Τ.Ε.Ι. ΘΕΣΣΑΛΟΝΙΚΗΣ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Ασαφή Συστήματα

Θεωρία και Εργαστηριακές Ασκήσεις

Παπαδάκης Στέλιος

Αδαμίδης Παναγιώτης

Θεσσαλονίκη, Μάιος 2004

Περιεχόμενα

1 Εισαγωγή στην Ασαφή Λογική

Ο στόχος των σημειώσεων είναι η παροχή των απαραίτητων θεωρητικών στοιχείων για την κατανόηση των βασικών εννοιών της ασαφούς λογικής. Μετά την πρωτοποριακή εργασία του *Lofti Zadeh* [3] στα μέσα της δεκαετίας του 1960, η ασαφής λογική γνώρισε αλματώδη ανάπτυξη και στις μέρες μας εφαρμογές της εμφανίζονται σχεδόν σε κάθε τομέα της ανθρώπινης δραστηριότητας.

Ο συνδυασμός της ασαφούς λογικής με τα νευρωνικά δίκτυα και τους εξελικτικούς αλγορίθμους οδήγησε στη δημιουργία συστημάτων με ικανότητες μάθησης, γενίκευσης και συμπερασμού (*cognitive systems*). Η μοντελοποίηση και ο έλεγχος σύνθετων φυσικών διεργασιών, των οποίων η σχέση διέγερσης-απόκρισης χαρακτηρίζεται από ισχυρές μη γραμμικότητες από ανακρίβειες, ακόμα και από αντιφάσεις, είναι εφικτή στο πλαίσιο της ασαφούς λογικής. Οι δυνατότητες αυτές που δεν παρέχονται επαρκώς από τα κλασικά αυστηρά μαθηματικά εργαλεία μοντελοποίησης και ελέγχου καθιέρωσαν την ασαφή λογική ως ένα από τα ποιο αποτελεσματικά και καθιερωμένα πλέον εργαλεία για το σύγχρονο μηχανικό και επιστήμονα. Η επιστημονική έρευνα στον τομέα είναι αχανής. Εφαρμογές εμφανίζονται σε πεδία μοντελοποίησης (*Modeling*), πρόβλεψης (*Prediction*), ελέγχου (*control*), κατάταξη δειγμάτων σε κατηγορίες (*Classification*), αναγνώριση προτύπων (*Pattern Regognition*), συστήματα υποστήριξης αποφάσεων (*Decision Support Systems*), Θεωρία δυνατοτήτων (*Possibility Theory*), βελτιστοποίηση (*Optimization*), ….

Αν και η ασαφής λογική είναι γέννημα του δυτικού πολιτισμού και οι θεωρητικές της βάσεις τέθηκαν και αναπτύχθηκαν επίσης στη δύση, γνώρισε μεγάλη άνθηση και εφαρμογή κυρίως στη Ιαπωνία. Η εξήγηση που έχει δοθεί έχει δύο άξονες. Ο ένας άξονας έχει τις ρίζες του στην εκπαίδευση που έχει ο Ιάπωνας και που του επιτρέπει να εφαρμόζει στην παραγωγή καινοτόμες ιδέες με ταχύτητα και αποτελεσματικότητα. Η άλλη είναι περισσότερο θεωρητική και ενδιαφέρουσα και σχετίζεται με την ιαπωνική γλώσσα και κουλτούρα. Η Δυτική κουλτούρα είναι στενά συνυφασμένη έως εγκλωβισμένη στην Αριστοτέλεια λογική Άσπρο-Μαύρο, Δεκτό-Μη αποδεκτό, Κατάλληλο – Ακατάλληλο κ.τ.λ. σε αντιδιαστολή με την Ανατολική Κουλτούρα που μπορεί εύκολα να δεχτεί και ‘*αποχρώσεις του γκρίζου*’. Επίσης στις Ασιατικές γλώσσες ο όρος *Fuzzy* που στη Δύση μεταφράζεται ασαφής και έχει μάλλον αρνητική απόχρωση στον έλεγχο συστημάτων (Φαντασθείτε ένα σύστημα πέδησης που λειτουργεί ασαφώς σε μια επικίνδυνη στροφή.) στις Ασιατικές Γλώσσες το *fuzzy* μεταφράζεται ως *kanji* και έχει την έννοια της σταδιακής κατάταξης αντικειμένων σε ολοένα και ποιο αυστηρά καθορισμένες κατηγορίες. Γιατί λοιπόν ένας ευρωπαίος να αγοράσει ένα προϊόν π.χ. ένα αυτοκίνητο που φρενάρει ‘*ασαφώς*’ . Και γιατί αυτό να είναι εμπόδιο για ένα Ασιάτη που ακούγοντας *kanji* καταλαβαίνει ότι το αυτοκίνητο φρενάρει με ένα τρόπο που γίνεται προοδευτικά αυστηρότερος και ακριβέστερος [1,2].

Ανάμεσα στις περίπου 2000 πατέντες που κατοχυρώθηκαν μέχρι το 1990 ξεχωρίζουν εφαρμογές οικείες στο ευρύ κοινό, όπως πλυντήρια ρούχων με *fuzzy control* όπου το πλυντήριο ρυθμίζει τις στροφές τον τύπο και την ποσότητα του απορρυπαντικού καθώς και τη θερμοκρασία του νερού, ανάλογα με την αντοχή τον τύπο και την ποιότητα των ρούχων, που μετράει με κατάλληλα αισθητήρια. Κλιματιστικά μηχανήματα που ρυθμίζουν την ταχύτητα την υγρασία και τη θερμοκρασία του αέρα ανάλογα με τις διαθέσεις του κλιματιζόμενου.

Στο εργαστήριο θα ασχοληθούμε κυρίως με την μοντελοποίηση και τον έλεγχο πραγματικών συστημάτων, που είναι και από τις περισσότερο πρακτικές και εφαρμόσιμες πλευρές της ασαφούς λογικής.

2 Βασικές Αρχές Της Ασαφούς Λογικής

Όπως κατά την εκμάθηση μιας γλώσσας προγραμματισμού τείνει να καθιερωθεί πλέον η πρώτη εφαρμογή να είναι η εκτύπωση στην οθόνη της φράσης *Hello World* , για τους ίδιους ακριβώς λόγους έχει σχεδόν καθιερωθεί η πρώτη προσέγγιση στις έννοιες της ασαφούς λογικής να πραγματοποιείται με το παράδειγμα του φρεναρίσματος ενός αυτοκινήτου.

Ας υποθέσουμε λοιπόν ότι οδηγούμε ένα όχημα με κάποια ταχύτητα και σε κάποια απόσταση αντιλαμβανόμαστε ένα εμπόδιο. Για απλότητα ας θεωρήσουμε ότι το οδόστρωμα είναι επίπεδο και ομαλό. Το ερώτημα είναι ο τρόπος με τον οποίο θα χειριστούμε την κατάσταση. Ένα πολύ μαθηματικό μυαλό θα μπορούσε να αναγνωρίσει αμέσως την αναλυτική μαθηματική φόρμουλα $F = f(d,u)$, η οποία θα του έδινε την δύναμη F με ακρίβεια κάποιων δεκαδικών ψηφίων που θα έπρεπε να ασκήσει στο φρένο του αυτοκινήτου, προκειμένου να σταματήσει ομαλά πριν από το εμπόδιο. Η σχέση βέβαια αυτή εξαρτάται από πολλούς παράγοντες, όπως ο τύπος του αυτοκινήτου, ο τύπος του φρένου, το βάρος του αυτοκινήτου η κατεύθυνση του ανέμου, κ.τ.λ. Μια τέτοια μαθηματική προσέγγιση επομένως είναι αδύνατη για τον κοινό νου. Ακόμα και αν χρησιμοποιούσαμε ένα σύστημα κλασσικού αυτόματου ελέγχου (*pid controler*), τα πράγματα θα ήταν εξαιρετικά δύσκολα διότι ο κλασσικός εκλεκτής θα έπρεπε να ρυθμίζει τα κέρδη του ανάλογα με τις συνθήκες, οι οποίες μπορεί να μεταβληθούν πολλές φορές κατά τη διάρκεια της πέδησης (π.χ. ένας ξαφνικός άνεμος κατά τη φορά του αυτοκινήτου). Στο παράδειγμα του φρεναρίσματος, δεν έχουμε ένα σταθερά διαμορφωμένο περιβάλλον η έστω ένα περιβάλλον που μεταβάλλεται με προβλέψιμο τρόπο. Οι συνθήκες μεταβάλλονται απρόβλεπτα η και αντιφατικά ορισμένες φορές (π.χ. αν ο δρόμος σε ένα σημείο έχει λάδια ο τρόπος του φρεναρίσματος θα πρέπει να ακολουθήσει εντελώς διαφορετική στρατηγική). Γεννιέται λοιπόν το ερώτημα: Πως καταφέρνει ο οδηγός να φρενάρει το όχημα ακόμα και σε δύσκολες γρήγορα μεταβαλλόμενες συνθήκες και μάλιστα κατά τρόπο ομαλό έτσι ώστε να μην ταλαιπωρηθούν επιβάτες και εμπορεύματα;

Ο οδηγός του αυτοκινήτου έχει συσσωρευμένη εμπειρία τόσο από την εκπαίδευση του όσο και από την εμπειρία του ως οδηγός. Η εμπειρία αυτή είναι οργανωμένη υπό μορφή κανόνων. Για παράδειγμα ο οδηγός γνωρίζει ότι αν το εμπόδιο είναι *μακριά* και η ταχύτητα του είναι *μικρή* δεν χρειάζεται να φρενάρει. Αν όμως το εμπόδιο είναι *κοντά* και η ταχύτητα του είναι *μεγάλη* τότε πρέπει να πατήσει *πολύ* φρένο. Οι όροι πολύ, λίγο, μικρή ταχύτητα μεγάλη ταχύτητα, μικρή απόσταση, μεγάλη απόσταση, κ.τ.λ. είναι *λεκτικοί (linguistic)* όροι. Ο οδηγός είναι σε θέση να εκτιμήσει πότε μια ταχύτητα είναι μικρή η μεγάλη. Π.χ. ταχύτητα 50 km/h σε στεγνό οδόστρωμα μπορεί να θεωρηθεί μικρή. Η ίδια ταχύτητα όμως υπό διαφορετικές συνθήκες π.χ. σε κατοικημένη περιοχή και βρεγμένο οδόστρωμα είναι μάλλον μέτρια και υπό άλλες συνθήκες π.χ. το οδόστρωμα έχει λάδια είναι μάλλον μεγάλη. Είναι δυνατόν να μεταφέρουμε την εμπειρία του οδηγού σε ένα μηχανικό ελεγκτή χρησιμοποιώντας τα κλασσικά μαθηματικά εργαλεία; Μια τέτοια μεταφορά θα δημιουργούσε κανόνες της μορφής:

```
Αν η ταχύτητα είναι από 10km/h έως 20km/h και η απόσταση από 80 έως 100μ τότε
άσκησε στο φρένο πίεση  0.1 atm
```

Αν η ταχύτητα είναι από 21km/h έως 40km/h και η απόσταση από 30 έως 79μ τότε άσκησε στο φρένο πίεση 0.4 atm

Αυτή η αντιμετώπιση έχει σημαντικές αδυναμίες. Τι θα συμβεί αν η ταχύτητα είναι 11 km/h? Ο ελεγκτής θα συμπεριφερθεί με τον ίδιο τρόπο που θα συμπεριφερθεί στην περίπτωση που η ταχύτητα είναι 19km/h? Μια λύση στο πρόβλημα φαίνεται να είναι η μεγαλύτερη διακριτοποίηση κάθε μεταβλητής, π.χ. ανά 1km/h. Τι θα συμβεί όμως αν έχουμε 10 μεταβλητές εισόδου και κάθε μια διακριτοποιείται σε 50 τμήματα? Θα δημιουργηθούν $50^{10} = ...$ κανόνες. Σε κάθε περίπτωση ο στόχος που είναι η μεταφορά της εμπειρίας του οδηγού φαίνεται ότι δεν επιτυγχάνεται τόσο αποτελεσματικά με τον αυστηρά αριθμητικό (*crisp*) τρόπο που επιχειρήσαμε. Το ανθρώπινο μυαλό χειρίζεται λεκτικούς όρους (*linguistic terms*) παρά αριθμούς και η μηχανή που φιλοδοξεί να μιμηθεί τον τρόπο λειτουργίας του θα πρέπει να το λάβει σοβαρά υπόψη. Ας δούμε πως θα αντιμετώπιζε ο άνθρωπος το παράδειγμα της πέδησης. Ένας έμπειρος οδηγός θα λειτουργούσε βάση λεκτικών κανόνων π.χ.

Αν η ταχύτητα είναι μικρή και η απόσταση είναι μέτρια τότε δύναμη πέδησης μικρή

Αν η ταχύτητα είναι μέτρια και η απόσταση είναι μέτρια τότε δύναμη πέδησης μέτρια

Αν η ταχύτητα είναι μεγάλη και η απόσταση είναι μικρή τότε δύναμη πέδησης μεγάλη κ.τ.λ.

Οι όροι μέτρια, μεγάλη, μικρή είναι οι λεκτικοί όροι οι οποίοι συμπλέκονται μεταξύ τους με συνδέσμους (*connectives*) και συνθέτουν κανόνες (*rules*). Πως όμως θα χειριστεί λεκτικούς όρους μια υπολογιστική μηχανή που χειρίζεται μόνο αριθμούς? Εδώ υπάρχει ασυμφωνία χαρακτήρων. Σε τέτοιες περιπτώσεις υποχωρεί ο ποιο λογικός που στην περίπτωση μας είναι ο άνθρωπος και δημιουργεί τη γέφυρα επικοινωνίας. Η γέφυρα αυτή είναι η *ασαφής λογική*. Ο *Zadeh* που θεωρείται ο ‘πατέρας’ της ασαφούς λογικής δήλωνε ότι η προσπάθεια μας να κατασκευάσουμε μηχανές με νοημοσύνη δεν μπορεί να προχωρήσει αν δεν βρεθεί ένας τρόπος ώστε οι μηχανές να σκέφτονται με παρόμοιους μηχανισμούς με τον άνθρωπο. Ας δούμε λοιπόν μερικές θεμελιώδεις έννοιες της ασαφούς λογικής.

2.1 Το ασαφές σύνολο

Το ασαφές σύνολο είναι ίσως η βασικότερη έννοια του οικοδομήματος της ασαφούς λογικής. Ο Zadeh παρατήρησε ότι ο παραδοσιακός τρόπος περιγραφής ενός συστήματος που στηρίζεται στην αυστηρή λογική ότι μια κατάσταση μπορεί να έχει δύο μόνο μορφές ύπαρξη ή απουσία συνεπάγεται απώλεια πληροφορίας καθώς η πολυπλοκότητα του συστήματος αυξάνεται (δες το παράδειγμα με τους $50^{10} = ...$ κανόνες). Αν υιοθετηθεί λοιπόν ο τρόπος περιγραφής ενός συστήματος με τον αυστηρά αριθμητικό τρόπο υπάρχουν δύο επιλογές α) απλό μαθηματικό μοντέλο με απώλεια πληροφορίας ιδιαίτερα στις οριακές καταστάσεις (π.χ. ταχύτητα 21 km/h) β) μη απώλεια πληροφορίας με πολύπλοκο μαθηματικό μοντέλο ($50^{10} = ...$ καταστάσεις). Ο *Zadeh* διατύπωσε αυτό το αδιέξοδο με την περίφημη *Αρχή της Ασυμβατότητας [4]:*

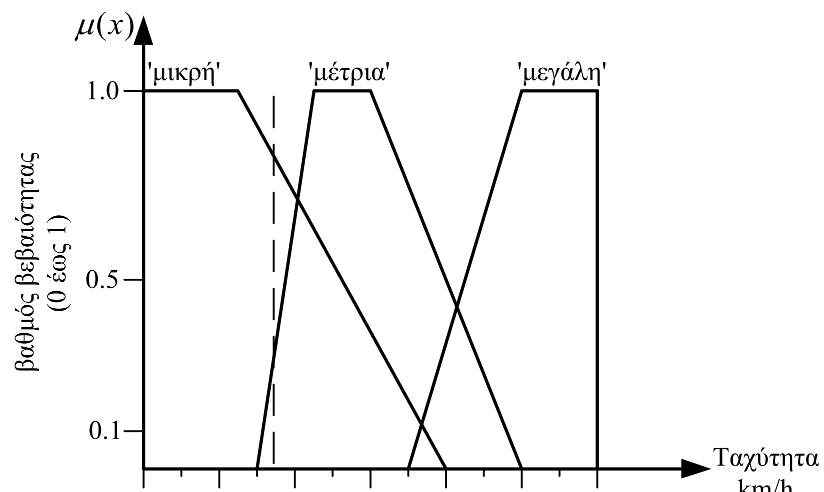
...καθώς η πολυπλοκότητα ενός συστήματος αυξάνεται, η ικανότητα για ακριβείς και ταυτόχρονα σημαντικές δηλώσεις που αφορούν τη συμπεριφορά του μειώνεται, και πέρα από ένα σημείο η ακρίβεια και η σημαντικότητα αποτελούν σχεδόν αμοιβαία αποκλειόμενα χαρακτηριστικά.

Ο Zadeh συνειδητοποίησε λοιπόν ότι ο πυρήνας του αδιεξόδου είναι ο δυαδικός τρόπος αναπαράστασης της πληροφορίας κατά τον οποίο μια τιμή μιας μεταβλητής είτε ανήκει είτε δεν ανήκει σε ένα υποσύνολο του πεδίου ορισμού της. Πρότεινε λοιπόν ένα διευρυμένο τρόπο αναπαράστασης όπου μια τιμή ανήκει ταυτόχρονα σε πολλά υποσύνολα, στο κάθε ένα με ένα βαθμό συμμετοχής. Κάθε τέτοιο υποσύνολο που περιλαμβάνει στοιχεία όπου κάθε ένα έχει ένα βαθμό συμμετοχής είναι το ασαφές σύνολο.

Ας θεωρήσουμε τη μεταβλητή που αντιπροσωπεύει την ταχύτητα ενός αυτοκινήτου σε ένα αυτοκινητόδρομο με πεδίο ορισμού από 0 km/h έως 120 km/h. Κάνουμε τις έξης ερωτήσεις στον οδηγό:

ΕΡ: 'πόσο σίγουρος είσαι ότι η ταχύτητα 15 km/h είναι μικρή?'
ΑΠ: 100%.
ΕΡ: 'πόσο σίγουρος είσαι ότι η ταχύτητα 20 km/h είναι μικρή?'
ΑΠ: 100%.
ΕΡ: 'πόσο σίγουρος είσαι ότι η ταχύτητα 40 km/h είναι μικρή?'
ΑΠ: 90% είναι και λίγο μέτρια.
ΕΡ: 'πόσο σίγουρος είσαι ότι η ταχύτητα 60 km/h είναι μικρή?'
ΑΠ: λίγο ένα 10-20% διότι 60 km/h ταχύτητα είμαι πιο σίγουρος ότι είναι μέτρια.

