ANEXOS

ANEXO 1

```
%CONTROLADOR FUZZY PARA ESTACIONAMIENTO SIN OBSTACULOS
%-----------------------------------------------------

clear all; clc; close all;

PI = 3.141592;
Lc=5.638;  %longitud real de cabina (=3718+1920)
Lt=5.667; %longitud real del trailer (=5570+2017-1920)

xini = input('Introduce coordenada inicial  x [0 a 100]: '); %trailer
yini = input('Introduce coordenada inicial  y : '); %trailer - no se considera como
entrada al controlador en el archivo FIS
CTini = input('Introduce angulo entre la cabina y el trailer CT [-90 a 90]: '); %angulo
entre el trailer y la cabina
Tini = input('Introduce inclinacion inicial del trailer T [-90 a 270]: ');
xdeseado = input('Introducir coordenada final de x [50]: ');

x = xini;
y = yini;
CT = CTini;
T = Tini;
velca = 35;
dt = 0.06;
r = velca*dt; % = v.At  % avance del camion en una etapa

camionfuzzy1 = readfis('sharafimod');
countmax = 800;

for count = 1:countmax

    xnuevo = x + 50 - xdeseado;
    DxG = evalfis([xnuevo,T,CT],camionfuzzy1);
    DxG = DxG*3.5; %forzar salida
    if( DxG > 65 ) %65
       DxG = 65;
    end
    if( DxG < -65 ) %-65
       DxG = -65;
    end

xn(count,1) = xnuevo;
xx(count,1) = x;
yy(count,1) = y;
Ct(count,1) = CT;
TT(count,1) = T;
gg(count,1) = DxG;
velc(count,1) = velca;

A = r*cos(DxG*PI/180);
B = A*cos(CT*PI/180);

Trad = (T*PI/180) + asin((A*sin(CT*PI/180))/Lt);
CTrad = (CT*PI/180) - asin((r*sin(DxG*PI/180))/(Lc + Lt));
```

```
CT = CTrad*180/PI;
if( Trad > (3*PI/2) ) %Trad > 270 [-90 a ->]
   Trad = Trad - 2*PI; %Trad -90 a 0
end
if( Trad < (-PI/2) ) %Trad < -90 [270 a ->]
   Trad = Trad + 2*PI; %Trad [90 a 0]
end
if( CT > 90 )
   CT = 90;
end
if( CT < -90 )
   CT = -90;
end
T = Trad*180/PI;

x = x + B*cos(Trad);
y = y + B*sin(Trad);

t1x = x; %parte trasera del camion
t1y = y;
t2x = x - Lt*cos(T*PI/180); %parte delantera del camion
t2y = y - Lt*sin(T*PI/180);
c1x = t2x; %parte trasera de la cabina
c1y = t2y;
C = CT + T;
c2x = c1x - Lc*cos(C*PI/180); %parte delantera de la cabina
c2y = c1y - Lc*sin(C*PI/180);

tx(count,1) = x;
ty(count,1) = y;
cx(count,1) = t2x;
cy(count,1) = t2y;
cxx(count,1) = c2x;
cyy(count,1) = c2y;
end

figure(1);
plot(gg);
grid;
title('Angulo del timon [grados]');

figure(2);
plot(xx,yy,'.b');
hold on;
axis([-200 250 -150 200]); %AXIS([XMIN XMAX YMIN YMAX])
grid;
hold on;
plot(cxx,cyy,'xr');
hold on;
title('Trayectoria de la Cabina (Rojo) y Trayectoria del Trailer (Azul)');

figure(3);
plot(velc);
grid;
```

```
title('Velocidad del Camion (Km/h)');
```

ANEXO 2

```
%CONTROLADOR FUZZY PARA ESTACIONAMIENTO CON OBSTACULOS
%-----------------------------------------------------

clear all; clc; close all;

PI = 3.141592;
Lc=5.638;  %longitud real de cabina (=3.718+1.920)
Lt=5.667; %longitud real del trailer (=5.570+2.017-1.920)

xini = input('Introduce coordenada inicial  x [0 a 100]: '); %trailer
yini = input('Introduce coordenada inicial  y : '); %trailer - no se considera como
entrada al controlador en el archivo FIS
CTini = input('Introduce angulo entre la cabina y el trailer CT [-90 a 90]: '); %angulo
entre el trailer y la cabina
Tini = input('Introduce inclinacion inicial del trailer T [-90 a 270]: ');
xdeseado = input('Introducir coordenada final de x [50]: ');

x = xini;
y = yini;
CT = CTini;
T = Tini;
velci = 30; %velocidad inicial del camion = 30Km/h
ace = 0; %aceleracion inicial del camion
dt = 0.05; %0.07
r = (velci*dt) + (0.5*ace*(dt^2)); % = v.At

%Obstaculo 1:
xc1 = 30; yc1 = 20; r1 = 3; % centro y radio del obstaculo
n1 = 50; k1 = 0:n1; fi1 = 2*PI*k1/n1;
x1 = xc1 + r1*cos(fi1);
y1 = yc1 + r1*sin(fi1);
%Obstaculo 2:
xc2 = 80; yc2 = 100; r2 = 5; % centro y radio del obstaculo
n2 = 10; k2 = 0:n2; fi2 = 2*PI*k2/n2;
x2 = xc2 + r2*cos(fi2);
y2 = yc2 + r2*sin(fi2);

camionfuzzy1 = readfis('sharafimod');
camionfuzzy2 = readfis('sharafiobst');
countmax = 1400; %1000
countmax2 = 1000;
c1=0;
c2=0;

%CONTROLADOR 1
%-------------
for count = 1:countmax

%-------------------------------------------------------------------
%Controlador 2:
