PROYECTO COCHE SIGUELÍNEAS

COCHE 101

FECHA: 19/12/2023

JUAN MARÍN PONCE

CARLOS PADILLA RUANO

IBRAHIM PÉREZ DE LEÓN

DEPARTAMENTO DE INGENIERÍA DE SISTEMAS Y AUTOMÁTICA





ÍNDICE

1.	Introducción	2
	1.1. CONTEXTO DEL PROYECTO	2
	1.2. OBJETIVO DEL PROYECTO	2
2.	Descripción del código	3
	2.1. CONFIGURACIÓN DE PINES	
	2.2. PARÁMETROS DE CONTROL	4
	2.3. INICIALIZACIÓN DE BLUETOOTH	4
	2.4. CONFIGURACIÓN DE MOTORES	
	2.5. CONFIGURACIÓN DE SENSORES	E
	2.6. VELOCIDAD DEL COCHE Y ÁNGULO DE GIRO	<i>6</i>
3.	Problemas encontrados	7
	3.1. DIFERENCIA DE VELOCIDADES EN LAS RUEDAS	7
	3.2. PROBLEMAS CON LA PLACA DE SENSORES	
	3.3. POCA FIABILIDAD DEL COCHE	7
4.	Soluciones aplicadas	7
5.	Conclusión	{





1. Introducción

El desarrollo de un coche siguelíneas con control automático constituye un proyecto orientado a la implementación de sistemas de control y sensores para lograr que el vehículo siga una trayectoria predefinida en función de la detección de líneas en su entorno. En este documento se presenta una memoria detallada del proceso de diseño, implementación y las dificultades encontradas durante la ejecución del proyecto.

1.1. CONTEXTO DEL PROYECTO

El proyecto surge en el contexto de la robótica aplicada a la movilidad autónoma. El objetivo principal es desarrollar un sistema capaz de seguir líneas trazadas en el suelo mediante el uso de sensores infrarrojos y un controlador proporcional-integral (PI). Este tipo de aplicaciones encuentra su utilidad en diversos campos, desde la educación en robótica hasta la exploración autónoma de entornos específicos.

1.2. OBJETIVO DEL PROYECTO

Los objetivos fundamentales de este proyecto son:

- Diseñar e implementar un sistema de control PI que permita al coche siguelíneas ajustar su dirección para seguir una línea en el suelo.
- Utilizar sensores infrarrojos para la detección de líneas y la obtención de información sobre la posición relativa del vehículo respecto a la trayectoria deseada.
- Analizar y abordar los problemas inherentes al diseño, tales como las diferencias de velocidad entre las ruedas y la sensibilidad de los sensores a la iluminación ambiental.
- Evaluar la fiabilidad del coche siguelíneas y proponer soluciones para mejorar su desempeño en diversas condiciones ambientales y de funcionamiento.





2. Descripción del código

El código principal del proyecto se encarga de gestionar el funcionamiento del coche siguelíneas, combinando la lectura de sensores infrarrojos, el cálculo del ángulo de giro necesario, y la actuación sobre los motores para mantener el vehículo en la trayectoria deseada. A continuación, se detalla cada componente del código:

2.1. CONFIGURACIÓN DE PINES

En esta sección, se asignan los pines necesarios para el control de los motores y la comunicación Bluetooth. Se utilizan pines específicos para la entrada y salida de señales que controlan la velocidad y dirección de los motores, así como para la comunicación serial Bluetooth, esencial para la monitorización y depuración del sistema.

void setup() {
pinMode(pinIN1, OUTPUT);
pinMode(pinIN2, OUTPUT);
pinMode(pinENA, OUTPUT);
pinMode(pinIN3, OUTPUT);
pinMode(pinIN4, OUTPUT);
pinMode(pinENB, OUTPUT);
BT.begin(9600);
}





