Obstacle detection robot controlled by PD controller

Members:

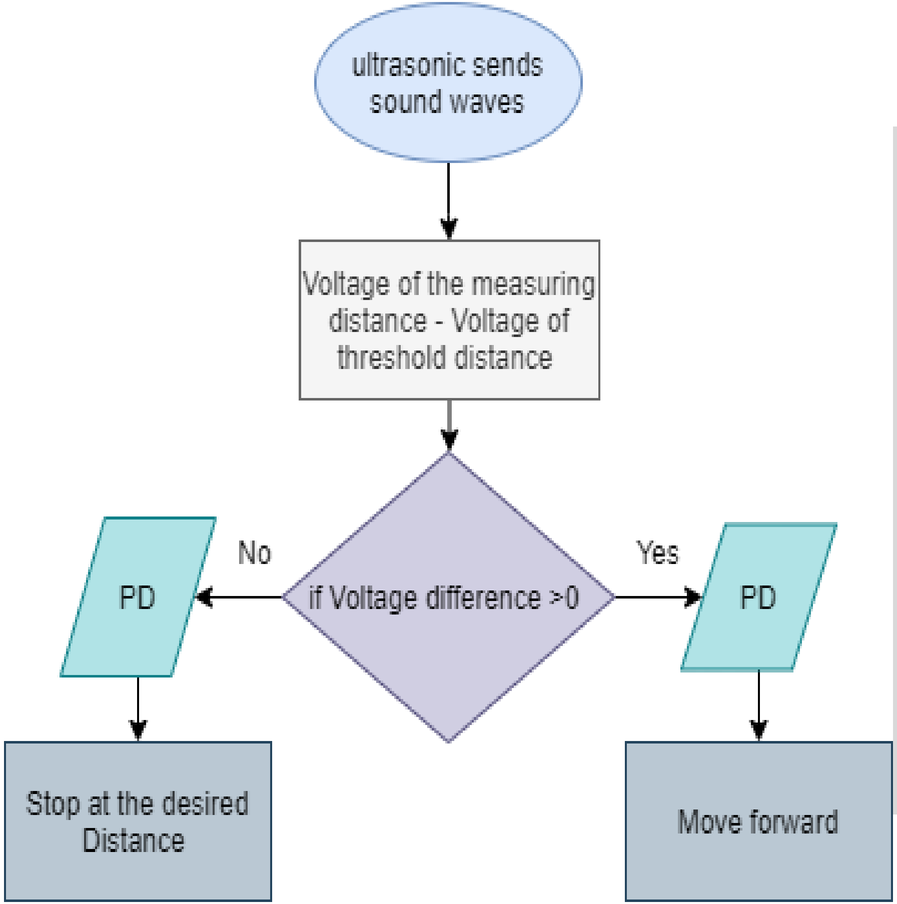
Alaa Ashraf Fawzy Elsayed
Alaa Elsayed Alaraby Ahmed Mahdy
Alzahraa Mohamed Abdelhamid Shaheen
Amir Hany Hussain Mohamed
Eman Abdelghany Sayed Eldeeb
Eman Mohamed Ahmed Gomaa
Manar Abdulla Tawfeeq Ibrahim
Hanaa Taha Ahmed Yousef
Wafaa Samir Abdelrheem Goda

Project Idea And Its Impact :

Control the distance between the robot and obstacles using ultrasonic sensor and control its output by “PD” controller.

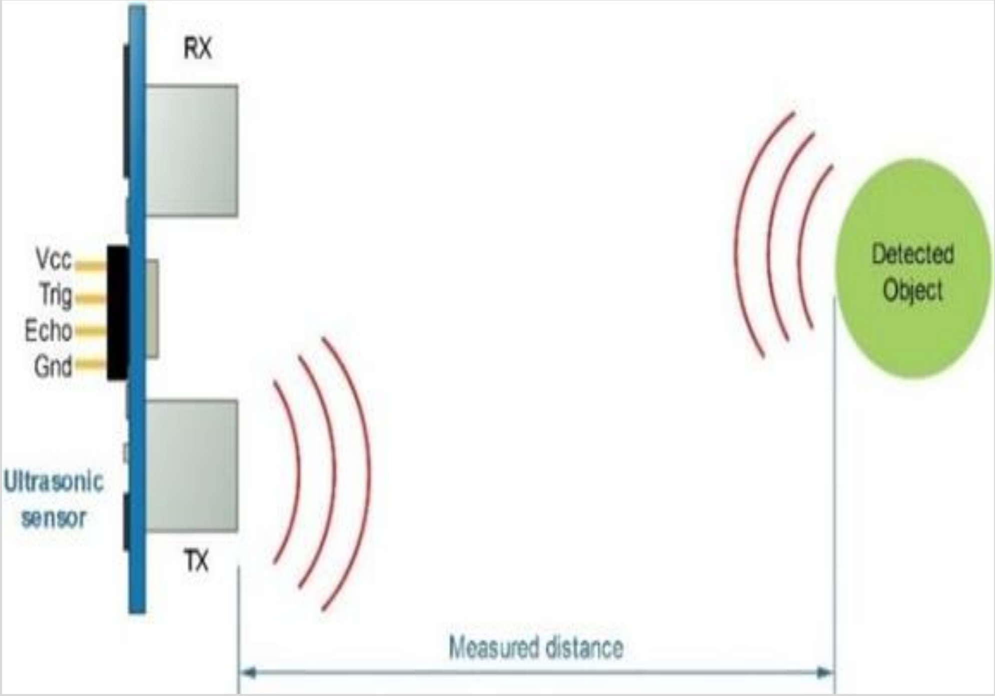
It can be used to keep a safe distance between cars and reduce traffic accidents

Flow chart


ultrasonic sends sound waves
Voltage of the measuring distance - Voltage of threshold distance
if Voltage difference >0
No
Yes
PD
PD
Stop at the desired Distance
Move forward