%----------------

%Determinando distancia del primer obstaculo
```

```
xd1 = xc1 - x;
yd1 = yc1 - y;
d1 = sqrt((xd1^2) + (yd1^2)) - r1;
%Determinando distancia del segundo obstaculo
xd2 = xc2 - x;
yd2 = yc2 - y;
d2 = sqrt((xd2^2) + (yd2^2)) - r2;

if ((d1>=-30 && d1<=30) || (d2>=-30 && d2<=30)) %distancia establecida en el
controlador 2
   for count2=1:countmax2
   %Optimizando entrada. No entra a controlador 2 si xdeseado esta cerca al
camion
      if ((xdeseado>x && x>xc1) || (xdeseado>x && x>xc2))
         break;
      elseif ((xdeseado<x && x<xc1) || (xdeseado<x && x<xc2))
         break;
      end
   %Determinando angulo del obstaculo 1
      if ((x>xc1) && (yc1>y))
         ang1 = -atand(xd1/yd1);
      elseif ((xc1>x) && (yc1>y))
         ang1 = abs(atand(xd1/yd1));
      end
      switch (y > yc1)
         case ((x>xc1) && (y>yc1))
            ang1 = -(180 - abs(atand(xd1/yd1)));
         case ((x<xc1) && (y>yc1))
            ang1 = 180 - abs(atand(xd1/yd1));
      end
   %Determinando angulo del obstaculo 2
      if ((x>xc2) && (yc2>y))
         ang2 = -atand(xd2/yd2);
      elseif ((xc2>x) && (yc2>y))
         ang2 = abs(atand(xd2/yd2));
      end
      switch (y > yc2)
         case ((x>xc2) && (y>yc2))
            ang2 = -(180 - abs(atand(xd2/yd2)));
         case ((x<xc2) && (y>yc2))
            ang2 = 180 - abs(atand(xd2/yd2));
      end
   %Llamando al controlador 2
      if (d1 < d2) %d1 y d2 siempre son positivos
         dobs = d1;
         angobs = ang1;
      else
         dobs = d2;
         angobs = ang2;
      end
      velobs = 0;
      DxGyvelca = evalfis([dobs,angobs,velobs],camionfuzzy2);
      DxG = DxGyvelca(1,1);
      velcf = DxGyvelca(1,2);
```

```
    ace = (velcf - velci)/dt;
    DxG = DxG*2.5; % 3.5 forzar salida

    if( DxG > 65 ) %65
       DxG = 65;
    end
    if( DxG < -65 ) %-65
       DxG = -65;
    end

dt = 0.07;
r = (velci*dt) + (0.5*ace*(dt^2));
A = r*cos(DxG*PI/180);
B = A*cos(CT*PI/180);

Trad = (T*PI/180) + asin((A*sin(CT*PI/180))/Lt);
CTrad = (CT*PI/180) - asin((r*sin(DxG*PI/180))/(Lc + Lt));
CT = CTrad*180/PI;

    if( Trad > (3*PI/2) ) %Trad > 270 [-90 a ->]
       Trad = Trad - 2*PI; %Trad -90 a 0
    end
    if( Trad < (-PI/2) ) %Trad < -90 [270 a ->]
       Trad = Trad + 2*PI; %Trad [90 a 0]
    end
    if( CT > 90 )
       CT = 90;
    end
    if( CT < -90 )
       CT = -90;
    end
       T = Trad*180/PI;

x = x + B*cos(Trad);
y = y + B*sin(Trad);

t1x = x; %parte trasera del camion
t1y = y;
t2x = x - Lt*cos(T*PI/180); %parte delantera del camion
t2y = y - Lt*sin(T*PI/180);
c1x = t2x; %parte trasera de la cabina
c1y = t2y;
C = CT + T;
c2x = c1x - Lc*cos(C*PI/180); %parte delantera de la cabina
c2y = c1y - Lc*sin(C*PI/180);

angob(count2,1) = angobs;
dob(count2,1) = dobs;
xx2(count2,1) = x;
yy2(count2,1) = y;
Ct2(count2,1) = CT;
TT2(count2,1) = T;
gg2(count2,1) = DxG;
velc2(count2,1) = velci;
```

```
    tx2(count2,1) = x;
    ty2(count2,1) = y;
    cx2(count2,1) = t2x;
    cy2(count2,1) = t2y;
    cxx2(count2,1) = c2x;
    cyy2(count2,1) = c2y;

    %Determinando distancia del primer obstaculo (chancando valores)
    xd1 = xc1 - x;
    yd1 = yc1 - y;
    d1 = sqrt((xd1^2) + (yd1^2)) - r1;
    %Determinando distancia del segundo obstaculo (chancando valores)
    xd2 = xc2 - x;
    yd2 = yc2 - y;
    d2 = sqrt((xd2^2) + (yd2^2)) - r2;

    c1=count;
    c2=count+count2;
    velci = velcf;

        if ((d1<-30 || d1>30) && (d2<-30 || d2>30)) %si d1 y d2 estan fuera de rango se
sale del bucle for
            break;
        end

    end
end

    xnuevo = x + 50 - xdeseado;
    DxG = evalfis([xnuevo,T,CT],camionfuzzy1);
    DxG = DxG*2.5; % 3.5 forzar salida
    ace = 0;
    velci = 30; %velocidad de estacionamiento = 30Km/h
    if( DxG > 65 ) %65
        DxG = 65;
    end
    if( DxG < -65 ) %-65
        DxG = -65;
    end

dt = 0.07;
r = velci*dt; % = v.At
A = r*cos(DxG*PI/180);
B = A*cos(CT*PI/180);

Trad = (T*PI/180) + asin((A*sin(CT*PI/180))/Lt);
CTrad = (CT*PI/180) - asin((r*sin(DxG*PI/180))/(Lc + Lt));
