



Pontificia Universidad Católica
Facultad de Ingeniería
IRB2001 - 1
Profesor: David Acuña

Tarea 3 - Taller 3: Control PID

Matías Moreno F.
Jaime Tagle D.
Santiago Larraín C.
05/05/2022

1. La evolución del código:

Al principio se utilizó el error como la diferencia entre la posición del auto y la referencia, pero rápidamente nos percatamos que este valor varía entre -3500 y 3500 , por lo que realizamos un escalado con una regla de 3 para que se mueva entre $-Power_{max}$ y $Power_{máx}$, siendo $Power_{máx}$ la potencia que se le suministraba a los motores, la cual puede variar entre -400 y 400 . Para el controlador, se utilizaron las ecuaciones dadas por el profesor en clases para K_0 , K_1 y K_2 , las cuales dependen de K_p , K_i y K_d ; Al comienzo, solo trabajamos con K_p , es decir, consideramos solamente el control proporcional.

Notamos que el auto seguía la línea, pero observamos una lentitud en el ajuste al error para distancias a la línea negra más grandes, por lo que optamos por usar curvas no lineales en el error $\tan\left(\frac{error}{ref}\right)$ o $\arcsin\left(\frac{error}{ref}\right)$, las cuales tuvieron una respuesta con mayor rapidez para errores más grandes. Sin embargo, notamos un pequeño error de código; se tenía una frecuencia de operación de 100 Hz (10 ms), pero debido a que se trabajó sobre un código de ejemplo que incluía un *delay* de 250 ms , la respuesta tenía retraso. También encontramos un error matemático en la ecuación del controlador que más tarde corregimos.

En este punto, el código se encontraba completo y solo faltaba testearlo para encontrar los mejores valores para $Power_{máx}$, K_p , K_i y K_d . Se comenzaron las pruebas con una potencia baja ($50 - 60$) y solo con la constante proporcional K_p ; el auto andaba correctamente, pero las curvas las recorría con lentitud, además en ocasiones no lograba completarlas y se salía del circuito. Luego de varios intentos, aumentamos el valor de la constante de integración K_i con el objetivo de que los errores provocados por las curvas se sopesaran. Finalmente, a modo de corregir oscilaciones y amortiguar los desfases, aumentamos el valor de la constante de derivación K_d . Finalmente, encontramos que los valores óptimos para el circuito de la clase fueron los de la **versión 3.0**:

- $Power_{máx} = 95.0$
- $K_p = 1.0$
- $K_i = 0.8$
- $K_d = 0.05$
- Curva = 1 (Tangente)

Si bien esta última configuración resultó óptima para la competencia, se observó que en ocasiones el auto oscilaba sobre la línea, ya que, según el control que aplicamos, el auto debía estar centrado en toda ocasión sobre la línea. Sin embargo, no es de nuestro interés que el auto se alinee de inmediato con la línea de manera brusca, por lo que buscamos una forma de *suavizar* la curva de error que se estaba utilizando; se encontró entonces la curva de error tangente suavizada, la cual ofrece cambios suaves (menores incluso que los de la curva lineal) para pequeños

errores y cambios más fuertes para grandes errores. Esta última curva se implementó en la versión 3.1 (por defecto esta versión tiene a la curva tangente seleccionada y no la suavizada) y si bien no se ha testado, se espera que pueda controlar mejor al auto en líneas rectas y curvas cerradas; el testeo aún se encuentra pendiente.

2. El equipo:

Durante los 3 talleres los aportes tendieron a seguir el hilo. Se empezó todos tratando de aportar en el código, sin embargo, Santiago era el con más dominio de este (Arduino) y fue el que se le ocurrió la idea de tomar la tangente y arcoseno de la curva para que así fuera un paso del robot más fluido. Matías en cambio fue clave en el tema de arreglar código, es decir, ver donde podría estar el problema de porque algo no está pasando, e investigar acerca del funcionamiento del código que se nos entregó inicialmente. Asimismo, Jaime fue un aporte en entender la calibración y en el testeo (junto con Matías y Santiago) del robot para que en conjunto como equipo se llegara a el producto final. Por último, todos aportaron su grano en todos los ámbitos, pero al mismo tiempo, siempre hubo tareas más específicas por integrante.

Consideraciones:

- El kit utilizado corresponde al kit N°5
- Todos los códigos se encuentran debidamente comentados, incluyendo las actualizaciones que incorpora cada versión más las ya implementadas en la versión anterior. Todo esto para un mejor entendimiento de la versión.
- El código implementado en la clase y la competencia corresponde a *Robot-seguidor-de-linea_v3/Robot-seguidor-de-linea_v3-0.ino*.
- Se incluye el archivo *FuncionError.nb* que muestra gráficamente las diferencias entre las distintas curvas de error.