

OBJETIVO:

Realizar el control con lógica difusa de un automóvil para que varíe su velocidad y sentido de giro en función de la distancia a la que se encuentre de algún obstáculo.

CONSIDERACIONES:

Sensor ultrasónico

Para que el proyecto pueda realizarse se requiere de algún sensor. Se emplea el sensor ultrasónico SRF05 ya que es muy sencillo de utilizar empleando Arduino. Este dispositivo puede trabajar máximo a una distancia de 4 metros y mínimo 2 cm. A continuación se muestra la distribución de las terminales del sensor así como el diagrama de tiempos de los pulsos para poder realizar una medición.

+5Vcc	Tensión positiva de alimentación
ECO	Salida del pulso cuya anchura determina el tiempo del recorrido de la señal ultrasónica
Disparo	Entrada de inicio de una nueva medida. Se aplica un pulso con una duración mínima de 10 μ s
Modo (N.C.)	Sin conexión se selecciona el modo 1 de compatibilidad con SRF04. Conectado a GND se selecciona el modo 2 de trabajo
GND	Tierra de alimentación.

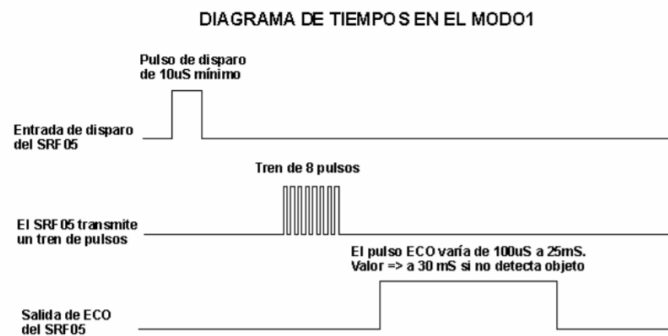
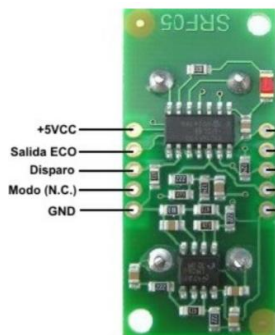


Figura 1. Terminales del sensor y diagrama de tiempos

Es decir, sólo se requiere generar un pulso de por lo menos 10 μ s y aplicarlo en la terminal de Disparo, luego se debe leer la señal recibida en la terminal ECO y convertir este pulso a su equivalente en centímetros.

PWM en arduino

La función *analogWrite(X)* genera una señal PWM con una frecuencia de alrededor de 500Hz. Donde la X puede tomar valores en el intervalo: $0 \leq X \leq 255$. Un 0 equivale a 0 %C.U. y 255 es 100 %C.U.

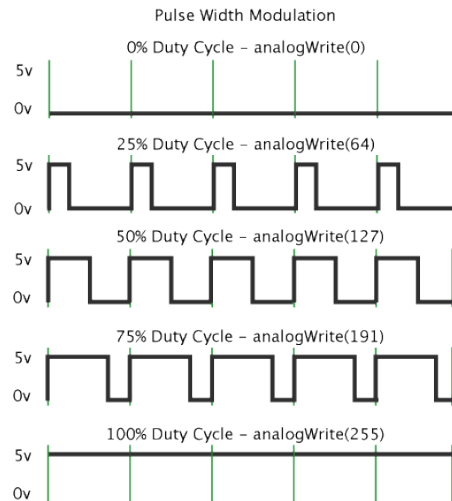


Figura 2. Señal PWM en Arduino

Motor, puente h y fuente de alimentación

El motor debe tener la capacidad para mover el carro con toda la carga necesaria (chasis, batería, sensor, tarjeta de arduino), además debe poder ser alimentado con un voltaje y corrientes relativamente bajos para que la batería sea lo más pequeña posible.

Se utiliza un motorreductor que puede ser alimentado por una batería de 9V - 1A/h. La corriente máxima exigida cuando el eje está bloqueado es de 600 mA.

Es importante mencionar que además de la fuente de alimentación para el motor, se requiere una segunda batería para alimentar al circuito de control, esto se hace con la finalidad de evitar daños en el mismo debido a las variaciones de corriente producidas por el motor.

El puente H utilizado, además de permitir el cambio de giro del motor, debe ser capaz de soportar la corriente exigida por el mismo. Se emplea el puente H L298N, a través del cual pueden circular hasta 3 Amp y puede ser alimentado por la batería antes mencionada, además puede trabajar de manera correcta con los pulsos del PWM proporcionados por el microcontrolador.