Αν κάνουμε παρόμοιες ερωτήσεις για μέτρια και για μεγάλη ταχύτητα και παραστήσουμε γραφικά τη βεβαιότητα από 0 έως 1 του οδηγού συναρτήσει της ταχύτητας θα πάρουμε το παρακάτω διάγραμμα (Εικόνα 2-1).



Εικόνα 2-1 : Γραφική παράσταση ασαφών συνόλων

Στο παραπάνω διάγραμμα παρατηρούμε μια λεκτική αναπαράσταση της ταχύτητας. Η ταχύτητα έχει '*διαμεριστεί'* όπως λέμε σε λεκτικούς (*linguistic*) όρους. Από το σημείο αυτό και μετά μπορούμε να μιλάμε για την ταχύτητα με λεκτικούς αντί αυστηρά αριθμητικούς όρους. Ας δούμε τι θα λέγαμε για μια ταχύτητα 32 km/h. Η ταχύτητα αυτή είναι μικρή με βαθμό βεβαιότητας 0.8 και μέτρια με βαθμό βεβαιότητας 0.3. Ο βαθμός αυτός βεβαιότητας ονομάζεται βαθμός συμμετοχής $\mu(x)$ μιας τιμής x στο αντίστοιχο ασαφές σύνολο που εκφράζεται από την συνάρτηση $\mu(x)$. Η περιγραφή μιας μεταβλητής x με λεκτικούς όρους ονομάζεται διαμερισμός της μεταβλητής και η περιγραφή μιας αυστηρά αριθμητικής τιμής με λεκτικούς όρους όπως για παράδειγμα η ταχύτητα των 32 km/h ονομάζεται '*ασαφοποίηση*' (*fuzzyfication*) της crisp τιμής. Ο αναγνώστης μπορεί να πειραματιστεί και να ασαφοποιήσει την ταχύτητα των 78 km/h.

Στο σημείο αυτό θα πρέπει να τονιστεί κάτι που συγχέεται συχνά ακόμα και από έμπειρους χρήστες της ασαφούς λογικής και αποτελεί το σημείο στο οποίο διαχωρίζεται η ασαφής λογική από τη στατιστική ανάλυση. Ένα ασαφές σύνολο εκφράζει κατανομή δυνατότητας (*possibility distribution*) και ένας βαθμός συμμετοχής μιας τιμής σε ένα ασαφές σύνολο αποτελεί το βαθμό βεβαιότητας (*degree of certainty*) ότι η πρόταση που διατυπώνουμε είναι αληθής (π.χ η ταχύτητα των 32 km/h είναι μικρή. Βεβαιότητα: 40%) και όχι κατανομή πιθανότητας (*probability distribution*). Για την αποσαφήνιση της εννοιολογικής διαφοράς μεταξύ πιθανότητας και βεβαιότητας αναφέρεται συχνά στη βιβλιογραφία το παράδειγμα με τα *δέκα αυγά*. Σύμφωνα με το παράδειγμα αυτό η *πιθανότητα* κάποιος στο πρωινό του να έχει φάει δέκα αυγά είναι μικρή αντίθετα η *βεβαιότητα* ότι ο άνθρωπος μπορεί να φάει δέκα αυγά είναι μεγάλη.

Η μορφή των συναρτήσεων του σχήματος (Εικόνα 2-1), που ονομάζονται συναρτήσεις συμμετοχής δεν είναι απαραίτητο να είναι τραπεζοειδής. Μπορεί να είναι τριγωνική, Γκαουσιανή γενικευμένη τραπεζοειδής η ακόμα και μια συγκεκριμένη αριθμητική τιμή. Διάφοροι τύποι συναρτήσεων συμμετοχής που αναπαριστούν ασαφή σύνολα απεικονίζονται στην Εικόνα 2-2.

Εικόνα 2-2:Διάφοροι τύποι συναρτήσεων συμμετοχής.

Η τελευταία περίπτωση ονομάζεται αυστηρό ή crisp ασαφές σύνολο και είναι ένα εκφυλισμένο ασαφές σύνολο. Προσοχή όμως! Η συνάρτηση συμμετοχής πρέπει να έχει συγκεκριμένες ιδιότητες προκειμένου να αναπαραστήσει ένα ασαφές σύνολο: Το πεδίο τιμών της πρέπει να είναι το σύνολο [0,1] και πρέπει να είναι κυρτή (*convex*) (Σημείωση: Η συνάρτηση συμμετοχής δεν είναι κυρτή με την αυστηρά ορισμένη μαθηματική έννοια του όρου. Μπορεί να είναι γνήσια αύξουσα μέχρι την μεγαλύτερη τιμή της και μετά γνήσια φθίνουσα μέχρι το τέλος του πεδίου ορισμού της).

2.1.1 Στοιχεία των ασαφών συνόλων

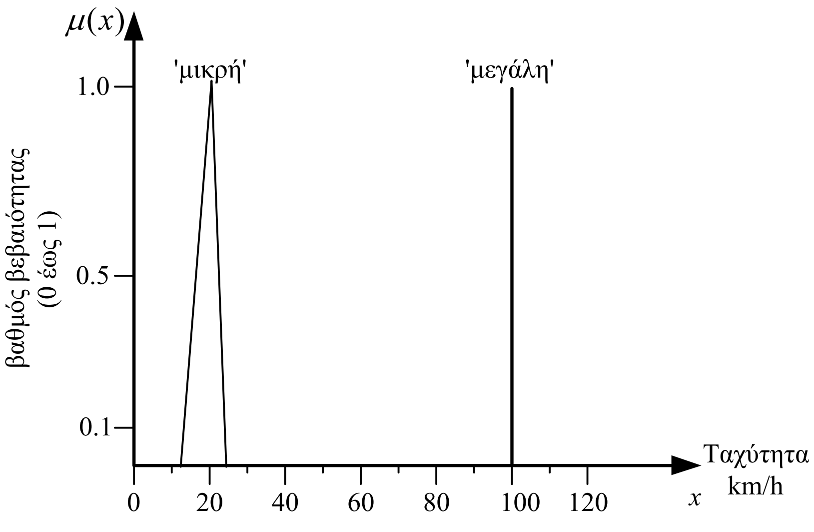
Για την αναπαράσταση ενός ασαφούς συνόλου έχουν καθιερωθεί διάφοροι τρόποι. Όταν ένα ασαφές σύνολο έχει πεδίο ορισμού X που αποτελείται από διακριτές και πεπερασμένες τιμές

$x_1, x_2 ..., x_n$. Το ασαφές σύνολο αναπαρίσταται ως η ένωση των διατεταγμένων ζευγών $x_i / \mu(x_i) : i = 1,2,...n$. Όπου $x_i \in X$ και $\mu(x_i) \in [0,1]$ ο αντίστοιχος βαθμός συμμετοχής της τιμής x_i. Αν $\underline{A}$ είναι ασαφές σύνολο τότε έχουμε την ακόλουθη αναπαράσταση $\underline{A} = \{x_1 / \mu(x_1) + x_2 / \mu(x_2) + ... + x_n / \mu(x_n)\}$. Στην περίπτωση που το X είναι συνεχές και μη πεπερασμένο τότε: $\underline{A} = \left\{\int \mu(x) / x\right\}$ ή απλούστερα $\mu(x)$. Προσοχή χρειάζεται στον παραπάνω συμβολισμό διότι το '+' δεν είναι το σύμβολο της αριθμητικής πρόσθεσης αλλά της ένωσης και το '/' δεν είναι το σύμβολο της διαίρεσης αλλά ένας διαχωριστής για να δείξει το διατεταγμένο ζεύγος 'τιμή / βαθμός συμμετοχής τιμής'. Επίσης το ολοκλήρωμα συμβολίζει την ένωση όλων των διαταγμένων ζευγών που όταν το X είναι συνεχές η ένωση ταυτίζεται με τη $\mu(x)$ που ονομάζεται συνάρτηση συμμετοχής. Το πεδίου ορισμού της *μ(x)* ονομάζεται και *universe of discourse* (***υπερσύνολο αναφοράς***) του ασαφούς συνόλου.

Το υποσύνολο του πεδίου ορισμού της συνάρτησης συμμετοχής για το οποίο το πεδίο τιμών λαμβάνει τιμές ίσες προς τη μονάδα ονομάζεται '*κόρος'* (*core*) του ασαφούς συνόλου. Αν C είναι ο κόρος της $\mu(x)$ τότε γράφουμε:

$x \in C \subseteq \Upsilon \Leftrightarrow \mu(x) = 1$.

Το πλάτος ενός ασαφούς συνόλου που είναι το εύρος του πεδίου ορισμού του, αποτελεί ένα μέτρο της ασάφειας (*fuzziness*) του όρου που περιγράφει, η οποία αντικατοπτρίζει το βαθμό αβεβαιότητας για το συγκεκριμένο όρο. Για παράδειγμα όσο ποιο ξεκάθαρο είναι το γεγονός ότι η ταχύτητα γύρω από τα 20km/h είναι μικρή τόσο ποιο 'στενό' ασαφές σύνολο με κόρο στο 20 χρειάζεται για να την περιγράψει. Όσο όμως η αβεβαιότητα μας για την περιοχή 'μικρή ταχύτητα' αυξάνεται τόσο ποιο 'πλατύ' πρέπει να είναι το ασαφές σύνολο που περιγράφει τον όρο '*μικρή ταχύτητα'*.



Εικόνα 2-3 : Ασάφεια ασαφούς συνόλου

Στο σχήμα (Εικόνα 2-3) παρουσιάζεται ο όρος 'μικρή ταχύτητα' ο οποίος περιγράφει την ταχύτητα γύρω από τα 20 km/h με μικρότερη ασάφεια από τον όρο 'μικρή ταχύτητα' του σχήματος (Εικόνα 2-1). Αν είμαστε απόλυτα βέβαιοι ότι η ταχύτητα των 100 km/h και μόνο αυτή είναι 'μεγάλη ταχύτητα', αυτό εκφράζεται με ένα αυστηρό (crisp) ασαφές σύνολο με κόρο την τιμή 100 και με μηδενικό πλάτος (δες Εικόνα 2-3).

Η *στήριξη (support*) ενός ασαφούς συνόλου είναι το υποσύνολο του πεδίου ορισμού του για το οποίο η συνάρτηση συμμετοχής λαμβάνει μη μηδενικές τιμές. Μαθηματικά αν Y είναι το *universe of discourse* της $\mu(x)$ και S το συμπλήρωμα της γράφουμε:

$x \in S \subseteq \mathrm{U} \Leftrightarrow \mu(x) \neq 0$

Κανονικό ασαφές σύνολο (normal fuzzy set) ονομάζεται το ασαφές σύνολο για το οποίο υπάρχει τουλάχιστον μια τιμή στο πεδίο ορισμού του με βαθμό συμμετοχής μονάδα.

Ύψος ασαφούς συνόλου (Height), ονομάζεται η μέγιστη τιμή της συνάρτησης συμμετοχής. Ένα κανονικό ασαφές σύνολο έχει ύψος 1 ενώ ένα ασαφές σύνολο με ύψος μικρότερο του 1 ονομάζεται *υποκανονικό*.

Τα όρια ενός ασαφούς συνόλου (boundaries) ονομάζονται τα σημεία του πεδίου ορισμού του για τα οποία ισχύει $0 < \mu(x) < 1$ είναι δηλαδή όλα τα στοιχεία του συμπληρώματος του ασαφούς συνόλου τα οποία δεν είναι κόρος.

Κυρτό ασαφές σύνολο (convex fuzzy set) είναι το ασαφές σύνολο το οποίο έχει μονότονα αύξουσα η μονότονα φθίνουσα συνάρτηση συμμετοχής.

2.1.2 Πράξεις μεταξύ ασαφών συνόλων

Μεταξύ ασαφών συνόλων ορίζονται ορισμένες πράξεις οι βασικότερες από τις οποίες είναι:

Α)Η ένωση (union)

Β)Η τομή (intersection)

Γ) Το συμπλήρωμα ενός ασαφούς συνόλου (complement)

Έστω τρία ασαφή σύνολα *A,B,C*, που ορίζονται στο πεδίο ορισμού *X*. Με συναρτήσεις συμμετοχής $\mu_A(x), \mu_B(x), \mu_C(x)$.

Η ένωση δίνεται από τη σχέση:

$$\mu_{A \cup B}(x) = \mu_A(x) \vee \mu_B(x) \forall x \in X \qquad \textit{2-1}$$

Η τομή δίνεται από τη σχέση:

$$\mu_{A \cap B}(x) = \mu_A(x) \wedge \mu_B(x) \forall x \in X \qquad \textit{2-2}$$

Και το συμπλήρωμα ενός ασαφούς συνόλου από τη σχέση:

$$\mu_{\overline{A}}(x) = 1 - \mu_A(x) \forall x \in X \qquad \textit{2-3}$$

Ο αναγνώστης καλείται να παραστήσει γραφικά τις λειτουργίες των σχέσεων 2-1 έως 2-3 για τα ασαφή σύνολα ‘μικρή’, ‘μέτρια’ και ‘μεγάλη’ ταχύτητα.

2.1.3 Τροποποιητές (modifiers) και φράκτες (hedges) ασαφών συνόλων

Τα ασαφή σύνολα χρησιμοποιούνται ως εργαλείο για να εκφραστούν βασικοί όροι της φυσικής ανθρώπινης διαλέκτου (*atoms*), χρειάζονται ένα μηχανισμό παραγωγής τροποποιημένων όρων από τους βασικούς όρους. Αν δηλαδή έχουμε ένα ασαφές σύνολο που εκφράζει τον λεκτικό όρο ‘*νέος’* τότε θα πρέπει από τον όρο αυτό να υπάρχει η δυνατότητα παραγωγής όρων όπως:

```
'Πολύ νέος'          (very young)

'Πολύ πολύ νέος'     (very very young)
```

`‘ελαφρώς νέος’        (slightly young)`

`‘νέος και κάτι’       (plus young)`

`‘νεος παρά κάτι’      (minus young)`

κ.τ.λ. Αν για παράδειγμα ένας λεκτικός όρος ‘Α’ έχει συνάρτηση συμμετοχής $\mu_A(x)$ τότε οι συναρτήσεις συμμετοχής των αντίστοιχων όρων παράγονται ως εξής:

‘Very A’: $\mu_{very\ A}(x) = \mu_A^2(x)$ *2-4*

‘Very Very A’: $\mu_{very\,very\ A}(x) = \mu_A^4(x)$ *2-5*

‘Plus A’: $\mu_{plus\ A}(x) = \mu_A^{1.25}(x)$ *2-6*

‘Minus A’: $\mu_{Minus\ A}(x) = \mu_A^{0.75}(x)$ *2-7*

‘Slightly A’: $\mu_{slightly\ A}(x) = \sqrt{\mu_A(x)}$ *2-8*

κ.τ.λ.

Φυσικά υπάρχουν πάρα πολλοί τροποποιητές (ή φράκτες) π.χ. ο φράκτης ‘μάλλον’ (*rather*). Οι φράκτες μπορούν να οργανωθούν σε υποκατηγορίες ανάλογα με την τροποποίηση που επιτελούν. Για παράδειγμα μπορεί να είναι:

Συγκεντρώσεις: ‘*concentrations*’ Εξισώσεις 2-*4* έως 2-*6*. Ονομάζονται έτσι διότι μειώνουν την τιμή του βαθμού συμμετοχής για όλες τις τιμές που έχουν βαθμό συμμετοχής μικρότερο από ένα.

Διαστολές: ‘*Dilations*’ Εξισώσεις 2-*7* έως 2-*8*. Επιτελούν την αντίστροφη διαδικασία από την συγκέντρωση.

Ενδυναμώσεις : ‘*intensifications*’ αποτελούν συνδυασμό της συγκέντρωσης και της διαστολής. Τιμές της συνάρτησης συμμετοχής μικρότερες από 0.5 γίνονται ακόμα μικρότερες ενώ τιμές μεγαλύτερες από 0.5 γίνονται ακόμα μεγαλύτερες. Η ενδυνάμωση ενός ασαφούς συνόλου μπορεί να εκφραστεί από την σχέση 2-*9*:

$$\mu_{Intensify\ A}(x) = \begin{cases} 2\mu_A^2(x) & if \quad 0 \le \mu_A(x) \le 0.5 \\ 1 - 2[1-\mu_A(x)]^2 & if \quad 0.5 < \mu_A(x) \le 1 \end{cases} \qquad \textit{2-9}$$

Ο αναγνώστης καλείται να υλοποιήσει τις γραφικές παραστάσεις των φρακτών για ένα ασαφές σύνολο. Επίσης καλείται να παράγει το ασαφές σύνολο ‘*not very small’ fruit* από τα ασαφή σύνολα ‘*small*’ και ‘*red*’.

2.1.4 Τομές άλφα (α-cuts)

Έστω ένα ασαφές σύνολο Α με συνάρτηση συμμετοχής $\mu_A(x)$. Η τομή-α (‘$\alpha - cut$’ - προφέρεται “*άλφα κατ*” και όχι εϊ κατ) του ασαφούς συνόλου Α είναι ένα νέο ασαφές σύνολο Α΄ με συνάρτηση συμμετοχής

$$\mu_a(x) = \begin{cases} \mu_A(x) & if \quad 0 \le \mu_A(x) \le a \\ \\ a & if \quad a \le \mu_A(x) \le 1 \end{cases} \qquad \textit{2-10}$$

Είναι προφανές από την παραπάνω σχέση ότι $0 \le \alpha \le 1$. Αν α=1 τότε το νέο ασαφές σύνολο είναι ίδιο με το αρχικό. Αν α<1 τότε το ασαφές σύνολο που προκύπτει είναι υποκανονικό. Ο αναγνώστης καλείται να υλοποιήσει τη γραφική παράσταση για την 0.4-τομή ενός ασαφούς συνόλου. Εφαρμογές των $\alpha - cut\,s$ θα συναντήσουμε παρακάτω.

