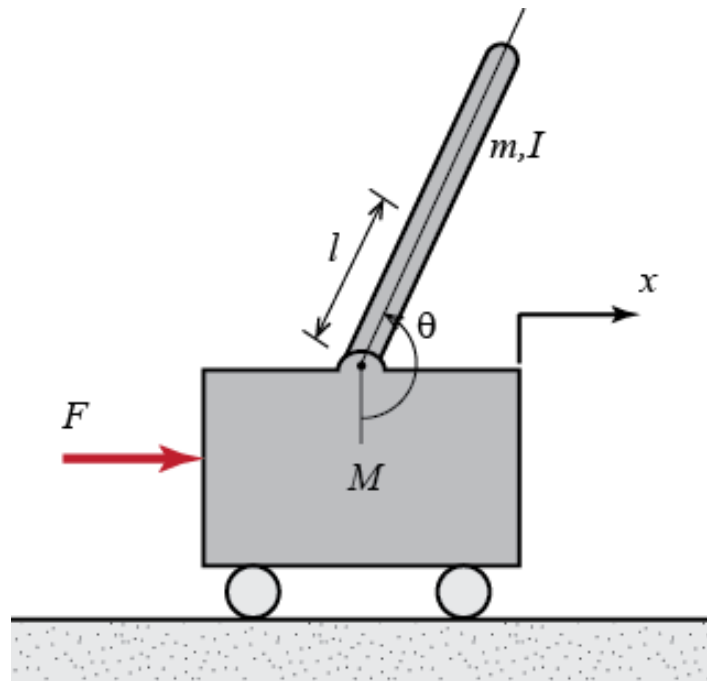
Ο ελεγκτής PID αλγόριθμος περιλαμβάνει τρεις ξεχωριστές σταθερές παραμέτρους, και, κατά συνέπεια, μερικές φορές ονομάζεται τριών όρων ελέγχου P,I,D. Με απλά λόγια, οι τιμές αυτές μπορούν να ερμηνευθούν με όρους χρόνου: P εξαρτάται από την παρούσα λάθος, I στη συσσώρευση των προηγούμενων λαθών, και D είναι μια πρόβλεψη των μελλοντικών σφαλμάτων, με βάση τις σημερινές εξελίξεις. Το σταθμισμένο άθροισμα αυτών των τριών ενεργειών χρησιμοποιείται για να ρυθμίσει τη διαδικασία μέσω ενός στοιχείου ελέγχου, όπως η θέση μίας βαλβίδας ελέγχου , έναν αποσβεστήρα , ή την ισχύ που παρέχεται σε ένα θερμαντικό στοιχείο.

Ορισμένες εφαρμογές μπορεί να απαιτούν τη χρήση μόνο ενός ή δύο δράσεων για την παροχή του κατάλληλου ελέγχου του συστήματος. Αυτό επιτυγχάνεται με τη ρύθμιση των άλλων παραμέτρων στο μηδέν. Ένας ελεγκτής PID θα λέγεται PI, PD.

