

*ΕΡΓΑΣΤΗΡΙΟ ΣΥΣΤΗΜΑΤΩΝ
ΑΥΤΟΜΑΤΟΥ ΕΛΕΓΧΟΥ*

ΣΥΣΤΗΜΑΤΑ ΑΥΤΟΜΑΤΟΥ ΕΛΕΓΧΟΥ (II)

MODERN CONTROL THEORY

(ΠΡΟΣΟΜΟΙΩΣΗ ΔΥΝΑΜΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ)

Διδάσκων : **Ι. Μπούταλης**
Αναπληρωτής Καθηγητής

ΠΡΟΣΟΜΟΙΩΣΗ ΔΥΝΑΜΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

- Θα αναφερθούμε στην επίλυση δυναμικών εξισώσεων με τη μέθοδο Runge-Kutta
- Θα αναφερθούμε σύντομα στη χρήση του προγραμματιστικού περιβάλλοντος MATLAB για την προσομοίωση απλών και σύνθετων συστημάτων

ΠΡΟΣΟΜΟΙΩΣΗ ΔΥΝΑΜΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

(ΕΠΙΛΥΣΗ ΔΥΝΑΜΙΚΩΝ ΕΞΙΣΩΣΕΩΝ ΜΕ ΤΗ ΒΟΗΘΕΙΑ ΥΠΟΛΟΓΙΣΤΗ)

ΜΕΘΟΔΟΣ RUNGE-KUTTA **C. Runge (1895) – W. Kutta (1901)**
(Επίλυση διαφορικών εξισώσεων)

Έστω : $\frac{dx}{dt} = f(x, t)$ Και ότι για $t=t_i$ είναι γνωστό ότι $x=x_i$.

h = Μικρή
χρονική αύξηση

Θέλουμε να βρούμε την τιμή του $x(t)$ για $t=t_{i+1}$, όπου $t_{i+1}=t_i+h$ $h = t_{i+1} - t_i$

Στη μέθοδο Runge-Kutta τέταρτης τάξης υπολογίζονται πρώτα οι ποσότητες

$$K_1 = h \bullet f(t_i, x_i) \qquad K_2 = h \bullet f(t_i + 0.5h, x_i + 0.5K_1)$$

$$K_3 = h \bullet f(t_i + 0.5h, x_i + 0.5K_2) \qquad K_4 = h \bullet f(t_i + h, x_i + K_3)$$

Αν ορίσουμε $\Delta x_i = \frac{1}{6}(K_1 + 2K_2 + 2K_3 + K_4)$

τότε $x_{i+1} = x_i + \Delta x_i = x_i + \frac{1}{6}(K_1 + 2K_2 + 2K_3 + K_4)$

ΠΡΟΣΟΜΟΙΩΣΗ ΔΥΝΑΜΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

(Ένα απλό παράδειγμα)

Σταγόνα μάζας m σε ελεύθερη πτώση

$$m \frac{dv}{dt} = mg - bv$$
$$\frac{dv}{dt} = g - cv, \quad c = b/m$$

V η ταχύτητα της σταγόνας, g η επιτ. Βαρύτητας, b η αντίσταση του αέρα

Υποθέτουμε τις τιμές $g = 9,81 \text{ m/s}^2$, $c = 0,1 \text{ s}^{-1}$, $v(0) = 0$

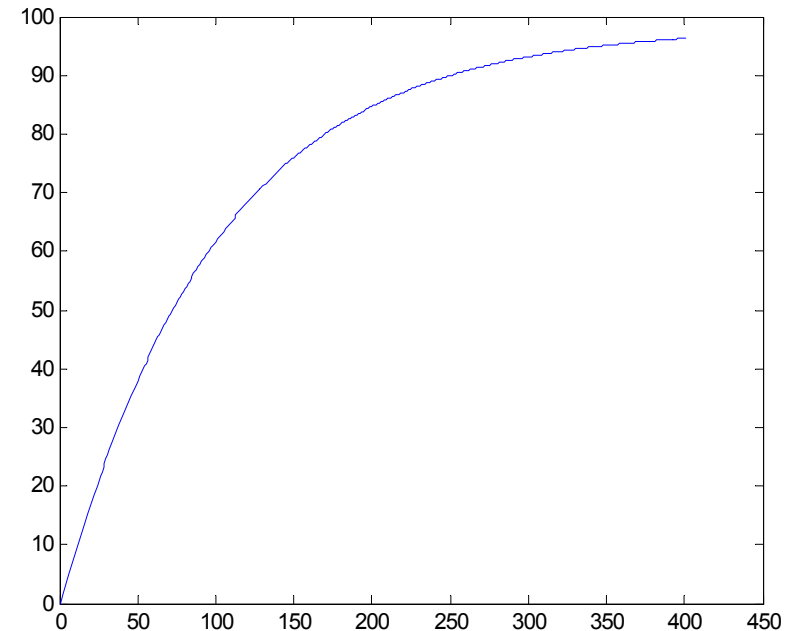
Θέτοντας $x=v$ η δυναμική Εξίσωση γράφεται

$$\dot{x} = 9.81 - 0.1x,$$
$$x(0) = 0$$

```
%Main program
%Drop in free fall
h = 0.1; tinit = 0; tfinal = 10; x=0;
xplot(1) = x; index = 1; t=tinit+h;
for i=0.1:0.1:40
```

```
function f=fdrop(t,x);
f = 9.81 - 0.1*x;
```

```
    xtemp = xplot(index);
    k1 = h*fdrop(t,xtemp);
    k2 = h*fdrop(t+0.5*h,xtemp+0.5*k1);
    k3 = h*fdrop(t+0.5*h,xtemp+0.5*k2);
    k4 = h*fdrop(t+h,xtemp+k3);
    index = index+1;
    xplot(index) = xplot(index-1) + (k1+2*k2+2*k3+k4)/6;
    t = t+h;
end;
plot(xplot)
```



ΠΡΟΣΟΜΟΙΩΣΗ ΔΥΝΑΜΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

(ΕΠΙΛΥΣΗ ΔΥΝΑΜΙΚΩΝ ΕΞΙΣΩΣΕΩΝ ΜΕ ΤΗ ΒΟΗΘΕΙΑ ΥΠΟΛΟΓΙΣΤΗ)

(ΠΟΛΥΜΕΤΑΒΛΗΤΑ ΣΥΣΤΗΜΑΤΑ)

Έστω : $\dot{x}_1 = f(t, x_1, x_2)$ Τότε σύμφωνα με τη μέθοδο $x_1(t_i + h) = x_1(t_i) + K$
 $\dot{x}_2 = g(t, x_1, x_2)$ Runge-Kutta τέταρτης τάξης $x_{1i+1} = x_{1i} + K$

$$x_2(t_i + h) = x_2(t_i) + L \quad \eta \quad x_{2i+1} = x_{2i} + L$$

όπου $K = \frac{1}{6}(K_1 + 2K_2 + 2K_3 + K_4)$ $L = \frac{1}{6}(L_1 + 2L_2 + 2L_3 + L_4)$

$$K_1 = h \bullet f(t_i, x_{1i}, x_{2i})$$

$$K_3 = h \bullet f(t_i + 0.5h, x_{1i} + 0.5K_2, x_{2i} + 0.5L_2)$$

$$L_1 = h \bullet g(t_i, x_{1i}, x_{2i})$$

$$L_3 = h \bullet g(t_i + 0.5h, x_{1i} + 0.5K_2, x_{2i} + 0.5L_2)$$

$$K_2 = h \bullet f(t_i + 0.5h, x_{1i} + 0.5K_1, x_{2i} + 0.5L_1)$$

$$K_4 = h \bullet f(t_i + h, x_{1i} + K_3, x_{2i} + L_3)$$

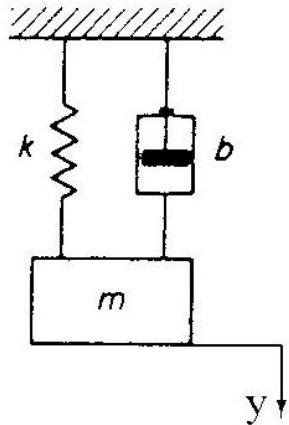
$$L_2 = h \bullet g(t_i + 0.5h, x_{1i} + 0.5K_1, x_{2i} + 0.5L_1)$$

$$L_4 = h \bullet g(t_i + h, x_{1i} + K_3, x_{2i} + L_3)$$

Τα παραπάνω επεκτείνονται σε n εξισώσεις και μπορούν να γραφούν σε πίνακοδιανυσματική συμπαγή μορφή

ΠΡΟΣΟΜΟΙΩΣΗ ΔΥΝΑΜΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

(ΠΑΡΑΔΕΙΓΜΑ ΠΟΛΥΜΕΤΑΒΛΗΤΟΥ ΣΥΣΤΗΜΑΤΟΣ)



