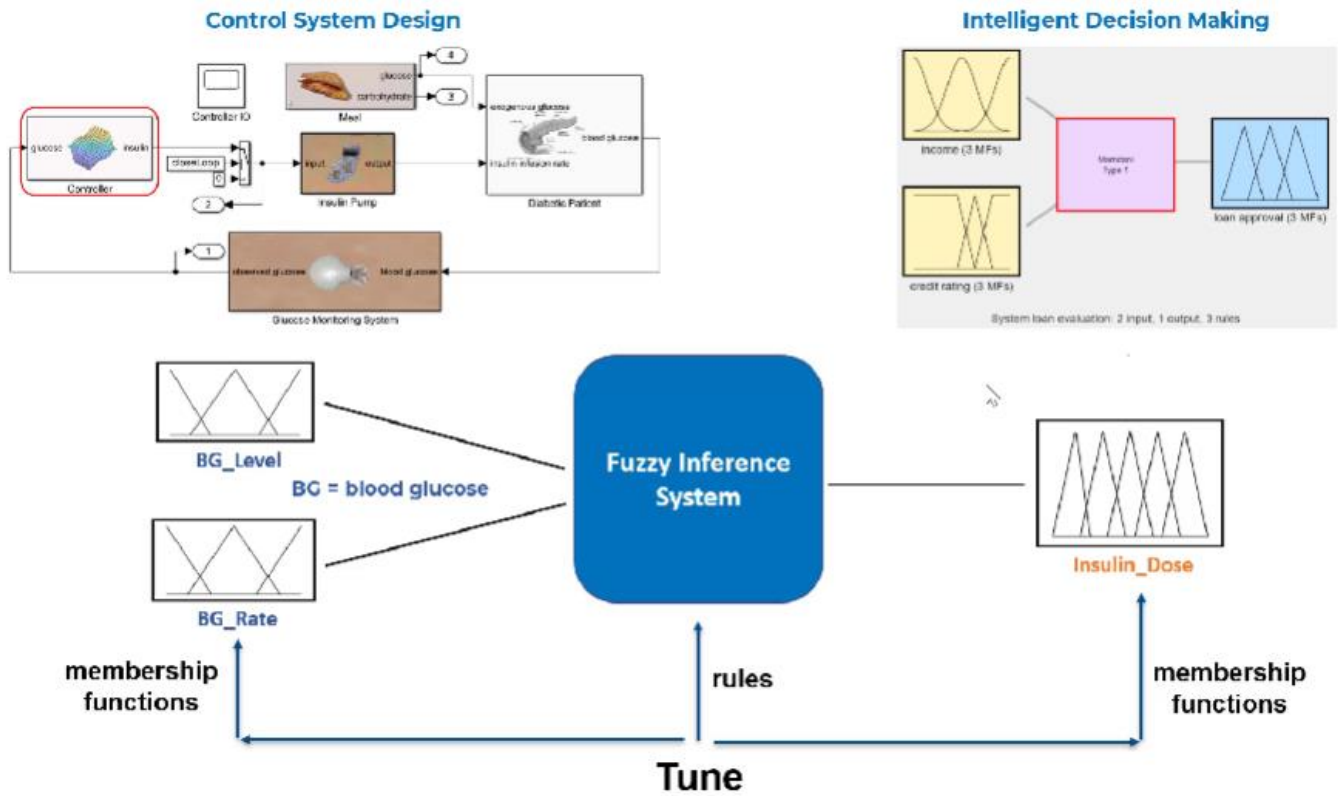


# ΥΠΟΛΟΓΙΣΤΙΚΗ ΝΟΗΜΟΣΥΝΗ

## ΕΡΓΑΣΤΗΡΙΑΚΗ ΑΣΚΗΣΗ ΑΣΑΦΗΣ ΛΟΓΙΚΗΣ



**Θέμα: Σύστημα φρεναρίσματος έκτακτης ανάγκης**

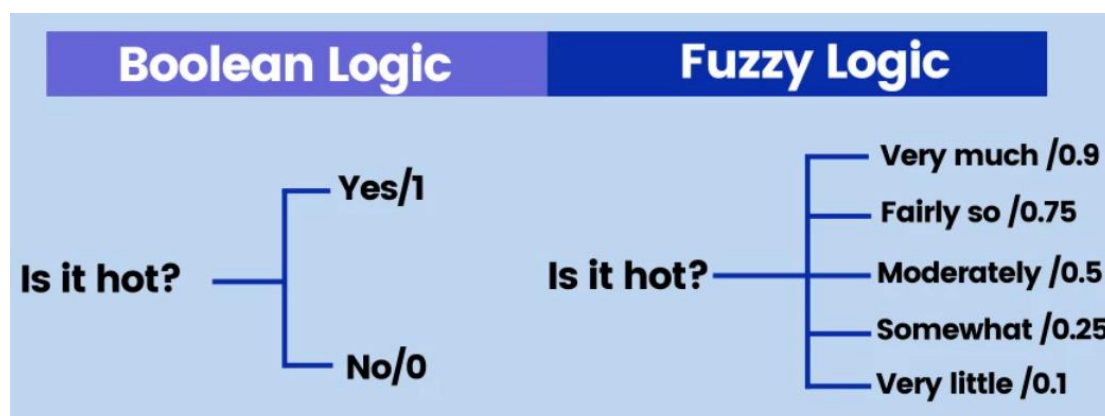
## Ασαφής Λογική (Fuzzy Logic)

Η θεωρία ασαφών συνόλων είναι ένας τρόπος να προσδιορίζουμε πόσο καλά ένα αντικείμενο ικανοποιεί μια αόριστη περιγραφή. Αυτός ο προσδιορισμός είναι υποκειμενικός, γιατί εξαρτάται από το πώς αντιλαμβάνεται ο καθένας τους λεκτικούς προσδιορισμούς ποσοτικών μεγεθών. Η ασαφής λογική χρησιμοποιεί ασαφή σύνολα, προτάσεις και κανόνες για να περιγράψει και να αναλύσει καταστάσεις, που εμπεριέχουν κάποια αβεβαιότητα.

