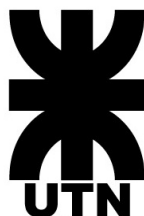


Universidad Tecnológica Nacional FRRO
Tecnologías para la Automatización



Trabajo Práctico Integrador – Robot sigue líneas

2024 - Comisión 402

Alumnos:

Nombre y Apellido	Legajo
Santiago Pedemonte	48069
Martín Glocer	48165
Alessandro Paolini (Com 403)	47999
Santiago Spini	49799
Mijael Nasatsky	48100

Docentes:

Cristian Bigatti
Gabriel Maurino
Hernán Garnica

Fecha Presentación: 26/07/2024

Índice

Abstract	3
Descripción breve del sistema a controlar	4
Información del Sistema	4
Funcionamiento del Sistema	5
Objetivos	5
Objetivo general	6
Objetivos específicos	6
Diseño	8
Materiales Utilizados	8
Descripción del Proceso	8
Diseño en Tinkercad	9
Código en Arduino	9
Logros y Resultados	13
Conclusiones	13

Abstract

En este informe se presenta el desarrollo y la implementación de un vehículo autónomo de seguimiento de líneas utilizando la plataforma Arduino. El proyecto se enmarca dentro del trabajo práctico integrador (TPI) de la asignatura Tecnologías para la Automatización de la carrera de Ingeniería en Sistemas de Información en la UTN - Facultad Regional Rosario.

El objetivo principal de este proyecto es diseñar y construir un coche capaz de seguir una línea negra sobre una superficie blanca, utilizando para ello sensores de reflectancia y un sistema de control basado en un microcontrolador Arduino Uno R3. El vehículo integra diversos componentes electrónicos, entre los que destacan un driver de motor L298N, motores de corriente continua (CC), sensores de reflectancia TCRT5000, y una estructura de chasis que aloja todos los elementos necesarios.

El funcionamiento del sistema se basa en la detección de la línea mediante los sensores de reflectancia, que emiten luz y leen la luz reflejada para distinguir entre superficies blancas y negras. Estos sensores envían la información al microcontrolador Arduino, el cual procesa los datos y ajusta la velocidad y dirección de los motores a través del driver de motor. De esta manera, se consigue que el vehículo corrija su trayectoria y se mantenga sobre la línea negra de forma autónoma.

El desarrollo del proyecto se llevó a cabo siguiendo una metodología estructurada que incluyó la fase de diseño del circuito utilizando la herramienta Tinkercad, la construcción y ensamblaje del chasis, el cableado de los componentes y la programación del microcontrolador. Se realizaron pruebas exhaustivas para asegurar la precisión y eficiencia del sistema, evaluando su capacidad para seguir la línea en diferentes condiciones.

Los resultados obtenidos demuestran que el vehículo es capaz de seguir la línea con una alta precisión y eficiencia, respondiendo adecuadamente a las desviaciones de la trayectoria. Este proyecto no solo permite aplicar conceptos teóricos de sistemas de control y automatización, sino que también proporciona una base práctica para el desarrollo de sistemas robóticos más complejos.

El trabajo finalizado refleja la integración exitosa de hardware y software para la creación de un sistema de control autónomo, destacando la versatilidad y capacidad de la plataforma Arduino para proyectos de automatización y robótica. Las lecciones aprendidas y los conocimientos adquiridos en este proyecto son de gran valor para el futuro desarrollo profesional en el campo de la ingeniería en sistemas de información.

Descripción breve del sistema a controlar

El problema principal a resolver en este proyecto es el diseño e implementación de un sistema regulador para un vehículo autónomo de seguimiento de líneas. Un sistema regulador se encarga de mantener una variable de salida constante en presencia de perturbaciones, lo cual es crucial en nuestro caso para asegurar que el vehículo siga una trayectoria definida por una línea negra sobre una superficie blanca.

