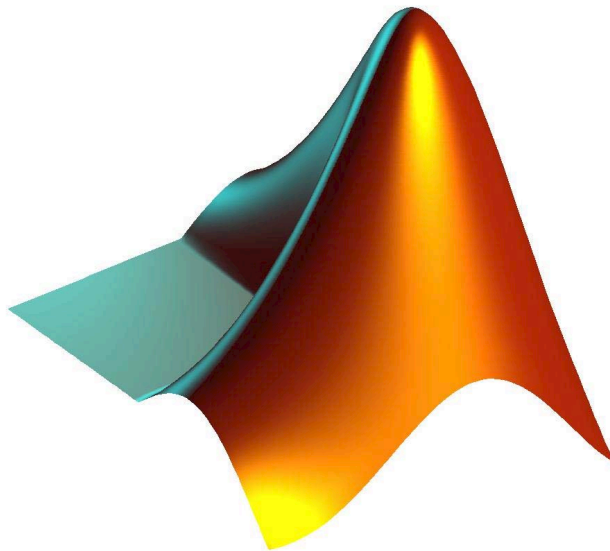


**Εθνικό Μετσόβιο Πολυτεχνείο
Σχολή Μηχανολόγων Μηχανικών**



Εγχειρίδιο MATLAB SIMULINK



**2.3.08.6 Εισαγωγή στη Θεωρία και Τεχνολογία
του Αυτομάτου Ελέγχου**

Καθηγητής Ευάγγελος Παπαδόπουλος

Εαρινό εξάμηνο 2012

Περιεχόμενα

1. Πρόλογος	3
2. Εκκίνηση MATLAB	3
3. Η Εντολή HELP	3
4. Διαχείριση Workspace	6
5. Πίνακες στο MATLAB.....	7
6. Πράξεις Πινάκων	10
7. Μαθηματικές πράξεις με διανύσματα	11
8. Γραφικές Παραστάσεις	13
9. Εντολές Workspace	18
10. Πολυωνυμικές Πράξεις.....	19
11. Λογικοί Τελεστές.....	20
12. Έλεγχος Ροής	22
13. Συναρτήσεις .m	24
14. Εφαρμογές Matlab στον Αυτόματο Έλεγχο	26
15. SIMULINK	36
Παράρτημα Α.....	57

1. Πρόλογος

Το MATLAB, έχει ονομαστεί από τις λέξεις MATrix LABoratory, και αποτελεί ένα λογισμικό περιβάλλον προγραμματισμού αριθμητικών μεθόδων, προσομοίωσης και γραφικής οπτικοποίησης. Το βασικό στοιχείο/ μεταβλητή του MATLAB είναι ο πίνακας και έτσι όλοι οι φιλόδοξοι μελλοντικοί προγραμματιστές πρέπει να γνωρίζουν τις βασικές μαθηματικές αρχές πινάκων και γραμμικής άλγεβρας.

Το εγχειρίδιο αποσκοπεί να αποτελέσει ένα σύντομο βοήθημα αναφορικά με τις κυριότερες συναρτήσεις και δυνατότητες του MATLAB και του SIMULINK. Αν κάποιος θέλει να εμβαθύνει και να εκμεταλλευτεί όλες τις δυνατότητες που προσφέρει το λογισμικό μπορεί να χρησιμοποιήσει το User's Manual ή την εντολή HELP η οποία θα αναλυθεί στο κεφάλαιο 3. Επίσης κάποιος μπορεί να χρησιμοποιήσει ένα από τα πάμπολλα FORUMS του διαδικτύου στα οποία μπορεί να βρει ειδικευμένες συναρτήσεις γραμμένες από άλλους χρήστες. Τέτοια FORUMS μπορούν να βρεθούν με ένα απλό search στο Google.

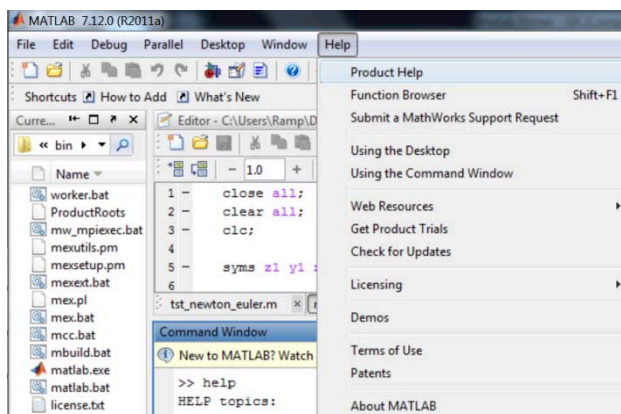
Σημαντικό να αναφερθεί ότι το MATLAB έχει αναπτυχθεί για την πλατφόρμα Windows στην οποία το mouse είναι το κύριο μέσο επικοινωνίας. Πολλές από τις εντολές του MATLAB, μπορούν να γίνουν είτε μέσω του menu ή μέσω εντολών στο Command Prompt. Οι μέθοδοι αυτοί θα αναλυθούν και αφήνεται στις προτιμήσεις του χρήστη για το ποια μέθοδο θα χρησιμοποιεί.

2. Εκκίνηση MATLAB

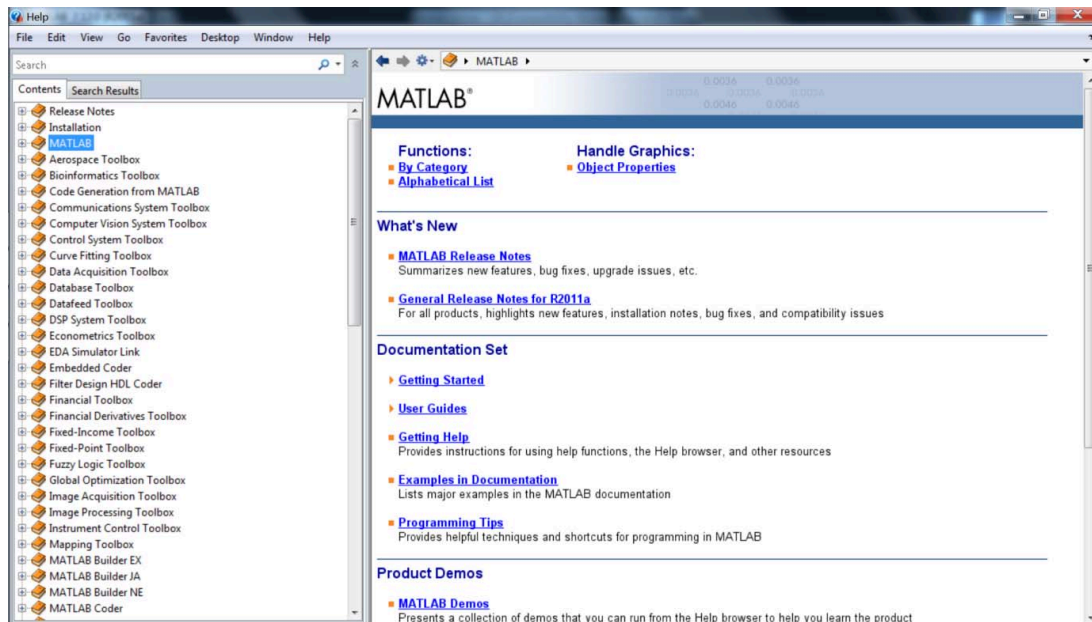
Για την εκκίνηση του MATLAB/ SIMULINK απλά βρείτε ένα ελεύθερο υπολογιστή στο PC Lab και κάνετε Login με τους κωδικούς σας ή με τους ελεύθερους κωδικούς που προσφέρονται (Username: temporal Password: !@#\$% ()). Το λογισμικό βρίσκεται στο Start Menu των Windows μέσα στο αρχείο MATH.

3. Η Εντολή HELP

Ο ευκολότερος τρόπος για να βρει κάποιος βοήθεια για κάποια διεργασία που τον ενδιαφέρει είναι μέσω της εντολής HELP την οποία μπορεί να πληκτρολογήσει στο Command Prompt ή να κάνει click στο Help που βρίσκεται στο menu πάνω δεξιά και μετά στο Product Help. Ποιο κάτω φαίνεται η διαδικασία:

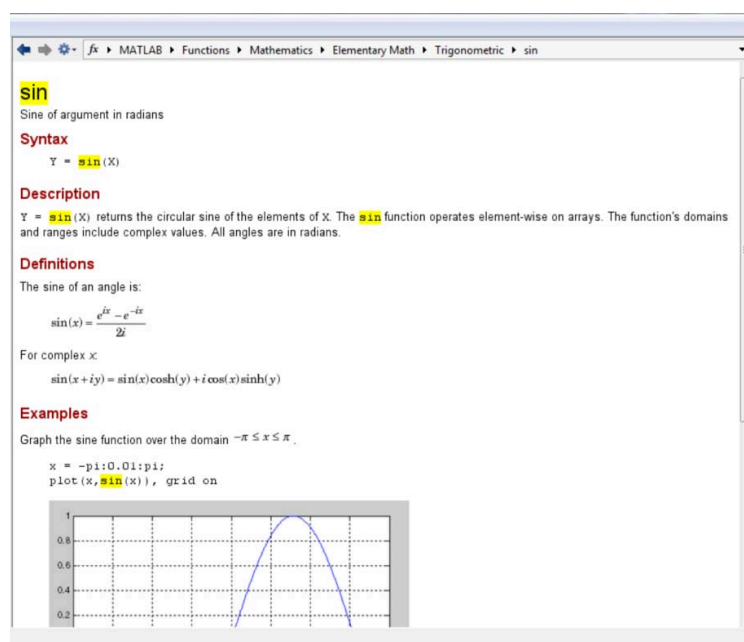
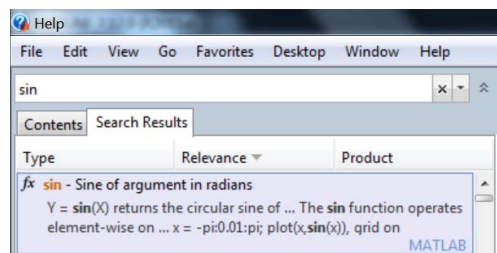



Θα εμφανιστεί ένα παράθυρο με όλα τα Toolboxes που υποστηρίζει το λογισμικό και εξερευνώντας τα, κάποιος μπορεί να βρει επεξηγήσεις για τις συναρτήσεις που τον ενδιαφέρουν μαζί με παραδείγματα εφαρμογής τους.



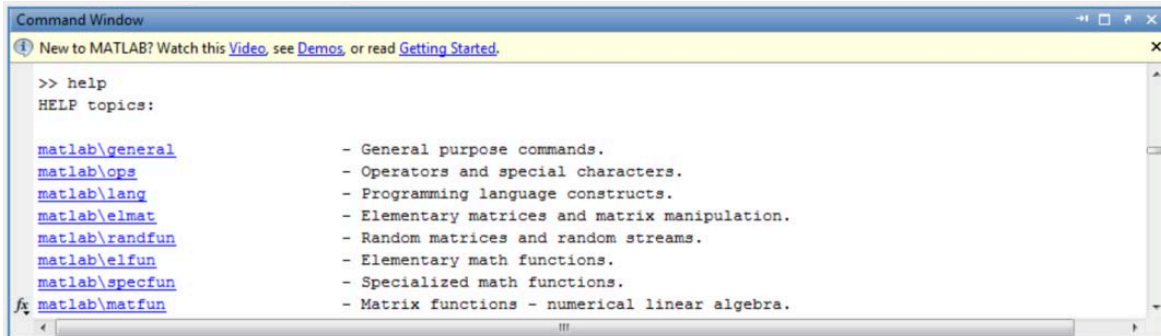
Ο ευκολότερος τρόπος για να βρει κάποιος την συνάρτηση που τον ενδιαφέρει είναι μέσω μιας απλής αναζήτησης με λέξεις κλειδιά για την σχετική συνάρτηση στο Product Help.

Στη συνέχεια φαίνεται η μορφή του περιβάλλοντος για την συνάρτηση ημιτόνου (sin):



Για έξοδο από το HELP κάποιος απλά πρέπει να κλείσει το παράθυρο του Product Help . Αν κάποιος θέλει άμεση πρόσβαση στις βιβλιοθήκες στήριξης του MATLAB χωρίς

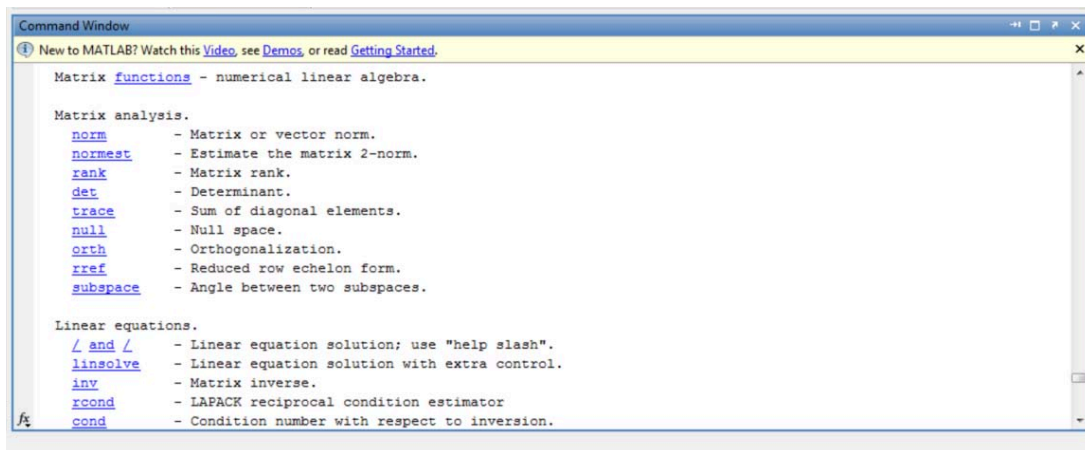
να ανοίξει ξεχωριστό παράθυρο το μόνο που χρειάζεται να κάνει είναι να πληκτρολογήσει **help** στο Command Prompt:



```
Command Window
New to MATLAB? Watch this Video, see Demos, or read Getting Started.
>> 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\randfun      - Random matrices and random streams.
matlab\elfun        - Elementary math functions.
matlab\specfun      - Specialized math functions.
matlab\matfun       - Matrix functions - numerical linear algebra.
```

Κάνοντας click στην οικογένεια συναρτήσεων που χρειάζεται να χρησιμοποιηθεί π.χ.: **matlab\matfun** μας δίνονται όλες οι συναρτήσεις:

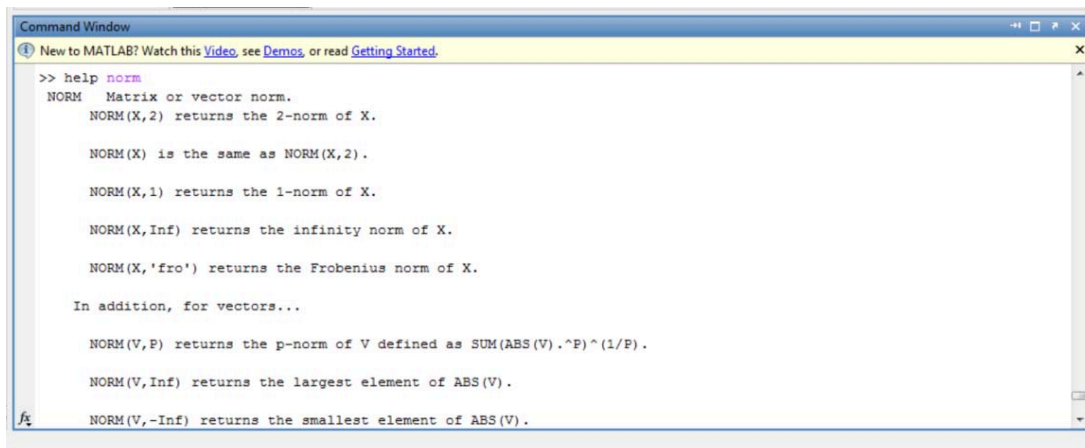


```
Command Window
New to MATLAB? Watch this Video, see Demos, or read Getting Started.
Matrix functions - numerical linear algebra.

Matrix analysis.
norm          - Matrix or vector norm.
normest       - Estimate the matrix 2-norm.
rank          - Matrix rank.
det           - Determinant.
trace         - Sum of diagonal elements.
null          - Null space.
orth          - Orthogonalization.
rref          - Reduced row echelon form.
subspace      - Angle between two subspaces.

Linear equations.
\ and /      - Linear equation solution; use "help slash".
linsolve     - Linear equation solution with extra control.
inv          - Matrix inverse.
rcond        - LAPACK reciprocal condition estimator
cond         - Condition number with respect to inversion.
```

Εναλλακτικά μπορεί κάποιος να γράψει απευθείας στο Command Prompt "help" -κενό- όνομα συνάρτησης και να βγει μία σύντομη περιγραφή της. Επίσης πρέπει να αναφερθεί ότι με την εντολή help παρέχεται μια σύντομη περιγραφή της συνάρτησης και του τρόπου λειτουργίας της (που πολύ συχνά είναι αρκετή). Για μια πιο αναλυτική περιγραφή, πρέπει κανείς να πάει μέσω του Help Menu, που περιγράφηκε πιο πάνω.



```
Command Window
New to MATLAB? Watch this Video, see Demos, or read Getting Started.
>> help norm
NORM Matrix or vector norm.
NORM(X,2) returns the 2-norm of X.

NORM(X) is the same as NORM(X,2).

NORM(X,1) returns the 1-norm of X.

NORM(X,Inf) returns the infinity norm of X.

NORM(X,'fro') returns the Frobenius norm of X.

In addition, for vectors...

NORM(V,P) returns the p-norm of V defined as SUM(ABS(V).^P)^(1/P).

NORM(V,Inf) returns the largest element of ABS(V).

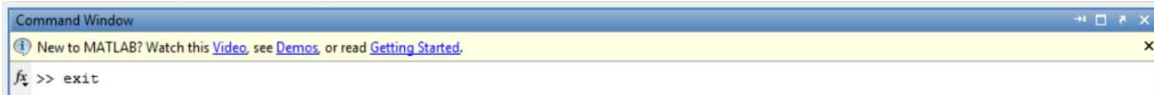
NORM(V,-Inf) returns the smallest element of ABS(V).
```

4. Διαχείριση Workspace

Υπάρχουν τέσσερις τρόποι για έξοδο από το MATLAB. Πληκτρολογώντας **quit**:

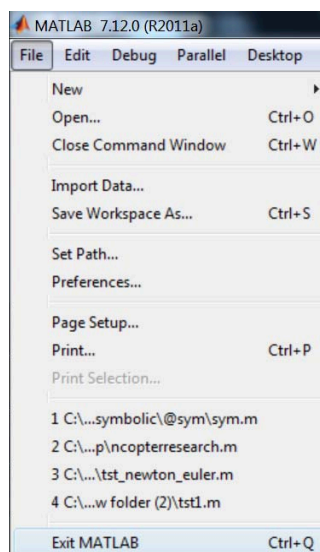



ή **exit**:



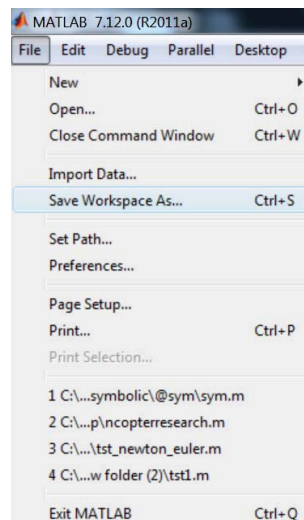
στο command prompt και πατώντας μετά το κουμπί **enter**

ή κάνοντας click στο **F**ile και μετά στο **E**xit MATLAB:

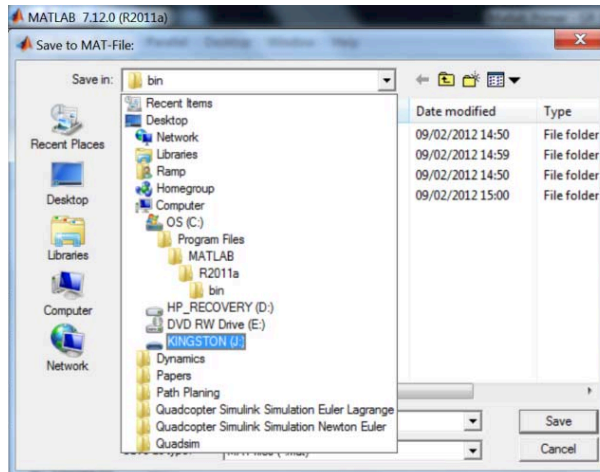


ή τέλος απλά κλείνοντας το παράθυρο από το .

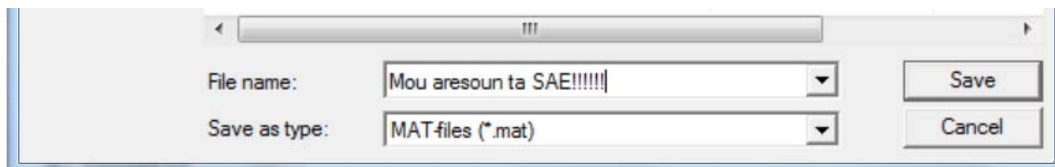
Πριν την έξοδο από το MATLAB καλή πρακτική είναι να γίνεται save το workspace για μελλοντική χρήση. Κάνουμε click στο **F**ile και μετά στο **S**ave **W**orkspace **A**s...



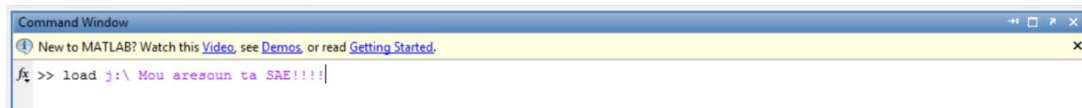
και στην συνέχεια επιλέγουμε την τοποθεσία προτίμησης. Η επιλογή μπορεί να γίνει κάνοντας click στο βελάκι δίπλα από το Save in:



μέχρι να δούμε την τοποθεσία που θέλουμε και επιλέγοντας την. Ακολούθως πληκτρολογούμε την ονομασία του αρχείου στο σημείο File name.

