ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΙΔΡΥΜΑ
ΑΝΑΤΟΛΙΚΗΣ ΜΑΚΕΔΟΝΙΑΣ ΚΑΙ ΘΡΑΚΗΣ

ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ Τ.Ε.

ΤΟΜΕΑΣ ΥΠΟΛΟΓΙΣΤΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΜΑΘΗΜΑ: ΕΙΣΑΓΩΓΗ ΣΤΗΝ ΥΠΟΛΟΓΙΣΤΙΚΗ ΝΟΗΜΟΣΥΝΗ

ΕΡΓΑΣΤΗΡΙΑΚΕΣ ΑΣΚΗΣΕΙΣ

Καθηγητής Βασ. Καμπουρλάζος

ΚΑΒΑΛΑ ΟΚΤΩΒΡΙΟΣ 2015

ΠΡΟΛΟΓΟΣ

Το εργαστήριο του μαθήματος «Εισαγωγή στην Υπολογιστική Νοημοσύνη (ΥΝ)» έχει αντικείμενο την πειραματική εξοικείωση με κάποιες «νέες τεχνολογίες» οι οποίες περιλαμβάνουν, μεταξύ άλλων, ασαφή συστήματα, νευρωνικά δίκτυα, κλπ. Το χαρακτηριστικό των «νέων τεχνολογιών» είναι ότι η αρχική έμπνευση για την ανάπτυξη τους προήλθε από την λειτουργία του ανθρώπινου εγκεφάλου, από εκεί προέρχεται και ο όρος *ευφυές* (σύστημα).

Ο σκοπός ενός ευφυούς συστήματος είναι, τελικά, να ελέγχει ένα άλλο (μη-ευφυές) σύστημα. Π.χ. ένα νευρωνικό δίκτυο μπορεί να ελέγχει την ομαλή λειτουργία μιας μηχανής μέσα σε μια βιομηχανική μονάδα, ένα ασαφές σύστημα μπορεί να ελέγχει μια παραγωγική διαδικασία, κλπ.

Στα πλαίσια αυτού του εργαστηρίου το πρόβλημα που καλείται να λύσει ένα ευφυές σύστημα είναι τελικά ένα πρόβλημα εκμάθησης μιας συνάρτησης $y= f(x)$, δοθέντος ενός συνόλου μετρήσεων (x_1,y_1), $(x_2,y_2),\ldots,(x_n,y_n)$.

Το εργαστήριο περιλαμβάνει μια σειρά εφαρμογών σε απλά προβλήματα ώστε να γίνει κατανοητή η χρήση και χρησιμότητα των ευφυών συστημάτων. Τα βασικό πακέτο λογισμικού που θα χρησιμοποιηθεί στο εργαστήριο είναι το MATLAB το οποίο πλέον θεωρείται ένα παγκόσμιο standard. Συγκεκριμένα θα χρησιμοποιηθούν διάφορα προγράμματα εφαρμογής (Toolboxes) του MATLAB.
Αυτές οι σημειώσεις περιλαμβάνουν επίσης ένα συνοπτικό εγχειρίδιο χρήσης του MATLAB, καθώς και οδηγίες χρήσης των Toolboxes που απαιτούνται. Δείχνονται επίσης βήμα-προς-βήμα παραδείγματα εφαρμογής σε συγκεκριμένα πρακτικά παραδείγματα όπως θα ζητηθεί και στις εξετάσεις.

ΠΕΡΙΕΧΟΜΕΝΑ

ΚΕΦΑΛΑΙΟ 1: ΣΥΝΟΠΤΙΚΟ ΕΓΧΕΙΡΙΔΙΟ ΕΝΤΟΛΩΝ MATLAB

ΚΕΦΑΛΑΙΟ 2: ΑΣΑΦΗ ΣΥΣΤΗΜΑΤΑ

ΚΕΦΑΛΑΙΟ 3: ΤΕΧΝΗΤΑ ΝΕΥΡΩΝΙΚΑ ΔΙΚΤΥΑ

ΚΕΦΑΛΑΙΟ 1: ΣΥΝΟΠΤΙΚΟ ΕΓΧΕΙΡΙΔΙΟ ΕΝΤΟΛΩΝ MATLAB

Το όνομα MATLAB προέρχεται από τις λέξεις MATrix (πίνακας) και LABoratory (εργαστήριο). Σήμερα το MATLAB θεωρείται παγκοσμίως ένα ενδεδειγμένο πακέτο λογισμικού γιά την ανάλυση γραμμικών και μη-γραμμικών συστημάτων, και αναγνωρίζεται σαν ένα ευέλικτο εργαλείο αριθμητικής ανάλυσης. Ο στόχος αυτού του εγχειριδίου είναι να δείξει περιληπτικά το εύρος των εντολών που είναι διαθέσιμες στο MATLAB.

1.1 Βασικές Αρχές

1.1.1 Είσοδος από το πληκτρολόγιο.

Εισαγωγή διανύσματος

```
>> A=[1 2 3 4]
>> A=[1,2,3,4];
```

Εισαγωγή πίνακα

```
>> A=[1 2 3 4;5 6 7 8]
>> A=[1 2 3 4];
```

Ανάστροφος πίνακα/διανύσματος

```
>> A=A'
```

1.1.2 Είσοδος από εξωτερικά αρχεία.

Γιά φόρτωμα ενός ASCII αρχείου

```
>> load dat.txt
```

Για αποθήκευση των μεταβλητών *VarName1 VarName2 VarName3 …* στο αρχείο *Filename* .

```
>> save Filename VarName1 VarName2 VarName3 …
```

Για αποθήκευση της μεταβλητής *VarName σε* ASCII μορφή στο αρχείο *Filename.Extension*.

```
>> save Filename.Extension VarName -ascii
```

1.1.3 Χρήσιμες εντολές.

Μορφοποίηση των μεταβλητών

```
>> format short
>> format long
```

Γιά βοήθεια από το ίδιο το MATLAB τύπωσε "help".

Οι γνωστές εντολές "dir", "cd" υπάρχουν και στο MATLAB .

Η εντολή "clear" σβήνει όλες τις μεταβλητές.

Η εντολή "clc" καθαρίζει μόνο την οθόνη.

1.2 Χειρισμός Διανυσμάτων και Πινάκων

1.2.1 Στοιχεία ενός πίνακα

Ενα από τα σημαντικότερα χαρακτηριστικά του MATLAB είναι ο μη-καθορισμός των διαστάσεων ενός πίνακα εκ των προτέρων.

Αναφερόμαστε σε ένα στοιχείο κάποιου πίνακα χρησιμοποιώντας τους κατάλληλους δείκτες

```
>> A(i,j)
```

Γιά να σβήσουμε ένα στοιχείο κάποιου πίνακα

```
>> x(4)=[]
```

Γιά να βρούμε το μέγεθος κάποιου πίνακα

```
>> size(A)
```

Γιά να βρούμε το μήκος κάποιου διανύσματος

```
>> length(A)
```

Γιά να βρούμε ποιές μεταβλητές υπάρχουν στο χώρο εργασίας

```
>> who
```

Γιά να βρούμε το μέγεθος των μεταβλητών του χώρου εργασίας

```
>> whos
```

1.2.2 Αναπαράσταση διαστήματος

Ενα διάστημα μπορεί να οριστεί περιγραφικά ορίζοντας την αρχή του, το τέλος του, καθώς και το βήμα σάρωσης του, π.χ.

```
>> x=0:0.01:5;
>> x=0:pi/4:pi
```

Ενα διάστημα μπορεί επίσης να οριστεί ορίζοντας την αρχή του, το τέλος του, καθώς και το πλήθος των σημείων του

```
>> k=linspace(-pi,pi,4)
```

Γιά λογαριθμική κλίμακα μεταξύ των άκρων ενός διαστήματος χρησιμοποιούμε την εντολή

```
>> k=logspace(p1, p2, #points)
```

όπου τα p_1 και p_2 είναι δυνάμεις (εκθέτες) του 10, π.χ. η εντολή logspace(1, 2) αντιστοιχεί στο διάστημα $[10^1, 10^2]$.

1.2.3 Πράξεις με πίνακες

Αναφερόμαστε σε τμήματα ενός πίνακα με διανύσματα πίνακα

```
>> B=A(1:3,1:3)
```

Εάν "a" και "k" είναι n-διάστατα διανύσματα και τα στοιχεία του "k" είναι 0 και 1 τότε το a(k) δίνει μόνο τα στοιχεία του "a" που βρίσκονται στην ίδια θέση με τα 1 του διανύσματος "k".

```
>> a=[2 4 45 22];
>> k=[0 1 0 1];
>> a(find(k))
ans=
        4   22
```

1.2.4 Ειδικοί πίνακες

Μοναδιαίος πίνακας

```
>> eye(4)
>> eye(3,4)
```

Πίνακας με μηδενικά

```
>> zeros(3)
```

Πίνακας με μονάδες

```
>> ones(1,5)
```

Διαγώνιος πίνακα (2 χρήσεις)

```
>> A=diag([1 2 3 4])
>> diag(A)
```

1.2.5 Σειρές χαρακτήρων

Μπορούμε να βάλουμε ένα σύνολο λέξεων σε ένα διδιάστατο πίνακα ως εξής

```
>> A=str2mat('today','it will','rain')
```

1.3 Βαθμωτές Πράξεις

Διάφορες συναρτήσεις που ορίζονται στο MATLAB

- ans το αποτέλεσμα μιάς πράξης όταν δεν ορίζεται μιά μεταβλητή.
- eps η ακρίβεια των υπολογισμών.
- computer ο τύπος του H/Y.
- pi ο αριθμός π.
- i, j ο μιγαδικός αριθμός $\sqrt{-1}$.
- Inf το άπειρο.
- NaN ο "μη-αριθμός", π.χ. 0/0.
- clock το ρολόι του H/Y.
- cputime ο χρόνος που πέρασε γιά την κεντρική μονάδα επεξεργασίας.
- date η ημερομηνία.
- realmax ο πιό μεγάλος αριθμός.
- realmin ο πιό μικρός αριθμός.
- nargin ο αριθμός των ορισμάτων εισόδου μιάς συνάρτησης.
- nargout ο αριθμός των ορισμάτων εξόδου μιάς συνάρτησης.

1.3.1 Αριθμητικές πράξεις

Οι γνωστές αριθμητικές πράξεις γίνονται χρησιμοποιώντας τους τελεστές + , - , * , / .

Το γινόμενο ενός βαθμωτού “a” με ένα διάνυσμα “x” βρίσκεται με την πράξη “a*x”.

Το εσωτερικό γινόμενο δύο διανυσμάτων στήλης “x” και “y” βρίσκεται με την πράξη “x’*y”.

Το γινόμενο δύο διανυσμάτων “x” και “y” στοιχείο-προς-στοιχείο με αποτέλεσμα και πάλι διάνυσμα βρίσκεται με την πράξη “x.*y”.
Παρόμοια ορίζεται η διαίρεση “x./y” και η ύψωση σε δύναμη “x.^y”.

1.3.2 Διάφορες συναρτήσεις

Στρογγυλοποίηση

- round στρογγυλοποίηση προς τον πλησιέστερο ακέραιο.
- fix στρογγυλοποίηση προς το μηδέν.
- floor στρογγυλοποίηση προς τον μικρότερο ακέραιο.
- ceil στρογγυλοποίηση προς τον μεγαλύτερο ακέραιο.

Προσέγγιση με ρητούς αριθμούς

- rem υπόλοιπο της διαίρεσης με ακεραίους.
- rat ανάπτυξη σε ρητούς αριθμούς.
- rats προσέγγιση με ρητούς αριθμούς.

Παραγοντοποίηση με ακέραιους

- gcd μέγιστος κοινός διαιρέτης.
- lcm ελάχιστο κοινό πολλαπλάσιο.

Μιγαδική αριθμητική

- real το πραγματικό μέρος ενός μιγαδικού αριθμού.
- imag το φανταστικό μέρος ενός μιγαδικού αριθμού.
- conj συζυγής ενός μιγαδικού αριθμού.
- abs το μέτρο ενός μιγαδικού αριθμού.
- angle το όρισμα ενός μιγαδικού αριθμού.

Από Καρτεσιανές συντεταγμένες σε πολικές (ή σε κυλινδρικές) : cart2pol.
Από πολικές (ή σε κυλινδρικές) συντεταγμένες σε Καρτεσιανές : pol2cart.
Από Καρτεσιανές συντεταγμένες σε σφαιρικές : cart2sph.
Από σφαιρικές συντεταγμένες σε Καρτεσιανές : sph2cart.

1.3.3 Παραδείγματα

- Στοιχειώδεις συναρτήσεις

```
>> x=(1:0.1:5)';
>> y=log(x);
>> [x y]
```

- Συναρτήσεις συναρτήσεων

Να υπολογιστεί ένας πίνακας τιμών της συνάρτησης $e^{3t}\sin(5\pi t)$

```
>> t=linspace(-2,2,45)';
>> y=exp(3*t).*sin(5*pi*t);
>> [t y]
```

• Ρητές συναρτήσεις

Πίνακας τιμών της συνάρτησης $f(s)=\frac{3s^2+5s+7}{s^3+5s^2+7s+12}$ με s=jω και ω∈[10^{-2}, 10^2].

```
>> omega=logspace(-2,2);
>> s=j*omega;
>> x1=3*s.^2+5*s+7;
>> x2=s.^3+5*s.^2+7*s+12;
>> f=x1./x2;
>> x=abs(f);
>> [s  x]
```

1.3.4 Λογικοί και σχεσιακοί τελεστές

Οι γνωστοί σχεσιακοί τελεστές <, <=, >, >=, ==, ~=.
Οι γνωστοί λογικοί τελεστές &, |, xor, ~.

Παράδειγμα

```
>> A=[1 2 3 4;5 6 7 8];
>> P=(rem(A,2)==0)

ans =
     0     1     0     1
     0     1     0     1
```

1.4 Πράξεις με Πίνακες

Οι προηγούμενες βαθμωτές συναρτήσεις είναι άμεσα διαθέσιμες και με πίνακες. Π.χ. γιά τον πίνακα $A=\begin{pmatrix}1 & 3\\ 4 & 2\end{pmatrix}$ η εντολή C=exp(A) υπολογίζει τον πίνακα $C=\begin{pmatrix}e^1 & e^3\\ e^4 & e^2\end{pmatrix}$.

1.4.1 Ανάστροφος ενός πίνακα

```
>> B=A'
```

(στην πραγματικότητα ο τελεστής " ' " δίνει τον ανάστροφο συζυγή).

1.4.2 Αλγεβρικές πράξεις

Οι τελεστές +, -, * ορίζονται και γιά πίνακες.

Ο αντίστροφος ενός πίνακα υπολογίζεται με την εντολή "inv(A)" ή με την εντολή "A^(-1)".

1.4.3 Συναρτήσεις με πίνακες

Η εντολή max(x) (min(x)) υπολογίζει το μέγιστο (ελάχιστο) ενός διανύσματος. Όταν αυτές οι εντολές χρησιμοποιούνται με πίνακες δίνουν ένα διάνυσμα με τα μέγιστα (ελάχιστα) κάθε στήλης του πίνακα. Το μέγιστο ενός πίνακα βρίσκεται με την εντολή max(max(A)).

Η εντολή sort(x) ταξινομεί ένα διάνυσμα σε αύξουσα σειρά. Η εντολή [y,ind]=sort(x) επιστρέφει τους δείκτες ind του ταξινομημένου διανύσματος y.

Η εντολή sum(x) υπολογίζει το άθροισμα, και η εντολή mean(x) δίνει τον μέσο όρο των στοιχείων ενός διανύσματος.

Η εντολή rank(A) υπολογίζει την τάξη ενός πίνακα.

Η εντολή det(A) υπολογίζει την ορίζουσα ενός πίνακα.

Η εντολή poly(A) υπολογίζει τους συντελεστές του χαρακτηριστικού πολυωνύμου p(λ) του πίνακα A, όπου p(λ)=|λI-A|.

Η εντολή trace(A) υπολογίζει το ίχνος ενός πίνακα A (δηλ. το άθροισμα των στοιχείων της κύριας διαγωνίου του πίνακα A).

Η εντολή norm(X) υπολογίζει τη νόρμα ενός πίνακα.

Η εντολή expm(A) υπολογίζει το $e^A = \sum_{n=0}^{\infty} \frac{A^n}{n!}$.

Παράδειγμα

Να ταξινομηθούν τα στοιχεία του πίνακα $R = \begin{pmatrix} 1 & 2 \\ 5 & 1 \\ 3 & 3 \\ 2 & 4 \end{pmatrix}$ ως προς την πρώτη στήλη αλλά χωρίς να χαθεί η συσχέτιση με τα στοιχεία της δεύτερης στήλης.

Προσέξτε ότι η εντολή A=sort(R) έχει σαν αποτέλεσμα $A = \begin{pmatrix} 1 & 1 \\ 2 & 2 \\ 3 & 3 \\ 5 & 4 \end{pmatrix}$, οπότε χάνεται η συσχέτιση με τα στοιχεία της δεύτερης στήλης. Ακολουθούν οι σωστές εντολές

```
>> [S,i1]=sort(R);
>> A=R(i1(:,1),:);
```

που έχουν σαν αποτέλεσμα $A = \begin{pmatrix} 1 & 2 \\ 2 & 4 \\ 3 & 3 \\ 5 & 1 \end{pmatrix}$.

1.4.4 Λογικές πράξεις

Η εντολή any(x), όπου το x είναι ένα διάνυσμα με 0 και 1, επιστρέφει 1 αν τουλάχιστον ένα από τα στοιχεία του x είναι 1.

Η εντολή all(x), όπου το x είναι ένα διάνυσμα με 0 και 1, επιστρέφει 1 αν όλα τα στοιχεία του x είναι 1.

Οι προηγούμενες εντολές ισχύουν κατά στήλη και γιά πίνακες, π.χ. η εντολή all(all(A<0.5)) επιστρέφει 1 εάν όλα τα στοιχεία του πίνακα A είναι μικρότερα του 0.5.

Η εντολή find επιστρέφει τους δείκτες των μη-μηδενικών στοιχείων του ορίσματός της.

Παράδειγμα

Οι εντολές

```
>> t=0:0.005:20;
>> y=sin(t);
>> i=find(abs(y-0.5)<0.05);
```

επιστρέφουν τους δείκτες εκείνων των στοιχείων του διανύσματος "y" που απέχουν λιγότερο από 0.05 από τον αριθμό 0.5 (Πρακτικά αυτό σημαίνει ότι μπορούμε να υπολογίσουμε τις τιμές της ανεξάρτητης μεταβλητής "t" των οποίων οι αντίστοιχες τιμές του "y" είναι 0.5).