Για παράδειγμα η πρόταση κάνει κρύο όταν η θερμοκρασία είναι 18 βαθμοί είναι αληθής; Εξαρτάται από το πώς αντιλαμβάνεται ο καθένας την έκφραση. Η ασάφεια δεν προκύπτει λόγω της αβεβαιότητάς μας για τον εξωτερικό κόσμο (η ακριβής θερμοκρασία είναι γνωστή). Προκύπτει γιατί ο γλωσσολογικός όρος «κρύο» δεν διαχωρίζει τη θερμοκρασία σε δύο κατηγορίες αυστηρά, αλλά σε πολλές διαβαθμίσεις.

Έχει εφαρμογές σε πολλούς τομείς, όπως η τεχνητή νοημοσύνη, η ρομποτική, η λήψη αποφάσεων, η επεξεργασία φυσικής γλώσσας και η επιστήμη των υπολογιστών γενικότερα. Σε αυτούς τους τομείς η ασάφεια μπορεί να είναι απαραίτητη για να μοντελοποιηθούν και να αντιμετωπιστούν προβλήματα όπου η σαφήνεια και η ακρίβεια δεν είναι εφικτές.

Η ασαφής λογική συνήθως βασίζεται σε ασαφή σύνολα, όπου η απόδοση της συμπεριφοράς ενός στοιχείου σε ένα σύνολο είναι πιθανοκρατική αντί για δυαδική. Αυτό επιτρέπει την απεικόνιση της βαθμωτής ασαφούς γλώσσας που περιγράφει την αβεβαιότητα. Η ασαφής λογική, στην ουσία επεκτείνει τη δυαδική λογική.



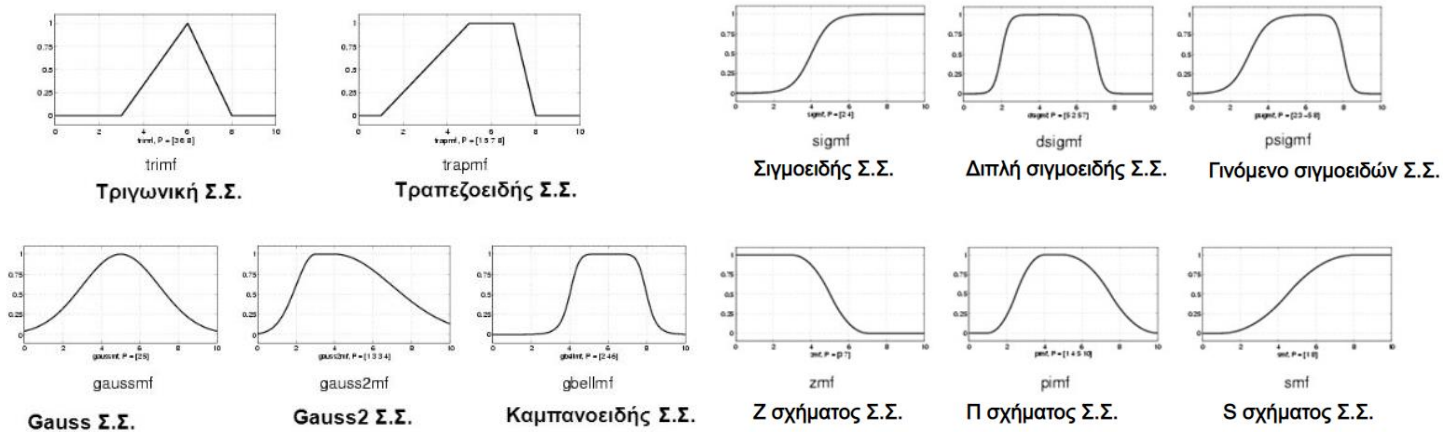
**Ασαφές Σύνολο:** Είναι μια ομάδα στοιχείων που μοιάζουν ως προς τις ιδιότητες του συνόλου σε κάποιο βαθμό.

Ένα ασαφές σύνολο μπορεί να περιλαμβάνει κάποιο στοιχείο μέχρι κάποιο βαθμό. Συνδυάζοντας ασαφή σύνολα με ασαφείς κανόνες μπορούμε να πάρουμε αποφάσεις.