A continuación se muestra un segmento de la hoja de especificaciones del puente H.

ABSOLUTE MAXIMUM RATINGS

Symbol	Parameter	Value	Unit
V_S	Power Supply	50	V
V_{SS}	Logic Supply Voltage	7	V
V_i, V_{en}	Input and Enable Voltage	-0.3 to 7	V
I_O	Peak Output Current (each Channel)		
	- Non Repetitive ($t = 100\mu s$)	3	A
	- Repetitive (80% on ~20% off; $t_{on} = 10ms$)	2.5	A
	-DC Operation	2	A
V_{sens}	Sensing Voltage	-1 to 2.3	V
P_{tot}	Total Power Dissipation ($T_{case} = 75^\circ C$)	25	W
T_{op}	Junction Operating Temperature	-25 to 130	$^\circ C$
T_{stg}, T_J	Storage and Junction Temperature	-40 to 150	$^\circ C$

Figura 3. Valores máximos de los parámetros para el puente H L298N

DESARROLLO Y RESULTADOS:

Se definen cinco conjuntos para la entrada (distancia en cm entre el carro y el objeto).

con1: muy poca distancia (menor a 7 cm), con2: poca distancia (entre 15 y 7 cm), con3: media distancia (alrededor de 25 cm), con4: distancia grande (alrededor de 50 cm) y con5: distancia muy grande (mayor a 100 cm), estos se muestran en la figura 4.

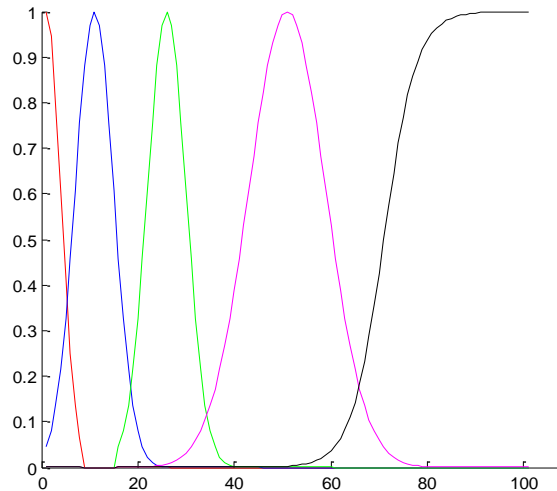


Figura 4. Conjuntos difusos para la entrada (Distancia en cm) vs membresía

EL circuito de control debe implementar las siguientes Reglas Difusas:

Regla 1 if Distancia Muy Grande (con5) then El auto avanza a máxima velocidad (S1).

Regla 2 If Distancia Grande (con4) then El auto avanza a media velocidad (S2).

Regla 3 If Distancia Media (con3) then El auto avanza a poca velocidad (S3).

Regla 4 If Distancia Poca (con2) then El auto se detiene (S4).

Regla 5 if Distancia Muy poca (con1) then El auto debe cambiar la dirección de giro y retroceder para alejarse del objeto (S5).

Para calcular el PWM de salida se emplea la siguiente ecuación

$$PWM = \frac{(\mu_{con1}(x) * S1) + (\mu_{con2}(x) * S2) + (\mu_{con3}(x) * S3) + (\mu_{con4}(x) * S4) + (\mu_{con5}(x) * S5)}{\mu_{con1}(x) + \mu_{con2}(x) + \mu_{con3}(x) + \mu_{con4}(x) + \mu_{con5}(x)}$$

Definición de los single tones

Los single tones de salida son los encargados de definir la velocidad del motor (%C.U. del PWM) y para poder seleccionar su valor es necesario conocer cómo se comporta el coche para diferentes %C.U. del PWM.

Al realizar pruebas se observa que con la batería de 9 Vdc-1A/h se logra mover al auto a partir de un valor de PWMArduino=50, es decir, aproximadamente 20 %C.U. y la máxima velocidad se alcanza con PWMArduino=525, es decir, 100 %C.U.

Se definen los single tones con los siguientes valores

$$\begin{aligned} S1 &= 100 \text{ (39 \%C.U.)}, & S2 &= 0 \text{ (0 \%C.U.)}, & S3 &= 50 \text{ (20 \%C.U.)}, \\ S4 &= 80 \text{ (31 \%C.U.)}, & S5 &= 180 \text{ (70 \%C.U.)} \end{aligned}$$

Estos valores se eligieron de manera experimental, ya que con ellos se logra ver un cambio suave en la velocidad a medida que el auto se acerca al objeto.