1.5 Πολυώνυμα

Ένα πολυώνυμο ορίζεται σαν ένα διάνυσμα με φθίνουσα σειρά των συντελεστών του. Π.χ. το πολυώνυμο $p(s)=s^3+4s^2+2s+5$ ορίζεται ως p=[1 4 2 5], ενώ το s^3+1 ορίζεται ως [1 0 0 1].

1.5.1 Βασικές πράξεις

Η εντολή poly(A) υπολογίζει τους συντελεστές του χαρακτηριστικού πολυωνύμου p(λ) του πίνακα A, όπου p(λ)=|λI-A|.

Η εντολή poly χρησιμοποιείται επίσης γιά να υπολογίσει ένα πολυώνυμο από τις ρίζες του, π.χ. το πολυώνυμο με ρίζες τις 1, -2, j, και -j υπολογίζεται με την εντολή poly([1,-2,j,-j]) και δίνεται από το διάνυσμα [1 1 -1 1 -2].

Η εντολή conv υπολογίζει το γινόμενο δύο πολυωνύμων, π.χ. με a=[1 2 3] και b=[4 5 6] η εντολή c=conv(a,b) δίνει c=[4 13 28 27 18].

Οι ρίζες ενός πολυωνύμου υπολογίζονται με την εντολή roots(p).

Η εντολή polyder(p) επιστρέφει την πρώτη παράγωγο dp(s)/ds του πολυωνύμου p.

Γιά να αναπτύξουμε ένα ρητό πολυώνυμο $\frac{b(s)}{a(s)}$ σε μερικά κλάσματα όπως φαίνεται παρακάτω

$$\frac{b(s)}{a(s)} = \frac{r_1}{s-p_1} + \frac{r_2}{s-p_2} + \ldots + \frac{r_n}{s-p_n} + k(s)$$

χρησιμοποιούμε την εντολή [r,p,k]=residue(b,a).

1.5.2 Παρεμβολή

Δοθείσης μιάς ανεξάρτητης μεταβλητής στο διάνυσμα "x" και των τιμών που πρόκειται να παρεμβληθούν στο διάνυσμα "y" η εντολή p=polyfit(x,y,n) επιστρέφει ένα πολυώνυμο n-στου βαθμού που παρεμβάλει τα ζεύγη (x_i,y_i) με τη μέθοδο των ελαχίστων τετραγώνων.

Παρεμβολή μπορεί να γίνει επίσης με τις εντολές spline, interpft, interp1, interp2.

1.6 Γραφικά

1.6.1 Διδιάστατα γραφικά

Παράδειγμα

```
>> t=0:0.05:4*pi;
>> y=sin(t);
>> plot(t,y)
```

1.6.2 Πολλαπλά γραφικά

Παράδειγμα

```
>> t1=(0:0.1:3)';
>> y1=sin(t1);
>> t2=(1:0.1:4)';
>> y2=cos(t2);
>> plot(t1,y1,t2,y2)
```

1.6.3 Η κλίμακα των αξόνων

Η εντολή axis([xmin xmax ymin ymax]) ορίζει τα άκρα των αξόνων x και y κατά τη σχεδίαση.

Η εντολή axis('square') επιβάλει μιά τετράγωνη περιοχή σχεδίασης.

Η εντολή axis('equal') επιβάλει ίσες μονάδες στους δύο άξονες σχεδίασης x και y.

Η εντολή grid σχεδιάζει ένα δικτύωμα επάνω στην περιοχή σχεδίασης.

1.6.4 Μιγαδικοί αριθμοί

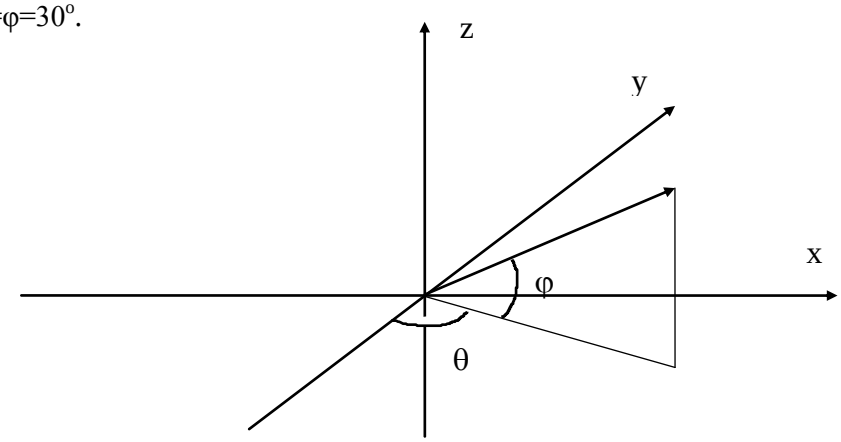
Η εντολή plot(X), όπου X είναι ένα διάνυσμα μιγαδικών αριθμών, έχει τα ίδια γραφικά αποτελέσματα όπως η εντολή plot(real(X),imag(X)).

1.6.5 Τρισδιάστατα γραφικά

Παράδειγμα : Συναρτήσεις δύο Μεταβλητών

```
>> x=0:0.1:4;
>> y=-2:0.1:1;
>> [X,Y]=meshgrid(x,y);
>> Z=sin(X).*cos(Y);
>> mesh(X,Y,Z);
```

Η εντολή view(az,el) ορίζει τη θέση παρατηρητή, όπου az και el είναι οι γωνίες παρατήρησης azimuth και elevation όπως φαίνονται στο παρακάτω σχήμα. Αρχικά είναι az=θ=-37.5^o και el=φ=30^o.



1.7 Προγραμματισμός στο MATLAB

1.7.1 Βασικές δομές

Η επιλογή στο MATLAB καθορίζεται ως εξής

```
if συνθήκη,
(σύνολο εντολών 1)
elseif συνθήκη,
(σύνολο εντολών 2)
else
(σύνολο εντολών 3)
  •
  •
end
```

Η επανάληψη στο MATLAB καθορίζεται με δύο τρόπους

1.
```
for i=1:n,
(σύνολο εντολών)
end
```

2.
```
while συνθήκη
(σύνολο εντολών)
end
```

1.7.2 Αρχεία γραφής (script files)

Παράδειγμα

```
% An M-file to compute the Fibonacci numbers
f=[1 1]; i=1;
while f(i)+f(i+1)<1000
  f(i+2)=f(i)+f(i+1);
  i=i+1;
end
plot(f)
```

1.7.3 Συναρτήσεις

Μιά συνάρτηση δημιουργείται όπως ένα αρχείο γραφής (script file) αλλά αρχίζει πάντοτε με την εξής σειρά

function *[output variables]=FunctionName(input variables)*

Παράδειγμα

```
function t = trace(a)
% TRACE   Sum of diagonal elements,
%   TRACE(A)   is the sum of the diagonal elements of A,
%   which is also the sum of the eigenvalues of A.

t =sum(diag(a));
```

1.8 Αριθμητική Ανάλυση

Θεωρείστε τη συνάρτηση

$humps(x) = \frac{1}{(x-0.3)^2+0.01} + \frac{1}{(x-0.9)^2+0.04} - 6$ της οποίας το αρχείο είναι

```
function y = humps(x)
y=1./((x-0.3).^2+0.01) + 1./((x-0.9).^2+0.04) - 6;
```

και της οποίας το γράφημα λαμβάνεται με τις εντολές

```
x=-1:0.01:2;
plot(x,humps(x))
```

1.8.1 Απειροστική ανάλυση

Η εντολή diff υπολογίζει τη διαφορά μεταξύ διαδοχικών στοιχείων ενός διανύσματος, π.χ. diff(x)=[x(2)-x(1), x(3)-x(2), … ,x(n)-x(n-1)].

Η εντολή Dy=diff(y)./diff(x) υπολογίζει την πρώτη παράγωγο.

Η εντολή s=quad('humps',0,1) υπολογίζει το ολοκλήρωμα της *humps* μεταξύ 0 και 1. (Υπολογίστηκε να είναι quad('humps',0,1)=29.858…).

1.8.2 Μη-γραμμικές εξισώσεις και βελτιστοποίηση

Η εντολή fzero('*FunctionName*',x0) υπολογίζει το πραγματικό (μη μιγαδικό) μηδενικό της συνάρτησης *FunctionName*' κοντύτερα στο x0.

Παράδειγμα

```
>> xz1=fzero('humps',0)
>> xz2=fzero('humps',1)
```

(Το xz1 ισούται με -0.131…, και το xz2 ισούται με 1.299…)

Ο υπολογισμός του τοπικού ελάχιστου μιάς συνάρτησης με μία μεταβλητή γίνεται με την εντολή x=fminbnd('*FunctionName*',x1,x2).

Παράδειγμα (υπολογισμός του αριθμού π)

```
>> pic= fminbnd ('cos',3,4)
```

1.8.3 Διαφορικές εξισώσεις

Το MATLAB έχει δύο μεθόδους επίλυσης διαφορικών εξισώσεων

1. Μέθοδος Runge-Kutta $2^{ης}$-$3^{ης}$ τάξης. : ode23, êáé
2. Μέθοδος Runge-Kutta $4^{ης}$-$5^{ης}$ τάξης. : ode45.

Θα πρέπει η προς ολοκλήρωση συνάρτηση να δίνεται σε *κανονική μορφή*, δηλ. $\frac{dx}{dt} = f(t,x)$.

Ενας τρόπος επίλυσης της πιό πάνω διαφορικής εξίσωσης είναι με την εντολή [x,t]=ode23('xDot',t0,tf,x0), όπου "xDot.m" είναι το αρχείο συνάρτησης που υπολογίζει την παραπάνω συνάρτηση *f,* t0 και tf είναι ο αρχικός και ο τελικός χρόνος της ολοκλήρωσης και x0 είναι το διάνυσμα των αρχικών καταστάσεων.

Παράδειγμα

Θεωρείστε την εξίσωση Van der Pol : $\ddot{x}+(x^2-1)\dot{x}+x=0$. Με $x_1=x$ και $x_2=\dot{x}$ η εξίσωση Van der Pol μπορεί επίσης να γραφτεί και ως

$$\dot{x}_1 = x_2$$
$$\dot{x}_2 = x_2(1-x_1^2)-x_1$$

Γράψτε το ακόλουθο αρχείο συνάρτησης "vdpol.m"

```
function xdot=vdpol(t,x)
xdot(1)=x(2);
xdot(2)=x(2).*(1-x(1).^2)-x(1);
```

Επιλύουμε την εξίσωση στο χρονικό διάστημα $0 \le t \le 20$ με αρχικές συνθήκες x0=[0.25; 1].

```
t0=0; tf=20;
x0=[0.25 1]';
[t,x]= ode23('vdpol',t0,tf,x0);
plot(t,x)
```

ΚΕΦΑΛΑΙΟ 2: ΑΣΑΦΗ ΣΥΣΤΗΜΑΤΑ

Σ΄αυτό το κεφάλαιο γίνεται μια σύντομη εισαγωγή στα ασαφή σύνολα, στην ασαφή λογική και στον ασαφή συμπερασμό. Επίσης ορίζεται ένα Ασαφές Σύστημα. Κατόπιν παρουσιάζονται δύο δημοφιλείς τρόποι σχεδιασμού Ασαφών Συστημάτων, αυτοί είναι 1) σχεδίαση κατά Mamdani, και 2) σχεδίαση κατά Sugeno. Σε κάθε περίπτωση παρουσιάζεται λεπτομερώς ένα παράδειγμα σχεδίασης Ασαφούς Συστήματος με το Fuzzy Logic Toolbox του MATLAB.

2.1 Περιληπτική Εισαγωγή στην Ασαφή Λογική

Ένα σύνολο Σ είναι μια συλλογή στοιχείων. Ένα στοιχείο α είτε ανήκει 100% στο Σ είτε ανήκει 0% στο σύνολο Σ (στην τελευταία περίπτωση λέμε ότι το στοιχείο α δεν ανήκει στο σύνολο Σ). Π.χ. εάν Σ είναι το σύνολο των φοιτητών που έχουν περάσει το μάθημα Μαθηματικά-Ι και α είναι ένας συγκεκριμένος φοιτητής τότε, κατά την κλασσική (Αριστοτέλεια) λογική, είτε $\alpha\in\Sigma$ είτε $\alpha\notin\Sigma$.

Υπάρχουν, όμως, περιπτώσεις όπου ένα σύνολο δεν μπορεί να καθοριστεί με σαφήνεια. Π.χ. Ποιο είναι το σύνολο Ψ των Ψηλών Φοιτητών μέσα στην αίθουσα; Ασφαλώς κάποιος φοιτητής α με ύψος 2.10 ανήκει στο Ψ με βεβαιότητα 100% ($\alpha\in\Psi$), ενώ κάποιος φοιτητής β με ύψος 1.55 ανήκει στο Ψ με βεβαιότητα 0% (δηλ. $\beta\notin\Psi$). Όμως τι μπορούμε να πούμε για κάποιον φοιτητή δ με ύψος 1.70 ή για κάποιον φοιτητή ε με ύψος 1.80; Τα ασαφή σύνολα επινοήθηκαν για να εκφράσουν το βαθμό βεβαιότητας που κάποιο στοιχείο ανήκει σε ένα σύνολο. Π.χ. στο προηγούμενο παράδειγμα μπορούμε να πούμε ότι $\delta\in\Psi$ με βεβαιότητα 30%, επίσης $\varepsilon\in\Psi$ με βεβαιότητα 70%.

Τα προηγούμενα επεκτείνονται και για να εκφράσουν το βαθμό αλήθειας μιας πρότασης. Είναι γνωστό ότι κατά κλασσική (Αριστοτέλεια) λογική μια πρόταση είτε ισχύει είτε δεν ισχύει. Π.χ. είτε τα ύψη ενός τριγώνου διέρχονται από το ίδιο σημείο είτε όχι. Υπάρχουν όμως περιπτώσεις όπου η κλασσική λογική δεν είναι χρήσιμη. Π.χ. κατά την κλασσική λογική ένα λιβάδι είτε θα είναι πράσινο είτε δεν θα είναι. Όμως από πρακτική άποψη είναι πιο χρήσιμο να πούμε ότι το λιβάδι είναι πράσινο στο βαθμό που το χόρτο του είναι πράσινο. Η ασαφής λογική χαρακτηρίζει κάθε πρόταση με έναν βαθμό αληθείας μεταξύ 0 (ψευδής πρόταση) και 1 (αληθής πρόταση). Ενώ, η κλασσική (Αριστοτέλεια) λογική χαρακτηρίζει κάθε πρόταση είτε ως αληθή (1) είτε ως ψευδή (0).

Η γνωστή διαδικασία της λογικής συνεπαγωγής στον προτασιακό συλλογισμό έχει επεκταθεί, με διάφορους τρόπους, και σε ασαφείς προτάσεις. Έτσι έχουν προκύψει διάφορες μορφές ασαφούς συμπερασμού.

Το ασαφές σύστημα είναι μια συνάρτηση f όπου το πεδίο ορισμού της f είναι μια ασαφής πρόταση, ή σε τελευταία ανάλυση ένα ασαφές σύνολο, και το πεδίο τιμών της f είναι είτε ένα ασαφές σύνολο (Ασαφές Σύστημα Mamdani) είτε μιά αλγεβρική συνάρτηση (Ασαφές Σύστημα Sugeno).

Τα ασαφή συστήματα έχουν καθιερωθεί σε πεδία εφαρμογών όπως ο αυτόματος βιομηχανικός έλεγχος, η ταξινόμηση δεδομένων, η ανάλυση αποφάσεων, τα έμπειρα συστήματα, κλπ. Τα ασαφή συστήματα είναι επίσης γνωστά και με διαφορετικά ονόματα, όπως ασαφή συστήματα βασισμένα σε κανόνες, έμπειρα ασαφή συστήματα, ασαφή μοντέλα, ελεγκτές ασαφούς λογικής, κλπ.

2.2 Μοντελοποίηση Συστημάτων με Λεκτικές Μεταβλητές

Θέλουμε να φέρουμε την θερμοκρασία ενός δωματίου σε κάποια επιθυμητά επίπεδα. Έστω ότι υπάρχουν δύο "μεταβλητές" ρύθμισης της θερμοκρασίας: 1) Η θέση του διακόπτη ενός αερόθερμου (από 0 έως 100%), και 2) το άνοιγμα μιας ενδιάμεσης πόρτας (από 0 έως 100%). Το πρόβλημα είναι τι τιμές να δώσουμε στις δύο μεταβλητές ρύθμισης της θερμοκρασίας έτσι ώστε να προκύψει μια επιθυμητή θερμοκρασία. Αυτό είναι ένα τυπικό πρόβλημα ελέγχου "κλειστού βρόχου". Συγκεκριμένα: υπάρχει το σύστημα ανοιχτού βρόχου "δωμάτιο" με δύο μεταβλητές εισόδου: 1) η θέση του διακόπτη του αερόθερμου, και 2) το άνοιγμα της ενδιάμεσης πόρτας. Επίσης υπάρχει μια μεταβλητή εξόδου: η επιθυμητή θερμοκρασία. Ο βρόχος κλείνει χρησιμοποιώντας ένα άλλο σύστημα, τον "ελεγκτή", που έχει μια είσοδο: την διαφορά μεταξύ επιθυμητής θερμοκρασίας και πραγματικής θερμοκρασίας του δωματίου. Επίσης ο ελεγκτής έχει δύο εξόδους: 1) τη θέση του διακόπτη του αερόθερμου, και 2) το άνοιγμα της ενδιάμεσης πόρτας.

Ο σχεδιασμός ενός αποτελεσματικού ελεγκτή για να διατηρεί τη θερμοκρασία σε επιθυμητά επίπεδα γίνεται με "παραδοσιακό" τρόπο ως εξής: διατυπώνεται ένα μαθηματικό (αλγεβρικό) μοντέλο του συστήματος ανοικτού βρόχου, δηλ. του συστήματος το οποίο θέλουμε να ελέγξουμε. Κατόπιν, υπολογίζεται ένα μαθηματικό (αλγεβρικό) μοντέλο του ελεγκτή έτσι ώστε να ικανοποιηθούν δοθείσες προδιαγραφές. Χαρακτηριστικό είναι ότι οι τιμές που λαμβάνει μια μεταβλητή εισόδου ή εξόδου είναι πραγματικοί αριθμοί.