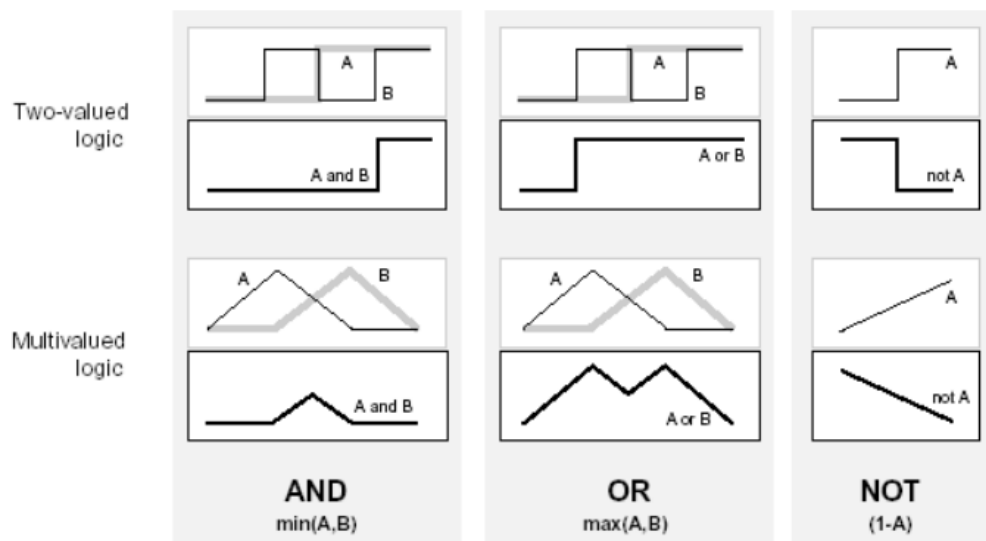
Στην κλασσική λογική μία πρόταση ή είναι ή δεν είναι αληθής. Αντίθετα, στην ασαφή λογική μια πρόταση είναι αληθής μέχρι κάποιο βαθμό. Ο βαθμός συμμετοχής στο σύνολο, ονομάζεται **συνάρτηση συμμετοχής**.

- $f(x)=0$  το αντικείμενο δεν ανήκει στο σύνολο.
- $f(x)=1$  είναι σίγουρα μέλος του συνόλου.
- Οι υπόλοιπες τιμές για την  $f(x)$  δείχνουν το βαθμό συμμετοχής.

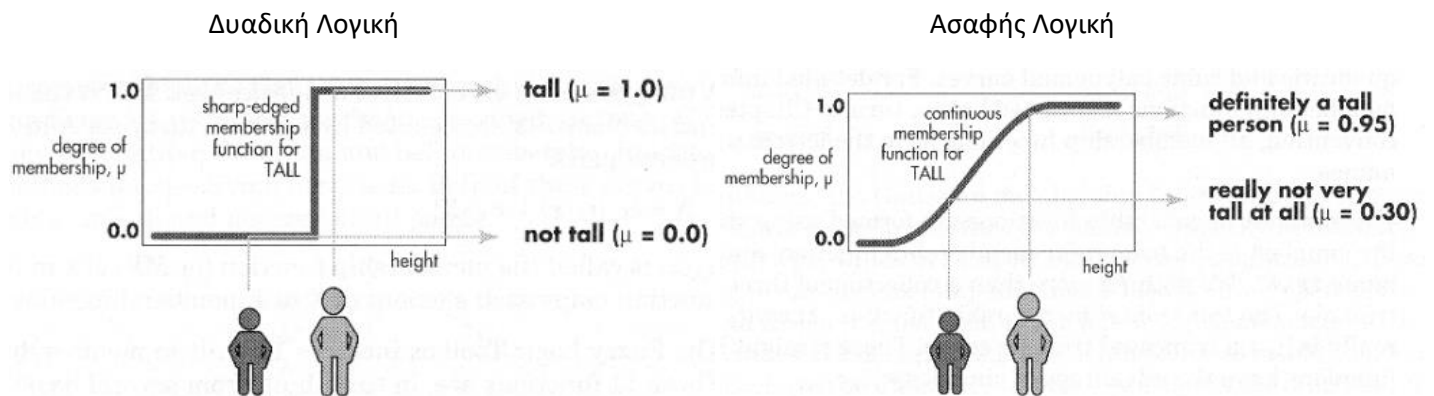
Παρακάτω φαίνονται διάφορες συναρτήσεις συμμετοχής που υπάρχουν στο toolbox του Matlab.



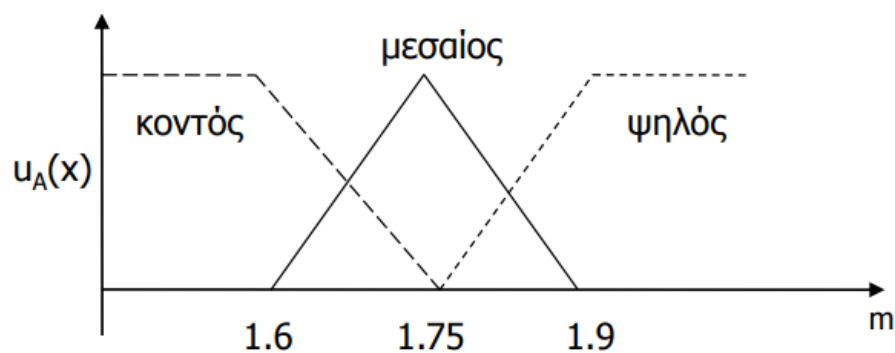
Στο παρακάτω σχήμα φαίνονται οι ασαφείς τελεστές. Στο πάνω μέρος έχουμε την περίπτωση συνόλων με δύο τιμές ενώ στο κάτω φαίνεται ο τρόπος με τον οποίο λειτουργούν οι τελεστές στην περίπτωση που οι τιμές αληθείας A και B μεταβάλλονται συνεχώς από το 0 στο 1.