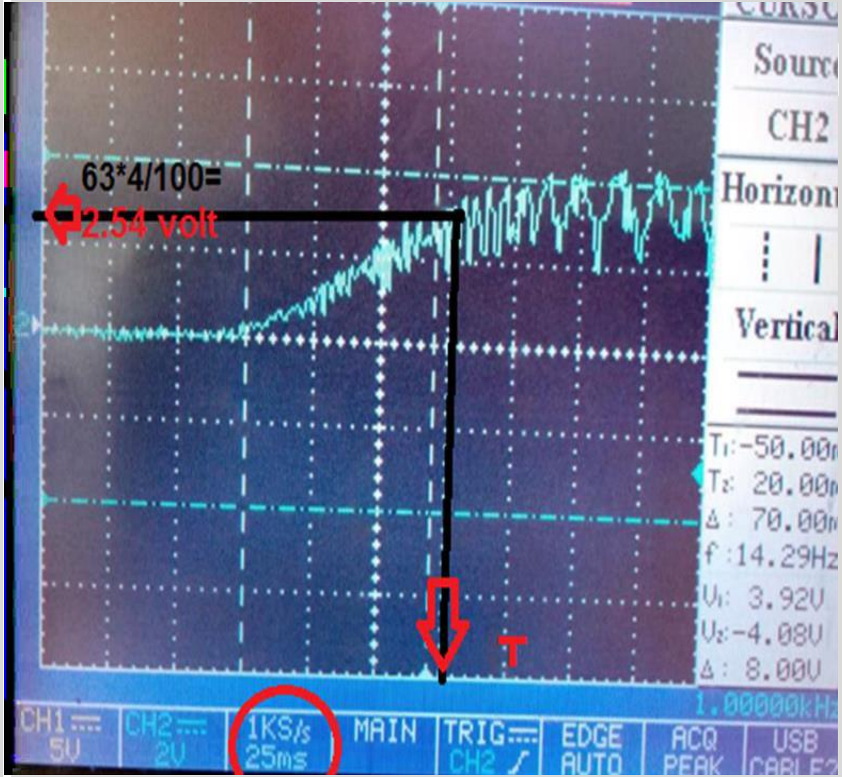
Ultrasonic sensor Working Principle:

When an electrical signal is applied to this transducer, it vibrates around the specific frequency range and generates a sound wave. These sound waves travel and whenever any obstacle comes, these sound waves will reflect the transducer inform of echo. And at the end of the transducer, this echo converts into an electrical signal. Here, the transducer calculates the time interval between the sending of the sound wave to the receiving the echo signal. The ultrasonic sensor sends the ultrasonic pulse at 40 kHz which travels through the air.



The distance can be measured by the following formula

$D = \frac{1}{2} * T * C$



Get Transfer Function of our DC

$$\frac{C(s)}{R(s)} = \frac{K}{Ts+1}$$

K is the D.C gain, T is the time constant of the system.

K=Steady state output /input=4/5= 0.8

T=time at which response reach 63% from the steady state

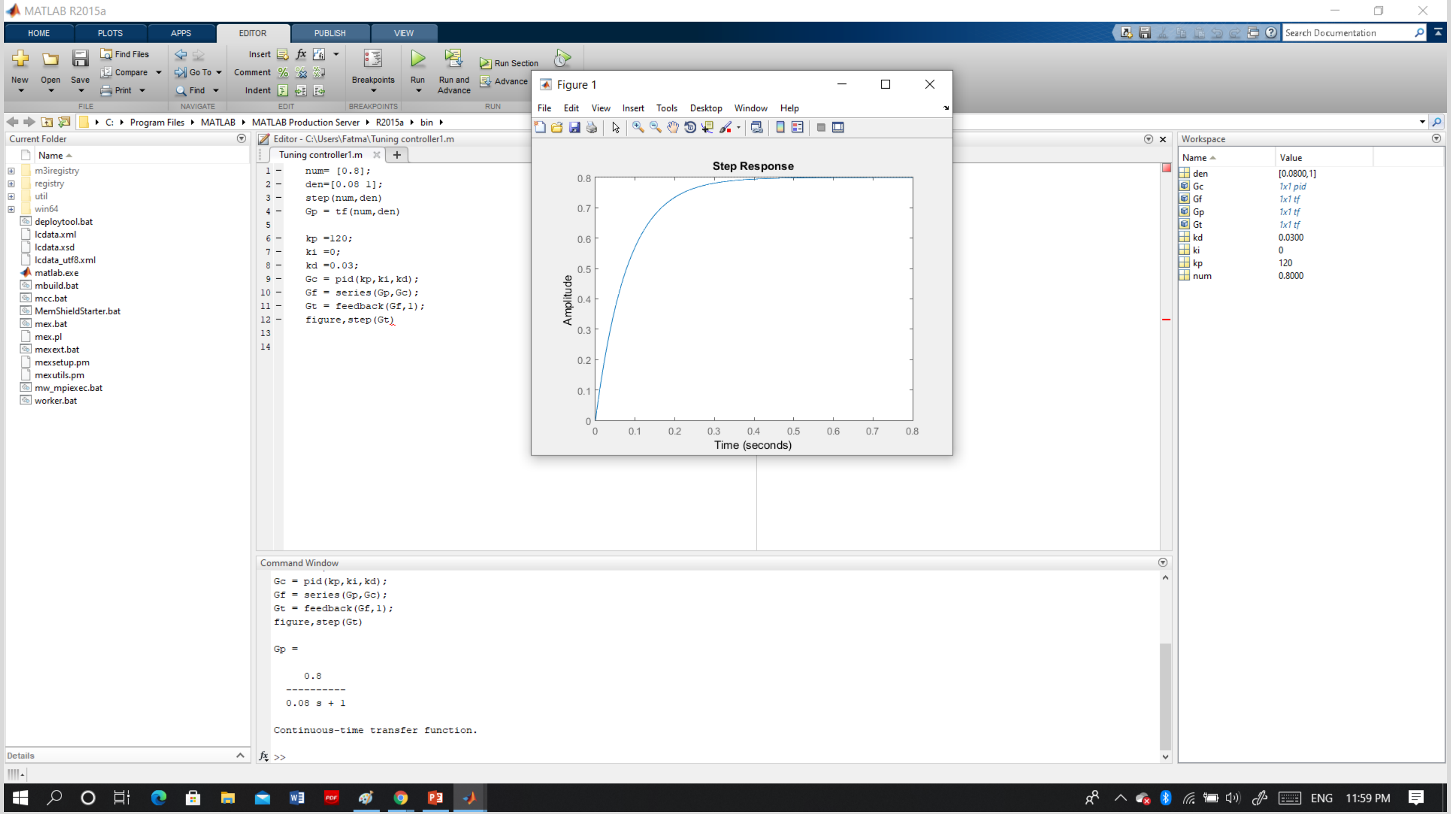
= 25*10^(-3)*3.2= 0.08sec

So transfer function = 0.8/(0.08s+1)

