

**ΣΥΝΟΠΤΙΚΟΣ ΟΔΗΓΟΣ ΜΑΤLAB – SIMULINK
ΓΙΑ ΤΟ ΜΑΘΗΜΑ ΣΑΕ Ι**

ΑΡΝΑΟΥΤΑΚΗΣ ΝΕΚΤΑΡΙΟΣ

**ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΑΡΑΓΩΓΗΣ ΚΑΙ ΔΙΟΙΚΗΣΗΣ
ΠΟΛΥΤΕΧΝΕΙΟ ΚΡΗΤΗΣ**

ΣΕΠΤΕΜΒΡΙΟΣ 2002

ΣΥΝΟΠΤΙΚΟΣ ΟΔΗΓΟΣ MATLAB – SIMULINK ΓΙΑ ΤΟ ΜΑΘΗΜΑ ΣΑΕ Ι

A. Περιγραφή και βασικές λειτουργίες της MATLAB (εκδόσεις 5.x)

1. Γενικά

Η γλώσσα προγραμματισμού MATLAB (το όνομα προήλθε από τις λέξεις Matrix Laboratory) λειτουργεί ως διερμηνέας εντολών (command interpreter), οι οποίες δίνονται μέσω του παραθύρου εντολών της (MATLAB command window). Οι εντολές αυτές μπορεί να είναι:

1. ορισμοί μεταβλητών και πράξεις
2. κλήση ενσωματωμένων συναρτήσεων της MATLAB και των εγκατεστημένων εργαλειοθηκών της (toolboxes)
3. κλήση συναρτήσεων (functions) ή αρχείων εντολών MATLAB (scripts) που κατασκευάζονται από τους χρήστες με τη μορφή m-file

Η ευρεία χρήση της MATLAB οφείλεται σε μεγάλο βαθμό στην επεκτασιμότητα της μέσω των διάφορων εργαλειοθηκών, κάθε μια από τις οποίες περιέχει ένα αριθμό συναρτήσεων για ένα συγκεκριμένο αντικείμενο. Η δομή των υπάρχοντων στοιχείων σε μια εγκατάσταση MATLAB παρουσιάζεται εκτελώντας την εντολή *help* (γράφουμε *help* δίπλα από την προτροπή » και πατάμε enter):

» help

HELP topics:

matlab\general	- General purpose commands.
matlab\ops	- Operators and special characters.
matlab\lang	- Programming language constructs.
matlab\elmat	- Elementary matrices and matrix manipulation.
matlab\elfun	- Elementary math functions.
matlab\specfun	- Specialized math functions.
matlab\matfun	- Matrix functions - numerical linear algebra.
matlab\datafun	- Data analysis and Fourier transforms.
matlab\polyfun	- Interpolation and polynomials.
matlab\funfun	- Function functions and ODE solvers.
matlab\sparfun	- Sparse matrices.
matlab\graph2d	- Two dimensional graphs.
matlab\graph3d	- Three dimensional graphs.
matlab\specgraph	- Specialized graphs.
matlab\graphics	- Handle Graphics.
matlab\uitools	- Graphical user interface tools.
matlab\strfun	- Character strings.
matlab\iofun	- File input/output.
matlab\timefun	- Time and dates.
matlab\datatypes	- Data types and structures.
matlab\winfun	- Windows Operating System Interface Files (DDE/ActiveX)
matlab\demos	- Examples and demonstrations.
toolbox\control	- Control System Toolbox.
.....	
MATLABR11\work	- (No table of contents file)
toolbox\local	- Preferences.

For more help on directory/topic, type "help topic".

Οι συναρτήσεις που περιέχονται σε κάθε ένα κατάλογο ή εργαλειοθήκη εμφανίζονται δίνοντας την εντολή *help όνομα_καταλόγου*. Για παράδειγμα, αν θέλουμε να δούμε τις συναρτήσεις που περιέχονται στην εργαλειοθήκη control, δίνουμε:

» help control

Control System Toolbox.
Version 4.2.1 (R11.1) 10-Sep-1999

What's new.

Readme - New features and enhancements in this version.

Creation of LTI models.

tf - Create a transfer function model.
zpk - Create a zero/pole/gain model.
ss - Create a state-space model.
dss - Create a descriptor state-space model.
frd - Create a frequency response data model.
filt - Specify a digital filter.
set - Set/modify properties of LTI models.
ltimodels - Detailed help on various types of LTI models.
ltiprops - Detailed help on available LTI properties.

.....

Για να πάρουμε πληροφορίες για μια συγκεκριμένη συνάρτηση, δίνουμε την εντολή *help όνομα_συνάρτησης*. Για παράδειγμα, αν θέλουμε πληροφορίες για τη συνάρτηση tf δίνουμε

» 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.

.....

conversion:

$SYS = TF(SYS)$ converts an arbitrary LTI model SYS to the transfer function representation. The result is a TF object.

$SYS = TF(SYS,'inv')$ uses a fast algorithm for conversion from state space to TF, but is typically less accurate for high-order systems.

See also LTIMODELS, FILT, SET, GET, TFDATA, SUBSREF, LTIPROPS, ZPK, SS.

Η τελευταία γραμμή της απάντησης παραθέτει τα ονόματα συναρτήσεων που είναι συναφείς με αυτή για την οποία ζητήθηκε βοήθεια.

Στα πλαίσια των εργαστηριακών ασκήσεων του μαθήματος ΣΑΕ Ι είναι απαραίτητη η χρήση ορισμένων βασικών συναρτήσεων και τελεστών της MATLAB, καθώς και η χρήση ορισμένων συναρτήσεων της εργαλειοθήκης συστημάτων ελέγχου (control systems toolbox). Οι συναρτήσεις αυτές θα παρουσιαστούν σε επόμενη ενότητα μαζί με ορισμένες υποδείξεις για τη χρήση τους στην επίλυση των εργαστηριακών ασκήσεων.

2. Ορισμοί μεταβλητών και βασικές πράξεις

Κάθε φορά που εκκινείται η MATLAB δημιουργείται στη μνήμη του υπολογιστή ο *χώρος εργασίας* (workspace) εντός του οποίου αποθηκεύονται οι οριζόμενες στο παράθυρο εντολών μεταβλητές. Οι μεταβλητές αυτές είναι διαθέσιμες μέχρι την έξοδο από τη MATLAB, ενώ είναι δυνατή η αποθήκευση τους στο δίσκο και η ανάκτηση τους σε επόμενη εκκίνηση της MATLAB. Ο εξ ορισμού τύπος μεταβλητής είναι πραγματικός διπλής ακρίβειας. Άλλοι τύποι μεταβλητών που θα χρησιμοποιηθούν στα πλαίσια των εργαστηριακών ασκήσεων είναι ο μιγαδικός αριθμός και η μεταβλητή-αντικείμενο συνάρτησης μεταφοράς. Είναι δυνατή η χρήση και άλλων τύπων μεταβλητών, η περιγραφή των οποίων είναι εκτός του σκοπού του παρόντος οδηγού.

Η δήλωση μιας μεταβλητής στο παράθυρο εντολών γίνεται ταυτόχρονα με την απόδοση τιμής σε αυτήν. Για παράδειγμα, ο ορισμός μιας μεταβλητής *a* στην οποία αποδίδουμε την τιμή 10 γίνεται με την εντολή

```
» a=10
```

```
a =  
    10
```

⇒ *Αν δεν επιθυμούμε να εμφανίζεται το αποτέλεσμα της εκτέλεσης μιας εντολής, βάζουμε στο τέλος της εντολής ένα ελληνικό ερωτηματικό:*

```
» A=5;
```

Για να δούμε ποιές μεταβλητές υπάρχουν στο χώρο εργασίας, δίνουμε την εντολή

```
» who
```

```
Your variables are:
```

```
A      a
```

⇒ *Η MATLAB κάνει διάκριση πεζών-κεφαλαίων στα ονόματα μεταβλητών.*

Για να δούμε την τιμή μιας μεταβλητής, δίνουμε το όνομα της στη γραμμή εντολής:

```
» A
```

```
A =  
    5
```

Η διαγραφή μιας μεταβλητής από το χώρο εργασίας γίνεται δίνοντας την εντολή *clear όνομα_μεταβλητής*. Αν θέλουμε να διαγράψουμε όλες τις υπάρχουσες μεταβλητές, δίνουμε απλώς *clear* (προσοχή στη χρήση...)

Μέχρι στιγμής είδαμε τον τρόπο ορισμού μιας βαθμωτής πραγματικής μεταβλητής. Στη συνέχεια παρουσιάζεται ο τρόπος ορισμού πραγματικών διανυσμάτων και πινάκων.

- Ορισμός διανύσματος γραμμής:

» $a=[1\ 2\ 3]$

a =

1 2 3

- Ορισμός διανύσματος στήλης:

» $b=[1;2;3]$

b =

1

2

3

- Ορισμός πίνακα:

» $A=[1\ 2\ 3;4\ 5\ 6]$

A =

1 2 3

4 5 6

- Ορισμός διανύσματος x τα στοιχεία του οποίου ανήκουν στο διάστημα [a,b] και απέχουν μεταξύ τους βήμα d: $x=a:d:b$. Για παράδειγμα

» $x=0:0.1:1$

x =

Columns 1 through 7

0 0.1000 0.2000 0.3000 0.4000 0.5000 0.6000

Columns 8 through 11

0.7000 0.8000 0.9000 1.0000

(το x ορίζεται ως διάνυσμα γραμμής).