Η παραπάνω διαδικασία σχεδίασης επαναλαμβάνεται και στα ασαφή συστήματα με μια σημαντική διαφορά: οι τιμές που λαμβάνει μια μεταβλητή εισόδου ή εξόδου σε ένα ασαφές σύστημα είναι ασαφή σύνολα, τα οποία (ασαφή σύνολα) θεωρούνται ως "λεκτικές" τιμές. Επιπλέον, μια μεταβλητή εισόδου ή εξόδου αποκαλείται "λεκτική" μεταβλητή. Συνεπώς, η λειτουργία του συστήματος δεν περιγράφεται, βασικά, με τον συνηθισμένο μαθηματικό (αλγεβρικό) τρόπο, αλλά η λειτουργία του συστήματος περιγράφεται με λεκτικό τρόπο. Η λεκτική (ασαφής) περιγραφή πολύπλοκων συστημάτων θεωρείται, από πολλούς ερευνητές, ότι είναι αξιόπιστη διότι μια τέτοια περιγραφή μπορεί να συλλάβει την ασάφεια η οποία είναι εγγενής σε πολύπλοκα συστήματα. Ασφαλώς, με ασαφή τρόπο περιγράφονται τόσο το φυσικό σύστημα ανοικτού βρόχου που πρόκειται να ελεγχθεί όσο και ο ελεγκτής ο οποίος πρόκειται να ελέγξει το φυσικό σύστημα. Γίνεται σημαντική έρευνα σε όλο το κόσμο για μοντελοποίηση και έλεγχο συστημάτων με ασαφείς τεχνικές.

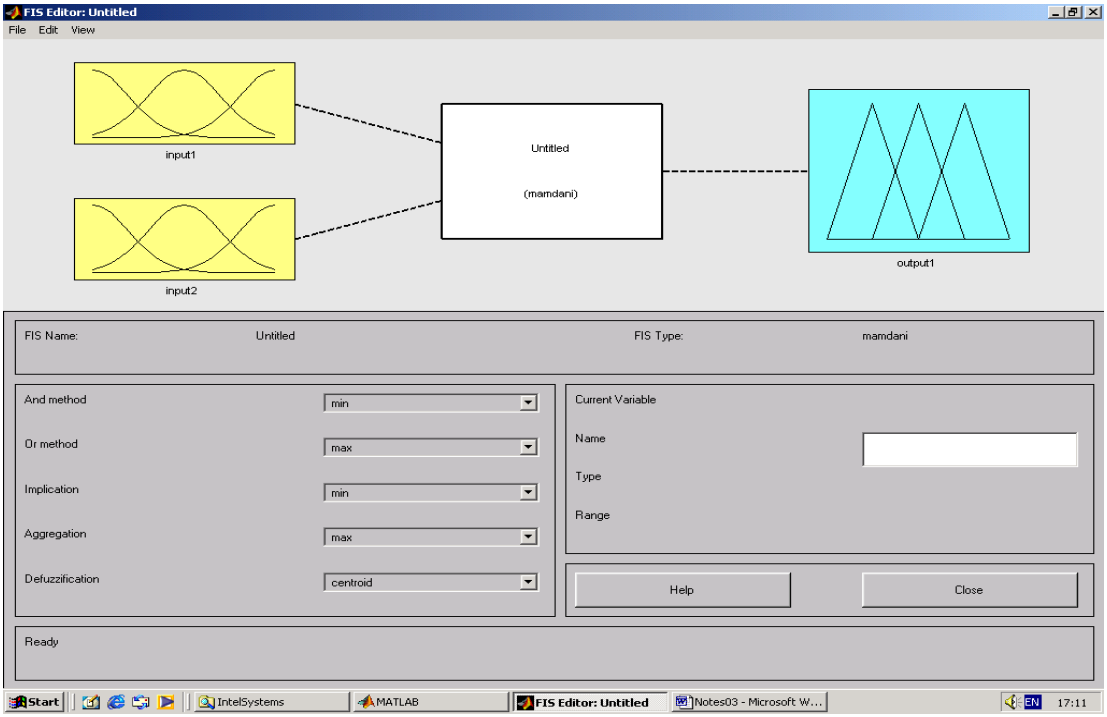
Στο εργαστήριο αυτό θα μελετηθούν πολύ απλά συστήματα με ασαφείς μεταβλητές τα οποία, κατ' αρχήν, θα είναι μοντέλα φυσικών συστημάτων ανοικτού βρόχου. Ο σκοπός αυτού του εργαστηρίου δεν είναι τόσο η αποτελεσματική σχεδίαση ασαφών συστημάτων σε συγκεκριμένες εφαρμογές, όσο η διδασκαλία κάποιων γενικών αρχών σχεδίασης με χρήση ασαφών συνόλων με χρήση του προγράμματος εφαρμογής 'Fuzzy' του MATLAB.

2.3 Το Πρόγραμμα Εφαρμογής 'Fuzzy' του MATLAB

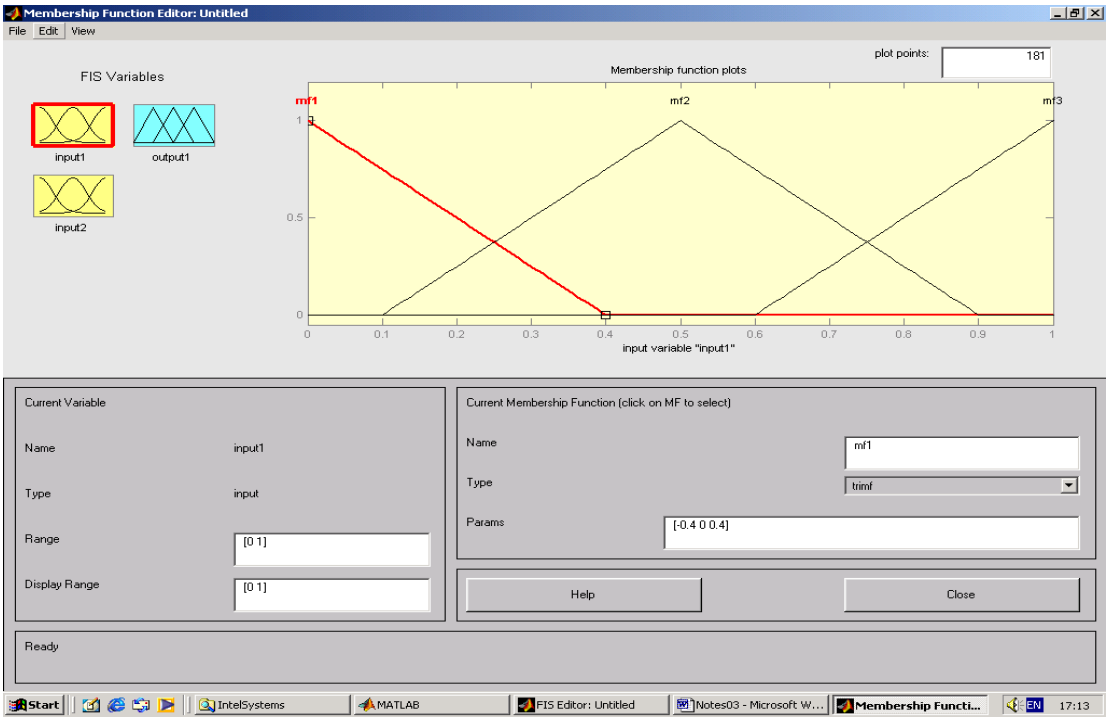
Το γραφικό σύστημα GUI (Graphical User Interface) ενεργοποιείται με την εντολή 'Fuzzy' (και μετά Enter) στην επιφάνεια εργασίας του MATLAB. Υπάρχουν διαθέσιμα πέντε βασικά εργαλεία GUI για τον σχεδιασμό, την διόρθωση, και την παρατήρηση συστημάτων ασαφούς λογικής: το Fuzzy Inference System ή FIS Editor, το Membership Function Editor, το Rule Editor, το Rule Viewer, και το Surface Viewer. Αυτά τα εργαλεία συνδέονται δυναμικά, με τέτοιο τρόπο ώστε αν γίνουν

αλλαγές σε ένα από αυτά ενημερώνονται και τα υπόλοιπα. Οποιαδήποτε από αυτά τα εργαλεία μπορεί να είναι ενεργά μια δεδομένη χρονική στιγμή.

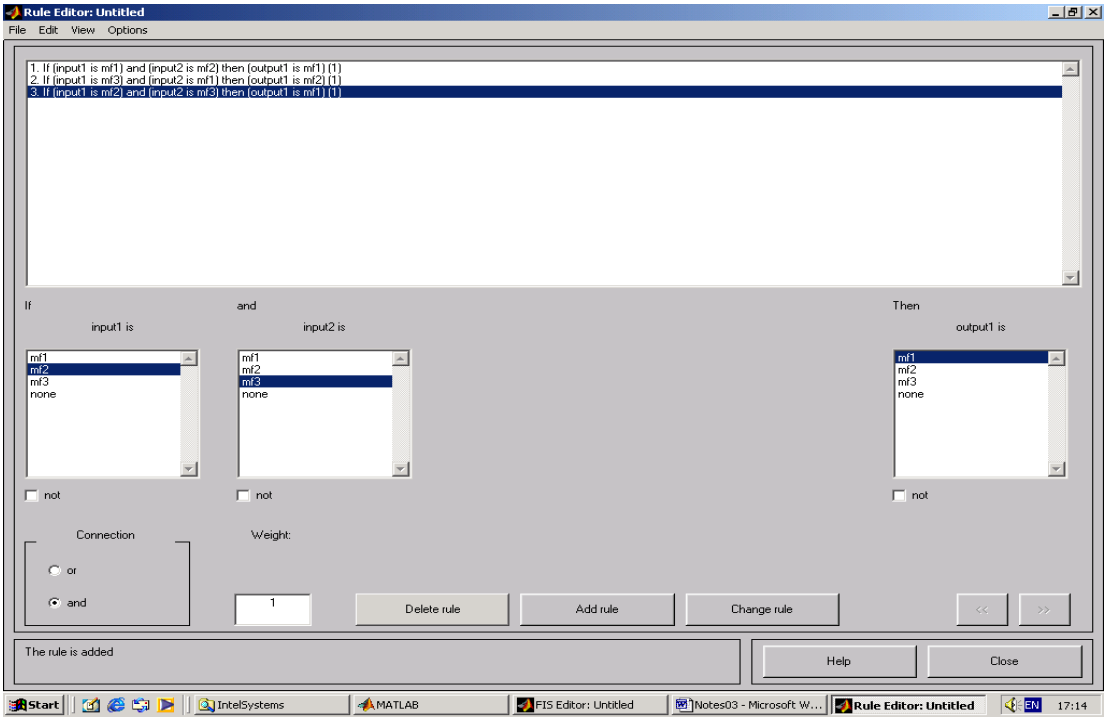
Τα βασικά εργαλεία απεικονίζονται στα Σχήματα 2.1(α) - (ε), ενώ η λειτουργία αυτών των βασικών εργαλείων αναλύεται στη συνέχεια.



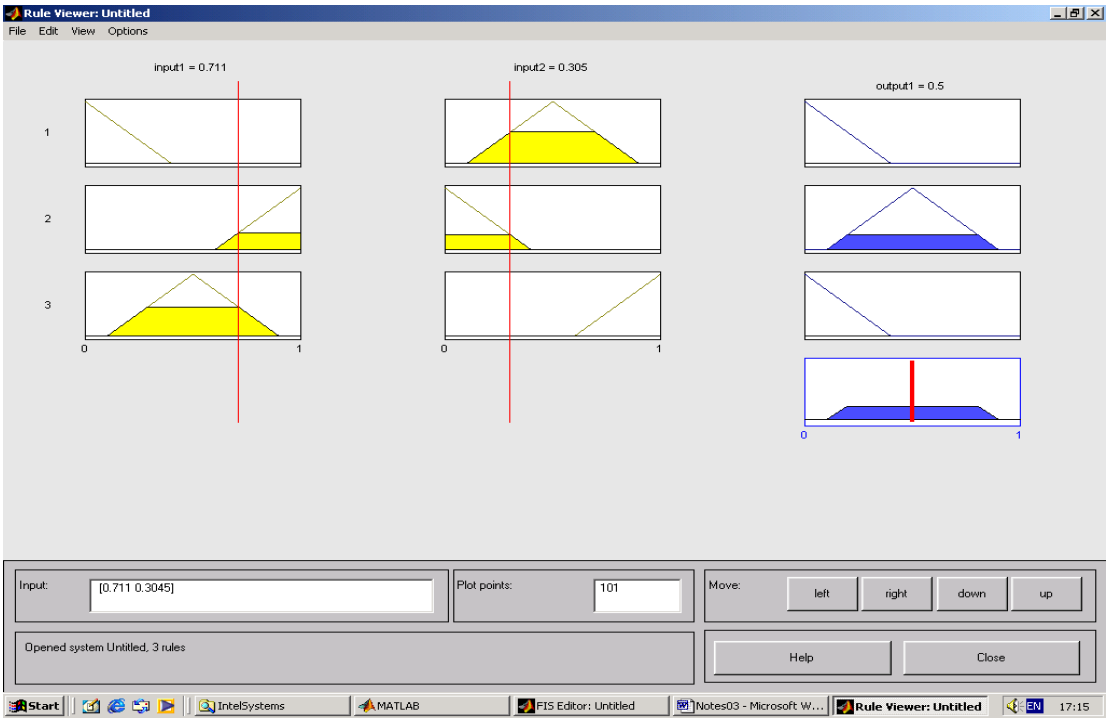
Σχήμα 2.1 (α): FIS Editor



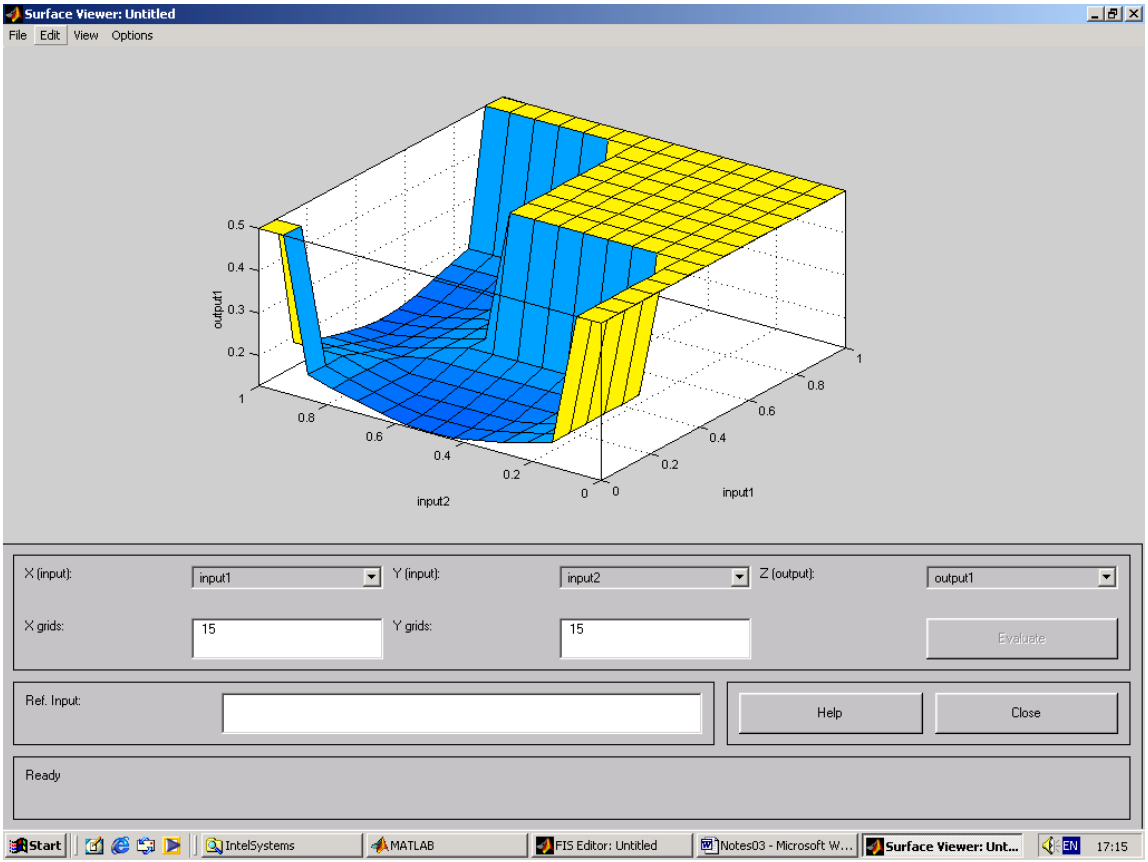
Σχήμα 2.1 (β): Membership Function Editor



Σχήμα 2.1 (γ): Rule Editor



Σχήμα 2.1 (δ): Rule Viewer



Σχήμα 2.1 (ε): Surface Viewer

Το εργαλείο **FIS Editor** χρησιμοποιείται για να οριστεί η γενική αρχιτεκτονική του συστήματος: Το πλήθος των εισόδων /εξόδων, τα ονόματα των εισόδων /εξόδων, κλπ.

Το εργαλείο **Membership Function Editor** χρησιμοποιείται για να οριστούν οι μορφές των συναρτήσεων συμμετοχής για κάθε μεταβλητή εισόδου /εξόδου.

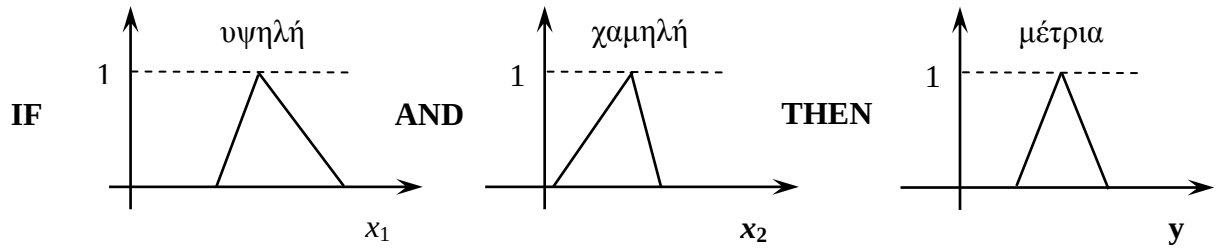
Το εργαλείο **Rule Editor** χρησιμοποιείται για να οριστούν οι λεκτικοί κανόνες λειτουργίας του συστήματος.