PID tuning on Matlab

Using Matlab PID tuning on our motor transfer function we apply proportional control to increase control signal proportionally to error value but Kp increases overshoot also so we add derivative control, we adjust Kp and Kd values to achieve a critically damping response

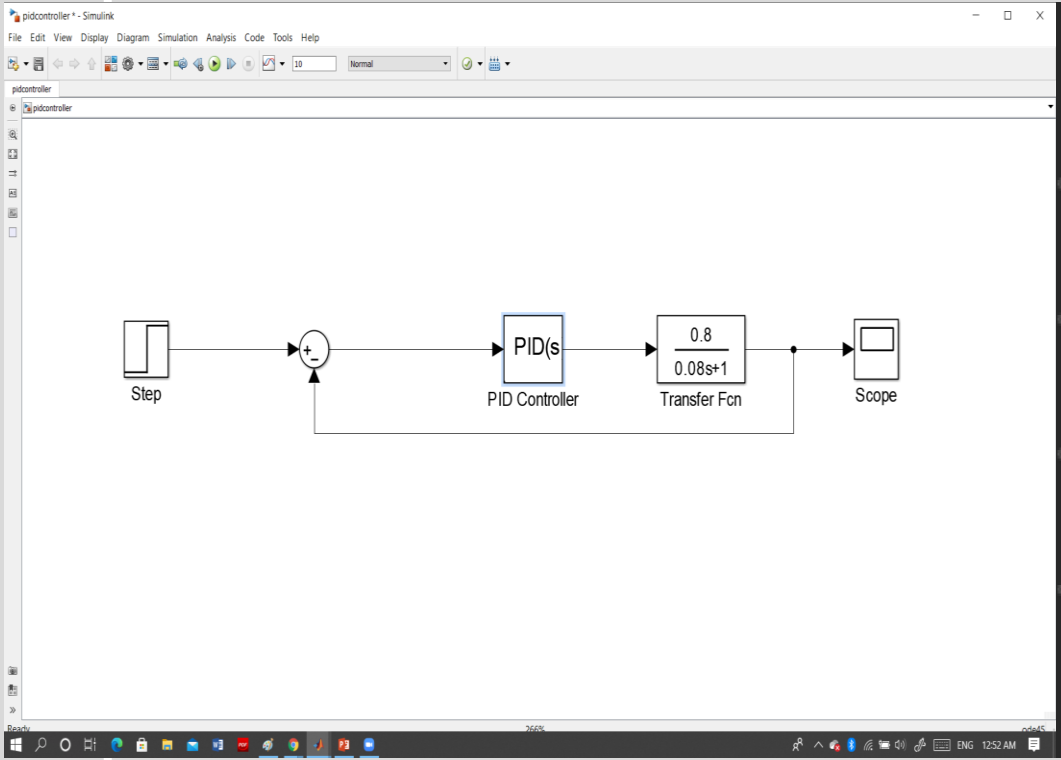
open loop response:

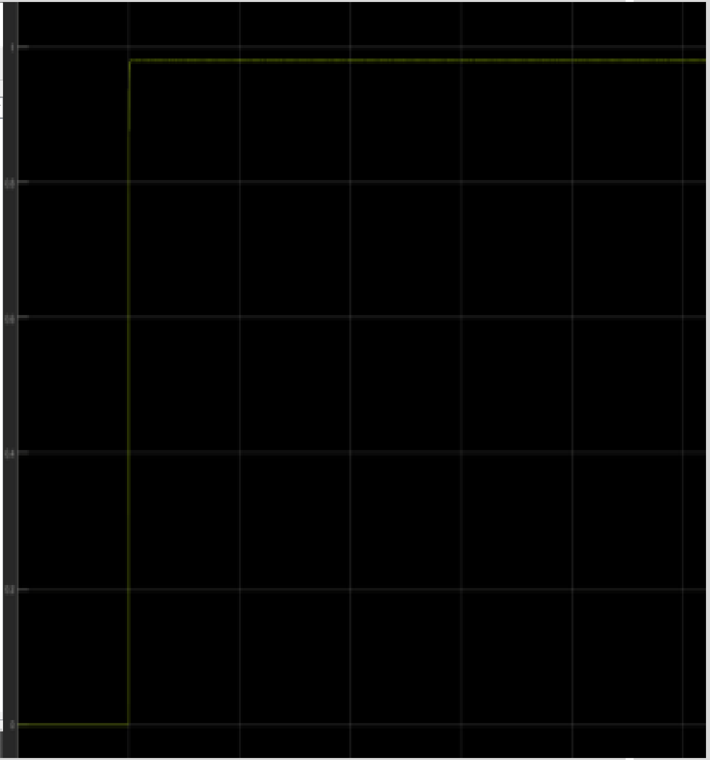


```
num= [0.8];
den=[0.08 1];
step(num,den)
Gp = tf(num,den)

kp =120;
ki =0;
kd =0.03;
Gc = pid(kp,ki,kd);
Gf = series(Gp,Gc);
Gt = feedback(Gf,1);
figure,step(Gt)
```

```
Gc = pid(kp,ki,kd);
Gf = series(Gp,Gc);
Gt = feedback(Gf,1);
figure,step(Gt)

Gp =

    0.8
  ----------
  0.08 s + 1

Continuous-time transfer function.

>>
```

Name	Value
den	[0.0800,1]
Gc	1x1 pid
Gf	1x1 tf
Gp	1x1 tf
Gt	1x1 tf
kd	0.0300
ki	0
kp	120
num	0.8000


pidcontroller * - Simulink
File Edit View Display Diagram Simulation Analysis Code Tools Help
10
Normal
pidcontroller
pidcontroller
Step
+
–
PID(s
PID Controller
0.8
0.08s+1
Transfer Fcn
Scope
Ready
266%
ode45
ENG 12:52 AM


1
0.8
0.6
0.4
0.2
0

we got Transfer Function from lab then:

1. plot response without controller

2. try values for Kp & Kd til we get the response we need

we want to get the ideal error "that makes output equal input"

