Shivansh Mishra

Calculating the speed of the DC Motor using the Ultrasonic Sensor

Apparatus Used—

(1) Arduino UNO
(2) Ultrasonic Sensor (HC-SR04)

What is the Ultrasonic sensor?

Ultrasonic sensor is the sensor that send the ultrasonic signals and receive it and measures the time of the signal's tour.

Now by multiplying the speed of the sound with the time it's easy to calculate the distance of the obstackle.

How does it actually fulfil our need?

a. The sensor HC-SR04 is triggered by the PWM and send the PWM of the pulse width for the time of travel of the sound wave.
b. Now we calculate the time for which the received pulse is on and calculate the distance.
c. Now RPM is the no of round taken in one minute.
d. So we set the motor such that the we calculate the time for which two continuous obstacles comes and then calculate the RPM.

Problem and fixings—

(i) While I was doing this, then, their came a problem that I could measure the speed at higher rpm accurately but on the lower RPM our model was not doing good.
(ii) The root cause we identified that this was due to our obstacle width.
(iii) Aniruddh bhaiya suggested that use the concept of the state machine for your purpose.
(iv) Concept of state machine simply say that if present in that state, then 1 otherwise zero.
(v) So we measured the time between two successive states and problem was fixed.

Code in Arduino—

```
#include <NewPing.h>


const int trigPin = 12;

const int echoPin = 13;

const float obstacle_distance = 4;  // in cm
```

```
#define MAX_DISTANCE 10
NewPing sonar(trigPin, echoPin, MAX_DISTANCE);

bool prevstate = false;
bool currstate = false;

unsigned long time1 = 0, time2 = 0;
int rpm = 0;
const unsigned long minTimeDiff = 80;

void setup() {
  Serial.begin(9600);
}

void loop() {
  delay(15);
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);

  unsigned int distance = sonar.ping_cm();
  if(distance>0 && distance <5){
    currstate = true;
  }else{
    currstate = false;
  }
  if(currstate == true && prevstate == false){
    time2 = millis();
    if (time1 != 0 && (time2 - time1) > minTimeDiff) {
      unsigned long timeDiff = time2 - time1;
      int exact_rpm = (60000.0 / timeDiff);
      rpm = (exact_rpm -(exact_rpm%10));

      Serial.print("RPM: ");
      Serial.println(rpm);
    }
    time1 = time2;
  }
  prevstate = currstate;
}
```