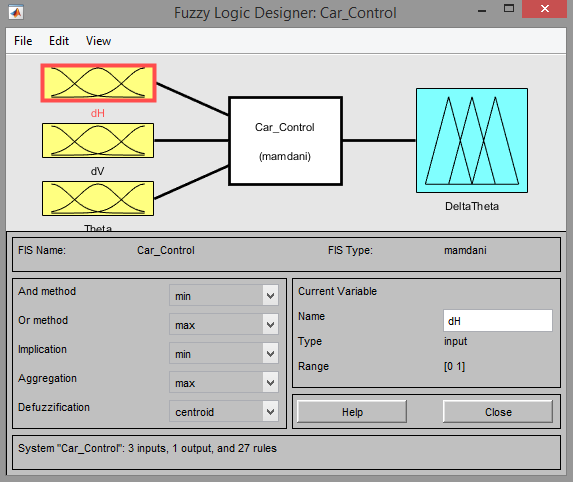
## **ΑΡΙΣΤΟΤΕΛΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΟΝΙΚΗΣ ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ**

## **ΥΠΟΛΟΓΙΣΤΩΝ ΤΟΜΕΑΣ ΗΛΕΚΤΡΟΝΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ**



**ΕΠΩΝΥΜΟ: ΠΑΠΑΚΩΝΣΤΑΝΤΙΝΟΥ**

**ΟΝΟΜΑ: ΑΝΤΩΝΙΟΣ**

**ΑΕΜ: 8977**

## ΘΕΣΣΑΛΟΝΙΚΗ 2020

Η παρούσα εργασία έχει ως σκοπό τον σχεδιασμό ενός ασαφούς ελεγκτή (FLC) για τον έλεγχο της κίνησης ενός οχήματος με σκοπό την αποφυγή των εμποδίων όπως παρουσιάζεται στην εικόνα 1.



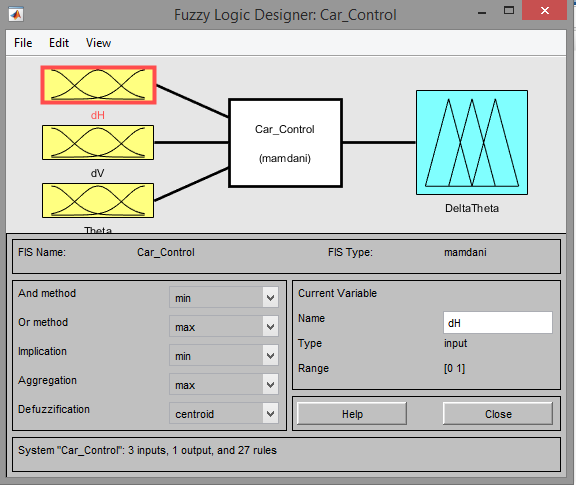
**Εικόνα 1**

Σκοπός του FLC είναι να οδηγήσει το όχημα με ασφάλεια (χωρίς να ακουμπήσει στα σταθερά εμπόδια) στην επιθυμητή θέση. Το όχημα διαθέτει κατάλληλους αισθητήρες ώστε να υπολογίζει κάθε χρονική στιγμή την κάθετη ( ) και την οριζόντια ( ) απόστασή του από τα εμπόδια.

Το μέτρο της ταχύτητάς του είναι σταθερό και ίσο με .