Τα εργαλεία **Rule Viewer** και **Surface Viewer** δίνουν μια εποπτική εικόνα του συστήματος. Το εργαλείο Rule Viewer δείχνει ποιοι κανόνες είναι ενεργοί, πως οι συγκεκριμένες μορφές των συναρτήσεων συμμετοχής καθορίζουν το αποτέλεσμα, κλπ. Το εργαλείο Surface Viewer δείχνει την εξάρτηση μιας εξόδου από τις εισόδους.

Τα πέντε βασικά εργαλεία του GUI ανταλλάσσουν πληροφορίες μεταξύ τους. Συγκεκριμένα, αν τα ονόματα των συναρτήσεων συμμετοχής αλλάξουν χρησιμοποιώντας το εργαλείο *Membership Function Editor*, τότε αυτές οι αλλαγές αντανακλώνται στους κανόνες που βλέπουμε στο εργαλείο *Rule Editor*. Η λεπτομερής χρήση του κάθε εργαλείου παρουσιάζεται παρακάτω σε ένα συγκεκριμένο παράδειγμα.

2.4 Υπολογισμός της Εξόδου Ασαφούς Συστήματος

Θεωρείστε έναν ασαφή κανόνα ο οποίος χαρακτηρίζεται από δύο λεκτικές μεταβλητές εισόδου και μια λεκτική μεταβλητή εξόδου, όπως στο Σχήμα 2.2.



Σχήμα 2.2: Ένας ασαφής κανόνας δύο λεκτικών εισόδων και μιας λεκτικής εξόδου.

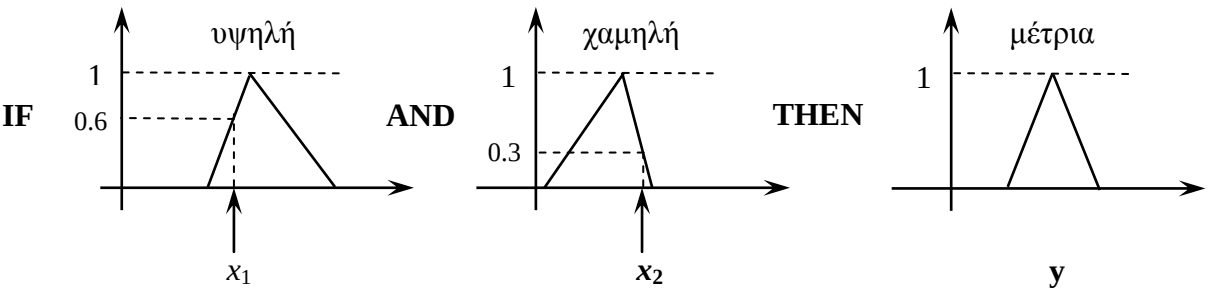
Στο πρόγραμμα εφαρμογής 'Fuzzy' του MATLAB υπάρχουν πέντε εργαλεία για τον υπολογισμό της αριθμητικής εξόδου ενός ασαφούς συστήματος από αριθμητικές εισόδους χρησιμοποιώντας το σύνολο των ασαφών κανόνων που διατίθενται στη βάση της γνώσης του ασαφούς συστήματος:

1) Ασαφοποίηση των αριθμητικών εισόδων,
2) Εφαρμογή ασαφών τελεστών (AND, OR),
3) Ασαφής συνεπαγωγή,
4) Συνάθροιση των μερικών εξόδων, και
5) Αποασαφοποίηση.

Τα παραπάνω εργαλεία περιγράφονται στη συνέχεια.

2.4.1 Ασαφοποίηση των αριθμητικών εισόδων

Αρχικά οι αριθμητικές είσοδοι ασαφοποιούνται όπως φαίνεται στο παρακάτω Σχήμα 2.3 όπου δείχνεται ένας κανόνας συζευγμένων (AND) προτάσεων-αιτίων Συγκεκριμένα, για κάθε αριθμητική είσοδο x_i , i=1,2 υπολογίζεται ο βαθμός συμμετοχής στο αντίστοιχο ασαφές σύνολο αιτίου του κάθε κανόνα. Ο βαθμός συμμετοχής σε κάποιο ασαφές σύνολο μπορεί να θεωρηθεί ως ο βαθμός "ενεργοποίησης/αλήθειας" της αντίστοιχης συζευγμένης πρότασης-αιτίου. Έτσι, στο Σχήμα 2.3, ο βαθμός αλήθειας της πρότασης-αιτίου "η μεταβλητή x_1 είναι υψηλή" είναι 0.6, ενώ ο βαθμός αλήθειας της πρότασης- αιτίου "η μεταβλητή x_2 είναι χαμηλή" είναι 0.3.



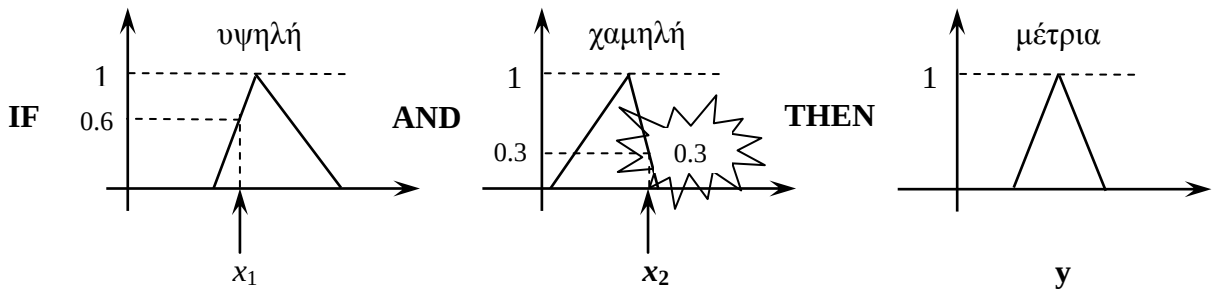
Σχήμα 2.3: Ασαφοποίηση δύο αριθμητικών εισόδων x_1 και x_2.

2.4.2 Εφαρμογή ασαφών τελεστών (AND, OR)

Με την ασαφοποίηση της κάθε εισόδου έχει υπολογιστεί ο βαθμός ικανοποίησης κάθε αιτίου (στο τμήμα *if*) του κανόνα. Εάν το τμήμα *if* του κανόνα είναι η σύζευξη περισσοτέρων από ένα αίτια, τότε εφαρμόζεται κάποιος ασαφής τελεστής για να υπολογιστεί συνολικά ένας αριθμός ο οποίος εκφράζει το συνολικό βαθμό ικανοποίησης του κανόνα. Αυτός ο αριθμός θα εφαρμοστεί τελικά στο αποτέλεσμα (στο τμήμα *then*) του κανόνα.

Συνεπώς, οι είσοδοι σε έναν ασαφή τελεστή είναι οι βαθμοί ικανοποίησης κάθε αιτίου (στο τμήμα *if*) του κανόνα. Η έξοδος είναι συνοπτικά ένας βαθμός ικανοποίησης που εκφράζει συνολικά το αίτιο του κανόνα. Στο πρόγραμμα εφαρμογής 'Fuzzy' του MATLAB υπάρχουν διαθέσιμοι δύο τελεστές τύπου AND: 1) min (του ελαχίστου), και 2) prod (του γινομένου). Επίσης υπάρχουν διαθέσιμοι δυο τελεστές τύπου OR: 1) max (του μεγίστου), και 2) probor. Ο τελευταίος τελεστής (που είναι γνωστός και ως αλγεβρικό άθροισμα) υπολογίζεται σύμφωνα με την εξίσωση 'probor(a,b)=a+b-a*b'.

Στο Σχήμα 2.4 φαίνεται ένα παράδειγμα υπολογισμού με τελεστή τύπου AND (min). Τα δύο διαφορετικά τμήματα του αιτίου παράγουν τις ασαφείς τιμές 0.6 και 0.3, αντίστοιχα. Ο ασαφής τελεστής 'AND' επιλέγει το ελάχιστο από αυτές τις δύο τιμές, δηλαδή το 0.3.

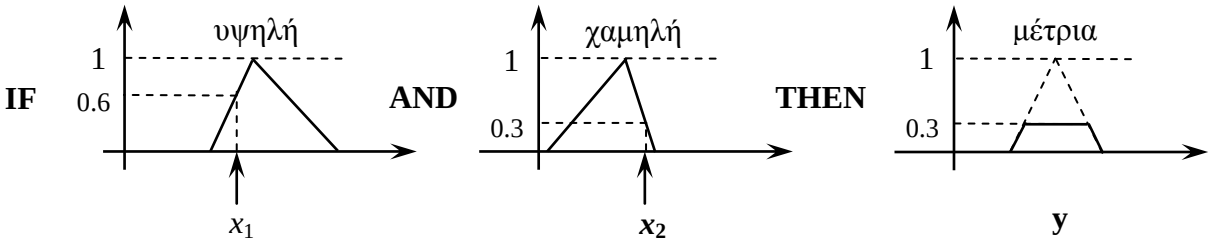


Σχήμα 2.4: Εφαρμογή του ασαφούς τελεστή AND (min).

2.4.3 Ασαφής συνεπαγωγή

Το αποτέλεσμα της ασαφούς συνεπαγωγής κατά Mamdani για κάθε κανόνα είναι το αντίστοιχο ασαφές σύνολο (αποτέλεσμα του κανόνα) το ύψος του οποίου είναι μικρότερο από 1 όπως έχει καθοριστεί από την προηγούμενη εφαρμογή ασαφών τελεστών. Η διαδικασία ασαφούς συνεπαγωγής με ασαφή τελεστή 'AND' φαίνεται στο Σχήμα 2.5 Χαρακτηριστικό είναι ότι η διαδικασία συνεπαγωγής πραγματοποιείται παράλληλα για όλους τους κανόνες.

Σημειώστε ότι το πρόγραμμα εφαρμογής 'Fuzzy' του MATLAB παρέχει τη δυνατότητα καθορισμού ενός συντελεστή βάρους (έναν αριθμό από το 0 ως το 1) για κάθε κανόνα. Έτσι, ένας συντελεστής βάρους εφαρμόζεται στο αποτέλεσμα που παράγεται από τα αίτια. Τυπικά, σ' αυτό το εργαστήριο, θα χρησιμοποιηθούν συντελεστές βάρους ίσοι με τη μονάδα.

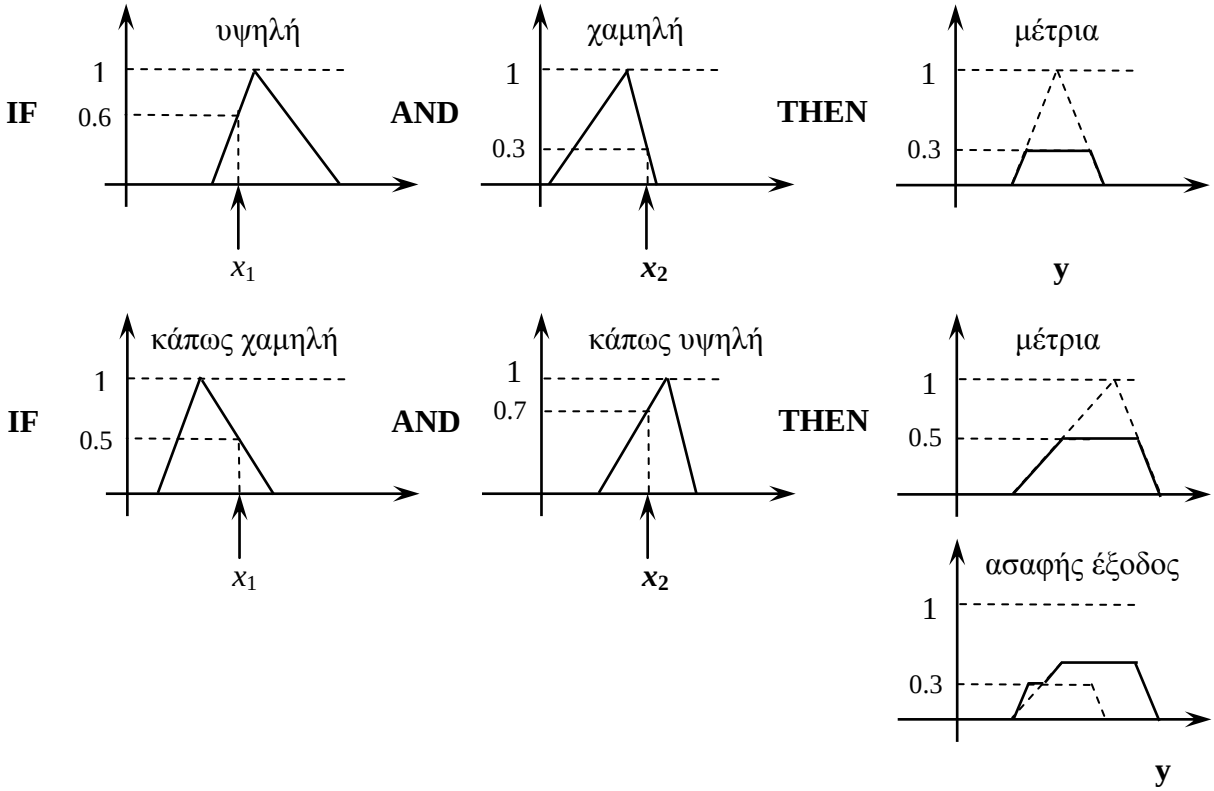


Σχήμα 2.5: Αποτέλεσμα ασαφούς συνεπαγωγής κατά Mamdani.

2.4.4 Συνάθροιση μερικών αποτελεσμάτων

Συνάθροιση είναι η διαδικασία κατά την οποία τα ασαφή σύνολα που αντιπροσωπεύουν την έξοδο του κάθε κανόνα συνδυάζονται σε ένα μοναδικό ασαφές σύνολο ανά λεκτική μεταβλητή εξόδου. Εκτελείται μόνο μια συνάθροιση για κάθε μεταβλητή εξόδου. Η είσοδος στην διαδικασία συνάθροισης είναι ένα σύνολο από περικομμένες συναρτήσεις εξόδου οι οποίες παράγονται από την μέθοδο συνεπαγωγής για κάθε κανόνα.

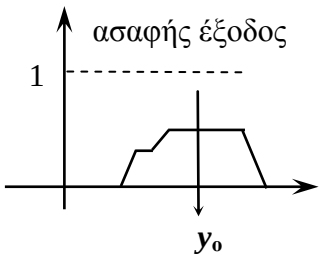
Το πρόγραμμα εφαρμογής 'Fuzzy' του MATLAB υποστηρίζει τρεις μεθόδους συνάθροισης: 1) τη max (μεγίστου), 2) τη probor (πιθανότητας), και 3) τη sum (αθροίσματος). Στο παρακάτω Σχήμα 2.6 δείχνεται ο τρόπος υπολογισμού του ασαφούς συνόλου για μια μεταβλητή εξόδου χρησιμοποιώντας τη συνάθροιση μεγίστου (max).



Σχήμα 2.6: Αποτέλεσμα συνάθροισης μερικών αποτελεσμάτων δύο κανόνων.

2.4.5 Αποασαφοποίηση

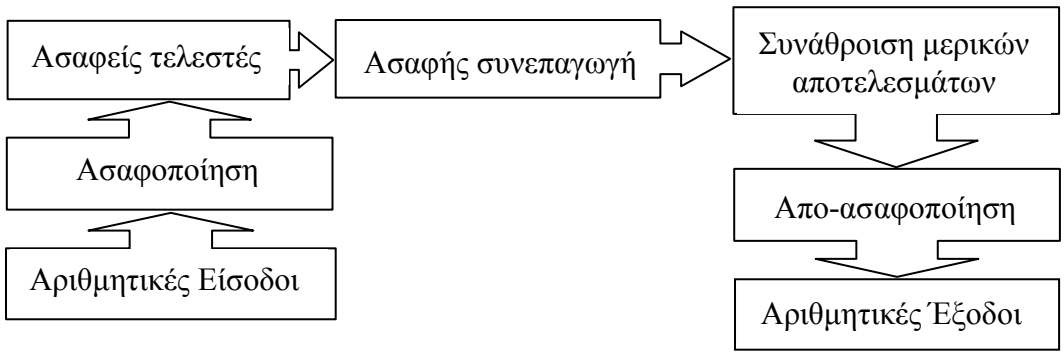
Από ένα ασαφές σύνολο που προέκυψε από την συνάθροιση μερικών αποτελεσμάτων του προηγούμενου βήματος τώρα θα υπολογιστεί ένας πραγματικός αριθμός με μια τεχνική απο-ασαφοποίησης. Η πιο δημοφιλής τεχνική απο-ασαφοποιήσης είναι υπολογίζοντας το "κέντρο βάρους" ενός ασαφούς συνόλου όπως φαίνεται στο Σχήμα 2.7. Το πρόγραμμα εφαρμογής 'Fuzzy' του MATLAB υποστηρίζει πέντε τεχνικές απο-ασαφοποίησης: 1) κέντρου βάρους, 2) διχοτόμησης, 3) μέσου των μεγίστων, 4) μεγαλύτερου από τα μέγιστα, και 5) μικρότερου από τα μέγιστα.



Σχήμα 2.7: Απο-ασαφοποίηση "κέντρου βάρους": Το ασαφές σύνολο εξόδου θεωρείται σαν μια ομογενής μάζα το κέντρο βάρους της οποίας υπολογίζεται. Τελικά, το ασαφές σύνολο εξόδου απεικονίζεται στον αριθμό y_o ο οποίος ορίζεται από την κατακόρυφη ευθεία που διέρχεται από το κέντρο βάρους.

2.4.6 Εποπτικό διάγραμμα υπολογισμού της εξόδου ασαφούς συστήματος

Το παρακάτω διάγραμμα του Σχήματος 2.8 συνοψίζει εποπτικά τα προηγούμενα "μερικά βήματα" που εμπλέκονται στον υπολογισμό της αριθμητικής εξόδου ενός ασαφούς συστήματος από αριθμητικές εισόδους.



Σχήμα 2.8: Εποπτικό διάγραμμα παραγωγής αριθμητικής εξόδου σε ένα ασαφές σύστημα από αριθμητικές εισόδους.