- Πρόσβαση σε μεμονωμένο στοιχείο διανύσματος ή πίνακα: χρήση δεικτών (indexes) γραμμής/στήλης. Για παράδειγμα

» $A(1,3)$

ans =

3

- Πρόσβαση σε μια γραμμή (στήλη) πίνακα: μέσω του τελεστή ":". Για παράδειγμα

» $A(1,:)$ δίνει την πρώτη γραμμή του A

» $A(:,2)$ δίνει τη δεύτερη στήλη του A

Σημείωση: οι δείκτες γραμμής/στήλης στη MATLAB παίρνουν μόνο θετικές ακέραιες τιμές (όχι 0 όπως συμβαίνει σε κάποιες γλώσσες προγραμματισμού).

Οι αριθμητικοί τελεστές της MATLAB είναι κατασκευασμένοι ώστε να δέχονται ως ορίσματα όχι μόνο βαθμωτές μεταβλητές, αλλά και μεταβλητές τύπου πραγματικού πίνακα ή διανύσματος. Το αποτέλεσμα της χρήσης ενός τελεστή εξαρτάται από τους

τύπους των ορισμάτων του. Οι τελεστές και οι ειδικοί χαρακτήρες της MATLAB παρουσιάζονται εκτελώντας την εντολή *help ops*.

Στη συνέχεια δίνονται ορισμένα παραδείγματα χρήσης των βασικών αριθμητικών τελεστών της MATLAB. Για την εκτέλεση των εντολών που ακολουθούν είναι απαραίτητο να έχουν προηγουμένως ορισθεί οι εξής μεταβλητές:

```
a = 2
b = [0 1 2]
c = [1 2 3]
```

» $b+c$ ($b-c$) Πρόσθεση (αφαίρεση) διανυσμάτων/πινάκων

```
ans =
    1    3    5   (-1  -1  -1)
```

⇒ Η μεταβλητή *ans* χρησιμοποιείται εξ' ορισμού για να αποθηκεύσει το αποτέλεσμα εντολής που δεν εκχωρείται σε μεταβλητή οριζόμενη από το χρήστη.

» $b.*c$ Πολλαπλασιασμός στοιχείων διανυσμάτων

```
ans =
    0    2    6
```

» $d=c'$ Ανάστροφος διανύσματος/πίνακα (τελεστής ')

```
d =
    1
    2
    3
```

» $d*b$ Πολλαπλασιασμός πινάκων

```
ans =
    0    1    2
    0    2    4
    0    3    6
```

» $a*b$ (ή $a.*b$) Πολλαπλασιασμός διανύσματος με αριθμό

```
ans =
    0    2    4
```

» $b./c$ Διάρθρωση στοιχείων διανυσμάτων

```
ans =
    0  0.5000  0.6667
```

» b/a (ή $b./a$) Διάρθρωση διανύσματος με αριθμό

```
ans =
    0  0.5000  1.0000
```

» $b.^a$ Ύψωση στοιχείων διανύσματος σε δύναμη

```
ans =  
    0    1    4
```

⇒ Αν και τα δυο ορίσματα είναι βαθμωτά, οι τελεστές $*$ και $./$ είναι ισοδύναμοι με τους τελεστές $*$ και $/$, αντίστοιχα.

⇒ Πολλά μηνύματα λάθους κατά την εκτέλεση πράξεων οφείλονται στην ασυμφωνία διαστάσεων των ορισμάτων.

Παράδειγμα (προς αποφυγή...): απόπειρα πρόσθεσης διανύσματος γραμμής με διάνυσμα στήλης

```
» b+d  
??? Error using ==> +  
Matrix dimensions must agree.
```

Οι εντολές *save* και *load* (αποθήκευση/ανάκτηση μεταβλητών)

Όπως αναφέρθηκε στην αρχή της ενότητας, είναι δυνατή η αποθήκευση μιας μεταβλητής (ή και όλων των μεταβλητών) του χώρου εργασίας στο δίσκο του υπολογιστή, αλλά και η ανάκτηση αποθηκευμένων μεταβλητών από το δίσκο στο χώρο εργασίας. Η αποθήκευση μεταβλητής γίνεται με την εντολή *save*, η οποία συντάσσεται ως

```
save fname X
```

όπου *fname* είναι το όνομα αρχείου και *X* η αποθηκευόμενη μεταβλητή. Εξ' ορισμού το αρχείο αποθήκευσης έχει την κατάληξη *.mat* και είναι δυαδικό (binary). Είναι δυνατή η αποθήκευση και σε αρχείο κειμένου (text) αν δοθεί η εντολή ως

```
save fname X -ascii
```

Η ανάκτηση μεταβλητών αποθηκευμένων σε αρχεία τύπου *.mat* ή αρχεία κειμένου γίνεται με την εντολή *load*. Η εντολή αυτή συντάσσεται ως

```
load fname
```

εάν το αρχείο με όνομα *fname* είναι τύπου *.mat*. Οι ανακτώμενες από το αρχείο μεταβλητές είναι διαθέσιμες στο χώρο εργασίας με τα ονόματα που είχαν χρησιμοποιηθεί κατά την αποθήκευσή τους. Στην περίπτωση όπου το αρχείο αποθήκευσης είναι τύπου κειμένου (περιέχει γραμμές αριθμητικών τιμών διαχωρισμένων με κενά), χρησιμοποιείται η σύνταξη

```
load fname.ext
```

όπου *.ext* η κατάληξη του αρχείου. Οι τιμές που περιέχει το αρχείο είναι διαθέσιμες στο χώρο εργασίας ως μια μεταβλητή με όνομα το όνομα του αρχείου χωρίς την κατάληξη.

⇒ Το *fname* είναι πλήρες όνομα αρχείου (περιλαμβάνει και τη διαδρομή στο δίσκο).

Παράδειγμα:

```
» a=[1 2 3];  
» save c:\alpha.dat a -ascii   Αποθήκευση του διανύσματος a σε αρχείο κειμένου  
» load c:\alpha.dat           Ανάκτηση των περιεχομένων του αρχείου alpha.dat στη  
                               μεταβλητή του χώρου εργασίας με όνομα alpha
```

3. Χρήση συναρτήσεων MATLAB για την επίλυση των εργαστηριακών ασκήσεων

Στην ενότητα αυτή δίνεται μια συνοπτική περιγραφή των συναρτήσεων της MATLAB και της εργαλειοθήκης συστημάτων ελέγχου που χρησιμεύουν στην επίλυση των εργαστηριακών ασκήσεων. Περισσότερες πληροφορίες για τις συναρτήσεις αυτές είναι διαθέσιμες μέσω της βοήθειας της MATLAB (δείτε σχετικά την ενότητα 1 του παρόντος).

Η κλήση μιας συνάρτησης γίνεται από το παράθυρο εντολών της MATLAB με την ακόλουθη γενική σύνταξη:

$$[A_1, A_2, \dots, A_n] = \text{function_name}(a_1, a_2, \dots, a_m)$$

όπου $[A_1, A_2, \dots, A_n]$: η λίστα μεταβλητών στις οποίες επιστρέφονται τα αποτελέσματα της συνάρτησης (ορίσματα εξόδου), *function_name*: το όνομα της συνάρτησης και a_1, a_2, \dots, a_m : η λίστα ορισμάτων εισόδου της συνάρτησης. Το πλήθος και ο τύπος των ορισμάτων εισόδου και εξόδου εξαρτώνται από την καλούμενη συνάρτηση. Αν η λίστα ορισμάτων εισόδου δεν είναι κατάλληλη, η εντολή κλήσης της συνάρτησης διακόπτεται και παράγεται μήνυμα λάθους. Η λίστα ορισμάτων εξόδου δύναται να είναι κενή, οπότε για την αποθήκευση του επιστρεφόμενου αποτελέσματος χρησιμοποιείται η εξ' ορισμού μεταβλητή *ans*.

Οι περισσότερες από τις συναρτήσεις που περιγράφονται στη συνέχεια επιστρέφουν αποτέλεσμα που εκχωρείται σε μια μεταβλητή, οπότε η εντολή κλήσης τους έχει τη μορφή

$$A = \text{function_name}(a_1, a_2, \dots, a_m)$$

ενώ ο τύπος των ορισμάτων και του αποτελέσματος είναι πραγματικό διάνυσμα με αναπαράσταση διπλής ακρίβειας. Στις περιπτώσεις όπου αυτό δεν συμβαίνει, αναφέρονται οι κατάλληλοι τύποι των ορισμάτων ή του αποτελέσματος.

Ας σημειωθεί τέλος ότι οι συναρτήσεις που περιγράφονται στο παρόν δίνονται υπό τη μορφή υπόδειξης: η επίλυση των εργαστηριακών ασκήσεων είναι ενδεχομένως δυνατή και με τη χρήση άλλων συναρτήσεων της MATLAB. Στους πίνακες που ακολουθούν δίνεται η λειτουργία και μια τυπική σύνταξη της εντολής κλήσης για κάθε μια από τις περιγραφόμενες συναρτήσεις.