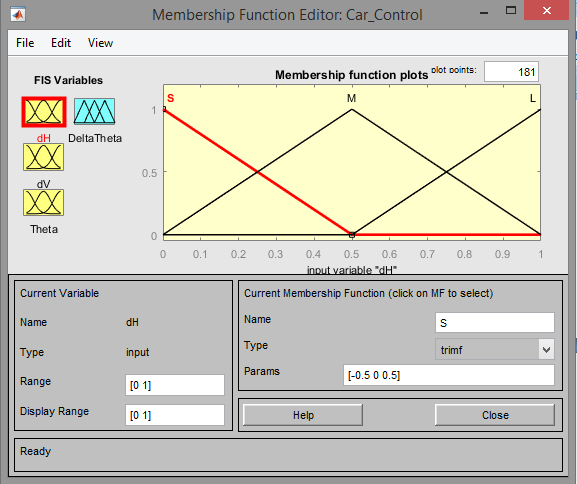
Έχοντας ως εισόδους την κάθετη και οριζόντια απόσταση του οχήματος από τα εμπόδια, αλλά και τη διεύθυνση της ταχύτητας θ, o FLC πρέπει να αποφασίζει τη μεταβολή στη διεύθυνση (Δθ) της ταχύτητάς του ώστε να μεταφερθεί στην επιθυμητή θέση:

Για το λόγο αυτό, στο περιβάλλον της Matlab δημιουργήθηκε ένας Fuzzy Control Logic (FLC) με το όνομα Car\_Control.fis, όπως φαίνεται στην εικόνα 2.



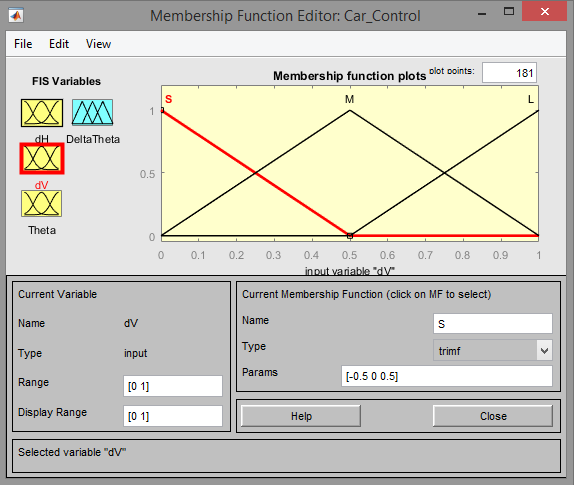
**Εικόνα 2:** Ο FLC

Οι τρεις είσοδοι , και θ παρουσιάζονται στις εικόνες 3,4 και 5 αντίστοιχα.



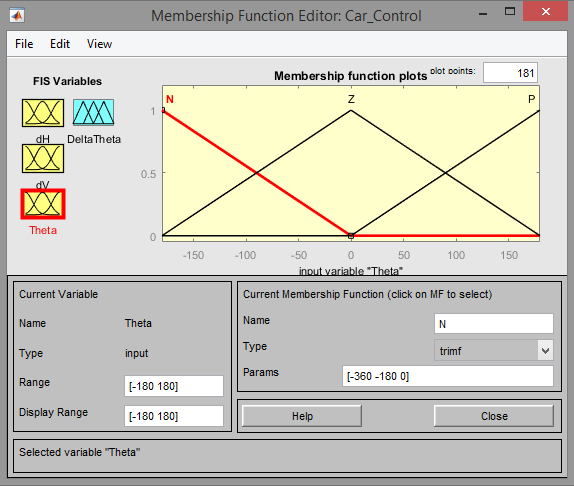
**Εικόνα 3:** Η είσοδος

Η εξίσωση της εισόδου δίνεται από τη σχέση:



**Εικόνα 4:** Η είσοδος

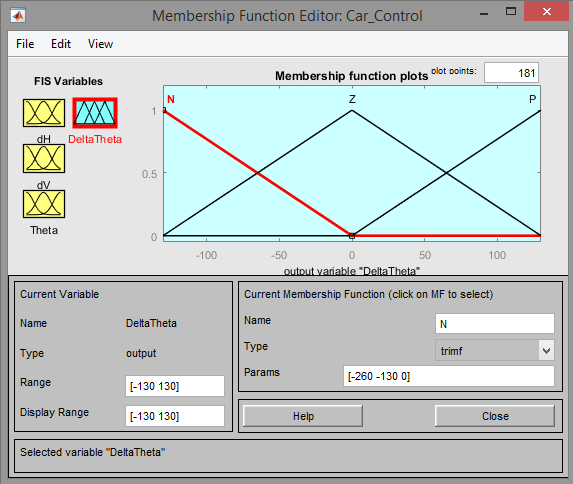
Η εξίσωση της εισόδου δίνεται από τη σχέση:



**Εικόνα 5:** Η είσοδος

Η εξίσωση της εισόδου δίνεται από τη σχέση:

Η έξοδος Δθ παρουσιάζεται στην εικόνα 6.