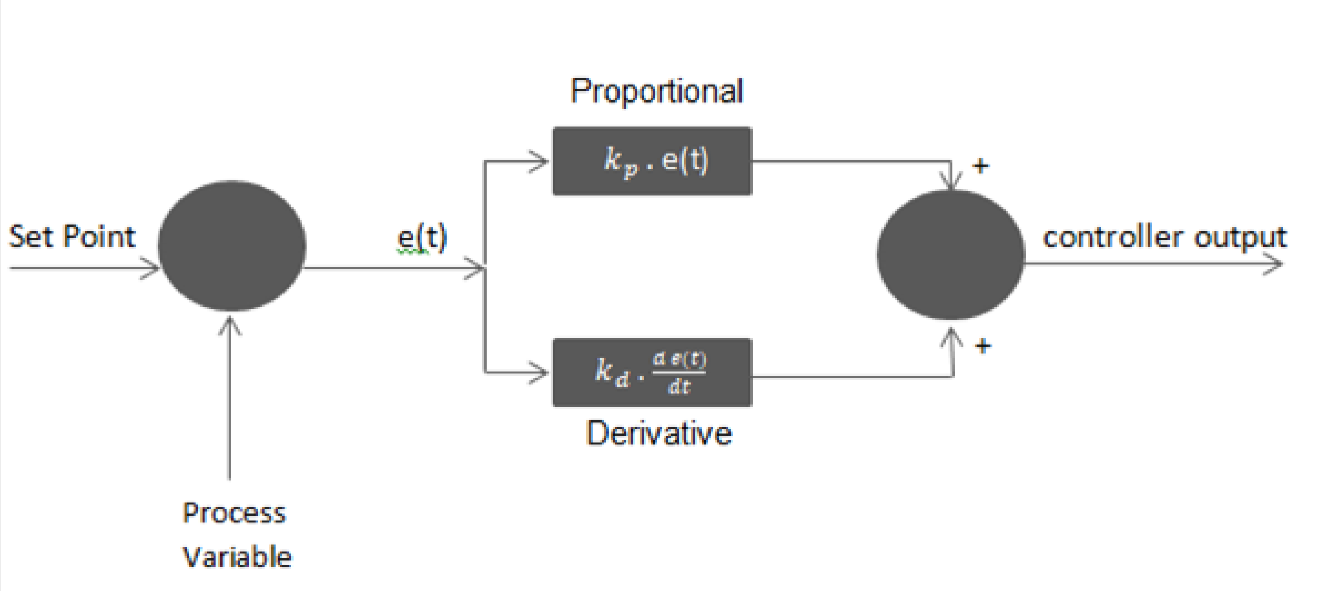
increasing Kp in range from 50 to 120 lead to:

-decreasing the error and steady state

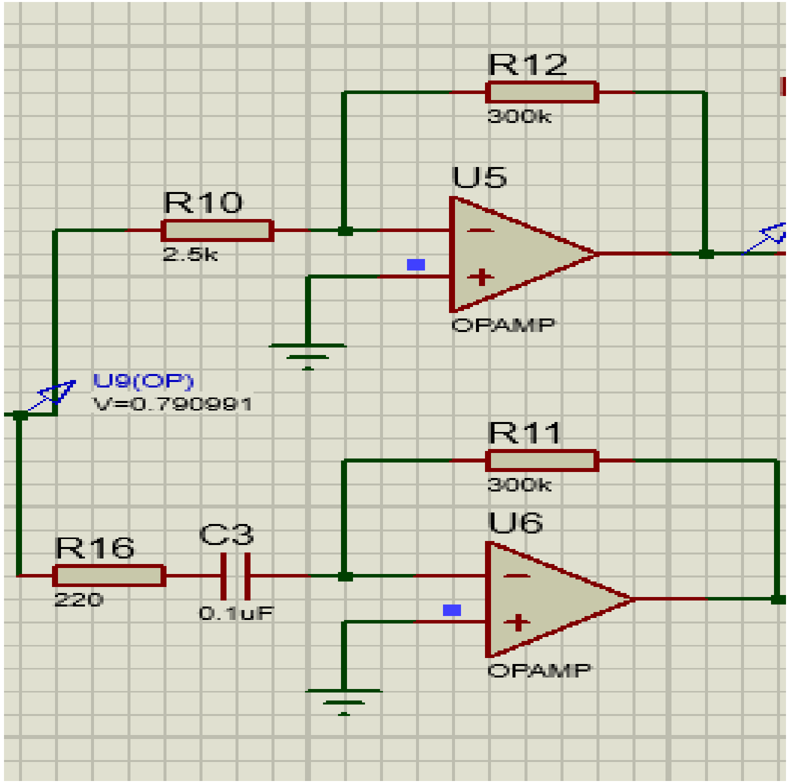
-increasing overshoot so we decrease Kd then try and try until we get stable system

3. the value of Kd that gives best response =0.03 & the value of Kp=120

4. finally we get that response:



$m(t) = K_p * e(t) + K_d * \frac{de(t)}{dt}$ The transfer function can be written as: $\frac{M(s)}{E(s)} = K_p + K_d * s$



Let Rf= 300K ohms

Kp =Rf / R1 = 120

Kd = Rf * c = 0.03

So, R1 =2.5k ohms

Rf =300k ohms

C =0.1uf

Proteus Schematic

Circuit:

- Sensor circuit

The trigger pin needs a pulse signal to start the sensor so we use the timer555 in the astable mode to generate pulses and give the output to the trigger. Then the sensor emits ultrasonic waves that travel in the air until obstacles then return back to The echo pin .

The output of the sensor is a pulse signal represent the distance between the sensor and the obstacle

To change sensor position virtually in proteus we use a potentiometer to divide the voltage representing the measuring distance when the distance increases the voltage increases this voltage that we compare with the threshold.