Información del Sistema

El sistema a controlar está compuesto por los siguientes componentes:

- Microcontrolador Arduino Uno R3: actúa como la unidad de control central, procesando las señales de entrada de los sensores y enviando comandos a los motores.
- Driver de motor L298N: este driver controla la velocidad y dirección de los motores de corriente continua (CC). Permite manejar motores con tensiones de 6V a 50V y corrientes de hasta 2A por canal.
- Motores de corriente continua (CC): dos motores CC se utilizan para proporcionar movimiento al vehículo. Estos motores están controlados por el driver de motor, permitiendo ajustar la velocidad y dirección de cada motor individualmente.
- Sensores TCRT5000: tres sensores de reflectancia se utilizan para detectar la línea negra. Estos sensores emiten luz infrarroja y miden la cantidad de luz reflejada para distinguir entre superficies blancas y negras. La disposición típica incluye un sensor central y dos sensores laterales.

- Chasis del vehículo: una estructura que aloja todos los componentes. En nuestro caso está hecho de plástico.
- Ruedas y rueda giratoria: dos ruedas controladas por motores y una rueda giratoria en la parte trasera para estabilidad.
- Batería de 9v como fuente de energía.
- Cables conectores.
- 3 resistencias de 10k Ω y 3 resistencias de 330 Ω

Funcionamiento del Sistema

El sistema funciona detectando la posición de la línea negra respecto al vehículo mediante los sensores de reflectancia. Los sensores envían señales al Arduino, que las procesa para determinar si el vehículo está siguiendo la línea correctamente o si necesita ajustar su trayectoria. Basado en esta información, el Arduino envía comandos al driver de motor para ajustar la velocidad y dirección de los motores, corrigiendo la trayectoria del vehículo.

Este sistema regulador debe ser capaz de:

- Detectar Desviaciones: Identificar cuando el vehículo se desvía de la línea
- Corregir la Trayectoria: Ajustar la velocidad y dirección de los motores para volver a alinear el vehículo con la línea.
- Mantener la Estabilidad: Asegurar que el vehículo se mantenga sobre la línea a pesar de las perturbaciones.

Esto garantiza que el vehículo siga la línea de manera eficiente y precisa, demostrando la aplicabilidad de los conceptos de regulación y control en un contexto práctico de automatización y robótica.

Objetivos

El proyecto del vehículo autónomo de seguimiento de líneas tiene como objetivo principal diseñar e implementar un sistema regulador que permita al vehículo seguir una línea negra sobre una superficie blanca de manera autónoma y precisa.

Objetivo general

- Diseñar y desarrollar un vehículo autónomo de seguimiento de líneas utilizando la plataforma Arduino y sensores TCRT5000.

Para alcanzar este objetivo general, se han definido los siguientes objetivos específicos:

Objetivos específicos

- Utilizar sensores TCRT5000 para detectar la presencia de la línea negra y enviar datos precisos al microcontrolador Arduino.
- Configurar y calibrar los sensores TCRT5000 para garantizar una detección confiable en diferentes condiciones.
- Controlar la velocidad y dirección de los motores:
 - Integrar un driver de motor L298N para controlar la velocidad y dirección de los motores de corriente continua (CC).
 - Desarrollar el código necesario en Arduino para procesar las señales de los sensores y ajustar la velocidad y dirección de los motores de manera precisa.
- Diseñar el chasis del vehículo:
 - Construir un chasis que soporte todos los componentes electrónicos y mecánicos del vehículo.
 - Asegurar que el diseño del chasis permita una distribución adecuada del peso y una estabilidad óptima del vehículo.
 - Garantizar que los sensores permanezcan estables y no se muevan fácilmente.
- Desarrollar la programación del sistema de control:
 - Escribir y depurar el código en Arduino para implementar el algoritmo de seguimiento de líneas, asegurando una respuesta rápida y eficiente a las desviaciones de la trayectoria, ayudándonos de un controlador PID.
 - Utilizar la plataforma Tinkercad para diseñar y simular el circuito antes de la implementación física, facilitando la detección y corrección de posibles errores.
- Realizar pruebas y ajustes del sistema:

- Evaluar el rendimiento del vehículo en diferentes escenarios, incluyendo variaciones en la iluminación y la superficie.
- Realizar ajustes y calibraciones en el sistema de control y los sensores para mejorar la precisión y eficiencia del seguimiento de líneas.
- Modificar los valores de las constantes del controlador PID en función de la salida que observamos.