**Εικόνα 6:** Η έξοδος

Η εξίσωση της εξόδου δίνεται από τη σχέση:

Στον πίνακα που ακολουθεί, Πίνακας 1, παρουσιάζονται οι κανόνες του FLC.

**Πίνακας 1:** Οι κανόνες του FLC

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| a/a | dH | dV | θ | Δθ |
| 1 | S | S | N | **P** |
| 2 | S | S | Z | **N** |
| 3 | S | S | P | **P** |
| 4 | S | M | N | **P** |
| 5 | S | M | Z | P |
| 6 | S | M | P | **Ν** |
| 7 | S | L | N | **N** |
| 8 | S | L | Z | **N** |
| 9 | S | L | P | **P** |
| 10 | M | S | N | **N** |
| 11 | M | S | Z | **P** |
| 12 | M | S | P | **P** |
| 13 | M | M | N | **Z** |
| 14 | M | M | Z | **Z** |
| 15 | M | M | P | **Z** |
| 16 | M | L | N | **P** |
| 17 | M | L | Z | **P** |
| 18 | M | L | P | **Z** |
| 19 | L | S | N | **P** |
| 20 | L | S | Z | **Z** |
| 21 | L | S | P | **Z** |
| 22 | L | M | N | **Z** |
| 23 | L | M | Z | **P** |
| 24 | L | M | P | **N** |
| 25 | L | L | N | **N** |
| 26 | L | L | Z | **Z** |
| 27 | L | L | P | **N** |

Οι νέες συντεταγμένες προκύπτουν από τους τύπους:

Όπου:

: Η επόμενη τετμημένη της θέσης του κινούμενου οχήματος

: Η επόμενη τεταμένη της θέσης του κινούμενου οχήματος

: Η κατεύθυνση της ταχύτητας

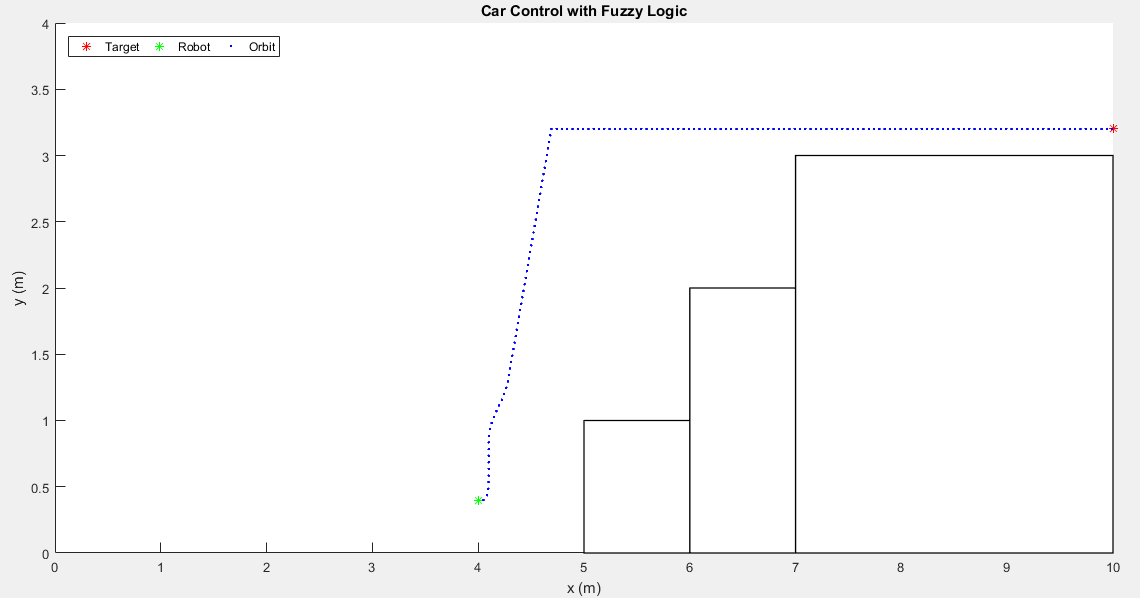
: Η κατεύθυνση της ταχύτηταςτην επόμενη χρονική στιγμή