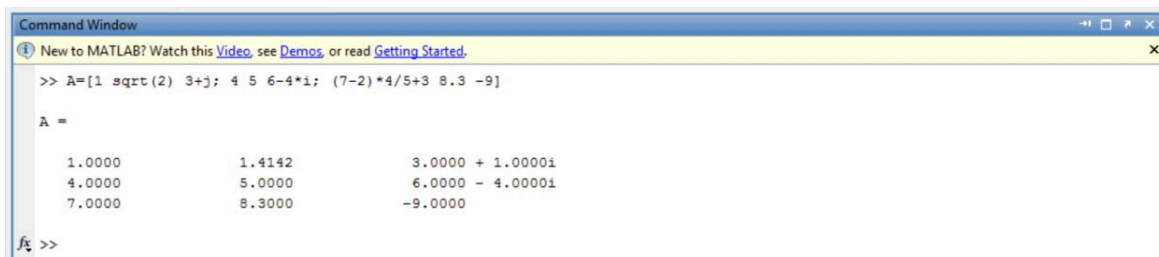


Τέλος κάνουμε click στο Save. Για να επαναφέρουμε το workspace πληκτρολογούμε **load** τοποθεσία **g** ή **f:λόνομα_αρχείου** χωρίς το extension **.mat** και πατάμε **enter**.



5. Πίνακες στο MATLAB

Η απλούστερη μέθοδος για να εισαγάγει κάποιος ένα πίνακα φαίνεται πιο κάτω όπου τα στοιχεία ενός πίνακα μπαίνουν ανάμεσα σε αγκύλες, τα στοιχεία μιας γραμμής χωρίζονται από κενό, ενώ οι γραμμές μεταξύ τους χωρίζονται με ";", όπως φαίνεται στο σχήμα:



Ισοδύναμα κάποιος μπορούσε να έχει το ίδιο αποτέλεσμα πληκτρολογώντας τα πιο κάτω όπου η αλλαγή γραμμής γίνεται με και Enter:


```

Command Window
New to MATLAB? Watch this Video, see Demos, or read Getting Started.
>> A=[1 sqrt(2) 3+
4 5 6-4*i
(7-2)*4/5+3 8.3 -9]

A =

    1.0000    1.4142    3.0000 + 1.0000i
    4.0000    5.0000    6.0000 - 4.0000i
    7.0000    8.3000   -9.0000
fx >> |

```

Το ποιο πάνω παράδειγμα θέλει να τονίσει ότι τα στοιχεία του πίνακα μπορούν να είναι μιγαδικά ή πραγματικά καθώς και με συναρτήσεις όπως η sqrt(2) ($\text{sqrt}(2) = \sqrt{2}$) ή και με πράξεις $\frac{(7-2) \times 4}{5} + 3$.

Οι συντεταγμένες των πινάκων, των βαθμωτών τους στοιχείων (1x1) και των διανυσμάτων (1 x n ή m x 1), δίνονται ποιο κάτω:

$$\begin{bmatrix} 1,1 & 1,2 & \dots & 1,n \\ 2,1 & 2,2 & \dots & 2,n \\ \vdots & \vdots & \ddots & \vdots \\ m,1 & m,2 & \dots & m,n \end{bmatrix}$$

Αν θέλουμε έτσι να απομονώσουμε το στοιχείο (2,3) του πίνακα **A**, πληκτρολογούμε:

```

Command Window
New to MATLAB? Watch this Video, see Demos, or read Getting Started.
>> A=[1 sqrt(2) 3+
4 5 6-4*i
(7-2)*4/5+3 8.3 -9]

A =

    1.0000    1.4142    3.0000 + 1.0000i
    4.0000    5.0000    6.0000 - 4.0000i
    7.0000    8.3000   -9.0000

>> B=A(2,3)

B =

    6.0000 - 4.0000i
fx >>

```

Αν είχαμε δύο διανύσματα $V = [1 \ -6 \ 7 \ 9E2 \ 22]$ (διάνυσμα-γραμμής) και $W = [1 \ 2 \ 3 \ 4e3 \ 0]$ (διάνυσμα-στήλης) με την χρήση του ";" να είναι η κύρια διαφορά.

```

Command Window
New to MATLAB? Watch this Video, see Demos, or read Getting Started.
>> V=[1 -6 7 9E2 22]

V =

    1    -6     7   900    22

>> W=[1; 2; 3; 4e3; 0]

W =

     1
     2
     3
   4000
     0
fx >>

```

τότε πληκτρολογούμε:


```

Command Window
New to MATLAB? Watch this Video, see Demos, or read Getting Started.
>> C=V(4)
C =
    900
>> D=W(4)
D =
    4000
fx >>

```

Για να απομονώσουμε το τέταρτο στοιχείο κάθε διανύσματος.

Για διανύσματα με μορφή σειράς αρκετό είναι να δώσουμε την στήλη που βρίσκετε το στοιχείο. Με τον ίδιο τρόπο για διανύσματα με μορφή στήλης αρκετό είναι να δώσουμε την σειρά του στοιχείου. Στο πιο πάνω παράδειγμα, $V(4)=V(1,4)$ και $W(4)=W(4,1)$.

Επίσης είναι δυνατός ο ορισμός μιας ακολουθίας στοιχείων χρησιμοποιώντας την άνω και κάτω τελεία (:). Για παράδειγμα:

```

Command Window
New to MATLAB? Watch this Video, see Demos, or read Getting Started.
>> t=0:5
t =
    0    1    2    3    4    5
fx >>

```

Επίσης μπορούμε να ρυθμίσουμε το βήμα μεταβολής της ακολουθίας από 1 σε 0.5 ως ακολούθως:

```

Command Window
New to MATLAB? Watch this Video, see Demos, or read Getting Started.
>> t=2:0.5:4
t =
    2.0000    2.5000    3.0000    3.5000    4.0000
fx >>

```

Έτσι αν θέλουμε η μεταβλητή **E** να έχει τις σειρές 2 και 3 του πίνακα **A** τότε πληκτρολογήστε:

```

Command Window
New to MATLAB? Watch this Video, see Demos, or read Getting Started.
>> E=A(2:3,1:3)
E =
    4.0000    5.0000    6.0000 - 4.0000i
    7.0000    8.3000    -9.0000
fx >>

```

Έτσι ένας πίνακας μπορεί να κατασκευαστεί από στοιχεία άλλων πινάκων καθώς και από άλλους πίνακες που είναι ήδη προγραμματισμένοι στις βιβλιοθήκες του MATLAB:

Προγραμματισμένοι Πίνακες	
zeros	Πίνακας με μηδενικά στοιχεία
ones	Πίνακας με όλα τα στοιχεία 1
rand	Πίνακας με τυχαία στοιχεία
eye	Ταυτοτικός πίνακας

Για παράδειγμα:

```

Command Window
New to MATLAB? Watch this Video, see Demos, or read Getting Started.
>> F=[ones(3,1) A(:,2) zeros(3,1); rand(2,1) eye(2)]

F =

    1.0000    1.4142     0
    1.0000    5.0000     0
    1.0000    8.3000     0
    0.8147    1.0000     0
    0.9058     0    1.0000

fx >>

```

Πολύ συχνά είναι χρήσιμο να μάθουμε την διάσταση ($m \times n$) ενός πίνακα. Η συνάρτηση **size** επιστρέφει ένα διάνυσμα διάστασης 1×2 με τον αριθμό των σειρών m και των στηλών n του πίνακα. Πληκτρολογώντας έτσι:

```

Command Window
New to MATLAB? Watch this Video, see Demos, or read Getting Started.
>> G=size(F)

G =

     5     3

fx >>

```

Μια γενικότερη μέθοδος αναγνώρισης/ διαχωρισμού δυο ή περισσότερων αποτελεσμάτων από μια συνάρτηση είναι ο ορισμός ίσου αριθμού μεταβλητών με κενό μεταξύ τους μέσα σε αγκύλες ([]). Για παράδειγμα:

```

Command Window
New to MATLAB? Watch this Video, see Demos, or read Getting Started.
>> [m n]=size(F)

m =

     5

n =

     3

fx >>

```

Κάποιες χρήσιμες συναρτήσεις όταν διαχειριζόμαστε πίνακες δίνονται στον πιο κάτω πίνακα:

Συναρτήσεις Στατιστικής	
max	Μέγιστη τιμή
min	Ελάχιστη τιμή
mean	Μέση τιμή
std	Τυπική απόκλιση
sum	Άθροισμα στοιχείων
prod	Γινόμενο στοιχείων

Αν κάποιος θέλει περισσότερες πληροφορίες για τις πιο πάνω συναρτήσεις μπορεί να χρησιμοποιήσει την εντολή **HELP**.

6. Πράξεις Πινάκων

Βασική αριθμητική/ πράξεις πινάκων γίνονται με μεγάλη ευκολία στο MATLAB και κάποιες από τις βασικότερες πράξεις συνοψίζονται πιο κάτω:

Πράξεις Πινάκων	
+	Πρόσθεση
-	Αφαίρεση
*	Πολλαπλασιασμός
/	Διαίρεση από δεξιά
\	Διαίρεση από αριστερά

^	Εκθετική Δύναμη
'	Ανάστροφο-συζυγής
det	Ορίζουσα
inv	Αντίστροφος
eig	Ιδιοτιμές/ ιδιοδιανύσματα

Πρέπει να σημειωθεί ότι οι μαθηματικοί κανόνες που διέπουν τις πράξεις μεταξύ πινάκων ισχύουν και στο MATLAB. Για παράδειγμα η πρόσθεση/ αφαίρεση είναι δυνατή μόνο όταν οι πίνακες έχουν τις ίδιες διαστάσεις ενώ για πολλαπλασιασμό πινάκων οι εσωτερικές διαστάσεις των πινάκων πρέπει να είναι ίδιες. Επιπρόσθετα για την δεξιά ή αριστερή διαίρεση η μέθοδος χρήσης των τελεστών γίνεται ως εξής:

$Z = X \setminus Y$ που είναι ισοδύναμο με $Z = \text{inv}(X) * Y$

$Z = X / Y$ που είναι ισοδύναμο με $Z = X * \text{inv}(Y)$.

Η εντολή, **A^2** είναι ταυτόσημη με την **A*A** όπου είναι αυτονόητο ότι ο πίνακας **A** πρέπει να είναι τετράγωνος. Η πράξη **2^A** είναι επίσης δυνατή χρησιμοποιώντας ιδιοτιμές και ιδιοδιανύσματα. Τέλος ο ερμιτιανός (ανάστροφο-συζυγής πίνακας) υπολογίζεται από την απόστροφο (') ενώ οι συναρτήσεις **det**, **inv**, και **eig** έχουν προφανείς λειτουργίες (για λεπτομερέστερη εξήγηση των συναρτήσεων χρησιμοποιήστε την εντολή **HELP** για την μέθοδο εφαρμογής τους).

7. Μαθηματικές πράξεις με διανύσματα

Αριθμητικές πράξεις στοιχείο - ανά - στοιχείο είναι δυνατές τοποθετώντας τελεία (.) μπροστά από την επιθυμητή μαθηματική πράξη όπως φαίνεται πιο κάτω:

Πράξεις Διανυσμάτων	
+	Πρόσθεση
-	Αφαίρεση
.*	Βαθμωτός πολλαπλασιασμός
./	Διαίρεση από δεξιά
.\	Διαίρεση από αριστερά
.^	Εκθετική Δύναμη
.'	Ανάστροφος

Παραδείγματα των πιο πάνω πράξεων δίνονται πιο κάτω:

Πρόσθεση Διανυσμάτων:

```

Command Window
New to MATLAB? Watch this Video, see Demos, or read Getting Started.
>> H=[1 2 3];
>> I=[4 5 6];
>> J=H+I

J =

     5     7     9
fx >>

```

Αφαίρεση Διανυσμάτων:

```

Command Window
New to MATLAB? Watch this Video, see Demos, or read Getting Started.
>> J=H-I

J =

    -3    -3    -3
fx >>

```

Βαθμωτός πολλαπλασιασμός Διανυσμάτων:



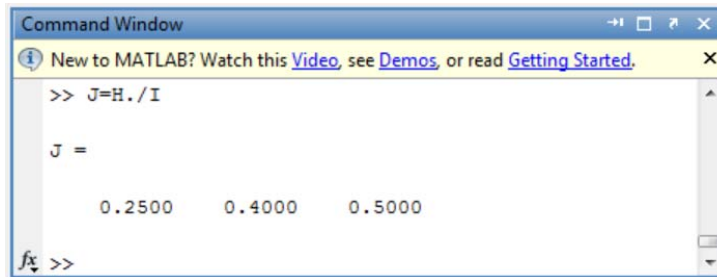
```
Command Window
New to MATLAB? Watch this Video, see Demos, or read Getting Started.
>> J=H.*I

J =

     4     10     18

fx >>
```

Διαίρεση από δεξιά



```
Command Window
New to MATLAB? Watch this Video, see Demos, or read Getting Started.
>> J=H./I

J =

    0.2500    0.4000    0.5000

fx >>
```

Διαίρεση από αριστερά



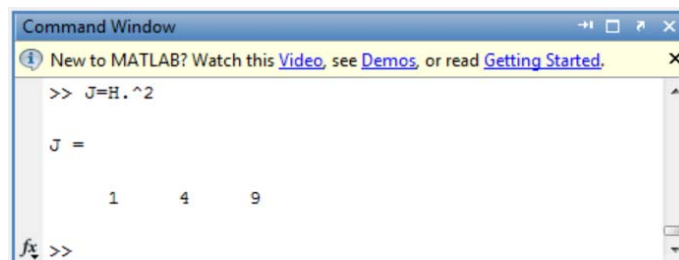
```
Command Window
New to MATLAB? Watch this Video, see Demos, or read Getting Started.
>> J=H.\I

J =

    4.0000    2.5000    2.0000

fx >>
```

Εκθετική Δύναμη



```
Command Window
New to MATLAB? Watch this Video, see Demos, or read Getting Started.
>> J=H.^2

J =

     1     4     9

fx >>
```

Σημαντικό να παρατηρηθεί ότι το ερωτηματικό (;) στο τέλος μιας εντολής δεν εμφανίζει το αποτέλεσμα/ έξοδο στο Command Prompt. Πιο κάτω συνοψίζονται βασικές συναρτήσεις. Αφήνεται στον αναγνώστη να μάθει πως να τις εφαρμόσει (π.χ. Help **atan2**):

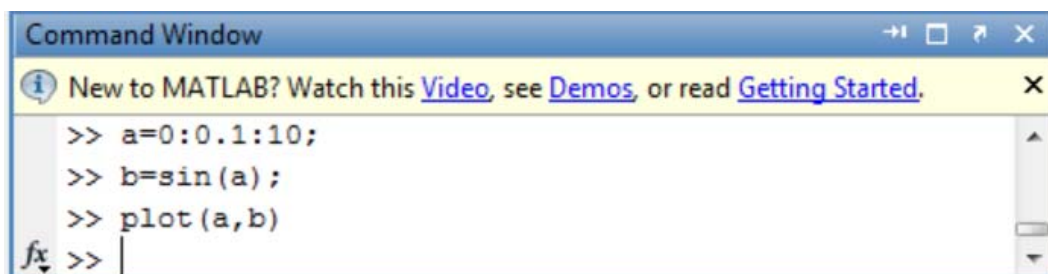
Τριγωνομετρικές Συναρτήσεις		Βασικές Μαθηματικές Συναρτήσεις	
sin	Ημίτονο	abs	Απόλυτη τιμή ή πλάτος μιγαδικού
cos	Συνημίτονο	angle	Γωνία φάσης
tan	Εφαπτόμενη	sqrt	Τετραγωνική ρίζα
asin	Τόξο ημιτόνου	real	Πραγματικό μέρος
acos	Τόξο συνημιτόνου	imag	Φανταστικό μέρος
atan	Τόξο εφαπτομένης	conj	Συζυγής Μιγαδικός
atan2	Four quadrant arctangent	round	Στρογγυλ. πλησιέστερο ακέραιο
sinh	Υπερβολικό ημίτονο	fix	Στρογγυλοποίηση προς το μηδέν
cosh	Υπερβολικό συνημίτονο	floor	Στρογγυλοποίηση προς $-\infty$
tanh	Υπερβολική εφαπτομένη	ceil	Στρογγυλοποίηση προς $+\infty$
asinh	Υπερβολικό τόξο ημιτόνου	sign	Επιστρέφει το πρόσημο μιας τιμής
acosh	Υπερβολικό τόξο	rem	Επιστρέφει το υπόλοιπο
atanh	συνημιτόνου	exp	Εκθετική ύψωση
	Υπερβολικό τόξο	log	Φυσικός λογάριθμος
	εφαπτομένης	log10	Λογάριθμος με βάση 10

8. Γραφικές Παραστάσεις

Ένας από τους κύριους λόγους της ευρείας χρήσης του MATLAB είναι η δυνατότητα οπτικοποίησης δεδομένων υπό την μορφή δισδιάστατων και τρισδιάστατων γραφικών παραστάσεων. Τα δεδομένα που προκύπτουν από τους αριθμητικούς υπολογισμούς, τα πειράματα και τις προσομοιώσεις μπορούν να επεξεργαστούν εύκολα και να οπτικοποιηθούν με την μορφή δισδιάστατων/ τρισδιάστατων γραφικών παραστάσεων. Μερικές από τις πιο διαδεδομένες εντολές οπτικοποίησης του MATLAB δίνονται πιο κάτω:

Εντολές Οπτικοποίησης	
plot	Δισδιάστατη γραφική παράσταση
loglog	Λογαριθμική γραφική παράσταση
semilogx	Λογαριθμικός άξονας x, γραμμικός άξονας y
semilogy	Λογαριθμικός άξονας y, γραμμικός άξονας x
polar	Πολική γραφική παράσταση
mesh	3-D γραφική πλέγματος
contour	Ισοψής γραφική παράσταση
bar	Γραφική παράσταση στηλών
stairs	Διακριτή γραφική παράσταση

Η εντολή **plot** είναι η πιο πολύ χρησιμοποιημένη εντολή οπτικοποίησης με τις εντολές **loglog**, **semilogx**, **semilogy**, και **polar** να χρησιμοποιούνται με τον ίδιο τρόπο. Παραδείγματος χάριν οι εντολές,

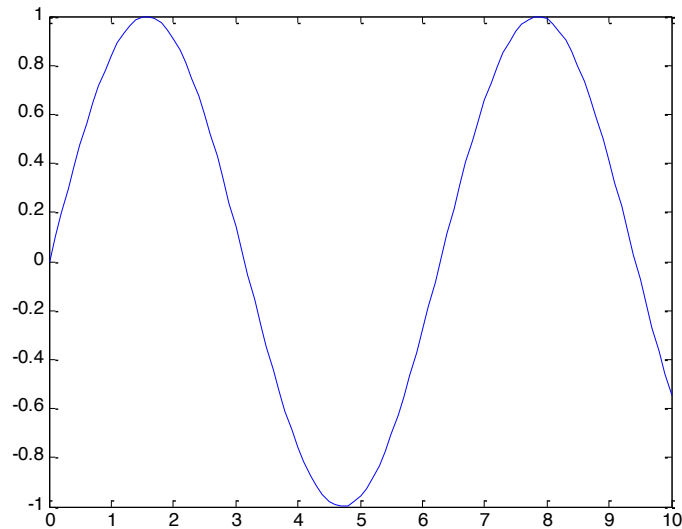


```

Command Window
New to MATLAB? Watch this Video, see Demos, or read Getting Started.
>> a=0:0.1:10;
>> b=sin(a);
>> plot(a,b)
fx >>

```

Παράγουν την δισδιάστατη γραφική όπου τα στοιχεία του διανύσματος **b** τυπώνονται συναρτήσει των στοιχείων του διανύσματος **a**.



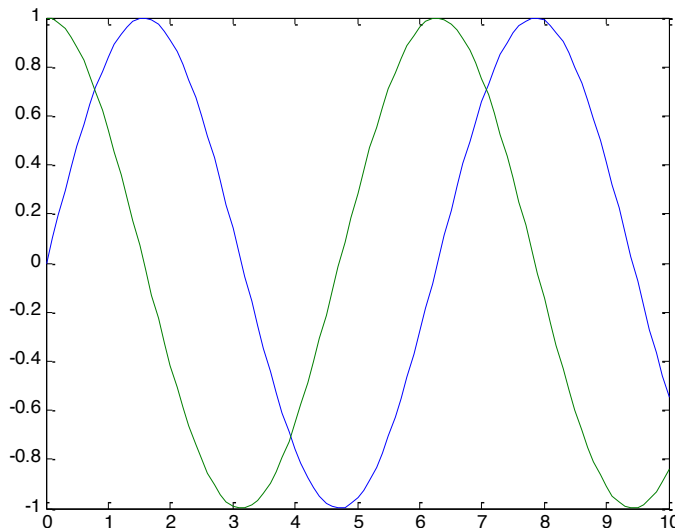
Επιπρόσθετα διανύσματα μπορούν να τυπωθούν στην ίδια γραφική όπως φαίνεται πιο κάτω:

```

Command Window
New to MATLAB? Watch this Video, see Demos, or read Getting Started.
>> c=cos(a);
>> plot(a,b,a,c)
fx >>

```

Με έξοδο στην γραφική:

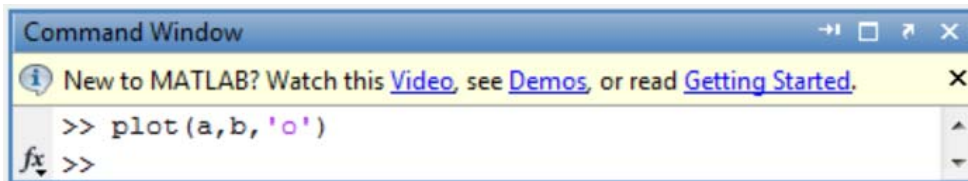


Αν η εντολή plot καλείτε με μια μόνο μεταβλητή τότε τα στοιχεία του διανύσματος εκτυπώνονται συναρτήσει των συντεταγμένων θέσης τους στον πίνακα ή στο διάνυσμα. Το MATLAB δίνει την δυνατότητα ρυθμίσεων στον τρόπο οπτικοποίησης των δεδομένων όπως αλλαγή του τύπου γραμμής, ρύθμιση χρωμάτων, τιτλοδότηση αξόνων έτσι ώστε ο χρήστης να μπορεί με ευκολία και αποτελεσματικότητα να αναλύει διάφορα δεδομένα.