2.2 Ο ασαφής κανόνας

Εννοιολογικά, ο ασαφής κανόνας είναι ένας μηχανισμός αναπαράστασης της γνώσης, ο οποίος προσιδιάζει στον ανθρώπινο τρόπο σκέψης. Τα ασαφή σύνολα που εκφράζουν λεκτικούς όρους συνδυάζονται μεταξύ τους και δημιουργούν ασαφείς κανόνες που αναπαριστούν τη γνώση που έχουμε για το σύστημα. Ένας ασαφής κανόνας αποτελείται από δύο βασικά μέρη α) το τμήμα υπόθεσης (*premise part*) και β) το τμήμα απόδοσης η απόφασης (*consequent part*). Ένας απλός κανόνας είναι της μορφής:

If x is A then y is B 2-11

Το τμήμα `if x is A` είναι το τμήμα υπόθεσης και το τμήμα `then y is B` το τμήμα απόφασης η συμπεράσματος. Όπου *A* και *B* ασαφή σύνολα. *x* είναι η τιμή μιας μεταβλητής εισόδου, η οποία *ασαφοποιείται (fuzzyfication)*, δηλαδή αποκτά ένα βαθμό συμμετοχής στο ασαφές σύνολο *A*. *y* είναι η έξοδος του συστήματος που εκφράζει την απόφαση του κανόνα και παρέχεται από το μηχανισμό του συμπεράσματος (*inference*) σε ασαφή μορφή. Στη συνέχεια το ασαφές συμπέρασμα από-ασαφοποιείται με το μηχανισμό της αποασαφοποίησης (*defuzzyfication*), και προκύπτει μια *crisp* τιμή που είναι το τελικό αριθμητικό συμπέρασμα που μπορεί να χειριστεί η υπολογιστική μηχανή η ένα αισθητήριο. Ένας κανόνας μπορεί να πάρει διάφορες μορφές εκτός από αυτή της εξίσωσης (2-11). Ο κανόνας της (2-11), του οποίου η έξοδος είναι ένα ασαφές σύνολο ονομάζεται κανόνας τύπου *mamdani* προς τιμή του Ebrahim Mamdani που ήταν από τους πρώτους που εφάρμοσε την ασαφή λογική για να κατασκευάσει ένα ασαφές σύστημα αυτόματου έλεγχου της ταχύτητας μιας ατμομηχανής. Άλλοι κύριοι χαρακτηριστικοί τύποι κανόνων είναι της μορφής:

If x is A then y is c 2-12

Όπου το c είναι αριθμός η μπορεί να θεωρηθεί και *crisp* ασαφές σύνολο και:

If x is A then y is c_0+c_1x 2-13

Όπου $c_0, c_1 \in \Re$. Η σχέση (2-12) προτάθηκε από τους *Sugeno-Takagi* και η επέκταση της (2-13) από τους *Takagi-Sugeno-Kang*. Ο ασαφής κανόνας (2-13) είναι ένας από τους κυριότερους τύπους ασαφούς κανόνα και χρησιμοποιείται ευρύτατα σε εφαρμογές ανάπτυξης ασαφών συστημάτων. Είναι γνωστός ως κανόνας *T-S-K* από τα αρχικά των ερευνητών *Takagi-Sugeno-Kang*. Για περισσότερες από μία εισόδους $x_1, x_2, ..., x_n$ οι κανόνες επεκτείνονται στις αντίστοιχες μορφές:

$if\ x_1\ is\ A_1\ and\ x_2\ is\ A_2\ and\ ...x_n\ is\ A_n\ Then\ y\ is\ B$ 2-14

Φυσικά μπορούμε να έχουμε και περισσότερες από μία εξόδους. Φαίνεται όμως εύκολα ότι ο κανόνας αυτός μπορεί να διασπαστεί σε περισσότερους από ένα κανόνες μιας εξόδου.

2.2.1 Ασαφής συμπερασμός του στοιχειώδους ασαφούς συστήματος ενός κανόνα

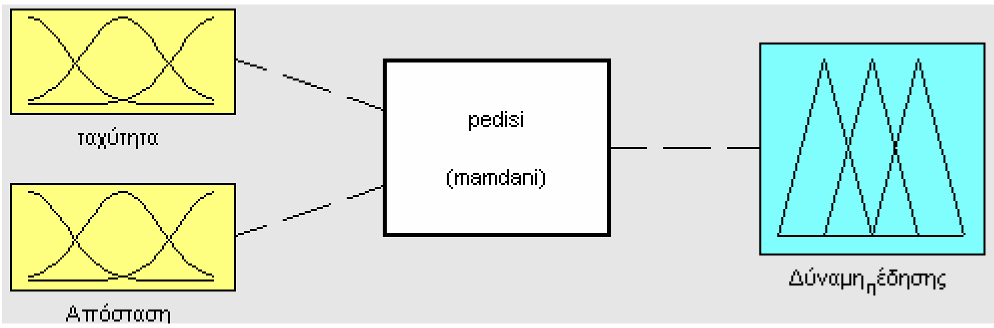
Το βασικό ζητούμενο είναι ο τρόπος λειτουργίας του ασαφούς κανόνα. Για ευκολία θα θεωρήσουμε ένα ασαφές σύστημα με ένα κανόνα. Ο τρόπος λειτουργίας του κανόνα διασπάται σε τρεις μεγάλες φάσεις:

A) Ασαφοποίηση (*fuzzification*)

B) Ασαφής συνεπαγωγή (*fuzzy implication*)

C) Αποασαφοποίηση (*defuzzyfication*)

Για να καταλάβουμε καλύτερα τις φάσεις αυτές μπορούμε να παρακολουθήσουμε τις αναγκαιότητες που εξυπηρετούν. Ας θυμηθούμε λοιπόν ότι η ασαφής λογική δημιουργήθηκε για να βοηθήσει τις υπολογιστικές μηχανές να χειριστούν λεκτικούς όρους. Οι λεκτικοί αυτοί όροι μικρό-μέτριο κ.τ.λ. αποθηκεύονται στη μνήμη του υπολογιστή υπό μορφή συναρτήσεων συμετοχής που $\mu_{A_1}, \mu_{A_2}, ..., \mu_{A_n}$ που εκφράζουν τα αντίστοιχα ασαφή σύνολα $A_1, A_2, ..., A_n$ (δες και §2.1.1 σελ. 6). Ας υποθέσουμε επίσης ότι έχουμε ένα φυσικό σύστημα με $x_1, x_2, ..., x_n$ εισόδους και μια έξοδο y η λειτουργία του οποίου διέπεται από ένα κανόνα της μορφής 2-14. για απλότητα ας θεωρήσουμε ότι θέλουμε να κατασκευάσουμε ένα ευφυές σύστημα το οποίο να φρενάρει ένα αυτοκίνητο με ένα κανόνα που ο οδηγός έχει αποκτήσει από την εμπειρία του. Έστω ότι οι επιλέγουμε δύο εισόδους $x_1 = u$ (ταχύτητα αυτοκινήτου) και $x_2 = d$ (απόσταση από το εμπόδιο).



Εικόνα 2-4: Το σύστημα της πέδησης του αυτοκινήτου με δύο εισόδους

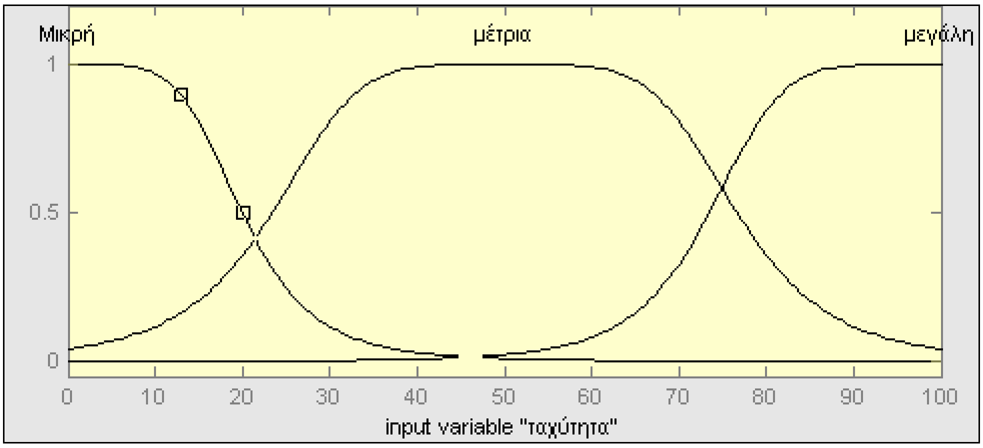
Η έξοδος του κανόνα $y = F$ (δύναμη πέδησης στο φρένο). Ο ειδικός διατυπώνει τον εξής κανόνα:

*Αν η ταχύτητα είναι **μεγάλη και** η απόσταση **μικρή τότε** δύναμη πέδησης μεγάλη* 2-15

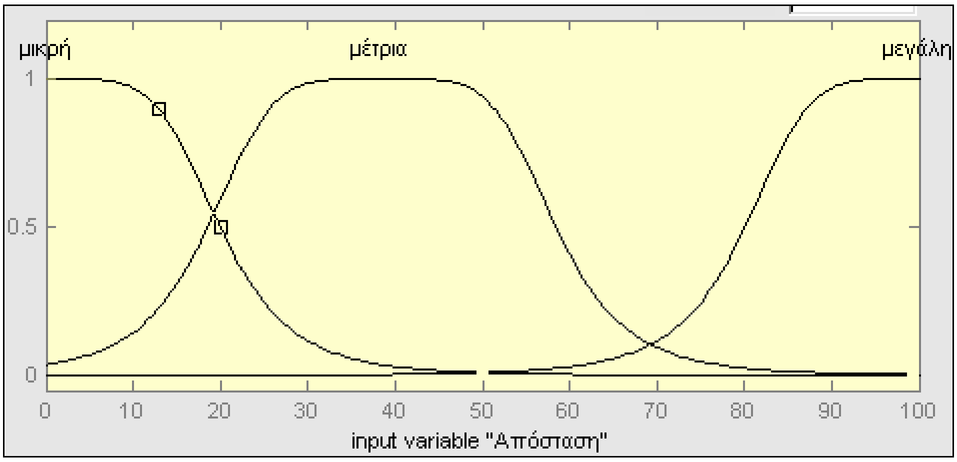
Ας δούμε πως μέσω της ασαφούς λογικής θα δώσουμε στον υπολογιστή να καταλάβει, τι θα μας απαντήσει και πως θα συνεννοηθούμε τελικά.

Ο σχεδιαστής του εκλεκτή θα πρέπει να επιτελέσει τα ακόλουθα βήματα:

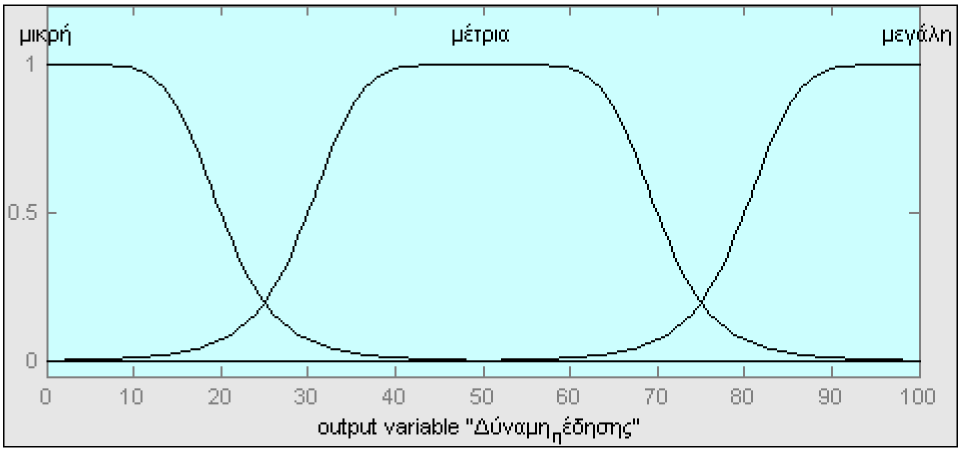
1. *Λεκτικός διαμερισμός των εισόδων:* Θα πρέπει δηλαδή να αναπαραστήσει τις μεταβλητές εισόδου και εξόδου με λεκτικούς όρους. Ένας διαμερισμός των εισόδων και των εξόδων φαίνεται στα σχήματα (Εικόνα 2-5 έως Εικόνα 2-7).



Εικόνα 2-5: Διαμερισμός της Εισόδου 'ταχύτητα' σε λεκτικούς όρους



Εικόνα 2-6: Διαμερισμός της εξόδου Απόσταση σε λεκτικούς όρους

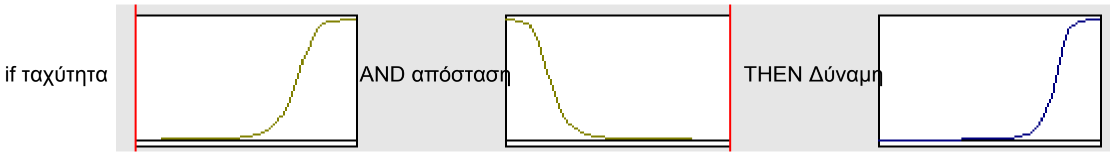


Εικόνα 2-7: Διαμερισμός της εξόδου 'Δύναμη πέδησης' σε λεκτικούς όρους

Ο αριθμός των λεκτικών όρων που θα διαμερίσουν μια μεταβλητή η μορφή και οι ακριβείς τους θέσεις καθορίζονται από το σχεδιαστή του συστήματος και αποτελεί ακόμα και σήμερα ένα από τα ποιο κρίσιμα και ενεργά πεδία έρευνας στον τομέα των ασαφών συστημάτων. Επίσης δεν είναι απαραίτητο όλα τα ασαφή σύνολα να είναι ίδιου τύπου. Στο συγκεκριμένο παράδειγμα επιλέγονται γενικευμένες τραπεζοειδής για την είσοδο και για την έξοδο τριγωνικές συναρτήσεις συμμετοχής.

2. *Διατύπωση των κανόνων:* Έχοντας διαμερίσει τις εισόδους και τις εξόδους τα ασαφή σύνολα μπορούν να αποθηκευτούν στον υπολογιστή υπό μορφή συναρτήσεων. Στη συνέχεια

διατυπώνονται οι κανόνες. Στην περίπτωση μας έχουμε ένα κανόνα (σχέση 2-15 σελ. 11). Επιλέγοντας τα αντίστοιχα ασαφή σύνολα από κάθε μεταβλητή ο κανόνας μπορεί να παρασταθεί γραφικά στην Εικόνα 2-8.



Εικόνα 2-8: Γραφική παράσταση του ασαφούς κανόνα

3. *Καθορισμός του τύπου της ασαφούς συνεπαγωγής (fuzzy implication)*: Για να κατανοήσουμε τις παραμέτρους της ασαφούς συνεπαγωγής θα πρέπει πρώτα να κατανοήσουμε το μηχανισμό της, δηλαδή τον τρόπο με τον οποίο λειτουργεί ο κανόνας. Ας υποθέσουμε λοιπόν ότι κατασκευάζεται το ασαφές σύστημα της πέδησης του αυτοκινήτου με τον *ένα* κανόνα που απεικονίζεται στην Εικόνα 2-8. Οι αισθητήρες που μετρούν την απόσταση και την ταχύτητα του αυτοκινήτου δίνουν *ταχύτητα= 70 km/h και απόσταση 30 m*. Οι τιμές αυτές εισάγονται στο ασαφές σύστημα και ασαφοποιούνται. Δηλαδή η ταχύτητα των 70 km/h είναι *μεγάλη* με βαθμό βεβαιότητας 0.4 και η απόσταση των 30 m είναι μικρή με βαθμό βεβαιότητας 0.2. Το ερώτημα είναι: πως θα ενεργοποιηθεί ο κανόνας που δίνεται στην Εικόνα 2-8 για δώσει αποτέλεσμα? Ο κανόνας λέει:

Αν η ταχύτητα είναι ***μεγάλη (είναι με βεβαιότητα 0.4)***

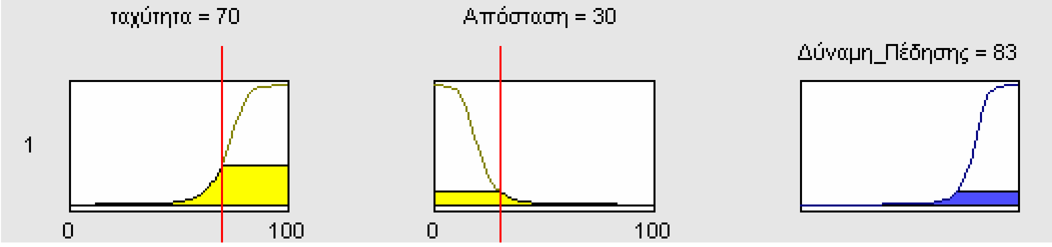
και

η απόσταση ***μικρή (είναι με βεβαιότητα 0.2)***

τότε

δυναμη πέδησης μεγάλη

Προφανώς θα πρέπει να βρούμε ένα τρόπο να υλοποιήσουμε το ‘***και (and)*** ’ και το ‘***τότε***’. Ο τρόπος με τον οποίο υλοποιείται αριθμητικά το ***και*** καθορίζεται από τον τύπο του ασαφούς συμπερασμού. Υπάρχουν διάφοροι τρόποι υλοποίησης του *AND* οι κυριότεροι είναι με τον τελεστή *MIN* και με τον τελεστή *product*. Ο τελεστής *Min* λαμβάνει το μικρότερο από τους βαθμούς συμμετοχής και παράγει το λεγόμενο *βαθμό εκπλήρωσης* (*degree of fulfillment*) του κανόνα. Ο τελεστής του *γινομένου* υπολογίζει το βαθμό εκπλήρωσης του κανόνα ως το *αριθμητικό γινόμενο* των βαθμών συμμετοχής των ασαφοποιημένων τιμών. Στην περίπτωση μας αν εφαρμοστεί το MIN ο κανόνας έχει βαθμό εκπλήρωσης 0.2. Αν εφαρμοστεί το γινόμενο ο κανόνας έχει βαθμό εκπλήρωσης 0.08. Εννοιολογικά ο βαθμός εκπλήρωσης του κανόνα εκφράζει τη βαρύτητα που έχει το αποτέλεσμα του κανόνα. Η βαρύτητα αυτή εκφράζεται (για συστήματα mamdani) με το αντίστοιχο *α-cut* του ασαφούς συνόλου που εκφράζει το αποτέλεσμα του κανόνα. Συνεπώς αν είναι w ο βαθμός εκπλήρωσης του κανόνα το αποτέλεσμα της εφαρμογής του είναι το *α-cut* ασαφές σύνολο της εξόδου του με $\alpha = w$. Στην Εικόνα 2-9 παριστάνεται γραφικά ο μηχανισμός του ασαφούς συμπερασμού για τις τιμές *ταχύτητα = 70 και απόσταση = 30*. Το αποτέλεσμα είναι το σκιασμένο ασαφές σύνολο της εξόδου.



Εικόνα 2-9: Γραφική παράσταση του ασαφούς συμπερασμού.