CT = CTrad*180/PI;

if( Trad > (3*PI/2) ) %Trad > 270 [-90 a ->]
   Trad = Trad - 2*PI; %Trad -90 a 0
end
if( Trad < (-PI/2) ) %Trad < -90 [270 a ->]
```

```
   Trad = Trad + 2*PI; %Trad [90 a 0]
end
if( CT > 90 )
   CT = 90;
end
if( CT < -90 )
   CT = -90;
end
T = Trad*180/PI;

xn(count,1) = xnuevo;
xx(count,1) = x;
yy(count,1) = y;
Ct(count,1) = CT;
TT(count,1) = T;
gg(count,1) = DxG;
velc(count,1) = velci;

xxori = xx;
yyori = yy;
Ctori = Ct;
TTori = TT;
ggori = gg; %para aumentar valores originales de gg que fueron sustituidos por gg2
velcori = velc; %para aumentar valores originales de velc que fueron sustituidos por
velc2

x = x + B*cos(Trad);
y = y + B*sin(Trad);

t1x = x; %parte trasera del camion
t1y = y;
t2x = x - Lt*cos(T*PI/180); %parte delantera del camion
t2y = y - Lt*sin(T*PI/180);
c1x = t2x; %parte trasera de la cabina
c1y = t2y;
C = CT + T;
c2x = c1x - Lc*cos(C*PI/180); %parte delantera de la cabina
c2y = c1y - Lc*sin(C*PI/180);

tx(count,1) = x;
ty(count,1) = y;
cx(count,1) = t2x;
cy(count,1) = t2y;
cxx(count,1) = c2x;
cyy(count,1) = c2y;

if ( y > 250)
  break;
end
end

if (c1~=0 && c2~=0)
    xx((c1:c2-1),:)=xx2;
    xxori2 = xxori((c1:length(xxori)),:);
```

```
    xx = [xx;xxori2];
    yy((c1:c2-1),:)=yy2;
    yyori2 = yyori((c1:length(yyori)),:);
    yy = [yy;yyori2];

    Ct((c1:c2-1),:)=Ct2;
    Ctori2 = Ctori((c1:length(Ctori)),:);
    Ct = [Ct;Ctori2];

    TT((c1:c2-1),:)=TT2;
    TTori2 = TTori((c1:length(TTori)),:);
    TT = [TT;TTori2];

    gg((c1:c2-1),:)=gg2;
    ggori2 = ggori((c1:length(ggori)),:);
    gg = [gg;ggori2];

    velc((c1:c2-1),:)=velc2;
    velcori2 = velcori((c1:length(velcori)),:);
    velc = [velc;velcori2];
end

figure(1);
plot(gg);
hold on;
grid;
title('Angulo del timon [grados]');

figure(2);
if (c1~=0 && c2~=0)
    plot(xx2,yy2,'.b');
    hold on;
    plot(cxx2,cyy2,'xr');
    hold on;
end

plot(xx,yy,'.b');
hold on;
axis([-200 250 -150 200]); %AXIS([XMIN XMAX YMIN YMAX])
grid;
hold on;
plot(cxx,cyy,'xr');
hold on;
plot(xc1,yc1,'x',x1,y1,'-p'); %grafica de obstaculo 1
hold on;
plot(xc2,yc2,'x',x2,y2,'-p'); %grafica de obstaculo 2
hold on;
title('Trayectoria de la Cabina (Rojo) y Trayectoria del Trailer (Azul)');

figure(3);
plot(velc);
hold on;
grid;
title('Velocidad del Camion (Km/h)');
```

ANEXO 3

```
%CONTROLADOR NEUROFUZZY PARA ESTACIONAMIENTO SIN
OBSTACULOS
%-------------------------------------------------------

clear all; clc; close all;

PI = 3.141592;
Lc=5.638;  %longitud real de cabina (=3718+1920)
Lt=5.667; %longitud real del trailer (=5570+2017-1920)

xini = input('Introduce coordenada inicial  x [0 a 100]: '); %trailer
yini = input('Introduce coordenada inicial  y : '); %trailer - no se considera como
entrada al controlador en el archivo FIS
CTini = input('Introduce angulo entre la cabina y el trailer CT [-90 a 90]: '); %angulo
entre el trailer y la cabina
Tini = input('Introduce inclinacion inicial del trailer T [-90 a 270]: ');
xdeseado = input('Introducir coordenada final de x [50]: ');

x = xini;
y = yini;
CT = CTini;
T = Tini;
velca = 35;
dt = 0.06;
r = velca*dt; % = v.At

camionfuzzy1 = readfis('sharafimod');
countmax = 800;

for count = 1:countmax
    xnuevo = x + 50 - xdeseado;
    DxG = evalfis([xnuevo,T,CT],camionfuzzy1);
    DxG = DxG*3.5; %forzar salida
    if( DxG > 65 ) %65
        DxG = 65;
    end
    if( DxG < -65 ) %-65
        DxG = -65;
    end

xn(count,1) = xnuevo;
xx(count,1) = x;
yy(count,1) = y;
Ct(count,1) = CT;
TT(count,1) = T;
gg(count,1) = DxG;
velc(count,1) = velca;

A = r*cos(DxG*PI/180);
B = A*cos(CT*PI/180);
Trad = (T*PI/180) + asin((A*sin(CT*PI/180))/Lt);
CTrad = (CT*PI/180) - asin((r*sin(DxG*PI/180))/(Lc + Lt));
CT = CTrad*180/PI;
```

```
if( Trad > (3*PI/2) ) %Trad > 270 [-90 a ->]
   Trad = Trad - 2*PI; %Trad -90 a 0
end
if( Trad < (-PI/2) ) %Trad < -90 [270 a ->]
   Trad = Trad + 2*PI; %Trad [90 a 0]
end
if( CT > 90 )
   CT = 90;
end
if( CT < -90 )
   CT = -90;
end
T = Trad*180/PI;

x = x + B*cos(Trad);
y = y + B*sin(Trad);

