Informe laboratorio #10

Daniel Avila
Yeison Lara
Cristian Mora

¿Qué se usó?

Robot móvil ensamblado con sus motores y sensores

Puerto ajustado para UDP ESP8266

Programador STLink

Teleplot o Qt para visualización

¿Qué se realizó?

Diseñamos e implementamos un perfil de velocidad trapezoidal dinámico, al igual que en el laboratorio anterior, para desplazamientos lineales y giros sobre el eje.

Creamos la librería PID_MM.c con dos controladores PID independientes (lineal y angular), cada uno con sus propias constantes Kp y Kd, y saturación en el PID angular para evitar picos de señal.

En la rutina principal, el robot ejecuta en bucle una cola de cinco movimientos activada por un botón físico.

Nueva funcionalidad: incorporamos un movimiento en curva usando una combinación de división y multiplicación de valores de velocidad:

Para cada valor de referencia, aplicamos una división por un factor α durante el inicio de la curva y luego una multiplicación por β al salir, con el fin de simular la aceleración interna y externa de la curva.

Implementamos las operaciones directamente en la función trapecio_perpetuo, ajustando los parámetros de velocidad en tiempo real.

Cada 0.005 s el robot transmite por UART (921 600 bps) la velocidad deseada, la velocidad real y las salidas PID para graficarlas en Teleplot.

¿Qué sucedió?

Al presionar el botón, el robot avanzó según el perfil trapezoidal estándar y, cuando correspondía, entró en el segmento de curva aplicando los factores de división/multiplicación.

El PID lineal corrigió desviaciones en la velocidad de avance, y el PID angular mantuvo la orientación en los giros sobre sí mismo.

Durante el tramo de curva, sin embargo, la combinación de operaciones aritméticas no produjo el efecto deseado: el robot tendía a ralentizarse demasiado o a acelerar bruscamente, sin mantener una trayectoria suave.

Tras completar la serie de cinco movimientos, el sistema volvió automáticamente a espera sin fallos de comunicación.

Complicaciones

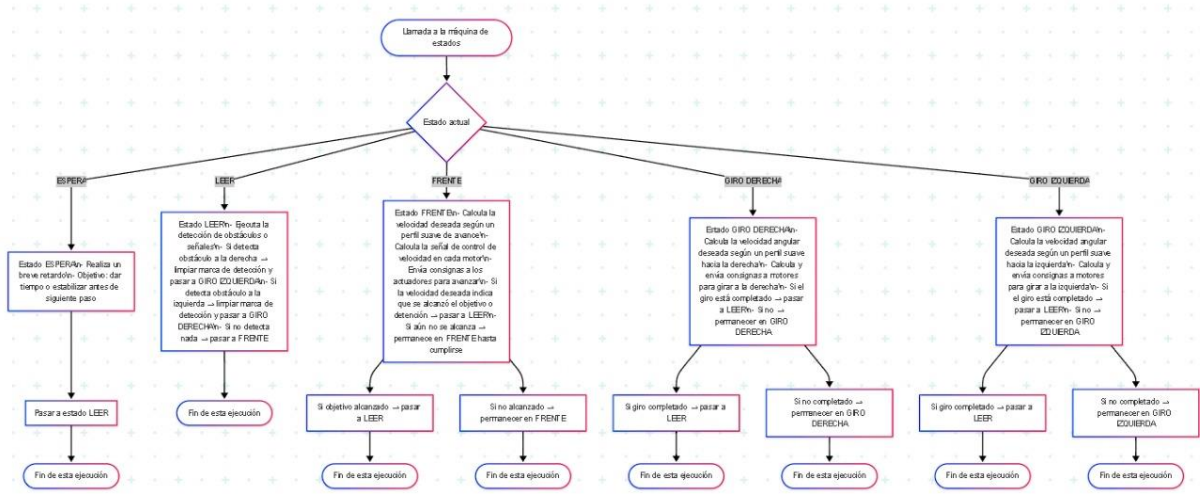
Inestabilidad en la curva: la simple división y multiplicación lineal de la velocidad no logró modelar correctamente la cinemática de un giro continuo, pero si queremos resaltar que para lógralo tuvimos en cuenta el trapecio angular de nuestro código, para saber la theta de ese momento, cosa que desborda y no fue nada preciso

Saturación del PID lineal: al reducir demasiado la referencia en el inicio de la curva, el controlador actuó con un error muy grande, por ello utilizamos un porcentaje más pequeño ya que provocaba oscilaciones en movimiento

Sincronización de operaciones: los cálculos aritméticos en tiempo real introdujeron retrasos de hasta 2–3 ms, afectando la resolución de la señal de control a 0.005 s.

Falta de modelo geométrico: al no incorporar parámetros de radio de giro ni una trayectoria basada en coordenadas, el efecto “curva” fue meramente escalar y no espacial.

Diagrama de flujo



Conclusión

Para futuros trabajos, proponemos en primer lugar elaborar un modelo geométrico de trayectoria circular basado en coordenadas polares y en una odometría correctamente calibrada, de modo que el robot calcule en todo instante el radio de giro y la ruta real del arco en el plano (en lugar de escalar linealmente la velocidad). Para ello incorporaremos el concepto de “giro de Kojima” —tal como se explica en el texto complementario que estudiamos—, que plantea generar un movimiento continuo con curvas infinitas mediante una máquina de estados y la manipulación de θ: basta con definir el número de grados que debe cambiar θ (por ejemplo, un giro en U de $180\,^\circ \simeq 3.14$ rad) y mantener simultáneamente una velocidad lineal constante para describir el arco deseado sin detener el avance. A continuación integraremos un término de feedforward angular en el PID, de forma que justo antes de entrar en el sector curvo el controlador compense anticipadamente la desviación esperada, reduciendo sobre impulsos y oscilaciones al iniciar y salir del giro. Finalmente, simplificaremos la ejecución acotando la llamada de control a un intervalo fijo de 5 ms.