Παρατηρώντας το αποτέλεσμα του κανόνα βλέπουμε ότι αυτό εκφράζεται από ένα υποκανονικό ασαφές σύνολο. Το γεγονός ότι είναι υποκανονικό δεν μας πειράζει καθόλου. Το ερώτημα όμως είναι το πως θα αξιοποιηθεί από το μηχανισμό που θα ασκήσει την πίεση στο φρένο. Το μηχανικό σύστημα που θα ασκήσει την πίεση καταλαβαίνει μόνο σαφείς αριθμητικές τιμές και όχι ασαφείς όρους. Στο σημείο αυτό αναφέρεται το τελικό βήμα της σχεδίασης

4. Μέθοδος *αποασαφοποίησης* (*defuzzyfication*). Η διαδικασία της αποασαφοποίησης είναι αντίθετη αυτής της ασαφοποίησης και παράγει μια αυστηρά αριθμητική (τιμή crisp τιμή) από ένα ασαφές σύνολο. Είναι δηλαδή μια απεικόνιση, η οποία απεικονίζει ένα ασαφές σύνολο σε ένα πραγματικό αριθμό. Υπάρχουν διάφορες μέθοδοι αποασαφοποίησης:

   I. *Αποασαφοποίηση κεντρικής τιμής (centroid defuzzyfication or center of area)*: Σύμφωνα με τη μέθοδο αυτή υπολογίζεται το *κέντρο βάρους* της κατανομής του ασαφούς συνόλου δίνεται από τη σχέση:

   $$x^* = \frac{\int x \cdot \mu(x)dx}{\int \mu(x)dx} \qquad 2\text{-}16$$

   ο υπολογισμός της κεντρικής τιμής της σκιασμένης επιφάνειας του ασαφούς συνόλου της εξόδου δίνει τον πραγματικό αριθμό 83. Αυτή είναι και η τιμή, η οποία θα δοθεί στο μηχανικό σύστημα το οποίο θα πιέσει το φρένο. Αν δηλαδή η δύναμη πέδησης μετριέται π.χ. σε Ν τότε στο φρένο θα πρέπει να ασκηθεί δύναμη 83Ν.

   II. *Αποασαφοποίηση μέσου όρου των μεγίστων (mean of maxima or mom)*: σύμφωνα με τη μέθοδο αυτή υπολογίζεται ο μέσος όρος των μέγιστων τιμών του ασαφούς συνόλου της εξόδου. Υπενθυμίζεται για πολλοστή φορά ότι το ασαφές σύνολο το οποίο αποασαφοποιείται είναι η σκιασμένη περιοχή της εξόδου του κανόνα (Εικόνα 2-9). Σύμφωνα με τη μέθοδο αυτή λαμβάνονται οι τιμές του πεδίου ορισμού που έχουν το μεγαλύτερο βαθμό συμμετοχής και υπολογίζεται η μέση τιμή τους. Η εφαρμογή της μεθόδου mom στο παράδειγμα που εξετάζουμε δίνει αποτέλεσμα *86.5 N*.

   III. *Αποασαφοποίηση άθροισης των μεγίστων (sum of maxima som)*: Η μέθοδος αυτή υπολογίζει το άθροισμα των μέγιστων τιμών.

2.2.2 Ασαφής συμπερασμός συστήματος με περισσότερους του ενός ασαφείς κανόνες.

Τα περισσότερα ασαφή συστήματα εμπλέκουν περισσότερους του ενός κανόνες. Στο παράδειγμα της πέδησης του αυτοκινήτου μπορούμε να προσθέσουμε περισσότερους κανόνες και να δημιουργήσουμε τη λεγόμενη *βάση κανόνων (rule base)*. Έστω λοιπόν ότι δημιουργούμε την ακόλουθη βάση που αποτελείται από τρεις κανόνες:

R^1 : **Αν ταχύτητα** μεγάλη και **απόσταση** μικρή τότε **δύναμη πέδησης** μεγάλη

R^2 : **Αν ταχύτητα** μεγάλη και **απόσταση** μέτρια τότε **δύναμη πέδησης** μέτρια

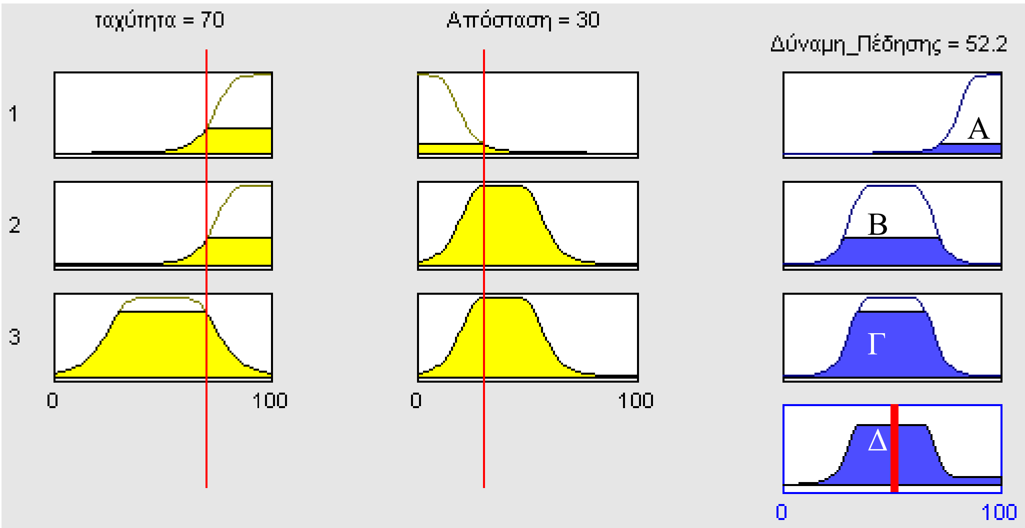
R^3 : **Αν ταχύτητα** μέτρια και **απόσταση** μέτρια τότε **δύναμη πέδησης** μέτρια

Εικόνα 2-10: Λεκτική διατύπωση τριών κανόνων

Για το σχεδιασμό του συστήματος θα πρέπει να ακολουθήσουμε τα βήματα της προηγούμενης παραγράφου. Συνεπώς:

1. *Λεκτικός διαμερισμός των εισόδων*: όπως και στην προηγούμενη παράγραφο (Εικόνα 2-5 έως Εικόνα 2-7).
2. *Διατύπωση των κανόνων*: δες Εικόνα 2-10.
3. *Καθορισμός του τύπου της ασαφούς συνεπαγωγής (fuzzy implication).* Επιλέγουμε *MIN*.
4. *Αποασαφοποίηση:* Επιλέγουμε *αποασαφοποίηση κεντρικής τιμής.*

Στο σημείο αυτό απαιτείται ένα ακόμα σημαντικότατο βήμα στο σχεδιασμό μετά το βήμα 3 και πρίν το βήμα 4. Το βήμα της Ασαφούς συνάθροισης των κανόνων (fuzzy aggregation). Για να κατανοήσουμε το βήμα αυτό ας παραστήσουμε γραφικά τους τρείς κανόνες (Εικόνα 2-10) και ας παρακολουθήσουμε τη λειτουργία τους. Με δεδομένο το λεκτικό διαμερισμό των εισόδων και της εξόδου (Εικόνα 2-5 έως Εικόνα 2-7), επιλέγουμε τα κατάλληλα ασαφή σύνολα οπότε οι κανόνες στην Εικόνα 2-10 απεικονίζονται γραφικά στην Εικόνα 2-11.

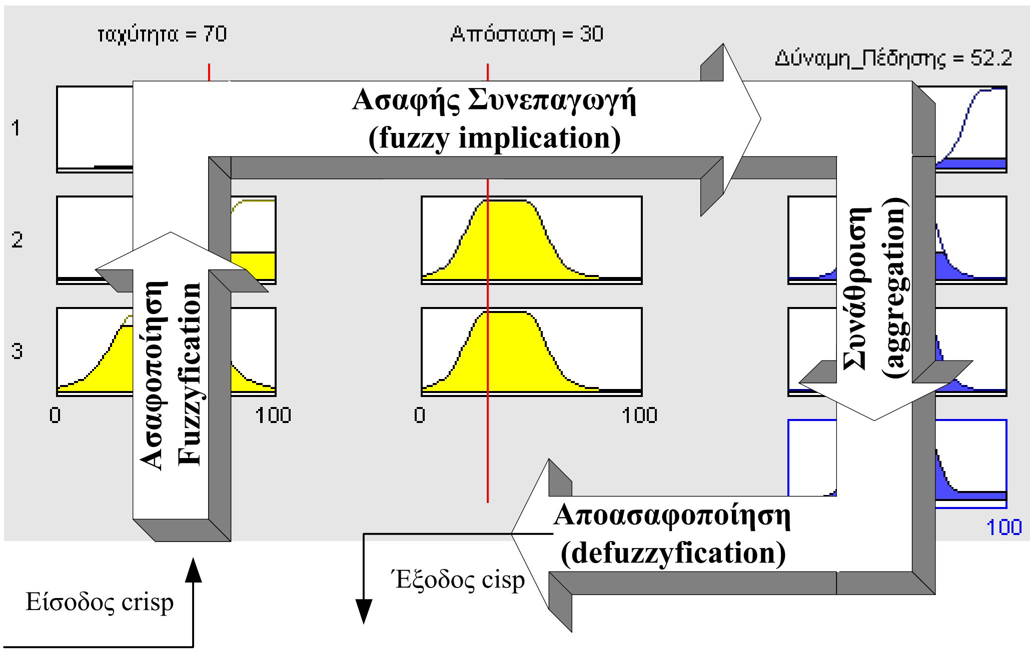


Εικόνα 2-11: Ασαφές σύστημα τριών κανόνων.

Για τιμές εισόδου [70,30] παρατηρούμε πως ασαφοποιούνται οι τιμές, πως ενεργοποιείται κάθε κανόνας και τα αντίστοιχα α-cuts που δημιουργούνται στις εξόδους κάθε κανόνα. Ενδιαφέρον παρουσιάζει το γεγονός ότι στην διατύπωση των κανόνων (Εικόνα 2-10) οι κανόνας 2 και 3 έχουν το ίδιο τμήμα απόφασης. Παρατηρούμε όμως για συγκεκριμένες τιμές (Εικόνα 2-11) τα συμπεράσματα του κανόνα 2 και 3 είναι διαφορετικά ασαφή σύνολα (σκιασμένες περιοχές) γεγονός που οφείλεται στο διαφορετικό βαθμό εκπλήρωσης κάθε κανόνα για τις ίδιες τιμές εισόδου. Άρα λοιπόν για ταχύτητα=70 km/h και απόσταση =30 m, οι κανόνες ενεργοποιούνται και κάθε ένας προτείνει το δικό του ασαφές συμπέρασμα ο κανόνας 1 προτείνει το ασαφές σύνολο A (σκιασμένη περιοχή), ο κανόνας 2 προτείνει το B (σκιασμένη περιοχή) και ο κανόνας 3 προτείνει το ασαφές σύνολο Γ (σκιασμένη περιοχή). Ενώ στην προηγούμενη παράγραφο είχαμε ένα ασαφές συμπέρασμα το οποίο αποασαφοποιήσαμε εδώ έχουμε τρία ασαφή συμπεράσματα. Τα τρία αυτά συμπεράσματα θα πρέπει να τα συνθέσουμε καταρχήν σε ένα ασαφές συμπέρασμα ο μηχανισμός που υλοποιεί αυτή τη σύνθεση ονομάζεται *ασαφής συνάθροιση των κανόνων (fuzzy aggregation)*. Υλοποιείται με διάφορους τρόπους ο βασικότερος είναι με την εφαρμογή του τελεστή *MAX*. Εφαρμόζοντας τον τελεστή *MAX* στα ασαφή σύνολα Α,Β,Γ, προκύπτει το ασαφές σύνολο Δ σύμφωνα με τη σχέση 2-1 σελ. 8. Στη συνέχεια το ασαφές αυτό σύνολο *αποασαφοποιείται* με εφαρμογή μιας μεθόδου αποασαφοποίησης (π.χ. centroid) και προκύπτει η τελική πραγματική τιμή που είναι το αποτέλεσμα του συστήματος των τριών κανόνων. Άρα

Είσοδος: {ταχύτητα=70 km/h , Απόσταση =30 m} $\rightarrow$ Έξοδος {Δύναμη πέδησης 52.2 N}.

Ο μηχανισμός του ασαφούς συμπερασμού μπορεί να παρασταθεί γραφικά στο παρακάτω διάγραμμα ροής:

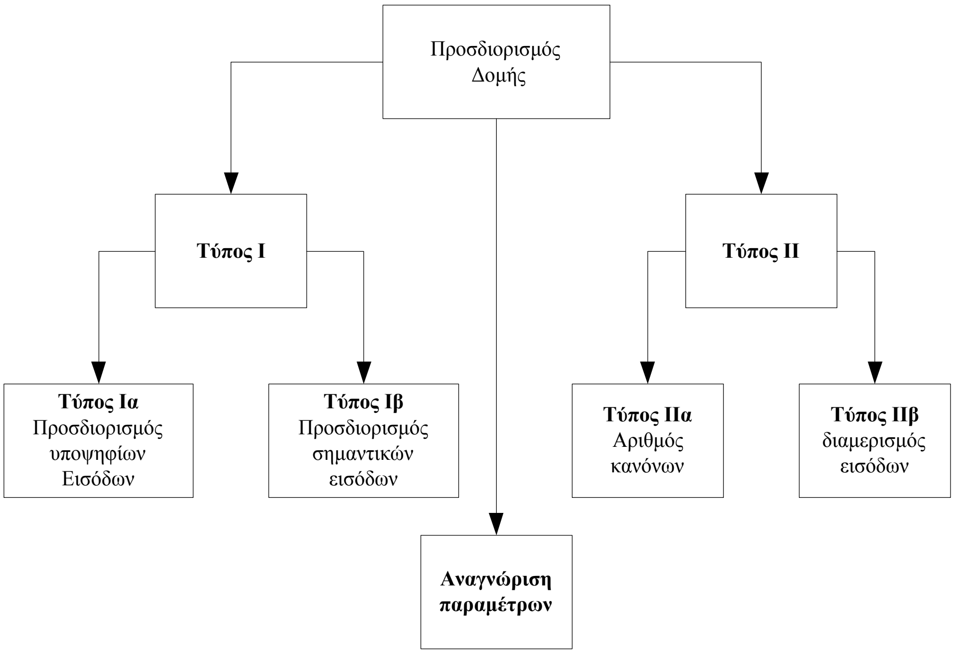


Εικόνα 2-12:Διάγραμμα ροής του Ασαφούς Συμπερασμού.

2.3 Εξαγωγή κανόνων από αριθμητικά δεδομένα

Στα προηγούμενα παραδείγματα διατυπώσαμε κανόνες που προέκυψαν από την εμπειρία ενός ειδικού. Συχνά όμως το σύστημα που θέλουμε να μοντελοποιήσουμε η να ελέγξουμε αντιμετωπίζεται ως ‘μαύρο κουτί’ και η πληροφορία που είναι διαθέσιμη για τη λειτουργία του είναι διαθέσιμη ως υπό μορφή ζευγών δεδομένων εισόδου-εξόδου που εκφράζουν διέγερση-απόκριση αντίστοιχα. Ο σχεδιαστής του συστήματος πρέπει να αναγνωρίσει τον αριθμό και τη θέση των κανόνων από τα δεδομένα. Το πρόβλημα αυτό είναι γνωστό στη βιβλιογραφία ως προσδιορισμός δομής (structure identification) και αποτελεί ένα από το ποιο ενεργά πεδία έρευνας στον τομέα των ασαφών συστημάτων.

Οι Sugeno-Yasukava συστηματοποίησαν το πρόβλημα του προσδιορισμού δομής στο διάγραμμα της Εικόνα 2-13. υποστηρίζουν μάλιστα ότι η σημαντικότητα του κάθε βήματος είναι αντίστοιχα 100:10:1 για τύπο I:II:προσδιορισμός παραμέτρων. Ο τύπος I σχετίζεται με τον προσδιορισμό των εισόδων του συστήματος. Ο τύπος Ια παριστάνει την επιλογή των εισόδων που επιδρούν στο σύστημα και ο μόνος τρόπος επιλογής είναι διαισθητικός. Η επιλογή αυτή καταλήγει σε ένα σύνολο υποψηφίων εισόδων που σύμφωνα με τη γνώση του ειδικού επηρεάζουν τη λειτουργία του συστήματος. Από αυτό το σύνολο κάποιες είσοδοι μπορεί να μην επιδρούν σημαντικά οπότε πρέπει να αφαιρεθούν. Στη βιβλιογραφία έχουν αναπτυχθεί μέθοδοι που με δεδομένο το σύνολο των υποψηφίων εισόδων υπολογίζουν ένα υποσύνολο από εισόδους που επιδρούν σημαντικά στην έξοδο του συστήματος. Ο προσδιορισμός του υποσυνόλου αυτού είναι ο προσδιορισμός δομής τύπου Ιβ.



Εικόνα 2-13: Το πρόβλημα του προσδιορισμού δομής του ασαφούς συστήματος.

Με δεδομένο το υποσύνολο των σημαντικών εισόδων ακολουθεί ο διαμερισμός τους σε λεκτικούς όρους (τύπος ΙΙ). Μετά το διαμερισμό των εισόδων (τύπος ΙΙβ) πρέπει να καθοριστεί ο αριθμός των κανόνων (τύπος ΙΙα) και οι ακριβείς τους θέσεις (αναγνώριση των παραμέτρων) που καθορίζονται από τον προσδιορισμό των παραμέτρων που καθορίζουν τη θέση και το πλάτος κάθε εμπλεκόμενου ασαφούς συνόλου (π.χ. προκειμένου για γκαουσιανές συναρτήσεις συμμετοχής πρέπει να καθοριστεί το μέσο m και η τυπική απόκλιση σ, $\exp(-\frac{(x-m)^2}{\sigma^2})$) κάθε μιας. Ο υπολογισμός των παραμέτρων ονομάζεται συχνά και εκπαίδευση του ασαφούς συστήματος και επιτυγχάνεται με μεθόδους μη γραμμικής βελτιστοποίησης όπως Γενετικούς η γενικότερα εξελικτικούς αλγόριθμους, μεθόδους κλίσης όπως back-propagation, Levemberg-marquat, κ.τ.λ. Όσον αφορά τον αριθμό των κανόνων μια μεγάλη ομάδα μεθόδων της βιβλιογραφίας στηρίζεται σε μεθόδους ομαδοποίησης δεδομένων (*clustering*).