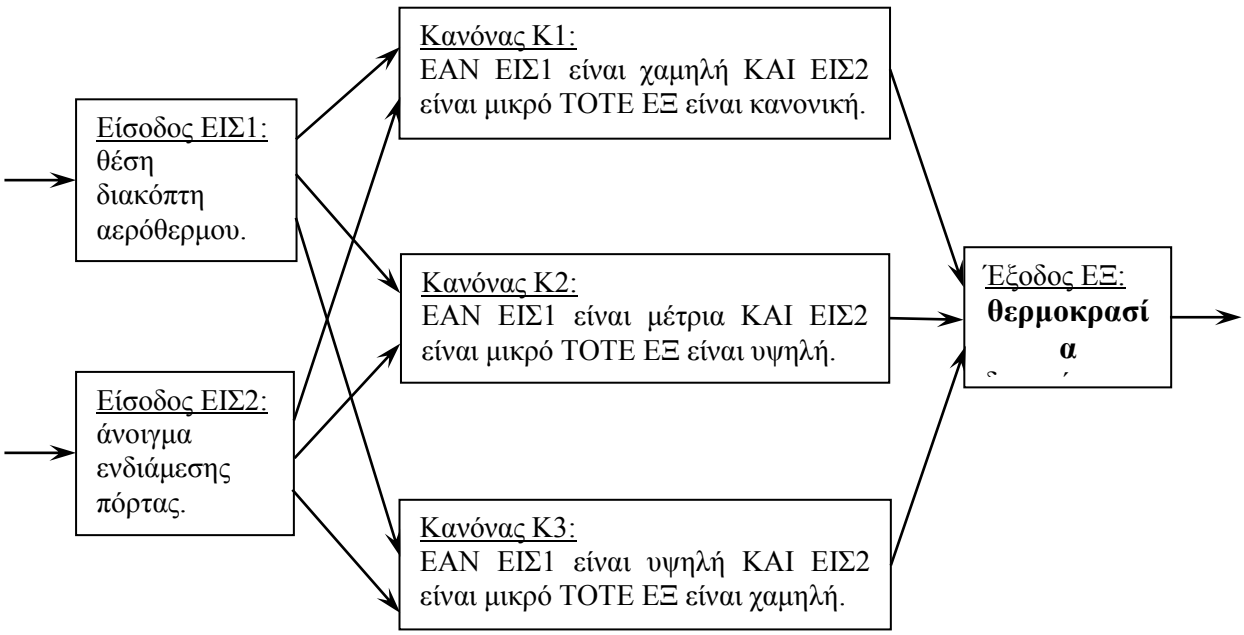
2.5 Τεχνικές Σχεδίασης Ασαφών Συστημάτων

Θα μελετηθούν συστήματα με εισόδους λεκτικές μεταβλητές οι τιμές των οποίων είναι ασαφή σύνολα. Τέτοια συστήματα ονομάζονται "ασαφή συστήματα". Η έξοδος ενός ασαφούς συστήματος είναι είτε λεκτική μεταβλητή (για συστήματα τύπου Mamdani) είτε μια αλγεβρική συνάρτηση (για συστήματα τύπου Sugeno).

2.5.1 Παράδειγμα σχεδίασης με τεχνική τύπου Mamdani

Η σχεδίαση κατά Mamdani είναι από τις πιο συνηθισμένες μεθοδολογίες σχεδίασης ασαφών συστημάτων. Το χαρακτηριστικό της είναι ότι τόσο οι είσοδοι όσο και οι έξοδοι του συστήματος είναι λεκτικές μεταβλητές οι τιμές των οποίων είναι ασαφή σύνολα. Με άλλα λόγια, τόσο οι υποθέσεις όσο και τα συμπεράσματα των κανόνων εμπλέκουν λεκτικές μεταβλητές οι τιμές των οποίων είναι ασαφή σύνολα. Το πρόγραμμα εφαρμογής 'Fuzzy' του MATLAB επιτρέπει τον ορισμό των συναρτήσεων μεταφοράς, οι οποίες μπορεί να είναι π.χ. τρίγωνα, τραπέζια, Gaussian, κλπ. Τελικά, το αποτέλεσμα της εφαρμογής όλων των κανόνων είναι ένα ασαφές σύνολο από το οποίο εξάγεται ένας αριθμός με την τεχνική της από-ασαφοποίησης.

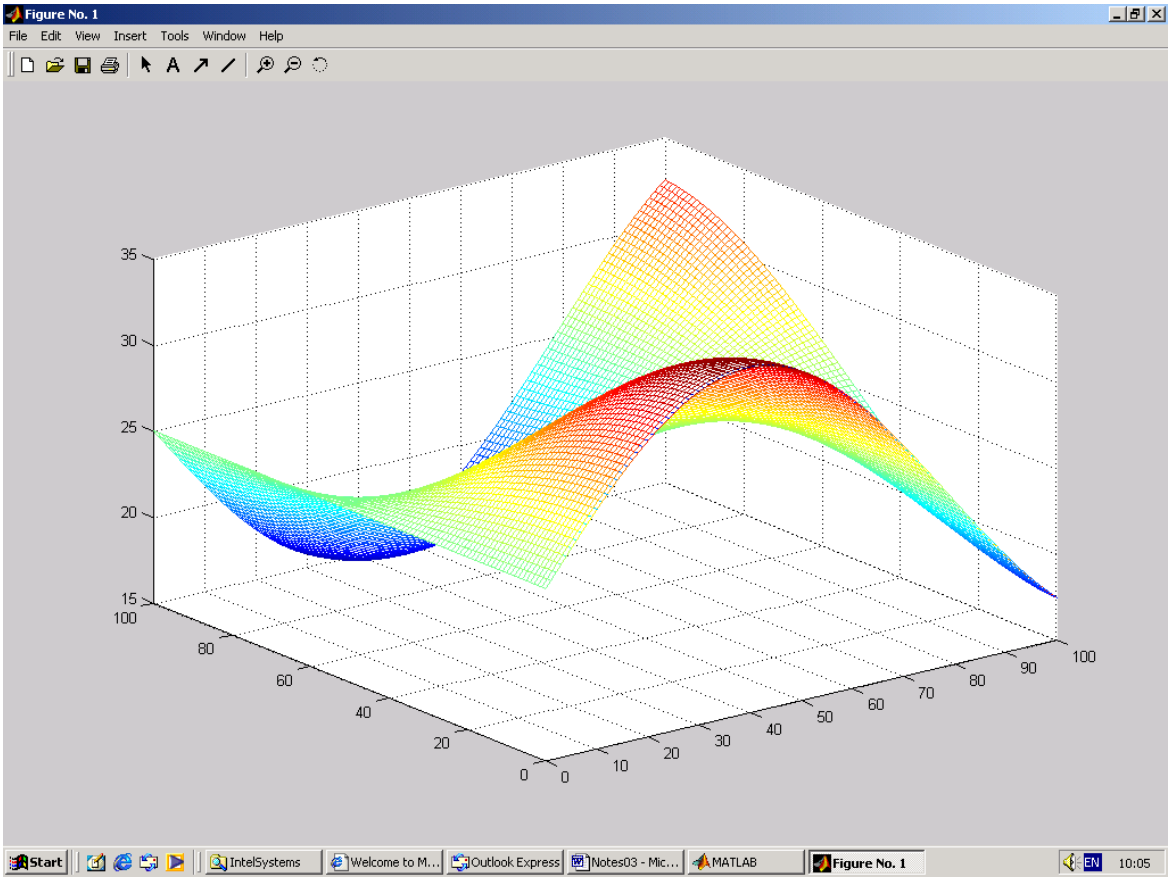
Θα μοντελοποιηθεί το σύστημα ανοιχτού βρόχου "δωμάτιο", το οποίο παρουσιάστηκε στην ενότητα 2.2, με δύο μεταβλητές εισόδου: 1) τη θέση του διακόπτη του αερόθερμου, και 2) το άνοιγμα της ενδιάμεσης πόρτας, και μια μεταβλητή εξόδου: την επιθυμητή θερμοκρασία δωματίου. Η βασική δομή αυτού του συστήματος παρουσιάζεται στο Σχήμα 2.9. Να σημειωθεί ότι η παράλληλη ενεργοποίηση των κανόνων είναι ένα από τα σημαντικότερα πλεονεκτήματα των συστημάτων ασαφούς λογικής (FIS).



Σχήμα 2.9: Μοντέλο FIS φυσικού συστήματος "δωματίου" ανοικτού βρόχου με δύο εισόδους και μια έξοδο.

Υποθέτουμε ότι και οι δύο είσοδοι μεταβάλλονται στο διάστημα 0-100%. Μια εκδοχή της διδιάστατης συνάρτησης ΕΞ (θερμοκρασία δωματίου) ως προς τις μεταβλητές ΕΙΣ1 (θέση διακόπτη αερόθερμου) και ΕΙΣ2 (άνοιγμα ενδιάμεσης πόρτας) φαίνεται στο Σχήμα 2.10 όπως προέκυψε από τις παρακάτω εντολές του MATLAB:

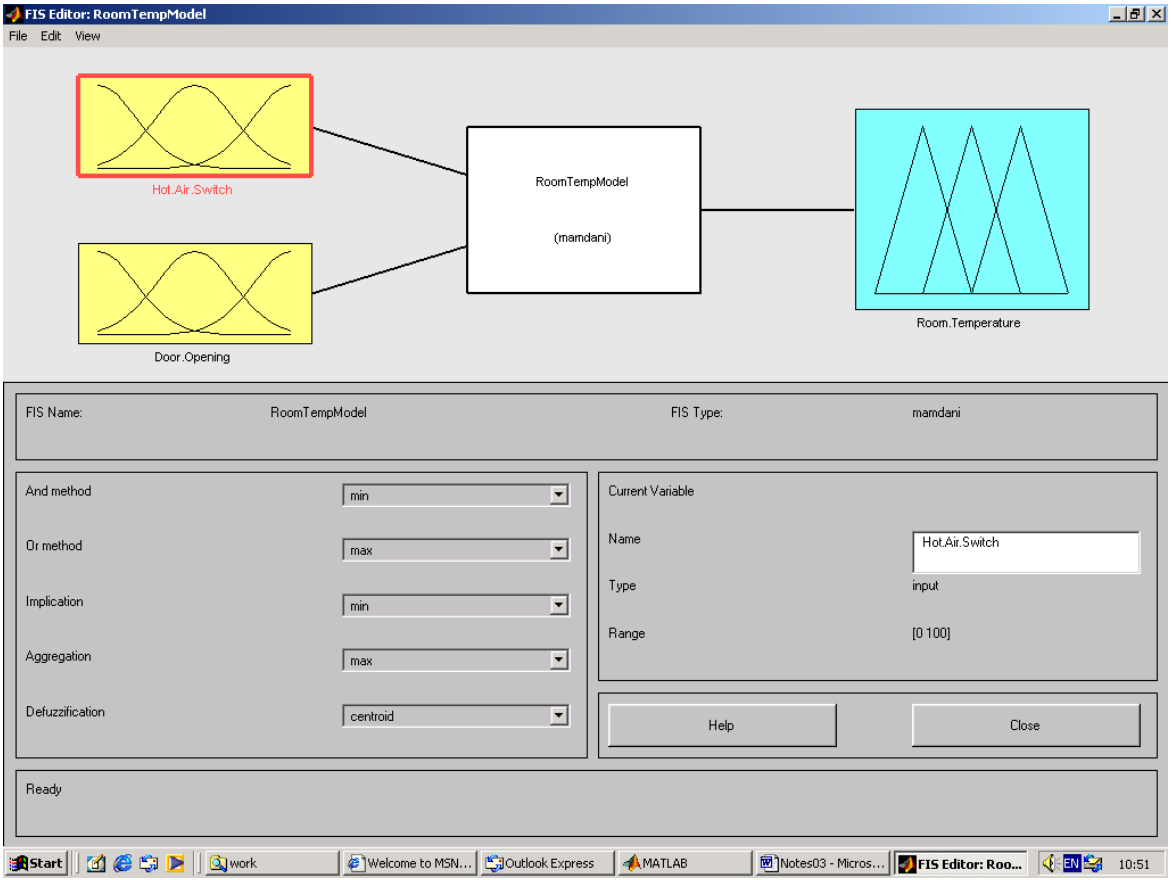
```
>> x=0:1:100;
>> y=0:1:100;
>> [X,Y]=meshgrid(x,y);
>> Z= 25+10*sin(X/25).*cos(Y/33);
>> mesh(X,Y,Z);
```



Σχήμα 2.10: Μια εκδοχή της μη-γραμμικής διδιάστατης συνάρτηση ΕΞ(ΕΙΣ1,ΕΙΣ2).

Χαρακτηριστική είναι η μη-γραμμικότητα της συνάρτησης ΕΞ(ΕΙΣ1,ΕΙΣ2). Για απλότητα θα χρησιμοποιηθούν έξι κανόνες για να προσεγγιστεί η συνάρτηση ΕΞ(ΕΙΣ1,ΕΙΣ2). Να σημειωθεί ότι ο αριθμός των κανόνων μπορεί να αυξηθεί για μεγαλύτερη ακρίβεια προσέγγισης.

Ξεκινάμε να δουλεύουμε με τα εργαλεία του GUI, για να κατασκευάσουμε ένα σύστημα ασαφούς λογικής το οποίο να προσεγγίζει τη συνάρτηση ΕΞ(ΕΙΣ1,ΕΙΣ2) του Σχήματος 2.10. Αρχικά καλείται ο FIS Editor, από την γραμμή των εντολών του MATLAB πληκτρολογώντας fuzzy, και τελικά ορίζεται το ασαφές σύστημα 'RoomTempModel' με 2 εισόδους και 1 έξοδο όπως φαίνεται στο Σχήμα 2.11.



Σχήμα 2.11: Ορισμός του ασαφούς συστήματος 'RoomTempModel' με τον FIS Editor.

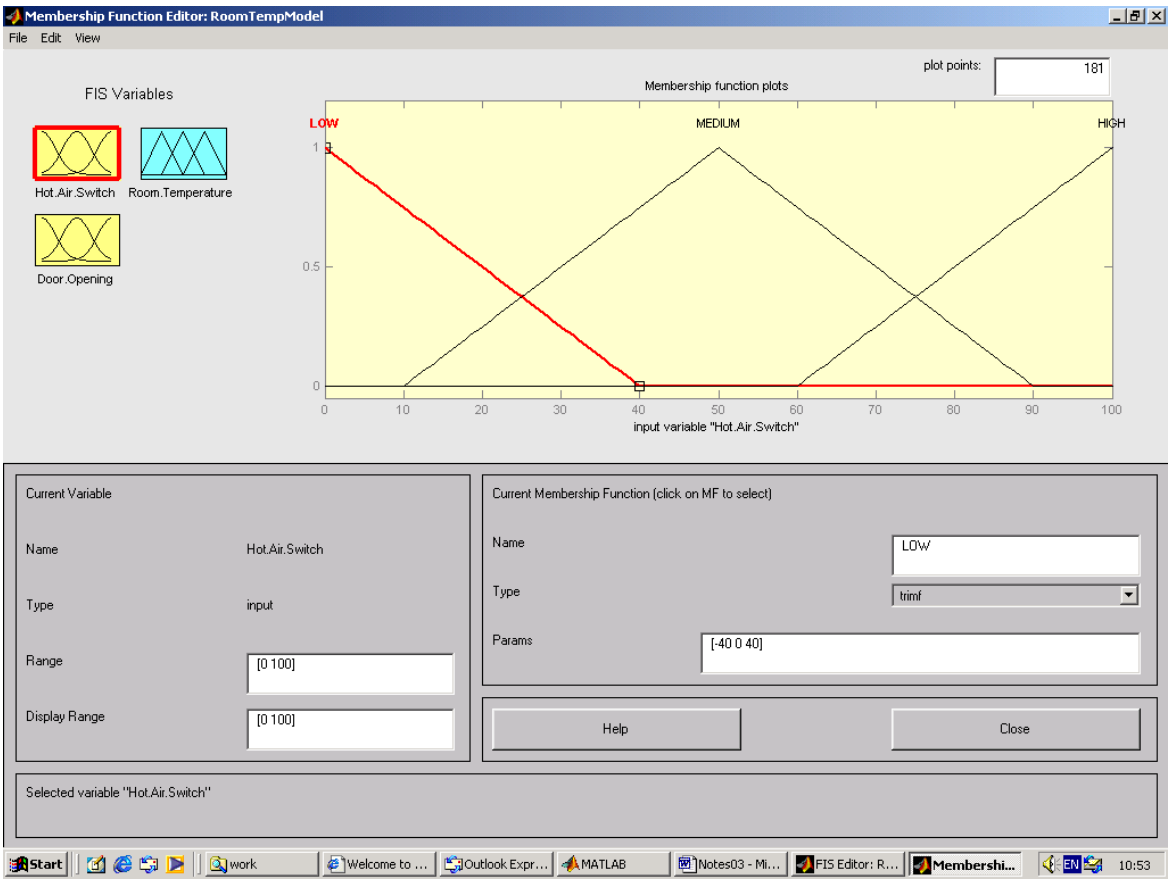
Ο FIS Editor διαθέτει όλα τα βασικά πλήκτρα για τον ορισμό ενός συστήματος ασαφούς λογικής. Στο επάνω μέρος του FIS Editor εμφανίζεται ένα block diagram όπου στο αριστερό μέρος δείχνονται οι μεταβλητές εισόδου (κίτρινα κουτάκια), ενώ στο δεξιό μέρος δείχνονται οι μεταβλητές εξόδου (μπλε κουτάκι). Τα υποδείγματα των συναρτήσεων συμμετοχής που παρουσιάζονται στα κουτάκια εισόδου/ εξόδου είναι εικονικά και δεν απεικονίζουν τις ακριβείς μορφές των συναρτήσεων συμμετοχής. Στο μέσον του block diagram δείχνεται το όνομα και ο τύπος του ασαφούς συστήματος (λευκό κουτάκι). Στο Σχήμα 2.11 χρησιμοποιείται ένα ασαφές σύστημα τύπου Mamdani με το όνομα 'RoomTempModel'. Να σημειωθεί ότι το όνομα και ο τύπος του ασαφούς συστήματος φαίνονται επίσης και στην πρώτη σειρά κάτω από το block diagram στον FIS Editor. Κάτω από το όνομα του ασαφούς συστήματος (στο αριστερό μέρος της εικόνας) υπάρχει ένας πίνακας με pop-up-menus που μας επιτρέπουν να ορίσουμε επί μέρους διαδικασίες ενός ασαφούς συστήματος. Κάτω από τον τύπο του ασαφούς συστήματος (στο δεξί μέρος της εικόνας) υπάρχει ένας πίνακας όπου ορίζονται το όνομα των μεταβλητών εξόδου ή εισόδου, τις σχετιζόμενες με αυτές βαθμούς συμμετοχής, και το φάσμα τιμών τους. Κάτω από αυτόν τον πίνακα υπάρχουν τα πλήκτρα help και close με προφανή χρήση. Τέλος, στο κατώτατο μέρος υπάρχει μια γραμμή η οποία δίνει πληροφορίες σχετικές με το σύστημα.