*: Το* μέτρο της ταχύτητας του αριστερού τροχού του οχήματος

*: Το* μέτρο της ταχύτητας του δεξιού τροχού του οχήματος

Για αρχική θέση:

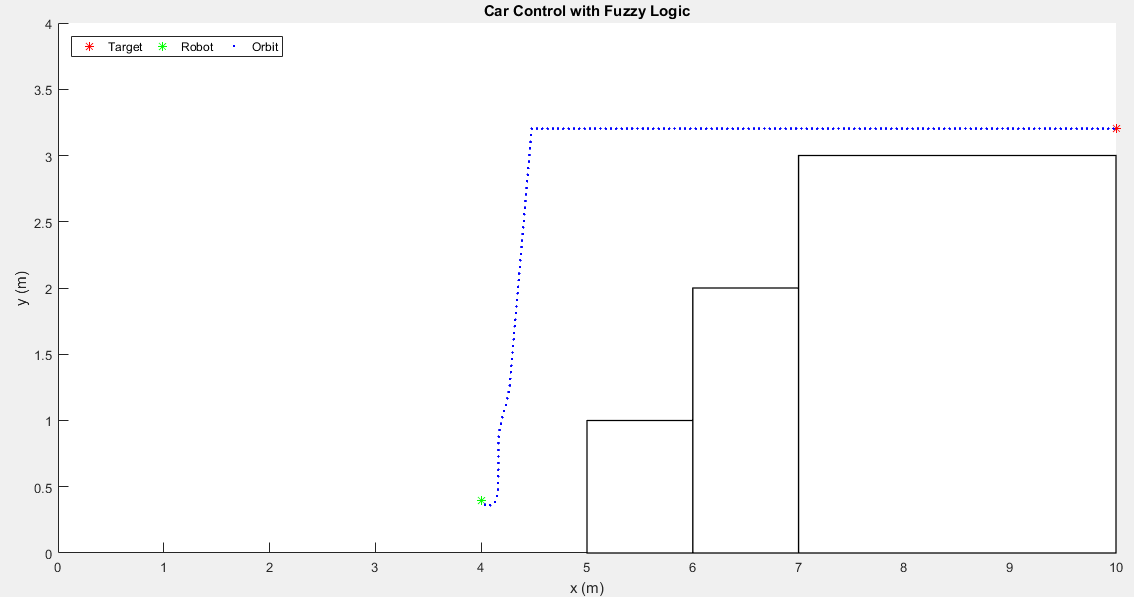
και με τη χρήση του κώδικα που παρουσιάζεται στο ΠΑΡΑΡΤΗΜΑ Α, έχουμε την κίνηση του οχήματος που φαίνεται στην εικόνα 7.



**Εικόνα 7:** Η κίνηση του οχήματος με τη χρήση του FCL.

Για αρχική θέση:

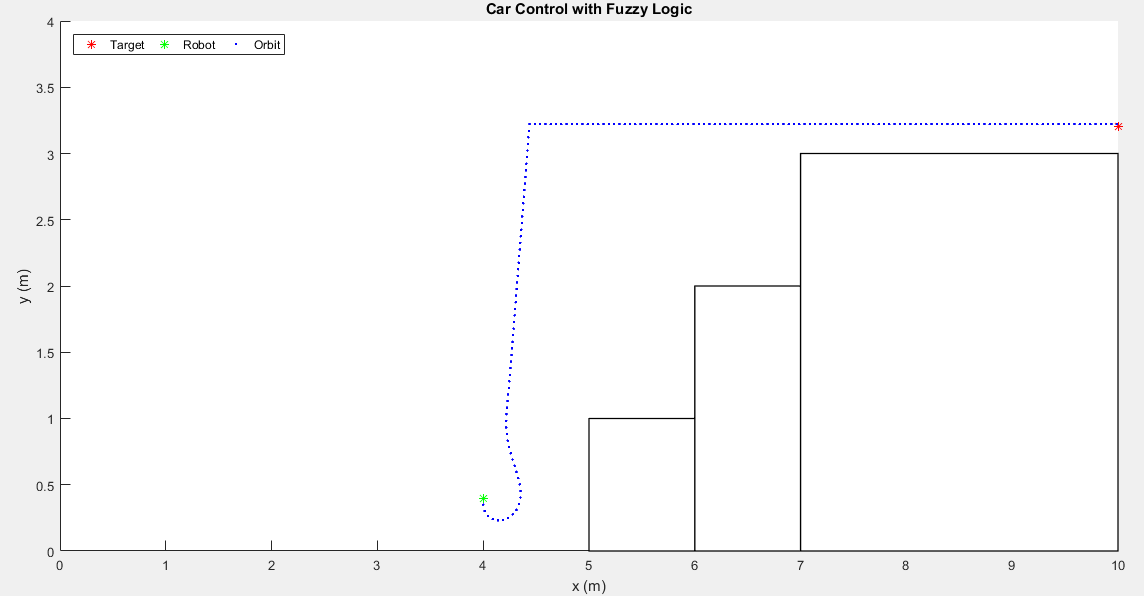
και με τη χρήση του κώδικα που παρουσιάζεται στο ΠΑΡΑΡΤΗΜΑ B, έχουμε την κίνηση του οχήματος που φαίνεται στην εικόνα 8.



**Εικόνα 8:** Η κίνηση του οχήματος με τη χρήση του FCL.

Για αρχική θέση:

και με τη χρήση του κώδικα που παρουσιάζεται στο ΠΑΡΑΡΤΗΜΑ Γ, έχουμε την κίνηση του οχήματος που φαίνεται στην εικόνα 9.



