PRÁCTICA CONTROL PID EN ROBOT SIGUELINEAS.

Introducción

Los AGV o vehículos guiados automáticamente son equipos móviles destinados al transporte automático de material dentro de las plantas industriales. Su capacidad de reprogramación les permite variar la trayectoria y posicionamiento dotándoles de gran versatilidad y flexibilidad. Este es el motivo por el cual su implantación está creciendo enormemente en los últimos años en los entornos de producción.

Objetivos generales

La práctica que se plantea tiene por objetivo el desarrollar un sistema de control básico para un vehículo de este tipo. Concretamente, se trata de realizar el sistema de control PID para que el mini robot móvil siga una línea marcada en el suelo.

Material

Para la realización de la práctica se empleará un pequeño robot móvil dotado básicamente de dos ruedas activadas por sendos motores controlados y un conjunto de sensores infrarrojos destinados a detectar la posición relativa del robot respecto a la línea a seguir. En base a esta información el robot corregirá la posición activando de forma controlada un motor u otro para mantenerse siempre posicionado sobre la línea.

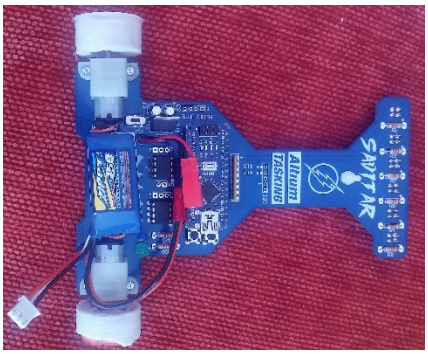


Fig.1. Mini robot que se utilizará en la práctica

El sistema de control se ejecutará en una tarjeta basada en microcontrolador de la familia Arduino. Arduino es una plataforma de prototipado electrónico de código abierto que facilita en gran medida el uso y programación de microcontroladores. Aunque para la programación se pueden utilizar varios lenguajes, el estándar es el C++, lenguaje con el cual el alumno se encuentra ya familiarizado pues es el utilizado en la asignatura de primer curso Fundamentos de Informática.

Además, el entorno que ofrece Arduino puede ser utilizado desde los sistemas operativos más utilizados: Windows, Mac OS y Linux.

Objetivos detallados

En esta práctica emplearemos un robot sigue-líneas. Este constituye un buen ejemplo para aplicar la teoría de control pues vamos a **desarrollar un control PID para un vehículo que se guiará automáticamente siguiendo una línea marcada en el suelo**.

Estructura del robot

Para controlar el robot (planta) este tiene integrados los tres elementos fundamentales de todo sistema controlado: **sensores** (seis CNY70), **controlador** (Arduino Nano) y **actuadores** (dos motores, uno para cada rueda) (figura 2).

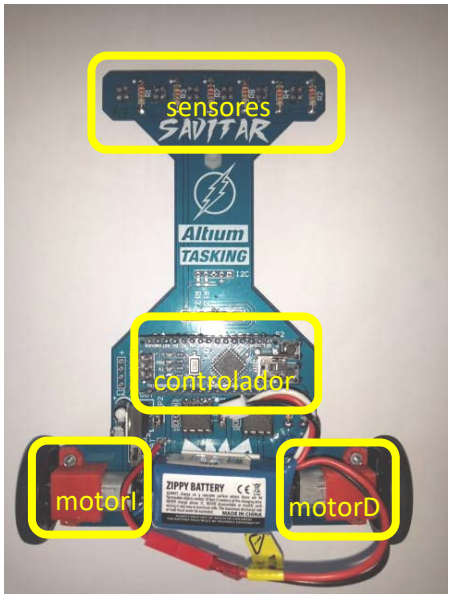
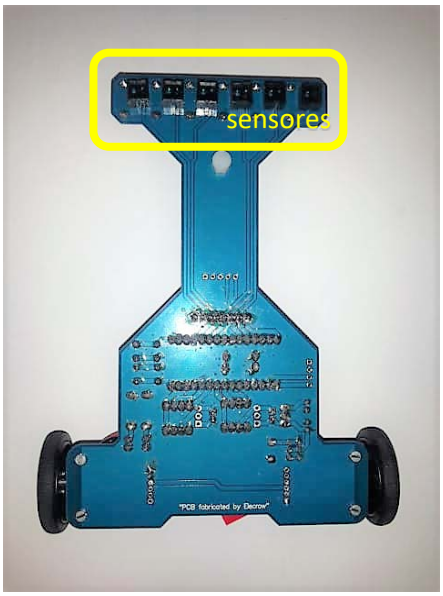


Figura 2.*Estructura del robot siguelíneas*

Sensores

En la parte frontal el robot integra una fila de **seis sensores** CNY70 (figura 2). Cada sensor está formado por una fuente de luz infrarroja (diodo emisor) y un fototransistor (receptor). El receptor captará la reflexión del diodo emisor solo cuando el sensor esté dirigido hacia una superficie clara devolviendo un valor bajo de tensión. Si el sensor se encuentra sobre la línea oscura no habrá reflexión, el fototransistor no conducirá y la tensión será alta (figura 3).