Θεωρούμε το σύστημα ελατηρίου, μάζας, αποσβεστήρα ταλαντώσεων που φαίνεται στο σχήμα. Υποθέτουμε ότι $m = 1$ slug, $b = 4$ και $k = 40$. Θα δώσουμε ένα πρόγραμμα σε BASIC για την εξαγωγή της λύσης της εξίσωσης του συστήματος όταν οι αρχικές συνθήκες είναι $y(0) = 1$ και $y'(0) = 0$. Θα δείξουμε τη γραφική παράσταση του $y(t)$ ως προς το χρόνο t για $0 \leq t \leq 5$

Εφαρμόζοντας το δεύτερο νόμο του Newton στο σύστημα προκύπτει

$$\ddot{y} + 4\dot{y} + 40y = 0$$
$$y(0) = 1, \dot{y}(0) = 0$$

Γράφουμε τις εξισώσεις χώρου κατάστασης του συστήματος κάνοντας τις αντικαταστάσεις

$$x_1 = y, \quad x_2 = \dot{y}$$

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -40 & -4 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$
$$y = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

... ΣΥΝΕΧΕΙΑ ΠΑΡΑΔΕΙΓΜΑΤΟΣ

Οι συναρτήσεις \dot{x}_1, \dot{x}_2
 Δηλώνονται ως FNX1D(T,X1,X2) και
 FNX2D(T,X1,X2) αντίστοιχα

```

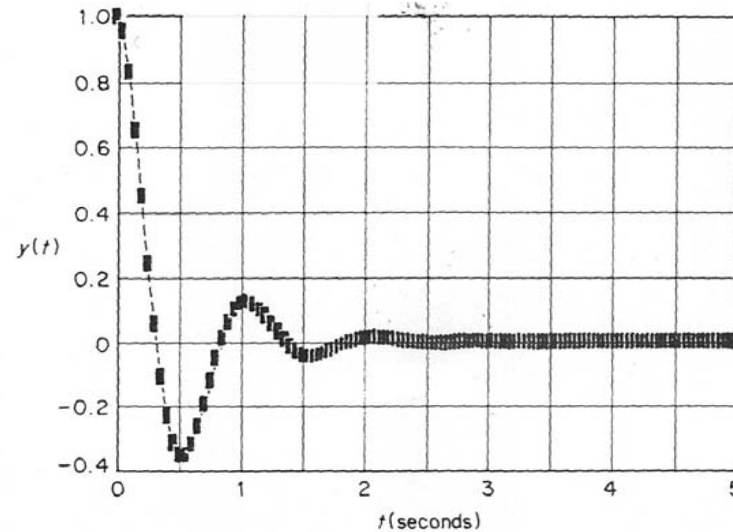
10 OPEN "O", #1, "ANS.BAS"
20 DEF NFX1D(T,X1,X2)=X2
30 DEF NFX2D(T,X1,X2)=- 40*X1- 4*X2
40 H=.05
50 T=0 : X1=1 : X2=0
60 TT=5
70 PRINT "    TIME      X1      X2"
80 PRINT "-----"
90 IF T > TT THEN GOTO 1000
100 PRINT #1, USING "#####.#####"; X1
110 PRINT USING "#####.#####"; T,X1,X2
120 K1= H*NFX1D(T,X1,X2)
130 L1= H*NFX2D(T,X1,X2)
140 K2= H*NFX1D(T+.5*H,X1+.5*K1,X2+.5*L1)
150 L2= H*NFX2D(T+.5*H,X1+.5*K1,X2+.5*L1)
160 K3= H*NFX1D(T+.5*H,X1+.5*K2,X2+.5*L2)

```

```

170 L3= H*NFX2D(T+.5*H,X1+.5*K2,X2+.5*L2)
180 K4= H*NFX1D(T+H,X1+K3,X2+L3)
190 L4= H*NFX2D(T+H,X1+K3,X2+L3)
200 X1=X1+(K1+2*K2+2*K3+K4)/6
210 X2=X2+(L1+2*L2+2*L3+L4)/6
220 T=T+H
230 GOTO 90
1000 CLOSE #1
1010 END

```



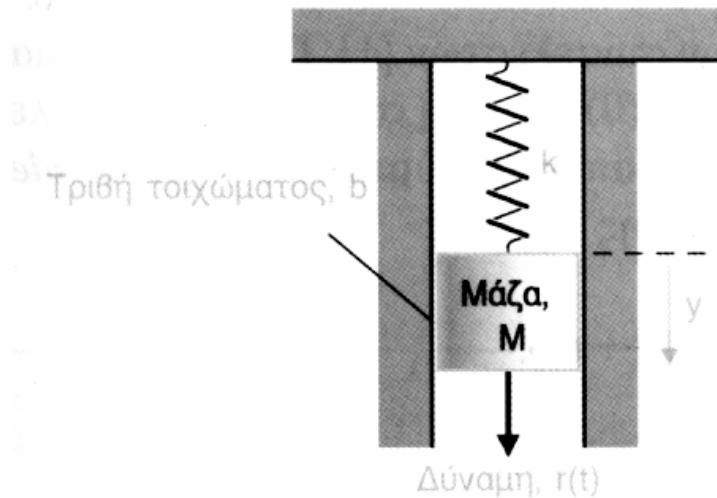
ΧΡΗΣΗ ΤΟΥ ΠΕΡΙΒΑΛΛΟΝΤΟΣ MATLAB

Η εφαρμογή πολλών εργαλείων σχεδίασης και ανάλυσης σύγχρονων αλλά και κλασικών συστημάτων αυτομάτου ελέγχου βασίζεται στην ύπαρξη κατάλληλου μαθηματικού μοντέλου. Το περιβάλλον λογισμικού MATLAB χρησιμοποιείται για την ανάλυση **συστημάτων των οποίων η περιγραφή δίνεται υπό μορφή συνάρτησης μεταφοράς**.

Θα δείξουμε πως μπορούμε να χρησιμοποιήσουμε το MATLAB για την ανάλυση ενός μηχανικού συστήματος με χρήση του κλασικού μαθηματικού μοντέλου του συστήματος αποσβεννύμενων ταλαντώσεων μάζας-ελατηρίου. Χρησιμοποιώντας ένα αρχείο εντολών (πρόγραμμα υπό μορφή λίστας εντολών του περιβάλλοντος, ή απλούστερα "*script*") θα αναπτύξουμε μια κατάλληλη μέθοδο ανάλυσης με σκοπό την μελέτη της επίδρασης της φυσικής συχνότητας και της απόσβεσης στην ελεύθερη απόκριση του συστήματος κατά την μετατόπιση της μάζας. Στην ανάλυση αυτή θα εκμεταλλευτούμε το γεγονός ότι **έχουμε στην διάθεση μας μια αναλυτική λύση** η οποία περιγράφει την ελεύθερη απόκριση της μετατόπισης της μάζας στο πεδίο του χρόνου.

Στη συνέχεια θα ασχοληθούμε με τις **συναρτήσεις μεταφοράς και τα λειτουργικά διαγράμματα**. Συγκεκριμένα μας ενδιαφέρει να διαπιστώσουμε με ποιον τρόπο μας χρησιμεύει το MATLAB στον χειρισμό των πολυωνύμων, στους υπολογισμούς των πόλων και των μηδενικών των συναρτήσεων μεταφοράς, στον υπολογισμό των συναρτήσεων μεταφοράς κλειστού βρόχου, στην διαδικασία απλοποίησης των λειτουργικών διαγραμμάτων και στον υπολογισμό της απόκρισης ενός συστήματος με είσοδο την **μοναδιαία βηματική συνάρτηση (βηματική απόκριση)**.

