

Line Follower Simulator with Fuzzy Control

Víctor Proaño

*Departamento de Eléctrica y Electrónica
Universidad de las Fuerzas Armadas ESPE
P.O.BOX:17-5-231B, Sangolquí, Ecuador
Email: vgproano@espe.edu.ec*

Marco Gualsaquí

*Departamento de Eléctrica y Electrónica
Universidad de las Fuerzas Armadas ESPE
P.O.BOX:17-5-231B, Sangolquí, Ecuador
Email: mvgualsaqui@espe.edu.ec*

Abstract—A Matlab program to simulate the Line Following Robot was developed as a didactic tool to learn the theory of fuzzy control. The Simulink environment is used to model the dynamics of mobile and the fuzzy controller, and a Script is used to model the line form and the deviation of the mobile from the line. The Simulator allows evaluating the performance of Proportional-Derivative (PD) Fuzzy Control by tuning the parameters of the controller and the speed of mobile. The PD Fuzzy Control shows a little better performance compared with a Proportional Derivative control. Each wheel of the mobile is modeled as a first order dynamic system. The parameters of mobile as the distance between wheels, the position of the sensor can be modified easily.

Keywords—line follower, robot, fuzzy control, Simulink, simulator, proporcional derivative control.

I. INTRODUCCIÓN

Los robots seguidores de línea son los más comunes en las competencias de robots móviles. El problema consiste en lograr que el robot realice una tarea y alcance el objetivo mediante pistas predefinidas en ambientes interiores. En la industria son usados para transportar bienes mediante una trayectoria predefinida. Estos robots son de fácil obtención y de bajo costo y por ello son una primera opción para el estudio de los algoritmos de control.

En [1] se afirma que los métodos de control basados en inteligencia artificial se utilizan en el diseño de sistemas de control complejos, que por su dificultad son inabordables por técnicas clásicas. Uno de tales métodos es la lógica difusa que ofrece una prometedora solución para manejar las incertidumbres del ambiente y que no requiere del modelo matemático del sistema a controlarse.

La terminología del control difuso se presenta con claridad en [2]. Allí se enseñan los cuatro bloques de un controlador difuso: la fusificación, la base de reglas, la máquina de inferencia y la defusificación. Se desarrolla en detalle el diseño de un control difuso de un péndulo invertido con la descripción de la formulación de los conjuntos difusos para las variables de entrada y salidas, así como también la formulación de la base de reglas.

El modelamiento de un robot diferencial se describe en [3], donde la cinemática del robot se expresa mediante tres estados, dos estados para la posición x, y y un tercer estado para el ángulo de dirección θ . En [4] se realiza la implementación de un móvil que dispone de un control con lógica difusa

cuyas entradas son la distancia con los bordes de una pista y el ángulo que forman el móvil y la trayectoria. El sistema utiliza una cámara para obtener la información indicada. La programación se realiza sobre una CPU en lenguaje C++ y se utiliza un controlador de movimiento embebido.

En [5] se desarrolla un sistema de control con lógica difusa que acepta la entrada de dos sensores de luz que entregan valores analógicos al controlador con los cuales se determina la desviación del móvil mediante el valor que resalta de la diferencia de los valores analógicos que entregan los sensores. El sistema de control produce sendas señales para cada rueda que corresponden a los valores difusos lento y rápido. El controlador se implementa con un microcontrolador AT89C52. En [6] se desarrolla un sistema de control con lógica difusa educacional de bajo costo para un robot seguidor de línea. En este trabajo dos sensores permiten detectar el nivel de sombra para obtener valores analógicos que determinan la desviación del móvil. Se definen tres conjuntos difusos (blanco, gris y negro) en la señal analógica de cada sensor para indicar el nivel de pertenencia del móvil a la posición izquierda o derecha. Se implementa el control mediante un microcontrolador PIC.

La simulación de la acción de un controlador en un móvil seguidor de línea sobre una pista con forma definida es conveniente para el análisis de los sistemas de control. Mediante la simulación se puede obtener mucha flexibilidad para generar condiciones concretas y obtener mucha variedad de información con la ventaja de utilizar pocos recursos. La simulación permite verificar la funcionalidad de los distintos métodos de control.