Ενδεικτικές σχετικές συναρτήσεις

$C = \text{conv}(a,b)$	C = το γινόμενο των πολυωνύμων a και b σημείωση: ένα πολώνυμο ορίζεται στη <i>MATLAB</i> μέσω του διανύσματος με στοιχεία τους συντελεστές του σε φθίνουσα διάταξη, π.χ. το $a(s) = 2s^2 + 3s + 1$ ορίζεται ως το διάνυσμα $a = [2 \ 3 \ 1]$
------------------------	---

Παράδειγμα:

```
>> a=[1 2 3];
>> b=[2 6 7];
>> c=conv(a,b)
```

$c =$

```
2 10 25 32 21
```

$B = \text{roots}(a)$	B = οι ρίζες του πολυωνύμου a (εν γένει μιγαδικές) σημείωση: η δήλωση/αναπαράσταση μιγαδικού αριθμού στη <i>MATLAB</i> γίνεται ως $x = m+ni$, όπου m,n πραγματικοί
-----------------------	--

Παράδειγμα:

```
>> a=[1 4 5];
>> b=roots(a)
```

$b =$

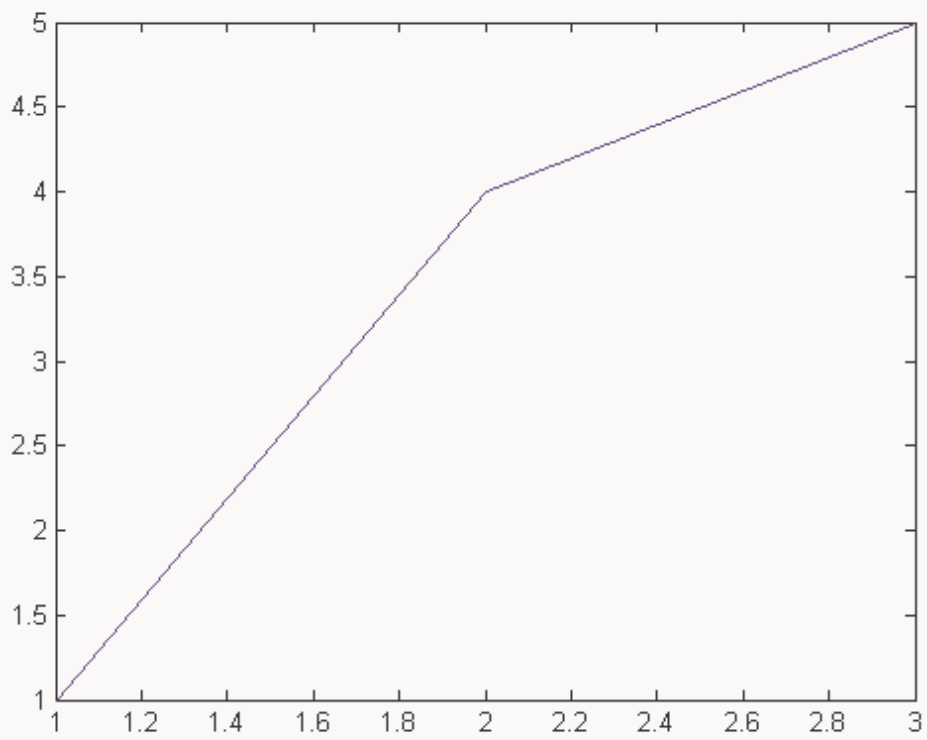
```
-2.0000 + 1.0000i
-2.0000 - 1.0000i
```

$\text{plot}(y)$	Σχεδιάζει τα στοιχεία του διανύσματος y συναρτήσει του δείκτη τους
$\text{plot}(x,y)$	Σχεδιάζει το διάνυσμα y συναρτήσει του διανύσματος x

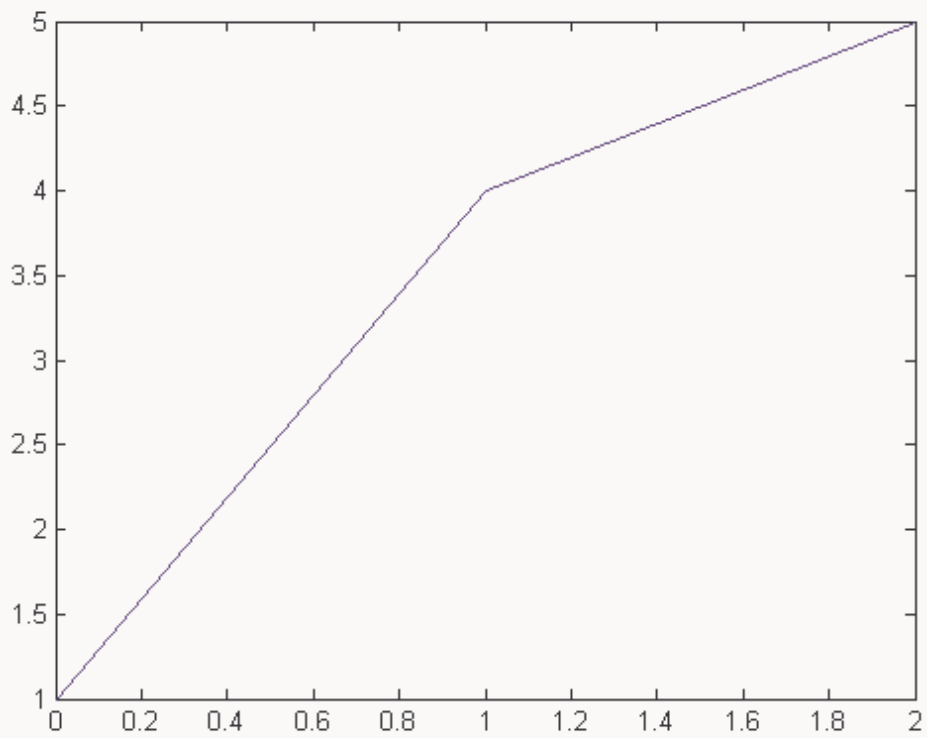
Παράδειγμα:

```
>> plot(a)
```

```
>>
```



```
> t=0:1:2;  
>> plot(t,a)  
>>
```



<code>max(x)</code>	Το μέγιστο στοιχείο του διανύσματος x
---------------------	---------------------------------------

Παράδειγμα:

```
>> a=[1 2 3];
>> max(a)

ans =

     3

>>
```

<code>length(x)</code>	Το μήκος του διανύσματος x
------------------------	----------------------------

Παράδειγμα:

```
>> length(a)

ans =

     3

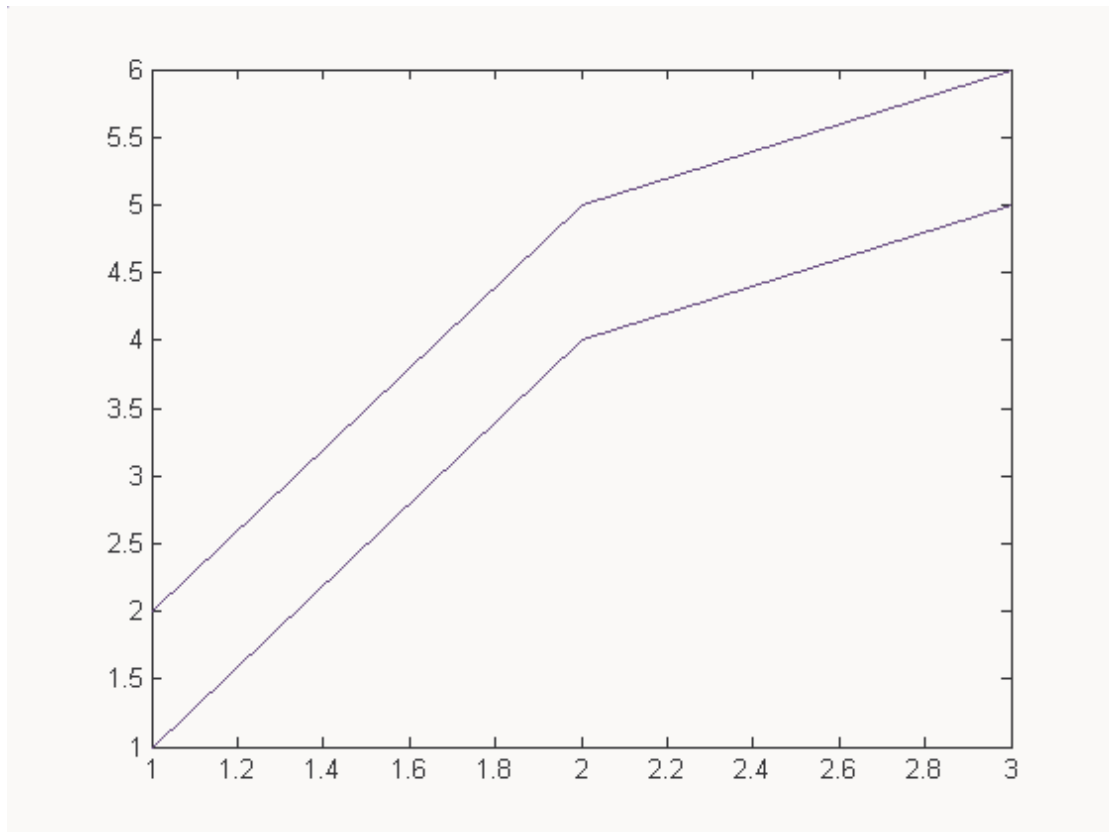
>>
```

<code>hold on</code>	Η επόμενη εντολή plot θα σχεδιάσει στο ήδη υπάρχον γράφημα
<code>hold off</code>	Αναίρεση της εντολής hold on (κάθε εντολή plot δημιουργεί νέο γράφημα)