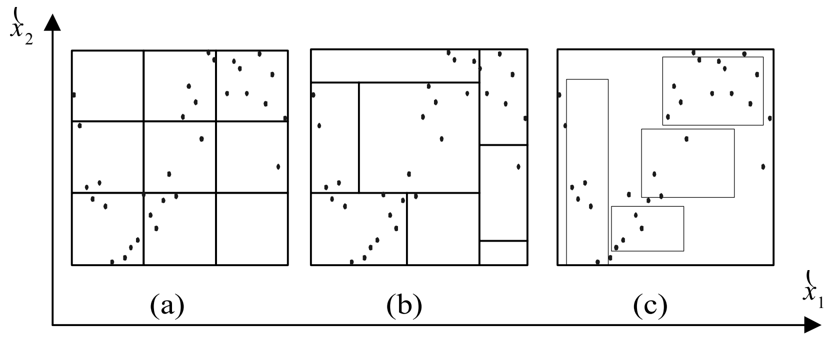
2.3.1 Τύποι διαμερισμού του χώρου των εισόδων

Υπάρχουν τρεις τύποι διαμερισμού του χώρου των εισόδων :

Διαμερισμός τύπου πλέγματος: Σύμφωνα με το διαμερισμό αυτό κάθε ασαφής μεταβλητή εισόδου $x_i \; |i = 1,2,...m$, όπου m ο αριθμός των εισόδων του μοντέλου, διαμερίζεται σε ένα προκαθορισμένο αριθμό px_i από λεκτικούς όρους. Αν $\widetilde{X}_i$ είναι το σύνολο των λεκτικών όρων που διαμερίζουν ασαφώς την x_i τότε ο χώρος των εισόδων διαμερίζεται σε $\prod_{i=1}^{m} px_i$ ασαφείς υποπεριοχές. Οι ασαφείς αυτές υποπεριοχές σχηματίζονται με όλους τους δυνατούς συνδυασμούς των λεκτικών όρων που εκφράζονται από το καρτεσιανό γινόμενο $\widetilde{X}_1 \times \widetilde{X}_2 \times \cdots \times \widetilde{X}_m$. Η αντιμετώπιση αυτή έχει το βασικό μειονέκτημα ότι ο αριθμός των υποπεριοχών που σχηματίζονται αυξάνεται εκθετικά με τον αριθμό των διαστάσεων.

Διαμερισμός τύπου δένδρου **:** Ο διαμερισμός αυτός δημιουργεί μια ιεραρχική δομή (*Hierarchical structure*), σύμφωνα με την οποία η πρώτη μεταβλητή εισόδου εξετάζεται με ένα σύνολο από κριτήρια για το αν θα πρέπει να διαιρεθεί σε δύο ασαφείς υποπεριοχές. Κάθε ασαφής υποπεριοχή εξετάζεται ως προς την επόμενη μεταβλητή για το αν θα πρέπει να διαιρεθεί σε υποπεριοχές κ.τ.λ. Ο αλγόριθμος συνεχίζεται αναδρομικά μέχρι την εξέταση όλων των μεταβλητών. Το μειονέκτημα αυτής της αντιμετώπισης είναι ότι το δένδρο που δημιουργείται από την αναδρομική διαδικασία αυξάνει απαγορευτικά σε μέγεθος, δημιουργώντας μεγάλο αριθμό από υποπεριοχές. Ο αριθμός των υποπεριοχών που δημιουργούνται είναι φυσικά αρκετά μικρότερος από τον αριθμό που δημιουργείται από το διαμερισμό τύπου πλέγματος. Επίσης, μειονέκτημα του διαμερισμού αυτού είναι η δημιουργία υποπεριοχών χωρίς δεδομένα, άρα κανόνων με χαμηλό πληροφοριακό περιεχόμενο και τέλος, οι μεταβλητές εξετάζονται μία προς μία, γεγονός που μπορεί να οδηγήσει το διαμερισμό σε λύσεις υποβέλτιστες.

*Διαμερισμός διανεμημένου τύπου***:** Σύμφωνα με τον διαμερισμό αυτό κάθε υποπεριοχή του χώρου των εισόδων καλύπτεται ανεξάρτητα από τις άλλες υποπεριοχές. Η κάλυψη αυτή των υποπεριοχών οδηγείται από τα δεδομένα του συνόλου δεδομένων εισόδου εξόδου που είναι διαθέσιμα. Ο τύπος αυτός είναι περισσότερο *ευέλικτος* από τους προηγούμενους και μπορεί να διαμερίσει το χώρο των εισόδων σε περιορισμένο αριθμό υποπεριοχών ανεξάρτητα από τον αριθμό των εισόδων. Κάθε υποπεριοχή είναι ανεξάρτητη από τις άλλες και έχει τη δυνατότητα να καλύψει μόνο εκείνες τις περιοχές του χώρου που είναι απαραίτητο να καλυφτούν. Η αντιμετώπιση αυτή δεν παρουσιάζει το μειονέκτημα της '*κατάρας των διαστάσεων*' (*The curse of dimensionallity*). Κάθε διαμερισμός πρέπει να ικανοποιεί ένα σύνολο από κριτήρια τα οποία καθορίζουν την ποιότητα του. Στην Εικόνα 2-14 απεικονίζονται οι τρεις τύποι διαμερισμού του χώρου των εισόδων.



Εικόνα 2-14: Οι τρεις τύποι διαμερισμού του χώρου των εισόδων. (a) Διαμερισμός τύπου πλέγματος, (b) Διαμερισμός τύπου δένδρου και (c) Διαμερισμός διανεμημένου τύπου

Στη συνέχεια δίνεται ένα παράδειγμα προσδιορισμού της δομής ενός ασαφούς συστήματος από αριθμητικά δεδομένα που βασίζεται σε διαμερισμό του χώρου των εισόδων διανεμημένου τύπου.

2.3.2 Παράδειγμα προσδιορισμού δομής ασαφούς συστήματος από αριθμητικά δεδομένα

Πρόβλημα **-** Για την αξιολόγηση των φοιτητών σε γραπτές εξετάσεις λαμβάνεται υπόψη εκτός από το βαθμό που έγραψε ένας φοιτητής και η δυσκολία του εξεταζόμενου θέματος. Ο βαθμός που γράφει ένας φοιτητής κυμαίνεται στο εύρος [0,10] και η δυσκολία του εξεταζόμενου θέματος στο εύρος [10,100]. Ενδεικτικές τιμές φαίνονται στον παρακάτω πίνακα:

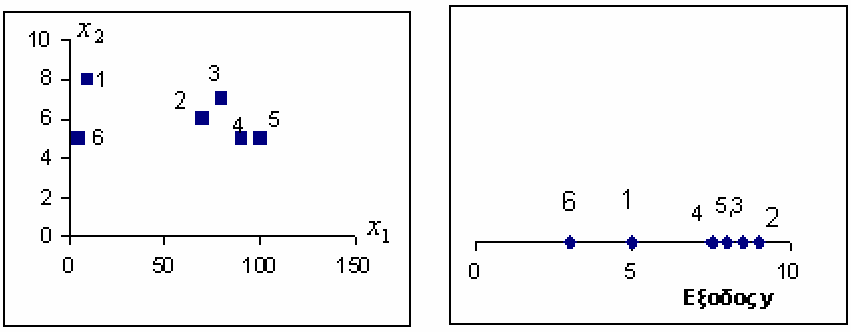
Βαθμός Δυσκολίας [10,100] Είσοδος x_1	Βαθμός Φοιτητή [0,10] Είσοδος x_2	Τελικός Βαθμός Έξοδος→y
10	8	5
70	6	9
80	7	8
90	5	7
100	5	8
5	5	3

Πίνακας 2-1: Προδιαγραφές μοντέλου εξέτασης

Να σχεδιαστεί ένα ασαφές σύστημα το οποίο να αποφασίζει για την τελική βαθμολογία ενός φοιτητή.

Αντιμετώπιση- Προφανώς θα πρέπει να προσδιορίσουμε τη δομή του συστήματος. Θεωρώντας ότι και οι δύο είσοδοι είναι σημαντικές θα ασχοληθούμε μόνο με τον προσδιορισμό δομής τύπου Ιιβ δηλαδή τον προσδιορισμό του αριθμού των κανόνων και το διαμερισμό των εισόδων και της εξόδου. Αν ακολουθήσουμε τη φιλοσοφία της ομαδοποίησης των δεδομένων (*clustering*) ο στόχος μας είναι να ομαδοποιήσουμε τα δεδομένα. Εργαζόμαστε λοιπόν ως εξής:

Στο χώρο των εισόδων $x_1 - x_2$ και στο χώρο της εξόδου y απεικονίζονται τα δεδομένα υπό μορφή διαγράμματος διασποράς (*scatter plot*).



Εικόνα 2-15:Διάγραμμα διασποράς των δεδομένων στο χώρο εισόδων και εξόδου

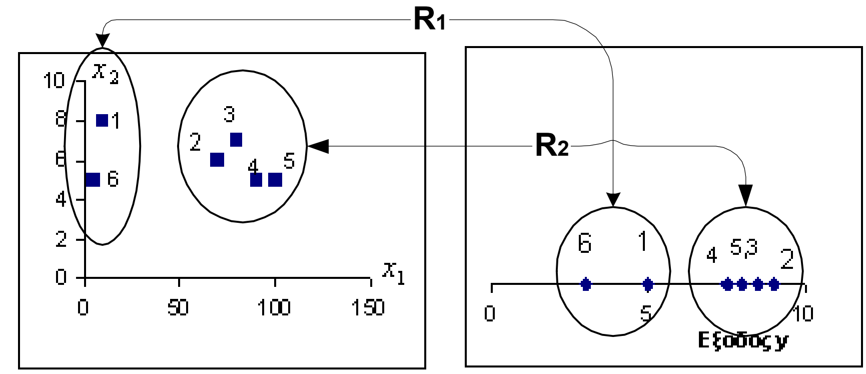
Διακρίνουμε δύο ομάδες δεδομένων. Η μία είναι για μικρό x_1 και μέτριο προς μεγάλο x_2 οπότε η έξοδος είναι μέτρια. Η άλλη είναι για μεγάλο x_1 και μέτριο x_2 οπότε η έξοδος είναι μεγάλη. Παρατηρούμε λοιπόν ότι ή φιλοσοφία του συστήματος διακρίνεται από δύο ασαφείς κανόνες:

R1: *Αν ο βαθμός δυσκολίας είναι μικρός και ο βαθμός που γράφει ο φοιτητής είναι μέτριος προς μεγάλος τότε ο τελικός βαθμός του είναι μέτριος.*

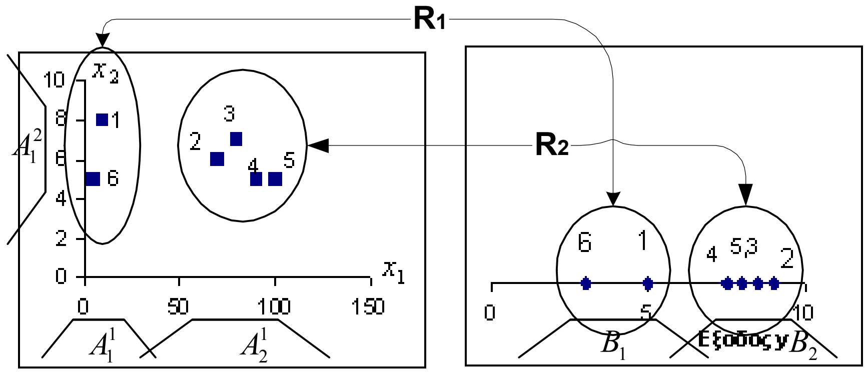
R2: *Αν ο βαθμός δυσκολίας είναι πολύ μεγάλος και ο βαθμός που γράφει ο φοιτητής είναι μέτριος τότε ο τελικός βαθμός του είναι μεγάλος προς μέτριο.*

Εικόνα 2-16:Εξαγωγή κανόνων από δεδομένα

Στο παρακάτω σχήμα παριστάνεται και γραφικά η ομαδοποίηση:



Εικόνα 2-17: Γραφική παράσταση της ομαδοποίησης των δεδομένων.

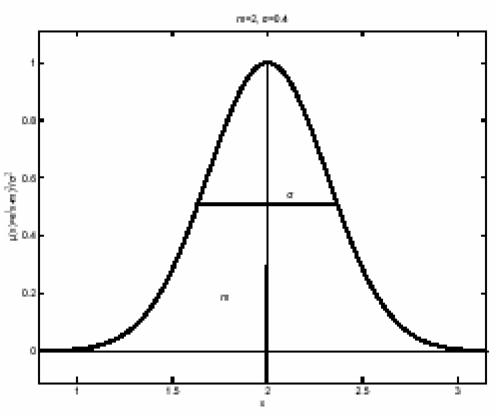


Εικόνα 2-18:Τοποθέτηση των ασαφών συνόλων διαμερισμού των εισόδων και της εξόδου.

Χρειαζόμαστε λοιπόν δύο κανόνες για να περιγράψουμε το σύστημα και κατά μια έννοια έχουμε επιτύχει και την εξαγωγή γνώσης (*knowledge extraction*) από τα δεδομένα υπό μορφή λεκτικών κανόνων (*rule extraction*). Το επόμενο βήμα είναι να γίνει ο διαμερισμός των εισόδων και της εξόδου. Η διαδικασία αυτή μπορεί να γίνει και πάλι γραφιστικά. Η είσοδος x_1 θα διαμεριστεί σε δύο ασαφείς όρους μικρό και πολύ μεγάλο. Η είσοδος x_2 θα διαμεριστεί σε μια περιοχή που εκφράζει το λεκτικό όρο μέτριο προς μεγάλο. Περίεργο? Καθόλου! Τα δεδομένα που έχουμε περιγράφονται έτσι. Αν είχαμε περισσότερα δεδομένα πιθανόν να είχαμε και περισσότερους κανόνες. Η έξοδος θα διαμεριστεί σε δύο λεκτικούς όρους μέτρια και μεγάλη. Ας δούμε τώρα πώς θα τοποθετήσουμε τα ασαφή σύνολα που αντιστοιχούν στους ασαφείς όρους. Ας υποθέσουμε ότι θέλουμε να χρησιμοποιήσουμε γκαουσιανές συναρτήσεις συμμετοχής για να περιγράψουμε τα ασαφή σύνολα. Οι συναρτήσεις αυτές δίνονται από τη σχέση:

$$\mu(x) = e^{-\frac{(x-m)^2}{\sigma^2}} \qquad 2\text{-}17$$

Στην Εικόνα 2-19 παρουσιάζεται η φυσική ερμηνεία των παραμέτρων της συνάρτησης συμμετοχής. Η παράμετρος m καθορίζει τη θέση της ενώ η παράμετρος σ το πλάτος που εκφράζει όπως έχουμε πει την ασάφεια του ασαφούς συνόλου.



Εικόνα 2-19:Παράμετροι γκαουσιανής

Με δεδομένο ένα σύνολο σημείων είναι προφανές ότι το κέντρο m θα προκύψει από το μέσο όρο των προβολών των σημείων που ανήκουν σε μια ομάδα στον αντίστοιχο άξονα. Η παράμετρος σ μπορεί να προκύψει υπολογίζοντας την τυπική απόκλιση της κατανομής. Η τυπική απόκλιση εκφράζει το πόσο συγκεντρωμένα είναι τα σημεία γύρω από τη μέση τιμή. Αν τα σημεία είναι πολύ συγκεντρωμένα γύρω από τη μέση τιμή η τυπική απόκλιση είναι μικρή άρα και το ασαφές σύνολο στενό (Λογικό? Γιατί?). Αντίθετα αν τα σημεία παρουσιάζουν μεγάλη διασπορά η τυπική απόκλιση είναι μεγάλη και το ασαφές σύνολο πλατύ (τι σημαίνει αυτό με όρους ασάφειας?). Μια άλλη φιλοσοφία είναι να τοποθετήσουμε τη συνάρτηση συμμετοχής στο μέσο όρο της κατανομής και το πλάτος να το καθορίσουμε έτσι ώστε όλα τα σημεία που ανήκουν στην αντίστοιχη ομάδα να έχουν βαθμό συμμετοχής μεγαλύτερο από ένα όριο π.χ. 0.5. αυτό σημαίνει ότι τα σημεία πού ανήκουν στο αντίστοιχο ασαφές σύνολο όταν ασαφοποιηθούν θα περιγράφονται από αυτό με βαθμό βεβαιότητας τουλάχιστον 0.5. Με βάση τα παραπάνω τοποθετούμε τις συναρτήσεις συμμετοχής σε κάθε άξονα και διαμερίζουμε τις μεταβλητές εισόδου και εξόδου. Ο αναγνώστης καλείται να κάνει τους υπολογισμούς για τις τιμές που αναφέρονται στον πίνακα (Πίνακας 2-1 σελ. 19). Στη συνέχεια ακολουθεί η κατασκευή των κανόνων επιλέγοντας τους αντίστοιχους λεκτικούς όρους από κάθε μεταβλητή. Η διαδικασία αυτή αναφέρεται στη βιβλιογραφία και ως ‘*απόδοση συντεταγμένων στους κανόνες*’ (*rule coordination*). Δημιουργούνται οι εξής κανόνες με βάση την Εικόνα 2-18:

$R^1 : if\ x_1\ is\ A_1^1\ and\ x_2\ is\ A_1^2\ then\ y\ is\ B_1$

$R^2 : if\ x_1\ is\ A_2^1\ and\ x_2\ is\ A_1^2\ then\ y\ is\ B_2$

Ο αναγνώστης καλείται να παραστήσει γραφικά τους προηγούμενους κανόνες όπως στην Εικόνα 2-11 και να υπολογίσει τις τιμές εξόδου για κάθε τιμή εξόδου ακολουθώντας τη διαδικασία του ασαφούς συμπερασμού που παρουσιάζεται στην Εικόνα 2-12 σελ. 16.

Όταν έχουμε δύο εισόδους και δεδομένα τα οποία είναι ομαδοποιημένα σε καθαρές ομάδες τότε τα πράγματα είναι εύκολα. Στην πράξη όμως αυτό συμβαίνει πολύ σπάνια. Συνήθως έχουμε να αντιμετωπίσουμε προβλήματα με 10 η και 50 εισόδους όπου δεν υπάρχει εποπτεία και τα δεδομένα δεν είναι ομαδοποιημένα σε καθαρά *clusters*. Για το λόγο αυτό έχει αναπτυχθεί πληθώρα μεθόδων ομαδοποίησης με κυριότερους εκπρόσωπου τις μεθόδους C-means, fuzzy

C—means, k- nearest clustering, subtractive clustering κ.τ.λ. το βασικό μειονέκτημα αυτής της φιλοσοφίας είναι ότι ομαδοποιούν δεδομένα στην είσοδο τα οποία είναι γειτονικά μεταξύ τους, χωρίς να τους ενδιαφέρει τι γίνεται στην έξοδο. Για το λόγο αυτό πολλές φορές οδηγούν σε κανόνες που δεν είναι αντιπροσωπευτικοί..