El trabajo está organizado en cinco secciones. Esta primera sección presenta los trabajos relacionados y la utilidad del simulador. En la sección II se presenta la descripción general del funcionamiento del simulador. La sección III muestra en detalle las dos partes complementarias para simular el robot con control difuso: un diagrama de bloques en Simulink y un Script. El diagrama de Simulink muestra un selector que permite conmutar entre un controlador Difuso y un controlador PD tradicional. La sección IV describe los resultados que genera el simulador así como también la comparación entre los controladores. En la sección V se presentan las conclusiones.

II. DESCRIPCIÓN GENERAL DEL SIMULADOR

El simulador se realiza mediante la interacción de un Script y un diagrama de bloques en Simulink. Tanto el modelo

dinámico del móvil como el controlador difuso se construyen mediante Simulink. En el Script se inicializan los valores de los parámetros del móvil, los parámetros del controlador y la ubicación inicial de móvil tanto en posición x y y como en dirección θ . La posición x y y corresponde al punto central del eje de las ruedas.

Los parámetros del móvil así como del controlador ingresan al diagrama de Simulink ya sea como bloques del tipo Constant ó como parámetros de condiciones iniciales en los bloques de Integración. El Script provoca el inicio y determina la duración de intervalos de simulación mediante el comando `sim('movilfuzzy',ts)` donde *movilfuzzy* es el nombre asignado al diagrama de Simulink y *ts* es el tiempo de duración de cada intervalo simulado. Durante cada intervalo simulado se actualizan las entradas del controlador, y al final del intervalo de simulación se obtienen los nuevos estados del móvil que se regresan al Script mediante bloques del tipo *simout*.

Para la simulación del móvil se utilizan como entradas la menor velocidad de las ruedas y la diferencia de velocidades angulares de las ruedas. Se modelan los estados que son la posición x y y junto con la dirección θ . El diagrama se muestra en la figura 1 donde se indican las ecuaciones en el espacio de estado ω_{movil} es la menor de las velocidades angulares de los motores que mueven las ruedas y se obtiene mediante la ecuación (1); ω_g es la velocidad de giro del móvil que se obtiene a partir de la diferente velocidad de las ruedas según la ecuación (2). v_d y v_i son las velocidades lineales de las ruedas derecha e izquierda, r es el radio de las ruedas y a es la distancia entre las ruedas.

$$\omega_{movil} = \min(v_d, v_i) / (2\pi r) \quad (1)$$

$$\omega_g = (v_d - v_i) / (2\pi a) \quad (2)$$

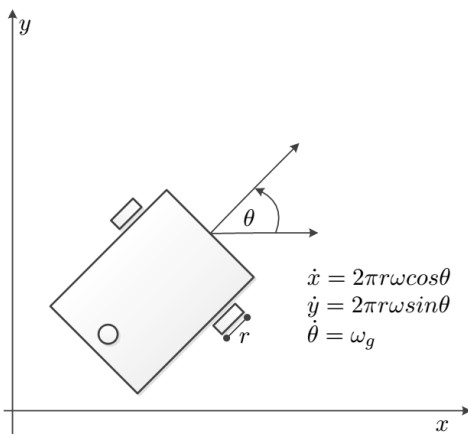


Fig. 1. Modelo matemático del móvil seguidor de línea

El sistema a simularse se basa en el kit seguidor de línea de dos ruedas de tracción y una rueda volante mostrado en la figura 2 que es de fácil adquisición en las tiendas de

dispositivos electrónicos. El controlador se implementa en una tarjeta stm32f4-discovery. El Kit se completa con un sensor de posición Pololu Qtr8A que permite medir la desviación en un rango de ± 38 mm.