**Εικόνα 9:** Η κίνηση του οχήματος με τη χρήση του FCL.

**ΠΑΡΑΡΤΗΜΑ Α: Ο ΚΩΔΙΚΑΣ ΣΕ MATLAB ΓΙΑ ΑΡΧΙΚΗ ΘΕΣΗ ΤΟΥ ΟΧΗΜΑΤΟΣ**

|  |
| --- |
| ΚΩΔΙΚΑΣ MATLAB |
| test = readfis('Car\_Control.fis');  rectangle('Position',[5,0,1,1],'FaceColor',[1 1 1],'EdgeColor','k','LineWidth',1)  rectangle('Position',[6,0,1,2],'FaceColor',[1 1 1],'EdgeColor','k','LineWidth',1)  rectangle('Position',[7,0,3,3],'FaceColor',[1 1 1],'EdgeColor','k','LineWidth',1)  hold on  plot(10,3.2,'r\*','DisplayName','Target');  plot(4,0.4,'g\*','DisplayName','Robot');  % Use 10mx4m simulation environment.  axis([0 10 0 4]);  xinit = 4;  yinit = 0.4;  theta = 0;  x(1) = xinit;  y(1) = yinit;  dH(1) = xinit;  dV(1) = yinit;  Theta(1) = theta;  for i=1:250  if ((5-dH(i)> 1) && yinit < 1)  dH(i+1) = dH(i) + 0.05\*cos(0);  dV(i+1) = dV(i) + 0.05\*sin(0);  Theta(i+1) = Theta(1);  x(i+1) = dH(i+1);  y(i+1) = dV(i+1);  elseif ((5-dH(i)<= 1) && (5-dH(i) >= 0)) && (yinit < 1)  dH(i+1) = dH(i) + 0.05\*cos(Theta(i)\*pi/180);  dV(i+1) = dV(i) + 0.05\*sin(Theta(i)\*pi/180);  DeltaTheta(i+1) = evalfis([5-dH(i) dV(i) Theta(i)], test);  Theta(i+1) = DeltaTheta(i+1) + Theta(i);  x(i+1) = dH(i+1);  y(i+1) = dV(i+1);  end  if ((6-dH(i)<= 1) && (6-dH(i) >= 0)) && ((2 - dV(i) <= 1) && (2 - dV(i) >= 0))  dH(i+1) = dH(i) + 0.05\*cos(Theta(i)\*pi/180);  dV(i+1) = dV(i) + 0.05\*sin(Theta(i)\*pi/180);  DeltaTheta(i+1) = evalfis([5-dH(i) dV(i) Theta(i)], test);  Theta(i+1) = DeltaTheta(i+1) + Theta(i);  x(i+1) = dH(i+1);  y(i+1) = dV(i+1);  end  if ((6-dH(i)<= 1) && (6-dH(i) >= 0)) && ((3 - dV(i) <= 1) && (3 - dV(i) >= 0))  dH(i+1) = dH(i) + 0.05\*cos(Theta(i)\*pi/180);  dV(i+1) = dV(i) + 0.05\*sin(Theta(i)\*pi/180);  DeltaTheta(i+1) = evalfis([5-dH(i) dV(i) Theta(i)], test);  Theta(i+1) = DeltaTheta(i+1) + Theta(i);  x(i+1) = dH(i+1);  y(i+1) = dV(i+1);  end  if ((6-dH(i)<= 1) && (6-dH(i) >= 0)) && ((3 - dV(i) <= 0))  dH(i+1) = dH(i) + 0.05\*cos(Theta(i)\*pi/180);  dV(i+1) = dV(i) + 0.05\*sin(Theta(i)\*pi/180);  DeltaTheta(i+1) = evalfis([5-dH(i) dV(i) Theta(i)], test);  Theta(i+1) = DeltaTheta(i+1) + Theta(i);  x(i+1) = dH(i+1);  y(i+1) = dV(i+1);  end  if (dV(i) > 3.2 && dV(i) <= 3.3)  dH(i+1) = dH(i) + 0.05\*cos(0);  dV(i+1) = dV(i) + 0.05\*sin(0);  Theta(i+1) = Theta(1);  x(i+1) = dH(i+1);  y(i+1) = dV(i+1);  end  if (dH(i) > 9.96 && dV(i) <3.3)  return;  end  hold on  plot(x(i+1),y(i+1),'b.','DisplayName','Orbit');  end  % Add labels, title, and legends.  xlabel('x (m)'),ylabel('y (m)')  title('Car Control with Fuzzy Logic')  hold off  legend({'Target','Robot','Orbit'},'Location','northwest','Orientation','horizontal') |