Στο παράδειγμα που φαίνεται στην παρακάτω εικόνα η συνάρτηση συμμετοχής είναι μια καμπύλη η οποία καθορίζει τον βαθμό στον οποίο κάθε σημείο του πεδίου ορισμού διαθέτει μια συγκεκριμένη ιδιότητα. Η συνάρτηση συμμετοχής ορίζεται στο διάστημα  $[0,1]$ .



Μια πιο απλή απεικόνιση της συνάρτησης μεταφοράς φαίνεται παρακάτω.



#### Σύστημα φρεναρίσματος έκτακτης ανάγκης

Θέλουμε να σχεδιάσουμε ένα σύστημα ασαφούς λογικής στο Matlab, στο οποίο να αναπτύσσεται ένα σύστημα φρεναρίσματος έκτακτης ανάγκης για αυτοκίνητα τα οποία κυκλοφορούν σε αστικές περιοχές.

Η λειτουργία του συστήματος ενεργοποιείται όταν το αυτοκίνητο κινείται με ταχύτητα **μικρότερη από 30 km/h** και όταν οι αισθητήρες του ανιχνεύουν εμπόδια στην πορεία κίνησης του σε απόσταση **μικρότερη από 10 m**. Οι είσοδοι του συστήματος είναι η ταχύτητα του αυτοκινήτου (km/h) και η απόσταση του εμποδίου (m), ενώ η έξοδος του είναι ο βαθμός φρεναρίσματος ο οποίος και θα εκφράζεται με μία τιμή από 0 μέχρι 1.

## Επίλυση του προβλήματος

Το πρόβλημα επιλύθηκε με τη βοήθεια του fuzzy logic toolbox του Matlab.

## Καθορισμός Προβλήματος

**Είσοδοι:** Ταχύτητα του αυτοκινήτου (km/h)<30km/h και Απόσταση του εμποδίου (m)<10m

Συναρτήσεις Συμμετοχής για ταχύτητα (slow, medium, fast)

Συναρτήσεις Συμμετοχής για απόσταση (near, far)

**Έξοδος:** Βαθμός φρεναρίσματος [0,1]

Συναρτήσεις Συμμετοχής (gentle, moderate, sudden)

## Κανόνες

If car is moving **slow** and the obstacle is **far** then the braking is **gentle**

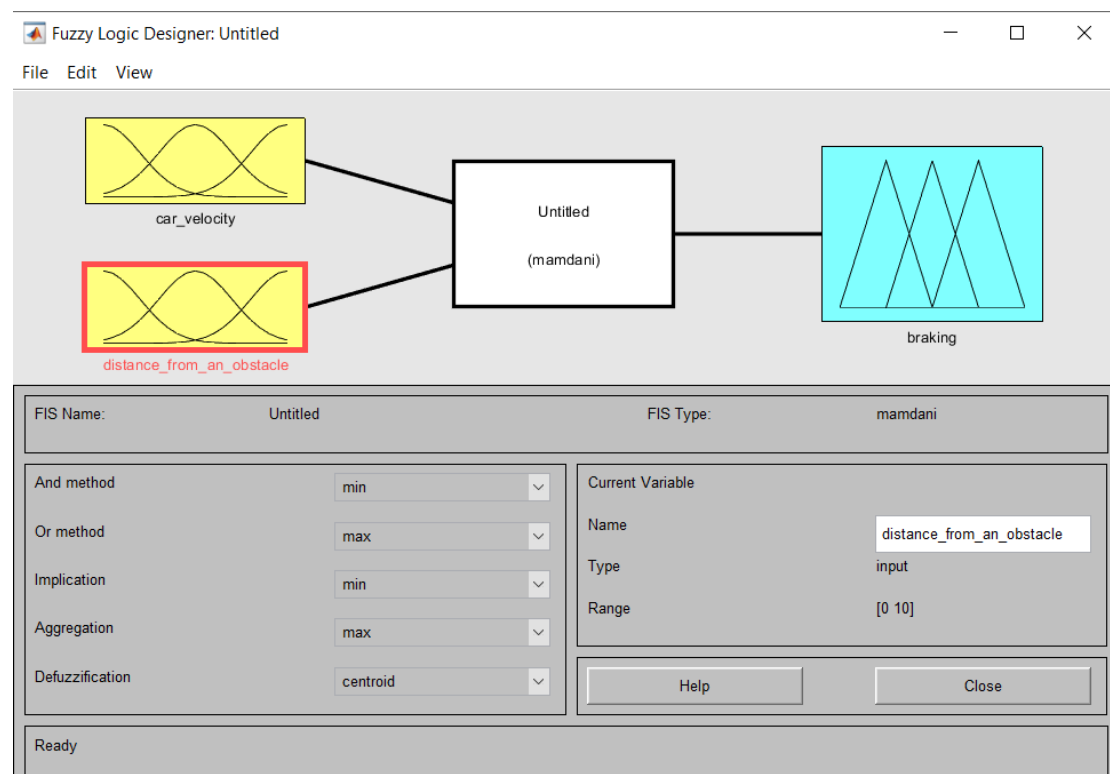
If car is moving with **medium** speed and the obstacle is **far** then the braking is **moderate**