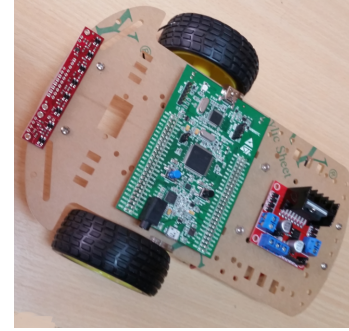


Fig. 2. Kit para Seguidor de línea

III. SIMULACIÓN

A. Diagrama en Simulink del simulador

El diagrama de Simulink que se muestra en la figura 3 se compone de cuatro bloques. Uno de los bloques realiza la simulación dinámica del móvil, un segundo bloque realiza la simulación dinámica de los motores. El tercer bloque realiza la simulación del controlador PD difuso y el cuarto bloque realiza la simulación de un controlador PD tradicional.

B. Simulación del móvil

El diagrama de simulación de los estados del móvil se realiza en Simulink y se muestra en la figura 4. La primera

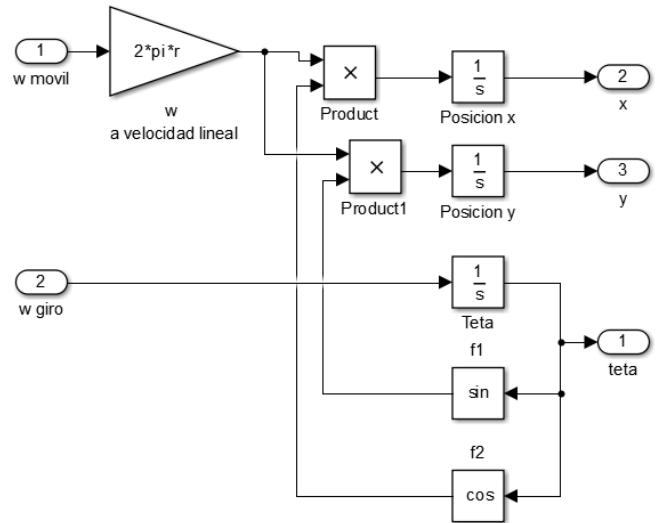


Fig. 4. Simulación de dinámica del móvil seguidor de línea

entrada representa la velocidad angular del móvil. El bloque de ganancia realiza la conversión de velocidad angular a velocidad lineal. Los bloques de integración permiten configurar los estados iniciales de la posición y dirección para la simulación en cada intervalo de tiempo.

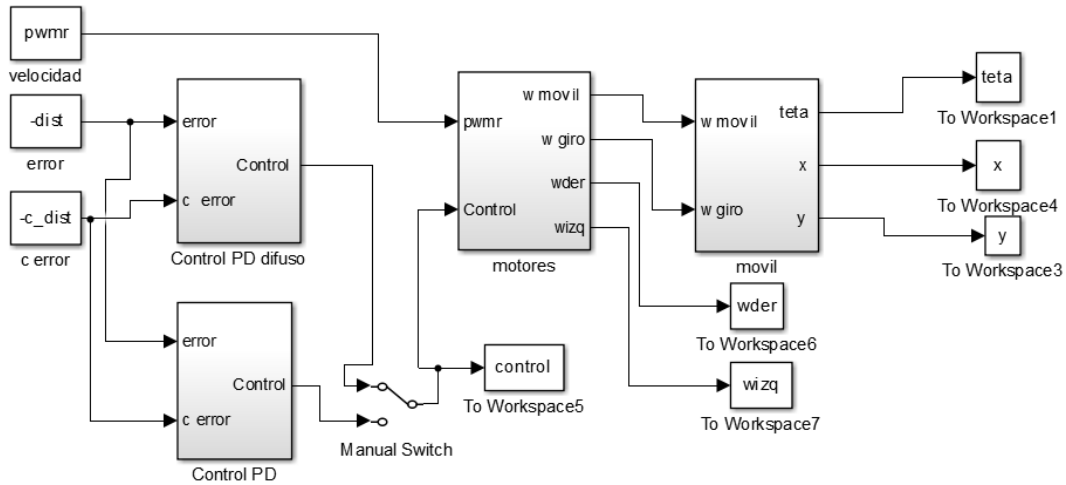


Fig. 3. Diagrama en Simulink de Simulador

C. Simulación de motores

El diagrama de simulación de los motores se muestra en la figura 5. La primera entrada representa la velocidad referencial