2.2. PARÁMETROS DE CONTROL

Los parámetros de control proporcional (kp), tiempo integral (Ti) y de tiempo de muestreo (Tm) son esenciales para el correcto funcionamiento del control PID. Estos valores determinan la respuesta del sistema ante las variaciones de error, la integral acumulativa y las correcciones derivativas.

```
const double kp = 4.3;

const double Ti = 1;

const double Tm = 0.3;

const double Td = 0;
```

2.3. INICIALIZACIÓN DE BLUETOOTH

Se establece una conexión Bluetooth mediante el uso de la biblioteca SoftwareSerial. Esto permite la comunicación inalámbrica con un dispositivo externo, en este caso una aplicación móvil, facilitando la configuración y monitorización del coche siguelíneas.

```
#include <SoftwareSerial.h>
SoftwareSerial BT(10, 11);
BT.begin(9600); //dentro de setup()
```

2.4. CONFIGURACIÓN DE MOTORES

Se definen los pines y parámetros para los motores A y B, que controlan las ruedas izquierda y derecha respectivamente. La función move se encarga de ajustar la dirección y velocidad de cada motor, asegurando un movimiento coordinado.

```
void move(const int pinMotor[3], int speed) {
  if (speed < 0) {
    digitalWrite(pinMotor[1], HIGH);
    digitalWrite(pinMotor[2], LOW);
    analogWrite(pinMotor[0], -speed);
    digitalWrite(pinMotor[0], -speed);
    analogWrite(pinMotor[1], LOW);
    digitalWrite(pinMotor[1], LOW);
    digitalWrite(pinMotor[1], LOW);
    digitalWrite(pinMotor[1], LOW);
    digitalWrite(pinMotor[2], HIGH);
    analogWrite(pinMotor[0], speed);
</pre>
```





2.5. CONFIGURACIÓN DE SENSORES

Se establece una función get_ang encargada de leer los valores de los sensores infrarrojos colocados en la parte inferior del coche. Estos sensores detectan la intensidad de la luz reflejada desde el suelo y proporcionan información sobre la posición relativa del coche respecto a una línea.

```
int get_ang()
{
  int umbral = 700;//Tensión umbral
  cochi_sensor[0] =analogRead(A5);
  cochi_sensor[1] =analogRead(A4);
  cochi_sensor[2] =analogRead(A3);
  cochi_sensor[3] =analogRead(A2);
  cochi_sensor[4] =analogRead(A1);
  cochi_sensor[5] =analogRead(A0);
  int valor [] = {0,1};
  for(int i = 0; i <6; i++){
    if (cochi_sensor[i] < umbral){
      cochi_sensor[i]=valor[0];
  }
}</pre>
```

```
else if(cochi_sensor[i] >= umbral){
  cochi_sensor[i]=valor[1];
int ang=0;
int n=0;
for(int i=0; i<6; i++)[
ang+=cochi_sensor[i]*10*(i+1);
n+=cochi_sensor[i];
ang=ang/n;
ang=ang-35;
if(n==0)[
 if(anga>=25){
 ang=30;
 else if(anga<=-25)[
 ang=-30;
return(ang);
```





2.6. VELOCIDAD DEL COCHE Y ÁNGULO DE GIRO

En el bucle principal (loop), se obtiene el ángulo de giro necesario a través de la función get_ang. Luego, se aplica el control PI mediante la función pid_control, y se ajusta la velocidad y dirección de los motores con la función set_speed. La información del ángulo se transmite a través de Bluetooth para su monitorización.