ΣΥΣΤΗΜΑ ΑΠΟΣΒΕΝΝΥΜΕΝΩΝ ΤΑΛΑΝΤΩΣΕΩΝ



Δίνεται η περιγραφή ενός μηχανικού συστήματος απόσβεσης των ταλαντώσεων, σύστημα ελατηρίου-μάζας. Η κίνηση της μάζας, που συμβολίζεται $y(t)$, περιγράφεται από την διαφορική εξίσωση

$$M\ddot{y}(t) + b\dot{y}(t) + ky(t) = r(t)$$

Η **ελεύθερη** δυναμική απόκριση $y(f)$ του συστήματος, είναι

$$y(t) = \frac{y(0)}{\sqrt{1-\zeta^2}} e^{-\zeta\omega_n t} \sin(\omega_n \sqrt{1-\zeta^2} t + \theta)$$

όπου $\theta = \cos^{-1}\zeta$. Η αρχική μετατόπιση είναι $y(0)$. Η **μεταβατική απόκριση** του συστήματος είναι **υποαποσβεννύμενη** όταν ισχύει $\zeta < 1$, **υπεραποσβεννύμενη** όταν ισχύει $\zeta > 1$, ενώ τέλος, για $\zeta = 1$ έχουμε την λεγόμενη **κρίσιμη απόσβεση**. Μπορούμε να χρησιμοποιήσουμε το MATLAB με σκοπό να σχηματίσουμε μια σαφή εικόνα της ελεύθερης απόκρισης του παραπάνω συστήματος θεωρώντας την αρχική συνθήκη $y(0)$. Έστω η περίπτωση της υποαποσβεννύμενης συμπεριφοράς όπου:

$$y(0) = 0.15m, \quad \omega_n = \sqrt{2} \text{ rad/sec}, \quad \zeta = \frac{1}{\sqrt{2}} \quad \left(\frac{k}{M} = 2, \frac{b}{M} = 1 \right)$$

ΣΥΣΤΗΜΑ ΑΠΟΣΒΕΝΝΥΜΕΝΩΝ ΤΑΛΑΝΤΩΣΕΩΝ

(ΠΡΟΓΡΑΜΜΑ MATLAB)

```
>>y0=0.15;
>>wn=sqrt(2);
>>zeta=1/(2*sqrt(2));
>>t=[0:0.1:10];
>>unforced

unforced.m
% Υπολογισμός Ελεύθερης Απόκρισης με Δεδομένες Αρχικές Συνθήκες
%
c=(y0/sqrt(1-zeta^2));
y=c*exp(-zeta*wn*t).*sin(wn*sqrt(1-zeta^2)*t+acos(zeta));
%
bu=c*exp(-zeta*wn*t);bl=-bu;
%
plot(t,y,t,bu,'--',t,bl,'--'), grid
xlabel('Χρόνος (sec)'), ylabel('y(t) (Μέτρα)')
legend(['\omega_n=',num2str(wn),' \zeta=',num2str(zeta)])
```

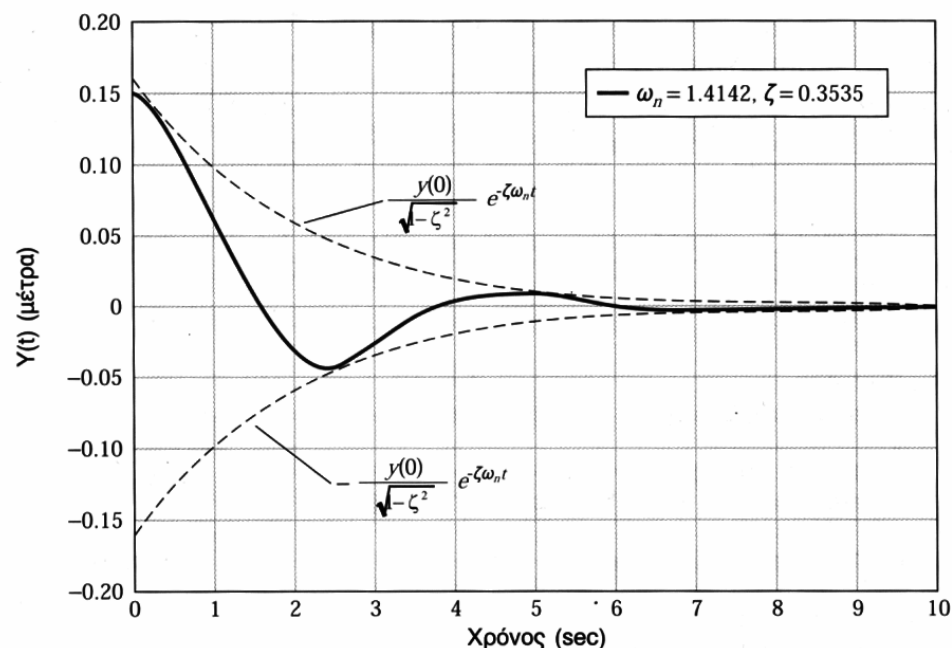
Οι εντολές του MATLAB με τις οποίες μπορούμε να δημιουργήσουμε την ζητούμενη γραφική παράσταση της απόκρισης, φαίνονται δίπλα. Στο πάνω μέρος φαίνονται οι εισαγωγές των αρχικών τιμών των μεταβλητών οι οποίες δίνονται με κατάλληλη πληκτρολόγηση στο περιβάλλον εργασίας (*workspace*), στην γραμμή εντολών (*command level* ή *command prompt*).

Στη συνέχεια εκτελούμε το αρχείο προγράμματος (θα ονομάζεται στο εξής για συντομία "*script*") "*unforced.m*", το οποίο παράγει τις ζητούμενες γραφικές παραστάσεις (*plots*). Με τον τρόπο αυτό έχουμε την δυνατότητα μιας αλληλοεπιδρώσας διαδικασίας ανάλυσης με την βοήθεια της οποίας μπορούμε να μελετήσουμε την επίδραση της φυσικής συχνότητας και της απόσβεσης στην ελεύθερη απόκριση του συστήματος κατά την μετατόπιση της μάζας.

ΣΥΣΤΗΜΑ ΑΠΟΣΒΕΝΝΥΜΕΝΩΝ ΤΑΛΑΝΤΩΣΕΩΝ

(ΠΡΟΓΡΑΜΜΑ ΜΑΤΛΑΒ)

Μπορούμε να διερευνήσουμε τις επιδράσεις που παρατηρούνται πάνω στην απόκριση του συστήματος εξαιτίας της φυσικής συχνότητας και του συντελεστή απόσβεσης, εισάγοντας κάθε φορά νέες τιμές για τις μεταβλητές ω_n και ζ στην γραμμή εντολών και στη συνέχεια εκτελώντας το script "unforced.m". Στο Σχήμα φαίνεται η γραφική παράσταση της απόκρισης του συστήματος στο πεδίο του χρόνου. Παρατηρήστε ότι το [script](#) τοποθετεί αυτόματα τις νέες κάθε φορά τιμές των μεταβλητών στις αντίστοιχες *ετικέτες (labels)* του γραφήματος, που αντιστοιχούν στην φυσική συχνότητα και τον συντελεστή απόσβεσης του συστήματος.



Στην περίπτωση του παραδείγματος, η λύση της αντίστοιχης διαφορικής εξίσωσης που δίνει την ελεύθερη απόκριση του, ήταν άμεσα διαθέσιμη. Γενικά, όταν θέλουμε να εξομοιώσουμε ένα σύστημα αυτομάτου ελέγχου κλειστού βρόχου στο οποίο εφαρμόζονται διάφορα σήματα εισόδου και για διάφορες τιμές αρχικών συνθηκών, είναι αρκετά δύσκολο να υπολογίσουμε αναλυτικά την αντίστοιχη λύση.

MATLAB ΚΑΙ ΑΝΑΛΥΣΗ ΣΥΣΤΗΜΑΤΩΝ

Το περιβάλλον του MATLAB χρησιμοποιείται για την ανάλυση συστημάτων που περιγράφονται από **συναρτήσεις μεταφοράς**. Εφόσον η συνάρτηση μεταφοράς είναι μια **ρητή συνάρτηση**, θα διερευνήσουμε τον τρόπο με τον οποίο το MATLAB **χειρίζεται παραστάσεις πολυωνύμων**, έχοντας υπόψη μας πάντοτε ότι όταν εργαζόμαστε με συναρτήσεις μεταφοράς θα πρέπει να καθορίζουμε τα **πολυώνυμα** του αριθμητή και του παρονομαστή.

Στο MATLAB, τα πολυώνυμα παριστάνονται από διανύσματα γραμμής τα οποία περιέχουν τους συντελεστές του εκάστοτε πολυωνύμου διατεταγμένους κατά τις κατιούσες δυνάμεις των αντίστοιχων όρων. Για παράδειγμα, το πολυώνυμο εισάγεται με τον τρόπο που φαίνεται στο σχήμα

$$p(s) = s^3 + 3s^2 + 4$$

```
>>p=[1 3 0 4];
>>r=roots(p)
r =
-3.3553
 0.1777+ 1.0773i
 0.1777- 1.0773i
>>p=poly(r)
p =
 1.0000  3.0000  0.0000  4.0000
```

$p(s) = s^3 + 3s^2 + 4$

Υπολογισμός των ριζών του $p(s) = 0$.

Σύνθεση του πολυωνύμου από τις ρίζες του.