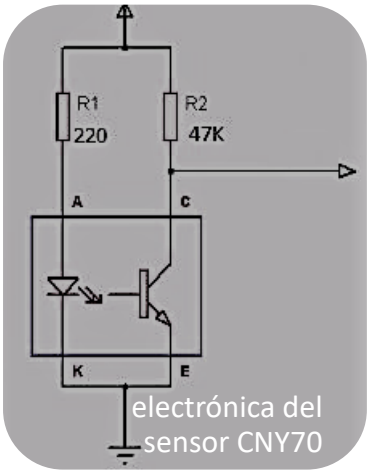


Figura 3. *El chasis del robot está constituido por la misma placa que soporta los componentes electrónicos y pistas. En la parte inferior están los seis sensores CNY70 que irán indicando la posición de la línea respecto al robot.*

Control PID

El robot debe intentar posicionarse sobre la línea marcada en el suelo de forma que esta se sitúe sobre el eje del robot. Puede avanzar y girar controlando ambos motores que actúan de forma independiente. Por ejemplo, si tiene que girar a la izquierda el control hará girar motor derecho más rápido que el izquierdo. Si tiene que corregir a la derecha hará lo contrario. Si los dos motores actúan con la misma velocidad el robot irá recto. Esta técnica se llama *tracción diferencial* y es típica en vehículos con orugas.

La **señal de error** en nuestro caso va a cuantificar la diferencia entre la posición de la línea y el eje del robot. Cuanto más lejos esté esta línea del eje mayor será el error en valor absoluto. El error tiene signo. Nuestro objetivo será controlar el giro de los motores de las ruedas para que la línea aparezca siempre entre los dos sensores centrales de la fila (figura 4).

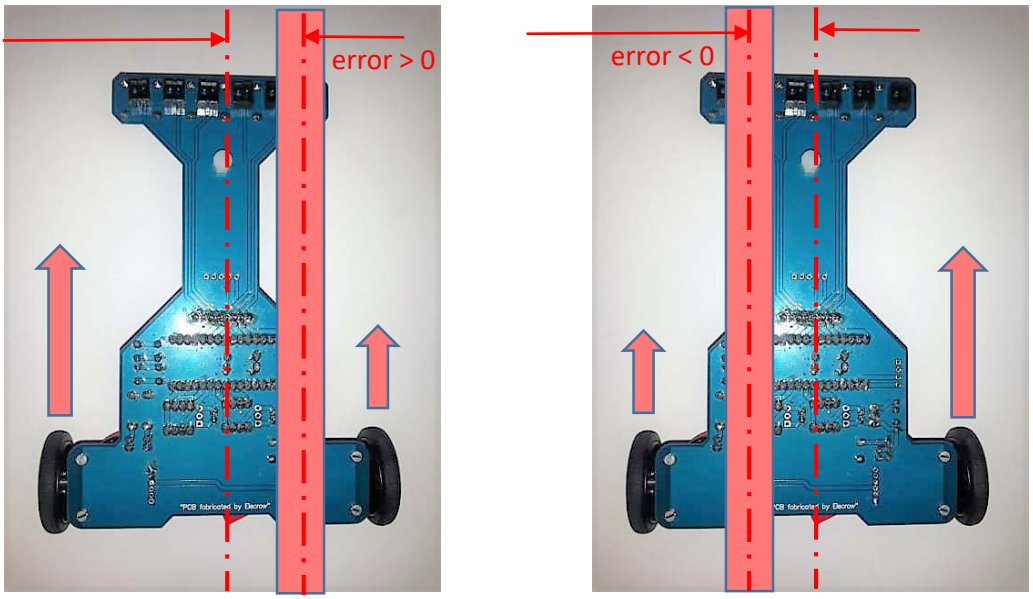


Figura 4. *Las velocidades de los motores se controlarán en base al error (diferencia entre la posición de la línea con el eje del robot)*

Una estrategia que podemos seguir para que el robot se posicione siempre sobre la línea, incluso en zonas de curvas es, que aplicar un voltaje adicional al motor al que se acerca la línea y quitárselo al motor más alejado. Parece razonable que este voltaje sea proporcional al error. Esto es lo que se conoce como **control proporcional**:

$$\Delta V = K_p \cdot e$$

Esta sencilla estrategia funciona perfectamente cuando la velocidad del robot no es elevada. Si el robot tiene que desplazarse rápidamente tendríamos que elevar la constante Kp para que pueda reaccionar a cambios bruscos en la dirección. Sin embargo, subir el valor de la constante Kp hace que el robot tienda a sobre-corregir produciendo grandes oscilaciones que le pueden llevar a sacarle de la línea. (Ver vídeos *videov80Kp45Kd0.mp4* y *oscilacionProporcionalCamaraLenta.mp4)*

Podemos mejorar la evolución del robot añadiendo al controlador una **acción de control derivativa.** Es decir, una acción que responda en base, no a la magnitud del error como la parte proporcional, sino a la velocidad con la que cambia el error.

$$\Delta V = K_d \cdot de/dt$$

Esta parte derivativa es muy adecuada para sistemas que deban actuar con rapidez como nuestro robot pues el controlador se anticipa a la propia señal del error. La parte derivativa mejorará por tanto la velocidad de respuesta en nuestro robot. (Ver vídeo *videov80Kp12Kd10.mp4*).

Plan de trabajo.

- Implantar un control proporcional para el robot. Comentar qué efectos se producen sobre el robot aumentar la constante proporcional.
- Implantar un control proporcional-derivativo PD. ¿Se puede implantar un control derivativo solo?
- Investiga si sería interesante implantar en el robot un control de tipo integral.

Referencias

http://elm-chan.org/works/ltc/report.html

http://crm-uam.github.io/actividades/2012_taller_arduino/Control_PD_Siguelineas.pdf

https://www.arduino.cc/