t1x = x; %parte trasera del camion
t1y = y;
t2x = x - Lt*cos(T*PI/180); %parte delantera del camion
t2y = y - Lt*sin(T*PI/180);
c1x = t2x; %parte trasera de la cabina
c1y = t2y;
C = CT + T;
c2x = c1x - Lc*cos(C*PI/180); %parte delantera de la cabina
c2y = c1y - Lc*sin(C*PI/180);

tx(count,1) = x;
ty(count,1) = y;
cx(count,1) = t2x;
cy(count,1) = t2y;
cxx(count,1) = c2x;
cyy(count,1) = c2y;

end

figure(1);
plot(gg);
grid;
title('Angulo del timon [grados]');

figure(2);
plot(xx,yy,'.b');
axis([-200 250 -150 200]); %AXIS([XMIN XMAX YMIN YMAX])
grid;
hold on;
plot(cxx,cyy,'xr');
hold on;
title('Trayectoria de la Cabina (Rojo) y Trayectoria del Trailer (Azul)');

figure(3);
plot(velc);
grid;
```

```
title('Velocidad del Camion (Km/h)');
```

ANEXO 4

```
%CONTROLADOR NEUROFUZZY PARA ESTACIONAMIENTO CON
OBSTACULOS
%-------------------------------------------------------

clear all; clc; close all;

PI = 3.141592;
Lc=5.638;  %longitud real de cabina (=3.718+1.920)
Lt=5.667; %longitud real del trailer (=5.570+2.017-1.920)

xini = input('Introduce coordenada inicial  x [0 a 100]: '); %trailer
yini = input('Introduce coordenada inicial  y : '); %trailer - no se considera como
entrada al controlador en el archivo FIS
CTini = input('Introduce angulo entre la cabina y el trailer CT [-90 a 90]: '); %angulo
entre el trailer y la cabina
Tini = input('Introduce inclinacion inicial del trailer T [-90 a 270]: ');
xdeseado = input('Introducir coordenada final de x [50]: ');

x = xini;
y = yini;
CT = CTini;
T = Tini;
velci = 30; %velocidad inicial del camion = 30Km/h
ace = 0; %aceleracion inicial del camion
dt = 0.05; %0.07
r = (velci*dt) + (0.5*ace*(dt^2)); % = v.At

%Obstaculo 1:
xc1 = 30; yc1 = 20; r1 = 3; % centro y radio del obstaculo
n1 = 50; k1 = 0:n1; fi1 = 2*PI*k1/n1;
x1 = xc1 + r1*cos(fi1);
y1 = yc1 + r1*sin(fi1);
%Obstaculo 2:
xc2 = 80; yc2 = 100; r2 = 5; % centro y radio del obstaculo
n2 = 10; k2 = 0:n2; fi2 = 2*PI*k2/n2;
x2 = xc2 + r2*cos(fi2);
y2 = yc2 + r2*sin(fi2);

camionfuzzy1 = readfis('sharafimod');
camionfuzzy2 = readfis('sharafiobst');
countmax = 1000; %1000
countmax2 = 400;
c1=0;
c2=0;

%CONTROLADOR 1
%-------------
for count = 1:countmax

%-------------------------------------------------------------------
%Controlador 2:
%----------------
```

```
%Determinando distancia del primer obstaculo
xd1 = xc1 - x;
yd1 = yc1 - y;
d1 = sqrt((xd1^2) + (yd1^2)) - r1;
%Determinando distancia del segundo obstaculo
xd2 = xc2 - x;
yd2 = yc2 - y;
d2 = sqrt((xd2^2) + (yd2^2)) - r2;

if ((d1>=-30 && d1<=30) || (d2>=-30 && d2<=30)) %distancia establecida en el
controlador 2
   for count2=1:countmax2
   %Optimizando entrada. No entra a controlador 2 si xdeseado esta cerca al
camion
      if ((xdeseado>x && x>xc1) || (xdeseado>x && x>xc2))
         break;
      elseif ((xdeseado<x && x<xc1) || (xdeseado<x && x<xc2))
         break;
      end
   %Determinando angulo del obstaculo 1
      if ((x>xc1) && (yc1>y))
         ang1 = -atand(xd1/yd1);
      elseif ((xc1>x) && (yc1>y))
         ang1 = abs(atand(xd1/yd1));
      end
      switch (y > yc1)
         case ((x>xc1) && (y>yc1))
            ang1 = -(180 - abs(atand(xd1/yd1)));
         case ((x<xc1) && (y>yc1))
            ang1 = 180 - abs(atand(xd1/yd1));
      end
   %Determinando angulo del obstaculo 2
      if ((x>xc2) && (yc2>y))
         ang2 = -atand(xd2/yd2);
      elseif ((xc2>x) && (yc2>y))
         ang2 = abs(atand(xd2/yd2));
      end
      switch (y > yc2)
         case ((x>xc2) && (y>yc2))
            ang2 = -(180 - abs(atand(xd2/yd2)));
         case ((x<xc2) && (y>yc2))
            ang2 = 180 - abs(atand(xd2/yd2));
      end
   %Llamando al controlador 2
      if (d1 < d2) %d1 y d2 siempre son positivos
         dobs = d1;
         angobs = ang1;
      else
         dobs = d2;
         angobs = ang2;
      end
      velobs = 0;
      DxGyvelca = evalfis([dobs,angobs,velobs],camionfuzzy2);
      DxG = DxGyvelca(1,1);
```

```
    velcf = DxGyvelca(1,2);
    ace = (velcf - velci)/dt;
    DxG = DxG*2.5; % 3.5 forzar salida