If car is moving **slow** and the obstacle is **near** then the braking is **moderate**

If car is moving with **medium** speed and the obstacle is **near** then the braking is **sudden**

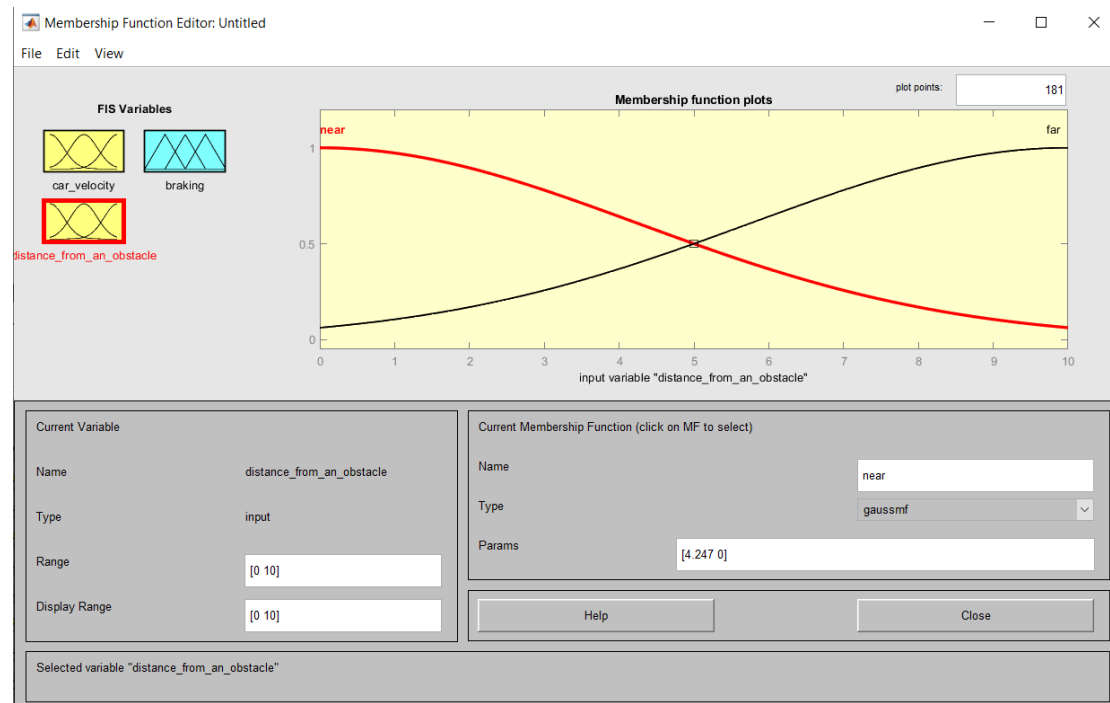
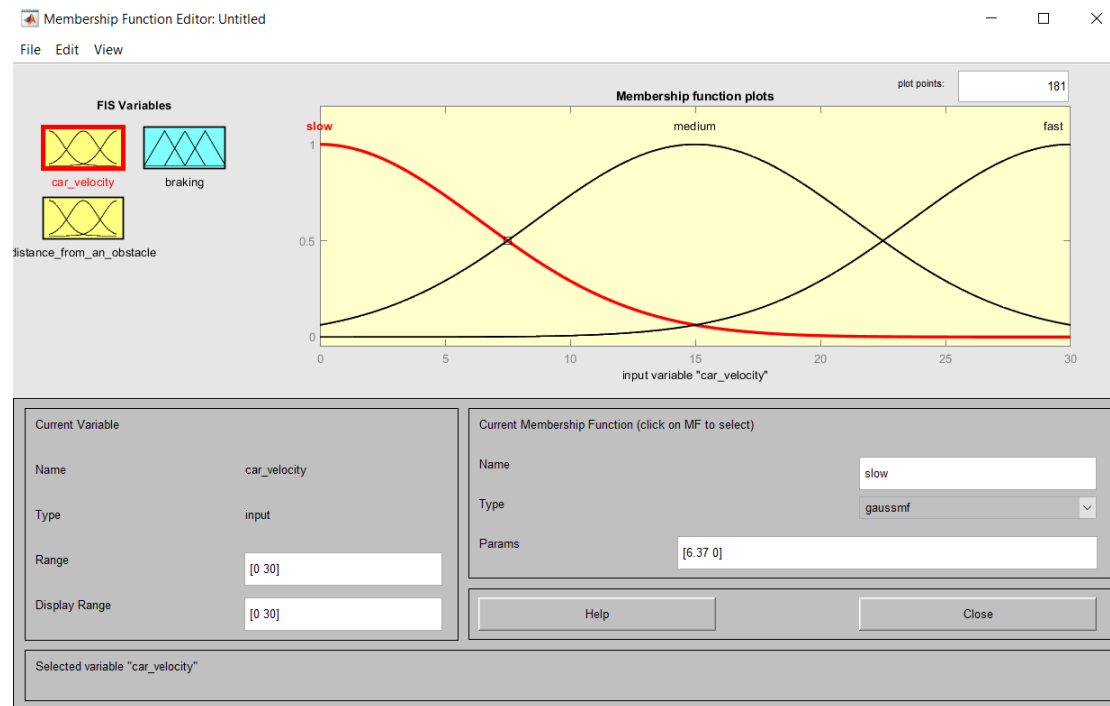
If car is moving **fast** then the braking is **sudden**