Αν p είναι ένα **διάνυσμα γραμμής** το οποίο περιλαμβάνει τους συντελεστές του πολυωνύμου $p(s)$ κατά τις κατιούσες δυνάμεις των αντίστοιχων όρων, τότε η συνάρτηση **roots**(p) επιστρέφει ένα **διάνυσμα στήλης** το οποίο περιλαμβάνει τις ρίζες του πολυωνύμου. Αντιστρόφως, αν r είναι ένα διάνυσμα στήλης το οποίο περιλαμβάνει τις ρίζες ενός πολυωνύμου, τότε η συνάρτηση **poly**(r) επιστρέφει ένα διάνυσμα γραμμής το οποίο περιλαμβάνει τους συντελεστές του πολυωνύμου κατά τις κατιούσες δυνάμεις των αντίστοιχων όρων

MATLAB ΚΑΙ ΑΝΑΛΥΣΗ ΣΥΣΤΗΜΑΤΩΝ

Ο πολλαπλασιασμός πολυωνύμων επιτυγχάνεται με χρήση της συνάρτησης **conv**. Έστω ότι θέλουμε να φέρουμε το πολυώνυμο $n(s)$ στην κανονική του μορφή, όπου

$$n(s) = (3s^2 + 2s + 1)(s + 4)$$

Οι εντολές του MATLAB οι οποίες χρησιμοποιούν την συνάρτηση conv, φαίνονται στο Σχήμα. Έτσι το πολυώνυμο προκύπτει με την ακόλουθη μορφή

```
>>p=[3 2 1]; q=[1 4];
>>n=conv(p,q)
n=
 3  14  9  4
>>value=polyval(n,-5)
value =
-66
```

Πολλαπλασιασμός των p και q.

$n(s) = 3s^3 + 14s^2 + 9s + 4$

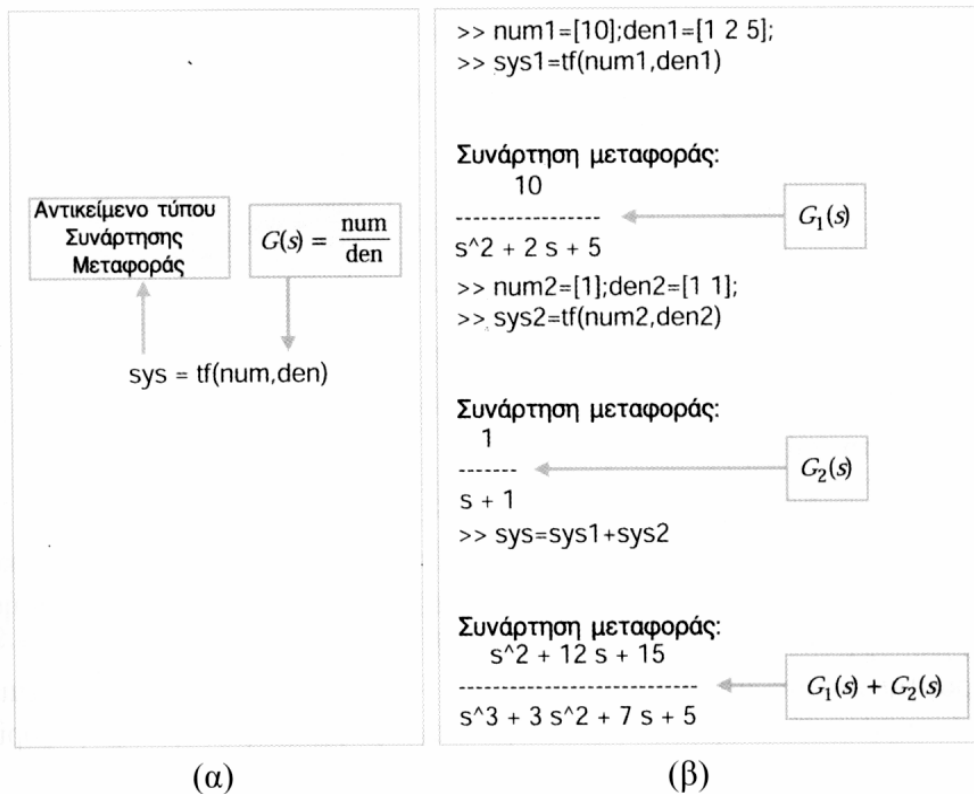
Υπολογισμός του $n(s)$ στο σημείο $s = -5$.

$$n(s) = 3s^3 + 14s^2 + 9s + 4$$

Η συνάρτηση **polyval** χρησιμοποιείται για τον υπολογισμό της τιμής ενός πολυωνύμου σε μια συγκεκριμένη τιμή της μεταβλητής του. Όπως φαίνεται στο σχήμα, το πολυώνυμο $n(s)$ λαμβάνει την τιμή $n(-5) = -66$

Η εργαλειοθήκη Control του MATLAB

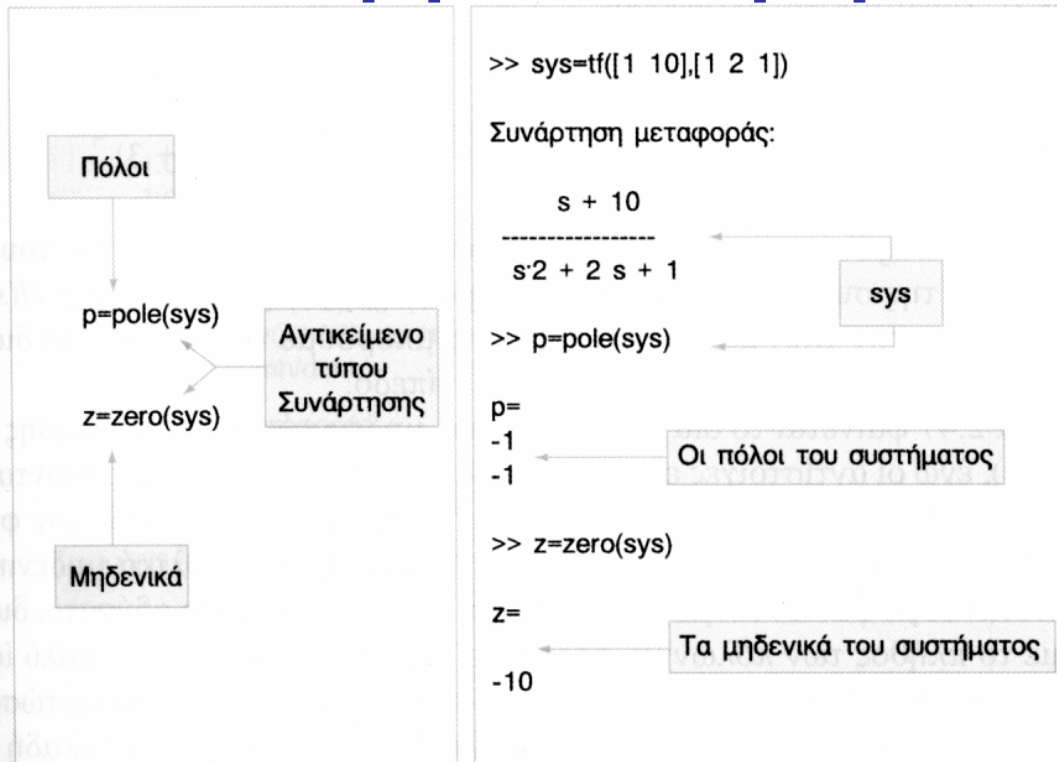
Η Εργαλειοθήκη των Συστημάτων Αυτομάτου Ελέγχου του MATLAB (*Control System Toolbox*) μεταχειρίζεται τα μοντέλα των γραμμικών και χρονικά αμετάβλητων συστημάτων ως **αντικείμενα** (*objects*), επιτρέποντας στον χρήστη να χειρίζεται τα διάφορα μοντέλα των συστημάτων ως απλές οντότητες



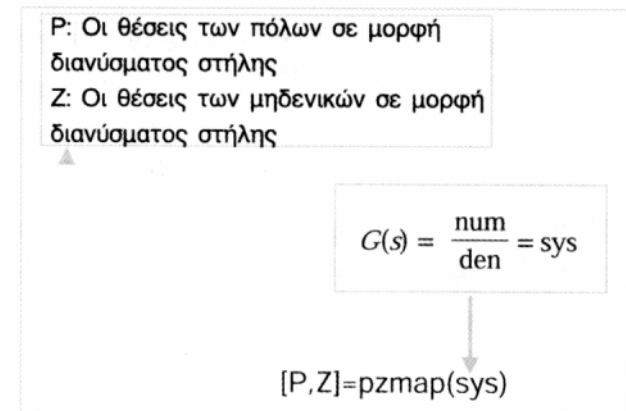
Στην περίπτωση των συναρτήσεων μεταφοράς, δημιουργούμε το μοντέλο του συστήματος με την βοήθεια της συνάρτησης **tf**. Για τα μοντέλα μεταβλητών κατάστασης (*state variable models*) χρησιμοποιείται η συνάρτηση **ss**