Παράδειγμα:

```
>> a=[1 4 5];
>> plot(a)
>> hold on
>> b=[2 5 6];
>> plot(b)
>> hold off
```

και τα αντίστοιχα διαγράμματα παρουσιάζονται στο παρακάτω διάγραμμα :



<code>rand(m,n)</code>	Ένας πίνακας $m \times n$ του οποίου τα στοιχεία είναι τυχαίοι αριθμοί που ακολουθούν ομοιόμορφη κατανομή στο διάστημα $(0,1)$
------------------------	--

Παράδειγμα:

```
>> rand(2,3)
```

```
ans =
```

```
0.9501 0.6068 0.8913
0.2311 0.4860 0.7621
```

```
>>
```

<code>abs(s)</code>	Η απόλυτη τιμή των στοιχείων του διανύσματος s . Αν το s είναι μιγαδικό διάνυσμα, το μέτρο των στοιχείων του s .
---------------------	--

Παράδειγμα:

```
>> a=[-1 5 -9];
```

```
>> abs(a)
```

```
ans =
```

```
1 5 9
```

>>

angle(s)	Η φάση των στοιχείων του μιγαδικού διανύσματος s.
----------	---

Παράδειγμα:

```
>> b=[-1 5 -8];  
>> angle(b)
```

ans =

```
3.1416    0 3.1416
```

>>

$X = A \setminus b$ (τελεστής \)	Η λύση του γραμμικού συστήματος $AX=b$
----------------------------------	--

Παράδειγμα:

```
>> a=[5 6;7 8];  
>> b=[7 4;9 6];  
>> X=a\b
```

X =

```
-1 2  
2 -1
```

>>

4. Κατασκευή συναρτήσεων και αρχείων εντολών MATLAB

Εκτός από την εκτέλεση εντολών και την κλήση ενσωματωμένων συναρτήσεων μέσω του παραθύρου εντολών της, η MATLAB παρέχει τη δυνατότητα κλήσης συναρτήσεων και αρχείων εντολών που κατασκευάζονται από το χρήστη. Οι *συναρτήσεις* (functions) και τα *αρχεία εντολών* (scripts) είναι αρχεία κειμένου τα οποία περιέχουν κώδικα MATLAB και χαρακτηρίζονται ως *m-files*, καθώς τα ονόματά τους έχουν την κατάληξη *.m*.

- Τα αρχεία εντολών δεν δέχονται ορίσματα εισόδου και δεν επιστρέφουν ορίσματα εξόδου. Χρησιμοποιούν δεδομένα (μεταβλητές) του χώρου εργασίας.
- Οι συναρτήσεις δέχονται ορίσματα εισόδου και επιστρέφουν ορίσματα εξόδου (όπως και οι ενσωματωμένες συναρτήσεις της MATLAB). Οι μεταβλητές που ορίζονται εντός μιας συνάρτησης είναι *τοπικές* (γνωστές μόνο στη συνάρτηση και όχι στο χώρο εργασίας).

Κάθε m-file, είτε είναι συνάρτηση είτε αρχείο εντολών, δημιουργείται ως αρχείο κειμένου είτε μέσω ενός απλού επεξεργαστή κειμένου (π.χ. notepad) είτε μέσω του MATLAB editor. Για να δημιουργήσουμε ένα νέο m-file μέσω του MATLAB editor, επιλέγουμε το μενού *File* του παραθύρου εντολών και στη συνέχεια *New* → *M-file*. Όταν ολοκληρωθεί η δημιουργία του αρχείου, αυτό αποθηκεύεται με όνομα το οποίο έχει την κατάληξη *.m*. Για να είναι γνωστό ένα m-file στο περιβάλλον της MATLAB, θα πρέπει να αποθηκευτεί σε έναν από τους καταλόγους που περιλαμβάνονται στη διαδρομή αναζήτησης της (MATLAB search path). Η διαδρομή αναζήτησης της MATLAB εμφανίζεται αν δώσουμε την εντολή

» path

Κάθε φορά που εκκινείται η MATLAB, ορίζεται ως *τρέχων κατάλογος* (working directory) ένας από τους καταλόγους της διαδρομής αναζήτησης. Στην έκδοση 5.3 ο κατάλογος αυτός είναι τυπικά ο *x:\MATLABR11\work*, ενώ σε παλαιότερες εκδόσεις ο *x:\MATLAB5\bin*. Ο τρέχων κατάλογος εμφανίζεται εκτελώντας την εντολή

» pwd

Μια καλή πρακτική για τους νέους χρήστες της MATLAB είναι να αποθηκεύουν τα m-files που δημιουργούν στον εξ' ορισμού τρέχοντα κατάλογο.

Χρειάζεται κάποια προσοχή στην επιλογή των ονομάτων των m-files που δημιουργούνται από το χρήστη, καθώς αν υπάρχουν *δυο m-files με το ίδιο όνομα η MATLAB εκτελεί αυτό που συναντά πρώτο στη διαδρομή αναζήτησης* (ξεκινώντας από τον τρέχοντα κατάλογο). Τα επιλεγόμενα ονόματα δεν θα πρέπει να ταυτίζονται με ονόματα ενσωματωμένων συναρτήσεων της MATLAB.

Αρχεία εντολών

Όταν καλείται ένα αρχείο εντολών από το παράθυρο εντολών της MATLAB, εκτελούνται οι εντολές που περιέχονται στο αρχείο. Οι εντολές αυτές μπορεί να είναι

ορισμοί μεταβλητών, πράξεις, και κλήσεις ενσωματωμένων συναρτήσεων ή συναρτήσεων που έχουν κατασκευαστεί από το χρήστη. Η κλήση ενός αρχείου εντολών γίνεται δίνοντας στο παράθυρο εντολών το όνομα του *χωρίς την κατάληξη .m*. Για παράδειγμα, αν έχει δημιουργηθεί το αρχείο εντολών *my_file.m*, η κλήση του γίνεται με την εντολή

» `my_file`

Τα αρχεία εντολών μπορούν να χρησιμοποιούν υπάρχοντα δεδομένα (μεταβλητές) του χώρου εργασίας, ή μπορούν να δημιουργούν και να χρησιμοποιούν νέες μεταβλητές. Παρ' όλο που τα αρχεία εντολών δεν επιστρέφουν ορίσματα εξόδου όπως οι συναρτήσεις, οποιεσδήποτε μεταβλητές που δημιουργούνται από αυτά παραμένουν στο χώρο εργασίας και είναι έτσι διαθέσιμες για περαιτέρω χρήση.

Στη συνέχεια δίνεται ένα παράδειγμα αρχείου εντολών. Οι δύο πρώτες γραμμές είναι γραμμές σχολίων (μια γραμμή σχολίων δηλώνεται από το σύμβολο `%` στην αρχή της). Η ύπαρξη κενών γραμμών και γραμμών σχολίων επιτρέπεται σε οποιοδήποτε σημείο ενός αρχείου εντολών. Το αρχείο αυτό έχει αποθηκευτεί με το όνομα *my_script.m*. Μέσω αυτού του αρχείου ορίζεται αρχικά ο πίνακας A και στη συνέχεια δημιουργείται ο ανάστροφος του B.

```
% Script example
% Declaration and transposition of a 3x3 matrix

A = [1 2 3;4 5 6;7 8 9];
B = A'
```

Αν δώσουμε την εντολή

» `my_script`

έχουμε σαν αποτέλεσμα

```
B =

     1     4     7
     2     5     8
     3     6     9
```

Οι μεταβλητές A και B που ορίζονται μέσω του αρχείου εντολών είναι γνωστές στο χώρο εργασίας, όπως διαπιστώνουμε δίνοντας την εντολή

» `who`

Your variables are:

```
A    B
```

Οι γραμμές σχολίων που τυχόν περιέχονται στην αρχή ενός αρχείου εντολών, εμφανίζονται αν δώσουμε την εντολή *help όνομα_αρχείου*:

» `help my_script`

Script example
Declaration and transposition of a 3x3 matrix

Συναρτήσεις

Οι συναρτήσεις είναι m-files τα οποία περιέχουν εντολές MATLAB, όπως ακριβώς και τα αρχεία εντολών, αλλά έχουν δυο βασικές διαφορές με αυτά: αφ' ενός δέχονται ορίσματα εισόδου και επιστρέφουν ορίσματα εξόδου και αφ' ετέρου οι μεταβλητές που ορίζονται στο εσωτερικό τους δεν είναι γνωστές στο χώρο εργασίας. Η κλήση μιας συνάρτησης που έχει δημιουργηθεί από το χρήστη γίνεται με τον ίδιο τρόπο που γίνεται η κλήση μιας ενσωματωμένης συνάρτησης της MATLAB (δείτε σχετικά την ενότητα 3 του παρόντος). Το όνομα της συνάρτησης πρέπει να είναι το ίδιο με το όνομα του m-file στο οποίο είναι αποθηκευμένη.