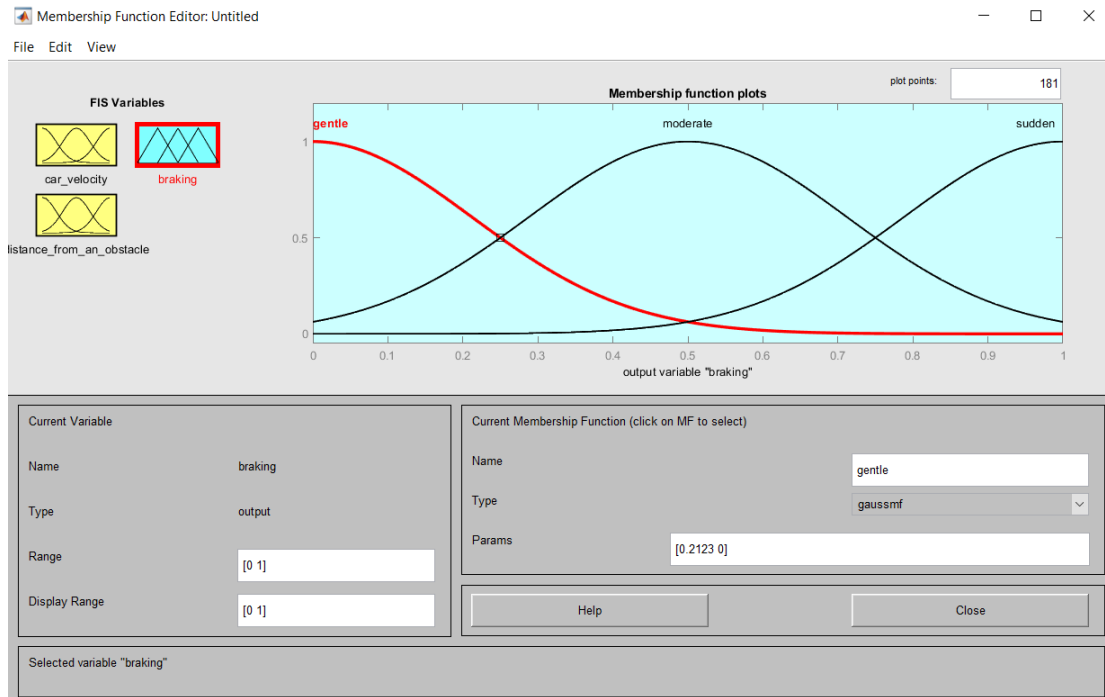
Στις παρακάτω εικόνες βλέπουμε την υλοποίηση της άσκησης στο toolbox.



Έχουμε δύο εισόδους *car\_velocity* με range [0,30] και *distance\_from\_an\_obstacle* με range [0,10]. Η έξοδος ονομάζεται *braking* με range [0,1].

Για συνάρτηση συμμετοχής χρησιμοποιήσαμε την *gaussmf* (Γκαουσσισιανή Συνάρτηση).





Παρακάτω φαίνονται οι κανόνες που ορίσαμε για το πρόβλημα.

Rule Editor: Untitled

File Edit View Options

1. If (car\_velocity is slow) and (distance\_from\_an\_obstacle is far) then (braking is gentle) (1)  
 2. If (car\_velocity is medium) and (distance\_from\_an\_obstacle is far) then (braking is moderate) (1)  
 3. If (car\_velocity is slow) and (distance\_from\_an\_obstacle is near) then (braking is moderate) (1)  
 4. If (car\_velocity is medium) and (distance\_from\_an\_obstacle is near) then (braking is sudden) (1)  
 5. If (car\_velocity is fast) then (braking is sudden) (1)