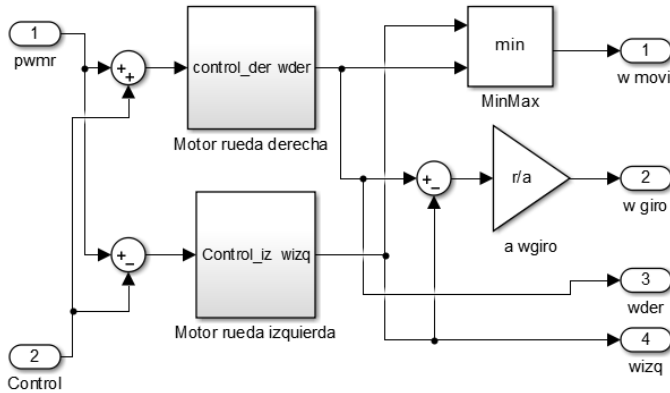


Fig. 5. Simulación de dinámica de los motores del móvil

del móvil que se ingresa como un valor de ciclo de trabajo en el rango $[0, 1]$. La segunda entrada es la señal de control que produce una diferencia de velocidad entre las ruedas e ingresa también como un valor de ciclo de trabajo entre 0 y 1. El bloque entrega como salida la velocidad angular del móvil y la velocidad angular de giro. Cada motor se simula como un sistema de primer orden que tiene como parámetro la ganancia K_m que representa la relación en estado estable de la velocidad angular con el ciclo de trabajo y τ_m que corresponde a la constante de tiempo del motor. La figura 6 muestra el diagrama de simulación del motor de la rueda derecha. Para la rueda izquierda se construye un diagrama similar.

D. Simulación del Controlador difuso

El controlador difuso acepta como entradas la desviación del móvil que constituye la señal de error y el cambio de la

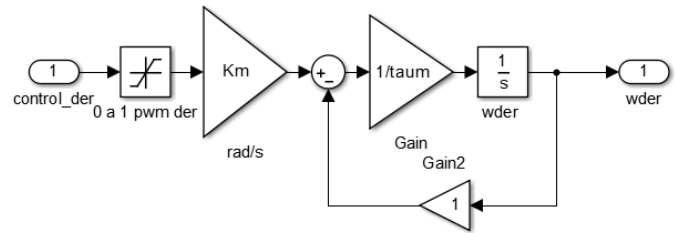


Fig. 6. Simulación de dinámica del motor de rueda derecha

desviación que es el cambio de error. La salida del controlador es la señal de control. El diagrama que se presenta en la figura 7 muestra las entradas y la salida. El bloque de control difuso se configura con una variable que almacena la estructura del control difuso. La estructura del control difuso contiene la definición de las funciones de pertenencia de las entradas y salidas, la base de reglas y la definición del tipo de inferencia difusa Mamdani.

Los valores lingüísticos para las variables error, cambio de error y señal de control son: grande negativo (gn), pequeño negativo (pn), cero (z), pequeño positivo (pp) y grande positivo (gp). Por ejemplo el conjunto difuso que representa la proposición: *El error es grande positivo* está nombrado con la variable *egp* y el conjunto difuso *El cambio de error es cero* se nombra con *cez*. El universo de discurso para cada variable está normalizada en el rango $[-1, 1]$. La figura 8 muestra la configuración del controlador difuso: en la figura 8(a) se definen los conjuntos difusos de la señal error, en la figura 8(b) los conjuntos difusos de la señal cambio de error; y en la figura 8(c) los conjuntos difusos de la señal de salida que corresponde al control. En la figura 8(d) se presenta la base de reglas. La regla de la casilla marcada se lee: *Si el error es pequeño negativo (epn) y el cambio de error es cero (cez),*

entonces la señal de control es pequeña negativa (upn).