Στο MATLAB, τα διάφορα αντικείμενα (**objects**) που αντιστοιχούν στο μοντέλο κάποιου συστήματος, διαθέτουν ορισμένες **ιδιότητες αντικειμένου** (**object properties**) τις οποίες μπορούμε να τροποποιήσουμε. Παρόμοια, οι συναρτήσεις που λειτουργούν πάνω στα αντικείμενα των μοντέλων συστήματος, καλούνται **μέθοδοι** (**methods**). Για παράδειγμα, αν έχουμε δύο μοντέλα **G1** και **G2** συστήματος μπορούμε να τα προσθέσουμε χρησιμοποιώντας τον τελεστή "+", οπότε λαμβάνουμε το σύνθετο σύστημα

Η εργαλειοθήκη Control του MATLAB



Ο υπολογισμός των **πόλων και των μηδενικών** που αντιστοιχούν σε μια συνάρτηση μεταφοράς επιτυγχάνεται με την χρήση των συναρτήσεων **pole** και **zero** πάνω στο αντικείμενο τύπου συνάρτησης μεταφοράς αντίστοιχα, όπως φαίνεται στο Σχήμα



Με Χρήση της συνάρτησης pzmap . (α) (β) (*pole-zero map*), δημιουργείται ένα διάγραμμα στο οποίο φαίνονται οι θέσεις των πόλων και των μηδενικών στο *μιγαδικό επίπεδο (complex plane)*. Στο διάγραμμα πόλων-μηδενικών, οι θέσεις των μηδενικών συμβολίζονται με το "o" ενώ οι θέσεις των πόλων με το "x". Αν η συνάρτηση pzmap κληθεί χωρίς ορίσματα τότε το διάγραμμα δημιουργείται αυτόματα.

Παράδειγμα: Συναρτήσεις μεταφοράς

```

>>numg=[6 0 1]; deng=[1 3 3 1];sysg=tf(numg,deng);
>>z=zero(sysg)

```

Υπολογισμός των πόλων και των μηδενικών της $G(s)$.

```

z =
 0 + 0.4082i
 0 - 0.4082i

```

```

>>p=pole(sysg)

```

Συνάρτηση μεταφοράς $H(s)$

```

p =
-1.0000
-1.0000 + 0.0000i
-1.0000 - 0.0000i

```

```

>>n1=[1 1]; n2=[1 2]; d1=[1 2*i]; d2=[1 -2*i]; d3=[1 3];
>>numh=conv(n1,n2); denh=conv(d1,conv(d2,d3));
>>sysh=tf(numh/denh)

```

Συνάρτηση μεταφοράς:

$$H(s) = \frac{s^2 + 3s + 2}{s^3 + 3s^2 + 4s + 12}$$

$\frac{G(s)}{H(s)} = sys$

```

>>sys=sysg/sysh

```

Συνάρτηση μεταφοράς:

$$\frac{6s^5 + 18s^4 + 25s^3 + 75s^2 + 4s + 12}{s^5 + 6s^4 + 14s^3 + 16s^2 + 9s + 2}$$

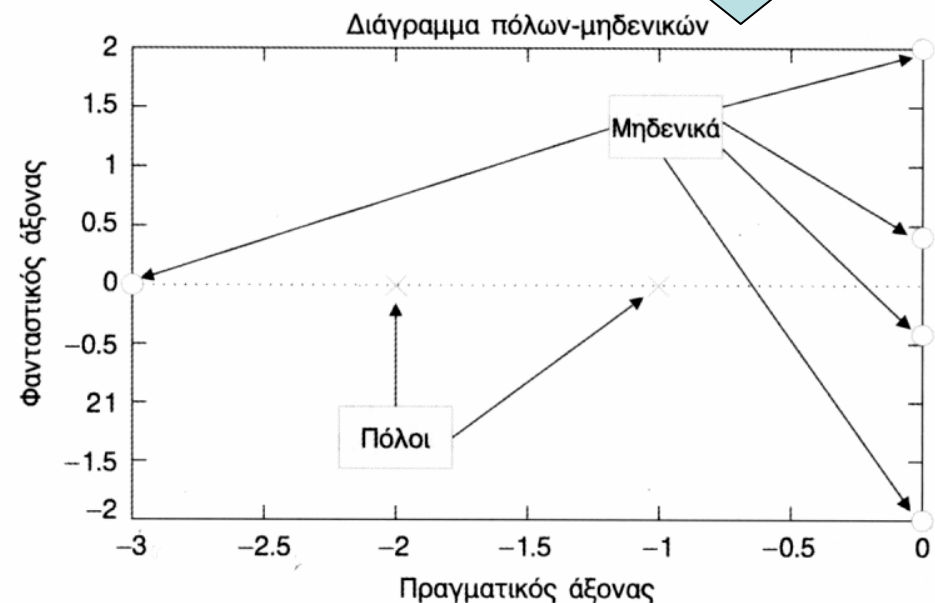
Διάγραμμα πόλων-μηδενικών

```

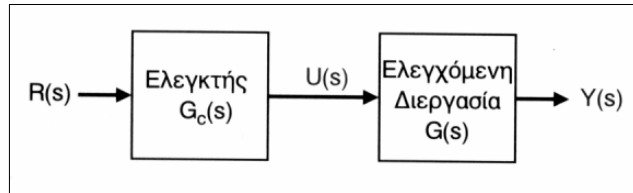
>>pzmap(sys)

```

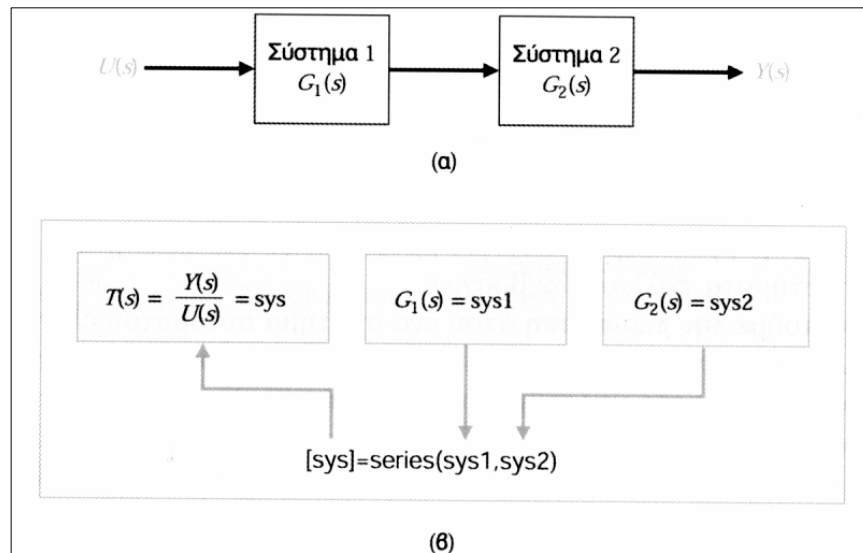
Χρησιμοποιώντας ένα script του MATLAB μπορούμε να υπολογίσουμε **τους πόλους και τα μηδενικά** της συνάρτησης $G(s)$, την χαρακτηριστική εξίσωση της $H(s)$ και να διαιρέσουμε τέλος την $G(s)$ με την $H(s)$. Επίσης μπορούμε να λάβουμε το **διάγραμμα πόλων-μηδενικών** της $G(s)/H(s)$ στο μιγαδικό επίπεδο



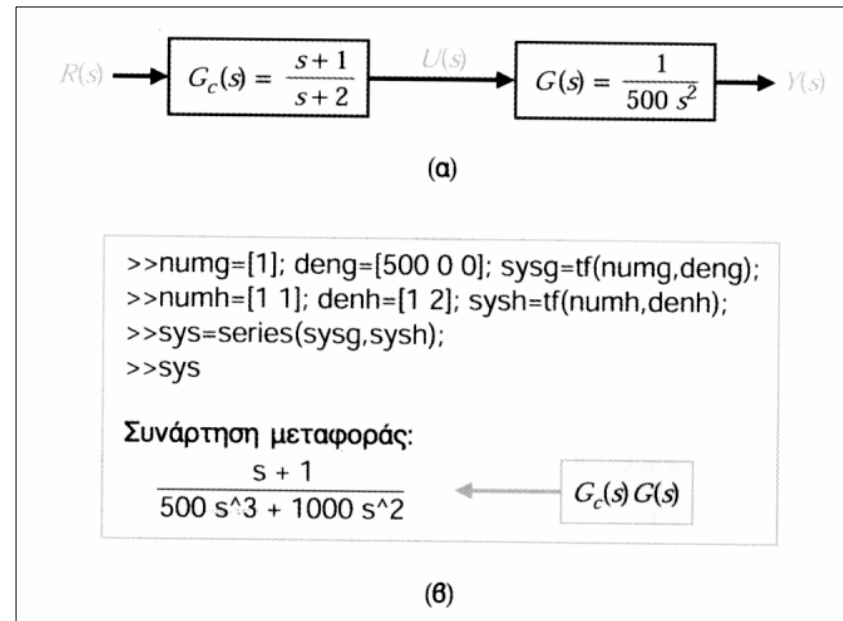
Σύνδεση συστημάτων σε σειρά



Είναι σύνηθες να συνδέουμε δύο ή περισσότερα συστήματα εν σειρά. (π.χ. τον ελεγκτή και το υπό έλεγχο σύστημα)