If car\_velocity is and distance\_from\_an\_ Then braking is

slow medium near gentle moderate  
 fast none sudden  
 none

☐ not ☐ not ☐ not

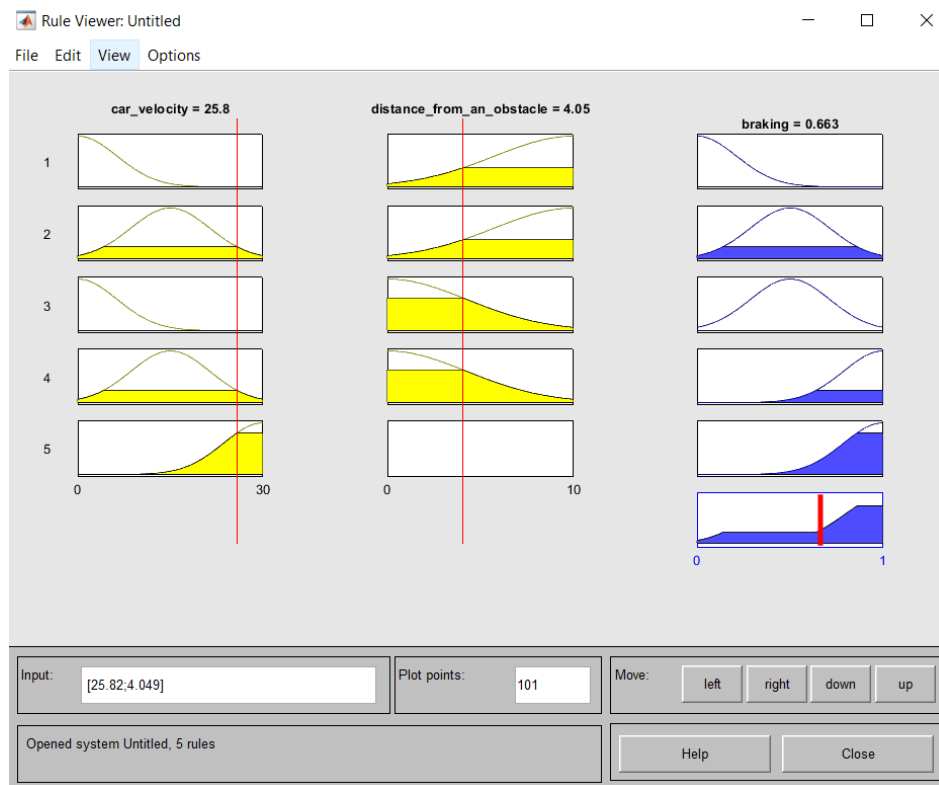
Connection Weight:

☒ or 1 ☐ and

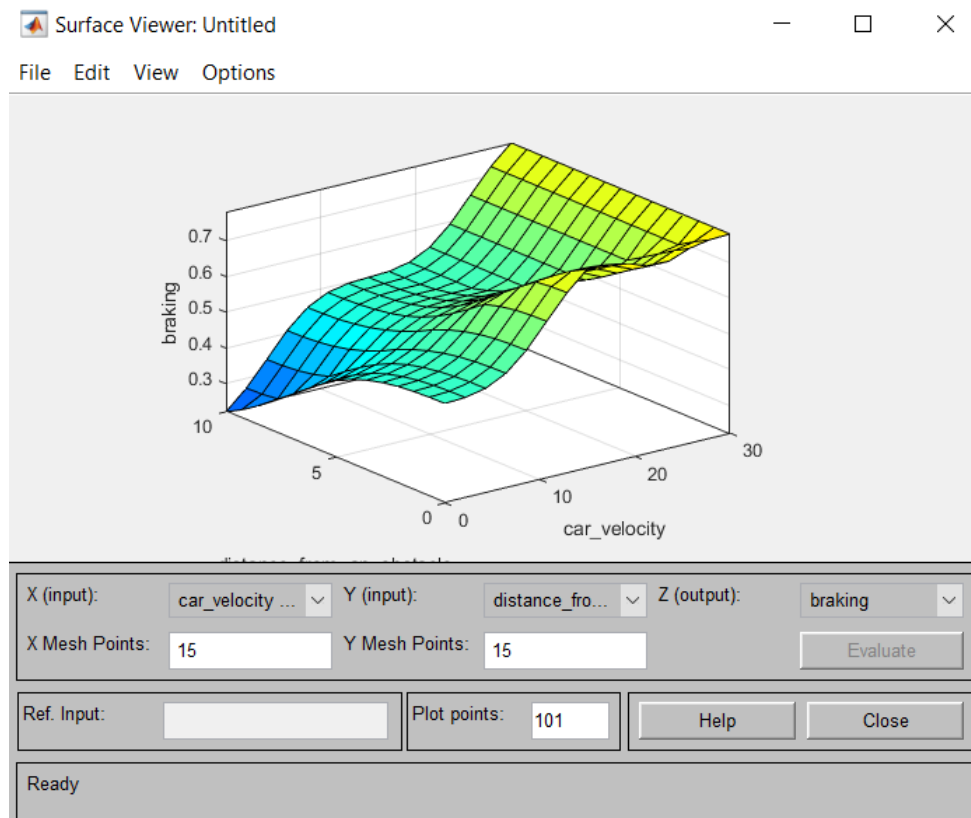
Delete rule Add rule Change rule << >>

The rule is added Help Close

Ακόμη, υπάρχει η δυνατότητα να προβάλλουμε τους κανόνες που θέσαμε σε μορφή διαγραμμάτων και να διαπιστώσουμε εάν λειτουργούν σωστά



Τέλος, υπάρχει η δυνατότητα να πάρουμε μια τρισδιάστατη γραφική παράσταση των εισόδων σε σχέση με την έξοδο.





## Αποτελέσματα και Συμπεράσματα

Για την καλύτερη κατανόηση των αποτελεσμάτων γράψαμε ένα μικρό αρχείο που χρησιμοποιεί τη συνάρτηση **evalfis** για να τυπώσει με τρόπο αρκετά περιγραφικό το αποτέλεσμα για 5 διαφορετικές εισόδους.

```
clc;clear;
fis = readfis('Emergency_Braking.fis');
showrule(fis)
% Speed, distance
inputs = [3 6 14 26 30; 10 4 6 9 2];
output = evalfis(fis, inputs);
disp("Braking:");

for i = 1:length(output)
    if inputs(1, i) < 10
        speed = ' slow ';
    elseif inputs(1, i) > 20
        speed = ' fast ';
    else
        speed = ' medium ';
    end

    if inputs(2, i) < 5
        distance = ' near ';
    else
        distance = ' far ';
    end

    if output(i) < 0.33
        braking = ' (gentle).';
    elseif output(i) > 0.66
        braking = ' (sudden).';
    else
        braking = ' (moderate).';
    end

    disp(['The velocity of the car is', speed, 'and the distance
of the obstacle is', distance, '-> Braking is ',
num2str(output(i)), braking]);
end
```

Με

την εντολή **showrule(fis)** μπορούμε να δούμε τους κανόνες που έχουμε θέσει.