A continuación se muestra una gráfica de la variación de la velocidad (variación del ciclo de trabajo del PWM) en función de la distancia.

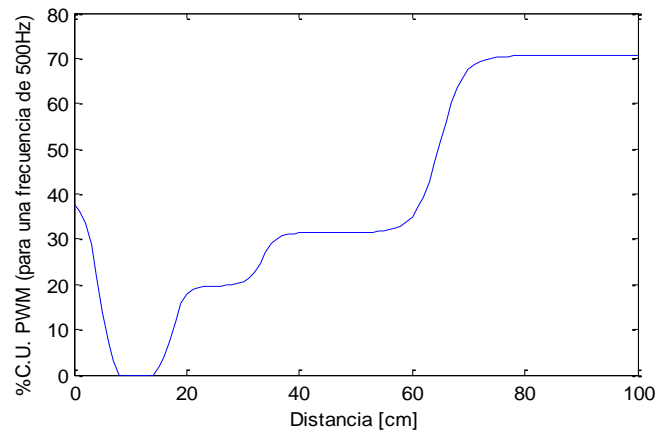


Figura 5. Variación de la velocidad en función de la distancia (simulación realizada en Matlab)

En la figura 5 se observa que los cambios de velocidad son suaves a medida que la distancia disminuye, que el motor se apaga cuando se está a una distancia de entre 7 y 15 cm y que la velocidad vuelve a aumentar cuando el obstáculo se acerca más allá de 7cm (aquí el sentido de giro del motor es cambiado para alejarse del obstáculo), esto se realizó con una simulación en Matlab, además en el prototipo físico el sistema de control también se comporta como se esperaba.

PROTOTIPO:

A continuación se muestra una fotografía del prototipo empleado para probar el sistema.

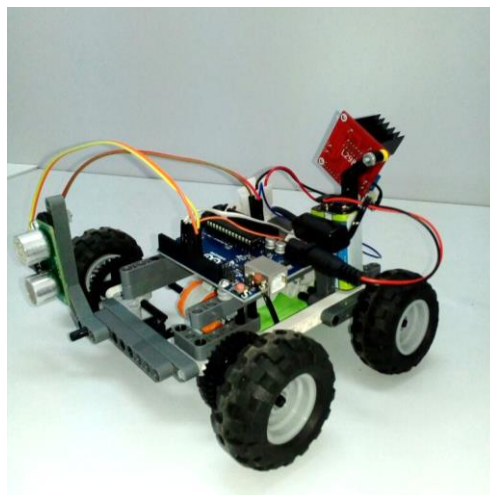


Figura 6. Prototipo

PROGRAMA EN ARDUINO:

```
#include <NewPing.h>

int a1=0, d1=3, a2=10, d2=4, a3=25, d3=4, a4=50, d4=8, a5=70;
int s1=100, s2=0, s3=50, s4=80, s5=180;
int Distancia_cm=0, val=0;
double con1,con2,con3,con4, con5;
double pwm=2;

/*Aqui se configuran los pines donde debemos conectar el sensor*/
#define TRIGGER_PIN 12
#define ECHO_PIN 11
#define MAX_DISTANCE 400

/*Crear el objeto de la clase NewPing*/
NewPing sonar(TRIGGER_PIN, ECHO_PIN, MAX_DISTANCE);

void setup()
{
}

void loop()
{
    // Esperar 1 segundo entre mediciones
    delay(200);
    // Obtener medicion de tiempo de viaje del sonido y guardar en variable uS
    int Distancia_uS = sonar.ping_median();
    int Distancia_cm = Distancia_uS/US_ROUNDTRIP_CM;//calcula la distancia en cm

    con1=exp((- (Distancia_cm-a1)^2)/(2*(d1^2)));
    con2=exp((- (Distancia_cm-a2)^2)/(2*(d2^2)));
    con3=exp((- (Distancia_cm-a3)^2)/(2*(d3^2)));
    con4=exp((- (Distancia_cm-a4)^2)/(2*(d4^2)));
    con5=1/(1+exp(-0.3*(Distancia_cm-a5)));

    if (Distancia_cm<=7)//Reversa
    {
        //Giro hacia atrás
        digitalWrite(7,LOW); //Enable 1
        digitalWrite(8,HIGH); //Enable 2
    }

    else
    {
        //Giro hacia adelante
        digitalWrite(8,LOW); //Enable 1
        digitalWrite(7,HIGH); //Enable 2
    }

    //*****STOP IMPORTANTE*****
    if (Distancia_cm>7 & Distancia_cm<15)
    {
        con1=0;
        con2=exp((- (Distancia_cm-a2)^2)/(2*(d2^2)));
        con3=0;
        con4=0;
        con5=0;
    }

    pwm= ((con1*s1)+(con2*s2)+(con3*s3)+(con4*s4)+(con5*s5))/(con1+con2+con3+con4+con5);
    analogWrite(9,pwm);
}
```