Ένα μπλοκ διάγραμμα ενός ελεγκτή PID σε ένα βρόχο ανάδρασης



Παράδειγμα Inverted Pendulum

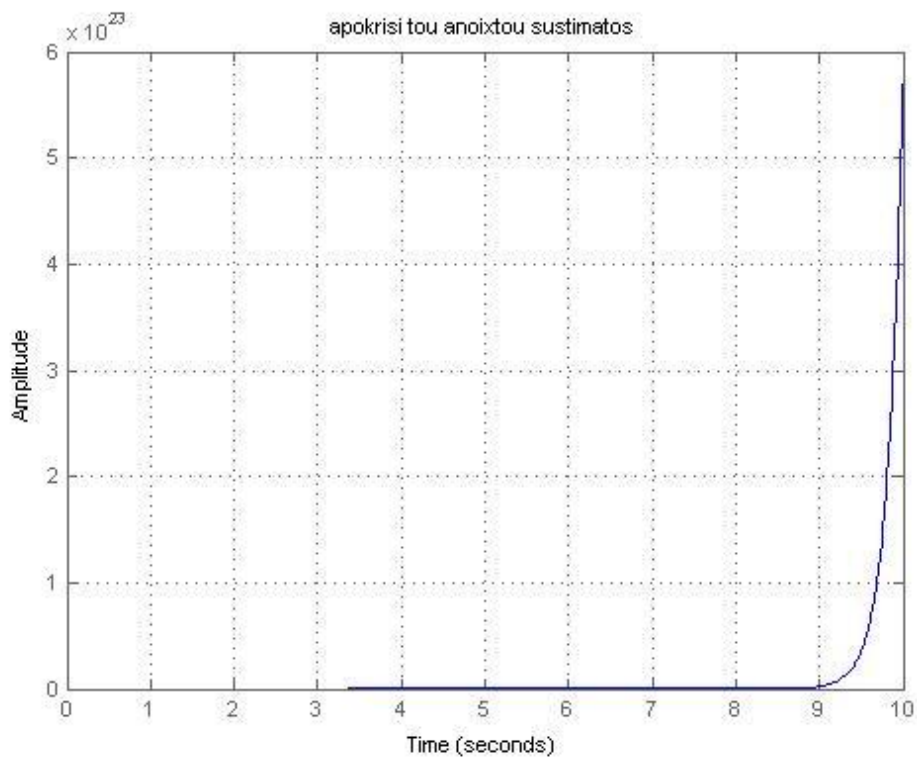


(M)	Μάζα του αμαξιδίου	0.5 kg
(m)	Μάζα του εκκρεμούς	0.2 kg
(b)	Συντελεστής τριβής αμαξιδίου	0.1 N/m/sec
(l)	Μήκος εκκρεμούς από το κέντρο μάζας του	0.3 m
(I)	ροπή αδράνειας του εκκρεμούς	0.006 kg.m ²
(F)	Δύναμη που εφαρμόζεται στο καλάθι	
(theta)	Γωνία εκκρεμούς από το κατακόρυφο άξονα	

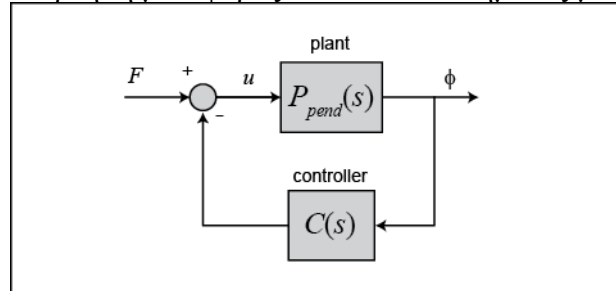
Κώδικας Matlab

Συνάρτηση Μεταφοράς Ανοιχτού Συστήματος

```
M = 0.5;  
m = 0.2;  
b = 0.1;  
I = 0.006;  
g = 9.8;  
l = 0.3;  
q = (M+m)*(I+m*l^2)-(m*l)^2;  
s = tf('s');  
P_pend = (m*I*s/q)/(s^3 + (b*(I + m*l^2))*s^2/q - ((M + m)*m*g*l)*s/q -  
b*m*g*l/q);  
t=0:0.01:10;  
impulse(P_pend,t)  
grid  
title('apokrisi tou anoixtou sustimatos');
```



Συνάρτηση μεταφοράς κλειστού συστήματος με PID



M = 0.5;

m = 0.2;

b = 0.1;

I = 0.006;

g = 9.8;

l = 0.3;

q = (M+m)*(I+m*l^2)-(m*l)^2;

s = tf('s');

P_pend = (m*I*s/q)/(s^3 + (b*(I + m*l^2))*s^2/q - ((M + m)*m*g*l)*s/q - b*m*g*l/q);

Kp = 100;

Ki = 1;

Kd = 1;

C = pid(Kp,Ki,Kd);