Ένα m-file το οποίο ορίζει μια συνάρτηση περιέχει υποχρεωτικά στην πρώτη γραμμή του την εντολή (στη γενική της μορφή)

$$\text{function } [A_1, A_2, \dots, A_n] = \text{function_name}(a_1, a_2, \dots, a_m)$$

όπου $[A_1, A_2, \dots, A_n]$ είναι η λίστα των ορισμάτων εξόδου, *function_name* το όνομα της συνάρτησης και (a_1, a_2, \dots, a_m) η λίστα των ορισμάτων εισόδου. Στη συνέχεια δίνεται ο κώδικας της συνάρτησης *my_function.m*, η οποία δέχεται ως όρισμα εισόδου έναν πίνακα και επιστρέφει ως αποτέλεσμα (όρισμα εξόδου) τον ανάστροφο του. Προφανώς η συνάρτηση αυτή κάνει ότι και το αρχείο εντολών *my_script.m*, αλλά για οποιονδήποτε πίνακα της μεταβιβάσουμε ως όρισμα εισόδου. Έχει συνεπώς γενικότερη εφαρμογή.

```
function b = my_function(a)
% Function example
% Transposition of a matrix
```

```
b = a';
```

Για να εξετάσουμε τον τρόπο λειτουργίας της συνάρτησης, δημιουργούμε στο χώρο εργασίας τον πίνακα A:

```
» A = [1 2 3;4 5 6;7 8 9]
```

```
A =
```

```
1 2 3
4 5 6
7 8 9
```

Στη συνέχεια καλούμε τη συνάρτηση ως εξής:

```
» B = my_function(A)
```

```
B =
```

```
1 4 7
2 5 8
```


Ο πίνακας A μεταβιβάζεται ως όρισμα εισόδου, ενώ στη μεταβλητή B (όρισμα εξόδου) αποθηκεύεται το αποτέλεσμα της κλήσης της συνάρτησης. Οι μεταβλητές a και b που ορίζονται και χρησιμοποιούνται από την συνάρτηση δεν είναι γνωστές στο χώρο εργασίας, όπως μπορούμε να διαπιστώσουμε δίνοντας την εντολή

» who

Your variables are:

A B

Οι γραμμές σχολίων που ακολουθούν την πρώτη γραμμή του m-file που ορίζει μια συνάρτηση εμφανίζονται όταν δώσουμε την εντολή *help όνομα_συνάρτησης*, όπως ακριβώς συμβαίνει και με τις ενσωματωμένες συναρτήσεις της MATLAB:

» help my_function

Function example
Transposition of a matrix

Καθολικές μεταβλητές

Στην περίπτωση όπου είναι αναγκαίο μια μεταβλητή να είναι γνωστή ταυτόχρονα σε περισσότερες από μια συναρτήσεις, αυτή δηλώνεται ως *καθολική* (global) σε κάθε μια από αυτές. Εάν είναι επιθυμητό η μεταβλητή αυτή να είναι γνωστή και στο χώρο εργασίας, ή σε αρχείο εντολών το οποίο καλεί τις εν λόγω συναρτήσεις, θα πρέπει να δηλωθεί και εκεί ως καθολική. Η δήλωση μιας μεταβλητής ως καθολικής πρέπει να προηγείται της χρήσης της και γίνεται χρησιμοποιώντας τη δεσμευμένη λέξη *global*, όπως φαίνεται και στο παράδειγμα που ακολουθεί:

```
global G
G = 9.81;
```

Είθισται οι καθολικές μεταβλητές να έχουν ονόματα με κεφαλαία γράμματα ώστε να διακρίνονται από τις υπόλοιπες μεταβλητές, χωρίς αυτό να είναι δεσμευτικό.

Έλεγχος ροής – λογικοί και συσχετιστικοί τελεστές

Η MATLAB διαθέτει πέντε δομές ελέγχου ροής:

- εντολές if
- εντολές switch
- βρόχους for
- βρόχους while
- εντολές break

Εντολές if

Το γενικό συντακτικό μιας δομής ελέγχου *if* έχει τη μορφή:

```

if λογική_συνθήκη_1
    μπλοκ_εντολών_1
elseif λογική_συνθήκη_2
    μπλοκ_εντολών_2
.....
elseif λογική_συνθήκη_n-1
    μπλοκ_εντολών_n-1
else
    μπλοκ_εντολών_n
end

```

Η δομή αυτή εξετάζει την ισχύ των λογικών συνθηκών που συνοδεύουν τις δεσμευμένες λέξεις *if* και *elseif*. Η πρώτη από τις λογικές συνθήκες που υπολογίζεται ως αληθής (true) συνεπάγεται την εκτέλεση του αντίστοιχου μπλοκ εντολών και την έξοδο από τη δομή *if*. Εάν καμία λογική συνθήκη δεν είναι αληθής, εκτελείται το μπλοκ εντολών που ακολουθεί το *else*. Οι δομές *elseif* και *else* είναι προαιρετικές, εν αντιθέσει με την ύπαρξη του *end* που σηματοδοτεί την ολοκλήρωση της δομής ελέγχου *if*.

Εντολές switch

Το γενικό συντακτικό μιας δομής ελέγχου *switch* έχει τη μορφή:

```

switch μαθηματική_έκφραση
    case τιμή_1
        μπλοκ_εντολών_1
.....
    case τιμή_n
        μπλοκ_εντολών_n
    otherwise
        μπλοκ_εντολών_n+1
end

```

Η δομή ελέγχου *switch* έχει ως αποτέλεσμα την εκτέλεση μπλοκ εντολών ανάλογα με την τιμή της μαθηματικής έκφρασης. Συγκεκριμένα, εκτελείται μόνο το μπλοκ εντολών του πρώτου *case* για το οποίο η τιμή συμπίπτει με την τιμή της έκφρασης. Αν αυτό δεν συμβαίνει για κανένα *case*, εκτελείται το μπλοκ εντολών που ακολουθεί το *otherwise*.

Βρόχοι for

Ένας βρόχος *for* εκτελεί ένα μπλοκ εντολών για ένα καθορισμένο αριθμό επαναλήψεων. Το γενικό συντακτικό μιας δομής ελέγχου *for* έχει τη μορφή:

```

for μετρητής = n1:step:n2
    μπλοκ_εντολών
end

```

Ως *μετρητής* χρησιμοποιείται μια μεταβλητή η οποία παίρνει τιμές από *n1* ως *n2* με βήμα *step*. Οι τιμές των *n1*, *n2* και *step* καθορίζουν και το πλήθος των εκτελούμενων επαναλήψεων. Αν η *step* παραληφθεί, θεωρείται ότι έχει την τιμή 1 (σύνθητες). Συνήθως η μεταβλητή *μετρητής* παίρνει ακέραιες τιμές και χρησιμοποιείται εντός του

μπλοκ εντολών (π.χ. ως δείκτης γραμμών/στηλών κάποιου πίνακα). Είναι δυνατή η ύπαρξη φωλιασμένων (nested) βρόχων *for*:

```
for μετρητής_α = n1_α:step_α:n2_α
    for μετρητής_β = n1_β:step_β:n2_β
        μπλοκ εντολών
    end
end
```

Βρόχοι *while*

Ένας βρόχος *while* επαναλαμβάνει το *μπλοκ_εντολών* για ένα ακαθόριστο αριθμό επαναλήψεων εφ' όσον η *λογική_συνθήκη* είναι αληθής. Το γενικό συντακτικό ενός βρόχου *while* έχει τη μορφή:

```
while λογική_συνθήκη
    μπλοκ_εντολών
end
```

Εντός του *μπλοκ εντολών* θα πρέπει να υπάρχει απαραίτητα μια εντολή η οποία επιδρά στην τιμή της λογικής συνθήκης έτσι ώστε αυτή κάποτε να γίνει ψευδής, διαφορετικά ο βρόχος θα γίνει ατέρμονας. Η λογική συνθήκη θα πρέπει να έχει αληθή τιμή πριν την είσοδο του προγράμματος στο βρόχο *while*.

Εντολές *break*

Η εντολή *break* προκαλεί τον πρόωρο τερματισμό ενός βρόχου *for* ή *while* και την έξοδο της ροής του προγράμματος από αυτόν. Σε φωλιασμένους βρόχους, η *break* προκαλεί την έξοδο μόνο από τον πλέον εσωτερικό βρόχο. Συνήθως η εντολή *break* περιλαμβάνεται στο *μπλοκ εντολών* κάποιας δομής ελέγχου *if* εντός βρόχου ανακύκλωσης, ώστε αυτός να τερματιστεί με την ικανοποίηση κάποιας λογικής συνθήκης.

Λογικοί και συσχετιστικοί τελεστές

Οι λογικές συνθήκες που χρησιμοποιούνται στις δομές ελέγχου *if* και *while* παίρνουν μια από τις τιμές *αληθής* (true) ή *ψευδής* (false) και κατασκευάζονται με τη βοήθεια των *λογικών* (logical) και των *συσχετιστικών* (relational) τελεστών. Οι κυριότεροι λογικοί τελεστές της MATLAB είναι οι ακόλουθοι:

Λογικό AND: &
Λογικό OR: /
Λογικό NOT: ~

Οι κυριότεροι συσχετιστικοί τελεστές της MATLAB είναι οι ακόλουθοι:

- Ισότητας: ==
- Ανισότητας: ~=
- Μικρότερο: <
- Μεγαλύτερο: >
- Μικρότερο ίσο: <=
- Μεγαλύτερο ίσο: >=

Οι συσχετιστικοί τελεστές μπορούν να δεχτούν ως τελεσταίους και διανύσματα ή πίνακες εκτός από βαθμωτές μεταβλητές ή παραστάσεις, αλλά κάτι τέτοιο χρειάζεται προσοχή καθώς η λειτουργία τους είναι διαφορετική στην περίπτωση αυτή. Περισσότερες πληροφορίες για τους λογικούς και συσχετιστικούς τελεστές είναι διαθέσιμες μέσω της βοήθειας της MATLAB (εντολή *help relop*).