    if( DxG > 65 ) %65
       DxG = 65;
    end
    if( DxG < -65 ) %-65
       DxG = -65;
    end

dt = 0.07;
r = (velci*dt) + (0.5*ace*(dt^2));
A = r*cos(DxG*PI/180);
B = A*cos(CT*PI/180);

Trad = (T*PI/180) + asin((A*sin(CT*PI/180))/Lt);
CTrad = (CT*PI/180) - asin((r*sin(DxG*PI/180))/(Lc + Lt));
CT = CTrad*180/PI;

    if( Trad > (3*PI/2) ) %Trad > 270 [-90 a ->]
       Trad = Trad - 2*PI; %Trad -90 a 0
    end
    if( Trad < (-PI/2) ) %Trad < -90 [270 a ->]
       Trad = Trad + 2*PI; %Trad [90 a 0]
    end
    if( CT > 90 )
       CT = 90;
    end
    if( CT < -90 )
       CT = -90;
    end
       T = Trad*180/PI;

x = x + B*cos(Trad);
y = y + B*sin(Trad);

t1x = x; %parte trasera del camion
t1y = y;
t2x = x - Lt*cos(T*PI/180); %parte delantera del camion
t2y = y - Lt*sin(T*PI/180);
c1x = t2x; %parte trasera de la cabina
c1y = t2y;
C = CT + T;
c2x = c1x - Lc*cos(C*PI/180); %parte delantera de la cabina
c2y = c1y - Lc*sin(C*PI/180);

angob(count2,1) = angobs;
dob(count2,1) = dobs;
xx2(count2,1) = x;
yy2(count2,1) = y;
Ct2(count2,1) = CT;
TT2(count2,1) = T;
gg2(count2,1) = DxG;
```

```
    velc2(count2,1) = velci;

    tx2(count2,1) = x;
    ty2(count2,1) = y;
    cx2(count2,1) = t2x;
    cy2(count2,1) = t2y;
    cxx2(count2,1) = c2x;
    cyy2(count2,1) = c2y;

    %Determinando distancia del primer obstaculo (chancando valores)
    xd1 = xc1 - x;
    yd1 = yc1 - y;
    d1 = sqrt((xd1^2) + (yd1^2)) - r1;
    %Determinando distancia del segundo obstaculo (chancando valores)
    xd2 = xc2 - x;
    yd2 = yc2 - y;
    d2 = sqrt((xd2^2) + (yd2^2)) - r2;

    c1=count;
    c2=count+count2;
    velci = velcf;

        if ((d1<-30 || d1>30) && (d2<-30 || d2>30)) %si d1 y d2 estan fuera de rango se
sale del bucle for
            break;
        end
    end
end

    xnuevo = x + 50 - xdeseado;
    DxG = evalfis([xnuevo,T,CT],camionfuzzy1);
    DxG = DxG*2.5; % 3.5 forzar salida
    ace = 0;
    velci = 30; %velocidad de estacionamiento = 30Km/h
    if( DxG > 65 ) %65
        DxG = 65;
    end
    if( DxG < -65 ) %-65
        DxG = -65;
    end

dt = 0.07;
r = velci*dt; % = v.At
A = r*cos(DxG*PI/180);
B = A*cos(CT*PI/180);

Trad = (T*PI/180) + asin((A*sin(CT*PI/180))/Lt);
CTrad = (CT*PI/180) - asin((r*sin(DxG*PI/180))/(Lc + Lt));
CT = CTrad*180/PI;

if( Trad > (3*PI/2) ) %Trad > 270 [-90 a ->]
   Trad = Trad - 2*PI; %Trad -90 a 0
end
if( Trad < (-PI/2) ) %Trad < -90 [270 a ->]
```

```
   Trad = Trad + 2*PI; %Trad [90 a 0]
end
if( CT > 90 )
   CT = 90;
end
if( CT < -90 )
   CT = -90;
end
T = Trad*180/PI;

xn(count,1) = xnuevo;
xx(count,1) = x;
yy(count,1) = y;
Ct(count,1) = CT;
TT(count,1) = T;
gg(count,1) = DxG;
velc(count,1) = velci;

xxori = xx;
yyori = yy;
Ctori = Ct;
TTori = TT;
ggori = gg; %para aumentar valores originales de gg que fueron chancados por gg2
velcori = velc; %para aumentar valores originales de velc que fueron chancados por
velc2

x = x + B*cos(Trad);
y = y + B*sin(Trad);

t1x = x; %parte trasera del camion
t1y = y;
t2x = x - Lt*cos(T*PI/180); %parte delantera del camion
t2y = y - Lt*sin(T*PI/180);
c1x = t2x; %parte trasera de la cabina
c1y = t2y;
C = CT + T;
c2x = c1x - Lc*cos(C*PI/180); %parte delantera de la cabina
c2y = c1y - Lc*sin(C*PI/180);

tx(count,1) = x;
ty(count,1) = y;
cx(count,1) = t2x;
cy(count,1) = t2y;
cxx(count,1) = c2x;
cyy(count,1) = c2y;

if ( y > 250)
  break;
end
end

if (c1~=0 && c2~=0)
    xx((c1:c2-1),:)=xx2;
    xxori2 = xxori((c1:length(xxori)),:);
```

```
    xx = [xx;xxori2];
    yy((c1:c2-1),:)=yy2;
    yyori2 = yyori((c1:length(yyori)),:);
    yy = [yy;yyori2];

    Ct((c1:c2-1),:)=Ct2;
    Ctori2 = Ctori((c1:length(Ctori)),:);
    Ct = [Ct;Ctori2];

    TT((c1:c2-1),:)=TT2;
    TTori2 = TTori((c1:length(TTori)),:);
    TT = [TT;TTori2];

    gg((c1:c2-1),:)=gg2;
    ggori2 = ggori((c1:length(ggori)),:);
    gg = [gg;ggori2];

    velc((c1:c2-1),:)=velc2;
    velcori2 = velcori((c1:length(velcori)),:);
    velc = [velc;velcori2];
end

figure(1);
plot(gg);
hold on;
grid;
title('Angulo del timon [grados]');