Este segmento de código principal establece la base para el control automático del coche siguelíneas, integrando la lectura de sensores, el control PI y la actuación sobre los motores para mantener una trayectoria predefinida.

```
void loop() {
 int angulo= get_ang();
 int d = pid_control(0, angulo); //velocidad de
giro
 set_speed(speed,d);
 BT.println(angulo);
 delay(2);
void set_speed(int v, int d) {
 int vi = constrain(v - d, -255, 255);
 int \ vd = constrain(v + d + 50, -255, 255);
if(vi<0)
vi=0:
if(vd<0)
vd=0:
 move(pinMotorA, vd);
 move(pinMotorB, vi);
```

```
int pid_control(double ref, double phi) {
 error = ref - phi;
 sum_ek+=error;
if (sum_ek > 150)//Maximo valor del integrador.
  sum_ek = 150;
 else if (sum_ek < -150)//Minimo valor de
integrador.
  sum_ek = -150;
 if(error<15&&error>-15)
 salida = kp*(error-sum_ek*Tm/Ti);
 else {
 salida = kp*(error+sum_ek*Tm/Ti);
 return salida;
```





3. Problemas encontrados

Durante la fase de desarrollo y pruebas del proyecto, se identificaron diversos desafíos que impactaron el rendimiento y la fiabilidad del coche siguelíneas. A continuación, se detallan los problemas encontrados:

3.1. DIFERENCIA DE VELOCIDADES EN LAS RUEDAS

Se observó una discrepancia en la velocidad de las ruedas, lo que generaba desviaciones en la trayectoria del coche. Aunque se implementó una solución provisional mediante la asignación de una mayor velocidad a una de las ruedas, esta medida no resolvió completamente el problema. La desigualdad en la velocidad de las ruedas afecta la capacidad de control del sistema, especialmente en curvas y tramos más complejos.

3.2. PROBLEMAS CON LA PLACA DE SENSORES

Los sensores infrarrojos utilizados resultaron ser altamente sensibles a las condiciones ambientales, principalmente a la iluminación del entorno. La falta de robustez en la sujeción de los sensores contribuyó a inconsistencias en las lecturas. Esto provocó que el umbral establecido en la función de cálculo del ángulo no fuera siempre suficiente, afectando la precisión y confiabilidad del sistema.

3.3. POCA FIABILIDAD DEL COCHE

Se observó una variabilidad significativa en el rendimiento del coche siguelíneas, incluso con el mismo conjunto de parámetros y condiciones iniciales. Pequeñas variaciones en la iluminación, el estado de las baterías u otros parámetros del entorno influenciaron de manera notable el comportamiento del vehículo. Esta falta de consistencia afecta la confianza en el sistema y su capacidad para operar de manera predecible y robusta.

4. Soluciones aplicadas

Durante la resolución de los desafíos encontrados en el desarrollo del coche siguelíneas, se aplicaron soluciones específicas para mejorar su rendimiento y confiabilidad. Para abordar la diferencia de velocidades entre las ruedas, se implementó una solución provisional asignando mayor velocidad a una de ellas, aunque se reconoce la necesidad de encontrar una solución más integral para lograr un equilibrio adecuado.





En respuesta a los problemas con la placa de sensores, se introdujo un umbral dinámico en la función de cálculo del ángulo, permitiendo adaptarse a las variaciones en las condiciones de iluminación. Además, se exploraron estrategias para mejorar la fiabilidad global, como el reforzamiento de la sujeción de los sensores y la consideración de técnicas de filtrado para suavizar las lecturas, con el objetivo de hacer que el sistema sea más robusto ante cambios ambientales.

Estas soluciones, aunque representan avances significativos, se reconocen como parte de un proceso continuo de mejora, y se espera que futuras iteraciones del diseño aprovechen los aprendizajes obtenidos durante este proceso de refinamiento. La adaptabilidad y ajuste constante de las estrategias implementadas son esenciales para lograr un coche siguelíneas más confiable y eficiente.

5. Conclusión

En conclusión, el desarrollo del coche siguelíneas con control automático ha proporcionado valiosas experiencias y desafíos. A pesar de los obstáculos encontrados, las soluciones aplicadas han contribuido a mejorar la estabilidad y confiabilidad del sistema. Este proyecto destaca la importancia de la adaptabilidad y la iteración continua para superar las complejidades asociadas con la movilidad autónoma. Las lecciones aprendidas sientan las bases para futuras mejoras, subrayando la necesidad de enfoques integrales y soluciones robustas en el diseño de sistemas robóticos.