Σχετικές με τις ασκήσεις εντολές:

1^η σειρά ασκήσεων

$\text{Sys} = \text{tf}(\text{num}, \text{den})$	Sys = μεταβλητή-αντικείμενο τύπου συνάρτησης μεταφοράς με αριθμητή και παρανομαστή τα πολυώνυμα num και den , αντίστοιχα
$\text{step}(\text{sys})$	Υπολογίζεται και σχεδιάζεται η μοναδιαία βηματική απόκριση του συστήματος με συνάρτηση μεταφοράς sys (sys =μεταβλητή-αντικείμενο τύπου συνάρτησης μεταφοράς)
$\text{impulse}(\text{sys})$	Υπολογίζεται και σχεδιάζεται η μοναδιαία κρουστική απόκριση του συστήματος με συνάρτηση μεταφοράς sys (sys =μεταβλητή-αντικείμενο τύπου συνάρτησης μεταφοράς)
$Y = \text{lsim}(\text{sys}, u, t)$	Y = η απόκριση του συστήματος sys στην είσοδο u κατά το χρονικό διάστημα t <i>υπόδειξη: ορισμός t ως $t1:dt:t2$, ενώ το u είναι το διάνυσμα τιμών της συνάρτησης εισόδου $u = f(t)$ που υπολογίζεται για το διάνυσμα t</i>
$Y = \text{polyval}(p, x)$	Y = η τιμή του πολυωνύμου $p(r)$ όταν $r = x$ (το x μπορεί να είναι και μιγαδικός)
$\text{Sys} = \text{series}(\text{sys1}, \text{sys2})$	Sys = μεταβλητή-αντικείμενο τύπου συνάρτησης μεταφοράς που προκύπτει από τη σύνδεση σε σειρά των συστημάτων με συναρτήσεις μεταφοράς sys1 και sys2
$\text{Sys} = \text{parallel}(\text{sys1}, \text{sys2})$	Sys = μεταβλητή-αντικείμενο τύπου συνάρτησης μεταφοράς που προκύπτει από την παράλληλη σύνδεση των συστημάτων με συναρτήσεις μεταφοράς sys1 και sys2
$\text{Sys} = \text{feedback}(\text{sys1}, \text{sys2})$	Sys = μεταβλητή-αντικείμενο τύπου συνάρτησης μεταφοράς που προκύπτει από τη σύνδεση των συστημάτων sys1 και sys2 σε σύστημα κλειστού βρόχου, όπου sys1 : συνάρτηση εγκατάστασης και sys2 : συνάρτηση ανατροφοδότησης (αρνητική ανατροφοδότηση)

3^η σειρά ασκήσεων

rlocus(sys)	Υπολογισμός και σχεδίαση του γ.τ.ρ. της εξίσωσης $1+KG(s) = 0$, όπου $G(s)$: η συνάρτηση μεταφοράς συστήματος που αναπαρίσταται με τη μεταβλητή-αντικείμενο τύπου συνάρτησης μεταφοράς sys
-------------	--

4^η σειρά ασκήσεων

<i>Συνάρτηση</i>	<i>Επιστρεφόμενο αποτέλεσμα</i>
[k,poles] = rlocfind(sys)	Εύρεση της τιμής του K και των αντίστοιχων ριζών της χαρακτηριστικής εξίσωσης $1+KG(s) = 0$ από το γ.τ.ρ. με γραφικό τρόπο (προϋποθέτει τη σχεδίαση του γ.τ.ρ. μέσω της εντολής rlocus(sys))
[num,den] = ord2(wn,z)	Δημιουργία συνάρτησης μεταφοράς (πολυνόμια αριθμητή-παρονομαστή: num και den, αντίστοιχα) πρότυπου συστήματος 2 ^{ης} τάξης με επιλεγμένα ζ και ω _n
[z,p,k] = tf2zp(num,den)	Εύρεση των μηδενικών και πόλων (z και p αντίστοιχα) της συνάρτησης μεταφοράς με πολυνόμια αριθμητή-παρονομαστή num και den, αντίστοιχα

Σημείωση: οι συναρτήσεις που υποδεικνύονται για την 4^η σειρά ασκήσεων (πλην της rlocfind) μπορούν να χρησιμοποιηθούν για τον υπολογισμό των παραμέτρων ενός PID ή PD ελεγκτή σύμφωνα με τη μεθοδολογία που παρουσιάζεται στην παράγραφο 5.1.3.5 των σημειώσεων.

B. Περιγραφή και βασικές λειτουργίες του SIMULINK (εκδόσεις 2.x και άνω)

1. Γενικά

Το SIMULINK είναι ένα λογισμικό πακέτο που επιτρέπει τη μοντελοποίηση, προσομοίωση και ανάλυση δυναμικών συστημάτων. Υποστηρίζει γραμμικά και μη γραμμικά συστήματα, μοντελοποιημένα σε συνεχή ή διακριτό χρόνο, ή ακόμη και υβριδικά συστήματα (εν μέρει μοντελοποιημένα σε συνεχή και εν μέρει σε διακριτό χρόνο). Υποστηρίζονται ακόμη συστήματα με τμηματικά διαφορετικούς χρόνους δειγματοληψίας.

Για τη μοντελοποίηση, το SIMULINK παρέχει ένα γραφικό περιβάλλον διεπαφής (GUI) που επιτρέπει την κατασκευή μοντέλων ως δομικών διαγραμμάτων, χρησιμοποιώντας λειτουργίες click-and-drag του ποντικιού. Το SIMULINK περιλαμβάνει ένα πλήθος βιβλιοθηκών δομικών στοιχείων (blocks), τα βασικότερα από τα οποία είναι οι πηγές (sources), τα στοιχεία «απορρόφησης» (sinks), τα συνεχή γραμμικά στοιχεία, τα μη γραμμικά στοιχεία και τα στοιχεία σημάτων και συστημάτων. Είναι επίσης δυνατή η τροποποίηση και η δημιουργία νέων δομικών στοιχείων από το χρήστη.




Τα μοντέλα SIMULINK είναι *ιεραρχικά* (ένα μοντέλο μπορεί να περιέχει μπλοκ τα οποία περιέχουν με τη σειρά τους άλλα μπλοκ), έτσι μπορούν να ιδωθούν σε διάφορα επίπεδα. Ένα σύστημα που έχει ιεραρχική δομή μπορεί να ιδωθεί αρχικά σε υψηλό επίπεδο ως ένα σύνολο διασυνδεδεμένων υποσυστημάτων, κάθε ένα από τα οποία μοντελοποιείται ως ένα μπλοκ. Στη συνέχεια, κάνοντας διπλό κλικ με το ποντίκι στα επί μέρους μπλοκ, ο χρήστης μπορεί να κατέβει σε χαμηλότερα επίπεδα ώστε να δει αυξανόμενους βαθμούς λεπτομέρειας.

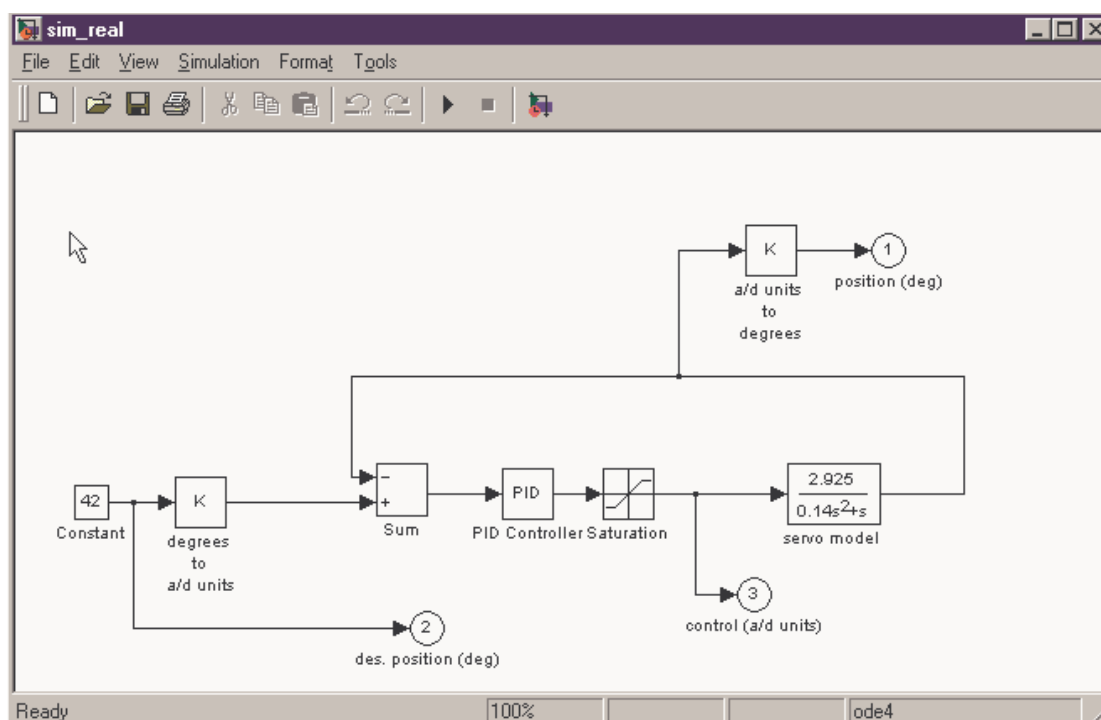
Μετά τη δημιουργία ενός μοντέλου, είναι δυνατή η προσομοίωση του, χρησιμοποιώντας μια από τις διάφορες μεθόδους ολοκλήρωσης που παρέχει το SIMULINK. Χρησιμοποιώντας παλμογράφους (scopes) και άλλα μπλοκ απεικόνισης, είναι δυνατή η παρακολούθηση των αποτελεσμάτων της προσομοίωσης καθώς αυτή εξελίσσεται. Επιπλέον, είναι δυνατή η εξαγωγή αποτελεσμάτων της προσομοίωσης στο χώρο εργασίας της MATLAB για περαιτέρω επεξεργασία. Είναι ακόμη δυνατή η χρήση του SIMULINK για προσομοίωση αλλά και έλεγχο συστημάτων σε πραγματικό χρόνο, μέσω της εργαλειοθήκης πραγματικού χρόνου (Real Time Workshop).