Diseño

Materiales Utilizados

- Microcontrolador Arduino Uno R3
- Driver de motor L298N
- 2 Motores de corriente continua
- 3 Sensores TCRT5000
- Chasis del vehículo
- 2 Ruedas
- 1 Rueda giratoria
- 1 batería de 9V
- Cables conectores
- 3 Resistencias de 10k Ω
- 3 Resistencias de 330 Ω

Descripción del Proceso

Detección de la línea:

Al encender el dispositivo, los sensores infrarrojos TCRT5000 detectan la luz reflejada de la superficie en la que está colocado el vehículo.

Los sensores emiten una luz infrarroja que se refleja contra el suelo; si el suelo es blanco o de un color claro, la luz se refleja, mientras que si es oscuro, no lo hará.

Los sensores captan este reflejo y envían la información a la placa Arduino.

Procesamiento de datos:

La placa Arduino procesa los datos de los sensores para determinar si el vehículo está siguiendo la línea correctamente o si necesita ajustar su trayectoria.

Ajuste de trayectoria:

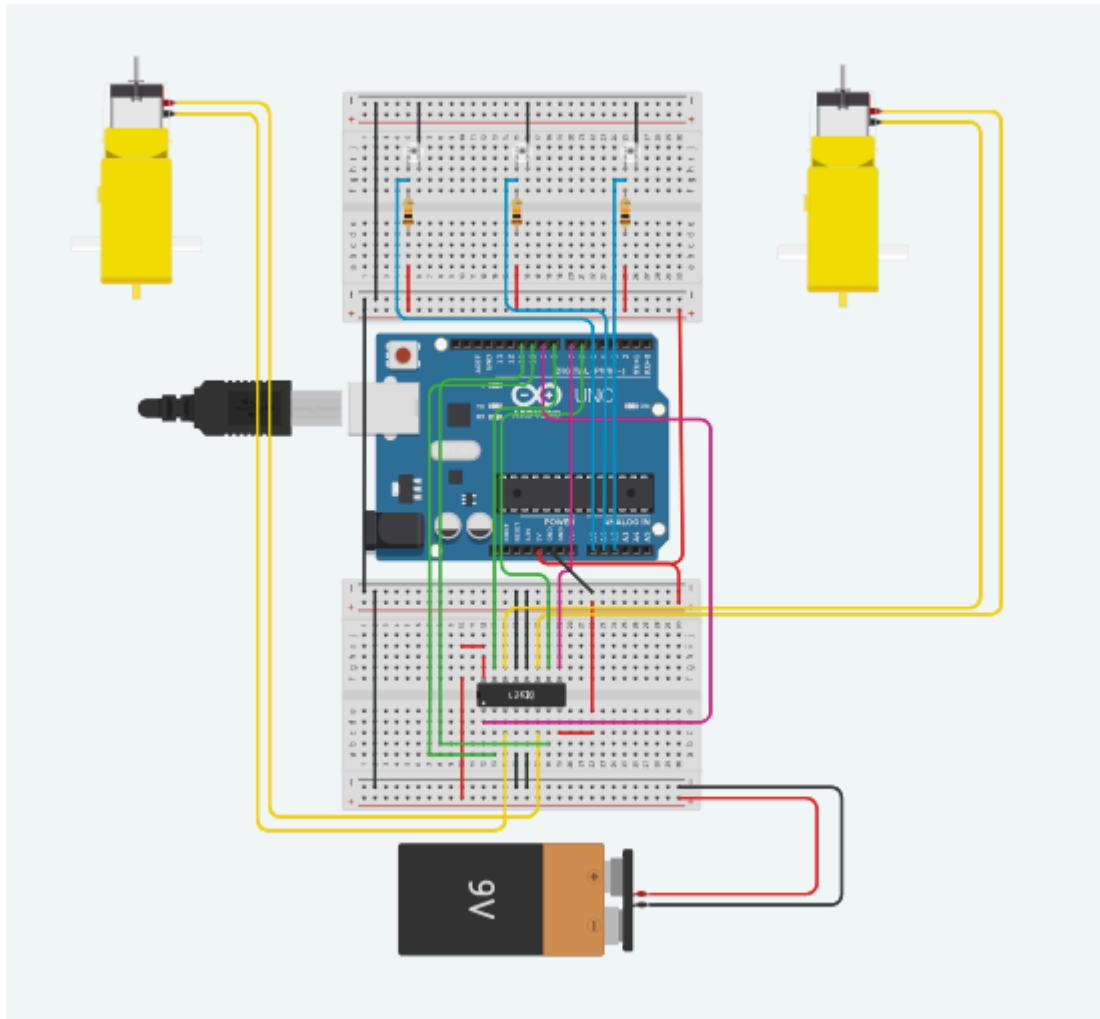
Basado en la información recibida, el Arduino envía comandos al driver de motor L298N para ajustar la velocidad y dirección de los motores, corrigiendo así la trayectoria del vehículo.