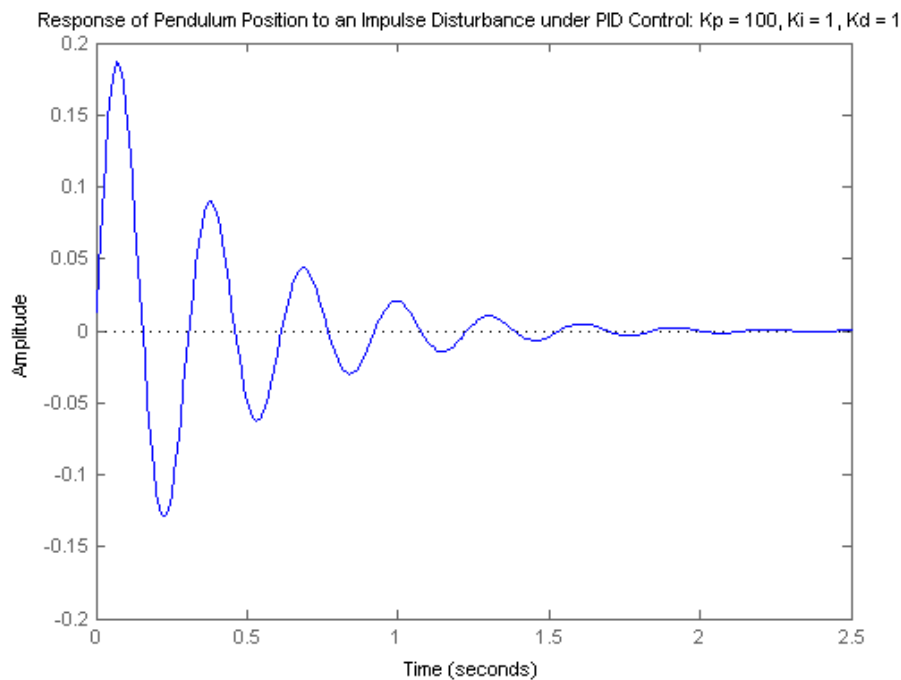
T = feedback(P_pend,C);

t=0:0.01:10;

impulse(T,t)

axis([0, 2.5, -0.2, 0.2]);

title('Response of Pendulum Position to an Impulse Disturbance under PID Control:
Kp = 100, Ki = 1, Kd = 1');



```

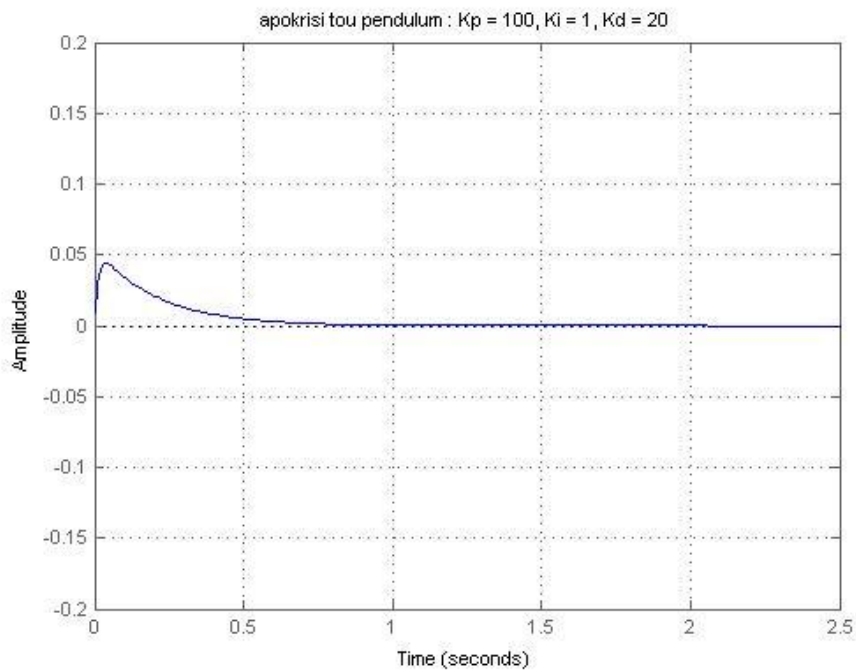
M = 0.5;
m = 0.2;
b = 0.1;
I = 0.006;
g = 9.8;
l = 0.3;
q = (M+m)*(I+m*l^2)-(m*l)^2;
s = tf('s');
P_pend = (m*l*s/q)/(s^3 + (b*(I + m*l^2))*s^2/q - ((M + m)*m*g*l)*s/q -
b*m*g*l/q);