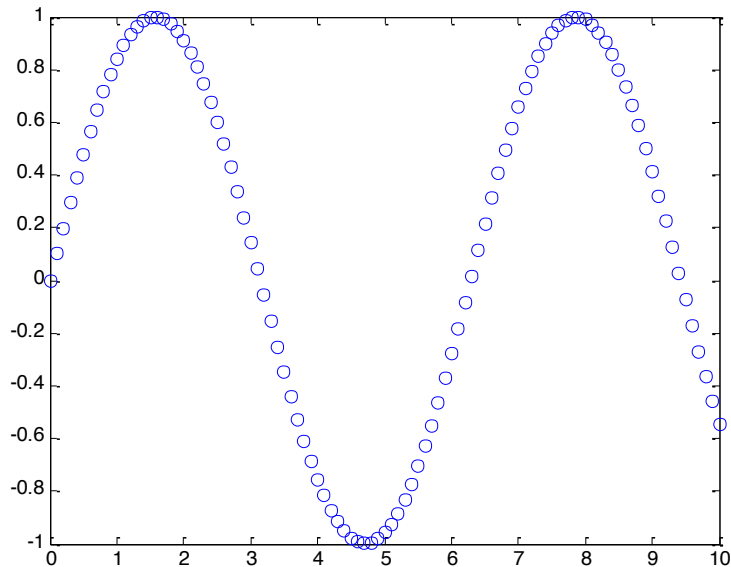
Μερικές από τις γραφικές ρυθμίσεις συνοψίζονται στον πίνακα πιο κάτω:

Τύπος Γραμμής		Χρώμα	
.	Σημείο	y	Κίτρινο
o	Κύκλος	g	Πράσινο
x	Τύπου X	c	Κυανό
+	Σταυροειδής	r	Κόκκινο
*	Αστεράκι	m	Μοβ
-	Συνεχής	b	Μπλε
:	Διακεκομμένη	w	Άσπρο
-.	Συνεχής με τελεία	k	Μαύρο
--	Συνεχής με κενό		

Η μέθοδος εκτέλεσης της εντολής δίνετε από: **plot(vector1,vector2,'line-type')**



Με έξοδο στην γραφική:



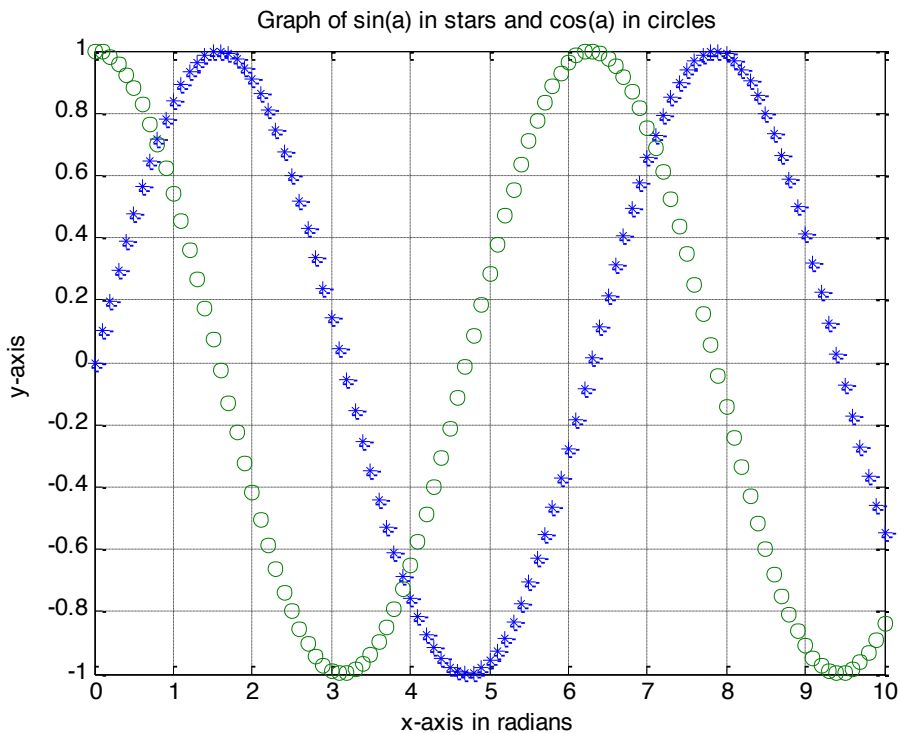
Οι εντολές ταυτοποίησης μιας γραφικής παράστασης δίνονται πιο κάτω:

Τιτλοδότηση Γραφικής	
title	Τίτλος
xlabel	Ονομασία άξονα x
ylabel	Ονομασία άξονα y
text	Κείμενο στην γραφική
gtext	Κείμενο με ρυθμισμένη θέση
grid	Πλέγμα

Η μέθοδος εφαρμογής τους είναι αρκετά απλή, για παράδειγμα:


```
Command Window
New to MATLAB? Watch this Video, see Demos, or read Getting Started.
>> plot(a,b,'*',a,c,'o')
>> xlabel('x-axis in radians')
>> ylabel('y-axis')
>> title('Graph of sin(a) in stars and cos(a) in circles')
>> grid
fx >>
```

Με έξοδο στη γραφική:



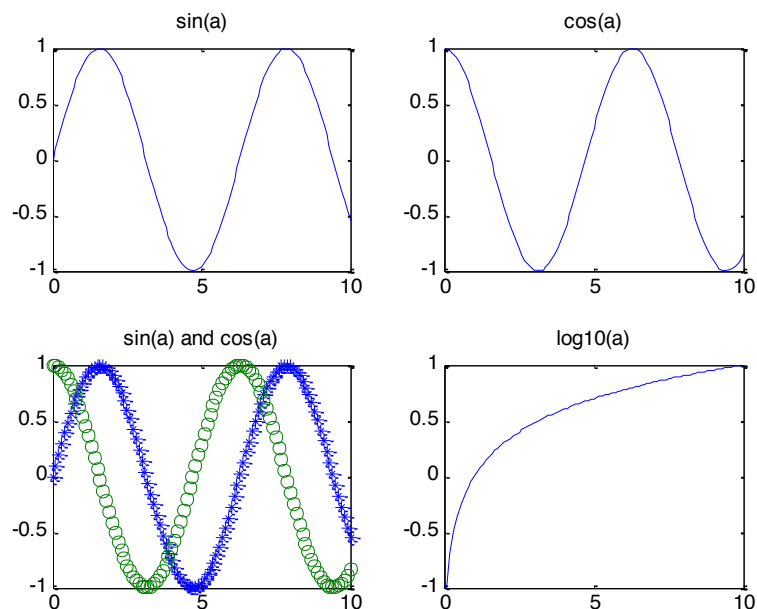
Συχνά είναι αναγκαία η οπτικοποίηση πολλών γραφικών σε ένα φύλλο. Η εντολή που διαιρεί την γραφική παράσταση σε 'υπό-γραφικές' παραστάσεις είναι η **subplot**. Ο τρόπος εφαρμογής της εντολής δίνεται ως εξής: **subplot(mnr)**. Όπου, **m** είναι ο αριθμός των σειρών και **n** είναι ο αριθμός των στηλών που θέλουμε να διαιρέσουμε την γραφική παράσταση. Η μεταβλητή **r** αναφέρετε σε ποιο παράθυρο θέλουμε να εμφανίσουμε τα δεδομένα. Για παράδειγμα οι πιο κάτω εντολές μας δίνουν τα ακόλουθα αποτελέσματα:

```

Command Window
New to MATLAB? Watch this Video, see Demos, or read Getting Started.
>> subplot(221),plot(a,b)
>> title('sin(a)')
>> subplot(222),plot(a,c)
>> title('cos(a)')
>> subplot(223),plot(a,b,'*',a,c,'o')
>> title('sin(a) and cos(a)')
>> subplot(224),plot(a,log10(a))
>> title('log10(a)')
fx >>

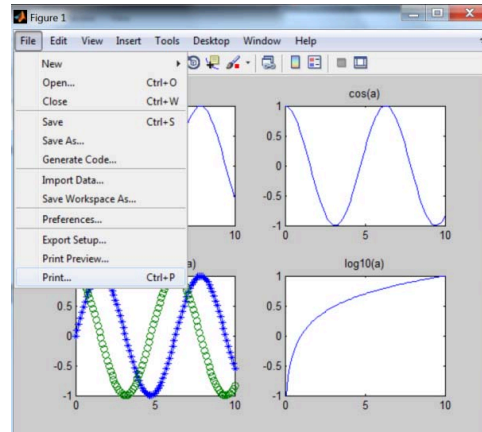
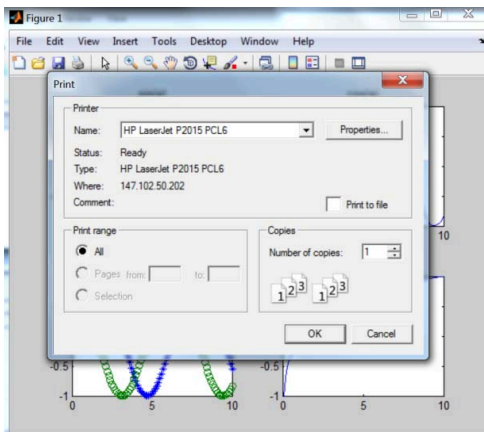
```

Με έξοδο στην γραφική:



Επιπρόσθετα αν είναι επιθυμητή η αλλαγή της κλίμακας των αξόνων για καλύτερη παρουσίαση των δεδομένων πληκτρολογώντας: **axis**([*x-min* *x-max* *y-min* *y-max*]). Όπου, **x-min** και **x-max** είναι οι ελάχιστες και μέγιστες τιμές του άξονα μας και οι **y-min** και **y-max** αντίστοιχα για τον κατακόρυφο άξονα.

Για την εκτύπωση μίας γραφικής, από το παράθυρο της γραφικής επιλέξετε **File**→ **Print**→ **OK**.



9. Εντολές Workspace

Ακολουθεί μια λίστα χρήσιμων εντολών που περιέχονται στο MATLAB και δεν έχουν προαναφερθεί:

Εντολές Workspace	
who	Λίστα αποθηκευμένων μεταβλητών
whos	Σαν την who με περισσότερες πληροφορίες
what	Λίστα αρχείων τύπου .M που χρησιμοποιούνται
clc	Καθαρίζει την οθόνη εντολών
clear	Καθαρίζει όλες τις μεταβλητές από την μνήμη
clear μεταβλητή	Καθαρίζει την μεταβλητή από την μνήμη
format	Αλλάζει τον τρόπο εμφάνισης αριθμητικών δεδομένων
...	Δηλώνει ότι η εντολή συνεχίζεται στην επόμενη γραμμή
hold	Διατηρεί την γραφική στην οθόνη
delete	Διαγραφή αρχείου
<Ctrl>-C	Τοπική ακύρωση κώδικα που τρέχει
<Esc>	Διαγράφει μια γραμμή εισόδου
<Home>	Μετακινεί τον δρομέα στην αρχή μιας γραμμής
<End>	Μετακινεί τον δρομέα στο τέλος μιας γραμμής

Στο Matlab, οι μεταβλητές **Z** και **z** (κεφαλαία-πεζά) είναι διαφορετικές.

```

Command Window
New to MATLAB? Watch this Video, see Demos, or read Getting Started.
>> Z= [1 3 7]

Z =

     1     3     7

>> z=144000

z =

    144000
  
```

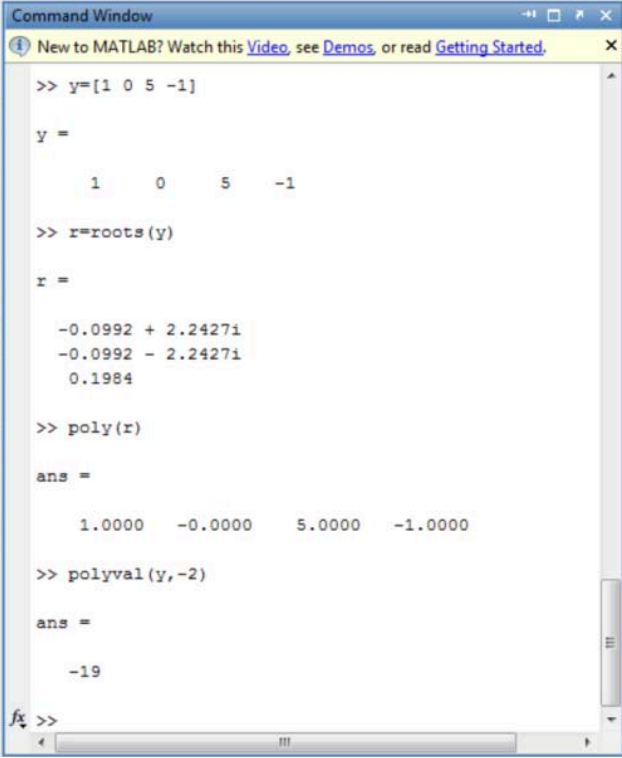
10. Πολυωνυμικές Πράξεις

Στο MATLAB τα πολυώνυμα συμβολίζονται με ένα διάνυσμα σειράς. Για ένα πολυώνυμο νιοστής τάξης το MATLAB χρειάζεται ένα διάνυσμα σειράς μήκους $n+1$. Αν θέλουμε να εκφράσουμε το πολυώνυμο $y=x^3+5x-1$ σαν διάνυσμα σειράς τότε απλά δίνουμε την εντολή $y=[1\ 0\ 5\ -1]$. Σημαντικό ότι οι συντελεστές της μεταβλητής x έχουν εισαχθεί στο διάνυσμα y έτσι το "0" σημαίνει ότι ο όρος " x^2 " έχει μηδενικό συντελεστή.

Το MATLAB προσφέρει πολύ χρήσιμες εντολές διαχείρισης πολυωνύμων οι οποίες δίνονται στον επόμενο πίνακα:

Πολυωνυμικές Πράξεις	
roots	Ρίζες του πολυώνυμου
poly	Χαρακτηριστικό πολυώνυμο
polyval	Υπολογισμός τιμής συναρτήσεως μεταβλητής
conv	Συνέλιξη πολυωνύμου
deconv	Αποσυνέλιξη πολυωνύμου μέσω διαίρεσης

Για παράδειγμα:



```
Command Window
New to MATLAB? Watch this Video, see Demos, or read Getting Started.
>> y=[1 0 5 -1]
y =
     1     0     5    -1
>> r=roots(y)
r =
   -0.0992 + 2.24271i
   -0.0992 - 2.24271i
    0.1984
>> poly(r)
ans =
    1.0000   -0.0000    5.0000   -1.0000
>> polyval(y,-2)
ans =
   -19
fx >>
```

Στο πιο πάνω παράδειγμα οι ρίζες του y έχουν υπολογιστεί και αποθηκευτεί στο διάνυσμα r . Επιπρόσθετα πληκτρολογώντας $\text{poly}(r)$ οι αρχικοί συντελεστές του y έχουν επιστραφεί. Τέλος το πολυώνυμο υπολογίστηκε στο $x=-2$ μέσω της συνάρτησης polyval .

Αν υπήρχε και δεύτερο πολυώνυμο π.χ. $w=2x^2-x-1$, τότε το αντίστοιχο διάνυσμα που το ορίζει δίνετε από $w=[2\ -1\ -1]$ και θα μπορούσαμε να κάνουμε τα ακόλουθα:

```

Command Window
New to MATLAB? Watch this Video, see Demos, or read Getting Started.
>> w=[2 -1 -1]

w =

     2     -1     -1

>> z=conv(w,y)

z =

     2     -1     9     -7     -4     1

>> [Q,R]=deconv(z,w)

Q =

     1     0     5     -1

R =

     0     0     0     0     0     0

fx >>

```

Έτσι η συνέλιξη των w και y μας επιστρέφει $z=2x^5-x^4+9x^3-7x^2-4x+1$ και η αποσυνέλιξη των z με w επιστρέφει τους συντελεστές του y χωρίς υπόλοιπα.

11. Λογικοί Τελεστές

Το MATLAB είναι εξοπλισμένο με τρεις λογικούς τελεστές οι οποίοι φαίνονται στον πίνακα κάτω και δρουν ανά στοιχείο σε ένα πίνακα. Το MATLAB είναι προγραμματισμένο να θεωρεί οποιαδήποτε μεταβλητή με μη-μηδενικό πραγματικό μέρος σαν ΑΛΗΘΗ-TRUE και επιστρέφει την τιμή "1" ενώ όλες οι άλλες περιπτώσεις θεωρούνται ΑΝΑΛΗΘΕΪΣ-FALSE και η αξία "0" επιστρέφεται.

Λογικοί Τελεστές	
&	Και
	Ή
~	Όχι/Δεν

Οι τελεστές & και | μπορούν να συγκρίνουν δύο αριθμούς ή δυο πίνακες ίσων διαστάσεων. Ο τελεστής ~ επιδρά μόνο σε μία μεταβλητή.

Για παράδειγμα:

```

Command Window
New to MATLAB? Watch this Video, see Demos, or read Getting Started.
>> v=[1 0 -3+i -4] & [-1 2 6 9]

v =

     1     0     0     1

>> v=[1 0 -3+i -4] | [-1 2 6 9]

v =

     1     1     1     1

>> ~v

ans =

     0     0     0     0

fx >>

```

Συγκρίσεις μεταξύ πινάκων μπορεί να γίνει μέσω των τελεστών που φαίνονται στον ποιο κάτω πίνακα και στα παραδείγματα:

Τελεστές Σύγκρισης	
<	Λιγότερο από
<=	Λιγότερο ή ίσο
>	Μεγαλύτερο από
>=	Μεγαλύτερο ή ίσο
==	Ίσο
~=	Άνισο

```

Command Window
New to MATLAB? Watch this Video, see Demos, or read Getting Started.
>> v=[1 -1 2 -2] <= [4 -5 6 -7]

v =

     1     0     1     0

>> v=[1 -1 2 -2] > [4 -5 6 -7]

v =

     0     1     0     1

fx >>

```

Δύο συναρτήσεις που χρησιμοποιούνται συχνά μαζί με λογικούς τελεστές είναι οι **any** και **all**. Πληκτρολογώντας **any(A,i)**, η τιμή "1" θα επιστραφεί αν το πραγματικό μέρος οποιουδήποτε στοιχείου του A είναι μη-μηδενικό για την δοθείσα διάσταση i. Αν το **A** είναι διάνυσμα τότε ο

ορισμός της διάστασης παραλείπεται. Η συνάρτηση **all** δουλεύει με τον ίδιο τρόπο με την διαφορά ότι όλα τα στοιχεία πρέπει να είναι μη-μηδενικά για να επιστραφεί η αξία "1".

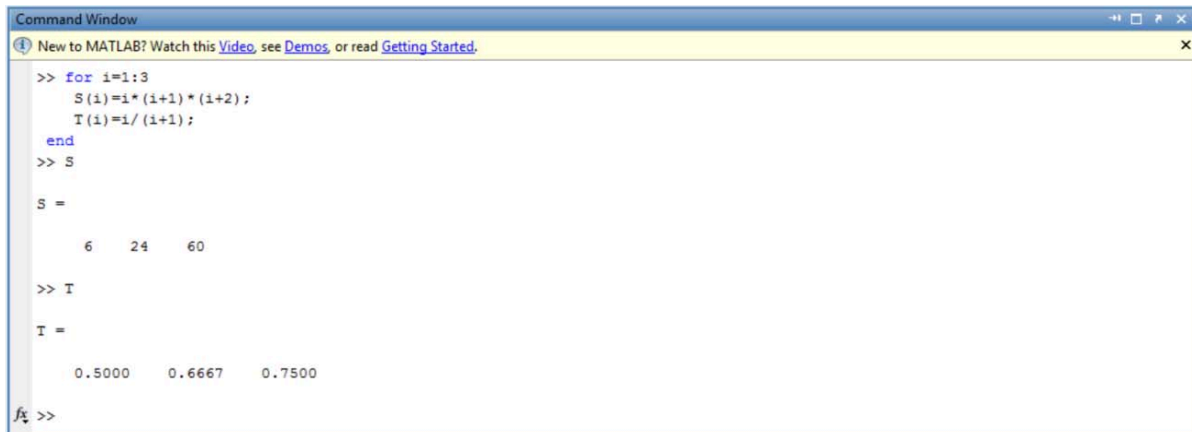
12. Έλεγχος Ροής

Το MATLAB έχει αρκετές εντολές έλεγχου ροής παρόμοιες με αυτές που χρησιμοποιούνται στις περισσότερες προγραμματιστικές γλώσσες. Οι εντολές αυτές μας δίνουν την δυνατότητα να δημιουργήσουμε ειδικευμένα προγράμματα για την εφαρμογή που χρειάζεται. Οι εντολές **FOR**, **WHILE**, **IF**, και **IF-ELSE** θα εξηγηθούν ποιο κάτω μέσω παραδειγμάτων:

Επαναληπτική εντολή **FOR**

```
for v = expression  
statements  
end
```

Εφαρμογή εντολής for:



```
Command Window  
New to MATLAB? Watch this Video, see Demos, or read Getting Started.  
>> for i=1:3  
    S(i)=i*(i+1)*(i+2);  
    T(i)=1/(i+1);  
end  
>> S  
  
S =  
  
     6     24     60  
  
>> T  
  
T =  
  
 0.5000  0.6667  0.7500  
fx >>
```

Η εντολή for δίνει αρχική αξία στην μεταβλητή *i*. Ακολουθως εκτελεί τις μαθηματικές εξισώσεις ορίζοντας τις τιμές των διανυσμάτων **S(i)** και **T(i)** βάση των εξισώσεων. Ακολουθως ανανεώνεται η αξία της μεταβλητής *i* κατά 1 και επαναλαμβάνονται οι υπολογισμοί. Η ποιο πάνω διαδικασία επαναλαμβάνεται μέχρι η μεταβλητή *i* να πάρει την τελική τις τιμή που ισούται με 3 στην περίπτωση μας.