Circuit:

- ❖ **Differential circuit**

We use this circuit to subtract the distance between the sensor and the obstacle from the setpoint reference (the distance we want the robot to stop) The result is negative so we put an inverter

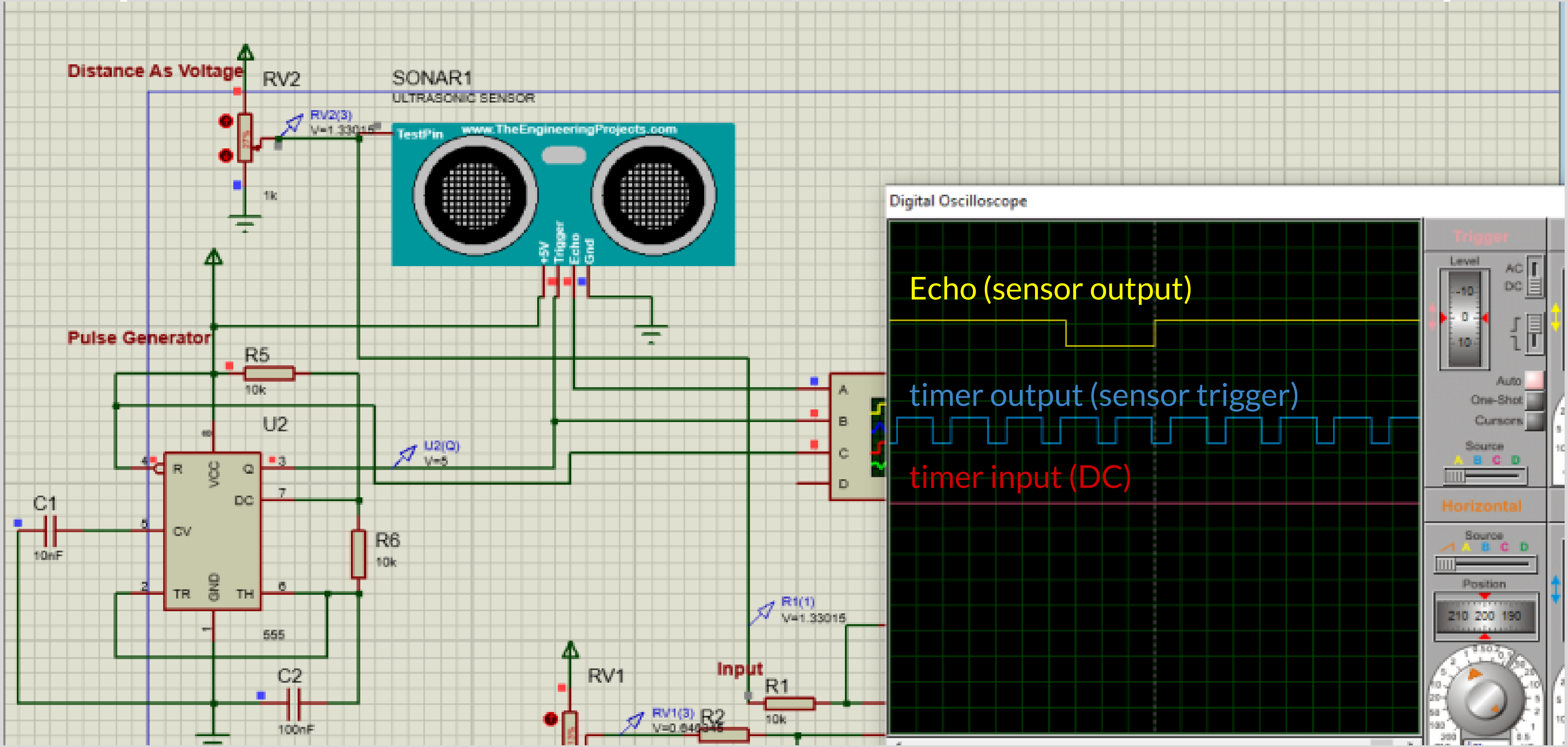
- ❖ **PD circuit**

We use PD circuit to control the output as we need so we use a summing inverting op amp to add the output of “P” and “D.”

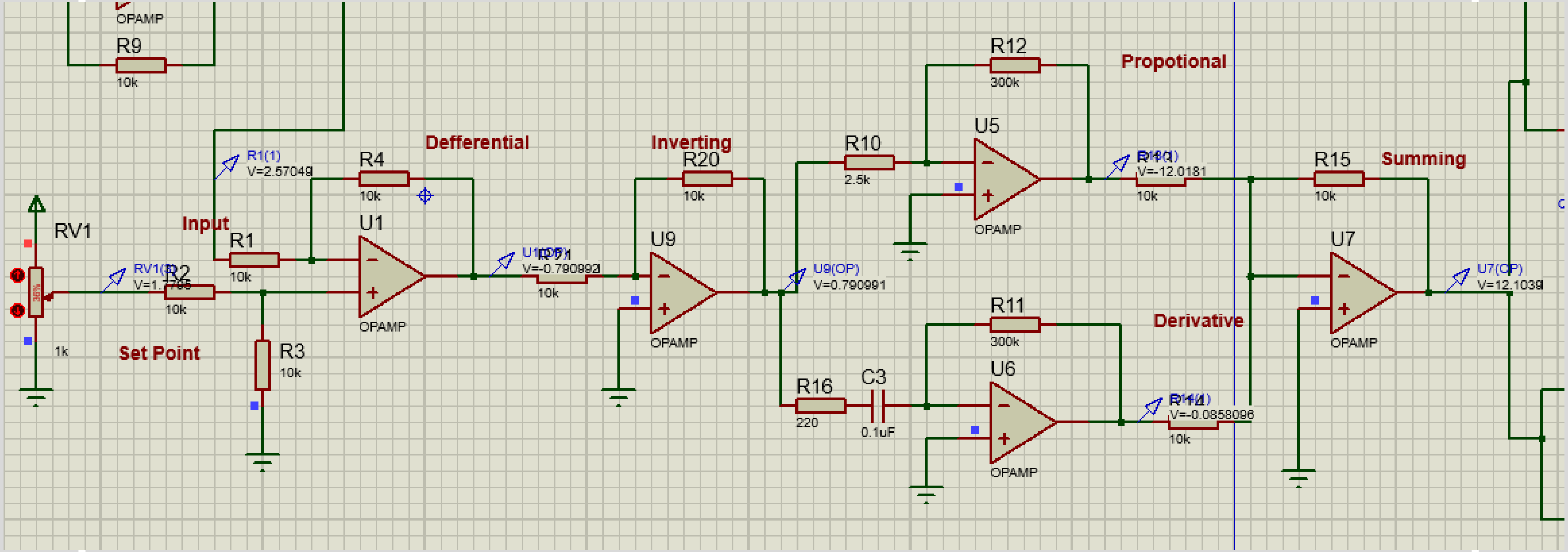
- ❖ **motor driver circuit**

It Contains ‘PNP ‘and ‘NPN’ transistors We use this driver to amplify the current to be able to rotate the motor .