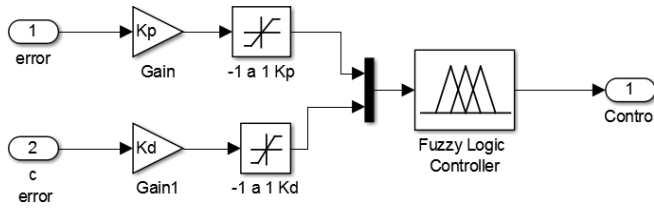
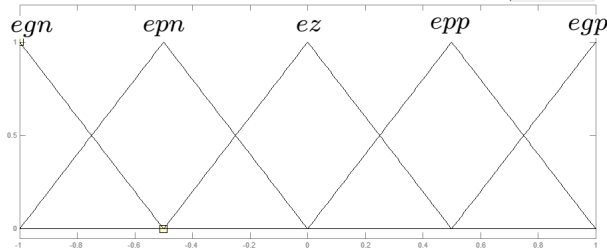
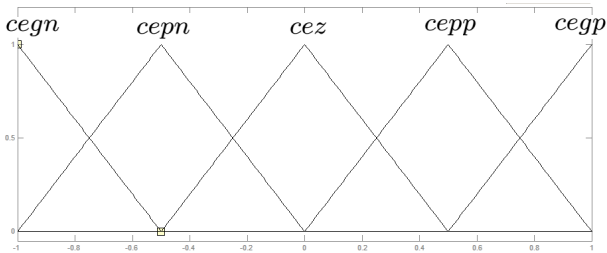


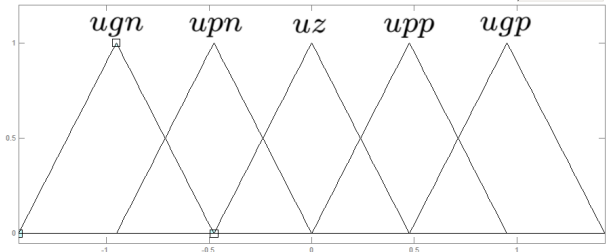
Fig. 7. Simulación del Control PD difuso



(a) Funciones de pertenencia para el error



(b) Funciones de pertenencia para el cambio de error



(c) Funciones de pertenencia para la señal de control

	egn	epn	ez	epp	egp
cegn	ugn	ugn	ugn	upn	uz
cepn	ugn	ugn	upn	uz	upp
cez	ugn	upn	uz	upp	ugp
cepp	upn	uz	upp	ugp	ugp
cegp	uz	upp	ugp	ugp	ugp

(d) Base de reglas

Fig. 8. Configuración del controlador difuso

E. Simulación del Controlador PD tradicional

El controlador PD acepta como entradas la desviación del móvil como señal de error y el cambio de la desviación como

cambio de error. La salida del controlador es la señal de control. El diagrama que se presenta en la figura 9 muestra el esquema del controlador. Se ha configurado el bloque de tal manera que produzca salidas limitadas en el rango $[-1, 1]$ de manera similar a lo que produce el control difuso. Las dos señales se restan para que la acción de control tenga el mismo signo que en el controlador difuso.

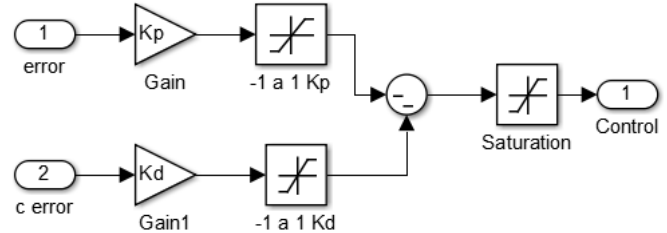


Fig. 9. Simulación del Control PD tradicional

F. Script de simulación de la Pista (línea)

La línea a seguir se ha generado mediante una matriz de dos columnas donde cada fila representa el par ordenado x y y que corresponde a un punto de la línea.

```

1 %Codigo para generar una pista circular
2 paso=0.01;
3 t=0: paso:2;
4 xp=.5*cos(pi*(t)+pi/2); yp=.5*sin(pi*(t)+pi/2);
5 plot(xp,yp);
6 title('Linea circular para movil');
7 axis equal

```

G. Script principal

El programa simulador del móvil seguidor de línea empieza definiendo los parámetros del controlador K_p y K_d y los parámetros del móvil como constante de tiempo de los motores, radio de las ruedas, voltaje aplicado a los motores, distancia entre ruedas, distancia del centro de eje de ruedas al centro del sensor.