Επαναληπτική εντολή **WHILE**

```
while expression  
statements  
end
```

Η εντολή while εκτελεί τον εσωτερικό βρόγχο επαναληπτικά όσο η συνθήκη που έχει οριστεί ικανοποιείται.

Εφαρμογή εντολής while:


```

Command Window
New to MATLAB? Watch this Video, see Demos, or read Getting Started.
>> x=1;
>> while x^3 <= 100
    x=x+1;
end
>> x

x =

    5
fx >> |

```

Επαναληπτική εντολή IF
<i>if expression</i>
<i>statements</i>
end

Η εντολή if εκτελεί τον εσωτερικό βρόγχο μια φορά αν η συνθήκη που έχει οριστεί ικανοποιείται. Εφαρμογή εντολής if:

```

Command Window
New to MATLAB? Watch this Video, see Demos, or read Getting Started.
>> d=5;
>> if d>0
    e=sqrt(d);
end
>> e

e =

    2.2361
fx >>

```

Επαναληπτική εντολή IF- ELSE
<i>if expression</i>
<i>statements</i>
else
<i>statements</i>
end

Η εντολή if-else ελέγχει αν η συνθήκη που έχει οριστεί ισχύει. Αν ισχύει εκτελεί τον πρώτο βρόγχο. Αν δεν ισχύει εκτελεί τον δεύτερο. Σημαντικό ότι μπορούμε να έχουμε όσους βρόγχους θέλουμε που λειτουργούν με την ίδια λογική.

Εφαρμογή εντολής if-else:

```

Command Window
New to MATLAB? Watch this Video, see Demos, or read Getting Started.
>> g=-2;
>> if g>=0
    g=g^2;
else
    g=abs(g);
end
>> g

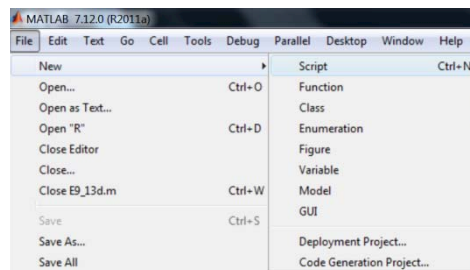
g =

    2
  
```

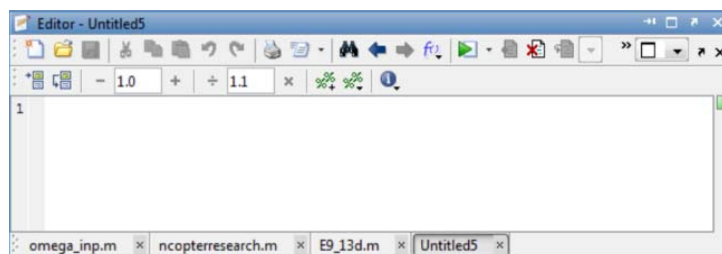
Η εντολή **if else** αποτελείται από δυο βρόγχους και η δυνατότητα εκτέλεσης του κάθε βρόγχου εξαρτάται εάν η συνθήκη εισόδου στον βρόγχο ικανοποιείται. Στην περίπτωση μας η αξία της μεταβλητής **g** ορίζει ποιος βρόγχος θα εκτελεστεί.

13. Συναρτήσεις .m

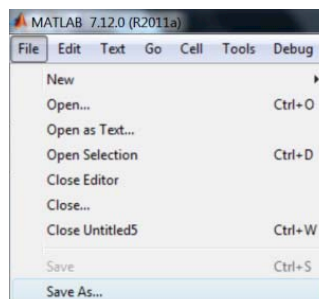
Το MATLAB επιτρέπει την δημιουργία από τον χρήστη συναρτήσεων μέσω των αρχείων .m. Για να γράψουμε ένα αρχείο .m κάνουμε click στο **File** μετά στο **New** και μετά στο **Script**:



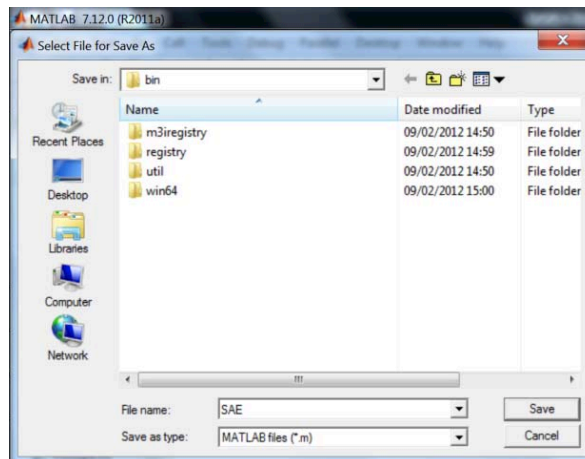
Ακολούθως ανοίγει το παράθυρο εργασίας όπου ο χρήστης μπορεί να βάλει τον κώδικα του.



Όταν ο χρήστης επιθυμεί να κάνει save τότε απλά πηγαίνει στο File → Save As

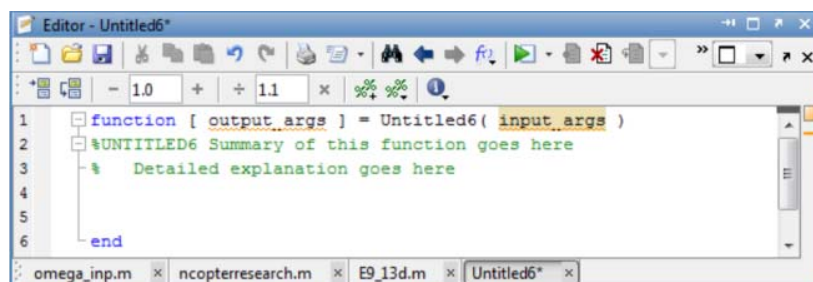


και μετά μπορεί να ονομάσει το αρχείο .m.

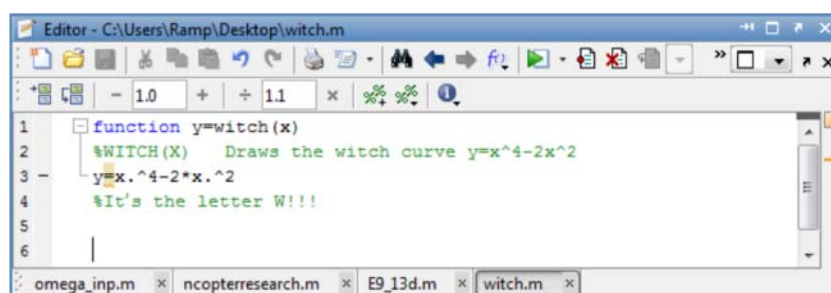


Όταν δημιουργείτε ένα αρχείο .m η πρώτη γραμμή πρέπει να ξεκινά με την εντολή function με την γενική μορφή να δίνετε ως:

function [output1, ...] = όνομα συνάρτησης (μεταβλητή1, μεταβλητή2, etc.) εάν η συνάρτηση έχει εξόδους ή function όνομα συνάρτησης (μεταβλητή1, μεταβλητή2, etc.) εάν δεν έχει. Οι μεταβλητές "μεταβλητή1" κ.λπ. αποτελούν τις εισόδους της συνάρτησης. Η συνάρτηση πρέπει να σώζεται ακριβώς με το "όνομα συνάρτησης" που φαίνεται στην πρώτη γραμμή της.



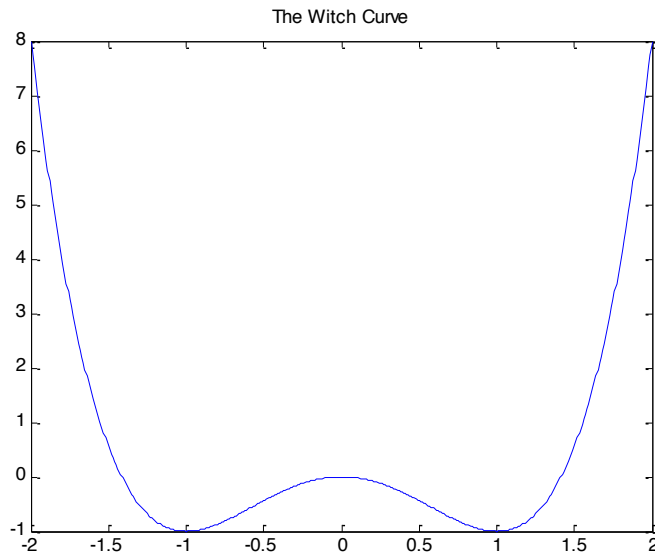
Για παράδειγμα:



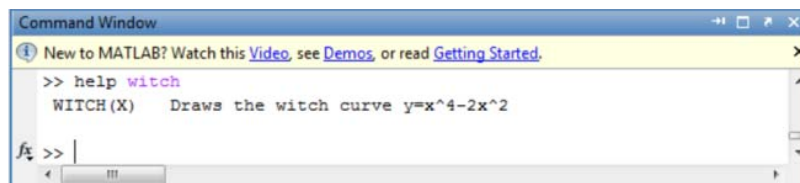
η συνάρτηση που περιγράφει το αρχείο .m ονομάζεται *witch.m* και καλείται ως εξής:



Και έχει ως έξοδο την:



Όπου το σύμβολο (%) χρησιμοποιείται για τον ορισμό σχόλιου αναφορικά με τον κώδικα και έτσι το compiler ξέρει να το αγνοήσει. Αν κάποιος δοκιμάσει να βάλει την εντολή **help witch**, τότε θα πάρει απλά τα σχόλια της πρώτης γραμμής:



14. Εφαρμογές Matlab στον Αυτόματο Έλεγχο

Στην θεωρία του αυτομάτου ελέγχου ένα δυναμικό σύστημα περιγράφεται είτε με συνάρτηση μεταφοράς (Σ.Μ) είτε στο χώρο κατάστασης (Χ.Κ). Το MATLAB περιέχει τις συναρτήσεις `tf` και `ss` καθώς και τη `zpk` (Πίνακας 14.1) για τις περιγραφές αυτές.

Επίσης, το MATLAB επιτρέπει την μετάβαση από το χώρο κατάστασης σε συνάρτηση μεταφοράς και αντίστροφα. Επιπλέον, εάν οι μηδενιστές, οι πόλοι και το κέρδος της Σ.Μ είναι γνωστά, τότε η Σ.Μ καθώς και ο Χ.Κ μπορούν να υπολογιστούν απευθείας και αντίστροφα. Οι απαραίτητες εντολές παρουσιάζονται στον Πίνακα 14.2.

Πίνακας 14.1

Βασικές Περιγραφές Συνεχών Συστημάτων	
<code>sys=tf(num,den)</code>	Συνάρτηση Μεταφοράς
<code>sys=ss(A,B,C,D)</code>	Χώρος Κατάστασης
<code>sys=zpk(Z,P,K)</code>	Μηδενιστές-Πόλοι-Κέρδος Συστήματος

Πίνακας 14.2

Μετατροπές Περιγραφών Συστημάτων	
<code>[num, den]=ss2tf(A,B,C,D)</code>	Χώρος Κατάστασης σε Συνάρτηση Μεταφοράς

[A,B,C,D]=tf2ss(num,den)	Συνάρτηση Μεταφοράς σε Χώρο Κατάστασης
[Z,P,K]=ss2zp(A,B,C,D)	Χώρος Κατάστασης σε Πόλους-Μηδενιστές-Κέρδος
[Z,P,K]=tf2zp(num,den)	Συνάρτηση Μεταφοράς σε Πόλους-Μηδενιστές-Κέρδος
[num, den]=zp2tf(Z,P,K)	Πόλοι-Μηδενιστές-Κέρδος σε Συνάρτηση Μεταφοράς
[A,B,C,D]=zp2ss(Z,P,K)	Πόλοι-Μηδενιστές-Κέρδος σε Χώρο Κατάστασης

Στους πίνακες 14.1 και 14.2, **num** και **den** είναι τα πολυώνυμα που ορίζουν τον αριθμητή και τον παρονομαστή, αντίστοιχα, της συνάρτησης μεταφοράς του συστήματος. Οι πίνακες **A,B,C,D** είναι οι πίνακες περιγραφής του συστήματος στο χώρο κατάστασης και **Z, P** είναι τα διανύσματα που περιέχουν τους μηδενιστές και τους πόλους του συστήματος και **K** είναι το κέρδος του.

Παράδειγμα 14.1: Να βρεθεί μια περιγραφή στον Χώρο Κατάστασης του συστήματος που έχει συνάρτηση μεταφοράς $G(s)=1/(s^2+2s+1)$.

Οι απαιτούμενες εντολές φαίνονται στο Σχήμα 14.1.

```
% Define system's Transfer Function (T.F)
num=1; % numerator of the T.F
den=[1 2 1]; % denominator of the T.F
% Conversion to State Space
[A, B, C, D]=tf2ss(num,den)
```

Σχήμα 14.1. Κώδικας για τον υπολογισμό του Χ. Κ. από Σ. Μ.

Για τον υπολογισμό της βηματικής και κρουστικής απόκρισης ενός συστήματος, το MATLAB διαθέτει τις εντολές **step** και **impulse** αντίστοιχα. Για οποιαδήποτε άλλη αυθαίρετη είσοδο χρησιμοποιείται η εντολή **lsim**. Επίσης, για απόκριση σε αρχικές συνθήκες υπάρχει η εντολή **initial** (σε αυτή την περίπτωση το σύστημα πρέπει να περιγραφεί στον Χ.Κ). Η σύνταξη των παραπάνω εντολών παρουσιάζεται στον Πίνακα 14.3.

Πίνακας 14.3

Απόκριση Συνεχών Συστημάτων	
step(sys)	Βηματική απόκριση
impulse(sys)	Κρουστική απόκριση
lsim(sys,u,t)	Απόκριση σε οποιαδήποτε είσοδο
initial(sys,x0,t)	Απόκριση σε αρχικές συνθήκες (Μόνο για Χ.Κ)

Στον Πίνακα 14.3, **sys** είναι η περιγραφή του συστήματος στο πεδίο της συχνότητας (Σ.Μ) ή στο πεδίο του χρόνου (Χ.Κ) όπως προκύπτει από τον Πίνακα 14.1, **t** είναι το διάνυσμα του χρόνου και **u** είναι το διάνυσμα της εισόδου (τα διανύσματα **t** και **u** πρέπει να έχουν τις ίδιες διαστάσεις). Τέλος **x0** είναι το διάνυσμα των αρχικών συνθηκών.

Η εντολή **step** μπορεί να συνταχθεί και ως **step(num,den)** ή ισοδύναμα **step(A,B,C,D)**. Το ίδιο ισχύει και για τις υπόλοιπες εντολές του Πίνακα 14.3. Επίσης η εντολή **step** δίνει την

απόκριση σε **μοναδιαία** βηματική είσοδο. Για αποκρίσεις σε βηματικές εισόδους πλάτους α , χρησιμοποιείται **step(α *sys)**.

Παρατήρηση: Για την δημιουργία του σήματος εισόδου στην **lsim**, πολλές φορές χρησιμοποιείται η εντολή:

[u,t]=gensig(type,tau),

Όπου **type** είναι το είδος εισόδου ('sin','square', 'pulse') και **tau** είναι η περίοδος του σήματος εισόδου.

Παράδειγμα 14.2: Να υπολογιστεί η απόκριση του συστήματος του Παραδείγματος 1 σε μοναδιαία αναρριχητική είσοδο.

Στο Σχήμα 14.2 παρουσιάζονται οι απαιτούμενες εντολές.

```

% Define system
num=1;% numerator of the T.F
den=[1 2 1]; % denominator of the T.F
sys=tf(num,den); % system's T.F
% Define simulation time
t=0:0.01:10;
% Define ramp input
a=1;
u=a*t;
% system response to ramp input
lsim(sys,u,t)

```

Σχήμα 14.2. Κώδικας για τον υπολογισμό της αναρριχητικής απόκρισης του συστήματος.

Η απόκριση ενός συστήματος καθορίζεται από τους πόλους και τους μηδενιστές του. Το MATLAB διαθέτει τις εντολές του Πίνακα 14.4 για τον υπολογισμό τους:

Πίνακας 14.4

Πόλοι – Μηδενιστές	
pole(sys)	Πόλοι Συστήματος
zero(sys)	Μηδενιστές Συστήματος
pzmap(sys)	Διάγραμμα Πόλων – Μηδενιστών
minreal(sys)	Απαλοιφή Κοινών Πόλων - Μηδενιστών

Παράδειγμα 14.3: Να υπολογιστεί η συνάρτηση μεταφοράς και στην συνέχεια να βρεθούν οι πόλοι και οι μηδενιστές καθώς και το διάγραμμα τους για το σύστημα που περιγράφεται στον χώρο κατάστασης από τους πίνακες:

$$A = \begin{bmatrix} -3 & -2 \\ 1 & 0 \end{bmatrix}, B = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, C = [1 \quad -1], D = 0$$

```

% Define system's State Space (S.S)
A=[-3 -2; 1 0];
B=[1; 0];
C=[1 -1];
D=0;
% Conversion to Transfer Function
[num,den]=ss2tf(A,B,C,D);
sys=tf(num,den);
% System zeros
midenika=zero(sys)
% System Poles
poloi=pole(sys)
% Pole-zero map
pzmap(sys)

```

Σχήμα 14.3. Κώδικας για το Παράδειγμα 3.

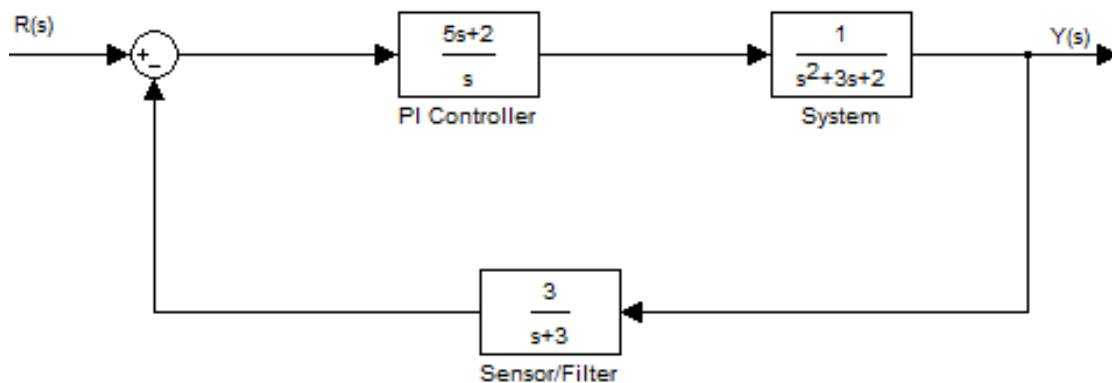
Συνήθως ένα σύστημα αποτελεί μέρος ενός ευρύτερου συνόλου συστημάτων, τα οποία συνδέονται μεταξύ τους. Το MATLAB περιέχει συναρτήσεις που δίνουν τη συνάρτηση μεταφοράς δύο συστημάτων τα οποία συνδέονται μεταξύ τους σε σειρά, παράλληλα, με θετική (ή αρνητική) ανατροφοδότηση. Η σύνταξη αυτών των εντολών παρουσιάζεται στον Πίνακα 14.5.

Πίνακας 14.5

Σύνδεση Συστημάτων	
sys=series(sys1,sys2)	Σύνδεση Συστημάτων σε Σειρά
sys=parallel(sys1,sys2)	Σύνδεση Συστημάτων Παράλληλα
sys=feedback(sys1,sys2,-1)	Σύνδεση Συστημάτων με Αρνητική Ανατροφοδότηση
sys=feedback(sys1,sys2,+1)	Σύνδεση Συστημάτων με Θετική Ανατροφοδότηση

Παρατήρηση: Η εντολή sys=feedback(sys1,sys2,-1) είναι ισοδύναμη με την εντολή sys=feedback(sys1,sys2).

Παράδειγμα 14.4: Να υπολογιστεί η συνάρτηση μεταφοράς του συστήματος κλειστού βρόχου του Σχήματος 14.4.



Σχήμα 14.4. Έλεγχος συστήματος με PI ελεγκτή.

Στο Σχήμα 14.5, παρουσιάζονται οι απαιτούμενες εντολές για τον υπολογισμό της Σ.Μ του συστήματος του Σχήματος 14.4.

```

% Define the controller
num1=[5 2]; % numerator of the T.F
den1=[1 0]; % denominator of the T.F
sys1=tf(num1,den1);
% Define the system
num2=1; % numerator of the T.F
den2=[1 3 2]; % denominator of the T.F
sys2=tf(num2,den2);
% Define the sensor/filter
num3=3; % numerator of the T.F
den3=[1 3]; % denominator of the T.F
sys3=tf(num3,den3);
% Transfer function of the upper part
% of the closed loop system
% (the controller and the system are
%connected in series)
sysol=series(sys1,sys2);
% Transfer function of the closed
% loop system
% (The system sysol and the sensor/filter
% system are connected by negative feedback)
syscl=feedback(sysol,sys3,-1);

```

Σχήμα 14.5. Κώδικας για τον υπολογισμό της Σ.Μ του συστήματος του Σχήματος 14.4.

Για τη μελέτη γραμμικών συστημάτων στο πεδίο της συχνότητας, το Matlab διαθέτει τις εντολές που παρουσιάζονται στον Πίνακα 14.6.