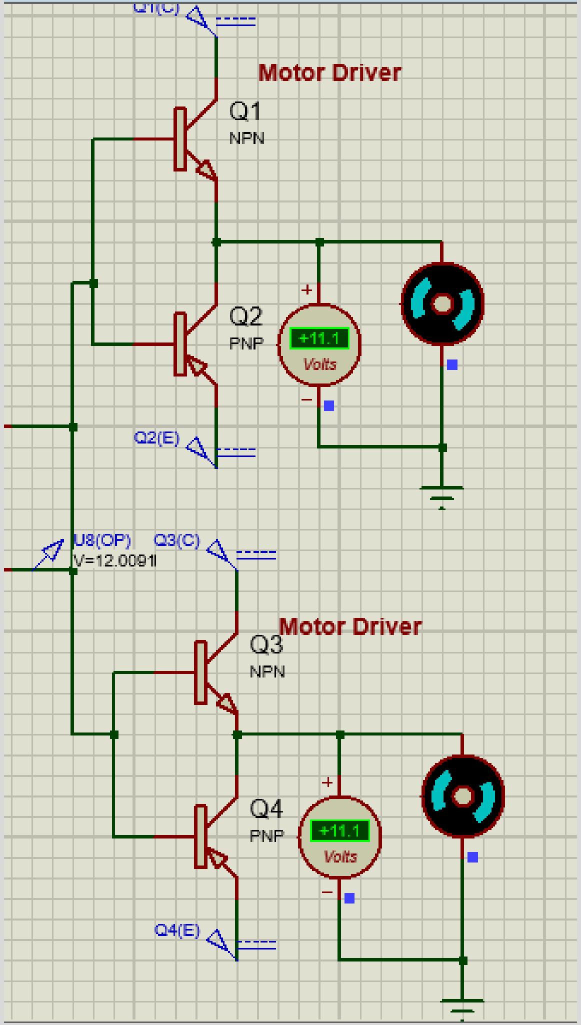
❖ Sensor circuit

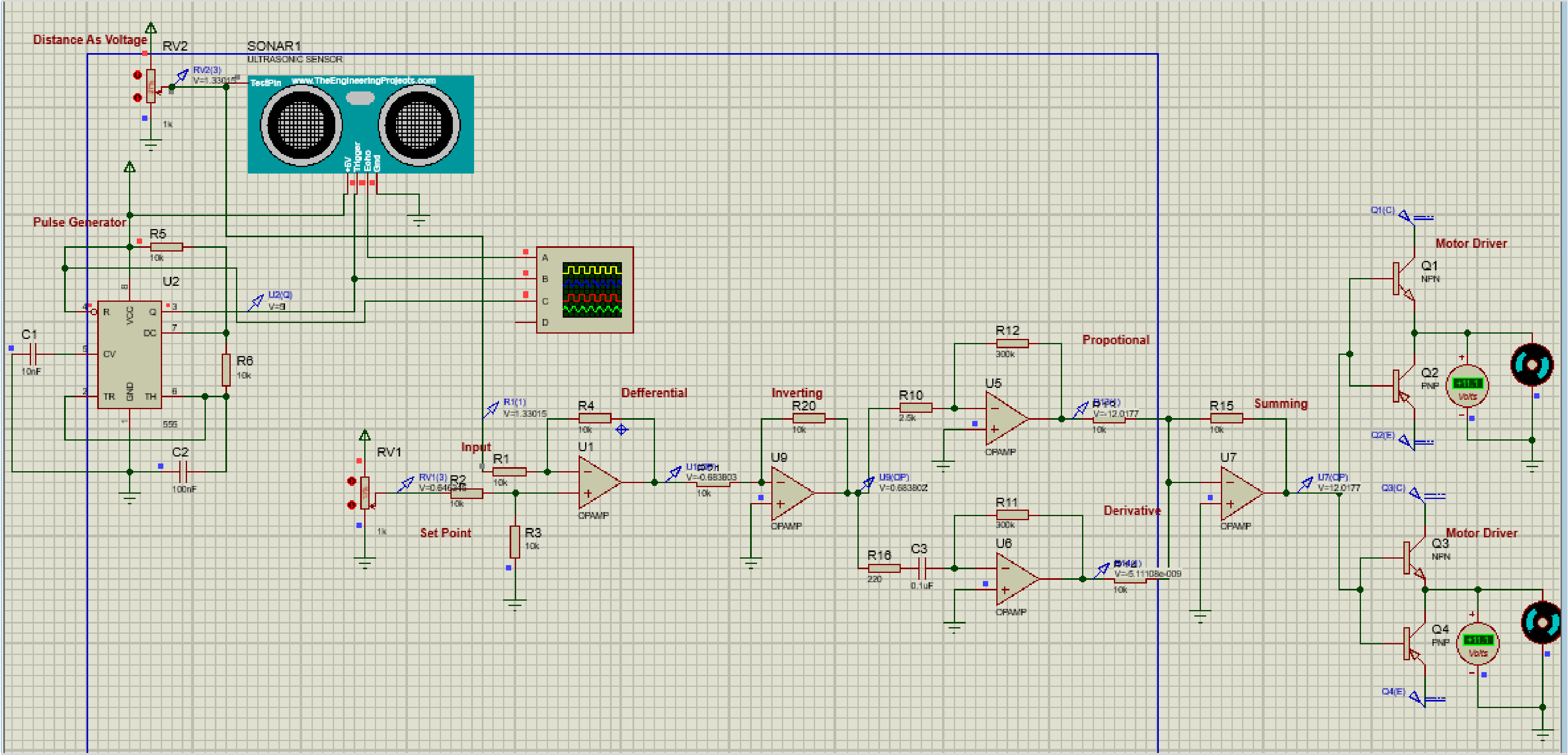


Differential circuit & PD circuit



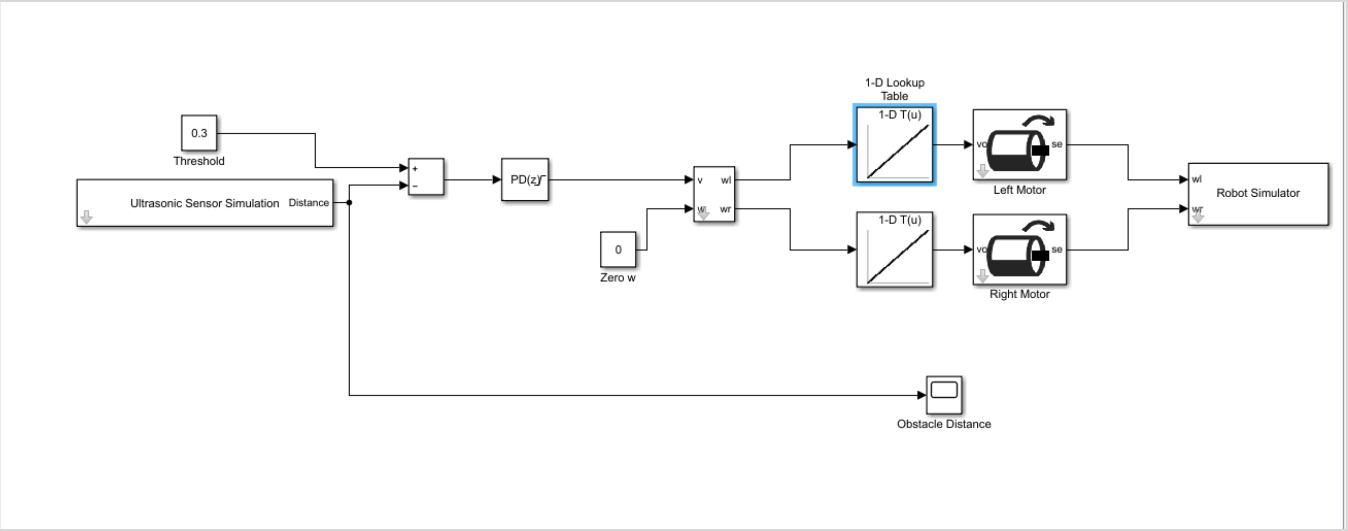
❖ motor driver circuit



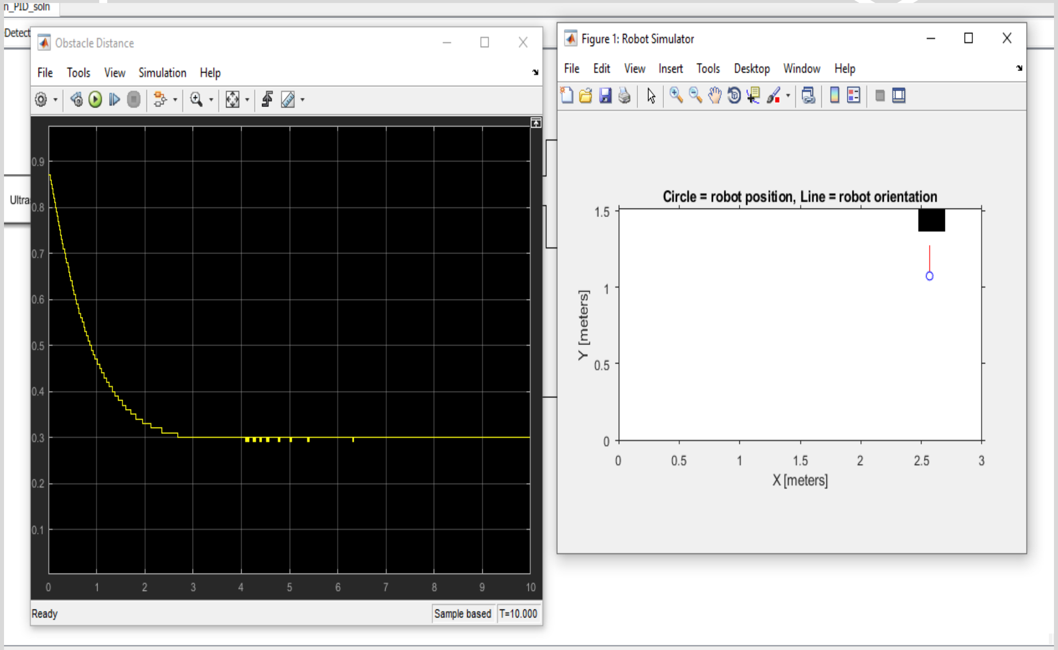


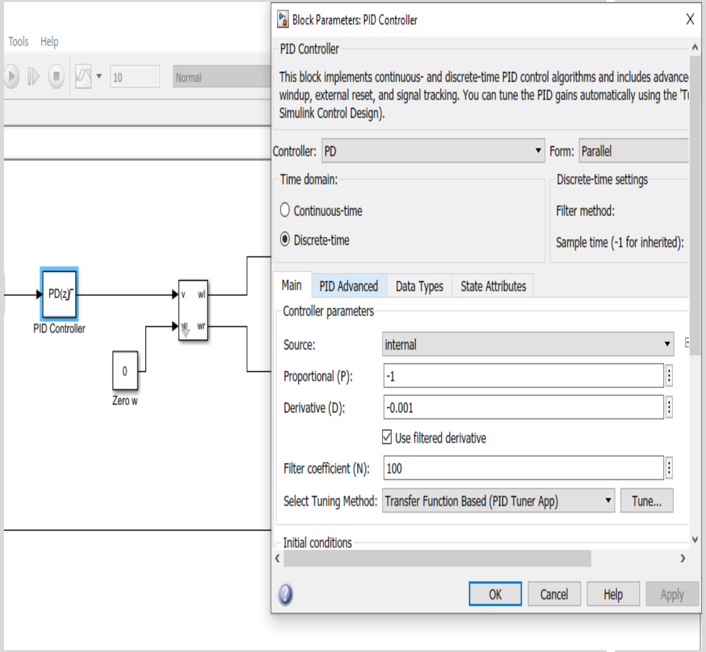
Distance As Voltage
RV2
1k
RV2(3)
V=1.33015
SONAR1
ULTRASONIC SENSOR
TechPin
www.TheEngineeringProjects.com
+5V
Trigger
Echo
Gnd
Pulse Generator
R5
10k
U2
R
VCC
Q
DC
CV
TR
GND
TH
555
U2(Q)
V=5
C1
10nF
R6
10k
C2
100nF
A
B
C
D
R1(1)
V=1.33015
Defferential
R4
10k
U1
OPAMP
Input
R1
10k
RV1
1k
RV1(3)
V=0.646248
R2
10k
Set Point
R3
10k
U1(OP)
V=-0.683803
Inverting
R20
10k
U9
OPAMP
10k
U9(OP)
V=0.683802
R10
2.5k
R12
300k
Propotional
U5
OPAMP
V=12.0177
10k
R11
300k
Derivative
R16
220
C3
0.1uF
U6
OPAMP
V=5.11108e-009
10k
R15
10k
Summing
U7
OPAMP
U7(OP)
V=12.0177
Q1(C)
Motor Driver
Q1
NPN
Q2
PNP
+11.1
Volts
Q2(E)
Q3(C)
Motor Driver
Q3
NPN
Q4
PNP
+11.1
Volts
Q4(E)

Running obstacle detection robot algorithms



Running obstacle detection robot algorithms





Block Parameters: PID Controller

PID Controller

This block implements continuous- and discrete-time PID control algorithms and includes advance windup, external reset, and signal tracking. You can tune the PID gains automatically using the 'Ti Simulink Control Design).