```

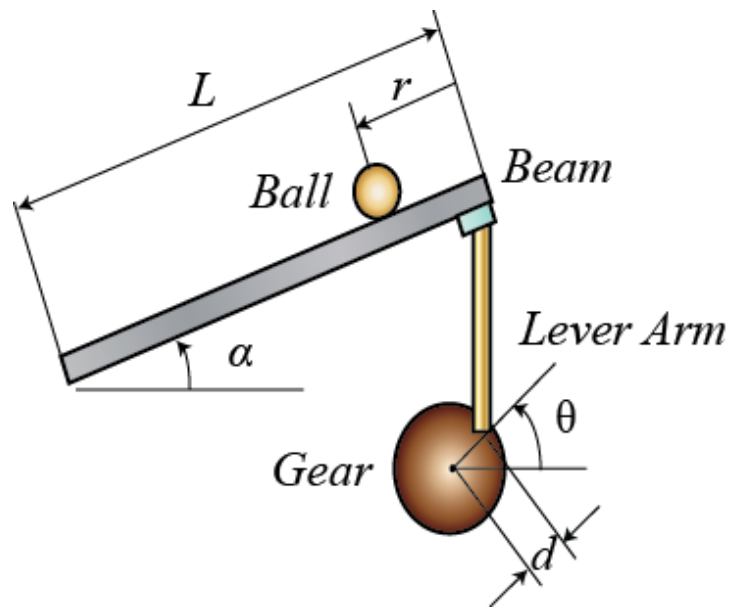
```

Kp = 100;
Ki = 1;
Kd = 20;
C = pid(Kp,Ki,Kd);
T = feedback(P_pend,C);
t=0:0.01:10;
impulse(T,t)
axis([0, 2.5, -0.2, 0.2]);
grid
title('apokrisi tou pendulum: Kp = 100, Ki = 1, Kd = 20');