Πίνακας 14.6

Συχνотική απόκριση	
bode(sys)	Διαγράμματα Bode
[Gm,Pm,Wcg,Wcp] = margin(sys)	Περιθώριο Κέρδους – Περιθώριο Φάσης

H=freqresp(sys,w)	Συχνοτική απόκριση
nyquist (sys)	Διαγράμματα nyquist
nichols (sys)	Διαγράμματα nichols

Στον Πίνακα 14.6, **sys** είναι το σύστημα, ορισμένο με κάποια από τις περιγραφές του Πίνακα 14.1, και **w** η συχνότητα σε rad/s. Επίσης **Gm** και **Pm** είναι τα περιθώρια κέρδους και φάσης αντίστοιχα ενώ **Wcg** και **Wcp** είναι οι αντίστοιχες συχνότητες.

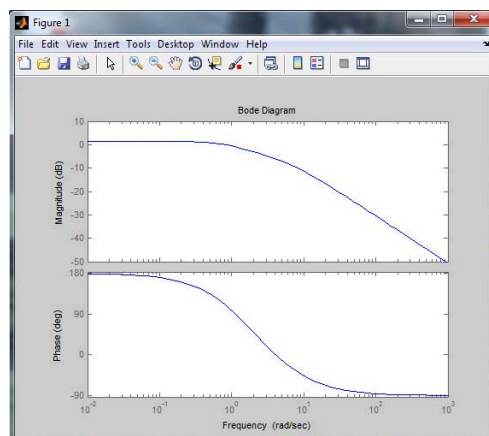
Παράδειγμα 14.5: Να σχεδιαστούν τα διαγράμματα Bode και Nyquist, για ένα σύστημα που έχει πόλους στα σημεία -1 και -5, μηδενιστή στο σημείο 2 και κέρδος $K = 3$.

Στο Σχήμα 14.6, παρουσιάζονται οι απαιτούμενες εντολές.

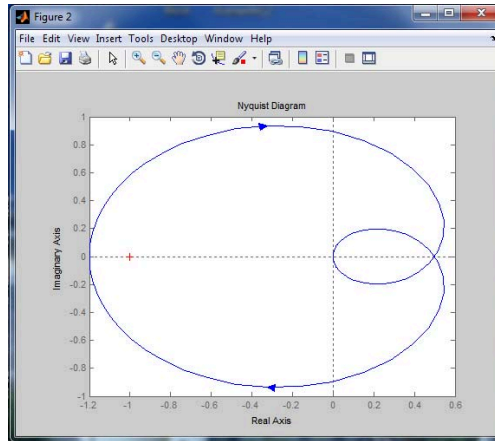
```
% Define the system
Z=2; % System's zeros
P=[-1 -5]; % System's poles
K=3; % System's gain
sys=zpk(Z,P,K); % System's transfer function
% Bode plot
figure(1)
bode(sys)
% Nyquist plot
figure(2)
nyquist(sys)
```

Σχήμα 14.6. Κώδικας για τον υπολογισμό των διαγραμμάτων Bode και Nyquist

Από τον κώδικα του Σχήματος 14.6 προκύπτουν τα παρακάτω διαγράμματα Bode και Nyquist.



Σχήμα 14.7. Διάγραμμα Bode του συστήματος του Παραδείγματος 5.



Σχήμα 14.8. Διάγραμμα Nyquist του συστήματος του Παραδείγματος 5.

Ένα από τα εργαλεία που χρησιμοποιείται στην ανάλυση και στην σχεδίαση συστημάτων αυτομάτου ελέγχου, είναι η μέθοδος του γεωμετρικού τόπου ριζών (γ.τ.ρ). Το Matlab διαθέτει, για τον σκοπό αυτόν, τις εντολές του Πίνακα 14.7.

Πίνακας 14.7

Γεωμετρικός Τόπος Ριζών	
rlocus(sys)	Διάγραμμα τόπου ριζών
[K, poles]=rlocfind(sys,P)	Υπολογισμός Κέρδους και πόλων

Όπου **P** είναι το διάνυσμα των επιθυμητών πόλων.

Παρατήρηση: Η εντολή rlocfind μπορεί να χρησιμοποιηθεί και χωρίς το όρισμα των επιθυμητών πόλων **P**. Σ' αυτή την περίπτωση η επιλογή του επιθυμητού πόλου γίνεται με επιλογή του από το διάγραμμα του γ.τ.ρ.

Παράδειγμα 14.6: Να υπολογιστεί ο γ.τ.ρ του συστήματος με συνάρτηση μεταφοράς:

$$G(s) = \frac{2}{s^3 + 6s^2 + 5s}$$

Στην συνέχεια να υπολογιστεί το κέρδος και οι αντίστοιχοι πόλοι για επιθυμητό πόλο κοντά στο σημείο -6 .

Στο Σχήμα 14.9 παρουσιάζονται οι απαιτούμενες εντολές.

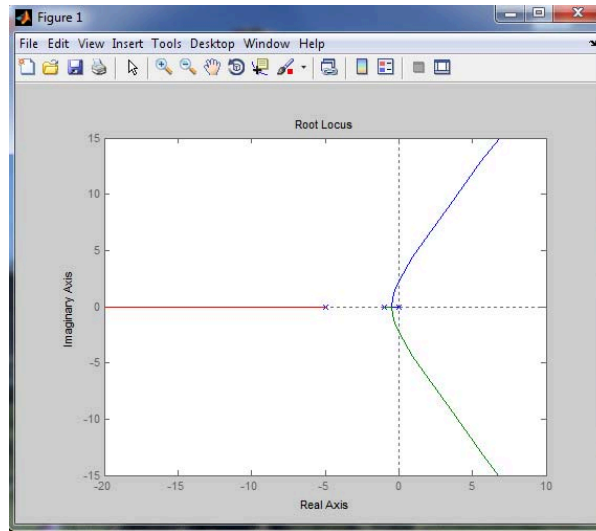
```

% Define system's transfer function
num=2;
den=[1 6 5 0];
sys=tf(num,den);
% Root locus plot
rlocus(sys)
% Desired pole
P=-6;
% Gain K and closed-loop poles for
% which one is near the desired pole
[K,poles] = rlocfind(sys,P);

```

Σχήμα 14.9. Κώδικας για την σχεδίαση γ.τ.ρ καθώς για τον υπολογισμό του κέρδους.

Παρακάτω παρουσιάζονται τα αποτελέσματα που δίνει ο κώδικας.



Σχήμα 14.11. Γεωμετρικός τόπος ριζών για το σύστημα του Παραδείγματος 14.6.

```
K =
    15

poles =
    -6.0000
    0.0000 + 2.2361i
    0.0000 - 2.2361i
```

Σχήμα 14.12. Το κέρδος και οι αντίστοιχοι πόλοι κλειστού βρόχου, για επιθυμητό πόλο -6.

Όπως τα συνεχή συστήματα, έτσι και τα διακριτά συστήματα περιγράφονται είτε με συνάρτηση μεταφοράς (Σ.Μ) είτε στο χώρο κατάστασης (Χ.Κ). Το MATLAB περιέχει τις συναρτήσεις `tf` και `ss` καθώς και τη `zpk` (Πίνακας 14.8) για τις περιγραφές αυτές.

Πίνακας 14.8

Βασικές Περιγραφές Διακριτών Συστημάτων	
<code>sys=tf(num,den,Ts)</code>	Συνάρτηση Μεταφοράς
<code>sys=ss(A,B,C,D,Ts)</code>	Χώρος Κατάστασης
<code>sys=zpk(Z,P,K,Ts)</code>	Μηδενιστές-Πόλοι-Κέρδος Συστήματος

όπου **Ts** είναι η περίοδος δειγματοληψίας. Αν αυτή δεν ορίζεται τότε θέτουμε **Ts = -1**.

Οι αντίστοιχες εντολές για την απόκριση των διακριτών συστημάτων σε διάφορες εισόδους παρουσιάζονται στον Πίνακα 14.9.

Πίνακας 14.9

Απόκριση Διακριτών Συστημάτων	
<code>dstep(num,den)</code>	Βηματική απόκριση
<code>dimpulse(num,den)</code>	Κρουστική απόκριση

<code>dlsim(num,den,u,t)</code>	Απόκριση σε οποιαδήποτε είσοδο
<code>dinitial(A,B,C,D,x0,t)</code>	Απόκριση σε αρχικές συνθήκες (Μόνο για Χ.Κ)

Παράδειγμα 14.7: Να υπολογιστεί η βηματική απόκριση του διακριτού συστήματος:

$$G(z) = \frac{1}{z-0.5}$$

Στο Σχήμα 14.13, δίνονται οι απαιτούμενες εντολές:

```
% Define the numerator and the denominator
% of the system's transfer function
num=1;
den=[1 -0.5];
% Step response of the system
dstep(num,den)
```

Σχήμα 14.13. Εντολές για την βηματική απόκριση διακριτού συστήματος.

Στον ψηφιακό έλεγχο, πολλές φορές, είναι απαραίτητη η μετατροπή ενός συνεχούς συστήματος σε διακριτό (διακριτοποίηση). Για τη διαδικασία αυτή, το Matlab διαθέτει τις εντολές του Πίνακα 14.10.

Πίνακας 14.10

Διακριτοποίηση συστημάτων.	
<code>sysd = c2d(sys,Ts,'method')</code>	Συνεχές σε Διακριτό
<code>[Ad,Bd,Cd,Db] = c2dm(A,B,C,D,Ts,'method')</code>	Συνεχές σε Διακριτό (Χ. Κ.)
<code>[numd,dend] = c2dm(num,den,Ts,'method')</code>	Συνεχές σε Διακριτό (Σ. Μ.)

Όπου **method** είναι η μέθοδος διακριτοποίησης ('zoh', 'foh', 'imp', 'tustin', 'prewarp', 'matched') και **Ts** είναι η περίοδος δειγματοληψίας.

Παρατηρήσεις: Όταν παραλείπεται η παράμετρος **method**, τότε θεωρείται ότι η διακριτοποίηση γίνεται με συγκρατητή μηδενικής τάξης (**zoh**). Η μέθοδος **matched** χρησιμοποιείται μόνο για συστήματα MEME. Επίσης όταν χρησιμοποιείται η μέθοδος **prewarp**, θα πρέπει να περιλαμβάνεται και η κρίσιμη συχνότητα **Wc** (rad/s) ως επιπλέον όρισμα στις εντολές του Πίνακα 14.8, δηλαδή:

$$\text{sysd}=\text{c2d}(\text{sys},\text{Ts},\text{'prewarp'},\text{Wc})$$

Όσο αφορά τις εντολές που αναφέρονται σε μετατροπές περιγραφών (Πίνακας 14.2), πόλους – μηδενιστές (Πίνακας 14.4), σύνδεση συστημάτων (Πίνακας 14.5), συχνотική απόκριση (Πίνακας 14.6) και γεωμετρικό τόπο ριζών (Πίνακας 14.7), ισχύουν και για τα διακριτά συστήματα. Επίσης, στην περίπτωση που το σύστημα έχει προκύψει από διακριτοποίηση, τότε για την απόκριση του μπορούν να χρησιμοποιηθούν οι εντολές του Πίνακα 14.3.

Παράδειγμα 14.8: Έστω το συνεχές σύστημα με συνάρτηση μεταφοράς:

$$G(s) = \frac{1}{s(s+0.5)}$$

Για τον έλεγχο του συστήματος έχει προταθεί ο παρακάτω PD ελεγκτής συνεχούς χρόνου:

$$G_c(s)=0.5+2s$$

Να διακριτοποιηθεί ο παραπάνω ελεγκτής με την μέθοδο Tustin, επιλέγοντας περίοδο δειγματοληψίας $T=0.1s$ και να σχεδιαστεί η βηματική απόκριση του κλειστού συστήματος.

Στο Σχήμα 14.14, παρουσιάζεται ο απαιτούμενος κώδικας Matlab:

```

% Define the system
num_sys=1;
den_sys=conv([1 0],[1 0.5]);
sys=tf(num_sys,den_sys);
% Define the controller
num_contr=[2 0.5];
den_contr=1;
contr=tf(num_contr,den_contr);
% Define the sample period
Ts=0.1;
% Discretization of the controller
contrd=c2d(contr,Ts,'tustin')
% Discretization of the system
sysd=c2d(sys,Ts,'zoh');
% Closed loop system
sysol=series(contrd,sysd);
syscl=feedback(sysol,1,-1);
% Step response of closed loop system
step(syscl)

```

Σχήμα 14.14. Κώδικας για την διακριτοποίηση και την απόκριση του κλειστού συστήματος.

Ο κώδικας του Σχήματος 14.14 δίνει την συνάρτηση μεταφοράς του διακριτοποιημένου ελεγκτή και την βηματική απόκριση του κλειστού συστήματος:

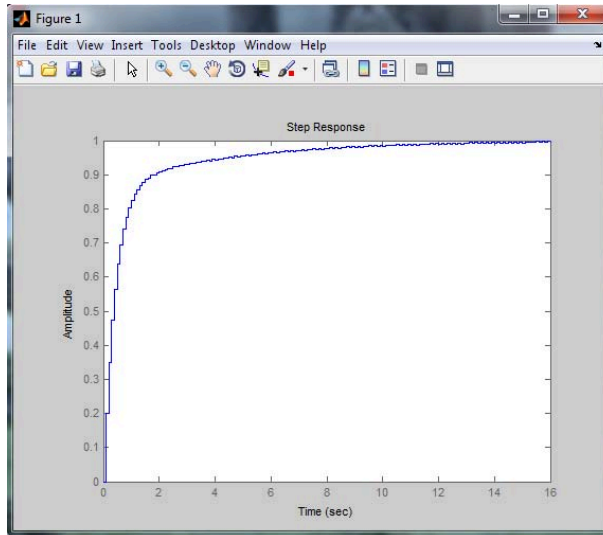
```

Transfer function:
40.5 z - 39.5
-----
z + 1

Sampling time: 0.1

```

Σχήμα 14.15. Συνάρτηση μεταφοράς του διακριτοποιημένου ελεγκτή του Παραδείγματος 14.8.



Σχήμα 14.16. Βηματική απόκριση του κλειστού συστήματος του Παραδείγματος 14.8.

15. SIMULINK

Το SIMULINK είναι ένα πρόγραμμα μέσα στο MATLAB που επιτρέπει την προσομοίωση δυναμικών συστημάτων χρησιμοποιώντας δομικά διαγράμματα (blocks). Τα διαγράμματα αυτά παρέχονται από την βιβλιοθήκη του SIMULINK οργανωμένα σε διάφορες κατηγορίες. Τοποθετώντας τα διαγράμματα αυτά σε ένα αρχείο και συνδέοντας τα κατάλληλα μπορούμε να προσομοιώσουμε οποιοδήποτε σύστημα. Για καλύτερη κατανόηση της λειτουργίας του SIMULINK, στην συνέχεια παρουσιάζονται τρία χαρακτηριστικά παραδείγματα.

Παράδειγμα 15.1: Ένα ερευνητικό υποβρύχιο κινείται κάτω από το επίπεδο του ωκεανού και παράλληλα σε αυτό. Η ισοδύναμη μάζα του όταν είναι βυθισμένο είναι $M=2400\text{kg}$. Ο συντελεστής οπισθέλκουσας είναι $B=320\text{kg/s}$ και η δύναμη προώθησης του είναι σταθερή ίση με $F=1130\text{N}$. Να σχεδιαστεί η ταχύτητα $v(t)$ του υποβρυχίου ως συνάρτηση του χρόνου εάν η αρχική ταχύτητα του είναι $v(0)=0\text{ m/s}$.

Καταρχήν, το υποβρύχιο μπορεί να μοντελοποιηθεί ως μια μάζα M στην οποία ασκείται η δύναμη προώθησης F καθώς και η οπισθέλκουσα δύναμη $-Bv(t)$. Για το σύστημα αυτό ισχύει ο δεύτερος νόμος του Newton. Έτσι:

$$\dot{v} + \frac{B}{M}v = \frac{F}{M} \quad (15.1)$$

Η Εξ. (15.1) είναι μια γραμμική διαφορική εξίσωση πρώτης τάξης, η οποία μπορεί να λυθεί αναλυτικά και να μας δώσει την ταχύτητα ως συνάρτηση του χρόνου:

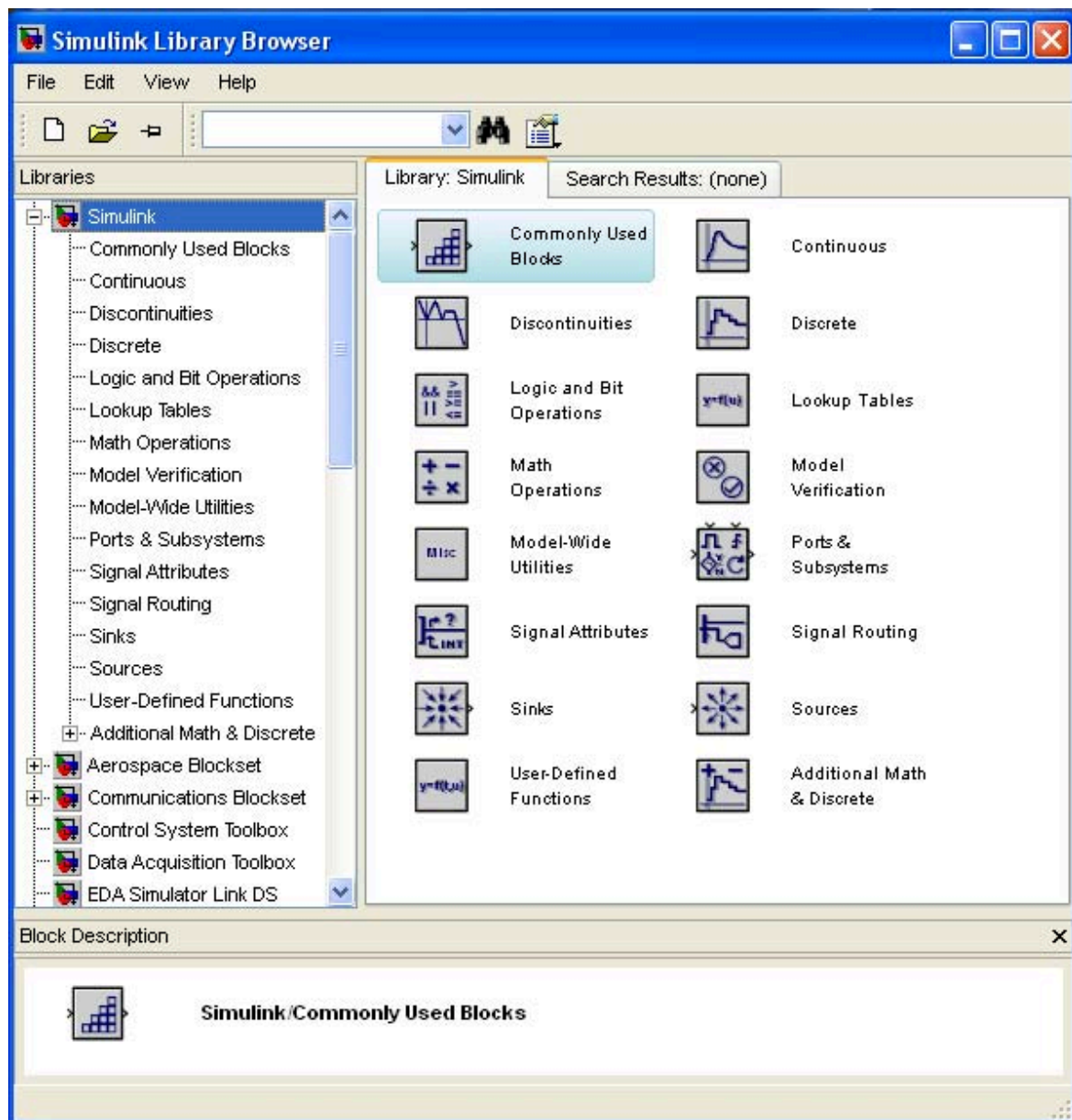
$$v(t) = \frac{F}{B}(1 - e^{-(B/M)t}) \quad (15.2)$$

Στην συνέχεια θα επιλύσουμε την ίδια διαφορική εξίσωση (αριθμητικά όμως) μέσω του SIMULINK.

Εισάγετε **simulink** στο Command Window

```
>>simulink
```

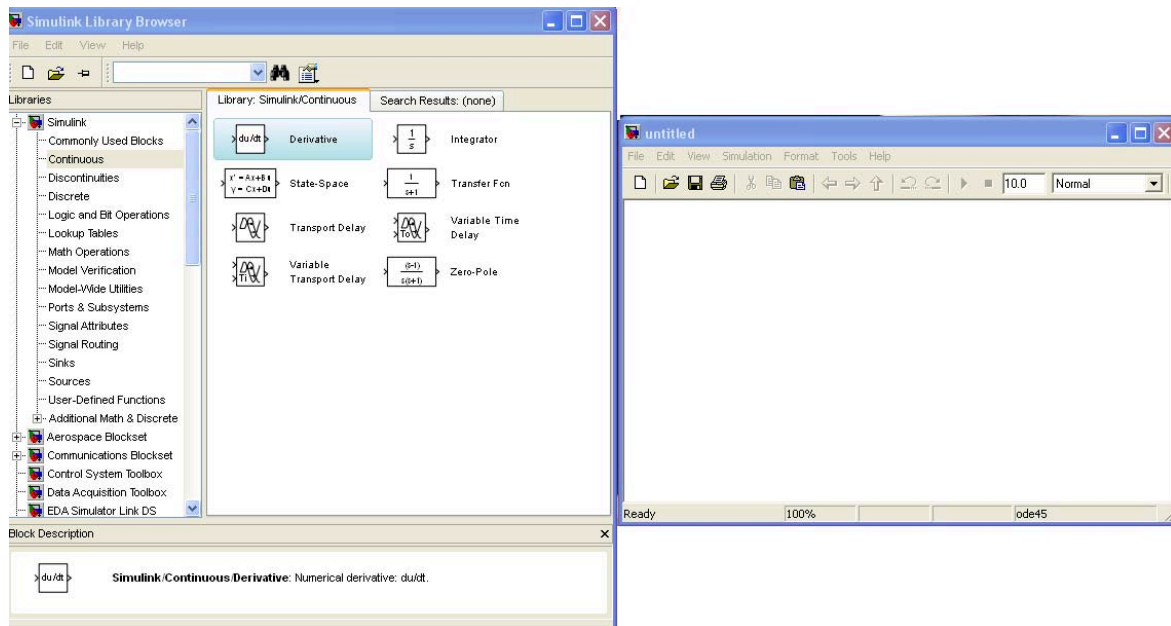

ή εναλλακτικά πατήστε το κουμπί που βρίσκεται στην πάνω μπάρα. Θα εμφανισθεί ένα παράθυρο που περιέχει την βιβλιοθήκη του SIMULINK, η οποία περιέχει διάφορες κατηγορίες (π.χ. Continuous, Discrete κ.λπ.), όπως δείχνει το Σχήμα 15.1:



Σχήμα 15.1. Βιβλιοθήκη του Simulink.