Controller: PD

Form: Parallel

Time domain:

○ Continuous-time

● Discrete-time

Discrete-time settings

Filter method:

Sample time (-1 for inherited):

Main | PID Advanced | Data Types | State Attributes

Controller parameters

Source: internal

Proportional (P): -1

Derivative (D): -0.001

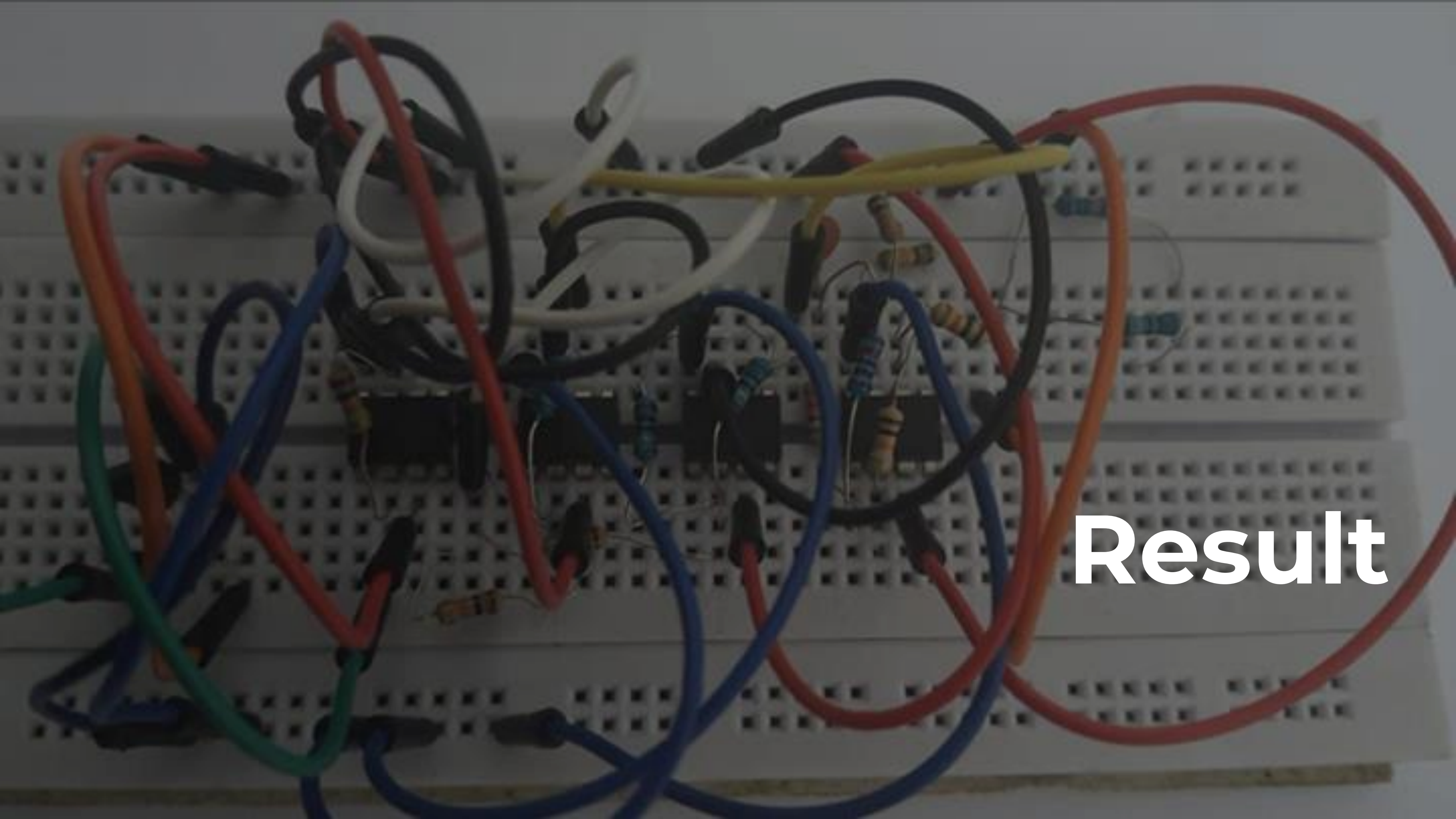
☑ Use filtered derivative

Filter coefficient (N): 100

Select Tuning Method: Transfer Function Based (PID Tuner App) Tune...

Initial conditions

OK Cancel Help Apply



Result

Problems:

1) Our system is analog and we use an ultrasonic sensor that needs pulses as trigger input and its output is a pulse also.

- sol: we have used timer 555 to generate pulses to the sensor .

2) PD circuit doesn’t provide suitable currents to motors .

- sol: we have built a driver circuit with transistors to amplify the current.

Problems:

3) Our actual motor that available has no datasheet so we couldn’t get the accurate parameters for the transfer function.

- sol: we tried to enter our college lab to get the motor response on the oscilloscope but the lab closed those days so we have contacted the older teams that used the same motor to get the response, then we have calculated the values from the graph .

4) We couldn’t do the hardware part Because we could not find some components.

- sol: we have simulated hardware on Matlab.

Proteus file

https://drive.google.com/file/d/1h3FvSPT3mQ4RD4tRz4jQOWNE3be6xeXS/view?usp=sharing

Matlab file

https://drive.google.com/file/d/1XYgZGSwNrPx8lR5AxnqWLNP-2U10VT0H/view?usp=sharing

Simulink file

https://drive.google.com/file/d/16cpODzTkMSjzoTstbLIIGV_dcuWJ2Md4/view?usp=sharing

Presentation Video

https://drive.google.com/file/d/1IE34Ru3oeWqXYwQARrZrd72ehUEmdYbb/view?usp=sharing

Thank You