CONCLUSIONES:

Inicialmente se definió que el automóvil se detuviera a 7 cm del objeto frente a él, sin embargo al llegar a esta distancia, y debido a la inercia que llevaba, no se lograba detener, haciendo que oscilara hacia adelante y atrás indefinidamente, por tal motivo fue necesario agregar una condición de STOP entre 7cm y 15 cm, esto es, en ese intervalo de distancias el motor debe estar “apagado”.

Cuando el automóvil se encuentra a una distancia mayor de 4 metros de algún objeto, el sensor envía un valor de 0, haciendo que el motor gire en reversa, lo cual es un inconveniente, pero es una limitante debida a la capacidad de medición del sensor.

A pesar de los problemas con la condición de STOP y la detección de objetos a más de 4 metros, el sistema de control con lógica difusa fue satisfactorio.

Referencias

<https://www.arduino.cc/en/Tutorial/PWM>

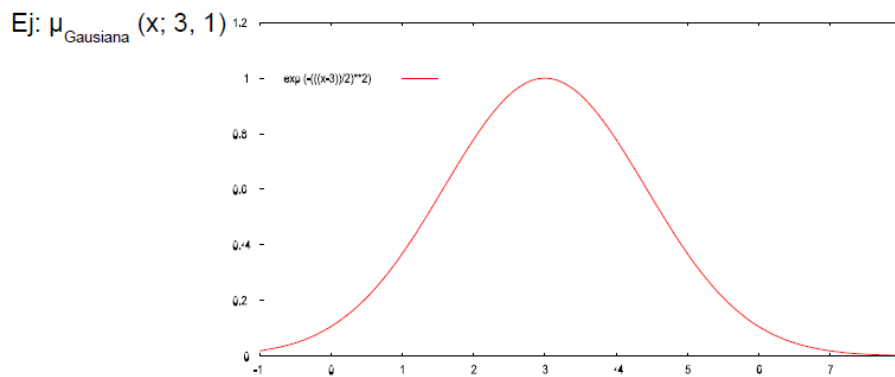
http://riobotics-test.weebly.com/uploads/9/3/0/9/9309609/medidor_ultrasonico_srf05.pdf

Apéndice

Para las funciones de pertenencia se utilizan Gaussianas para las velocidades pequeñas y una sigmoide para la velocidad máxima

Funciones de pertenencia (cont):

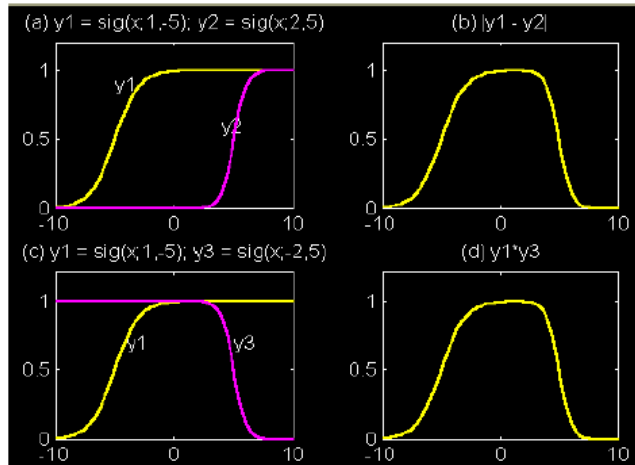
- Gausiana
- $$\mu_{Gausiana}(x; c, \sigma) = e^{-\frac{1}{2} \left(\frac{x-c}{\sigma} \right)^2}$$



El σ nos da el ancho de la campana

- Sigmoide:

$$s i g m f (x ; a , b , c) = \frac{1}{1 + e^{-a(x-c)}}$$



Referencias:

- [1] Yager, R., Filev, D., Essentials of Fuzzy Modeling and Control, Wiley Interscience, NY, 1994
- [2] Kartalopoulos, S., Understanding Neural Networks and Fuzzy Logic, IEEE PRESS, NY, 1994
- [3] Jang, J., et al, Neuro-Fuzzy and Soft Computing, Prentice Hall, 1997
- [4] www.seattlerobotics.org