Για να ξεκινήσουμε τον σχεδιασμό ενός συστήματος πληκτρολογούμε fuzzy στη σελίδα εργασίας του MATLAB. Το γενικό περιβάλλον του FIS Editor (χωρίς τίτλο) ανοίγει με μια είσοδο input1 και μια έξοδο output1. Σ' αυτό το παράδειγμα θα δημιουργήσουμε ένα σύστημα δύο εισόδων μιας εξόδου. Έτσι, πηγαίνουμε στο Edit και επιλέγουμε διαδοχικά Add Variable/Input. Ένα δεύτερο κίτρινο κουτάκι input2 θα εμφανιστεί. Οι δύο είσοδοι που θα έχουμε στο παράδειγμα μας θα είναι Hot.Air.Switch (θέση διακόπτη αερόθερμου) και Door.Opening (άνοιγμα ενδιάμεσης πόρτας), ενώ η έξοδος μας θα είναι η Room.Temperature (θερμοκρασία δωματίου). Τα ονόματα των μεταβλητών μπορούν να αλλαχτούν με προφανή τρόπο. Ο χώρος εργασίας σώζεται με το όνομα 'RoomTempModel' επιλέγοντας File/Export/To Disk και δίνοντας το όνομα RoomTempModel. Αντίστροφα, κάποιο ασαφές σύστημα το οποίο έχει σωθεί στον δίσκο μπορεί να φορτωθεί επιλέγοντας File/Import/From Disk.

Στην συνέχεια ορίζουμε τις συναρτήσεις συμμετοχής κάθε λεκτικής μεταβλητής εισόδου/εξόδου. Για να γίνει αυτό καλούμε το Membership Function Editor με έναν από τους παρακάτω τρεις τρόπους:

1. Επιλέγοντας την είσοδο/έξοδο που μας ενδιαφέρει (δηλ. το αντίστοιχο κουτάκι στον FIS Editor), κατόπιν πιέζοντας το αριστερό κουμπί του ποντικιού επιλέγουμε Edit/Membership Functions ...
2. Κάνοντας διπλό κλικ στην είσοδο/έξοδο που μας ενδιαφέρει (δηλ. στο αντίστοιχο κουτάκι στον FIS Editor), και
3. Πληκτρολογώντας mfedit στην γραμμή εντολών.

Ένα επί μέρους αποτέλεσμα φαίνεται στο Σχήμα 2.12.

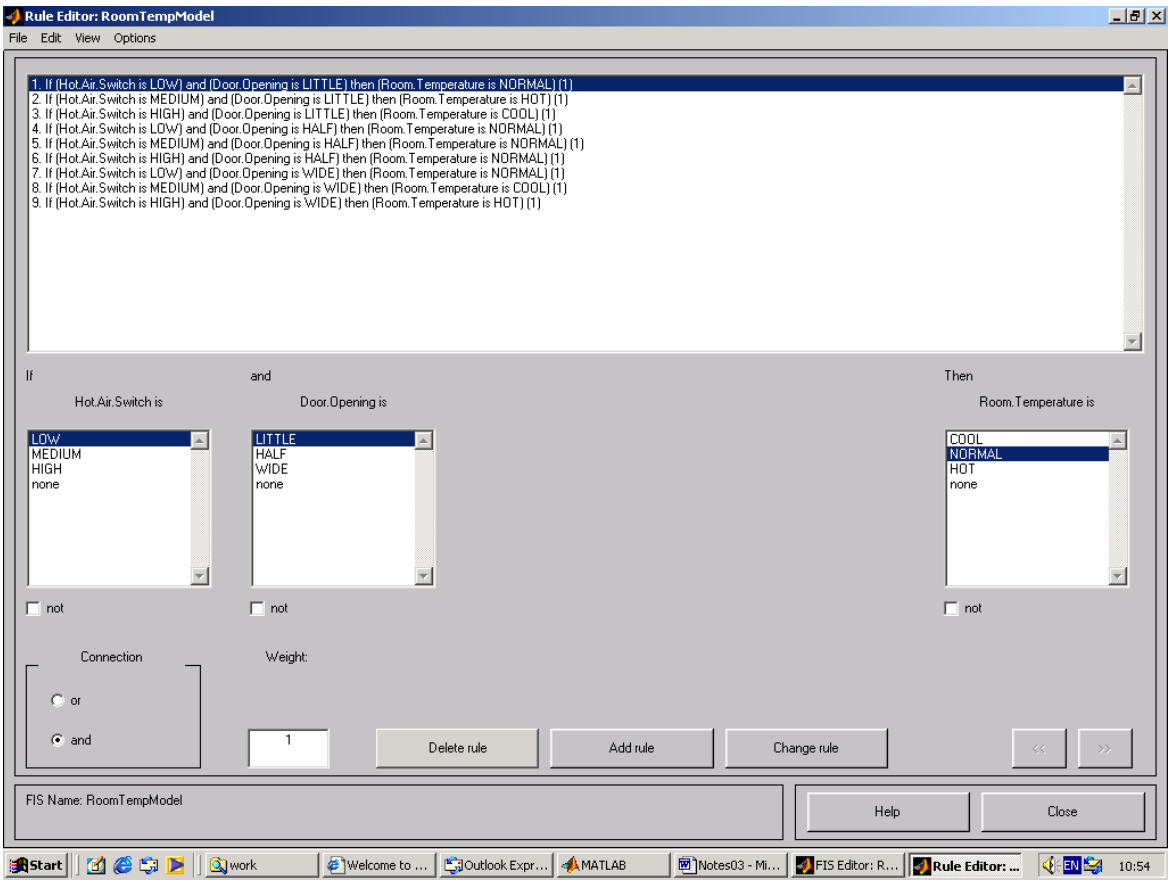


Σχήμα 2.12: Το γραφικό περιβάλλον του Membership Function Editor.

Στον Membership Function Editor καθορίσουμε ασαφή σύνολα για την είσοδο/έξοδο που μας ενδιαφέρει. Επιλέγοντας Edit/Add MFS … μπορούμε να προσθέσουμε επιπλέον ασαφή σύνολα. Επίσης μπορούμε να σβήσουμε κάποιο ασαφές σύνολο επιλέγοντας Edit/Remove Selected MF …, κλπ. Ένα ασαφές σύνολο μπορεί να επιλεγεί με το ποντίκι και, στη συνέχεια, μπορούμε να αλλάξουμε το όνομά του, τον τύπο του (type), και τις παραμέτρους που το ορίζουν (params).

Οι συναρτήσεις συμμετοχής των ασαφών συνόλων που έχουν οριστεί εμφανίζονται στο κυρίως τμήμα του γραφικού περιβάλλοντος. Μια συνάρτηση συμμετοχής που έχει επιλεγεί μπορεί να διαμορφωθεί με δύο τρόπους. Πρώτον, διαστέλλοντας ή/και συστέλλοντας μια συνάρτηση συμμετοχής με τη βοήθεια του ποντικιού. Δεύτερον, ορίζοντας με το χέρι τις παραμέτρους (params) μιας συνάρτηση συμμετοχής.

Τώρα είμαστε έτοιμοι να δημιουργήσουμε τους κανόνες. Για να εμφανιστεί ο Rule Editor, είτε επιλέγουμε Edit/Rules … είτε κάνουμε διπλό κλικ στο λευκό κουτάκι με το όνομα και τον τύπος του ασαφούς συστήματος. Ο Rule Editor φαίνεται στο Σχήμα 2.13.



Σχήμα 2.13: Το γραφικό περιβάλλον του Rule Editor.

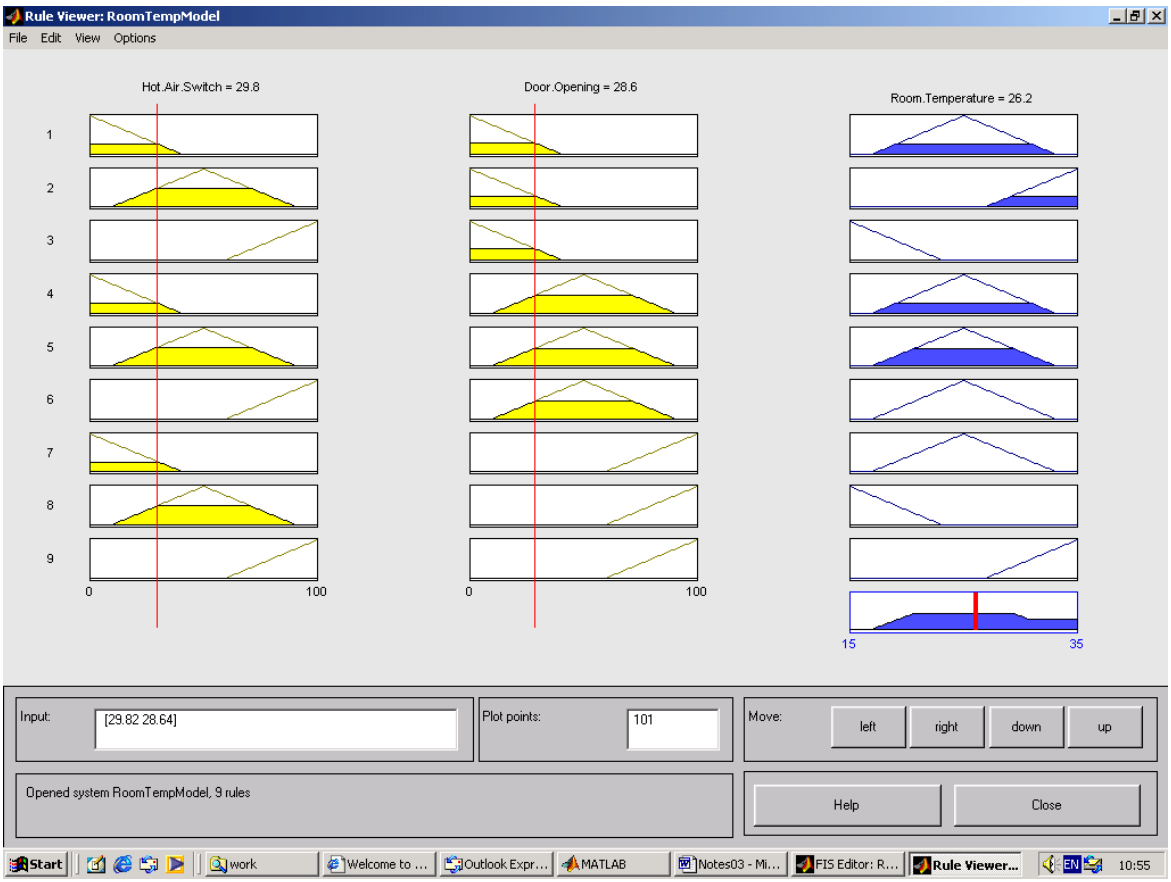
Η δημιουργία των κανόνων με τον Rule Editor είναι μία εμπειρική διαδικασία. Χρησιμοποιήθηκαν οι ακόλουθοι 9 κανόνες :

1. If (Hot.Air.Switch is LOW) and (Door.Opening is LITTLE) then (Room.Temperature is NORMAL) (1)

2. If (Hot.Air.Switch is MEDIUM) and (Door.Opening is LITTLE) then (Room.Temperature is HOT) (1)
3. If (Hot.Air.Switch is HIGH) and (Door.Opening is LITTLE) then (Room.Temperature is COOL) (1)
4. If (Hot.Air.Switch is LOW) and (Door.Opening is HALF) then (Room.Temperature is NORMAL) (1)
5. If (Hot.Air.Switch is MEDIUM) and (Door.Opening is HALF) then (Room.Temperature is NORMAL) (1)
6. If (Hot.Air.Switch is HIGH) and (Door.Opening is HALF) then (Room.Temperature is NORMAL) (1)
7. If (Hot.Air.Switch is LOW) and (Door.Opening is WIDE) then (Room.Temperature is NORMAL) (1)
8. If (Hot.Air.Switch is MEDIUM) and (Door.Opening is WIDE) then (Room.Temperature is COOL) (1)
9. If (Hot.Air.Switch is HIGH) and (Door.Opening is WIDE) then (Room.Temperature is HOT) (1)

Οι αριθμοί εντός των τελευταίων παρενθέσεων σε κάθε κανόνα αντιστοιχούν στο βάρος κάθε κανόνα. Τυπικά τα βάρη θεωρούνται να είναι μοναδιαία (1).

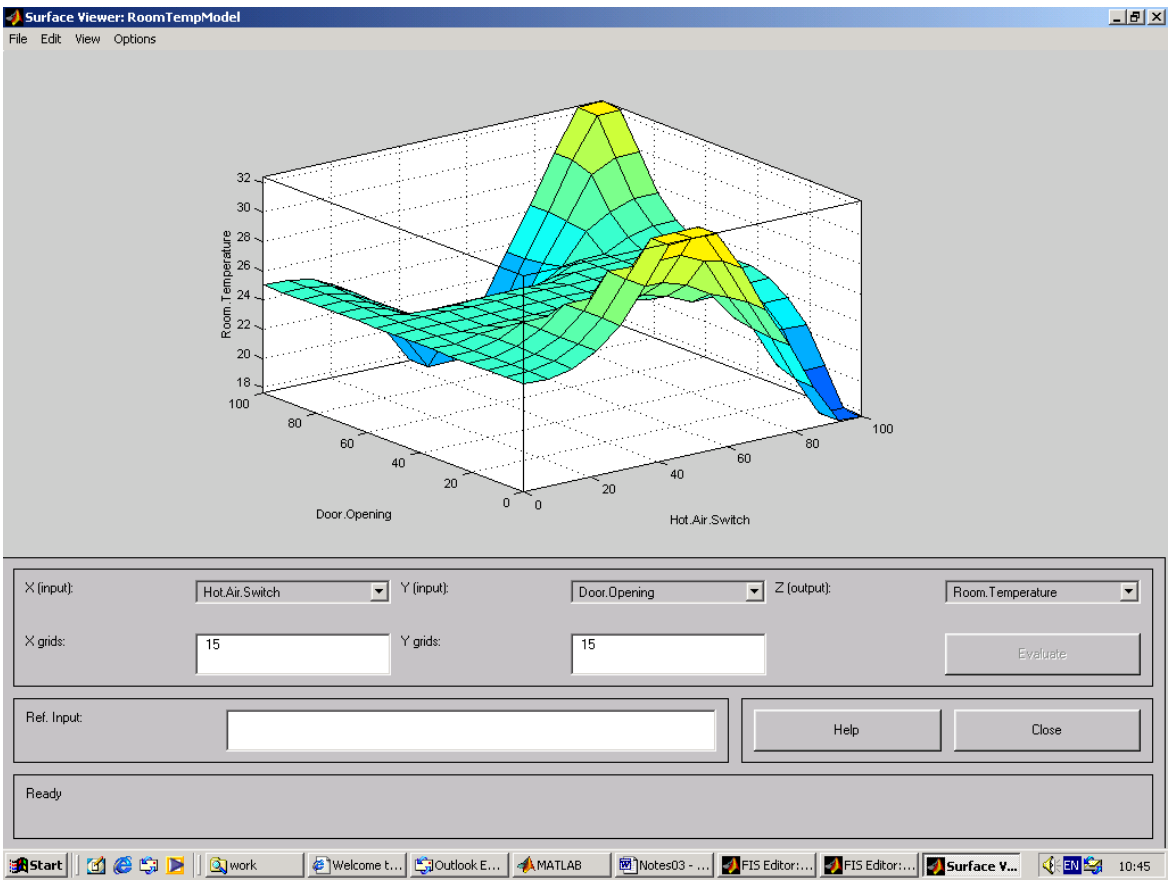
Σε αυτό το σημείο το ασαφές σύστημα έχει οριστεί πλήρως. Ο FIS Editor προσφέρει τη δυνατότητα για δύο "οπτικές πιστοποιήσεις" του ασαφούς συστήματος. Συγκεκριμένα, υπάρχουν οι δυνατότητες επιλογής 1) του View/Rules, και 2) του View/Surface. Ο Rule Viewer φαίνεται στο Σχήμα 2.14



Σχήμα 2.14: Το γραφικό περιβάλλον του Rule Viewer.

Ο Rule Viewer δείχνει το σύνολο των ασαφών κανόνων. Στο Σχήμα 2.14 φαίνονται συνολικά 28 κουτάκια. Τα 3 κουτάκια κατά μήκος της κορυφής αντιστοιχούν στον πρώτο κανόνα. Κάθε κανόνας αντιστοιχεί σε μια γραμμή από κουτάκια. Κάθε στήλη αντιστοιχεί σε μια μεταβλητή. Οι πρώτες 2 στήλες από κουτάκια αντιστοιχούν στο τμήμα if κάθε κανόνα. Η τρίτη στήλη από κουτάκια αντιστοιχεί στο τμήμα then κάθε κανόνα. Αν πιεστεί το αριστερό πλήκτρο του ποντικιού μια φορά επάνω στον αριθμό του κανόνα, ο κανόνας αυτός θα φανεί στο κάτω μέρος της εικόνας. Το δέκατο κουτάκι της τρίτης στήλης δείχνει το αποτέλεσμα της συνάθροισης των μερικών αποτελεσμάτων όλων των ενεργών κανόνων.

Αριθμητικά δεδομένα εισόδου μπορούν να εισαχθούν στο ασαφές σύστημα είτε μετακινώντας, με το ποντίκι, τις κόκκινες κατακόρυφες γραμμές είτε γράφοντας τα αριθμητικά δεδομένα με το χέρι. Σε κάθε περίπτωση το ασαφές σύστημα κανόνων ενεργοποιείται, κατά τα γνωστά, και τελικά παράγεται στην έξοδο ένας αριθμός. Στο Σχήμα 2.15 φαίνεται ο Surface Viewer.



Σχήμα 2.15: Το γραφικό περιβάλλον του Surface Viewer.