Επιλέγοντας “**File**→**New**→**Model**” ανοίγει ένα νέο κενό αρχείο με το όνομα untitled στο οποίο θα δημιουργηθεί το σύστημα.

Πατώντας μια φορά σε οποιοδήποτε κατηγορία, μια λίστα από διαγράμματα (blocks) θα εμφανισθεί σε ξεχωριστό παράθυρο. Αυτά τα διαγράμματα μπορούν να εισαχθούν στο αρχείο untitled όπου θα πραγματοποιηθεί η προσομοίωση. Η εισαγωγή τους μπορεί να γίνει είτε με αντιγραφή – επικόλληση είτε με απλό σύρσιμο τους στο αρχείο. Στο Σχήμα 15.2 φαίνονται τα blocks που αντιστοιχούν στην κατηγορία **Continuous** μαζί με το αρχείο untitled.




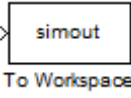




Σχήμα 15.2. Αντιπροσωπευτικά διαγράμματα και αρχείο προσομοίωσης.

Στον Πίνακα 15.1 παρουσιάζονται μερικά βασικά διαγράμματα τα οποία θα χρησιμοποιηθούν στα επόμενα παραδείγματα.

Πίνακας 15.1

Κατηγορία	Continuous		
	Όνομα	Σχήμα	Λειτουργία
	Transfer Function	 Transfer Fcn	Περιγραφή Συστήματος με Συνάρτηση Μεταφοράς
	State Space	 State-Space	Περιγραφή Συστήματος στον Χώρο Κατάστασης
	Integrator	 Integrator	Ολοκλήρωση Σήματος Εισόδου
	Derivative	 Derivative	Παραγωγή Σήματος Εισόδου
Κατηγορία	Sources		
	Όνομα	Σχήμα	Λειτουργία
	Step	 Step	Βηματική Είσοδος

	Ramp		Αναρριχητική Είσοδος
	Clock		Εισαγωγή Χρόνου Προσομοίωσης
Κατηγορία	Sinks		
	Όνομα	Σχήμα	Λειτουργία
	Scope		Απεικόνιση του Σήματος Εισόδου Συναρτήσει του Χρόνου
	To Workspace		Αποθήκευση Τιμών του Σήματος Εισόδου στον Χώρο Εργασίας
Κατηγορία	Math operations		
	Όνομα	Σχήμα	Λειτουργία
	Gain		Πολλαπλασιασμός Σήματος Εισόδου με μια Σταθερά
	Sum		Άθροιση Σημάτων Εισόδου

Στο παράδειγμα αυτό, η ταχύτητα του υποβρυχίου είναι η μεταβλητή εξόδου και η προωθητική δύναμη είναι η μεταβλητή εισόδου. Η συνάρτηση μεταφοράς του συστήματος μπορεί να υπολογιστεί από την διαφορική εξίσωση (15.1) χρησιμοποιώντας τις ιδιότητες του μετασχηματισμού Laplace. Έτσι:

$$G(s) = \frac{v(s)}{F(s)} = \frac{1}{Ms + B} \quad (15.3)$$

Και εφόσον η δύναμη F είναι σταθερή ξεκινώντας από $t=0$ s, μπορεί να θεωρηθεί ως βηματική είσοδος:

$$F(s) = \frac{F}{s} \quad (15.4)$$

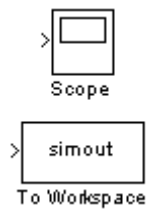
Για την προσομοίωση του συστήματος επιλέγονται και εισάγονται στο untitled αρχείο τα παρακάτω διαγράμματα:

A) Από την κατηγορία **Sources** επιλέγουμε τα διαγράμματα **Step** και **Clock**. Το διάγραμμα **Step** αντιπροσωπεύει την σταθερή (βηματική) δύναμη προώθησης. Ενώ το διάγραμμα **Clock** εισάγει τον χρόνο προσομοίωσης.



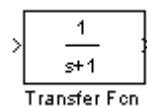
Σχήμα 15.3. Τα διαγράμματα **Step** και **Clock**.

Β) Από την κατηγορία **Sinks** επιλέγονται τα διαγράμματα **Scope** και **To Workspace**. Το **Scope** επιτρέπει την καταγραφή της μεταβολής της εξόδου συναρτήσεως του χρόνου ενώ το διάγραμμα **To Workspace** αποθηκεύει τις τιμές της εξόδου σε ένα διάνυσμα με το όνομα **simout** στον χώρο εργασίας (workspace) του MATLAB και το οποίο μπορεί να χρησιμοποιηθεί για περαιτέρω ανάλυση.



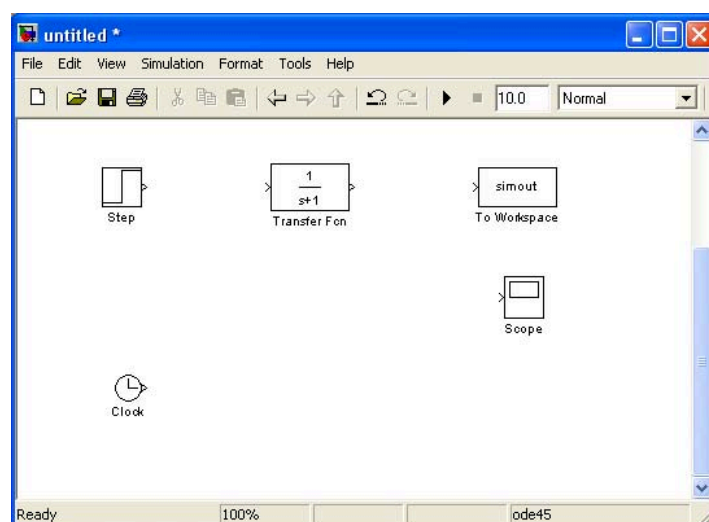
Σχήμα 15.4. Τα διαγράμματα **Scope** και **To Workspace**.

Γ) Από την κατηγορία **Continuous** επιλέγετε το διάγραμμα **Transfer Fcn**. Αυτό επιτρέπει την εισαγωγή της συνάρτησης μεταφοράς που υπολογίσθηκε προηγουμένως.



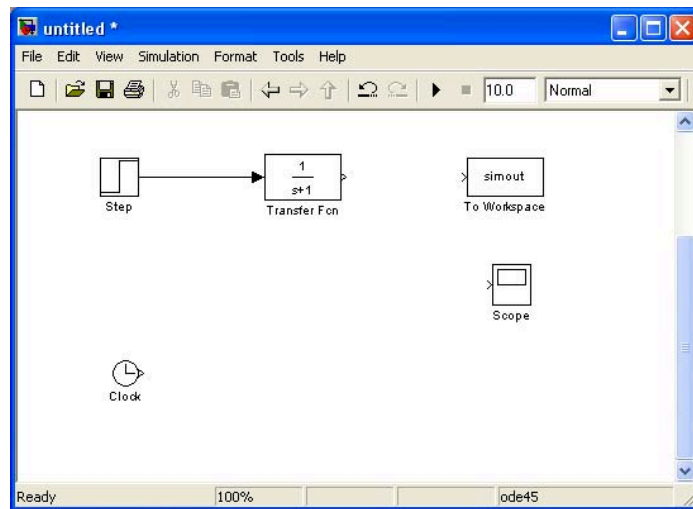
Σχήμα 15.5. Το διάγραμμα **Transfer Fcn**.

Όπως προαναφέρθηκε, τα διαγράμματα αυτά μπορούν να μεταφερθούν στο αρχείο είτε με αντιγραφή – επικόλληση είτε με σύρσιμο τους από την βιβλιοθήκη στο αρχείο. Με τον τρόπο αυτό προκύπτει το ακόλουθο αρχείο.



Σχήμα 15.6. Τα απαιτούμενα διαγράμματα συγκεντρωμένα στο αρχείο untitled.

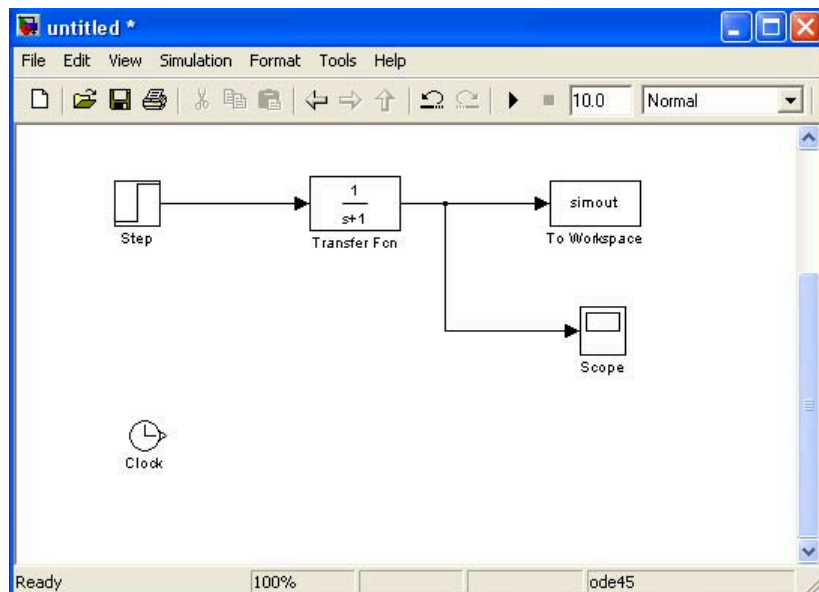
Έπειτα, ενώνουμε τα διαγράμματα **Step** και **Transfer Fcn**. Αυτό γίνεται με σύρσιμο από το βελάκι εξόδου (>) του **Step** και κατάληξη στο βελάκι εισόδου (>) του **Transfer Fcn**. Εάν η σύνδεση είναι επιτυχής, θα σχηματιστεί ένα μαύρο βέλος που ενώνει τα δυο αυτά διαγράμματα όπως φαίνεται στο Σχήμα 15.7.



Σχήμα 15.7. Σύνδεση των διαγράμματα **Step** και **Transfer Fcn**.

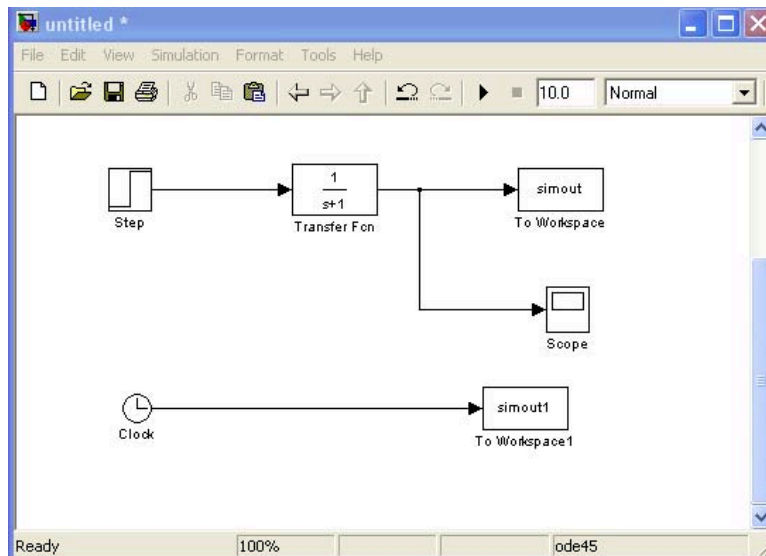
Με τον ίδιο τρόπο συνδέονται και τα υπόλοιπα διαγράμματα σύμφωνα με το Σχήμα 15.8.

Αξίζει να σημειωθεί ότι η σύνδεση του διαγράμματος **Scope**, γίνεται με σύρσιμο από το βελάκι εισόδου (>) του **Scope** και κατάληξη στο βέλος που συνδέει τα διαγράμματα **Transfer Fcn** και **To Workspace**.



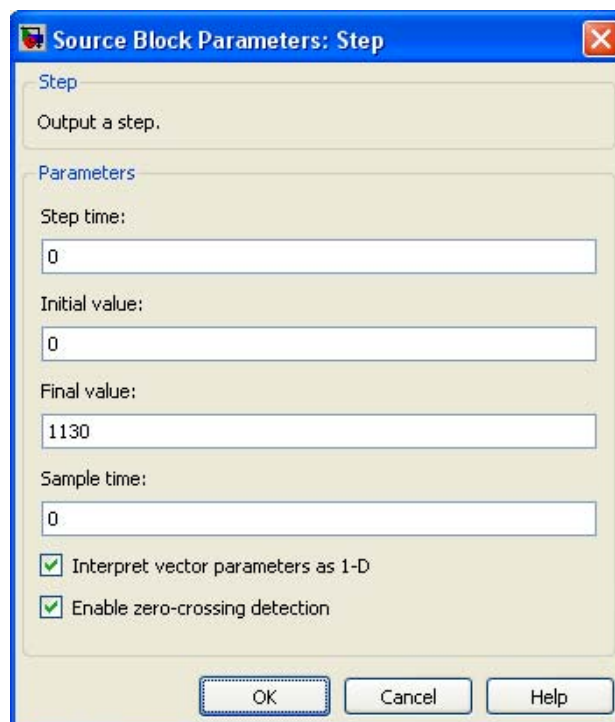
Σχήμα 15.8. Σύνδεση και των υπόλοιπων διαγραμμάτων.

Έπειτα, με αντιγραφή – επικόλληση του διαγράμματος **To Workspace** στο αρχείο, δημιουργούμε το διάγραμμα **To Workspace1** το οποίο συνδέεται με το διάγραμμα **Clock**.



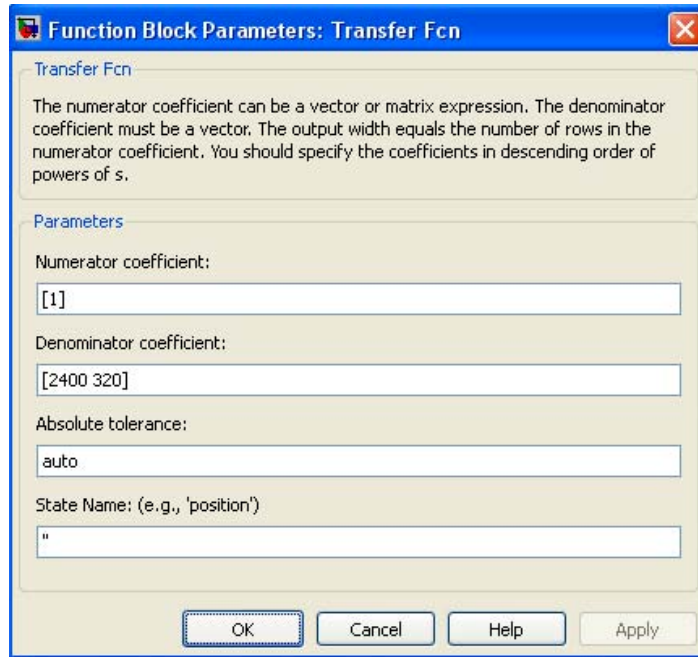
Σχήμα 15.9. Δημιουργία του διαγράμματος **To Workspace1** και σύνδεση του με το διάγραμμα **Clock**.

Τώρα πρέπει να εισαχθούν στα διαγράμματα οι κατάλληλες παράμετροι που αντιστοιχούν στο υποβρύχιο. Πατώντας δύο φορές πάνω στο διάγραμμα **Step** αλλάζουμε την παράμετρο **Step time**: σε 0 και την παράμετρο **Final value**: σε 1130 και πατάμε OK.



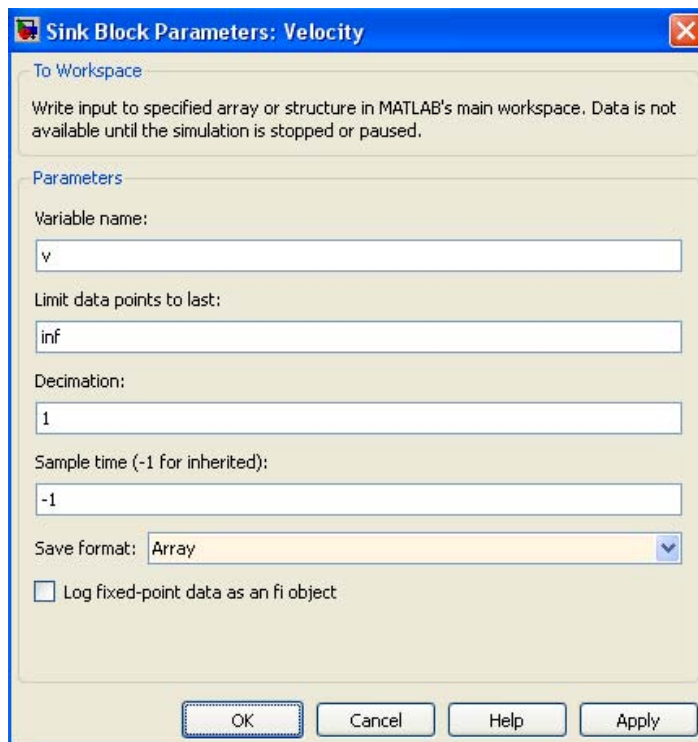
Σχήμα 15.10. Εισαγωγή παραμέτρων του διαγράμματος **Step**.

Με τον ίδιο τρόπο στο διάγραμμα **Transfer Fcn** αλλάζουμε την παράμετρο **Denominator**: βάζοντας [2400 320] και πατάμε Apply και μετά OK.



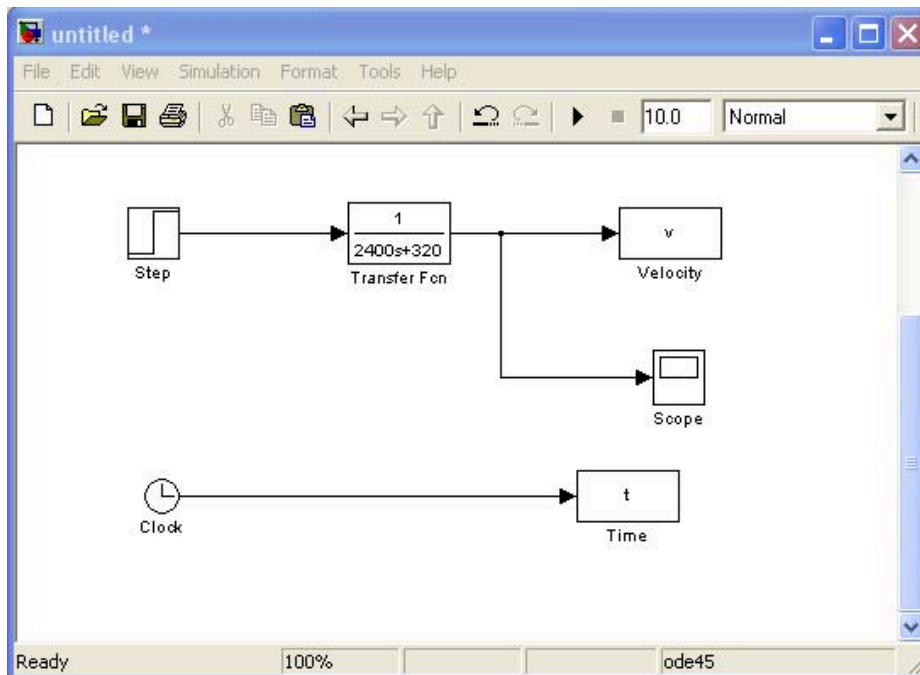
Σχήμα 15.11. Εισαγωγή παραμέτρων του διαγράμματος *Transfer Fcn*.

Στο διάγραμμα *To Workspace* αλλάζουμε τη παράμετρο **Variable name**: σε *v* (αντίστοιχα του *To Workspace1* σε *t*) και την παράμετρο **Save format** σε *Array* και πατάμε *Apply* και μετά *OK*. Έτσι, οι τιμές της ταχύτητας αποθηκεύονται σε ένα διάνυσμα με το όνομα *v* και του χρόνου σε ένα αντίστοιχο με το όνομα *t*.



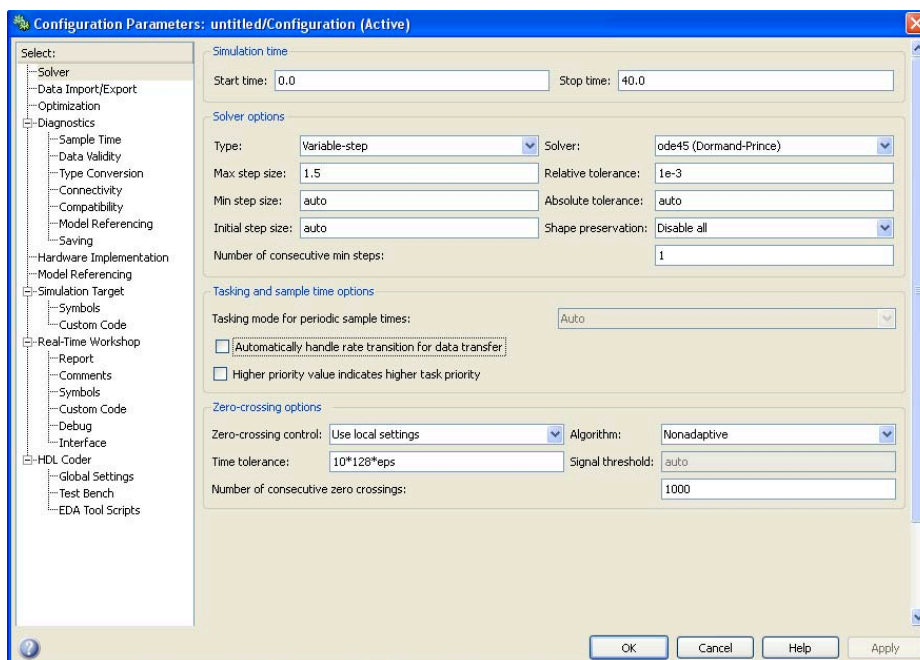
Σχήμα 15.12. Εισαγωγή παραμέτρων του διαγράμματος *To Workspace*.