2.3.3 Εκπαίδευση του συστήματος

Το ασαφές σύστημα που προκύπτει από τη διαδικασία που είδαμε στα προηγούμενα παρέχουν ένα αρχικό ασαφές μοντέλο για το οποίο έχουν καθοριστεί ο αριθμός των κανόνων και η αρχική τους θέση. Αν δημιουργήσουμε το ασαφές σύστημα με τον τρόπο αυτό και εισάγουμε τις τιμές εισόδου των προδιαγραφών του πίνακα (Πίνακας 2-1 σελ. 19), θα παρατηρήσουμε ότι μας δίνουν αποκλίσεις από τις επιθυμητές τιμές που μπορεί να είναι και μη αποδεκτές. Αν μετακινήσουμε λίγο τα ασαφή σύνολα στην είσοδο και την έξοδο θα δούμε ότι η κατάσταση μπορεί να βελτιωθεί. Η τελευταία φάση λοιπόν στο σχεδιασμό του συστήματος είναι ο ακριβής προσδιορισμός των παραμέτρων του, δηλαδή των κέντρων και των αποκλίσεων των ασαφών συνόλων ώστε να προσαρμοστεί όσο το δυνατόν ποιο πιστά στις δοθείσες προδιαγραφές. Η προσαρμογή αυτή που ονομάζεται εκπαίδευση του συστήματος πραγματοποιείται με μεθόδους μη γραμμικής βελτιστοποίησης. Εξέχουσα θέση κατέχουν οι γενετικοί αλγόριθμοι οι οποίοι αρχικά χρησιμοποιήθηκαν ως αλγόριθμοι προσδιορισμού, των παραμέτρων προκαθορισμένων δομών ασαφών συστημάτων. Στις μεθόδους αυτές αναγνωρίστηκε η ικανότητα των γενετικών αλγορίθμων σε σχέση με άλλες μεθόδους βελτιστοποίησης (π.χ. μεθόδους κλίσης όπως back-propagation) να απεγκλωβίζονται από τοπικά βέλτιστα. Το γεγονός αυτό οφείλεται στο ότι οι γενετικοί κατανέμουν τη διαδικασία της διερεύνησης σε όλο το χώρο λύσεων διερευνώντας παράλληλα και ανεξάρτητα διαφορετικές περιοχές. Η ιδιότητα τους αυτή βέβαια είναι ευλογία και κατάρα ταυτόχρονα όταν χρησιμοποιούνται για εκπαίδευση, διότι ενώ εντοπίζουν σχετικά γρήγορα την περιοχή που βρίσκεται η βέλτιστη λύση δυσκολεύονται να την προσδιορίσουν ακριβώς αφού η διερεύνηση είναι κατανεμημένη και σε άλλες περιοχές του χώρου λύσεων. Για το σκοπό αυτό οι γενετικοί εφοδιάζονται με ειδικούς τελεστές που συγκεντρώνουν η ανακατανέμουν δυναμικά τη διαδικασία της διερεύνησης κατά τη διάρκεια της εξελικτικής διαδικασίας με στόχο την επιτάχυνση της εύρεσης λύσης. Ιδιαίτερη σημασία κατά τη διάρκεια της εκπαίδευσης έχει το φαινόμενο της *υπερεκπαίδευσης* (*overtrainning*). Όταν ένα σύστημα υπερεκπαδευτεί τότε προσεγγίζει ακριβώς τις δεδομένες προδιαγραφές για τις οποίες εκπαιδεύεται υστερεί όμως σημαντικά στις ικανότητες γενίκευσης (generalization). Αν δηλαδή στην είσοδο του έρθει μια τιμή που δεν ανήκει στα δεδομένα εκπαίδευσης το σύστημα θα δώσει εντελώς λανθασμένη έξοδο. Όταν ένα σύστημα υπερκπαιδευτεί αποστηθίζει τα δεδομένα εκπαίδευσης και δεν μαθαίνει τη δυναμική που διέπει τα δεδομένα, συνεπώς λειτουργεί ως μνήμη και είναι άχρηστο. Το πρόβλημα αυτό αντιμετωπίζεται συνήθως διασπώντας το σύνολο εκπαίδευσης σε δύο υποσύνολα. Το σύστημα εκπαιδεύεται χρησιμοποιώντας τα δεδομένα του ενός συνόλου και η συμπεριφορά του ελέγχεται και στα δύο υποσύνολα. Ο στόχος είναι η προσαρμογή των παραμέτρων έτσι ώστε το σύστημα να προσεγγίζει ικανοποιητικά τα δεδομένα και των δύο συνόλων. Ένας εμπειρικός σχεδιαστικός κανόνας για την αποφυγή του φαινομένου είναι:

ο αριθμός των προσαρμοζόμενων παραμέτρων του συστήματος πρέπει να είναι το πολύ ο μισός από τον αριθμό των δεδομένων.

3 Το περιβάλλον προγραμματισμού matlab

Το *MATLAB®* είναι μια υψηλής απόδοσης γλώσσα η οποία συνοδεύεται από ένα εύχρηστο προγραμματιστικό περιβάλλον για προγραμματισμό, τεχνικούς υπολογισμούς και οπτικοποίηση (*visualization*) των δεδομένων. Η βασική έννοια στο περιβάλλον MATLAB® είναι ο πίνακας (*matrix*). Μάλιστα το όνομα MATLAB® προέρχεται από τα αρχικά ***Mat**rix **Lab**oratory*. Οι πίνακες (*matrices*), τα διανύσματα (*vectors*) που είναι μονοδιάστατοι πίνακες και οι μεταβλητές, δηλώνονται και δημιουργούνται αυτόματα με τη χρήση τους χωρίς περιττές δηλώσεις που αφορούν τον τύπο το μέγεθος και τις διαστάσεις (dimensioning προκειμένου για πίνακες.). το γεγονός αυτό παρέχει μεγάλη ευχρηστία και ταχύτητα στον προγραμματισμό ιδιαίτερα στην επίλυση προβλημάτων που μορφοποιούνται υπό μορφή πινάκων. Επίσης, είναι εφοδιασμένο με πληθώρα συναρτήσεων π.χ. εύρεση ορίζουσας, υπολογισμό αντίστροφου, διαχείριση υποπινάκων χωρίς να είναι απαραίτητη η δημιουργία πολύπλοκων συναρτήσεων, δέσμευση και αποδέσμευση μνήμης (γίνεται αυτόματα). Επίσης, παρέχει ισχυρές δυνατότητες οπτικοποίησης των αποτελεσμάτων με ταχύτητα και ευκολία. Το μειονέκτημα που έχει είναι ότι οι εντολές που δίνουμε δεν μπορούν να δημιουργήσουν αυτόνομη εφαρμογή (*standalone executable*), αλλά χρειάζονται το περιβάλλον για να εκτελεστούν. Το *MATLAB®* δηλαδή είναι ένας αλληλεπιδραστικός διερμηνευτής εντολών (*Interactive command interpreter*) και με αυτή τη φιλοσοφία πρέπει να χρησιμοποιείται. Βέβαια πρόσφατες εκδόσεις του MATLAB® έχουν ενσωματώσει το MATLAB Application Program Interface (*M-API*) που παρέχει τη δυνατότητα να κληθούν οι συναρτήσεις του από γλώσσες προγραμματισμού όπως η *C* και η *FORTRAN*, υπό μορφή κλήσεων DLL (Dynamic Link Library).

3.1 Γραμμή εντολών και βασικές πράξεις

3.1.1 Απλοί αριθμητικοί υπολογισμοί

Με την ενεργοποίηση του περιβάλλοντος εμφανίζεται η γραμμή εντολών περιμένοντας τις εντολές μας:

```
Using Toolbox Path Cache.  Type "help toolbox_path_cache" for more info.
 To get started, select "MATLAB Help" from the Help menu.
>>
```

έστω ότι θέλουμε να ορίσουμε δύο μεταβλητές x, y και το άθροισμα να το αποθηκεύσουμε στην μεταβλητή z. Αυτό πραγματοποιείται με τις ακόλουθες εντολές.

```
>> x=2
x =

     2
>> y=4
y =
     4
>> z=x+y
z =
     6
>>
```

Με την εντολή *whos* μπορούμε να δούμε το *workspace* δηλαδή το χώρο που αποθηκεύονται οι μεταβλητές. Μπορούμε να δούμε ποιες μεταβλητές είναι αποθηκευμένες τον τύπο τους και το μέγεθος τους:

```
>> whos
  Name      Size                    Bytes  Class

  x         1x1                         8  double array
  y         1x1                         8  double array
  z         1x1                         8  double array

Grand total is 3 elements using 24 bytes
>>
```

Αυτό σημαίνει ότι στο χώρο εργασίας έχουν αποθηκευτεί οι μεταβλητές x,y,z. Αν θέλουμε να εμφανίσουμε μια μεταβλητή π.χ. τη μεταβλητή y, γράφουμε y στη γραμμή εντολών και πιέζουμε *enter*.

```
>> y
y =
     4
>>
```

3.1.2 υπολογισμοί με διανύσματα

Ένα διάνυσμα, δηλαδή ένας μονοδιάστατος πίνακας μπορεί να οριστεί με διάφορους τρόπους. Ο ποιο εύκολος είναι με τη βοήθεια του τελεστή : που καθορίζει το βήμα μεταξύ της μέγιστης και της ελάχιστης τιμής. Έστω ότι θέλουμε να δημιουργήσουμε ένα διάνυσμα από –1 έως 2 με βήμα 0.5 αυτό γίνεται με την ακόλουθη εντολή:

```
>> a=-1:0.5:2
a =
   -1.0000   -0.5000         0    0.5000    1.0000    1.5000    2.0000
>>
```

Διαφορετικά θα μπορούσαμε να δώσουμε τις τιμές σε αγκύλες διαχωρισμένες μεταξύ τους με κενό:

```
b=[-1 -0.5 0 0.5 1 1.5 2]
b =
   -1.0000   -0.5000         0    0.5000    1.0000    1.5000    2.0000
>>
```

μπορούμε να προσθέσουμε τα διανύσματα και να πάρουμε το άθροισμα σε μια νέα μεταβλητή με μια απλή πρόσθεση:

```
>> c=a+b
c =
    -2    -1     0     1     2     3     4
>>
```

μπορούμε να πάρουμε τον ανάστροφο ενός διανύσματος με τον τελεστή ΄ π.χ:

```
>> can=c'
can =
    -2
    -1
     0
     1
     2
     3
     4
>>
```

Η ίδια διαδικασία σε C η άλλη γλώσσα προγραμματισμού δεν είναι και τόσο απλή. Ένα άλλο σημαντικό στοιχείο είναι η ευκολία με την οποία μπορούμε να χειριστούμε υποπίνακες η υποδιανύσματα. Έστω ότι θέλουμε να πάρουμε από το τρίτο μέχρι το πέμπτο στοιχείο του διανύσματος a και να βρούμε το ανάστροφο του διανύσματος που προκύπτει. Αυτό γίνεται επίσης σε μία εντολή ως εξής:

```
>> e=a(3:1:5)'
e =
         0
    0.5000
    1.0000
>>
```

Προσέξτε τις παρενθέσεις αντί για τις αγκύλες. Το 3:1:5 σημαίνει από το στοιχείο 3 με βήμα 1 έως το 5. Ο τόνος μας δίνει το ανάστροφο διάνυσμα του αποτελέσματος. Έστω τώρα ότι θέλουμε να δούμε ποιες μεταβλητές έχουμε αποθηκεύσει κατά τα γνωστά *whos*

```
>> whos
  Name      Size                   Bytes  Class
  a         1x7                       56  double array
  b         1x7                       56  double array
  c         1x7                       56  double array
  can       7x1                       56  double array
  e         3x1                       24  double array
  x         1x1                        8  double array
  y         1x1                        8  double array
  z         1x1                        8  double array
Grand total is 34 elements using 272 bytes
>>
```

Ας υποθέσουμε ότι θέλουμε να δούμε τα περιεχόμενα του διανύσματος a. Γράφουμε:

```
>> A
??? Undefined function or variable 'A'.
>>
```

Γιατί? Διότι ***το*** MATLAB® ***είναι case sensitive*** και η μεταβλητή a είναι διαφορετική μεταβλητή από την A.

Ας υποθέσουμε ότι θέλουμε να προσθέσουμε το a με το e γράφουμε:

```
>> a+e
??? Error using ==> +
Matrix dimensions must agree.
>>
```

Γιατί? Είπαμε ότι το MATLAB® κάνει τα πάντα με σεβασμό όμως στις μαθηματικές παραδόσεις. Και επειδή υπάρχει μια τάση να αγνοούμε τα μαθηματικά καλό είναι να εξοικειωθούμε με το παραπάνω μήνυμα σφάλματος γιατί θα το δούμε πολλές φορές.

Και κάτι ακόμα: έστω ότι γράφουμε a=15. Τι θα συμβεί? Το a ήταν ένα διάνυσμα 1Χ7 τώρα θα διαγραφεί και στο a θα είναι πλέον αποθηκευμένος ο αριθμός 15. Προσοχή λοιπόν γιατί δεν θα υπάρξει προειδοποίηση. Την εντολή *whos* καλό είναι να τη χρησιμοποιούμε συχνά για να βλέπουμε τι γίνεται (διαστάσεις διανυσμάτων, τύποι μεταβλητών κ.τ.λ.). Είναι εντελώς δωρεάν και πολύ χρήσιμη.

Για δισδιάστατους πίνακες (*matrixes*) τα πράγματα είναι ανάλογα:

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

Η διαφορετικά:

```
>> B=[ [1 3 2]' [2 3 1]' [5 6 7]' ]
B =
     1     2     5
     3     3     6
     2     1     7
>>
```

οι γραμμές διαχωρίζονται με «;» και τα στοιχεία κάθε γραμμής με κενό όπως στα μονοδιάστατα διανύσματα. Εύκολα μπορούμε να βρούμε τον ανάστροφο του πίνακα A:

```
>> B=A'
B =
     1     4     7
     2     5     8
     3     6     9
>>
```

Μπορούμε να πολλαπλασιάσουμε πίνακες μεταξύ τους, να βρούμε την ορίζουσα ενός πίνακα και τον αντίστροφό του:

```
>> B=A'
B =
     1     4     7
     2     5     8
     3     6     9
>> C=A*B
C =
    13    11    38
    31    29    92
    49    47   146
>> d=det(C)
d =
     0
>> C=inv(A)
Warning: Matrix is close to singular or badly scaled.
         Results may be inaccurate. RCOND = 1.541976e-018.
C =
  1.0e+016 *
   -0.4504    0.9007   -0.4504
    0.9007   -1.8014    0.9007
   -0.4504    0.9007   -0.4504
>>
```

Εδώ η ορίζουσα του πίνακα C είναι d=0 και επομένως ο αντίστροφος δεν υπάρχει. Το MATLAB® υπολογίζει προσεγγιστικά τον αντίστροφο που φυσικά ο αλγόριθμος δεν συγκλίνει αφού ο αντίστροφος δεν υπάρχει και απαντά με το μήνυμα:

```
Warning: Matrix is close to singular or badly scaled.
         Results may be inaccurate. RCOND = 1.541976e-018.
```

Το αποτέλεσμα που δίνει δεν είναι ορθό και θα πρέπει να αγνοηθεί. Αν ο πίνακας έχει αντίστροφο π.χ. ο πίνακας B τότε όλα είναι φυσιολογικά:

```
>> det(B)
ans =
   -18
>> inv(B)
ans =
   -0.8333    0.5000    0.1667
    0.5000    0.1667   -0.5000
    0.1667   -0.1667    0.1667
>>
```

Ένα στοιχείο του πίνακα B μπορεί να ληφθεί με B(i,j) π.χ:

```
>> B(2,3)
ans =
     6
>>
```

και ένας υποπίνακας ως B(from_row : step_row : to_row, from_col : step_col : to_col) π.χ.

```
>> B(1:2:3, 2:1:3)
ans =
     2     5
     1     7
>>
```

Εδώ μπορούμε να παρατηρήσουμε και κάτι ακόμα: Αν δεν αποθηκεύσουμε κάποιο αποτέλεσμα σε μια δικιά μας μεταβλητή το MATLAB® αποθηκεύει το αποτέλεσμα σε μια μεταβλητή με το όνομα *ans* η οποία κρατάει πάντα το τελευταίο *ορφανό* αποτέλεσμα. Αν θέλουμε να διαγράψουμε όλες τις μεταβλητές από το χώρο εργασίας χρησιμοποιούμε την εντολή *clear*. Προσοχή όμως μετά τη χρήση της εντολής *clear* οι μεταβλητές και τα δεδομένα δεν είναι πλέον διαθέσιμα αυτό φαίνεται και με την *whos* που δεν μας επιστρέφει πλέον τίποτα:

```
>> clear
>> whos
>>
```