Στη συνέχεια ανάλογα με τις τιμές των εισόδων και την τιμή της εξόδου κάνουμε τις κατάλληλες εκτυπώσεις. **Βέβαια τα threshold που θέσαμε δεν είναι εντελώς σωστά, αλλά για αυτές τις τιμές εισόδων τροποποιήθηκαν ώστε να μας εκτυπώνουν αυτό που θέλουμε.**

Αν η εκτύπωση έρχεται σε αντίθεση με κάποιον κανόνα, τότε πρέπει να αλλάξουμε τα threshold. **Δεν μπορεί να οριστεί τιμή threshold, γιατί υπάρχει ασάφεια. Εδώ ορίστηκε απλά μια προσεγγιστική τιμή για να κάνουμε τον έλεγχο με τους κανόνες.**

Παρατηρούμε ότι όλα τα αποτελέσματα συμφωνούν με τους κανόνες.

```
'1. If (car_velocity is slow) and (distance_from_an_obstacle is far) then (braking is gentle) (1) '
'2. If (car_velocity is medium) and (distance_from_an_obstacle is far) then (braking is moderate) (1) '
'3. If (car_velocity is slow) and (distance_from_an_obstacle is near) then (braking is moderate) (1) '
'4. If (car_velocity is medium) and (distance_from_an_obstacle is near) then (braking is sudden) (1) '
'5. If (car_velocity is fast) then (braking is sudden) (1) '
```

Braking:

```
The velocity of the car is slow and the distance of the obstacle is far -> Braking is 0.29923 (gentle).
The velocity of the car is slow and the distance of the obstacle is near -> Braking is 0.49999 (moderate).
The velocity of the car is medium and the distance of the obstacle is far -> Braking is 0.53435 (moderate).
The velocity of the car is fast and the distance of the obstacle is far -> Braking is 0.66918 (sudden).
The velocity of the car is fast and the distance of the obstacle is near -> Braking is 0.77832 (sudden).
```

Για το παράδειγμα speed=25 και distance=10, το αποτέλεσμα είναι:

The velocity of the car is fast and the distance of the obstacle is far -> Braking is 0.6368 (moderate).

**Το αποτέλεσμα αυτό έρχεται σε αντίθεση με τον κανόνα 5. Αυτό είπαμε ότι συμβαίνει γιατί ορίσαμε αυθαίρετα κάποια threshold, ενώ υπάρχει αβεβαιότητα.**

Άρα, ας αφήσουμε απλά το πρόγραμμα να τυπώνει τις τιμές εισόδου και εξόδου. Αυτό είναι και το πιο σωστό.

```
clc; clear;
fis = readfis('Emergency_Braking.fis');
%showrule(fis)

% Speed, distance
inputs = [3 6 14 25 30; 10 4 6 10 2];
output = evalfis(fis, inputs);
disp("Braking:");

for i = 1:length(output)
    disp(['The velocity of the car is ', num2str(inputs(1,i)), '
and the distance of the obstacle is ', num2str(inputs(2,i)), ' ->
Braking is ', num2str(output(i))]);
end
```

Οι παρακάτω εκφράσεις τονίζουν την ασάφεια που περιέχεται στο πρόβλημα.

Braking:

```
The velocity of the car is 3 and the distance of the obstacle is 10 -> Braking is 0.29923
The velocity of the car is 6 and the distance of the obstacle is 4 -> Braking is 0.49999
The velocity of the car is 14 and the distance of the obstacle is 6 -> Braking is 0.53435
The velocity of the car is 25 and the distance of the obstacle is 10 -> Braking is 0.6368
The velocity of the car is 30 and the distance of the obstacle is 2 -> Braking is 0.77832
```

Τελικά, καταλήξαμε σε μια ικανοποιητική λύση, με τη χρήση ασαφούς λογικής για το πρόβλημα του φρεναρίσματος έκτακτης ανάγκης. Δεν εξαλείψαμε την αβεβαιότητα που εμπεριέχει το πρόβλημα, αλλά τη χρησιμοποιήσαμε για να το επιλύσουμε. Είδαμε ότι αν εξαλείψουμε την αβεβαιότητα (χρησιμοποιώντας thresholds), οδηγούμαστε σε ανακριβή, μπορεί και λανθασμένη λύση.

### Βιβλιογραφία

- Διαφάνειες μαθήματος και εργαστηρίου, “Υπολογιστική Νοημοσύνη” Δημοκρίτειο Πανεπιστήμιο-Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών.
- <https://www.mathworks.com/help/fuzzy/what-is-fuzzy-logic.html>
- <https://slideplayer.gr/slide/11139705/>
- <https://hackr.io/blog/what-is-fuzzy-logic>
- [https://eclass.upatras.gr/modules/document/file.php/EE846/fuzzy\\_logic.pdf](https://eclass.upatras.gr/modules/document/file.php/EE846/fuzzy_logic.pdf)