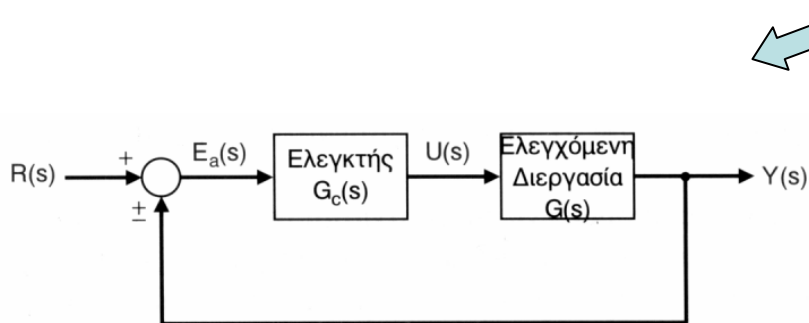
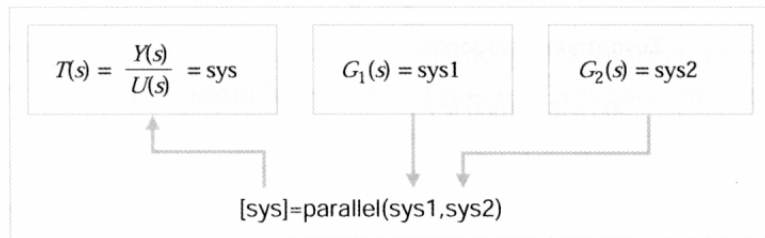
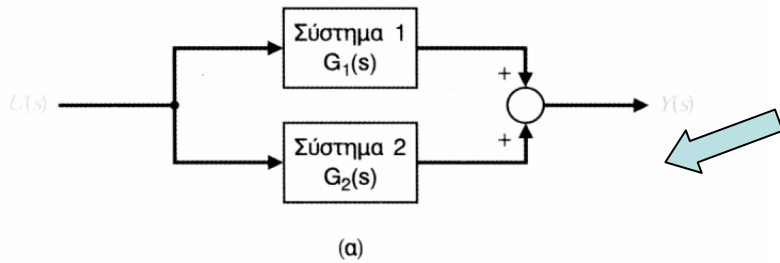


Στο MATLAB η σειριακή σύνδεση επιτυγχάνεται με τη χρήση της συνάρτησης **series**



Εδώ βλέπουμε την εφαρμογή σε ένα παράδειγμα. Είναι γνωστό ότι το σύνθετο σύστημα έχει συνάρτηση μεταφοράς ίση με το γινόμενο των δύο συναρτήσεων μεταφοράς των δύο συστημάτων

Παράλληλη σύνδεση και Χρήση Ανάδρασης

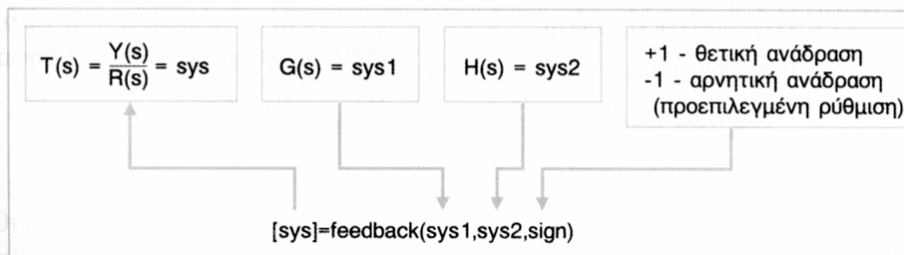
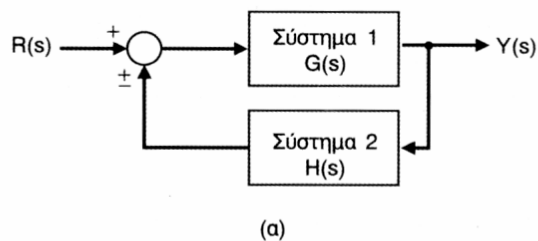
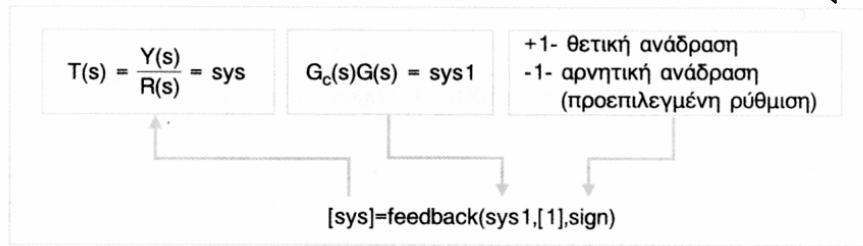
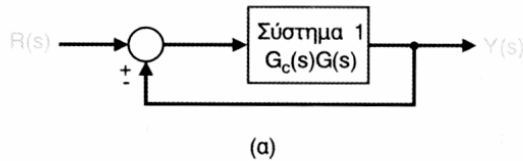


Στα λειτουργικά διαγράμματα εμφανίζονται συχνά συναρτήσεις μεταφοράς των οποίων οι αντίστοιχες μονάδες συνδέονται **παράλληλα**. Στις περιπτώσεις αυτές χρησιμοποιείται η συνάρτηση **parallel**. Η χρήση της συνάρτησης parallel, περιγράφεται στο σχήμα. Η σύνθετη συνάρτηση μεταφοράς προκύπτει από το άθροισμα των δύο επιμέρους συναρτήσεων

Μπορούμε επίσης να εισάγουμε ένα **σήμα ανάδρασης** στο σύστημα ελέγχου, κλείνοντας τον βρόχο με την λεγόμενη μοναδιαία ανάδραση (**unity feedback**), όπως φαίνεται στο σχήμα. Το σήμα $E_a(s)$ είναι το λεγόμενο σήμα σφάλματος (**error signal**). Το σήμα $R(s)$ αντιστοιχεί στην είσοδο αναφοράς (**reference input**). Στο συγκεκριμένο σύστημα, ο ελεγκτής βρίσκεται στον ευθύ κλάδο και η συνάρτηση μεταφοράς κλειστού βρόχου είναι,

$$T(s) = \frac{G_c(s)G(s)}{1 \mp G_c(s)G(s)}$$

Ανάδραση με τη συνάρτηση Feedback

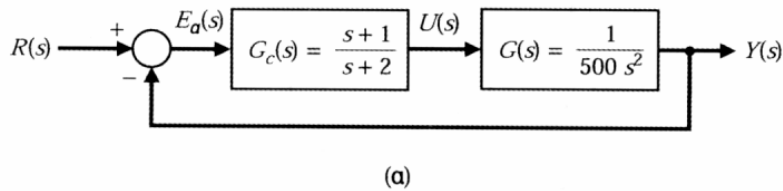


Μπορούμε να κάνουμε χρήση της συνάρτησης *feedback* για να διευκολύνουμε την διαδικασία απλοποίησης του λειτουργικού διαγράμματος, με σκοπό να υπολογίσουμε την τελική *συνάρτηση μεταφοράς κλειστού βρόχου* τόσο για συστήματα ενός βρόχου, όσο και για συστήματα πολλαπλών βρόχων.

Συχνά συναντούμε την περίπτωση όπου ένα σύστημα αυτομάτου ελέγχου κλειστού βρόχου περιλαμβάνει έναν κλάδο μοναδιαίας ανάδρασης, όπως φαίνεται στο σχήμα. Στις περιπτώσεις αυτές, μπορούμε να χρησιμοποιήσουμε την συνάρτηση *feedback* για τον υπολογισμό της αντίστοιχης συνάρτησης μεταφοράς κλειστού βρόχου, θέτοντας $H(s)=1$.