figure(2);
if (c1~=0 && c2~=0)
    plot(xx2,yy2,'.b');
    hold on;
    plot(cxx2,cyy2,'xr');
    hold on;
end

plot(xx,yy,'.b');
hold on;
axis([-200 250 -150 200]); %AXIS([XMIN XMAX YMIN YMAX])
grid;
hold on;
plot(cxx,cyy,'xr');
hold on;
plot(xc1,yc1,'x',x1,y1,'-p'); %grafica de obstaculo 1
hold on;
plot(xc2,yc2,'x',x2,y2,'-p'); %grafica de obstaculo 2
hold on;
title('Trayectoria de la Cabina (Rojo) y Trayectoria del Trailer (Azul)');

figure(3);
plot(velc);
hold on;
grid;
title('Velocidad del Camion (Km/h)');
```

ANEXO 5

```
%CONTROLADOR NEUROFUZZY PARA SEGUIMIENTO DE TRAYECTORIA
%-----------------------------------------------------

clear all; clc; close all;

PI = 3.141592;
Lc=5.638;  %longitud real de cabina (=3718+1920)
Lt=5.667; %longitud real del trailer (=5570+2017-1920)

xini = input('Introduce coordenada inicial  x [0 a 100]: '); %trailer
yini = input('Introduce coordenada inicial  y : '); %trailer - no se considera como
entrada al controlador en el archivo FIS
CTini = input('Introduce angulo entre la cabina y el trailer CT [-90 a 90]: '); %angulo
entre el trailer y la cabina
Tini = input('Introduce inclinacion inicial del trailer T [-90 a 270]: ');
xdeseado = input('Introducir coordenada final de x [50]: ');

%Define cantidad de divisiones de la variables

kX = 5;
kT = 7;
kCT = 3;
kDxG = 7;

x = -50:0.07:150;
x = x';
cx1 = -11.5;
ax1 = 7.5;
fpx1 = 1.0./(1+exp((x-cx1)./ax1));
figure(1);
subplot(3,1,1);
hold on;
plot(x,fpx1,':b');

cx2 = 30;
ax2 = 8.492;
fpx2 = exp(-((x-cx2)./ax2).^2);
subplot(3,1,1);
plot(x,fpx2,':b');

cx3 = 50.0;
ax3 = 4.248;
fpx3 = exp(-((x-cx3)./ax3).^2);
subplot(3,1,1);
plot(x,fpx3,':b');

cx4 = 70.0;
ax4 = 8.492;
fpx4 = exp(-((x-cx4)./ax4).^2);
subplot(3,1,1);
plot(x,fpx4,':b');

cx5 = 107.0;
```

```
ax5 = 7.5;
fpx5 = 1.0./(1+exp(-(x-cx5)./ax5));
subplot(3,1,1);
plot(x,fpx5,':b');

%--------
tphi = -95:0.07:275;
tphi = tphi';
ct1 = -45.0;
at1 = 23.36;
fpt1 = exp(-((tphi-ct1)./at1).^2);
figure(1);
subplot(3,1,2);
hold on;
plot(tphi,fpt1,':b');

ct2 = 25.0;
at2 = 14.86;
fpt2 = exp(-((tphi-ct2)./at2).^2);
figure(1);
subplot(3,1,2);
hold on;
plot(tphi,fpt2,':b');

ct3 = 65.0;
at3 = 10.62;
fpt3 = exp(-((tphi-ct3)./at3).^2);
figure(1);
subplot(3,1,2);
hold on;
plot(tphi,fpt3,':b');

ct4 = 90.0;
at4 = 8.493;
fpt4 = exp(-((tphi-ct4)./at4).^2);
figure(1);
subplot(3,1,2);
hold on;
plot(tphi,fpt4,':b');

ct5 = 115.0;
at5 = 10.62;
fpt5 = exp(-((tphi-ct5)./at5).^2);
figure(1);
subplot(3,1,2);
hold on;
plot(tphi,fpt5,':b');

ct6 = 155.0;
at6 = 14.86;
fpt6 = exp(-((tphi-ct6)./at6).^2);
figure(1);
subplot(3,1,2);
hold on;
```

```
plot(tphi,fpt6,':b');

ct7 = 225.0;
at7 = 23.36;
fpt7 = exp(-((tphi-ct7)./at7).^2);
figure(1);
subplot(3,1,2);
hold on;
plot(tphi,fpt7,':b');

%--------
ctphi = -95:0.07:95;
ctphi = ctphi';
cct1 = -35.0; %42.0;
act1 = 10.0; %36.0;
fpct1 = 1.0./(1+exp((ctphi-cct1)./act1));
figure(1);
subplot(3,1,3);
hold on;
plot(ctphi,fpct1,':b');

cct2 = 0.0;
act2 = 3.397;
fpct2 = exp(-((ctphi-cct2)./act2).^2);
figure(1);
subplot(3,1,3);
hold on;
plot(ctphi,fpct2,':b');

cct3 = 35.0;
act3 = 10.0;
fpct3 = 1.0./(1+exp(-(ctphi-cct3)./act3));
figure(1);
subplot(3,1,3);
hold on;
plot(ctphi,fpct3,':b');

%--------------
%Base de Reglas CT = NE
BaseReg1 = [ 2  3  3  1  1  1  1
             1  3  3  1  1  1  1
             1  1  2  4  2  2  1
             1  1  1  2  2  2  1
             1  1  1  2  3  3  2];
%NB=-65;NM=-35;NS=-14;ZE=0;PS=14;PM=35;PB=65; 1 -> 7
%  1      2     3     4    5     6     7
BaseReg1 = [ -35.0  -14.0  -14.0  -65.0  -65.0  -65.0  -65.0
             -65.0  -14.0  -14.0  -65.0  -65.0  -65.0  -65.0
             -65.0  -65.0  -35.0    0.0  -35.0  -35.0  -65.0
             -65.0  -65.0  -65.0  -35.0  -35.0  -35.0  -65.0
             -65.0  -65.0  -65.0  -35.0  -14.0  -14.0  -35.0];

%matriz vertical de valores representativos de f
k = 1;
```

```
for i1 = 1:kX
  for j1 = 1:kT
     deltanf1(k,1) =  BaseReg1(i1,j1);     %
     k = k + 1;
  end
end

%--------------
%Base de Reglas CT = ZR