**ΠΑΡΑΡΤΗΜΑ B: Ο ΚΩΔΙΚΑΣ ΣΕ MATLAB ΓΙΑ ΑΡΧΙΚΗ ΘΕΣΗ ΤΟΥ ΟΧΗΜΑΤΟΣ**

|  |
| --- |
| ΚΩΔΙΚΑΣ MATLAB |
| test = readfis('Car\_Control.fis');  rectangle('Position',[5,0,1,1],'FaceColor',[1 1 1],'EdgeColor','k','LineWidth',1)  rectangle('Position',[6,0,1,2],'FaceColor',[1 1 1],'EdgeColor','k','LineWidth',1)  rectangle('Position',[7,0,3,3],'FaceColor',[1 1 1],'EdgeColor','k','LineWidth',1)  hold on  plot(10,3.2,'r\*','DisplayName','Target');  plot(4,0.4,'g\*','DisplayName','Robot');  % Use 10mx4m simulation environment.  axis([0 10 0 4]);  xinit = 4;  yinit = 0.4;  theta = -45;  x(1) = xinit;  y(1) = yinit;  dH(1) = xinit;  dV(1) = yinit;  Theta(1) = theta;  for i=1:250  if ((5-dH(i)> 1) && yinit < 1)  dH(i+1) = dH(i) + 0.05\*cos(0);  dV(i+1) = dV(i) + 0.05\*sin(0);  Theta(i+1) = Theta(1);  x(i+1) = dH(i+1);  y(i+1) = dV(i+1);  elseif ((5-dH(i)<= 1) && (5-dH(i) >= 0)) && (yinit < 1)  dH(i+1) = dH(i) + 0.05\*cos(Theta(i)\*pi/180);  dV(i+1) = dV(i) + 0.05\*sin(Theta(i)\*pi/180);  DeltaTheta(i+1) = evalfis([5-dH(i) dV(i) Theta(i)], test);  Theta(i+1) = DeltaTheta(i+1) + Theta(i);  x(i+1) = dH(i+1);  y(i+1) = dV(i+1);  end  if ((6-dH(i)<= 1) && (6-dH(i) >= 0)) && ((2 - dV(i) <= 1) && (2 - dV(i) >= 0))  dH(i+1) = dH(i) + 0.05\*cos(Theta(i)\*pi/180);  dV(i+1) = dV(i) + 0.05\*sin(Theta(i)\*pi/180);  DeltaTheta(i+1) = evalfis([5-dH(i) dV(i) Theta(i)], test);  Theta(i+1) = DeltaTheta(i+1) + Theta(i);  x(i+1) = dH(i+1);  y(i+1) = dV(i+1);  end  if ((6-dH(i)<= 1) && (6-dH(i) >= 0)) && ((3 - dV(i) <= 1) && (3 - dV(i) >= 0))  dH(i+1) = dH(i) + 0.05\*cos(Theta(i)\*pi/180);  dV(i+1) = dV(i) + 0.05\*sin(Theta(i)\*pi/180);  DeltaTheta(i+1) = evalfis([5-dH(i) dV(i) Theta(i)], test);  Theta(i+1) = DeltaTheta(i+1) + Theta(i);  x(i+1) = dH(i+1);  y(i+1) = dV(i+1);  end  if ((6-dH(i)<= 1) && (6-dH(i) >= 0)) && ((3 - dV(i) <= 0))  dH(i+1) = dH(i) + 0.05\*cos(Theta(i)\*pi/180);  dV(i+1) = dV(i) + 0.05\*sin(Theta(i)\*pi/180);  DeltaTheta(i+1) = evalfis([5-dH(i) dV(i) Theta(i)], test);  Theta(i+1) = DeltaTheta(i+1) + Theta(i);  x(i+1) = dH(i+1);  y(i+1) = dV(i+1);  end  if (dV(i) > 3.2 && dV(i) <= 3.3)  dH(i+1) = dH(i) + 0.05\*cos(0);  dV(i+1) = dV(i) + 0.05\*sin(0);  Theta(i+1) = Theta(1);  x(i+1) = dH(i+1);  y(i+1) = dV(i+1);  end  if (dH(i) > 9.96 && dV(i) <3.3)  return;  end  hold on  plot(x(i+1),y(i+1),'b.','DisplayName','Orbit');  end  % Add labels, title, and legends.  xlabel('x (m)'),ylabel('y (m)')  title('Car Control with Fuzzy Logic')  hold off  legend({'Target','Robot','Orbit'},'Location','northwest','Orientation','horizontal') |