Simulación y pruebas:

Utilizando Tinkercad, se diseña y simula el circuito para detectar y corregir posibles errores antes de la implementación física.

Se realizan pruebas y ajustes en diferentes condiciones para asegurar la precisión y eficiencia del sistema de seguimiento de líneas.

Diseño en Tinkercad



Nota: en el Tinkercad, al no existir el driver L298N, tuvimos que representar inicialmente el modelo con el circuito integrado L293D. La existencia de 2 protoboards en el Tinkercad se debe simplemente a que el modelo sea más fácil de interpretar y que no se junten muchos cables. En nuestro hardware final usamos los mismos pines pero adaptados a lo que requiere un controlador de motores L298N.

A continuación dejamos el link que direcciona al Tinkercad:

<https://www.tinkercad.com/things/cZR7QITOfzb-tpi-2024?sharecode=Kpli3Vyx1jsHM-t-XUk2WI9Q786CR5ORw6B4METqi2E>

Código en Arduino

```

/*
✓ Robot Sigue Linea:
  - En base al error de los sensores externos establece los valores del PID
  - En base a los valores del PID establece cuanto gira cada motor

✓ Tener en cuenta:
  - Los sensores miden que tan negro o que tan blanco es la superficie (no son binarios).
  - Cuanto mas blanco mayor el error

*/

// === Establecemos Velocidades
int baseSpeed = 200;
int maxSpeed = 255;
int minSpeed = 75;

// FineTuning del PID
float Kp2 = 0.0008; //Kp = Kp2 * (1000 - leer_sensor(1));
float Kd2 = 0.1; //Kd = Kd2 * Kp;
float Ki2 = 0.0001; // Ki = Ki2

// === Desde aca es todo codigo, no se deberia tener que modificar nada =====

✓ // Pin definitions
  // Motores
const int AIA = 3;
const int AIB = 11;
const int BIA = 10;
✓ const int BIB = 6;
  // Sensores
const int irIzquierda = 9;
const int irDerecha = 8;
const int irCentro = 7;
const int ledTest = 13;

// === Sensor pins =====
// A0 -> Sensor Derecha
// A1 -> Sensor Medio
// A2 -> Sensor Izquierda
const int sensorPins[3] = {A0, A1, A2};
const int irSensor[3] = {irDerecha, irCentro, irIzquierda};

// === Inicializacion de variables =====
int P, D, I = 0, previousError = 0, PIDvalue, error;
int lsp, rsp;

float Kp = 0; // Inicializadas en cero por si son comentadas mas adelante tome este valor y no de error
float Kd = 0;
float Ki = 0;

int ref; // Para la funcion leer_sensor