Επίσης, μπορούμε να αλλάξουμε και τις ετικέτες των διαγραμμάτων. Πατώντας, λοιπόν, μια φορά στη λέξη “To Workspace” την αλλάζουμε σε “Velocity”. Ομοίως, αλλάζουμε το “To Workspace1” σε “Time”. Τελικά θα προκύψει το αρχείο του Σχήματος 15.13:



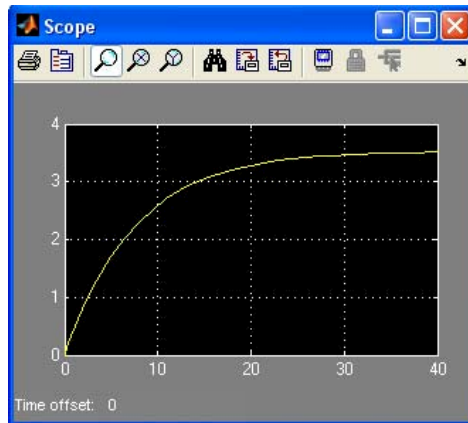
Σχήμα 15.13. Τελική μορφή του αρχείου.

Προτού ξεκινήσει η προσομοίωση, θα πρέπει να οριστούν οι παράμετροι της (π.χ. χρόνος προσομοίωσης, μέθοδος ολοκλήρωσης, βήμα ολοκλήρωσης κ.λπ.). Αυτές επιλέγονται πατώντας “**Simulation** → **Configuration Parameters**” όπως φαίνεται και στο Σχήμα 15.14.

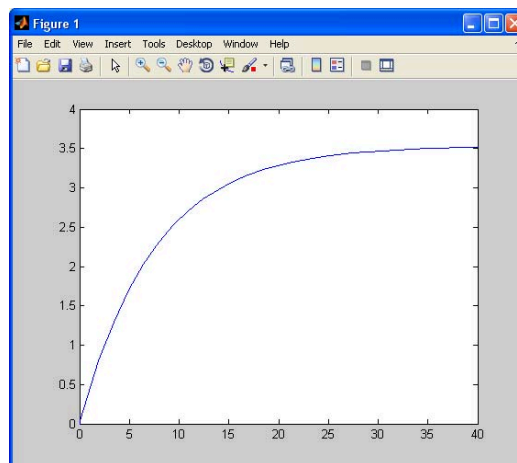


Σχήμα 15.14. Εισαγωγή παραμέτρων προσομοίωσης.

Για το παράδειγμα μας, αλλάζουμε την παράμετρο **Stop Time**: σε 40, την **Max Step Size**: σε 1.5 και μετά πατάμε Apply και OK. Τις περισσότερες φορές ο αλγόριθμος επίλυσης των διαφορικών εξισώσεων του μοντέλου (solver), ode45 είναι ο πιο κατάλληλος. Σε άλλη περίπτωση (π.χ. stiff problems) πρέπει να επιλεγθεί διαφορετικός αλγόριθμος επίλυσης (π.χ. ode15s). Για την έναρξη της προσομοίωσης επιλέγουμε **Simulation** και μετά **Start** (ή εναλλακτικά πατάμε το κουμπί play). Μετά το πέρας της προσομοίωσης πατώντας δυο φορές το διάγραμμα **Scope**, λαμβάνουμε την ταχύτητα του υποβρυχίου συναρτήσει του χρόνου όπως φαίνεται στο Σχήμα 15.15. Το ίδιο αποτέλεσμα θα λάβουμε, Σχήμα 15.16, εάν εισάγουμε την εντολή **plot(t,v)** στο Command Window.



Σχήμα 15.15. Η απόκριση της ταχύτητας όπως φαίνεται από το διάγραμμα **Scope**.



Σχήμα 15.16. Η απόκριση της ταχύτητας όπως προκύπτει από την εντολή **plot**.

Για να αποθηκεύσουμε το αρχείο πατάμε **File** και μετά **Save As...** και πληκτρολογούμε το όνομα του αρχείου (π.χ. **submarine**) αντί του **untitled**.

Παρατήρηση: Οι τιμές των παραμέτρων προσομοίωσης δεν επιλέχθηκαν αυθαίρετα αλλά με βάση έναν γενικό κανόνα που ισχύει για τα συστήματα πρώτης τάξης. Έτσι, η παράμετρος **Stop Time**: επιλέχθηκε ίση με πέντε σταθερές χρόνου και η παράμετρος **Max Step Size**: επιλέχθηκε ίση με το ένα-πέμπτο της σταθεράς χρόνου. Η σταθερά χρόνου του υποβρυχίου ισούται με $\tau=M/B=7.5$ s.

Παράδειγμα 15.2: Να λυθεί το Παράδειγμα 1 για την περίπτωση που η αρχική ταχύτητα του υποβρυχίου είναι $v(0)=2$ m/s.

Η μεθοδολογία που αναπτύχθηκε προηγουμένως βασίστηκε στην συνάρτηση μεταφοράς του συστήματος. Όπως είναι γνωστό, όμως, η συνάρτηση μεταφοράς ορίζεται για μηδενικές

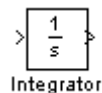
αρχικές συνθήκες. Στην συνέχεια θα δείξουμε πως επιλύεται αριθμητικά, μέσω Simulink, μια γραμμική διαφορική εξίσωση με **μη μηδενικές** αρχικές συνθήκες.

Η Εξ. (1) μπορεί να γραφεί ως:

$$\dot{v} = -\frac{B}{M}v + \frac{F}{M} \quad (15.5)$$

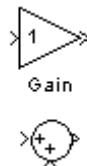
Για την επίλυση της Εξ. (15.5) στο Simulink απαιτούνται τα παρακάτω διαγράμματα:

A) Από την κατηγορία **Continuous** επιλέγουμε το διάγραμμα **Integrator**. Το διάγραμμα αυτό χρησιμοποιείται για την ολοκλήρωση ενός σήματος. Στην περίπτωση μας χρησιμοποιείται για τη δημιουργία της ταχύτητας v από την επιτάχυνση \dot{v} .



Σχήμα 15.17. Το διάγραμμα **Integrator**.

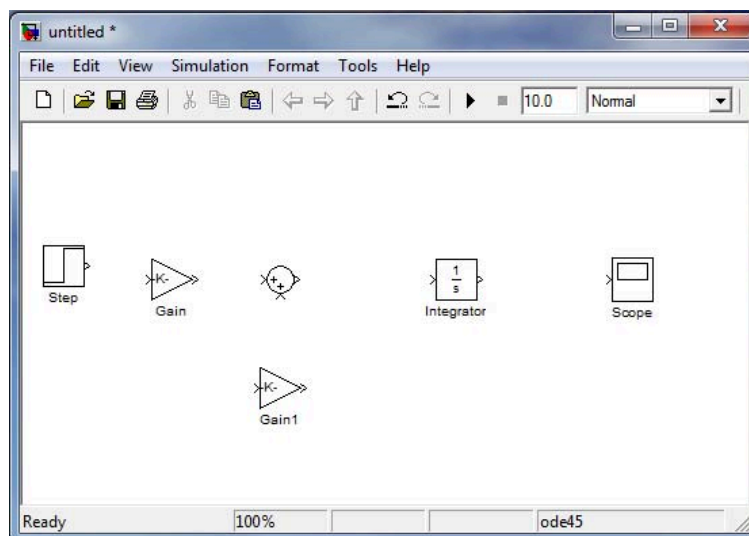
B) Από την κατηγορία **Math operations** επιλέγονται τα διαγράμματα **Gain** και **Sum**. Το διάγραμμα **Gain** πολλαπλασιάζει την είσοδο του με μια σταθερά. Η είσοδος μπορεί να είναι βαθμωτό ή διανυσματικό σήμα και η τιμή της σταθεράς μπορεί να είναι βαθμωτή ή διανυσματική. Εδώ χρησιμοποιείται για τον σχηματισμό των συντελεστών της εξίσωσης. Το διάγραμμα **Sum** χρησιμοποιείται για άθροιση σημάτων.



Σχήμα 15.18. Τα διαγράμματα **Gain** και **Sum**.

Γ) Επίσης τα διαγράμματα **Step** και **Scope** που χρησιμοποιηθήκαν στο Παράδειγμα 15.1.

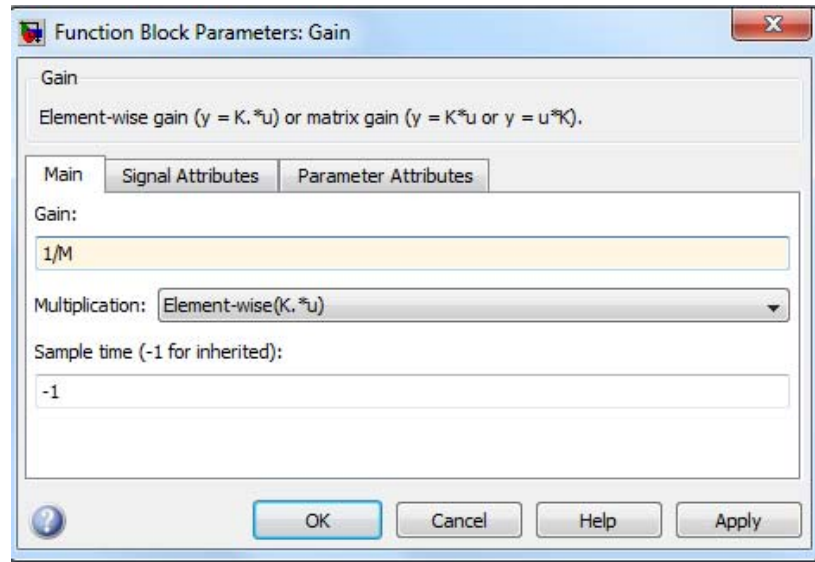
Εισάγουμε τα παραπάνω διαγράμματα σε ένα νέο αρχείο, οπότε προκύπτει το Σχήμα 15.19.



Σχήμα 15.19. Το αρχείο με τα απαιτούμενα διαγράμματα.

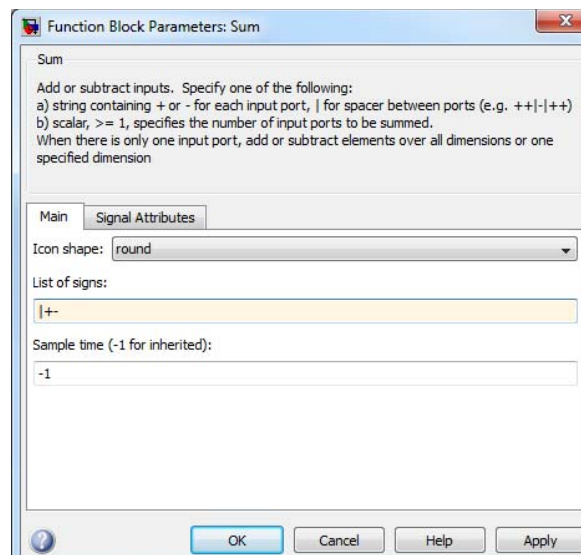
Τώρα πρέπει να εισαχθούν στα διαγράμματα οι κατάλληλες παράμετροι που αντιστοιχούν στη διαφορική εξίσωση. Πατώντας δύο φορές πάνω στο διάγραμμα **Gain** αλλάζουμε την παράμετρο **Gain**: σε $1/M$ ενώ η παράμετρος **Multiplication**: παραμένει σε Element-wise ($K.*u$). Αντίστοιχα στο διάγραμμα **Gain1** αλλάζουμε την παράμετρο **Gain**: σε B/M . Τέλος πατάμε Apply και μετά OK.

Προσοχή: Οι μεταβλητές B και M , θα πρέπει να έχουν ορισθεί πριν την προσομοίωση διαφορετικά θα πρέπει να εισαχθούν στα διαγράμματα οι αντίστοιχες αριθμητικές τους τιμές.



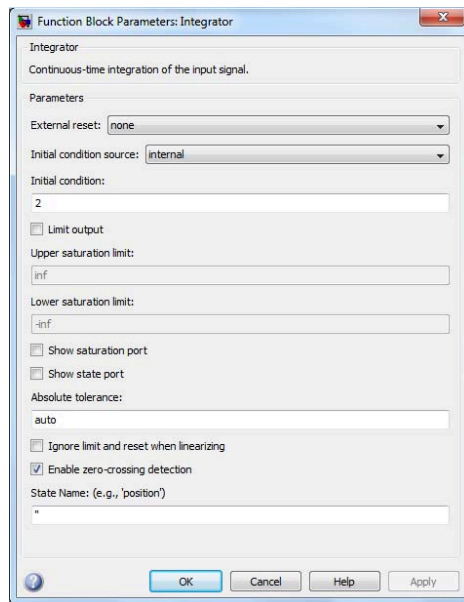
Σχήμα 15.20. Εισαγωγή παραμέτρων του διαγράμματος **Gain**.

Πατώντας δύο φορές πάνω στο διάγραμμα **Sum** αλλάζουμε την παράμετρο **List of signs**: σε $+ -$ και πατάμε Apply και μετά OK.



Σχήμα 15.21. Εισαγωγή παραμέτρων του διαγράμματος **Sum**.

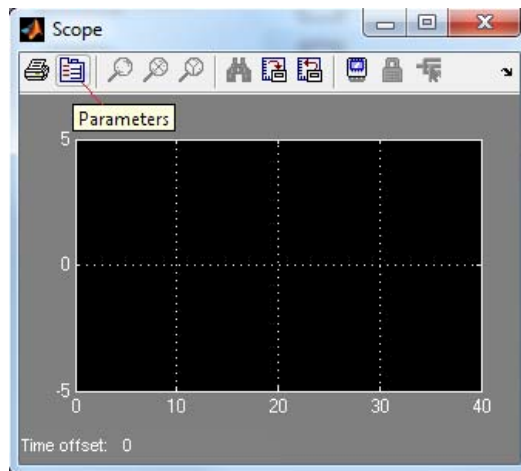
Οι αρχικές συνθήκες του συστήματος εισάγονται μέσω του διαγράμματος **Integrator**. Έτσι πατώντας δύο φορές πάνω του, αλλάζουμε την παράμετρο **Initial condition**: σε 2 και πατάμε Apply και μετά OK.



Σχήμα 22. Εισαγωγή παραμέτρων του διαγράμματος **Integrator**.

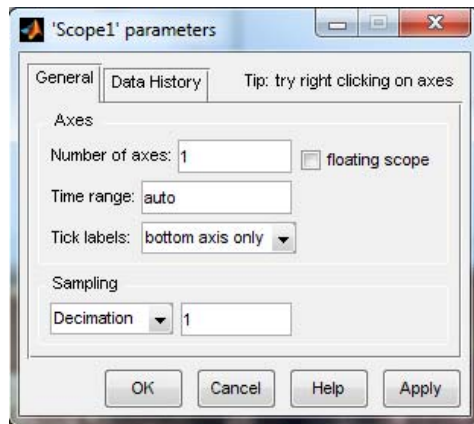
Στο παράδειγμα 15.1, χρησιμοποιήθηκε το διάγραμμα **To Workspace** για την αποθήκευση των τιμών του χρόνου t και της ταχύτητας v στον χώρο εργασίας και την μετέπειτα χρήση τους. Το ίδιο μπορεί και να γίνει μέσω του διαγράμματος **Scope**.

Αρχικά πατώντας μια φορά στο διάγραμμα **Scope** προκύπτει:



Σχήμα 15.23. Αποθήκευση μεταβλητών μέσω του διαγράμματος **Scope**.

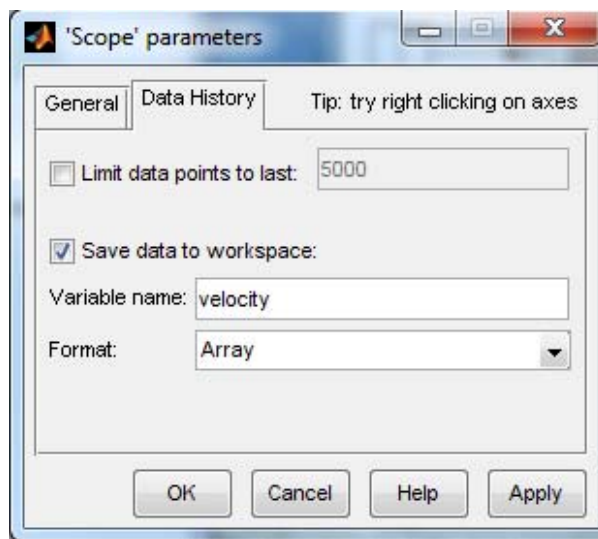
Έπειτα πατάμε το κουμπί Parameters όπως δείχνει το Σχήμα 15.23. Προκύπτει:



Σχήμα 15.24. Αποθήκευση μεταβλητών μέσω του διαγράμματος **Scope**.

Επιλέγοντας **Data History** και στη συνέχεια **Save data to workspace** αλλάζουμε τη μεταβλητή **Variable name:** σε **velocity** τη μεταβλητή **Format:** σε **Array**. Στην πρώτη στήλη του πίνακα με το όνομα **velocity** θα αποθηκευτούν οι τιμές του χρόνου προσομοίωσης και στις υπόλοιπες οι τιμές των εισόδων του διαγράμματος **Scope** (εδώ της ταχύτητας).

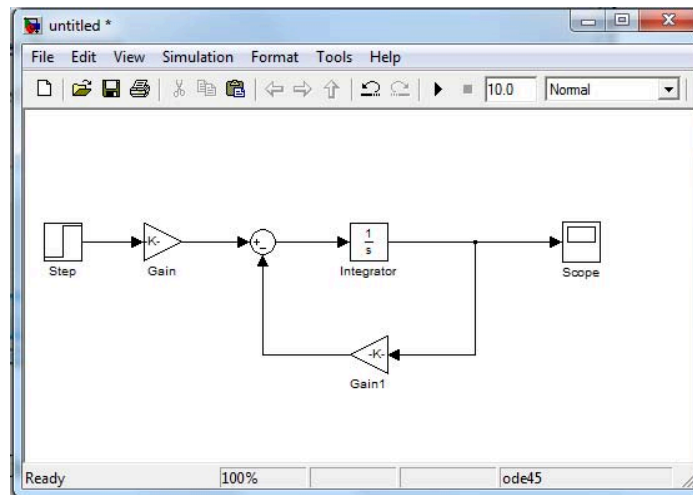
Επίσης αποεπιλέγουμε το **Limit data points to last:**, για να μην υπάρχει περιορισμός στο πλήθος των τιμών των μεταβλητών που μπορούν να αποθηκευτούν, όπως φαίνεται στο Σχήμα 15.25.



Σχήμα 15.25. Αποθήκευση μεταβλητών μέσω του διαγράμματος **Scope**.

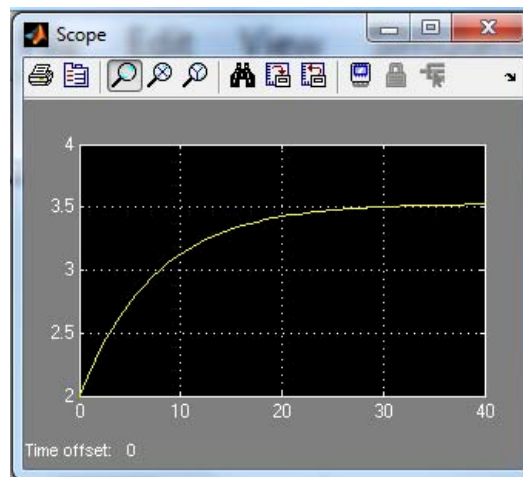
Στην συνέχεια συνδέουμε τα διαγράμματα όπως φαίνεται στο Σχήμα 15.26. Η αναστροφή του διαγράμματος **Gain1** έγινε πατώντας δεξί κλικ στο **Gain1** και επιλέγοντας **Format→Flip Block**.

Παρατηρήστε ότι το σήμα εξόδου του αθροιστή (διάγραμμα **Sum**), δίνει την επιτάχυνση όπως προκύπτει από την Εξ. (15.5).



Σχήμα 15.26. Τελική μορφή του αρχείου.

Επιλέγοντας όπως και στο παράδειγμα 15.1 τις παραμέτρους επίλυσης και τρέχοντας το αρχείο προκύπτει η απόκριση της ταχύτητας του υποβρυχίου:



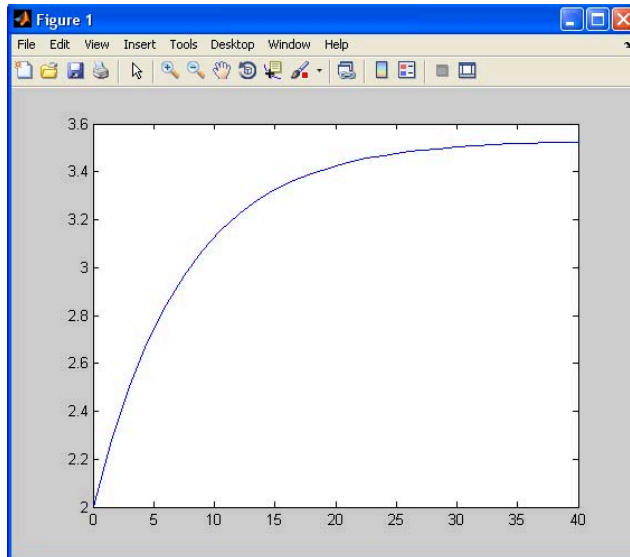
Σχήμα 15.27. Η απόκριση της ταχύτητας όπως φαίνεται από το διάγραμμα **Scope**.