Στο διπλανό σχήμα φαίνεται η συνάρτηση *feedback* μαζί με την αντίστοιχη σύνθεση του συστήματος, όπου στον κλάδο ανάδρασης περιλαμβάνεται η συνάρτηση $H(s)$. **Αν παραλείψουμε την παράμετρο εισόδου "sign", τότε θεωρείται ότι έχουμε αρνητική ανάδραση**

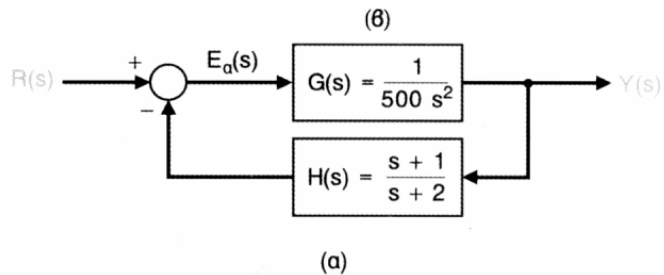
Παράδειγμα με τη συνάρτηση Feedback



```
>>numg=[1]; deng=[500 0 0]; sys1=tf(numg,deng);
>>numc=[1 1]; denc=[1 2]; sys2=tf(numc,denc);
>>sys3=series(sys1,sys2);
>>sys=feedback(sys3,[1])
```

Transfer function:

$$\frac{s + 1}{500 s^3 + 1000 s^2 + s + 1}$$

$$\frac{Y(s)}{R(s)} = \frac{G_c(s)G(s)}{1 + G_c(s)G(s)}$$


```
>>numg=[1]; deng=[500 0 0]; sys1=tf(numg,deng);
>>numh=[1 1]; denh=[1 2]; sys2=tf(numh,denh);
>>sys=feedback(sys1,sys2);
>>sys
```

Συνάρτηση μεταφοράς:

$$\frac{s + 2}{500 s^3 + 1000 s^2 + s + 1}$$

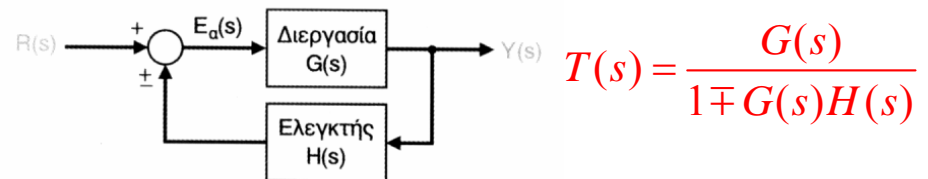
$$\frac{Y(s)}{R(s)} = \frac{G(s)}{1 + G(s)H(s)}$$

(β)

Η Συνάρτηση Feedback σε Σύστημα με Κλάδο Μοναδιαίας Ανάδρασης

Έστω η διεργασία και ο αντίστοιχος ελεγκτής που παριστάνονται από τις συναρτήσεις μεταφοράς $G_1(s)$ και $G_2(s)$ αντίστοιχα, όπως δίνονται στο διπλανό σχήμα. Για να χρησιμοποιήσουμε την συνάρτηση *feedback* θα πρέπει πρώτα να εκτελέσουμε την συνάρτηση *series*, ώστε να προκύψει το σύστημα $G_1(s)G_2(s)$ και στη συνέχεια εκτελούμε την συνάρτηση *feedback* για να κλείσουμε τον βρόχο. Η ακολουθία των εντολών που απαιτείται για τον σκοπό αυτό δίνεται στο σχήμα

Μια άλλη βασική δομή ενός συστήματος αυτομάτου ελέγχου, φαίνεται παρακάτω.

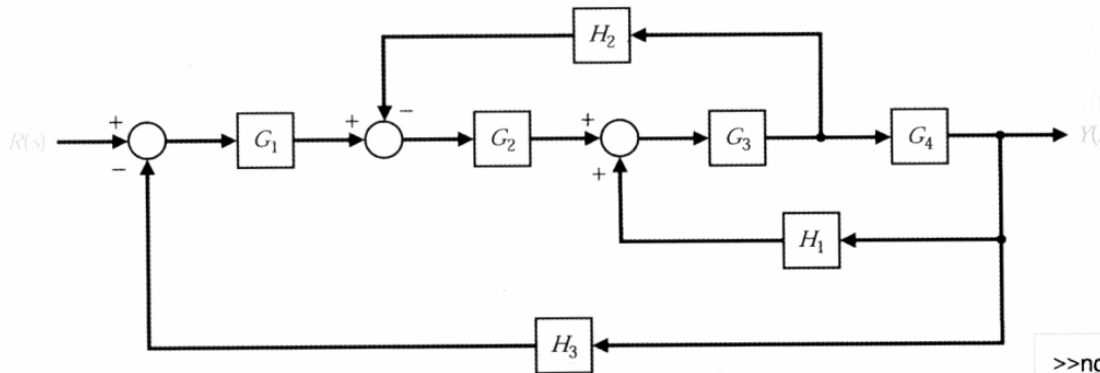


Στην περίπτωση αυτή η μονάδα του ελεγκτή τοποθετείται στον κλάδο της ανάδρασης.

Για τον υπολογισμό της συνάρτησης μεταφοράς κλειστού βρόχου με την μονάδα του ελεγκτή στον κλάδο της ανάδρασης, χρησιμοποιούμε την συνάρτηση *feedback*.

Απλοποίηση λειτουργικών διαγραμμάτων

Οι συναρτήσεις του MATLAB, *series*, *parallel* και *feedback*, μπορούν να χρησιμοποιηθούν για την διαδικασία μετασχηματισμών των λειτουργικών διαγραμμάτων με πολλαπλούς βρόχους. Ας θεωρήσουμε το σύνθετο σύστημα του σχήματος. Θέλουμε να υπολογίσουμε την αντίστοιχη συνάρτηση μεταφοράς κλειστού βρόγχου.



$$G_1(s) = \frac{1}{s+10} \quad G_2(s) = \frac{1}{s+1}$$

$$G_3(s) = \frac{s^2+1}{s^2+4s+4} \quad G_4(s) = \frac{s+1}{s+6}$$

$$H_1(s) = \frac{s+1}{s+2} \quad H_2(s) = 2 \quad H_3(s) = 1$$

Στο παράδειγμα αυτό ακολουθούμε μια διαδικασία πέντε βημάτων:

1. Εισάγουμε τις συναρτήσεις μεταφοράς στο περιβάλλον του MATLAB.
2. Μετακινούμε την μονάδα H_2 πίσω από την μονάδα G_4 .
3. Απαλείφουμε τον βρόχο $G_3 G_4 H_1$.
4. Απαλείφουμε τον βρόχο που περιλαμβάνει την μονάδα H_1 .
5. Απαλείφουμε τον βρόχο που απομένει και υπολογίζουμε την $T(s)$.

```
>>ng1=[1]; dg1=[1 10]; sysg1=tf(ng1,dg1);
>>ng2=[1]; dg2=[1 1]; sysg2=tf(ng2,dg2);
>>ng3=[1 0 1]; dg3=[1 4 4]; sysg3=tf(ng3,dg3);
>>ng4=[1 1]; dg4=[1 6]; sysg4=tf(ng4,dg4);
>>nh1=[1 1]; dh1=[1 2]; sysh1=tf(nh1,dh1);
>>nh2=[2]; dh2=[1]; sysh2=tf(nh2,dh2);
>>nh3=[1]; dh3=[1]; sysh3=tf(nh3,dh3);
>>sys1=sys2/sysg4;
>>sys2=series(sysg3,sysg4);
>>sys3=feedback(sys2,sysh1,+1);
>>sys4=series(sysg2,sys3);
>>sys5=feedback(sys4,sys1);
>>sys6=series(sysg1,sys5);
>>sys=feedback(sys6,[1]);
```

Συνάρτηση μεταφοράς:

$$\frac{s^5 + 4s^4 + 6s^3 + 6s^2 + 5s + 2}{12s^6 + 205s^5 + 1066s^4 + 2517s^3 + 3128s^2 + 2196s + 712}$$

- Βήμα 1
- Βήμα 2
- Βήμα 3
- Βήμα 4
- Βήμα 5

Χειρισμός των μοντέλων κατάστασης

Το μοντέλο ενός Γραμμικού, Χρονικά Αμετάβλητου συστήματος στο χώρο κατάστασης δίνεται από τη σχέση

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t)$$

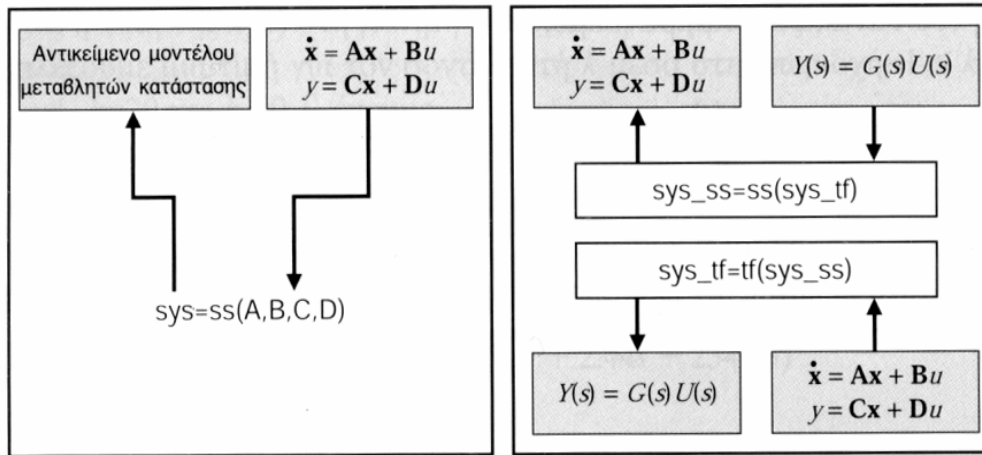
$$\mathbf{y}(t) = \mathbf{C}\mathbf{x}(t) + \mathbf{D}\mathbf{u}(t)$$