```

```
// === Setup =====
void setup(){

    pinMode(13, OUTPUT); // ledTest
    digitalWrite(13, LOW);

    // Inicializamos los sensores
    pinMode(irIzquierda, OUTPUT);
    pinMode(irDerecha, OUTPUT);
    pinMode(irCentro, OUTPUT);

    // Inicializamos los motores
    pinMode(AIA, OUTPUT);
    pinMode(AIB, OUTPUT);
    pinMode(BIA, OUTPUT);
    pinMode(BIB, OUTPUT);

    // Apagamos los sensores
    digitalWrite(irIzquierda, LOW);
    digitalWrite(irDerecha, LOW);
    digitalWrite(irCentro, LOW);

    // Iniciamos los motores como apagados (para evitar movimientos no deseados)
    digitalWrite(AIA, LOW);
    digitalWrite(AIB, LOW);
    digitalWrite(BIA, LOW);
    digitalWrite(BIB, LOW);

    // Initial blink
    digitalWrite(ledTest, LOW);
    delay(500);
    digitalWrite(ledTest, HIGH);
    delay(500);
    digitalWrite(ledTest, LOW);
    delay(500);

    Serial.begin(9600);
}
```

```
// === Loop =====
void loop(){

    if (leer_sensor(1) > 450){ // Si el sensor del medio es mayor a 450(Negro)

        /* Otra forma de hacerlo es establecer valores fijos en la declaracion de variables */
        Kp = Kp2 * (1000 - leer_sensor(1)); // Si esta centrado, kp sera casi 0
        Kd = Kd2 * Kp;
        Ki = Ki2;

        linefollow();
    }
}
```

```
// === LineFollowFunction =====
void linefollow(){

    int error = ((leer_sensor(2)) - leer_sensor(0));

    // Calculo valores segun el error
    P = error;
    I = I + error;
    D = error - previousError;

    PIDvalue = (Kp * P) + (Ki * I) + (Kd * D);
    previousError = error;

    // Asigna velocidades a los motores | Se resta o suma dependiendo de cual restamos para el error
    lsp = baseSpeed - PIDvalue;
    rsp = baseSpeed + PIDvalue;

    // Establece valores maximos y minimos en caso de ser necesario
    if (lsp > maxSpeed) {
        lsp = maxSpeed;
    }
    if (lsp < minSpeed) {
        lsp = minSpeed;
    }
    if (rsp > maxSpeed) {
        rsp = maxSpeed;
    }
    if (rsp < minSpeed) {
        rsp = minSpeed;
    }

    // Asignamos velocidad a los motores
    analogWrite(AIB, lsp);
    analogWrite(BIA, rsp);
}

// === Leer Sensor =====
int leer_sensor(int i){
    digitalWrite(irSensor[i], HIGH);
    ref = analogRead(sensorPins[i]);
    digitalWrite(irSensor[i], LOW);

    return ref;
}
```

El código se encuentra adjunto (ta_tpi_2024.ino).

Logros y Resultados

- Diseño y Construcción del Sistema:
 - Se ha diseñado y construido un chasis adecuado que alberga todos los componentes necesarios, logrando una estructura estable y funcional.
 - La integración de los sensores TCRT5000 y el driver de motor L298N ha permitido un control preciso de la trayectoria del vehículo.
- Implementación del algoritmo de control:
 - Se ha desarrollado un algoritmo eficiente para el seguimiento de líneas, implementado en el microcontrolador Arduino Uno R3, que procesa las señales de los sensores y ajusta la velocidad y dirección de los motores en tiempo real con un controlador PID.
 - Las pruebas realizadas han demostrado que el sistema puede detectar desviaciones de la línea y corregir la trayectoria del vehículo de manera rápida y precisa.

Dejamos el video de la prueba de nuestro proyecto adjunto.
(video_robot_sigue_linea.mp4)

Conclusiones

El desarrollo del vehículo autónomo de seguimiento de líneas ha sido un ejercicio integral que ha permitido la aplicación de conocimientos teóricos y prácticos en el ámbito de la automatización y los sistemas de control. A lo largo de este proyecto, se han alcanzado con éxito los objetivos planteados, demostrando la viabilidad y eficacia de utilizar la plataforma Arduino junto con sensores de reflectancia y drivers de motor para crear un sistema autónomo de seguimiento de líneas.

El proyecto ha resaltado la capacidad de integración de diversos componentes electrónicos y ha permitido experimentar con la programación y calibración de sensores para obtener un comportamiento preciso y confiable del vehículo. Además, el uso de herramientas de simulación como Tinkercad ha facilitado la identificación y corrección de errores antes de la implementación física, optimizando el proceso de desarrollo.

Este proyecto no solo ha validado la utilidad de la plataforma Arduino en aplicaciones de automatización y robótica, sino que también ha proporcionado una base sólida para futuros desarrollos en sistemas de control autónomos, reforzando las competencias técnicas y prácticas necesarias en el campo de la ingeniería en sistemas de información.

Encontramos enriquecedor poder ver lo estudiado sobre controladores y sistemas de control en la teoría aplicado prácticamente.