Luego se carga la línea que se desea seguir y el tiempo de duración de cada intervalo simulado. Se define la posición inicial del móvil que corresponde al punto central del eje de las ruedas y el ángulo de dirección θ . Otro parámetro que se define es la velocidad deseada del móvil en términos de ciclo de trabajo de la onda PWM que se genera para controlar los motores. Esta variable es pwmr.

El lazo del programa realiza la actualización de las condiciones iniciales del próximo intervalo y el cálculo de la desviación del móvil respecto de la línea. Las condiciones iniciales que se actualizan corresponden a los estados del móvil x , y y θ y las velocidades angulares de los motores izquierdo w_{izq} y derecho w_{der} .

```

1 %Programa principal para simular un movil en una
  pista
2 Kd=200; %Kd= 200 para ajustar universo de
  discurso de cambio de posicion a +-5mm/s

```

```

3   Kp=26 %Kp=26 para ajustar universo de posicion
    a +-38mm
4   d=0.13; taum=10e-3;r=0.025;V= 18;a=0.25;rs=.076;
    Km=V/9*7.14*2*pi;
5   %d es la distancia desde el eje de ruedas a sensor
6   %taum es constante de tiempo del motor en segundos
7   %r es el radio de las ruedas
8   %V es el voltaje aplicado al motor la velocidad del
    motor es 7.14rps con 9 voltio
9   %a es la distancia entre ruedas
10  %rs es el rango del sensor
11  %Km es la relacion de velocidad angular rad/s a
    voltaje V del motor
12  pista4; %carga la linea a seguir
13  ts=10e-3;% tiempo retardo de la accion de
    control
14  figure(1);subplot(221);plot(xp,yp,'r');axis
    equal;hold on
15  %valores iniciales
16  dist=0;dist_ant=0;c_dist=0;int_dist=0; %
    distancia anterior a pista y cambio inicial
    de distancia, valor integral inicial
17  xini=.0;yini=-0.5 %valores de posicion inicial
    del centro de sensor
18  teta0=0; %los valores de posicion y direccion
    inicial de movil
19  x0=xini-d*cos(teta0);y0=yini-d*sin(teta0);%
    posicion inicial de centro de eje de ruedas
20  xv=[];yv=[];tetav=[]; %vectores que almacenan
    la evolucion de los estados del movil
21  distv=[]; %vector que almacena el error de
    seguimiento de linea
22  controlv=[];%para acumular control
23  wder0=0;wizq0=0;% inicializacion de velocidades
    de motores
24  pwmr=1; %valor de velocidad del motor expresada
    como ciclo de trabajo.
25  for i=1:1:200
26  sim('movil_fuzzy',ts)
27  x1=x+d*cos(teta);y1=y+d*sin(teta);
28  xsi=x1-rs/2*sin(teta);ysi=y1+rs/2*cos(teta); %
    posicion extrema izquierda de sensor
29  xsd=x1+rs/2*sin(teta);ysd=y1-rs/2*cos(teta); %
    posicion extrema derecha de sensor
30  xi=x-a/2*sin(teta);yi=y+a/2*cos(teta); %posicion
    de rueda izquierda
31  xd=x+a/2*sin(teta);yd=y-a/2*cos(teta); %posicion
    de rueda derecha
32  tetav=[tetav teta];controlv=[controlv control];%
    para graficar direccion teta y control
33  wiz0=wizq;wder0=wder;% se actualiza la condicion
    inicial de velocidades angulares
34  teta0=teta;% se actualiza la condicion inicial
    para siguiente simulacion
35  xv=[xv x1];% xv es un vector para mostrar la
    evolucion del punto central del sensor en x
36  x0=x; %se actualiza la condicion inicial de
    posicion x para siguiente simulacion
37  yv=[yv y1];% yv es un vector para mostrar la
    evolucion del punto central del sensor en y
38  y0=y; %se actualiza la condicion inicial para
    siguiente simulacion
39  subplot(221);plot(xp,yp,'r');axis equal;hold on
40  plot(x1,y1,'*r');plot(xi,yi,'or');plot(xd,yd,'or')
41  line([x x1],[y y1]);line([xsi xsd],[ysi ysd])
42  line([xi xd],[yi yd])
43  title('Linea a seguir y Trayectoria de movil')
44  plot(xv,yv);hold off
45  xsiv=xsi*ones(1,length(xp)); ysiv=ysi*ones(1,
    length(xp));
46  xsdv=xsd*ones(1,length(xp)); ysdv=ysd*ones(1,
    length(xp));
47  [distiz,iiz]=min((xsiv-xp).^2+(ysiv-yp).^2);%