Ο Surface Viewer δείχνει μια διδιάστατη επιφάνεια η οποία αποτελεί προσέγγιση της μη-γραμμικής διδιάστατης συνάρτησης ΕΞ(ΕΙΣ1,ΕΙΣ2) του σχήματος 2.10. Να σημειωθεί ότι ο σκοπός ανάπτυξης του ασαφούς συστήματος σ' αυτήν την άσκηση ήταν ο υπολογισμός μιας ικανοποιητικής προσέγγισης της συνάρτησης ΕΞ(ΕΙΣ1,ΕΙΣ2). Συγκρίνοντας τις επιφάνειες που δείχνονται στα σχήματα 2.10 και 2.15 είναι φανερό ότι αν και υπάρχει μεγάλη ομοιότητα οι δύο επιφάνειες απέχουν από το να είναι ίδιες. Ωστόσο η τελευταία παρατήρηση δεν μειώνει την αξία προσέγγισης μιας συνάρτησης με ένα ασαφές σύστημα κανόνων για τους παρακάτω λόγους:

1. Σε πρακτικές εφαρμογές, η προς προσέγγιση συνάρτηση, δεν προκύπτει από κάποια αλγεβρική έκφραση όπως η προηγούμενη συνάρτηση ΕΞ(ΕΙΣ1,ΕΙΣ2), αλλά η συνάρτηση που θέλουμε να προσεγγίσουμε προκύπτει από δειγματοληφθέντα δεδομένα. Συνεπώς, ένα ασαφές σύστημα παράγει μια ικανοποιητική προσέγγιση.
2. Ένα ασαφές σύστημα υπολογίζει ένα σύνολο λογικών κανόνων για την παραγωγή των εξόδων του από τις εισόδους του, έτσι προτείνεται ένας λογικός τρόπος παραγωγή των αριθμητικών εξόδων από τις αριθμητικές εισόδους του ασαφούς συστήματος.

2.5.2 Σχεδίαση Ασαφούς Συστήματος με τεχνική τύπου Sugeno

Η σχεδίαση κατά Sugeno είναι από τις πιο συνηθισμένες μεθοδολογίες σχεδίασης ασαφών συστημάτων. Το χαρακτηριστικό της σχεδίασης κατά Sugeno είναι ότι οι είσοδοι είναι λεκτικές μεταβλητές οι τιμές των οποίων είναι ασαφή σύνολα, ενώ οι έξοδοι του συστήματος είναι αλγεβρικές εκφράσεις (συνήθως γραμμικές εκφράσεις ή σταθερές). Το αποτέλεσμα της εφαρμογής όλων των κανόνων είναι ένα σύνολο αριθμών οι οποίοι συνυπολογίζονται, ο καθένας με κάποιον συντελεστή βαρύτητας, και τελικά εξάγεται ένας αριθμός. Το πρόγραμμα εφαρμογής 'Fuzzy' του MATLAB επιτρέπει τον ορισμό είτε γραμμικών συναρτήσεων εξόδου είτε σταθερών εξόδου. Το παρακάτω παράδειγμα εξηγεί την σχεδίαση ασαφών συστημάτων με τεχνική τύπου Sugeno.

Σε μια βιομηχανική διαδικασία καθαρισμού νερού δύο μετρήσιμες μεταβλητές 1) το PH του νερού, και 2) η αλκαλικότητα AL του νερού, καθορίζουν την ποσότητα χημικών ουσιών τύπου PAC που θα χρησιμοποιηθούν για τον καθαρισμό μιας ποσότητας νερού. Έστω x_1 είναι η ακριβής μετρούμενη τιμή του PH, x_2 είναι η ακριβής μετρούμενη τιμή του AL, και y είναι η ακριβής ποσότητα PAC που πρέπει να χρησιμοποιηθεί για τον καθαρισμό μιας ποσότητας νερού. Οι παρακάτω τρεις κανόνες χρησιμοποιούνται για τον υπολογισμό της ακριβούς ποσότητας PAC που πρέπει να χρησιμοποιηθεί.

K1. Εάν PH είναι κανονικό και AL είναι χαμηλό τότε $y= 2x_1+x_2+1$.

K2. Εάν PH είναι υψηλό και AL είναι μέτριο τότε $y= x_1+2x_2+3$.

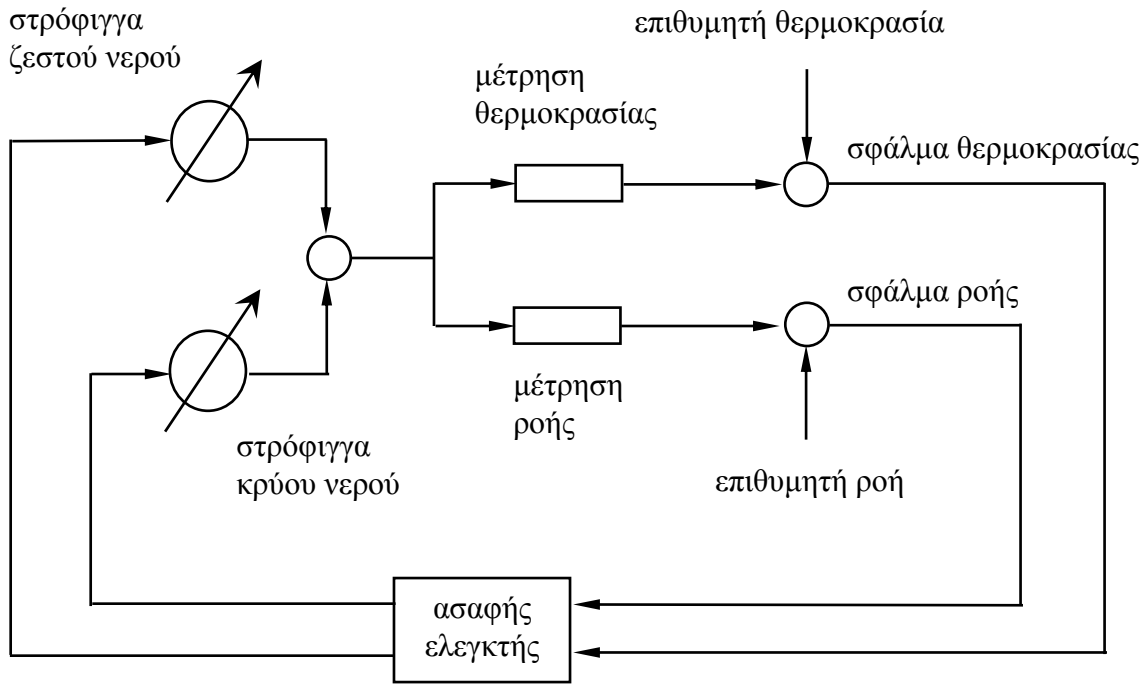
K3. Εάν PH είναι χαμηλό και AL είναι μέτριο τότε $y= 2x_1+3x_2-1$.

Ζητείται να χρησιμοποιηθεί η εφαρμογή 'Fuzzy' του MATLAB και κατόπιν επιλέγοντας File/New FIS…/Sugeno ορίσετε το παραπάνω ασαφές σύστημα τύπου Sugeno.

- Θεωρείστε ότι το διάστημα τιμών της μεταβλητής PH είναι 6-8, ενώ το διάστημα τιμών της μεταβλητής AL είναι 40-60.
- Χρησιμοποιείστε τριγωνικές συναρτήσεις συμμετοχής για τα ασαφή σύνολα που θα ορίσετε.
- Χρησιμοποιώντας τον Rule Viewer βρείτε την ποσότητα y που πρέπει να χρησιμοποιηθεί για τον καθαρισμό μιας ποσότητας νερού για τα παρακάτω ζεύγη τιμών των μεταβλητών PH και AL: (6.5, 50), (7, 45), (7.5, 55).
- Κάνετε συγκρίσεις με τα αποτελέσματα των άλλων φοιτητών. Σχολιάστε.

2.6 Σχεδίαση Ασαφούς Συστήματος με Δύο Εξόδους

Θα σχεδιαστεί ένα ασαφές σύστημα που θα μοντελοποιεί έναν ελεγκτή (εμπειρογνώμονα, άνθρωπο) δύο εισόδων και δύο εξόδων σε ένα σύστημα κλειστού βρόχου. Έστω το παρακάτω υδραυλικό σύστημα κυκλοφορίας ζεστού-κρύου νερού.



Σχήμα 2.16: Ασαφής ελεγκτής συστήματος δύο εισόδων και δύο εξόδων.

Στη θέση του παραπάνω Ασαφούς Ελεγκτή θα μπορούσε να είναι ένας άνθρωπος ο οποίος να ρυθμίζει τις στρόφιγγες ζεστού/κρύου νερού ώστε να επιτυγχάνει επιθυμητή θερμοκρασία και ροή. Η χρήση Ασαφούς Ελεγκτή προτιμάται στο παραπάνω πρόβλημα ελέγχου για διάφορους λόγους όπως:

1. Οι ρυθμίσεις του παραπάνω συστήματος γίνονται σύμφωνα με τις προσωπικές προτιμήσεις κάποιου, συνεπώς η χρήση διαφορικών εξισώσεων ή εξισώσεων διαφορών δεν είναι ο πλέον ενδεδειγμένος τρόπος σχεδίασης ενός ελεγκτή,
2. Η λύση στο παραπάνω πρόβλημα (αναμένεται να) είναι μη-γραμμική, και
3. Η λύση στο παραπάνω πρόβλημα μπορεί να περιγραφεί λεκτικά.

Καλούνται οι φοιτητές να σχεδιάσουν δύο ασαφείς ελεγκτές (1 ελεγκτή τύπου Mamdani, και 1 ελεγκτή τύπου Sugeno) για το παραπάνω σύστημα σύμφωνα με τις παρακάτω προδιαγραφές:

- Θεωρείστε ότι το διάστημα τιμών θερμοκρασίας νερού του υδραυλικού συστήματος κυκλοφορίας είναι 10 – 60 °C, ενώ το αντίστοιχο διάστημα τιμών για τη ροή είναι 0.1 – 2.1 lt/sec. Επίσης η θέση μιας στρόφιγγας κυμαίνεται μεταξύ -100% (πλήρως κλειστή) και 100% (πλήρως ανοικτή) από το σημείο 0 (σημείο λειτουργίας).
- Χρησιμοποιώντας τον Rule Viewer υπολογίστε τη θέση στρόφιγγας (ζεστού, κρύου) νερού για τα ακόλουθα ζεύγη τιμών (σφάλμα θερμοκρασίας, σφάλμα ροής) του νερού: (-5, 0.1), (20, -0.5), (-10, 1), (40,0), (-30,-0.5).

2.7 Σχεδίαση που Βασίζεται σε Αριθμητικά Δεδομένα Εισόδου-Εξόδου

Σχεδιάστε έναν ασαφή ελεγκτή που να ελέγχει την ταχύτητα περιστροφής ενός ανεμιστήρα στο θάλαμο εντατικής παρακολούθησης ενός νοσοκομείου ώστε η θερμοκρασία και η σχετική υγρασία του θαλάμου να διατηρούνται σε επίπεδα σύμφωνα με τις υποδείξεις των γιατρών. Ο ελεγκτής διαθέτει αισθητήρα θερμοκρασίας που μετράει τη θερμοκρασία από 0 °C έως 50 °C, και αισθητήρα που μετράει τη σχετική υγρασία από 20 % έως 100 %. Η ταχύτητα περιστροφής του ανεμιστήρα κυμαίνεται από 120 έως 1200 RPM (Rotations Per Minute).

Γιατροί έχουν ορίσει τις παρακάτω 5 προδιαγραφές:

	Θερμοκρασία	Σχετική Υγρασία	Επιθυμητή ταχύτητα περιστροφής ανεμιστήρα.
1	8	20	300
2	15	40	400
3	21	50	600
4	32	70	1000
5	41	90	900

Για τεχνικούς λόγους θα πρέπει, επιπλέον, να ικανοποιούνται οι παρακάτω δύο προδιαγραφές:

1. Η μέση ανοχή σφάλματος στην έξοδο (η τελευταία είναι η επιθυμητή ταχύτητα περιστροφής του ανεμιστήρα) να είναι το πολύ 5%, και
2. Θα πρέπει να χρησιμοποιηθούν το πολύ 4 κανόνες οι οποίοι πρέπει να καλύπτουν όλο το εύρος των τιμών.

ΚΕΦΑΛΑΙΟ 3: ΤΕΧΝΗΤΑ ΝΕΥΡΩΝΙΚΑ ΔΙΚΤΥΑ

Στο κεφάλαιο αυτό θα περιγραφεί η βασική λειτουργία ενός απλού Τεχνητού Νευρωνικού Δικτύου (ΤΝΔ) και οι τύποι των υπολογισμών που μπορεί να εκτελέσει. Επίσης, θα σχολιαστούν τα χαρακτηριστικά των βιολογικών νευρωνικών δικτύων και πώς αυτά έχουν χρησιμεύσει ως έμπνευση στους ερευνητές για την κατασκευή μοντέλων νευρωνικών δικτύων, τα οποία ονομάζονται τεχνητά νευρωνικά δίκτυα. Ακολουθεί μία ιστορική αναδρομή μέσα από την οποία φαίνεται η πρόοδος στον τομέα αυτόν τα τελευταία 50 χρόνια. Το εισαγωγικό τμήμα κλείνει με περιγραφή μερικών γενικών εφαρμογών τους.

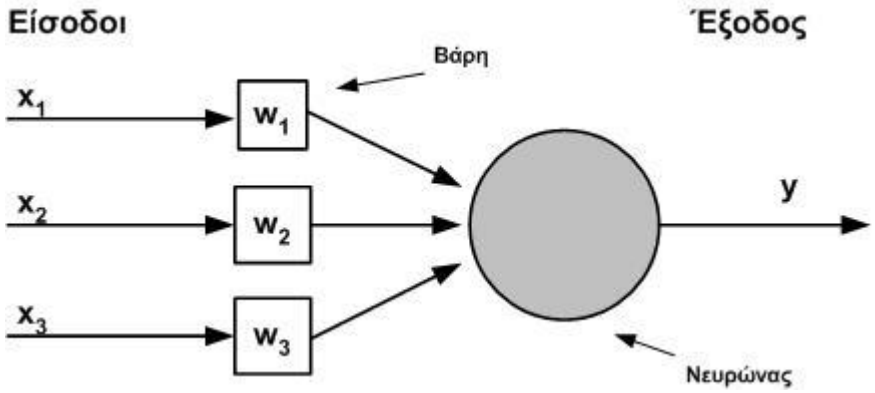
3.1 Εισαγωγή

Τα Τεχνητά Νευρωνικά Δίκτυα (ΤΝΔ) είναι απλοποιημένα μοντέλα του κεντρικού νευρικού συστήματος. Είναι δίκτυα με υψηλής αλληλοσύνδεσης νευρωνικά υπολογιστικά στοιχεία, τα οποία έχουν την ικανότητα να αποκρίνονται σε ερεθίσματα εισόδου και τα οποία μαθαίνουν να προσαρμόζονται στο περιβάλλον. Υποστηρίζεται από πολλούς ερευνητές στον χώρο αυτόν ότι τα μοντέλα νευρωνικού δικτύου προσφέρουν την πιο ελπιδοφόρα ενοποιημένη προσέγγιση για την κατασκευή πραγματικά νοήμονων συστημάτων υπολογιστών. Επίσης πιστεύεται ότι η χρήση κατανεμημένων, παράλληλων υπολογισμών που εκτελούνται στα ΤΝΔ είναι ο καλύτερος τρόπος για την υπερνίκηση της συνδυαστικής έκρηξης, που σχετίζεται με τους συμβολικούς σειριακούς υπολογισμούς, όταν χρησιμοποιούνται οι αρχιτεκτονικές υπολογιστών von Neumann. Έχει αποδειχθεί ότι τα ΤΝΔ μπορούν να χρησιμοποιηθούν αποτελεσματικά ως υπολογιστικοί επεξεργαστές σε ποικίλες εργασίες όπως η αναγνώριση προτύπου (speech and visual image recognition), η ταξινόμηση (classification), η συμπίεση δεδομένων (data compression), η πρόβλεψη (forecasting), η προσομοίωση (modeling) και ο προσαρμοστικός έλεγχος (adaptive control). Επιδεικνύουν έναν αριθμό επιθυμητών ιδιοτήτων που δεν συναντώνται στα συμβατικά, συμβολικά υπολογιστικά συστήματα, όπως η υψηλή απόδοση, όταν σχετίζονται με θόρυβο ή ατελή πρότυπα εισόδου, ο υψηλός βαθμός ανοχής σφάλματος, οι υψηλοί ρυθμοί παράλληλου υπολογισμού, η ικανότητα να γενικεύουν και η προσαρμοστική μάθηση.

3.2 Ένα απλό Τεχνητό Νευρωνικό Δίκτυο

Ένα νευρωνικό δίκτυο χαρακτηρίζεται από (1) το πρότυπο των συνδέσεων μεταξύ των νευρώνων (καλείται αρχιτεκτονική), (2) την μέθοδο καθορισμού των βαρών του στις συνδέσεις (καλείται αλγόριθμος εκπαίδευσης ή μάθησης) και (3) την συνάρτηση ενεργοποίησης.
Ένα ΤΝΔ αποτελείται από έναν μεγάλο αριθμό υπολογιστικών στοιχείων που ονομάζονται νευρώνες, μονάδες, κύτταρα ή κόμβοι. Κάθε νευρώνας είναι συνδεδεμένος με άλλους νευρώνες με ευθύγραμμους συνδέσμους επικοινωνίας. Κάθε σύνδεσμος εισόδου έχει ένα συσχετισμένο εξωτερικό σήμα εισόδου ή μία διέγερση, ένα αντίστοιχο βάρος και ένα μικρό φίλτρο που είναι τμήμα του σημείου σύνδεσης μεταξύ της εισόδου και του νευρώνα.

Παρόλο που οι αρχιτεκτονικές των ΤΝΔ διαφέρουν σε μερικά χαρακτηριστικά σημεία, μπορεί να ειπωθεί ότι ένας τυπικός νευρώνας ενός ΤΝΔ ή ένα υπολογιστικό στοιχείο είναι βασικά ένας συγκριτής ο οποίος παράγει μία έξοδο όταν η συνολική επίδραση της διέγερσης εισόδου υπερβαίνει ένα κατώφλι. Στο Σχήμα 3.1 παρουσιάζεται ένας απλός νευρώνας ΤΝΔ με τρεις εισόδους και μία έξοδο.



Σχήμα 3.1: Ένας απλός νευρώνας.