Εφόσον η μεταβλητή τύπου πίνακα (ή διανύσματος) αποτελεί μια από τις βασικές υπολογιστικές δομές στο περιβάλλον MATLAB, το μοντέλο των μεταβλητών κατάστασης προσαρμόζεται αρκετά ικανοποιητικά στο περιβάλλον αυτό. Στην πραγματικότητα, το πακέτο λογισμικού MATLAB καλύπτει πάρα πολλά θέματα σχετικά με τις διάφορες μεθόδους που χρησιμοποιούνται στα μοντέλα των μεταβλητών κατάστασης ενός συστήματος

Δύο νέες συναρτήσεις του MATLAB που θα γνωρίσουμε αμέσως παρακάτω, είναι η συνάρτηση **ss** και η συνάρτηση **lsim**. Επίσης θα χρησιμοποιήσουμε και την συνάρτηση **expm** για τον υπολογισμό του πίνακα μετάβασης.

Αν γνωρίζουμε την συνάρτηση μεταφοράς ενός συστήματος μπορούμε να κατασκευάσουμε το αντίστοιχο μοντέλο μεταβλητών κατάστασης και αντίστροφα. Η συνάρτηση **tf** χρησιμοποιείται για την μετατροπή της παράστασης ενός συστήματος από το μοντέλο των μεταβλητών κατάστασης στο μοντέλο της συνάρτησης μεταφοράς. Αντίθετα, η συνάρτηση **ss** χρησιμοποιείται για την μετατροπή της παράστασης ενός συστήματος από το μοντέλο της συνάρτησης μεταφοράς στο μοντέλο των μεταβλητών κατάστασης

Χειρισμός των μοντέλων κατάστασης



Ας θεωρήσουμε, για παράδειγμα την ακόλουθη συνάρτηση μεταφοράς ενός συστήματος τρίτης τάξης

$$T(s) = \frac{Y(s)}{R(s)} = \frac{2s^2 + 8s + 6}{s^3 + 8s^2 + 16s + 6}$$

Μπορούμε να χρησιμοποιήσουμε την συνάρτηση **ss** για να λάβουμε την αντίστοιχη παράσταση του συστήματος στο μοντέλο των μεταβλητών κατάστασης που είναι

$$\dot{\mathbf{x}}(t) = \begin{bmatrix} -8 & -2 & -0.75 \\ 8 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \mathbf{x}(t) + \begin{bmatrix} 2 \\ 0 \\ 0 \end{bmatrix} \mathbf{u}(t)$$

$$\mathbf{y}(t) = [1 \quad 0.5 \quad 0.375] \mathbf{x}(t) + [0] \mathbf{u}(t)$$

```
num = [2 8 6];
den = [1 8 16 6];
sys_tf = tf(num,den);
sys_ss = ss(sys_tf)
```

Και αντίστροφα

```
a = [-8 -2 -0.75; 8 0 0; 0 1 0];
b = [2; 0; 0];
c = [1 0.5 0.375]; d = [0];
sys_ss = ss(a,b,c,d);
sys_tf = tf(sys_ss)
```

Χειρισμός των μοντέλων κατάστασης

Υπολογισμός του μεταβατικού πίνακα $\exp(At)$

Η χρονική απόκριση του συστήματος που περιγράφεται από τις Εξισώσεις κατάστασης προκύπτει από τη λύση της διανυσματικής διαφορικής εξίσωσης που για ΓΧΑ συστήματα είναι η

$$\mathbf{x}(t) = \exp(\mathbf{A}t)\mathbf{x}(0) + \int_0^t \exp[\mathbf{A}(t-\tau)]\mathbf{B}u(t)d\tau$$

Η συνάρτηση **expm** υπολογίζει την εκθετική συνάρτηση πίνακα. Για παράδειγμα οι εντολές

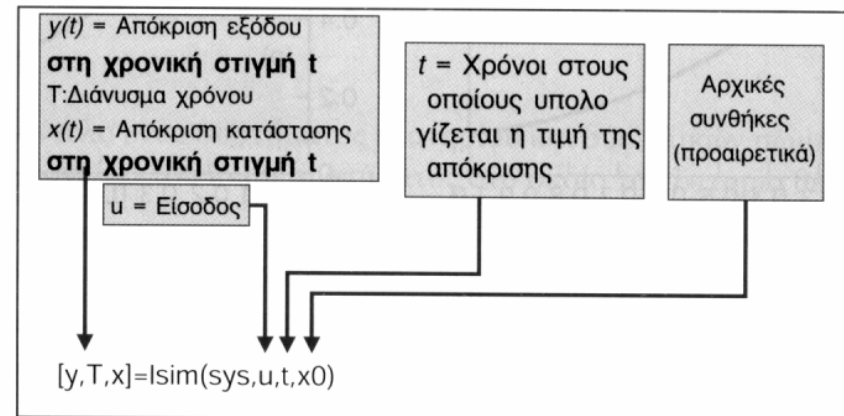
```
> a=[0 -2; 1 -3];  
>> t=0.2;  
>> pinexp=expm(a*t);
```

Δίνουν
αποτέλεσμα

```
pinexp =  
0.9671 -0.2968  
0.1484 0.5219
```

Υπολογισμός της χρονικής απόκρισης με τη συνάρτηση **lsim**

Η συνάρτηση **lsim** μπορεί να δεχθεί μη μηδενικές αρχικές συνθήκες καθώς επίσης και μια οποιαδήποτε συνάρτηση εισόδου. Με την βοήθεια της συνάρτησης **lsim** μπορούμε να υπολογίσουμε σε κάθε χρονική στιγμή την τιμή της κατάστασης και κατά συνέπεια της εξόδου. Η συνάρτηση πραγματοποιεί αριθμητική επίλυση των διαφορικών εξισώσεων (Θυμηθείτε τη μέθοδο Runge-Kutta)



Περισσότερες δυνατότητες – Χρήση βοήθειας

>> help control

Control System Toolbox.
Version 5.2 (R13) 28-Jun-2002

General.

ctrlpref - Set Control System Toolbox preferences.

ltimodels - Detailed help on the various types of LTI models.

ltiprops - Detailed help on available LTI model properties.

Creating linear models.

tf - Create transfer function models.

zpk - Create zero/pole/gain models.

ss, dss - Create state-space models.

frd - Create a frequency response data models.

filt - Specify a digital filter.

set - Set/modify properties of LTI models.

Data extraction.

tfdata - Extract numerator(s) and denominator(s).

... κλπ

Μπορούμε να δούμε περισσότερα για τις δυνατότητες της εργαλειοθήκης ΣΑΕ με τη χρήση της εντολής help από την γραμμή εντολών ή ανατρέχοντας στα εγχειρίδια χρήσης

>> help tf

TF Creation of transfer functions or conversion to transfer function.

Creation:

SYS = TF(NUM,DEN) creates a continuous-time transfer function SYS with numerator(s) NUM and denominator(s) DEN. The output SYS is a TF object.

SYS = TF(NUM,DEN,TS) creates a discrete-time transfer function with sample time TS (set TS=-1 if the sample time is undetermined).

... κλπ

ΠΡΟΣΟΜΟΙΩΣΗ ΜΕ ΧΡΗΣΗ ΤΟΥ ΓΡΑΦΙΚΟΥ ΠΕΡΙΒΑΛΛΟΝΤΟΣ SIMULINK (2/2)

- Για να βάλουμε είσοδο στο σύστημα πηγαίνουμε στο μενού Sources και διαλέγουμε την επιθυμητή είσοδο π.χ. step
- Για την απόκριση του συστήματος χρησιμοποιούμε το block **Scope** το οποίο βρίσκεται στην βιβλιοθήκη Simulink → Sinks
- ✓ Αν θέλουμε να περιστρέψουμε ένα block τότε το επιλέγουμε και πατάμε το συνδυασμό των πλήκτρων Ctrl+R
- Για να ξεκινήσει η προσομοίωση επιλέγουμε Start από το μενού Simulation και με διπλό κλικ στο Scope λαμβάνουμε τα αποτελέσματα.