```



Παράδειγμα Ball & Beam

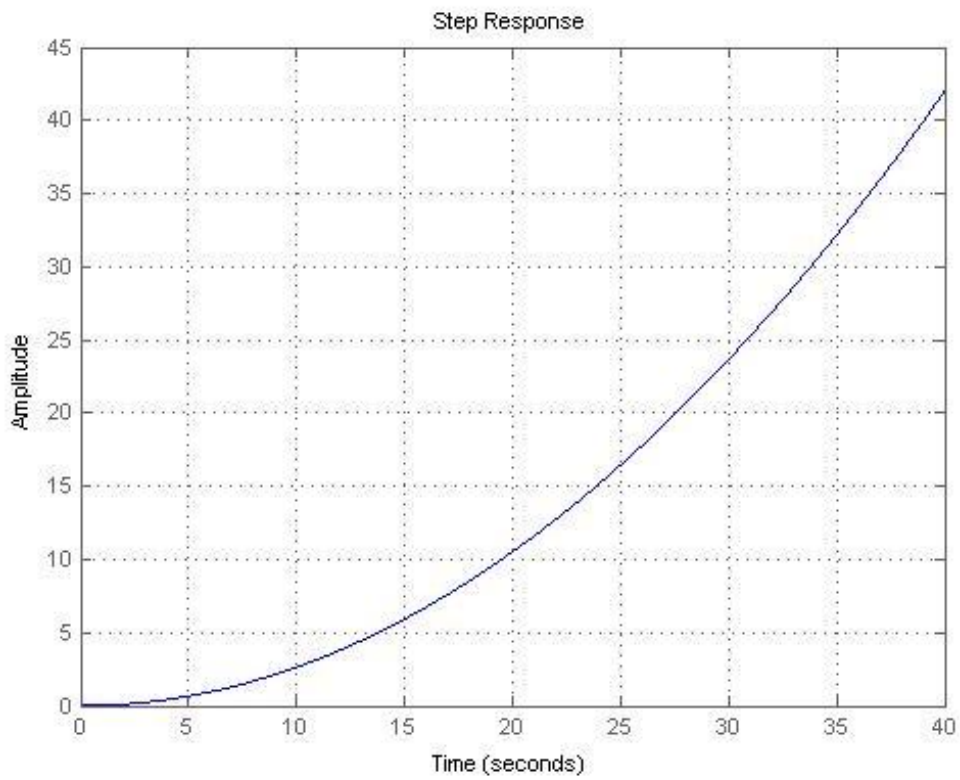


(m)	Μάζα της μπάλας	0.11 kg
(R)	Η ακτίνα της σφαίρας	0.015 m
(d)	Ύψος βραχίονα offset	0.03 m
(g)	Επιτάχυνση της βαρύτητας	9.8 m/s ²
(L)	Μήκος της δοκού	1.0 m
(J)	Ροπή αδράνειας της σφαίρας	9.99e-6 kg.m ²
(r)	Θέση της μπάλας	
(alpha)	Γωνία της δοκού	
(theta)	Γωνία γραναζιού servo	

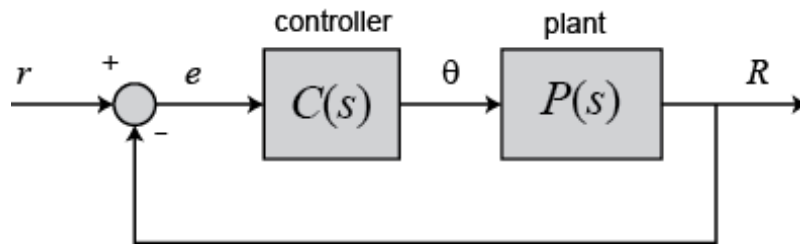
Κώδικας Matlab

Συνάρτηση Μεταφοράς Ανοιχτού Συστήματος

```
m = 0.111;  
R = 0.015;  
g = -9.8;  
L = 1.0;  
d = 0.03;  
J = 9.99e-6;  
s = tf('s');  
P_ball = -m*g*d/L/(J/R^2+m)/s^2;  
step(0.25*P_ball)  
grid
```