Στη συνέχεια θα δοθούν κάποιες βασικές οδηγίες χρήσης με έμφαση στην προσομοίωση έτοιμων μοντέλων SIMULINK και θα γίνει μια συνοπτική περιγραφή των βιβλιοθηκών και ορισμένων δομικών στοιχείων τα οποία είναι πιθανό να χρησιμοποιηθούν στα πλαίσια των εργαστηριακών ασκήσεων του μαθήματος ΣΑΕ Ι. Περαιτέρω πληροφορίες για τα δομικά στοιχεία και τις βιβλιοθήκες του SIMULINK είναι διαθέσιμες μέσω της βοήθειας της MATLAB.

Σημείωση: η οργάνωση των βιβλιοθηκών και το γραφικό περιβάλλον παρουσιάζουν ορισμένες διαφορές μεταξύ των εκδόσεων 2.x και 3.x του SIMULINK, παρ' όλα αυτά οι βασικές λειτουργίες παραμένουν οι ίδιες. Οι πληροφορίες που δίνονται στη συνέχεια αφορούν την έκδοση 3.

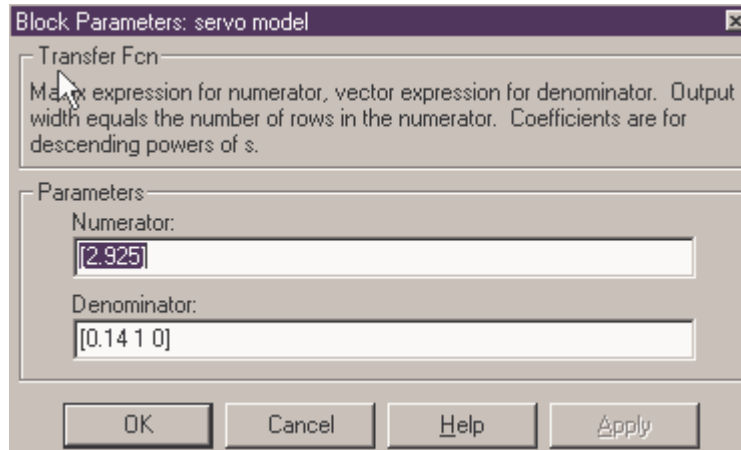
2. Οι βασικές λειτουργίες του SIMULINK

Τα μοντέλα δυναμικών συστημάτων που κατασκευάζονται με το SIMULINK αποθηκεύονται ως αρχεία με την κατάληξη *.mdl*. Προκειμένου να δημιουργήσουμε ένα νέο μοντέλο ή να ανοίξουμε ένα αποθηκευμένο μοντέλο, στο παράθυρο εντολών της MATLAB κάνουμε αρχικά κλικ στο εικονίδιο  το οποίο ανοίγει τον browser των βιβλιοθηκών του SIMULINK. Από το παράθυρο του browser μπορούμε να δημιουργήσουμε ένα νέο μοντέλο, κάνοντας κλικ στο εικονίδιο  ή να ανοίξουμε ένα αποθηκευμένο μοντέλο, κάνοντας κλικ στο εικονίδιο . Ένα τυπικό μοντέλο γραμμικού χρονικά αναλοίωτου συστήματος αυτομάτου ελέγχου μιας εισόδου μιας εξόδου με PID ελεγκτή παρουσιάζεται στο σχήμα που ακολουθεί.

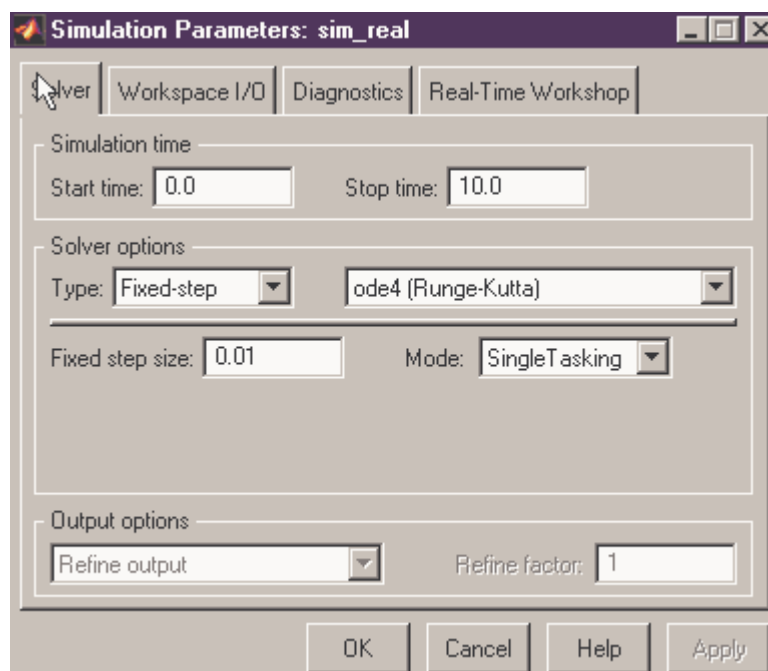


Τα βασικά δομικά στοιχεία του μοντέλου είναι: α) το μοντέλο συνάρτησης μεταφοράς του ελεγχόμενου συστήματος (μπλοκ “servo model”), β) το μοντέλο του ελεγκτή PID (μπλοκ “PID Controller”) γ) η είσοδος του συστήματος, σταθερά στην περίπτωση αυτή (μπλοκ “Constant”, ανήκει στην κατηγορία των source blocks) και δ) ο αθροιστής (μπλοκ “Sum”) ο οποίος κατασκευάζει το σήμα σφάλματος. Τα μπλοκ που περιέχουν το “K” πραγματοποιούν πολλαπλασιασμό με μια σταθερά (μπλοκ κέρδους, εδώ χρησιμοποιούνται για μετατροπή μονάδων). Τα αριθμημένα κυκλικά μπλοκ (ανήκουν στην κατηγορία των μπλοκ σημάτων και συστημάτων) αποτελούν τις θύρες εξόδου του μοντέλου (παρακολουθούμενα σήματα) προς το χώρο εργασίας της MATLAB, όπως θα δούμε αργότερα. Το μπλοκ με το όνομα “Saturation” είναι ένα μη γραμμικό στοιχείο κόρου.

Κάθε ένα από τα μπλοκ χαρακτηρίζεται από ορισμένες παραμέτρους. Οι τιμές των παραμέτρων αυτών μπορούν να καθοριστούν κάνοντας διπλό κλικ πάνω στο μπλοκ, οπότε ανοίγει ένα παράθυρο διαλόγου, όπως αυτό που φαίνεται στο σχήμα που ακολουθεί για το μπλοκ μοντέλου συνάρτησης μεταφοράς συστήματος μιας εισόδου μιας εξόδου:

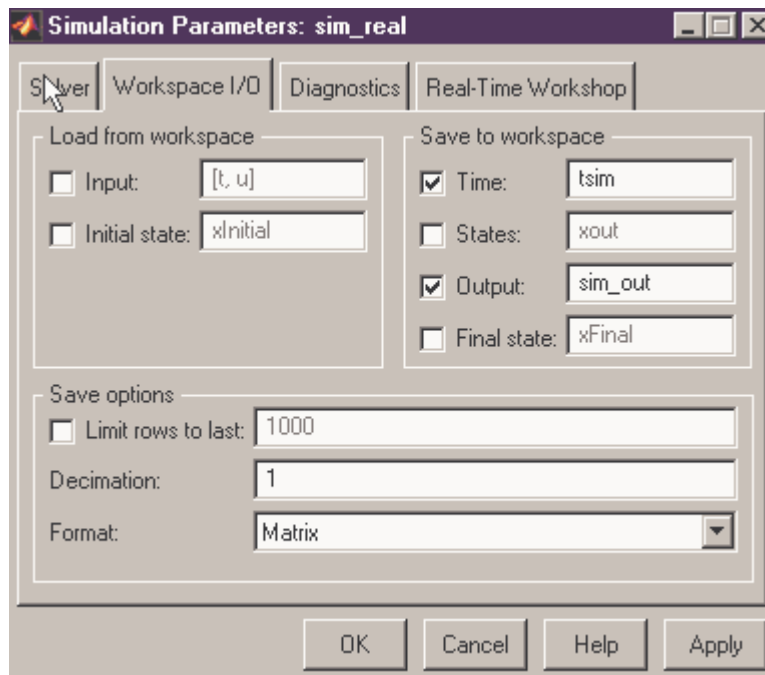


Μετά τον καθορισμό όλων των απαιτούμενων παραμέτρων για τα μπλοκ που περιλαμβάνονται στο μοντέλο, πρέπει να καθοριστούν οι παράμετροι της προσομοίωσης. Αυτό είναι δυνατό μέσω του μενού “Simulation” του παραθύρου του μοντέλου. Επιλέγοντας το μενού “Parameters...” εμφανίζεται το παράθυρο που φαίνεται στο σχήμα που ακολουθεί:





Μέσω της καρτέλας “Solver” καθορίζονται ο χρόνος έναρξης και λήξης της προσομοίωσης (σε δευτερόλεπτα), η μέθοδος ολοκλήρωσης που θα χρησιμοποιηθεί και ανάλογα με αυτήν οι απαιτούμενες παράμετροι (σημείωση: τα συστήματα που μοντελοποιούνται στα πλαίσια του μαθήματος ΣΑΕ Ι είναι όλα συνεχούς χρόνου, επομένως και η επιλεγόμενη μέθοδος ολοκλήρωσης πρέπει να είναι συνεχούς χρόνου). Μέσω της καρτέλας “Workspace I/O”, που παρουσιάζεται στο σχήμα που ακολουθεί, είναι δυνατή η επικοινωνία με το χώρο εργασίας της MATLAB. Πιο αναλυτικά:

- Μέσω των επιλογών “Load from workspace” καθορίζεται αν το μοντέλο θα δεχτεί είσοδο ή κάποια αρχική κατάσταση μέσω μεταβλητών του χώρου εργασίας (δεν έχει επιλεγεί κάτι τέτοιο στο συγκεκριμένο μοντέλο).





- Μέσω των επιλογών “Save to workspace” καθορίζονται οι εξόδοι προς το χώρο εργασίας. Στο συγκεκριμένο μοντέλο έχει επιλεγεί η αποθήκευση στο χώρο εργασίας του χρόνου προσομοίωσης (διάνυσμα με τις χρονικές στιγμές της ολοκλήρωσης) ως μεταβλητή *tsim* και των εξόδων του μοντέλου ως μεταβλητή *sim_out* (οι τιμές των τριων σημάτων που οδηγούνται στα sink blocks κατά τις χρονικές στιγμές της ολοκλήρωσης).
- Μέσω των επιλογών “Save options” καθορίζονται ορισμένες επιλογές για την αποθήκευση μεταβλητών στο χώρο εργασίας. Η επιλογή του “Limit rows to last:” έχει ως αποτέλεσμα την αποθήκευση μόνο του αριθμού των τελευταίων βημάτων της ολοκλήρωσης που εμφανίζεται στο παράθυρο (εξ’ ορισμού 1000).

Σημείωση: αν δεν καθοριστούν παράμετροι προσομοίωσης, το SIMULINK χρησιμοποιεί τις εξ’ ορισμού παραμέτρους (αυτό συμβαίνει γενικά με όλα τα μπλοκ που περιλαμβάνουν παραμέτρους).

Μετά και τον καθορισμό των παραμέτρων της προσομοίωσης είναι δυνατή η εκτέλεση της. Αυτό μπορεί να γίνει είτε από το μενού “Simulation”, όπου επιλέγουμε “Start”, είτε κάνοντας κλικ στο εικονίδιο . Η προσομοίωση σταματάει μόλις ο χρόνος φτάσει την τιμή που είχε τεθεί στην παράμετρο “Stop time”. Μπορούμε να διακόψουμε την προσομοίωση αν επιλέξουμε “Stop” από το μενού “Simulation” ή κάνουμε κλικ στο εικονίδιο  (όταν δεν τρέχει η προσομοίωση είναι απενεργοποιημένο).

Προκειμένου να δημιουργήσουμε ένα νέο μοντέλο SIMULINK εξ’ αρχής, η διαδικασία είναι η ακόλουθη:

- κάνουμε κλικ στο εικονίδιο  του browser βιβλιοθηκών, οπότε ανοίγει ένα παράθυρο νέου μοντέλου (untitled)

- εισάγουμε τα επιθυμητά μπλοκ από τις κατάλληλες βιβλιοθήκες κάνοντας click-and-drag με το ποντίκι απ' ευθείας στο παράθυρο του νέου μοντέλου (τα περιεχόμενα μιας βιβλιοθήκης εμφανίζονται κάνοντας διπλό κλικ πάνω στο εικονίδιο  του browser).
- Κάνουμε τις κατάλληλες διασυνδέσεις με click-and-drag από την έξοδο ενός μπλοκ στην είσοδο του άλλου
- Καθορίζουμε τις παραμέτρους των μπλοκ και της προσομοίωσης
- Αποθηκεύουμε το μοντέλο μέσω του μενού "File"

Μπορούμε να διαγράψουμε, να αντιγράψουμε, ή να κόψουμε και να επικολλήσουμε μπλοκ ή συνδέσεις -αφού πρώτα τα επιλέξουμε κάνοντας κλικ πάνω τους- με το συνήθη τρόπο. Μπορούμε επίσης να επιλέξουμε μια ομάδα μπλοκ και συνδέσεων κάνοντας click-and-drag με το ποντίκι και περιλαμβάνοντας τα.

3. Οι βασικές βιβλιοθήκες δομικών στοιχείων του SIMULINK

Στον πίνακα που ακολουθεί περιγράφονται ορισμένα από τα δομικά στοιχεία των βιβλιοθηκών του SIMULINK, οι οποίες χρησιμοποιούνται συνήθως κατά τη μοντελοποίηση γραμμικών χρονικά αναλλοίωτων συστημάτων συνεχούς χρόνου.

Βιβλιοθήκη	Δομικό στοιχείο	Λειτουργία
Sources - περιέχει μπλοκ τα οποία είναι πηγές σημάτων (δεν έχουν είσοδο, παράγουν ως έξοδο ένα σήμα)	Constant	Σταθερά
	Step	Βηματική συνάρτηση
	Ramp	Συνάρτηση αναρρίχησης
	Pulse generator	Γεννήτρια παλμών
	Random number	Γεννήτρια τυχαίου σήματος (κανονική κατανομή)
	Sine wave	Γεννήτρια ημιτόνου
	Signal generator	Γεννήτρια σημάτων (παράγει διάφορες κυματομορφές)
Sinks - περιέχει μπλοκ τα οποία είναι στοιχεία «απορρόφησης» σημάτων (δεν έχουν έξοδο, δέχονται μόνο είσοδο)	Display	Οθόνη απεικόνισης τιμών
	Scope	Παλμογράφος
	Stop Simulation	Τερματισμός προσομοίωσης
	To Workspace	Αποθήκευση στο χώρο εργασίας
Continuous - περιέχει μπλοκ για τη μοντελοποίηση γραμμικών συστημάτων συνεχούς χρόνου	Derivative	Παραγωγή
	Integrator	Ολοκλήρωση
	State space	Μοντέλο συστήματος στο χώρο κατάστασης
	Transfer function	Μοντέλο συστήματος συνάρτησης μεταφοράς
	Zero-pole	Μοντέλο συστήματος πόλων – μηδενικών
Nonlinear - περιέχει μπλοκ που μοντελοποιούν μη γραμμικά στοιχεία	Saturation	Στοιχείο κόρου
	Manual Switch	Χειροκίνητος διακόπτης
	Switch	Διακόπτης
	Quantizer	Κβαντιστής σήματος

Βιβλιοθήκη	Δομικό στοιχείο	Λειτουργία
Math - Περιέχει μπλοκ που μοντελοποιούν μαθηματικές πράξεις και συναρτήσεις	Abs	Απόλυτη τιμή
	Gain	Κέρδος
	Math function	Διάφορες μαθηματικές συναρτήσεις
	Matrix Gain	Πίνακας κερδών
	MinMax	Ελάχιστο ή μέγιστο
	Product	Πολλαπλασιασμός ή διαίρεση
	Rounding Function	Συνάρτηση στρογγύλευσης
	Sign	Εύρεση προσήμου
	Slider gain	Μεταβλητό κέρδος
	Sum	Άθροιση ή αφαίρεση
Trigonometric function	Τριγωνομετρικές συναρτήσεις	
Signals and Systems - Περιέχει στοιχεία διασύνδεσης σημάτων και συστημάτων	In1	Θύρα εισόδου υποσυστήματος ή μοντέλου
	Demux	Αποπλέκτης σημάτων
	Mux	Πολυπλέκτης σημάτων
	Terminator	Τερματισμός ασύνδετων σημάτων (δέχεται μόνο είσοδο)
	Out1	Θύρα εξόδου υποσυστήματος ή μοντέλου
Control Systems Toolbox - Περιέχει στοιχεία μοντελοποίησης συστημάτων ελέγχου	LTI System	Μοντελοποίηση γραμμικού χρονικά αναλλοίωτου συστήματος με διάφορους τρόπους (μέσω συνάρτησης μεταφοράς, στο χώρο κατάστασης, αναπαράσταση πόλων – μηδενικών)

Υπενθυμίζεται ότι η οργάνωση των βιβλιοθηκών του SIMULINK όπως παρουσιάστηκε στον προηγούμενο πίνακα αναφέρεται στην έκδοση 3.