BaseReg2 = [ 2  3  4  6  6  7  7
             1  2  4  6  6  6  2
             1  1  2  4  6  6  7
             7  1  1  2  4  4  6
             7  1  1  2  4  4  6];
%NB=-65;NM=-35;NS=-14;ZE=0;PS=14;PM=35;PB=65; 1 -> 7
%  1      2      3     4    5     6     7
BaseReg2 = [ -35.0  -14.0    0.0   35.0  35.0  65.0   65.0
            -65.0  -35.0    0.0   35.0  35.0  35.0  -35.0
            -65.0  -65.0  -35.0    0.0  35.0  35.0   65.0
             65.0  -65.0  -65.0  -35.0   0.0   0.0   35.0
             65.0  -65.0  -65.0  -35.0   0.0   0.0   35.0];

%matriz vertical de valores representativos de f
k = 1;
for i2 = 1:kX
  for j2 = 1:kT
     deltanf2(k,1) =  BaseReg2(i2,j2);     %
     k = k + 1;
  end
end

%--------------
%Base de Reglas CT = PO
BaseReg3 = [ 7  5  5  6  7  7  7
             7  5  5  6  7  7  7
             7  7  6  4  6  7  7
             7  7  7  7  7  7  6
             7  7  7  7  5  5  7];
%NB=-65;NM=-35;NS=-14;ZE=0;PS=14;PM=35;PB=65; 1 -> 7
%  1      2      3     4    5     6     7
BaseReg3 = [ 65.0  14.0  14.0  35.0  65.0  65.0  65.0
             65.0  14.0  14.0  35.0  65.0  65.0  65.0
             65.0  65.0  35.0   0.0  35.0  65.0  65.0
             65.0  65.0  65.0  65.0  65.0  65.0  35.0
             65.0  65.0  65.0  65.0  14.0  14.0  65.0];

%matriz vertical de valores representativos de f
k = 1;
for i3 = 1:kX
  for j3 = 1:kT
     deltanf3(k,1) =  BaseReg3(i3,j3);     %
     k = k + 1;
  end
end
```

```
deltanf = [deltanf1;deltanf2;deltanf3];

x = xini;
y = yini;
CT = CTini;
T = Tini;
velca = 30;
dt = 0.07;
r = velca*dt; % = v.At  %3 avance del carro en una etapa

countmax = 400;

for count = 1:countmax

% %Para R1
% xdeseado = (5*x - 9*y + 60); %(5*x - 9*y + 60); %100 punto final de recta
% Tdeseado = 180 + atand(10/18); %angulo deseado de trayectoria
% CTdeseado = 0;

%       %Para C1
%       xdeseado = ((38.5-x)^2) + ((35-y)^2) - (6.1^2); %seguimiento de
circunferencia -- 30 radio circ, 100 centro circ
%       Tdeseado = 180/pi*abs(atan((38.5-x)/(35-y)));           %seguimiento de
circunferencia
%       CTdeseado = 0;
%
%       %Para R2
%       xdeseado = (x - y + 5); %la recta empieza en el xdeseado y termina en
45(en Y)
%       Tdeseado = 45;
%       CTdeseado = 0;
%
%       %Para R3
%       xdeseado = (8*x - 11*y + 175); %la recta empieza en el xdeseado y termina
en 45(en Y)
%       Tdeseado = atand(16/22);
%       CTdeseado = 0;
%
%       %Para R4
%       xdeseado = (672 - x - 10*y); %la recta empieza en el xdeseado y termina en
45(en Y)
%       Tdeseado = -atand(1/10);
%       CTdeseado = 0;
%
%       %Para C2
%       xdeseado = ((x-73.5)^2) + ((y-62.5)^2) - (2.92^2); %seguimiento de
circunferencia -- 30 radio circ, 100 centro circ
%       Tdeseado = 180/pi*abs(atan((x-73.5)/(y-62.5)));           %seguimiento de
circunferencia
%       CTdeseado = 0;
%
%       %Para R5
```

```
%       xdeseado = (335 - x - 4*y); %la recta empieza en el xdeseado y termina en
45(en Y)
%       Tdeseado = 90 + atand(20/5);
%       CTdeseado = 0;
%
%       %Para R6
%       xdeseado = (2*x - 5*y + 240); %la recta empieza en el xdeseado y termina
en 45(en Y)
%       Tdeseado = 180 + atand(10/25);
%       CTdeseado = 0;
%
%       %Para R7
%       xdeseado = (3*x - y - 30); %la recta empieza en el xdeseado y termina en
45(en Y)
%       Tdeseado = 180 + atand(30/10);
%       CTdeseado = 0;
%
%       %Para R8
%       xdeseado = (380 - 13*x - 4*y); %la recta empieza en el xdeseado y termina
en 45(en Y)
%       Tdeseado = -atand(13/4);
%       CTdeseado = 0;
%
%       %Para C3
%       xdeseado = ((x-19.5)^2) + ((y-16)^2) - (4.61^2);
%       Tdeseado = 180/pi*abs(atan((x-19.5)/(y-16)));
%       CTdeseado = 0;
%
%       %Para R9
%       xdeseado = (8*x - y - 105);
%       Tdeseado = atand(16/2);
%       CTdeseado = 0;
%
    %Para R10
    xdeseado = (36*x - 5*y - 457);
    Tdeseado = atand(36/5);
    CTdeseado = 0;
%
%       %Para R11
%       xdeseado = (15*x - 19*y + 943);
%       Tdeseado = atand(30/38);
%       CTdeseado = 0;
%
%       %Para R12