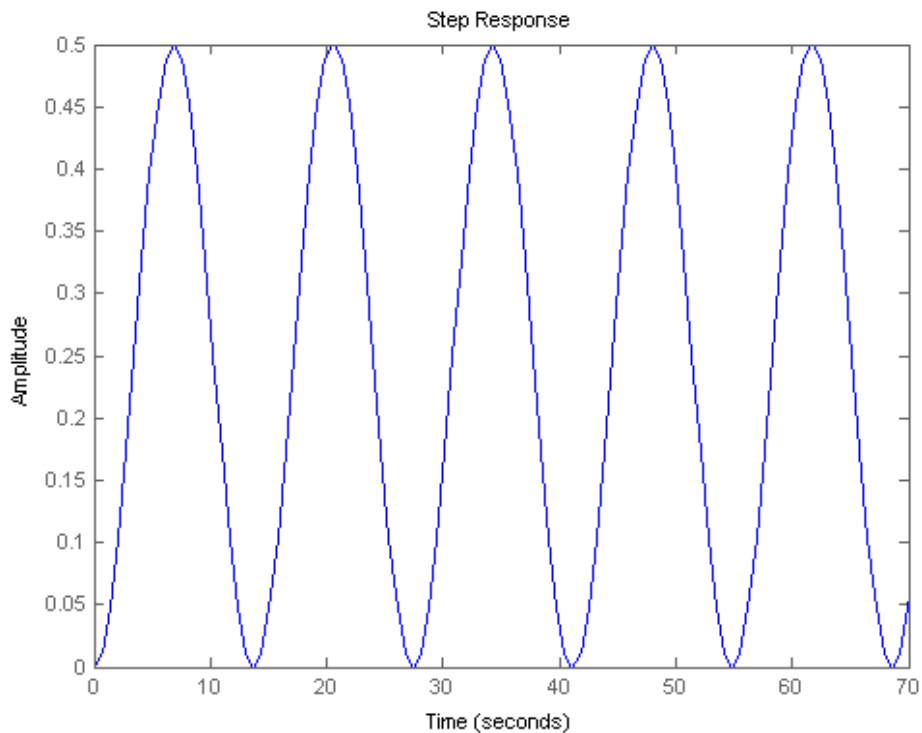


Συνάρτηση μεταφοράς κλειστού συστήματος με PID

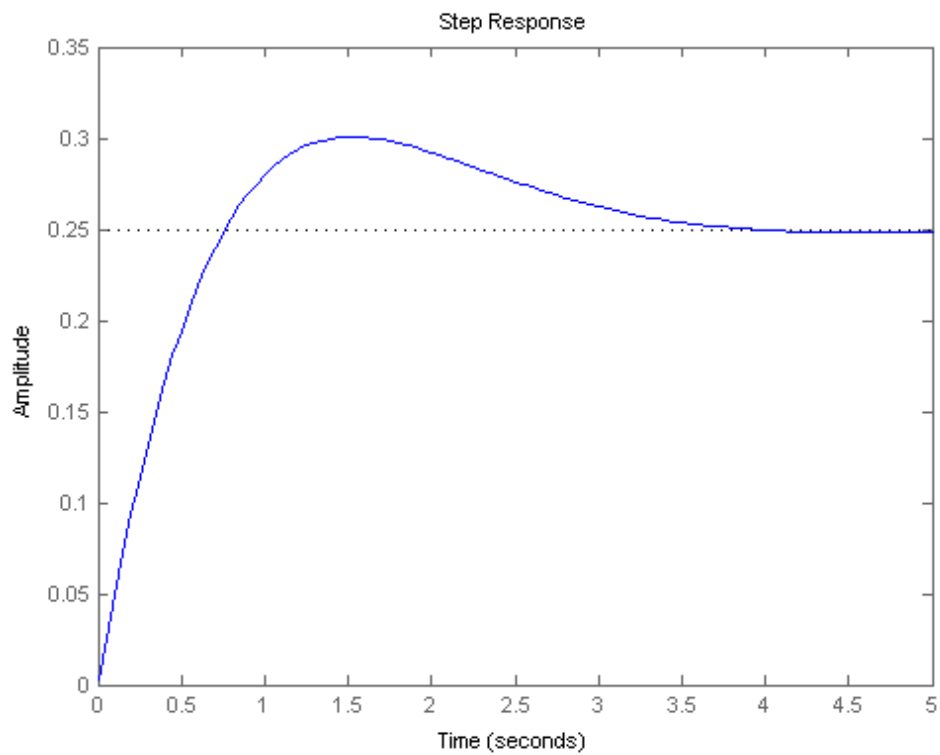


```
m = 0.111;
R = 0.015;
g = -9.8;
L = 1.0;
d = 0.03;
J = 9.99e-6;
s = tf('s');
P_ball = -m*g*d/L / (J/R^2+m) / s^2;
```

```
Kp = 1;
C = pid(Kp);
sys_cl=feedback(C*P_ball,1);
step(0.25*sys_cl)
axis([0 70 0 0.5])
```