3.3 Βασικές έννοιες των νευρωνικών υπολογισμών

Στο Σχ. 3.1 υπάρχουν i (i =1,2,3) σύνδεσμοι, ο καθένας από τους οποίους έχει ένα εξωτερικό σήμα εισόδου x_i και ένα αντίστοιχο βάρος w_i. Οι τιμές εισόδου x_i μπορούν να είναι πραγματικές (+ ή -), δυαδικές (0,1) ή διπολικές (-1,+1). Τα βάρη τα οποία προσομοιώνουν τις συναπτικές νευρωνικές συνδέσεις των βιολογικών δικτύων, ενεργούν έτσι ώστε, είτε να αυξάνουν (διεγερτική είσοδος) είτε να μειώνουν (ανασταλτική είσοδος) τα σήματα εισόδου του νευρώνα. Τα βάρη μπορούν να έχουν επίσης δυαδικές ή πραγματικές τιμές. Συνήθως όμως θεωρούνται ότι είναι πραγματικές: θετικές για διεγερτικούς και αρνητικές για ανασταλτικούς συνδέσμους. Η έξοδος του ΤΝΔ μπορεί επίσης να πάρει τιμή πραγματική, δυαδική ή διπολική.
Ο νευρώνας συμπεριφέρεται σαν μία συνάρτηση ενεργοποίησης ή απεικόνισης $f(\cdot)$ και παράγει μία έξοδο $y=f(net)$ όπου net είναι η συνολική διέγερση εισόδου στον νευρώνα και f είναι μία μη γραμμική συνάρτηση του net. Για παράδειγμα, το net λαμβάνεται συνήθως ως το άθροισμα των εισόδων, φορτισμένων με τα αντίστοιχα βάρη:

$$net = x_1w_1+x_2w_2+x_3w_3 = \sum_{i=1}^{3} x_i w_i$$

και f είναι μία μονότονη αύξουσα συνάρτηση του net. Φυσικά, οι i είσοδοι σε ένα δίκτυο είναι συνήθως πολύ περισσότερες από τρεις. Σε αυτήν την περίπτωση χρησιμοποιούμε έναν δείκτη n για να δηλώσουμε ότι έχουμε έναν αυθαίρετο αριθμό εισόδων και η έξοδος του δικτύου υπολογίζεται μέσω της εξίσωσης

Έξοδος Τεχνητού Δικτύου *Νευρωνικού* $$y = f\left(\sum_{i=1}^{n} x_i w_i\right)$$

3.4 Νευρωνικά Δίκτυα Perceptrons Πολλαπλών Στρωμάτων

Σε αυτήν την παράγραφο θα μελετηθεί μία σημαντική κατηγορία νευρωνικών δικτύων, που ονομάζονται *πρόσω τροφοδοτούμενα πολλών στρωμάτων δίκτυα (multilayer feedforward networks)*. Ένα τυπικό δίκτυο αποτελείται από ένα σύνολο μονάδων αισθητηρίων (πηγαίοι κόμβοι) που αποτελούν το *στρώμα εισόδου (input layer)*, ένα ή περισσότερα *κρυφά στρώματα (hidden layers)* με κόμβους υπολογισμού και ένα *στρώμα εξόδου (output layer)* με κόμβους υπολογισμού. Το σήμα εισόδου διαδίδεται μέσω του δικτύου με προς τα εμπρός κατεύθυνση από στρώμα σε στρώμα. Αυτά τα νευρωνικά δίκτυα συνήθως αναφέρονται ως *perceptrons πολλαπλών στρωμάτων (MLPs),* τα οποία αποτελούν γενίκευση του *perceptron* ενός στρώματος.
Τα *MLPs* έχουν εφαρμοστεί επιτυχώς για την επίλυση δύσκολων και ποικίλων προβλημάτων, μέσω της εκπαίδευσής τους με εποπτευόμενο τρόπο, με τη βοήθεια ενός πολύ διαδεδομένου αλγόριθμου γνωστού ως *error back-propagation algorithm.* Αυτός ο αλγόριθμος βασίζεται *στον* κανόνα *error-correction learning rule* και μπορεί να θεωρηθεί ως γενίκευση ενός εξίσου διαδεδομένου προσαρμοστικού αλγόριθμου φιλτραρίσματος, που ονομάζεται *least-mean-square (LMS) algorithm.*
Η διαδικασία του *error back-propagation* συνίσταται από δύο διαβάσεις μέσω των διαφορετικών στρωμάτων του δικτύου: μίας διάβασης προς τα εμπρός και μίας προς τα πίσω. Στην προς τα εμπρός διάβαση, ένα πρότυπο δραστηριότητας (διάνυσμα εισόδου) εφαρμόζεται στους *αισθητήριους* κόμβους του δικτύου και η δράση του διαδίδεται μέσω του δικτύου από στρώμα σε στρώμα. Τελικά ένα σύνολο εξόδων παράγεται ως πραγματική απόκριση του δικτύου. Κατά την διάρκεια της προς τα εμπρός διάβασης τα συναπτικά βάρη *(synaptic weights)* είναι όλα σταθερά. Από την άλλη μεριά, κατά την διάρκεια της προς τα πίσω διάβασης τα συναπτικά βάρη είναι όλα προσαρμοσμένα σύμφωνα με τον κανόνα *error-correction.* Συγκεκριμένα, η πραγματική απόκριση του δικτύου αφαιρείται από την επιθυμητή απόκριση –που καλείται στόχος– για την παραγωγή του σήματος σφάλματος. Τότε αυτό το σήμα σφάλματος, διαδίδεται προς τα πίσω μέσω του δικτύου αντίθετα προς την κατεύθυνση τον συναπτικών συνδέσεων· γι' αυτό προέκυψε η ονομασία *"error back-propagation"*. Τα συναπτικά βάρη προσαρμόζονται κατά τέτοιον τρόπο ώστε να φέρουν την πραγματική απόκριση του δικτύου όλο και πιο κοντά στην επιθυμητή απόκριση. Ο αλγόριθμος *error back-propagation* αναφέρεται επίσης στην βιβλιογραφία ως αλγόριθμος *back-propagation* ή απλά *back-prop.*

Ένα *perceptron* πολλαπλών στρωμάτων έχει τρία ιδιαίτερα χαρακτηριστικά:

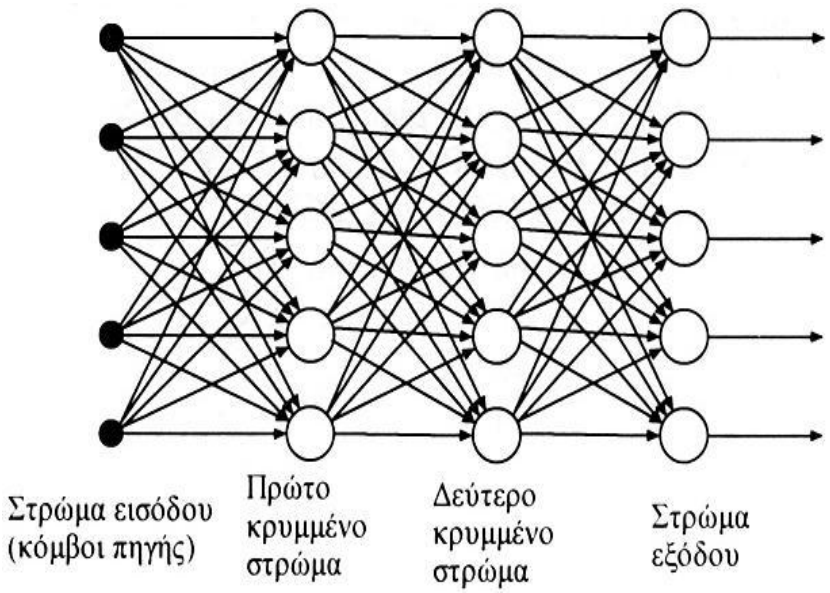
1. Το πρότυπο του κάθε νευρώνα στο δίκτυο περιλαμβάνει μία *μη γραμμικότητα* στο τέλος *της* εξόδου. Το σημαντικό σημείο που πρέπει να τονιστεί εδώ, είναι ότι η μη γραμμικότητα είναι ομαλή (παντού διαφορίσιμη), σε αντίθεση με τον ισχυρό περιορισμό που χρησιμοποιείται στο perceptron του Rosenblatt. Μία συνηθισμένη μορφή της μη γραμμικότητας που χρησιμοποιείται και η οποία ικανοποιεί αυτές τις απαιτήσεις, είναι η σιγμοειδής μη γραμμικότητα (*sigmoidal nonlinearity*) που ορίζεται από την λογιστική συνάρτηση:

$$y_j = \frac{1}{1+e^{-u_j}}$$

όπου u_j είναι το δικτυακό εσωτερικό επίπεδο ενεργοποίησης του νευρώνα j και y_j είναι η έξοδος του νευρώνα. Η παρουσία των μη γραμμικοτήτων είναι σημαντική,

επειδή σε διαφορετική περίπτωση η σχέση μεταξύ εισόδου και εξόδου του δικτύου μπορεί να μειωθεί σε αυτή ενός perceptron ενός στρώματος. Επιπλέον η χρήση της λογιστικής συνάρτησης είναι βιολογικά παρακινούμενη, αφού προσπαθεί να δικαιολογήσει την ανυπότακτη φάση των πραγματικών νευρώνων.
2. Το δίκτυο περιέχει ένα ή περισσότερα στρώματα κρυμμένων νευρώνων, τα οποία δεν είναι μέρος της εισόδου ή της εξόδου του δικτύου. Αυτοί οι κρυμμένοι νευρώνες επιτρέπουν το δίκτυο να μάθει πολύπλοκες απεικονίσεις με την προοδευτική εξαγωγή των σημαντικότερων χαρακτηριστικών από τα πρότυπα εισόδου (διανύσματα).
3. Το δίκτυο επιδεικνύει υψηλό βαθμό *συνεκτικότητας*, που καθορίζεται από τις συνάψεις του δικτύου. Μία αλλαγή στην συνεκτικότητα του δικτύου απαιτεί αλλαγή στον αριθμό των συναπτικών συνδέσεων ή στα βάρη τους.

Είναι αλήθεια ότι τα perceptrons πολλαπλών στρωμάτων αντλούν την υπολογιστική τους ισχύ από τον συνδυασμό αυτών των χαρακτηριστικών μαζί με την ικανότητα της μάθησης που προκύπτει από την εμπειρία που προέρχεται από την εκπαίδευση. Αυτά τα ίδια χαρακτηριστικά είναι ωστόσο υπεύθυνα για τις ατέλειες της παρούσας κατάστασης της γνώσης μας, σε ό,τι αφορά την συμπεριφορά των δικτύων. Πρώτον, το παρουσιαστικό μίας διανεμημένης μορφής μη γραμμικότητας και η υψηλή συνεκτικότητα των δικτύων, κάνουν δύσκολη την θεωρητική ανάλυση ενός perceptron πολλαπλών στρωμάτων. Δεύτερον, η χρήση των κρυφών νευρώνων κάνει την διαδικασία μάθησης πιο δύσκολη. Προκύπτει η αίσθηση ότι η διαδικασία μάθησης πρέπει να αποφασίσει ποια χαρακτηριστικά του προτύπου εισόδου πρέπει να αναπαρασταθούν από τους κρυμμένους νευρώνες. Η διαδικασία μάθησης γίνεται συνεπώς πολύ πιο δύσκολη, επειδή η έρευνα πρέπει να διεξαχθεί σε πολύ μεγαλύτερο διάστημα πιθανών συναρτήσεων και η επιλογή θα πρέπει να γίνει μεταξύ εναλλακτικών αναπαραστάσεων του προτύπου εισόδου.
Η δομή ενός MLP τριών στρωμάτων φαίνεται στο παρακάτω Σχήμα 3.2.



Σχήμα 3.2: Multiperceptron feed-forward δίκτυο με δυο κρυμμένα στρώματα

Έχει αποδειχθεί θεωρητικά ότι το *MLP* έχει αυξημένες δυνατότητες απεικόνισης και συγκεκριμένα χαρακτηρίζεται από την ιδιότητα της *παγκόσμιας προσέγγισης* (universal approximation). Παραλείποντας τον αυστηρό μαθηματικό ορισμό η ιδιότητα αυτή μας λέει το εξής: ένα *MLP* με τουλάχιστον ένα κρυμμένο επίπεδο με μη γραμμικές κρυμμένες μονάδες μπορεί να προσεγγίσει οποιαδήποτε συνάρτηση με οποιαδήποτε ακρίβεια, αυξάνοντας κατάλληλα τον αριθμό των κρυμμένων μονάδων. Η ιδιότητα αυτή είναι θεωρητικά μόνο σημαντική, διότι μας εξασφαλίζει ότι το *MLP* μπορεί να υλοποιήσει οποιαδήποτε απεικόνιση, αλλά δεν είναι πρακτική διότι δεν μας λέει τίποτα για το πώς θα υλοποιήσουμε την απεικόνιση (πόσες κρυμμένες μονάδες να βάλουμε, τι αλγόριθμο εκπαίδευσης να χρησιμοποιήσουμε κτλ.). Το πρόβλημα του καθορισμού του αριθμού των κρυμμένων μονάδων που απαιτούνται για ένα δεδομένο σύνολο εκπαίδευσης αποτελεί σήμερα βασικό ζήτημα σχετικά με τα *MLP*.
Εξαιτίας του γεγονότος ότι, ένα *MLP* έχει την ικανότητα να απεικονίζει ένα διάνυσμα πραγματικών εισόδων σε ένα διάνυσμα πρόβλεψης (prediction), μπορεί να χρησιμοποιηθεί για την κατασκευή μοντέλων από δεδομένα(data fitting), για τον έλεγχο συστημάτων, ακόμη και για την επίλυση διαφορικών εξισώσεων.
Η βασικότερη όμως εφαρμογή του *MLP* είναι τα προβλήματα ταξινόμησης (classification). Στην περίπτωση αυτή τα δεδομένα είναι της μορφής (πρότυπο, κατηγορία) και προκειμένου να εκπαιδευτεί το *MLP* απαιτείται μία διαδικασία που ονομάζεται κωδικοποίηση των κατηγοριών.

3.5 Νευρωνικά Δίκτυα και MATLAB

Το MATLAB διαθέτει ένα πλήρες εργαλείο ανάπτυξης εφαρμογών για Νευρωνικά Δίκτυα, το οποίο ονομάζεται Neural Network Toolbox. Το εργαλείο αυτό περιλαμβάνει ένα σύνολο συναρτήσεων με τις οποίες μπορούμε να κατασκευάσουμε, εκπαιδεύσουμε και να χρησιμοποιήσουμε ένα μεγάλο αριθμό διαφορετικών κατηγοριών νευρωνικών δικτύων. Επειδή στα πλαίσια των εργαστηριακών μαθημάτων θα περιοριστούμε στην κατηγορία των *MLPs*, είναι χρήσιμο να μελετήσουμε τις αντίστοιχες συναρτήσεις που μας παρέχει το σχετικό toolbox του MATLAB.

Συνάρτηση κατασκευής δικτύου

```
net = newff(PR,[S1 S2...SNl],{TF1 TF2...TFNl},BTF)
```

`PR` : Είναι ένα διάνυσμα `Rx2` που περιέχει τις ελάχιστες και μέγιστες τιμές των `R` εισόδων του δικτύου.
`Si` : Είναι το μέγεθος του i επιπέδου
`TFi`: Είναι η συνάρτηση μεταφοράς του i επιπέδου
`BTF`: Είναι η μέθοδος εκπαίδευσης του δικτύου

Συνάρτηση εκπαίδευσης του δικτύου

```
net = train(net,P,T)
```

`net` : Προς εκπαίδευση δίκτυο
`P` : Είσοδοι εκπαίδευσης (Patterns)
`T` : Έξοδοι εκπαίδευσης (Targets)

Συνάρτηση λειτουργίας του δικτύου

```
Y = sim(net,X)
```

`net` : Εκπαιδευμένο δίκτυο
`X` : Είσοδος δικτύου
`Y` : Έξοδος δικτύου

Ο συνδυασμός των τριών παραπάνω συναρτήσεων μας δίνει τη δυνατότητα να κατασκευάσουμε, εκπαιδεύσουμε και να χρησιμοποιήσουμε ένα νευρωνικό δίκτυο ανάλογα με τις ανάγκες του κάθε προβλήματός μας.

3.6 Ασκήσεις

Στη συνέχεια παρατίθενται δύο παραδείγματα εφαρμογής νευρωνικών δικτύων στην προσέγγιση μίας συνάρτησης μίας μεταβλητής και στην ταξινόμηση δεδομένων που ανήκουν σε δύο κλάσεις.

Άσκηση 1η: (Προσέγγιση Συνάρτησης Ημιτόνου)

```
%%%%%%% Training Data
x_train  = 0:pi/10:2*pi;
y_train  = sin(x_train);

%%%%%%% Test Data
x_test   = 0:pi/7:2*pi;
y_test   = sin(x_test);

%%%%%%% Network Construction
net = newff(x_train,y_train,3,{'tansig'});

%%%%%%% Network Training
net.trainParam.epochs = 200;
net = train(net,x_train,y_train);

%%%%%%% Network Operation
Y   = sim(net,x_test);

%%%%%%% Plot results
plot(x_test,y_test,x_test,Y,'r');
```

Άσκηση 2η: (Ταξινόμηση Προτύπων 2 Κλάσεων)

```
%%%%%%% Training Data
P_train(1,1:20) = rand(1,20)*5;
P_train(2,1:20) = rand(1,20)+5;
P_train(1,21:40)= rand(1,20)+5;
P_train(2,21:40)= rand(1,20)*5;

T_train(1:20) = zeros(1,20);
T_train(21:40)= ones(1,20);

plot(P_train(1,1:20),P_train(2,1:20),'*b',P_train(1,21:40),P_train(2,
21:40),'*r');

%%%%%%% Test Data
P_test = [1.5 2.3 6.7 9.1;6.3 3 5.7 2.1];
T_test = [0 1 0 1];

%%%%%%% Network Construction
net = newff(P_train,T_train,10,{'tansig','logsig'});

%%%%%%% Network Training
net.trainParam.epochs = 200;
net = train(net,P_train,T_train);

%%%%%%% Network Operation
Y   = sim(net,P_test);

%%%%%%% Plot results
figure;
subplot(1,2,1);
for i=1:length(T_test)
    if T_test(i)==0
     plot(P_test(1,i),P_test(2,i),'*b');
    else
     plot(P_test(1,i),P_test(2,i),'*r');
    end
    hold on;
end
title('Desired Points');


subplot(1,2,2);
for i=1:length(T_test)
    if Y(i)<=0.5
     plot(P_test(1,i),P_test(2,i),'*b');
    else
     plot(P_test(1,i),P_test(2,i),'*r');
    end
    hold on;
End
title('Recognized Points');
```