3.2 Γραφικές παραστάσεις

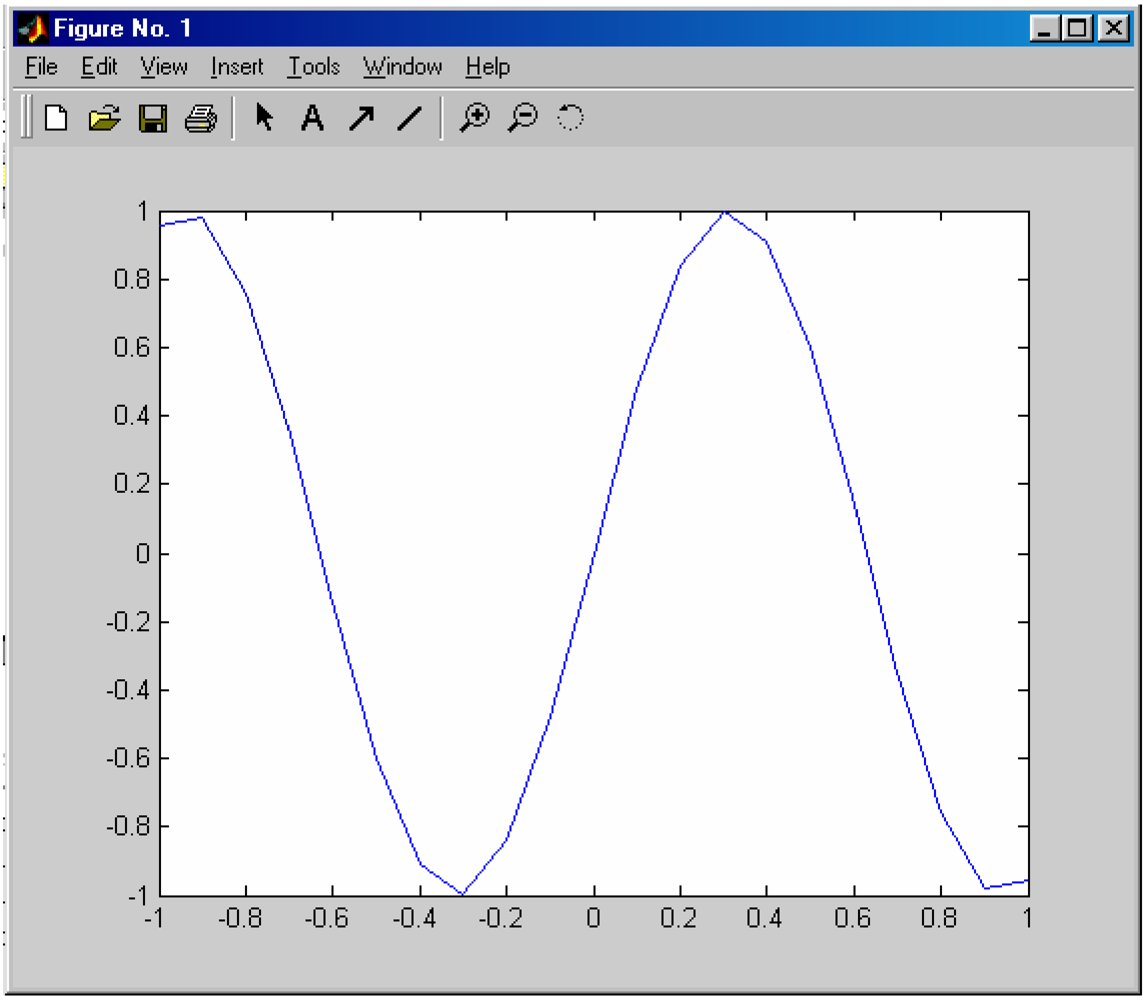
3.2.1 γραφικά στις δύο διαστάσεις

Το MATLAB® δίνει την ευκολία εύκολων γραφικών παραστάσεων των δεδομένων. Έστω ότι θέλουμε να δημιουργήσουμε τη γραφική παράσταση της συνάρτησης $y = \sin(5x) \big| x \in [-1,1]$. Πρώτα δημιουργούμε τις τιμές του x έπειτα υπολογίζουμε τις αντίστοιχες τιμές y βάση της προηγούμενης σχέσης και στη συνέχεια απεικονίζουμε τα δεδομένα με την εντολή *plot*.

```
>> x=-1:0.1:1;
>> y=sin(5*x);
>> whos
  Name      Size                   Bytes  Class

  x         1x21                     168  double array
  y         1x21                     168  double array
Grand total is 42 elements using 336 bytes
>> plot(x,y)
>>
```

Εδώ το *whos* δεν είναι απαραίτητο απλώς τέθηκε για να δούμε τα αποτελέσματα των δύο εντολών. Γιατί δεν εμφανίστηκαν στην οθόνη? Διότι στο τέλος κάθε εντολής τέθηκε το «;». Ο τελεστής αυτός όταν μπαίνει στο τέλος μιας εντολής αποτρέπει την εμφάνιση των αποτελεσμάτων της εντολής, στην οθόνη. Είναι πολύ χρήσιμος σε περιπτώσεις που έχουμε μεγάλα αποτελέσματα. Φυσικά το αποτέλεσμα δημιουργείται κανονικά όπως φαίνεται και με τη *whos*. Το αποτέλεσμα του plot φαίνεται στο παρακάτω σχήμα:



Εικόνα 3-1:Γραφική παράσταση με το MATLAB®

Με την εντολή *help plot* μπορούμε να δούμε τις παραμέτρους και τη σύνταξη της και επιμέρους πληροφορίες που αφορούν την εντολή.

```
>> help plot
 PLOT   Linear plot.
    PLOT(X,Y) plots vector Y versus vector X. If X or Y is a matrix,
    then the vector is plotted versus the rows or columns of the matrix,
    whichever line up.  If X is a scalar and Y is a vector, length(Y)
    disconnected points are plotted.
     PLOT(Y) plots the columns of Y versus their index.
    If Y is complex, PLOT(Y) is equivalent to PLOT(real(Y),imag(Y)).
    In all other uses of PLOT, the imaginary part is ignored.
     Various line types, plot symbols and colors may be obtained with
    PLOT(X,Y,S) where S is a character string made from one element
    from any or all the following 3 columns:
           b     blue          .     point              -     solid
           g     green         o     circle             :     dotted
           r     red           x     x-mark             -.    dashdot
           c     cyan          +     plus               --    dashed
           m     magenta       *     star
           y     yellow        s     square
           k     black         d     diamond
                               v     triangle (down)
                               ^     triangle (up)
                               <     triangle (left)
                               >     triangle (right)
                               p     pentagram
                               h     hexagram
    For example, PLOT(X,Y,'c+:') plots a cyan dotted line with a plus
    at each data point; PLOT(X,Y,'bd') plots blue diamond at each data
    point but does not draw any line.
    PLOT(X1,Y1,S1,X2,Y2,S2,X3,Y3,S3,...) combines the plots defined by
    the (X,Y,S) triples, where the X's and Y's are vectors or matrices
    and the S's are strings.
    For example, PLOT(X,Y,'y-',X,Y,'go') plots the data twice, with a
    solid yellow line interpolating green circles at the data points.
    The PLOT command, if no color is specified, makes automatic use of
    the colors specified by the axes ColorOrder property.  The default
    ColorOrder is listed in the table above for color systems where the
    default is blue for one line, and for multiple lines, to cycle
    through the first six colors in the table.  For monochrome systems,
    PLOT cycles over the axes LineStyleOrder property.
    PLOT returns a column vector of handles to LINE objects, one
    handle per line.
    The X,Y pairs, or X,Y,S triples, can be followed by
    parameter/value pairs to specify additional properties
    of the lines.
    See also SEMILOGX, SEMILOGY, LOGLOG, PLOTYY, GRID, CLF, CLC, TITLE,
    XLABEL, YLABEL, AXIS, AXES, HOLD, COLORDEF, LEGEND, SUBPLOT, STEM.
>>
```

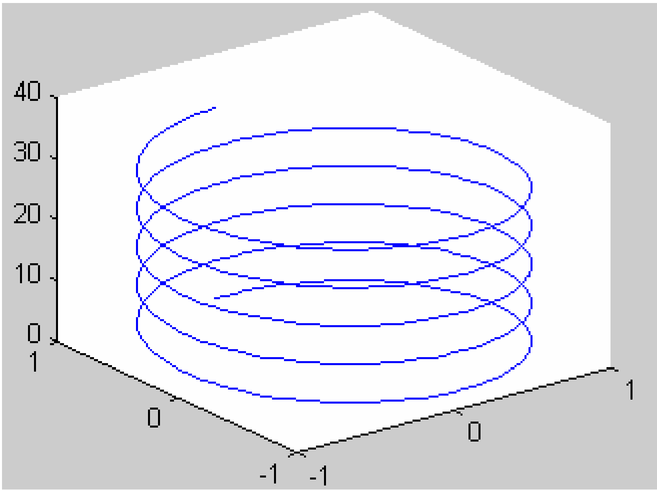
Το ίδιο συμβαίνει με όλες τις εντολές που χρησιμοποιούμε. Π.χ. *help inv* μας δίνει πληροφορίες για την εντολή που υπολογίζει τον αντίστροφο ενός πίνακα. Ο αναγνώστης καλείται να πειραματιστεί με τις επιμέρους παραμέτρους της plot να θέσει ετικέτες στους άξονες και να αλλάξει χρώμα- στυλ στη γραφική παράσταση.

3.2.2 Γραφικές παραστάσεις στις τρεις διαστάσεις

Το MATLAB® παρέχει τη δυνατότητα τρισδιάστατων γραφικών παραστάσεων που απεικονίζουν επιφάνειες. Η απεικόνιση τρισδιάστατων παραστάσεων είναι εννοιολογικά περισσότερο περίπλοκη από αυτήν στις δύο διαστάσεις και απαιτεί γνώσεις γραφικών Η/Υ (computer graphics) που αφορούν στο χώρο παρατήρησης σε φωτορεαλλισμό (rendering) κ.τ.λ. Απλές γραφικές παραστάσεις γραμμών στο χώρο είναι δυνατό να πραγματοποιηθούν με χρήση της εντολής plot3(), που είναι η αντίστοιχη της plot στις δύο διαστάσεις. Η παράσταση μιας έλικας στις τρείς διαστάσεις μπορεί να γίνει με τον ακόλουθο κώδικα:

```
>>  t = 0:pi/50:10*pi;
       plot3(sin(t),cos(t),t);
 >>
```

το αποτέλεσμα απεικονίζεται στην Εικόνα 3-2.



Εικόνα 3-2:Γραφική παράσταση έλικας στις τρεις διαστάσεις

Η απεικόνιση περισσότερο πολύπλοκων επιφανειών είναι ποιο σύνθετη και έξω από το στόχο των σημειώσεων. Ο αναγνώστης που επιθυμεί περισσότερα σχετικά με την απεικόνιση στις τρεις διαστάσεις μπορεί να συμβουλευτεί το εγχειρίδιο του *MATLAB®* και βιβλιογραφία σχετικά με γραφικά Η/Υ για τις βασικές έννοιες.

3.3 Εντολές ελέγχου της ροής του προγράμματος

Το MATLAB® μας δίνει τη δυνατότητα να γράψουμε δομημένο κώδικα ελέγχοντας τη ροή του προγράμματος με εντολές if-else και βρόγχους επανάληψης for, while κ.τ.λ.

3.3.1 Η εντολή ελέγχου for

```
>> for i = 1:2:10,
i^2
end
ans =
     1
ans =
     9
ans =
    25
ans =
    49
ans =
    81
>>
```

Παραπάνω εμφανίζεται ένα παράδειγμα σύνταξης της εντολής for η σύνταξη είναι

```
for μεταβλητήεπαναληψης=from_value:step:to_value
        command-1
        command-2
        .
        .
        .
        command-n
end
```

Η μεταβλητή επανάληψης έχει αρχικά την τιμή from_value την οποία σε κάθε επανάληψη αυξάνει κατά step. Σε κάθε επανάληψη εκτελούνται οι εντολές που υπάρχουν μέχρι το end. Οι επαναλήψεις συνεχίζονται μέχρι η τιμή της μεταβλητής επανάληψης να γίνει to_value. Μπορούμε να έχουμε και for μέσα σε for όπως γνωρίζουμε και από τις γλώσσες προγραμματισμού κάποια δηλαδή από τις εντολές command-ι. μπορεί να είναι ένα block εντολών αντί για μια απλή εντολή.

3.3.2 Η εντολή while

Κατά παρόμοιο τρόπο έχουμε και την εντολή while. Όπου οι εντολές που βρίσκονται μεταξύ της συνθήκης και του end εκτελούνται μέχρι η συνθήκη ελέγχου να γίνει ψευδής. Υπενθυμίζεται ότι πρέπει να προσέχουμε ώστε η συνθήκη να αλλάζει από το σώμα της while και κάποτε να γίνει ψευδής διαφορετικά το πρόγραμμα θα εγκλωβιστεί σε ένα ατέρμονο βρόγχο.

```
while sinthiki
command-1
command-2
.
.
.
command-n
end
```

3.3.3 Η εντολή switch

Η σύνταξη της έχει ως ακολούθως

```
switch switch_expr
        case case_expr,
          statement, ..., statement
        case {case_expr1, case_expr2, case_expr3,...}
          statement, ..., statement
       ...
        otherwise,
          statement, ..., statement
      END
```

3.3.4 Η εντολή ελέγχου if και else-if

Η εντολή ελέγχου if

```
Example
       if I == J
         A(I,J) = 2;
       elseif abs(I-J) == 1
         A(I,J) = -1;
       else
         A(I,J) = 0;
       end
```

3.4 Υπολογισμοί με σύμβολα (symbolic toolbox)

Το MATLAB® μας δίνει τη δυνατότητα να επιτελούμε υπολογισμούς με σύμβολα αντί για αριθμούς. Π.χ έστω ότι θέλουμε να υπολογίσουμε το αόριστο ολοκλήρωμα της συνάρτησης $f(x)=\dfrac{1}{1+x^4}$. θα πρέπει να γράψουμε τον ακόλουθο κώδικα.

```
>> x=sym('x')  %Δηλώνουμε το χ ως μεταβλητή σύμβολο.

x =

x

>> int(1/(1+x^4)) %υπολογίζει το αόριστο ολοκλήρωμα.

ans =

1/8*2^(1/2)*log((x^2+x*2^(1/2)+1)/(x^2-
x*2^(1/2)+1))+1/4*2^(1/2)*atan(x*2^(1/2)+1)+1/4*2^(1/2)*atan(x*2^(1/2)-1)

>> pretty(ans) %εμφανίζει τα αποτελέσματα σε 'κομψή μορφή'

                2      1/2
       1/2     x  + x 2    + 1         1/2         1/2
  1/8 2    log(---------------) + 1/4 2    atan(x 2    + 1)
                2      1/2
               x  - x 2    + 1

                1/2         1/2
         + 1/4 2    atan(x 2    - 1)
>>
```

όμοια αν θέλουμε την παράγωγο θα πρέπει να γράψουμε :

```
>> diff(1/(1+x^4))
 ans =
 -4/(1+x^4)^2*x^3
 >> pretty(ans)
                                        3
                                      x
                                -4 ---------
                                         4 2
                                   (1 + x )
>>
```

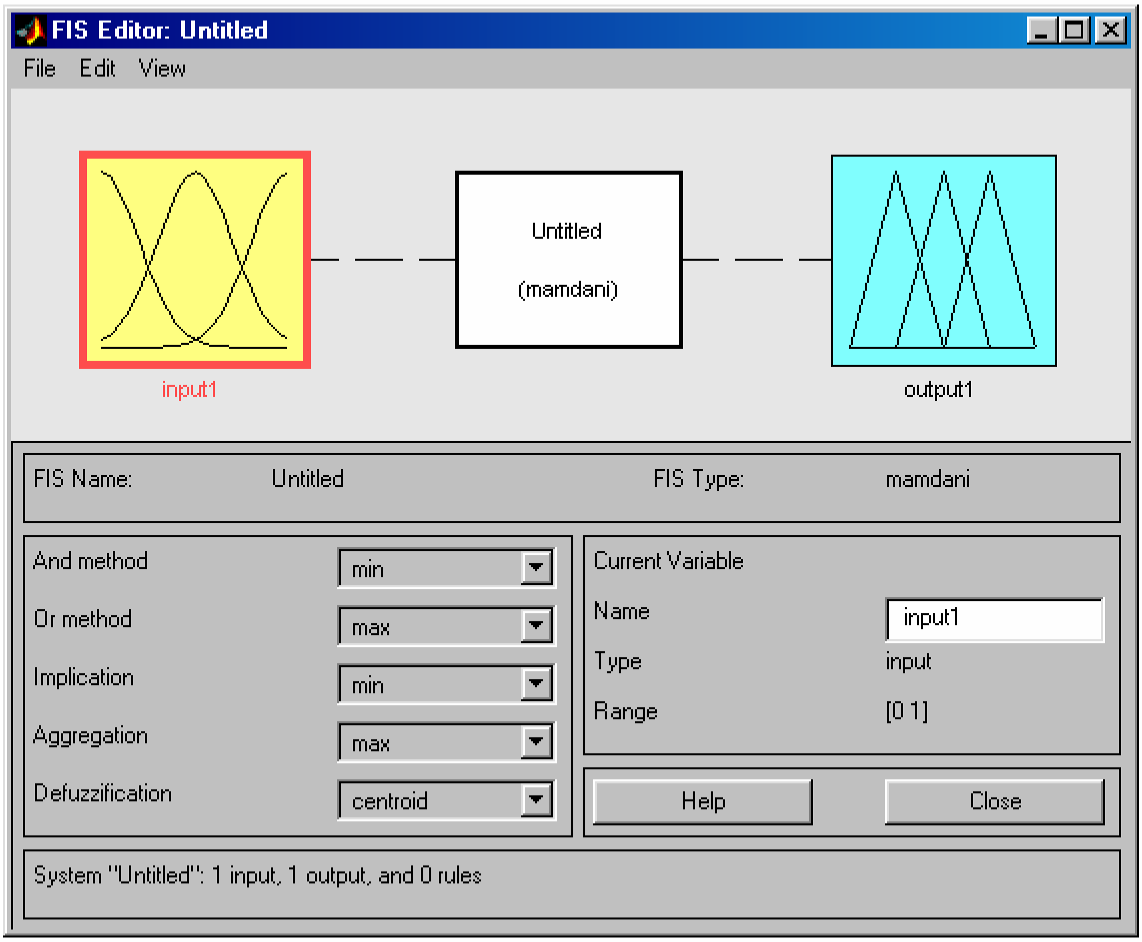
με την εντολή *help symbolic* και τη βοήθεια του πακέτου μπορούν να βρεθούν περισσότερες λεπτομέρειες για τις δεκάδες εντολών συμβολικών υπολογισμών.

3.5 Η ασαφής Εργαλειοθήκη (fuzzy toolbox)

Εκτός από τη μεγάλη ευκολία που μας παρέχει σχετικά με επεξεργασία και απεικόνιση δεδομένων το *MATLAB®* είναι εφοδιασμένο και με ένα σύνολο από εργαλειοθήκες (*toolboxes*) που είναι ολοκληρωμένα περιβάλλοντα και βιβλιοθήκες που μας δίνουν τεράστια ευκολία σχεδιασμού και ανάλυσης σε εξειδικευμένα πεδία της επιστήμης. Στις σημειώσεις αυτές θα ασχοληθούμε με την εργαλειοθήκη ασαφούς λογικής (*fuzzy toolbox*) που μας δίνει τη δυνατότητα σχεδιασμού ασαφών συστημάτων. Η εργαλειοθήκη της ασαφούς λογικής ενεργοποιείται με την εντολή:

```
>> fuzzy
>>
```

Το MATLAB® ανταποκρίνεται ενεργοποιώντας τον συντάκτη συστημάτων ασαφούς συμπερασμού (*FIS **F**uzzy **I**nference **S**ystem **E**ditor*) όπως απεικονίζεται στην Εικόνα 3-3, ο οποίος μας επιτρέπει να σχεδιάσουμε πλήρως ένα ασαφές σύστημα. Ο προκαθορισμένος τύπος ασαφούς συστήματος είναι ο τύπος mamdani (δες §2.2 σελ.10.). μπορούμε όμως να καθορίσουμε και ασαφές σύστημα τύπου TSK, από το *menu\file\new fis*.



Εικόνα 3-3: Ο συντάκτης ασαφούς συμπερασμού FIS.

Καθορίζοντας ένα νέο ασαφές σύστημα μπορούμε να καθορίσουμε τον αριθμό και το εύρος των μεταβλητών εισόδου το διαμερισμό τους, τον τύπο του ασαφούς συμπερασμού τους ασαφείς κανόνες και γενικά όλα τα βήματα σχεδιασμού που περιγράφονται στην §2.2. (δες και Άσκηση 4.2 σελ. 39).