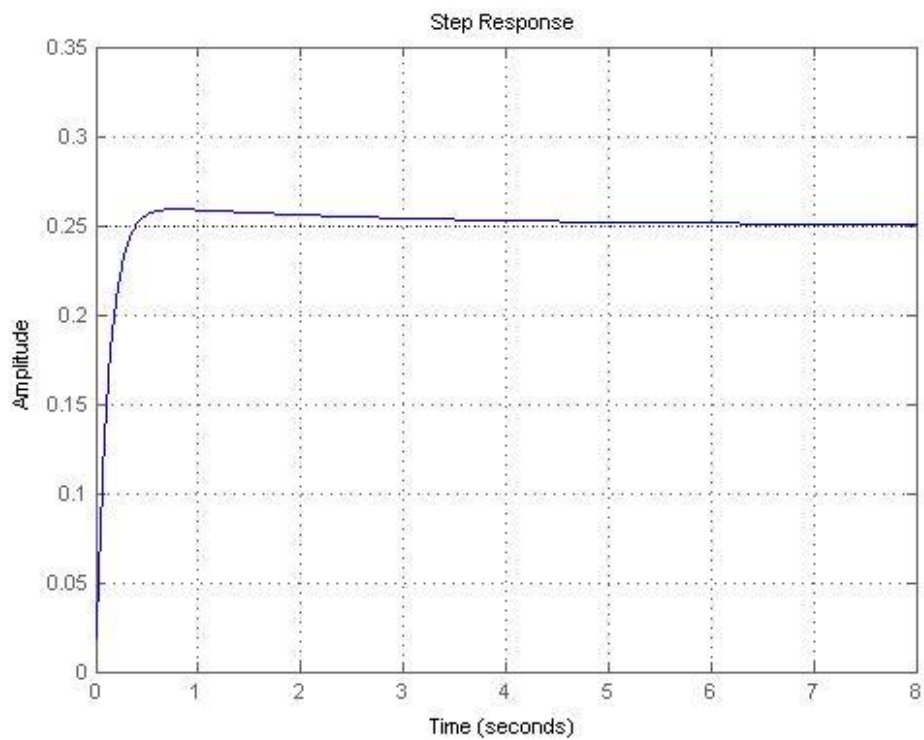



```
m = 0.111;  
R = 0.015;  
g = -9.8;  
L = 1.0;  
d = 0.03;  
J = 9.99e-6;  
s = tf('s');  
P_ball = -m*g*d/L/(J/R^2+m)/s^2;  
  
Kp = 10;  
Kd = 10;  
C = pid(Kp,0,Kd);  
  
sys_cl=feedback(C*P_ball,1);  
  
t=0:0.01:5;  
step(0.25*sys_cl)
```



```
m = 0.111;  
R = 0.015;  
g = -9.8;  
L = 1.0;  
d = 0.03;  
J = 9.99e-6;  
s = tf('s');  
P_ball = -m*g*d/L/(J/R^2+m)/s^2;
```

```
Kp = 15;  
Ki = 0;  
Kd = 40;  
C = pid(Kp,Ki,Kd);  
sys_cl=feedback(C*P_ball,1);  
step(0.25*sys_cl)  
grid
```



Πρακτικό Άσκησης #10

Άσκηση #1

1) Αλλάξτε τις τιμές των ακόλουθων παραμέτρων για το inverted pendulum:

A) $K_p = 1, 10, 1000$

B) $K_i = 10, 100, 1000$

Γ) $K_d = 1, 100, 1000$

Σχολιάστε πως μεταβάλλεται η χρονική απόκριση του συστήματος ανάλογα με τις τιμές που δίνετε στις παραμέτρους.

Άσκηση #2

1) Αλλάξτε τις τιμές των ακόλουθων παραμέτρων για το ball & beam:

A) $K_p = 1.5, 150, 1500$

B) $K_i = 10, 100, 1000$

Γ) $K_d = 4, 400, 4000$

Σχολιάστε πως μεταβάλλεται η χρονική απόκριση του συστήματος ανάλογα με τις τιμές που δίνετε στις παραμέτρους.