Μέσω του διαγράμματος **Scope**, έχουν αποθηκευτεί οι τιμές του χρόνου και της ταχύτητας σε έναν πίνακα με το όνομα **velocity**. Η πρώτη στήλη του πίνακα περιέχει τις τιμές του χρόνου **t** ενώ η δεύτερη στήλη περιέχει τις τιμές της ταχύτητας **v**.

Έτσι πληκτρολογώντας στο Command Window:

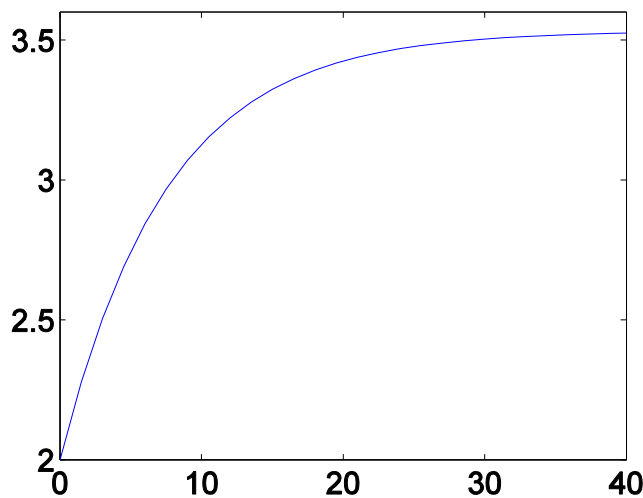
```
>> t=velocity(:,1);
>> v=velocity(:,2);
>> plot(t,v)
```

Προκύπτει:



Σχήμα 15.28α. Η απόκριση της ταχύτητας όπως προκύπτει από την εντολή **plot**.

Στο σημείο αυτό θα πρέπει να αναφερθεί πως σε ένα αρχείο Word, η απόκριση δεν εισάγεται με τη μορφή που έχει στο Σχήμα 15.28α αλλά ως κάποιο αρχείο JPEG, Bitmap, Eps κ.λπ. όπως φαίνεται στο Σχήμα 15.28β. Η διαδικασία εισαγωγής εικόνας σε αρχείο word παρουσιάζεται στο Παράρτημα Α.



Σχήμα 15.28β. Η απόκριση της ταχύτητας σε μορφή Eps.

Παράδειγμα 15.3: Να λυθεί το Παράδειγμα 15.1 για την περίπτωση που η οπισθέλκουσα δύναμη είναι της μορφής $-Bv^2$ και η αρχική ταχύτητα του υποβρυχίου είναι $v(0)=2$ m/s.

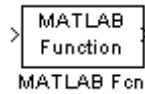
Στην περίπτωση αυτή, η εξίσωση κίνησης του υποβρυχίου είναι:

$$\dot{v} + \frac{B}{M} v^2 = \frac{F}{M} \quad (15.6)$$

Η Εξ. (15.6) είναι **μη γραμμική** διαφορική εξίσωση. Στην συνέχεια, θα παρουσιαστεί ένας τρόπος αριθμητική της επίλυσης μέσω του Simulink.

Θα χρησιμοποιηθούν τα παρακάτω διαγράμματα:

A) Από την κατηγορία **User-defined-functions** επιλέγουμε το διάγραμμα **MATLAB Fcn**. Το διάγραμμα αυτό εφαρμόζει μια συνάρτηση χρησιμοποιώντας κώδικα Matlab. Μπορεί να δεχτεί διανυσματικές εισόδους και να δώσει αντίστοιχα διανυσματικές εξόδους.



Σχήμα 15.29. Το διάγραμμα **MATLAB Fcn**.

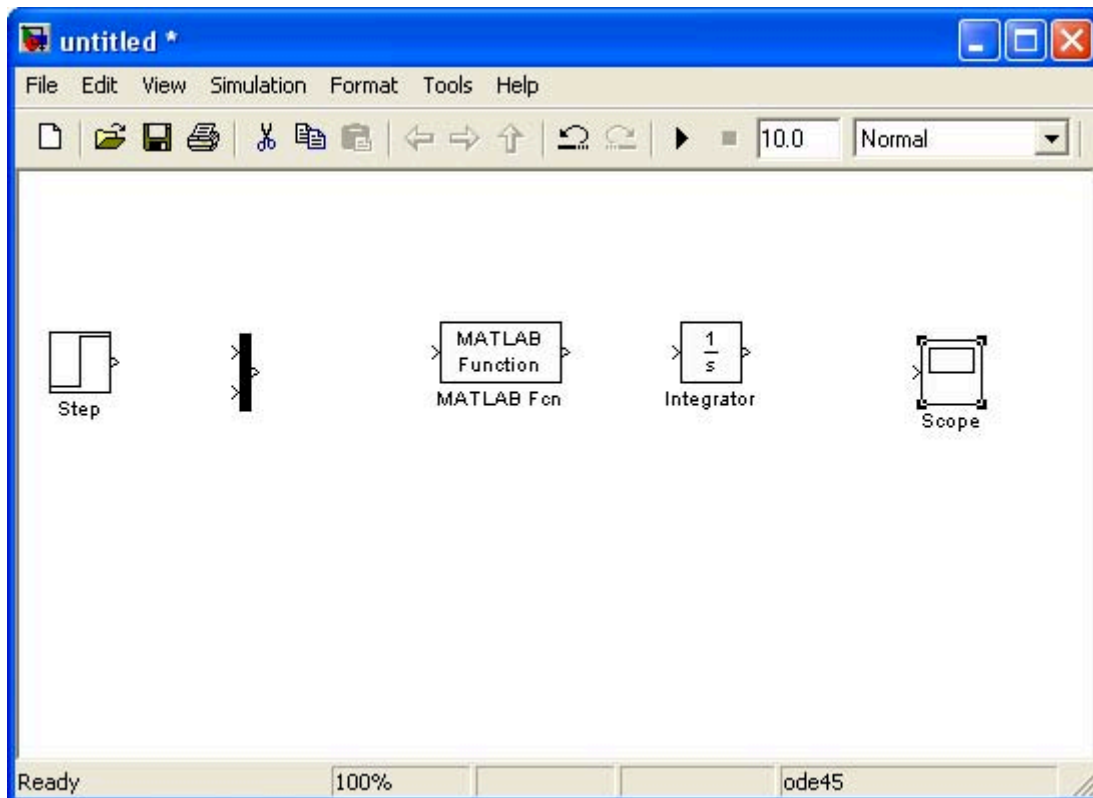
B) Από την κατηγορία **Signal Routing** επιλέγουμε το διάγραμμα **Mux**. Το διάγραμμα αυτό συνδυάζει δύο ή περισσότερα βαθμωτά ή/και διανυσματικά σήματα εισόδου. Σαν έξοδο δίνει ένα διανυσματικό σήμα που περιέχει τα σήματα εισόδου.



Σχήμα 15.30. Το διάγραμμα **Mux**.

Γ) Επίσης επιλέγονται και τα διαγράμματα **Step**, **Scope** και **Integrator** που χρησιμοποιήθηκαν και στο Παράδειγμα 15.2.

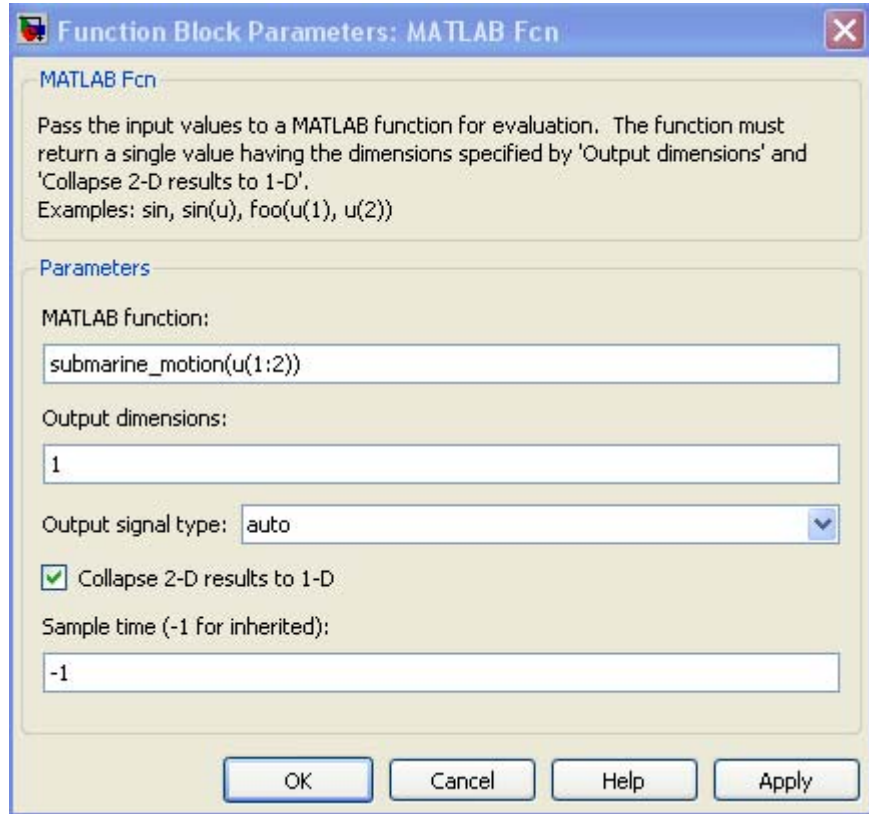
Εισάγουμε τα παραπάνω διαγράμματα σε ένα νέο αρχείο, οπότε προκύπτει το Σχήμα 15.31.



Σχήμα 15.31. Το αρχείο με τα απαιτούμενα διαγράμματα.

Τώρα πρέπει να εισαχθούν στα διαγράμματα οι κατάλληλες παράμετροι. Για τα διαγράμματα **Step**, **Integrator** και **Scope** ισχύουν οι παράμετροι του Παραδείγματος 15.2. Πατώντας δύο φορές πάνω στο διάγραμμα **MATLAB Fcn** αλλάζουμε την παράμετρο **MATLAB Fcn**: σε `submarine_motion(u(1:2))`. Το `submarine_motion` είναι το όνομα του function αρχείου που δίνει τη μη γραμμική διαφορική εξίσωση (βλέπε Σχήμα 15.33). Η δημιουργία τέτοιων

αρχείων έχει αναφερθεί στο Κεφ. 13. Η μεταβλητή $u(1:2)$ δίνει το όνομα και το πλήθος των εισόδων που δέχεται το διάγραμμα **MATLAB Fcn** (στο παράδειγμα αυτό είναι η προωθητική δύναμη F και η ταχύτητα v του υποβρυχίου). Επίσης, αλλάζουμε την παράμετρο **Output Dimensions**: σε 1 που αντιστοιχεί στο πλήθος των εξόδων του διαγράμματος **MATLAB Fcn** (στο παράδειγμα αυτό είναι η επιτάχυνση \dot{v} του υποβρυχίου). Τέλος πατάμε Apply και μετά OK.

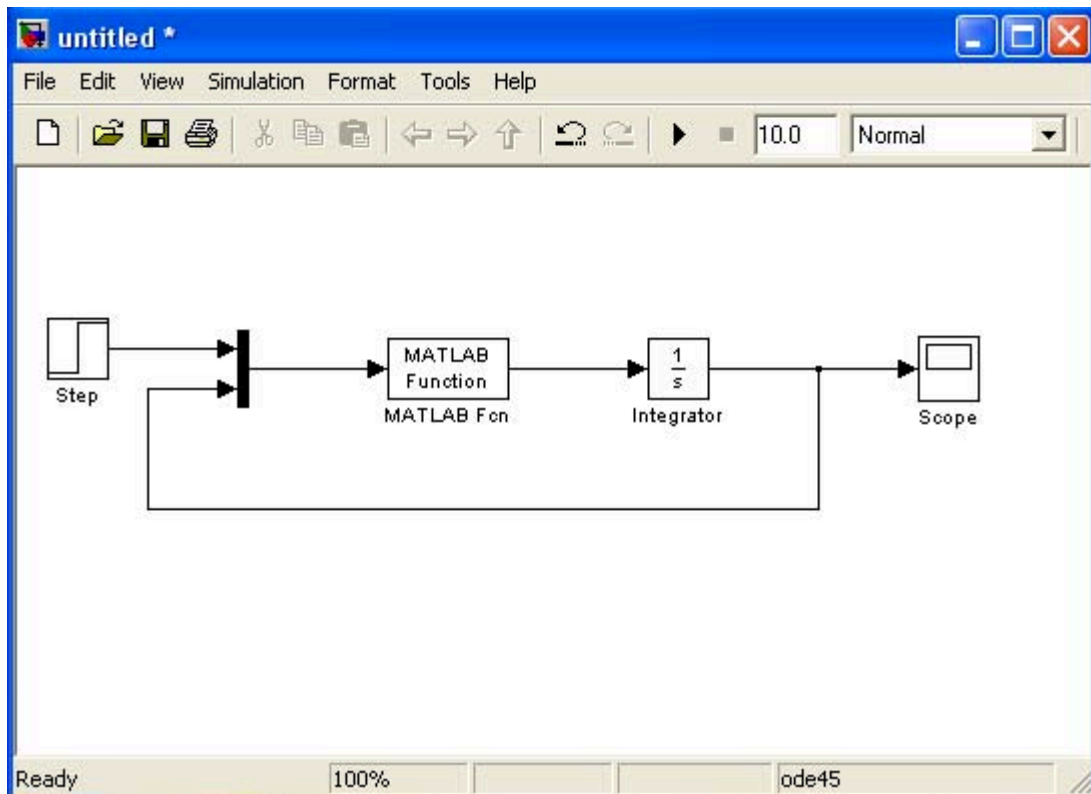


Σχήμα 15.32. Εισαγωγή παραμέτρων του διαγράμματος **MATLAB Fcn**.

```
function vdot=submarine_motion(u)
%Define the system parameters
M=2400;
B=320;
%Define the block inputs
F=u(1); % First input (start from the top of the Mux block)
v=u(2); % Second input
%Define the block outputs
vdot=-(B/M)*v^2+(1/M)*F;
```

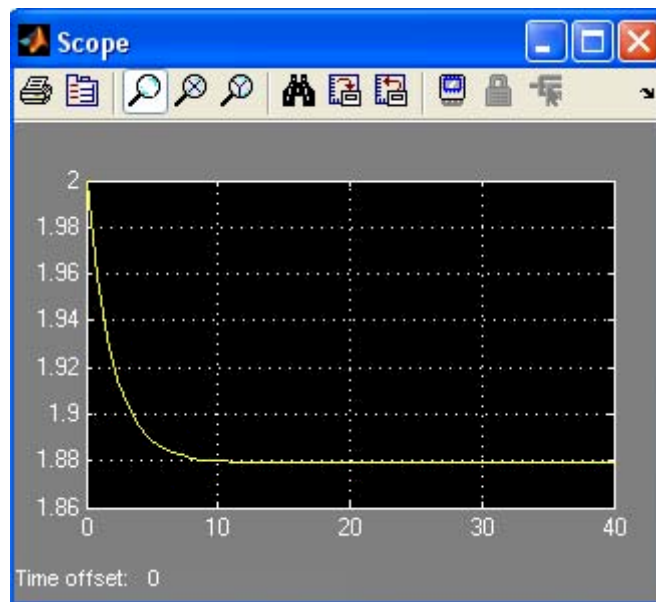
Σχήμα 15.33. Αρχείο function που χρησιμοποιείται στο διάγραμμα **MATLAB Fcn**.

Στην συνέχεια συνδέουμε τα διαγράμματα όπως φαίνεται στο Σχήμα 15.34.



Σχήμα 15.34. Τελική μορφή του αρχείου.

Επιλέγοντας όπως και στο παράδειγμα 15.1 τις παραμέτρους επίλυσης και τρέχοντας το αρχείο προκύπτει η απόκριση της ταχύτητας του υποβρυχίου:



Σχήμα 15.35. Η απόκριση της ταχύτητας όπως φαίνεται από το διάγραμμα **Scope**.

Στα παραπάνω παραδείγματα προσομοιώσαμε την απόκριση ενός συστήματος (γραμμικού και μη γραμμικού) με ή χωρίς αρχικές συνθήκες σε μια βηματική είσοδο. Στα συστήματα αυτομάτου ελέγχου, η συμπεριφορά ενός τέτοιου συστήματος ελέγχεται από έναν ελεγκτή. Ο ελεγκτής συνδέεται με το προς έλεγχο σύστημα και μπορεί να είναι είτε συνεχούς χρόνου

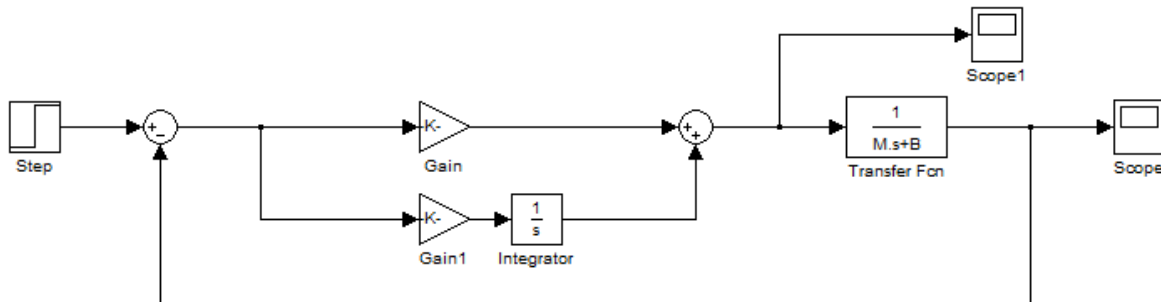
είτε διακριτού. Στο Παράδειγμα 15.4 παρουσιάζεται η σύνδεση ενός PI ελεγκτή με το προς έλεγχο σύστημα μέσω του Simulink.

Παράδειγμα 15.4: Να γίνει έλεγχος, μέσω του Simulink, του συστήματος του υποβρυχίου με συνάρτηση μεταφοράς που δίνεται από την Εξ. (15.3), με έναν PI ελεγκτή που έχει κέρδη:

$$k_p=160, \quad k_i=240$$

και να παρουσιαστούν γραφικά η απόκριση της ταχύτητας για επιθυμητή τιμή αναφοράς $v=1$ m/s καθώς και ο νόμος ελέγχου που προκύπτει από τον ελεγκτή.

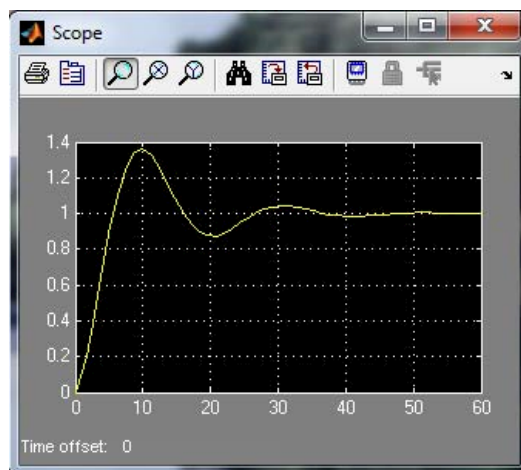
Στο Σχήμα 15.36 δίνεται η δομή του συστήματος κλειστού βρόχου.



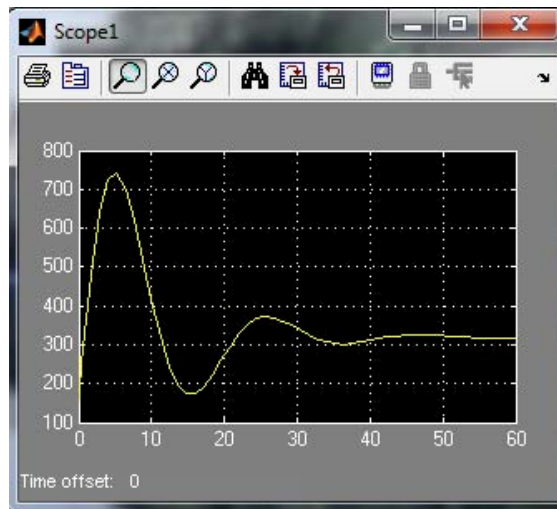
Σχήμα 15.36. Έλεγχος του υποβρυχίου με PI ελεγκτή.

όπου στο διάγραμμα **Step** εισάγεται η επιθυμητή τιμή της ταχύτητας, σύμφωνα με το Σχήμα 15.10 και στα διαγράμματα **Gain** και **Gain1** εισάγονται οι αντίστοιχες τιμές των κερδών k_p και k_i του ελεγκτή σύμφωνα με το Σχήμα 15.20.

Από το διάγραμμα **Scope** προκύπτει η απόκριση της ταχύτητας όπως φαίνεται στο Σχήμα 15.37, ενώ ο νόμος ελέγχου προκύπτει από το διάγραμμα **Scope1** και παρουσιάζεται στο Σχήμα 15.38.



Σχήμα 15.37. Απόκριση ταχύτητας υποβρυχίου μετά τον έλεγχο.



Σχήμα 15.38. Νόμος έλεγχου PI ελεγκτή.

Παράρτημα Α

Εισαγωγή εικόνας σε αρχείο word

Επιλέγουμε από το Figure 1 του Σχήματος 15.28α:

“File → Export Setup → Επιλέγουμε επιθυμητά χαρακτηριστικά για Size, Rendering, Fonts και Lines → Export → Επιλογή είδους (Eps, JPEG κ.λπ.) και ονομασίας αρχείου → Save → OK”.

Στην συνέχεια στο αρχείο Word εκτελούμε:

“Εισαγωγή → Εικόνα από αρχείο → Επιλέγουμε το επιθυμητό αρχείο” και προκύπτει το Σχήμα 15.28β.