3.5.1 Το διάγραμμα

Το διάγραμμα (Εικ. 3-3) δείχνει τις εισόδους, εξόδους και ένα κεντρικό επεξεργαστή ασαφών κανόνων. Εάν επιλέξουμε με το ποντίκι ένα από τα κουτιά μεταβλητές, τότε αυτό θα γίνει η τρέχουσα μεταβλητή. Το κουτί θα πρέπει να τονιστεί με κόκκινο χρώμα. Διπλή επιλογή με το ποντίκι (double-click) σε κάποια από τις μεταβλητές θα πρέπει να εμφανίσει το παράθυρο του συντάκτη συναρτήσεων συμμετοχής (*Membership Function Editor*). Διπλή επιλογή με το ποντίκι (double-click) στον επεξεργαστή ασαφών κανόνων θα εμφανίσει τον συντάκτη κανόνων

(*Rule Editor*). Εάν μία μεταβλητή υπάρχει αλλά δεν αναφέρεται στη βάση κανόνων, τότε συνδέεται με τον επεξεργαστή κανόνων με διακεκομμένη γραμμή.

3.5.2 Επιλογές του Menu

Οι επιλογές της μπάρας εργαλείων του συντάκτη συστημάτων ασαφούς συμπερασμού (*FIS Editor*) μας επιτρέπυν να ανοίξουμε σχετικά GUI εργαλεία, να ανοίξουμε και να αποθηκεύσουμε συστήματα κλπ.

- Κάτω από το **File** επιλέξτε

  New Mamdani FIS... για να ανοίξουμε ένα νέο σύστημα Mamdani-style χωρίς μεταβλητές και χωρίς κανόνες, το οποίο ονομάζεται `Untitled`.

  New Sugeno FIS... για να ανοίξουμε ένα νέο σύστημα Sugeno-style χωρίς μεταβλητές και χωρίς κανόνες, το οποίο ονομάζεται `Untitled`.

  Open from disk... για να ανοίξουμε ένα σύστημα από ένα συγκεκριμένο `.fis` αρχείο του δίσκου.

  Save to disk για να αποθηκεύσουμε το τρέχον σύστημα σε ένα `.fis` αρχείο στο δίσκο.

  Save to disk as... για να αποθηκεύσουμε το τρέχον σύστημα στο δίσκο με την δυνατότητα να το μετονομάσετε ή να του αλλάξετε θέση.

  Open from workspace... για να ανοίξουμε ένα σύστημα από μία συγκεκριμένη μεταβλητή (FIS δομή) του χώρου εργασίας (workspace).

  Save to workspace... για να αποθηκεύσουμε το σύστημα στην τρέχουσα μεταβλητή (FIS δομή) του χώρου εργασίας (workspace).

  Save to workspace as... για να αποθηκεύσουμε το σύστημα σε μία συγκεκριμένη μεταβλητή (FIS δομή) του χώρου εργασίας (workspace).

  Close window για να κλείσουμε το παράθυρο.

- Κάτω από το **Edit** επιλέξτε

  Add input για να προσθέσουμε μία ακόμη είσοδο στο σύστημα.

  Add output για να προσθέσουμε μία ακόμη έξοδο στο σύστημα.

  Remove variable για να διαγράψουμε μία συγκεκριμένη μεταβλητή.

  Undo για να ματαιώσουμε την πιο πρόσφατη μεταβολή.

- Κάτω από το **View** επιλέξτε

  Edit MFs... για να καλέσουμε τον συντάκτη συναρτήσεων συμμετοχής (*Membership Function Editor*).

  Edit rules... για να καλέσουμε τον συντάκτη κανόνων (*Rule Editor*).

  Edit anfis... για να καλέσουμε τον συντάκτη ANFIS (*ANFIS Editor*) για συστήματα Sugeno με μία μόνο έξοδο.

View rules... για να καλέσουμε την επισκόπηση των κανόνων (*Rule Viewer*).

View surface... για να καλέσουμε την επισκόπηση επιφανειών (*Surface Viewer*).

3.5.3 Επιλογές Μεθόδων Συνεπαγωγής

Το MATLAB® παρέχει πέντε pop-up menus για αλλάξουμε την λειτουργία των πέντε βασικών βημάτων στην διαδικασία ασαφούς συνεπαγωγής (*fuzzy implication process*)

- **And method**: Επιλέγουμε `min`, `prod`, ή `Custom` για μία μέθοδο σύμφωνα με τις οδηγίες μας.
- **Or method**: Επιλέγουμε `max`, `probor` (probabilistic or), or `Custom` για μία μέθοδο σύμφωνα με τις οδηγίες μας.
- **Implication method**: Επιλέγουμε `min`, `prod`, or `Custom` για μία μέθοδο σύμφωνα με τις οδηγίες μας. Αυτή η επιλογή δεν είναι διαθέσιμη για ασαφή συνεπαγωγή τύπου Sugeno.
- **Aggregation method**: Επιλέγουμε `max`, `sum`, `probor`, or `Custom` για μία μέθοδο σύμφωνα με τις οδηγίες μας. Αυτή η επιλογή δεν είναι διαθέσιμη για ασαφή συνεπαγωγή τύπου Sugeno.
- **Defuzzification method**: Για συνεπαγωγή τύπου Mamdani, επιλέγουμε `centroid`, `bisector`, `mom` (middle of maximum), `som` (smallest of maximum), `lom` (largest of maximum), or `Custom` για μία μέθοδο σύμφωνα με τις οδηγίες μας. Για συνεπαγωγή τύπου Sugeno, επιλέγουμε μεταξύ `wtaver` (weighted average) ή `wtsum` (weighted sum).

4 Εργαστηριακές Ασκήσεις

Στο κεφάλαιο αυτό περιγράφονται διάφορες ασκήσεις, τις οποίες ο αναγνώστης καλείται να φέρει σε πέρας προκειμένου να εξοικειωθεί και να αποσαφηνίσει τις βασικές έννοιες της ασαφούς λογικής.

4.1 Γνωριμία με το περιβάλλον προγραμματισμού MATLAB®

4.1.1 Επίλυση συστήματος εξισώσεων

Να επιλυθεί το παρακάτω σύστημα εξισώσεων της (4.18) με χρήση του προγράμματος MATLAB®.

$$\begin{aligned} 5x+2y+3z+4w&=7 \\ 2x+6y+1z+9w&=11 \\ 6x+3y+1z+5w&=5 \\ 2x+4y+7z+9w&=3 \end{aligned} \tag{4.18}$$

4.1.2 Διαχείριση αρχείων

Να δημιουργηθεί ένα αρχείο δεδομένων το οποίο να αποτελείται από 4 στήλες και 300 γραμμές. Κάθε γραμμή αποτελείται από τέσσερα στοιχεία της μορφής Πίνακας 4-1.

Α/Α	x_1	x_2	x_3	y
1	0.1	1.2	2.7	3.1
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
300	1.1	2.7	1.8	4.2

Πίνακας 4-1

Κάθε $x_i \,|\, i=1,2,3$ είναι ένας τυχαίος πραγματικός αριθμός στο διάστημα [0,3] και η έξοδος y δίνεται από τη σχέση:

$$y=[x_1^{-1.5}+\sin(5x_1)]^2+3\cos(x_2)+5x_3 \qquad 4.19$$

Στη συνέχεια να αποθηκευτούν σε ένα αρχείο με όνομα *data.dat*. Το αρχείο να ανοιχτεί με το *notepad* και να επιβεβαιωθεί η ορθότητα του.

Ακολούθως να διαβαστεί το αρχείο από το MATLAB® και για κάθε στήλη να υπολογιστεί η μέση τιμή η τυπική απόκλιση η μέγιστη και η ελάχιστη τιμή.

Τέλος να γίνει η γραφική παράσταση υπό μορφή scatter plot κάθε στήλης (άξονας χ) σε σχέση με την στήλη y. Οι τρείς γραφικές παραστάσεις να γίνουν σε ξεχωριστά διαγράμματα τα οποία όμως να φαίνονται ταυτόχρονα.

4.2 Σχεδιασμός Ασαφούς συστήματος με FIS.

Ορισμένες εξελιγμένες μάρκες αυτοκινήτων είναι εφοδιασμένες με το λεγόμενο σύστημα *αυτόματου πιλότου* το οποίο ρυθμίζει την επιτάχυνση του αυτοκινήτου ανάλογα με την κλίση του εδάφους [-40,40]% και το φορτίο του οχήματος [1000-1500]kg, ώστε η ταχύτητα του να διατηρείται σταθερή. Η επιτάχυνση του αυτοκινήτου λαμβάνει κανονικοποιημένες τιμές στο εύρος [-1,1]. Αρνητικές τιμές έχουν την έννοια της επιβράδυνσης ενώ θετικές τιμές την έννοια της επιτάχυνσης του αυτοκινήτου. Ενδεικτικές τιμές αναφέρονται στον παρακάτω πίνακα:

Κλίση εδάφους [-40,40]%	Βάρος Αυτοκινήτου [1000-1500] kg	Επιτάχυνση [-1,1]
-20	1100	-0.7
0	1200	0.1
10	1500	0.9
20	1200	0.7
30	1500	1.0
-15	1000	-0.5

Πίνακας 4-2: Αυτόματος Πιλότος.

Με βάση τις παραπάνω ενδεικτικές τιμές να σχεδιαστεί ασαφής ελεγκτής τύπου Mamdani, ο οποίος να ρυθμίζει την επιτάχυνση του αυτοκινήτου, με σφάλμα 5%. Ο σχεδιασμός του συστήματος να γίνει ακολουθώντας τη μεθοδολογία που δόθηκε στην §2.2 σελ. 10.

4.3 Κατάταξη δειγμάτων σε κατηγορίες Classification.

Σε ένα εργοστάσιο επεξεργασίας φρούτων χρησιμοποιείται ένας ασαφής εκλεκτής για την διαλογή των φρούτων σε ποιότητες. Η ποιότητα κάθε φρούτου καθορίζεται από το βάρος και την σκληρότητα του. Θεωρούμε ότι το βάρος των φρούτων κυμαίνεται από [10-100 γρ] και η σκληρότητα σε μια κλίμακα από [10-100]% ως προς την ιδανική σκληρότητα. Το αποτέλεσμα του ελεγκτή είναι η αυτόματη κατηγοριοποίηση των φρούτων σε τρεις κατηγορίες 0- (κακή ποιότητα), έως 10 -(Άριστη Ποιότητα), σύμφωνα με τις παρακάτω προδιαγραφές:

Βάρος [10-100]	Σκληρότητα [10-100]	Ποιότητα [0-10]
20	20	2
30	25	5
80	40	8
90	45	9
100	100	1
10	10	0

Πίνακας 4-3: Κατάταξη Δειγμάτων σε Κατηγορίες

Να σχεδιαστεί ασαφής ελεγκτής ο οποίος να αποφασίζει για την θερμοκρασία του ξηραντήριου με σφάλμα στις παραπάνω ενδεικτικές μετρήσεις μικρότερο του 3%. Ο σχεδιασμός του συστήματος να γίνει με τη μεθοδολογία που δόθηκε στην § 2.3 σελ. 16.

4.4 Αναγνώριση μη γραμμικού συστήματος με Νευρο-Ασαφή συστήματα

Δίνεται ένα μη γραμμικό σύστημα δύο εισόδων και μιας εξόδου το οποίο περιγράφεται από τη μη γραμμική εξίσωση:

$$y = (1 + x_1^{-1.5} + 3x_2^2) \qquad (4.20)$$

Κάθε $x_i \,|\, i = 1,2,3$ είναι ένας τυχαίος πραγματικός αριθμός στο διάστημα [1,5]

Να σχεδιαστεί ασαφές σύστημα τύπου α) T-S-K, β) mamdani γ) crisp output, το οποίο να αναγνωρίζει το σύστημα με όσο το δυνατό μικρότερο σφάλμα. Για το σύστημα τύπου Tα-S-K μπορεί να χρησιμοποιηθεί το *ANFIS* της ασαφούς εργαλειοθήκης του MATLAB®.

4.5 Κατάταξη δειγμάτων σε κατηγορίες με Νευρο-Ασαφή συστήματα

Μια οικογένεια κρίνων (*IRIS)* διαχωρίζεται σε τρεις κατηγορίες ανάλογα με το μήκος των πετάλων το μήκος των σέπαλων το πλάτος των πέταλων και το πλάτος των σέπαλων. Για την αυτόματη κατάταξη των κρίνων συλλέγονται μετρήσεις 150 μετρήσεις στη μορφή:

A/A	Sepal-lenght	Petal-length	Sepal-width	Petal-width	Category
1	5	3.5	1.3	0.3	1
2	5.6	3	4.1	1.3	2
.	.	.	.	.	.
.	.	.	.	.	.
.	.	.	.	.	.
150	6.8	3.2	5.9	2.3	3

Πίνακας 4-4: Μορφή δειγμάτων εισόδου εξόδου για το IRIS

Να σχεδιαστεί και να εκπαιδευτεί ένα ασαφές μοντέλο τύπου TSK το οποίο να λειτουργεί ως ταξινομητής *classifier*. Τα πρώτα 75 δείγματα να χρησιμοποιηθούν για την εκπαίδευση του ταξινομητή και τα υπόλοιπα 75 για τον έλεγχο της ικανότητας ταξινόμησης του.

5 Βιβλιογραφία

[1] Rogers, M. Hoshuai Y. "The future looks 'fuzzy'", Newsweek, May 28, p 46, 1990.

[2] R. Timothy, "Fuzzy Logic with Engineering Applications", New York: McGraw Hill, 1994

[3] Zadeh, Lotfi, "Fuzzy Sets", Information and Control, 1965.

[4] Zadeh, Lotfi, "Outline of a New Approach to the Analysis of Complex Systems", IEEE Trans. on Sys., Man and Cyb. 3, 1973.

[5] Ruspini E., Bonissone P., Pedrycz W.(Eds.), "*Handbook of Fuzzy Computation*", Oxford Uivesrity Press, 1997

[6] Asai K, "*Fuzzy Systems for Information Processing*", Burke: IOS Press, Sept. 1995

[7] Asai K, "*Fuzzy Systems for Management*", Burke: IOS Press, April 1995

[8] Baldwin J, "*Fuzzy Logic*", New York: John Wiley and Sons, July 1996

[9] Bandemer-Hans, "*Fuzzy Sets, Fuzzy Logic, Fuzzy Methods with Applications*", New York: John Wiley and Sons, 1995

[10] Bandler, W., and Kohout, L.J., "Fuzzy Power Sets and Fuzzy Implication Operators", *Fuzzy Sets and Systems* 4:13-30, 1980.

[11] Berkan-Riza-C, Trubatch-Sheldon, "*Fuzzy System Design Principles: Identification and Resolution of the Practical Problems in Designing Fuzzy If-Then Rule Base*", Piscataway: IEEE, May 1997

[12] Bezdek, James C, "Fuzzy Models --- What Are They, and Why*?", IEEE Transactions on Fuzzy Systems,* 1:1, pp. 1-6, 1993.

[13] Cox, Earl, "*The Fuzzy Systems Handbook*", Academic Press Limited, 1998.

[14] Cox, Earl, "*Fuzzy Models with Java*", Rockland: Charles-River-Media, 1999

[15] Driankov Dimiter, Hellendoorn Hans, and Reinfrank, "*An Introduction to Fuzzy Control*", [16]Springer-Verlag, New York, 1993.

[17] Dubois, Didier, and Prade, Henri, "A Class of Fuzzy Measures Based on Triangle Inequalities", *Int. J. Gen. Sys*. 8.

[18] Dubois, Didier, and Prade, Henri, "Fuzzy Numbers: An Overview", *Analysis of Fuzzy Information* 1:3-39, CRC Press, Boca Raton, 1987.

[19] Dubois, Didier, and Prade, Henri, "Mean Value of a Fuzzy Number", *Fuzzy Sets and Systems* 24(3):279-300, 1987.

[20] Jamshidi-Mohammad, Vadiee-Nader, Ross-Timothy*, "Fuzzy Logic and Control: Software and Hardware Applications*", Paramus: Prentice Hall, 1993

[21] Kaufmann, A., and Gupta, M.M., "*Introduction to Fuzzy Arithmetic*", Reinhold, New York, 1985.

[22] Klir George J, Yuan Bo, "*Fuzzy Sets and Fuzzy Logic: Theory and Applications*", Paramus: Prentice-Hall, 1995

[23] Klir G J, Yuan Yu Hsien, "*Fuzzy Theory and Applications*", Paramus: Prentice-Hall, 1997

[24] Kosko B., "*Fuzzy Engineering*", Paramus: Prentice-Hall, 1996

[25] Kosko B., "*Fuzzy Thinking: The New Science of Fuzzy Logic*", New York: Hyperion, 1993

[26] Kruse R., Gebhardt J-E, Klowon-F, "*Foundations of Fuzzy Systems: Theory and Applications*", New York: John Wiley, 1994

[27] McNeil, Martin, and Thro, Ellen, "*Fuzzy Logic, A Practical Approach*". Academic Press, 1994.

[28] Mizumoto, "*Improvement Methods of Fuzzy Control*", in Proceedings of the 3rd IFSA Congress, 1989.

[29] Nguyen H-T, Walker-E, "*A First Course in Fuzzy Logic*", Boca Raton: CRC Press, 1996

[30] Nguyen H-T, (Editor), Sugeno M, Tong R, Yager R-R, "*Theoretical Aspects of Fuzzy Control*", New York: John Wiley, 1995

[31] Passino K. M, "*Fuzzy Control: Theory and Applications*", Reading: Addison Wesley Longman, 1997

[32] Pedrycz W., "*Fuzzy Control and Fuzzy Systems*", New York: John-Wiley, 1993

[33] Pedrycz W., Gomide F., "*An Introduction to Fuzzy Sets - Analysis and Design*", MIT Press, 1998

[34] Pedrycz W., Zadeh Lofti, "*Fuzzy Sets Engineering*", Boca Raton: CRC Press, 1995.

[35] Terano T., Asai K., Sugeno M., editors, "*Applied Fuzzy Systems*", translated by C. Ascchmann, AP Professional.

[36] Wang Li Xin, "*A Course in Fuzzy Systems and Control*", Paramus: Prentice-Hall, 1996

[37] Yager, R.R., and Zadeh, L.A., "*An Introduction to Fuzzy Logic Applications in Intelligent Systems*", Kluwer Academic Publishers, 1991.

[38] Zadeh, Lotfi, "The Calculus of Fuzzy Restrictions", *Fuzzy Sets and Applications to Cognitive and Decision Making Processes*, edited by L. A. Zadeh et. al., Academic Press, New York, 1975, pages 1-39.

[39] Zimmermann-H-J, "*Fuzzy Set Theory -- and Its Applications*", Kluwer Academic Publishers, 1996

6 Ευρετήριο