**ΠΑΡΑΡΤΗΜΑ Γ: Ο ΚΩΔΙΚΑΣ ΣΕ MATLAB ΓΙΑ ΑΡΧΙΚΗ ΘΕΣΗ ΤΟΥ ΟΧΗΜΑΤΟΣ**

|  |
| --- |
| ΚΩΔΙΚΑΣ MATLAB |
| test = readfis('Car\_Control.fis');  rectangle('Position',[5,0,1,1],'FaceColor',[1 1 1],'EdgeColor','k','LineWidth',1)  rectangle('Position',[6,0,1,2],'FaceColor',[1 1 1],'EdgeColor','k','LineWidth',1)  rectangle('Position',[7,0,3,3],'FaceColor',[1 1 1],'EdgeColor','k','LineWidth',1)  hold on  plot(10,3.2,'r\*','DisplayName','Target');  plot(4,0.4,'g\*','DisplayName','Robot');  % Use 10mx4m simulation environment.  axis([0 10 0 4]);  xinit = 4;  yinit = 0.4;  theta = -90;  x(1) = xinit;  y(1) = yinit;  dH(1) = xinit;  dV(1) = yinit;  Theta(1) = theta;  for i=1:250  if ((5-dH(i)> 1) && yinit < 1)  dH(i+1) = dH(i) + 0.05\*cos(0);  dV(i+1) = dV(i) + 0.05\*sin(0);  Theta(i+1) = Theta(1);  x(i+1) = dH(i+1);  y(i+1) = dV(i+1);  elseif ((5-dH(i)<= 1) && (5-dH(i) >= 0)) && (yinit < 1)  dH(i+1) = dH(i) + 0.05\*cos(Theta(i)\*pi/180);  dV(i+1) = dV(i) + 0.05\*sin(Theta(i)\*pi/180);  DeltaTheta(i+1) = evalfis([5-dH(i) dV(i) Theta(i)], test);  Theta(i+1) = DeltaTheta(i+1) + Theta(i);  x(i+1) = dH(i+1);  y(i+1) = dV(i+1);  end  if ((6-dH(i)<= 1) && (6-dH(i) >= 0)) && ((2 - dV(i) <= 1) && (2 - dV(i) >= 0))  dH(i+1) = dH(i) + 0.05\*cos(Theta(i)\*pi/180);  dV(i+1) = dV(i) + 0.05\*sin(Theta(i)\*pi/180);  DeltaTheta(i+1) = evalfis([5-dH(i) dV(i) Theta(i)], test);  Theta(i+1) = DeltaTheta(i+1) + Theta(i);  x(i+1) = dH(i+1);  y(i+1) = dV(i+1);  end  if ((6-dH(i)<= 1) && (6-dH(i) >= 0)) && ((3 - dV(i) <= 1) && (3 - dV(i) >= 0))  dH(i+1) = dH(i) + 0.05\*cos(Theta(i)\*pi/180);  dV(i+1) = dV(i) + 0.05\*sin(Theta(i)\*pi/180);  DeltaTheta(i+1) = evalfis([5-dH(i) dV(i) Theta(i)], test);  Theta(i+1) = DeltaTheta(i+1) + Theta(i);  x(i+1) = dH(i+1);  y(i+1) = dV(i+1);  end  if ((6-dH(i)<= 1) && (6-dH(i) >= 0)) && ((3 - dV(i) <= 0))  dH(i+1) = dH(i) + 0.05\*cos(Theta(i)\*pi/180);  dV(i+1) = dV(i) + 0.05\*sin(Theta(i)\*pi/180);  DeltaTheta(i+1) = evalfis([5-dH(i) dV(i) Theta(i)], test);  Theta(i+1) = DeltaTheta(i+1) + Theta(i);  x(i+1) = dH(i+1);  y(i+1) = dV(i+1);  end  if (dV(i) > 3.2 && dV(i) <= 3.3)  dH(i+1) = dH(i) + 0.05\*cos(0);  dV(i+1) = dV(i) + 0.05\*sin(0);  Theta(i+1) = Theta(1);  x(i+1) = dH(i+1);  y(i+1) = dV(i+1);  end  if (dH(i) > 9.96 && dV(i) <3.3)  return;  end  hold on  plot(x(i+1),y(i+1),'b.','DisplayName','Orbit');  end  % Add labels, title, and legends.  xlabel('x (m)'),ylabel('y (m)')  title('Car Control with Fuzzy Logic')  hold off  legend({'Target','Robot','Orbit'},'Location','northwest','Orientation','horizontal') |