```

```

    calcula distancia de linea a extremo
    izquierdo de sensor
48  [distder,ider]=min((xsdv-xp).^2+(ysdv-yp).^2);%
    calcula distancia de linea a extremo derecho
    de sensor
49  if xsdv(ider)^2+ysdv(ider)^2 > xp(ider)^2+yp(
    ider)^2 && xsiv(iiz)^2+ysiv(ider)^2 < xp(iiz
    )^2+yp(iiz)^2
50  dist=rs/2-rs*distiz/(distiz+distder);%
    desviacion de movil a pista
51  else
52  display('Sensor fuera de linea')
53  break
54  end
55  distv=[distv dist]; %para graficar desviacion
56  subplot(222);plot(distv);title('Desviacion_m')
57  subplot(223);plot(tetav);title('Direccion_de_
    movil_rad')
58  subplot(224);plot(controlv);title('Control')
59  drawnow
60  ddist=(dist+dist_ant)/2*ts;c_dist=(dist-dist_ant
    )/ts;
61  dist_ant=dist;
62  end

```

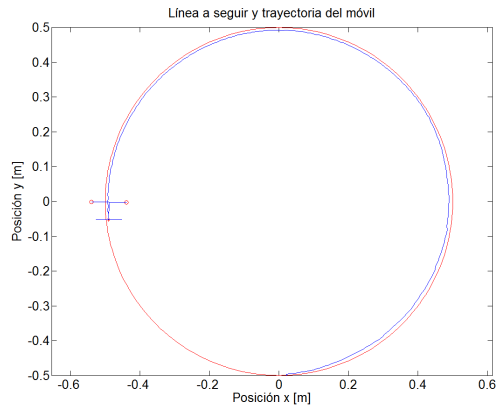
IV. RESULTADOS

La figura 10 muestra los resultados obtenidos con los parámetros definidos en el Script principal. En la subfigura 10(a) se observa la pista junto con la trayectoria que sigue el móvil como resultado de la acción de control. La subfigura 10(b) muestra la evolución de la señal de desviación del móvil en metros durante dos segundos. La subfigura 10(c) muestra la variación de la dirección θ en radianes. La subfigura 10(d) muestra la acción de control que está restringida al intervalo $[-1, 1]$.

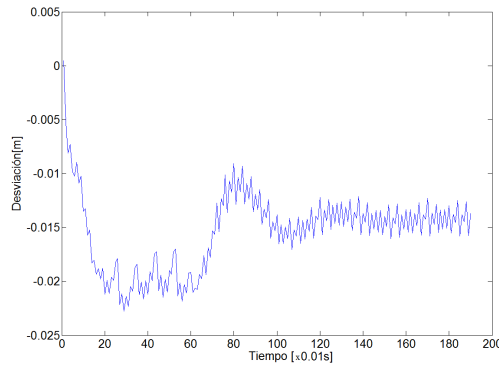
La tabla I se ha construido para realizar la comparación entre el desempeño de un control difuso proporcional derivativo PDD y un control proporcional derivativo tradicional PD. La tabla muestra el ángulo en radianes que recorre el móvil cuando se le aplica cada controlador. Los dos controladores se han sintonizado con los mismos valores de ganancia proporcional K_p y ganancia derivativa K_d . Se ha determinado como intervalo de prueba 50 pasos de simulación que significan 500 ms. En las columnas PDD y PD se indica el recorrido del móvil en radianes, considerando que la pista de prueba es circular. La última columna indica el porcentaje de diferencia %D entre los recorridos. Se observa que el mejor desempeño se obtiene para cuando $K_p = 130$ y $K_d = 40$ que es la sintonización que produce el máximo recorrido. Se puede notar también que en general solo se obtiene una leve ventaja del controlador PD Difuso frente al controlador PD tradicional para la mayoría de valores de ganancia K_p , K_d .