%       xdeseado = (3*x - 16*y + 1372);
%       Tdeseado = atand(6/32);
%       CTdeseado = 0;
%
%       %Para R13
%       xdeseado = (298 - x - 2*y);
%       Tdeseado = -atand(5/10);
%       CTdeseado = 0;
%
%       %Para C4
```

```
%       xdeseado = ((x-103)^2) + ((y-101)^2) - (3.16^2);
%       Tdeseado = 180/pi*abs(atan((x-103)/(y-101)));
%       CTdeseado = 0;
%
%       %Para R14
%       xdeseado = (1352 - 12*x - y); %la recta empieza en el xdeseado y termina
en 45(en Y)
%       Tdeseado = 90 + atand(1/12);
%       CTdeseado = 0;
%
%       %Para R15
%       xdeseado = (3*x - y - 193); %la recta empieza en el xdeseado y termina en
45(en Y)
%       Tdeseado = atand(6/2);
%       CTdeseado = 0;
%
%       %Para R16
%       xdeseado = (1975 - 13*x - 5*y); %la recta empieza en el xdeseado y termina
en 45(en Y)
%       Tdeseado = 90 + atand(5/13);
%       CTdeseado = 0;
%
%       %Para R17
%       xdeseado = (3*x - 2*y - 30); %la recta empieza en el xdeseado y termina en
45(en Y)
%       Tdeseado = atand(3/2);
%       CTdeseado = 0;

xnuevo = x + 50 - xdeseado;
Tnuevo = T + 90 - Tdeseado;
CTnuevo = CT + 0 - CTdeseado;

cx1 = -11.5;
ax1 = 7.5;
fpx(1,1) = 1.0./(1+exp((xnuevo-cx1)./ax1));
cx2 = 30;
ax2 = 8.492;
fpx(2,1) = exp(-((xnuevo-cx2)./ax2).^2);
cx3 = 50.0;
ax3 = 4.248;
fpx(3,1) = exp(-((xnuevo-cx3)./ax3).^2);
cx4 = 70.0;
ax4 = 8.492;
fpx(4,1) = exp(-((xnuevo-cx4)./ax4).^2);
cx5 = 107.0;
ax5 = 7.5;
fpx(5,1) = 1.0./(1+exp(-(xnuevo-cx5)./ax5));

ct1 = -45.0;
at1 = 23.36;
fpt(1,1) = exp(-((Tnuevo-ct1)./at1).^2);
ct2 = 25.0;
at2 = 14.86;
fpt(2,1) = exp(-((Tnuevo-ct2)./at2).^2);
```

```
ct3 = 65.0;
at3 = 10.62;
fpt(3,1) = exp(-((Tnuevo-ct3)./at3).^2);
ct4 = 90.0;
at4 = 8.493;
fpt(4,1) = exp(-((Tnuevo-ct4)./at4).^2);
ct5 = 115.0;
at5 = 10.62;
fpt(5,1) = exp(-((Tnuevo-ct5)./at5).^2);
ct6 = 155.0;
at6 = 14.86;
fpt(6,1) = exp(-((Tnuevo-ct6)./at6).^2);
ct7 = 225.0;
at7 = 23.36;
fpt(7,1) = exp(-((Tnuevo-ct7)./at7).^2);

cct1 = -42.0;
act1 = 36.0;
fpct(1,1) = exp(-((CTnuevo-cct1)./act1).^2);
cct2 = 0.0;
act2 = 3.397;
fpct(2,1) = exp(-((CTnuevo-cct2)./act2).^2);
cct3 = 38.0;
act3 = 36.0;
fpct(3,1) = exp(-((CTnuevo-cct3)./act3).^2);

k = 1;
   for n1 = 1:kCT %se obtienen las salidas de reglas
      for i4 = 1:kX
         for j4 = 1:kT
            fpxT(k,1) =  fpx(i4,1) * fpt(j4,1) * fpct(n1,1);     %
            k = k + 1;
         end
      end
   end

sumfpxT = sum(fpxT);
fpxT = fpxT./sumfpxT;
%fpxP = 5*fpxP;
DxG = deltanf'*fpxT;
DxG = DxG*2.5;
velca = 30; %velocidad del camion 35Km/h

if( DxG > 65 ) %65
   DxG = 65;
end
if( DxG < -65 ) %-65
   DxG = -65;
end

A = r*cos(DxG*PI/180);
B = A*cos(CT*PI/180);

Trad = (T*PI/180) + asin((A*sin(CT*PI/180))/Lt);
```

```
CTrad = (CT*PI/180) - asin((r*sin(DxG*PI/180))/(Lc + Lt));
CT = CTrad*180/PI;
if( Trad > (3*PI/2) ) %Trad > 270 [-90 a ->]
   Trad = Trad - 2*PI; %Trad -90 a 0
end
if( Trad < (-PI/2) ) %Trad < -90 [270 a ->]
   Trad = Trad + 2*PI; %Trad [90 a 0]
end
if( CT > 90 )
   CT = 90;
end
if( CT < -90 )
   CT = -90;
end
T = Trad*180/PI;

xn(count,1) = xnuevo;
xx(count,1) = x;
yy(count,1) = y;
Ct(count,1) = CT;
TT(count,1) = T;
gg(count,1) = DxG;
velc(count,1) = velca;

dt = 0.07;
r = velca*dt; % = v.At

x = x + B*cos(Trad);
y = y + B*sin(Trad);

t1x = x; %parte trasera del camion
t1y = y;
t2x = x - Lt*cos(T*PI/180); %parte delantera del camion
t2y = y - Lt*sin(T*PI/180);
c1x = t2x; %parte trasera de la cabina
c1y = t2y;
C = CT + T;
c2x = c1x - Lc*cos(C*PI/180); %parte delantera de la cabina
c2y = c1y - Lc*sin(C*PI/180);

tx(count,1) = x;
ty(count,1) = y;
cx(count,1) = t2x;
cy(count,1) = t2y;
cxx(count,1) = c2x;
cyy(count,1) = c2y;

if ( y > 250)
  break;
end
end

figure(2);
plot(gg);
```

```
grid;
title('Angulo del timon [grados]');

figure(3);
plot(xx,yy,'.b');
hold on;
axis([-200 250 -150 200]); %AXIS([XMIN XMAX YMIN YMAX])
grid;
hold on;
plot(cxx,cyy,'xr');
hold on;
title('Trayectoria de la Cabina (Rojo) y Trayectoria del Trailer (Azul)');

figure(4);
plot(velc);
grid;
title('Velocidad del Camion (Km/h)');
```

ANEXO 6