V. CONCLUSIONES

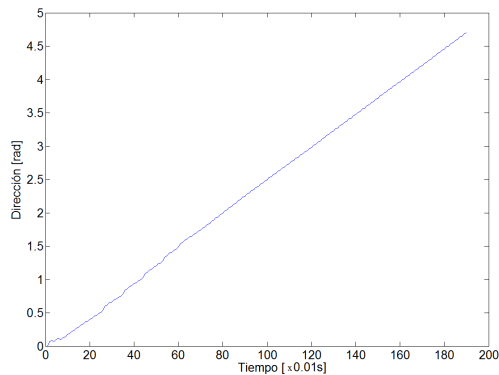
En este artículo se ha presentado un simulador de un móvil que realiza seguimiento de una línea gracias a la acción de un controlador Proporcional Derivativo Difuso. Los resultados que se muestran permiten observar la trayectoria que sigue el móvil, la variación en metros de la desviación respecto de la trayectoria deseada, el valor de la dirección en radianes del



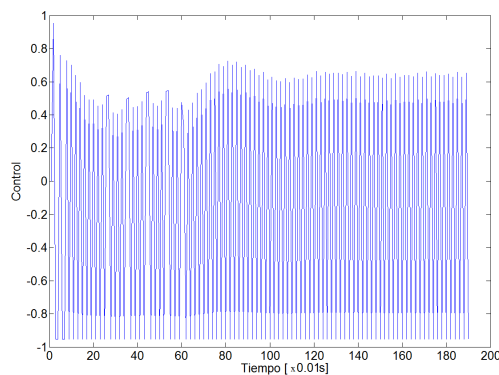
(a) Pista y trayectoria



(b) Desviación



(c) Dirección



(d) Control

Fig. 10. Resultados de la simulación

Tabla I
COMPARACIÓN DE CONTROLADORES.

K_p	K_d	PDD (rad)	PD (rad)	%D
26	200	1.23	1.19	3.4
52	100	1.14	1.05	8.5
130	40	1.54	1.52	1.3
260	20	1.50	1.37	9.5
520	10	1.41	1.41	0.0

móvil en cada instante, así como también el valor de la acción de control normalizada que se aplica en forma diferencial a las ruedas del móvil.

El simulador tiene mucha flexibilidad para la definición de los parámetros del móvil, de la pista y del controlador. Para el móvil se puede modificar el tamaño de las ruedas, la distancia entre ruedas, la ubicación del sensor, la longitud del sensor. Para el controlador se puede cambiar las constantes de ganancia proporcional y derivativa. La forma de la pista puede definirse fácilmente mediante una secuencia de puntos.

Se ha realizado una comparación entre la acción del controlador PD difuso frente a un controlador PD tradicional. El parámetro de desempeño para la comparación es el recorrido total del móvil luego de un tiempo de 500ms. De la comparación se concluye que existe un leve mejor desempeño con el controlador difuso.

REFERENCIAS

- [1] R. Galán, A. Jimenez, R. Sanz, F. Matía "Control inteligente," Revista Iberoamericana de Inteligencia Artificial, 2000.
- [2] K. Passino, S. Yurkovich, 'Fuzzy control', Addison- Wesley, 1998.
- [3] H. Cruz, E. Luz, D. Alcántara, " Robot de tracción diferencial con navegación mediante seguidor de línea y posicionamiento GPS," Instituto Politécnico Nacional, México D.F. 2013.
- [4] S. Pawlikowski, "Development of a fuzzy logic speed and steering control system for an autonomous vehicle," (editorial), 1999.
- [5] U. Farooq, M. Amr, M. Usman Asad, G. Abbas, A. Hanif, "Fuzzy logic reasoning system for line following robot," IACSIT International Journal of Engineering and technology, 2014.
- [6] D. Ibrahim, T. Alshanableh, "An undergraduate fuzzy logic control lab using a line following robot," Wiley InterScience, 2009.