Pre informe

Sistemas controlados por computadora

Diseño e implementacion de un robotin

Alumnos: Tania Ferreira, Sansoni Sebastián

Neuquén, 2018

1. Ideas

Cuestiones Importantes:

- Modelado del sistema (cinemático vs dinámico). Robot no holonómico.
- Modelado de los motores. Ensayos con carga y sin carga.
- Sistema de medición de RPM y línea.
- Problema del delay en la medición de los encoders.
- Ziegler-Nichols: para los motores no andaba bien por el delay variable, para el sistema total no lo podíamos implementar.
- PID tuner: una vez estimado el sistema da buenos resultados.
- Discretización del PID: el Tf era FUNDAMENTAL para el sistema completo.
- Identificación del sistema completo.
- Hipótesis simplificativas.
- Controlador final.

2. Resumen

El presente informe trata el diseño y desarrollo de un robot seguidor de línea. Este trabajo se encuentra enmarcado como proyecto de final de materia de Sistemas Controlados por Computadora, materia de la carrera Ingeniería Electrónica de la Universidad Nacional del Comahue.

El objetivo de estudio para la materia era la discretización e implementación en la vida real de un controlador PID. Con esto en mente se propuso la realización de un robot seguidor de línea del estilo utilizado en diversas competencias de robótica, donde el objetivo de diseño es que el robot sea capaz de seguir una pista en un plano horizontal de forma autónoma en el menor tiempo posible. El presente trabajo se orientó a la obtención de un seguidor de línea básico, de velocidad moderada, con perspectiva de mejorarse en versiones posteriores.

3. Introducción

Esto fue copiado de la propuesta presentada 19/12/17

3.1. Motivación

En el marco de las Jornadas Argentinas de Robótica (JAR) 2019 a realizarse en la Universidad Nacional del Comahue, se desea dar los primeros pasos en la implementación de sistemas robóticos. En particular, se desea trabajar en el inicio de un movimiento en la facultad para la utilización de la robótica educativa.
La robótica educativa, también conocida como robótica pedagógica, es una disciplina que tiene por objeto la concepción, creación y puesta en funcionamiento de prototipos robóticos y programas especializados con fines pedagógicos (Ruiz-Velasco, 2007). Partiéndose de un proyecto concreto a realizar, el proceso de concepción, diseño, armado y puesta en marcha del prototipo enriquece el proceso de aprendizaje del alumno, fortaleciendo sus capacidades de liderazgo y motivándolo a resolver retos cada vez más complejos.
Este tipo de herramienta pedagógica se encuentra dentro de la metodología ABP (aprendizajes basados en proyectos). El ABP se desarrolla en un entorno real y experimental, lo que ayuda a los alumnos a relacionar los contenidos teóricos con el mundo real y mejora la receptividad para aprender los conceptos teóricos. Los alumnos toman un papel activo en el proyecto, ya que tienen que marcar el ritmo y profundidad del aprendizaje, y fijar los objetivos de la realización del proyecto. Este tipo de metodología motiva al alumno, por lo que resulta una estrategia de particular interés para mejorar el rendimiento académico y la persistencia en los estudios.
En la UNCo, particularmente en nuestra carrera, existe una fuerte inclinación a priorizar la enseñanza de conceptos casi exclusivamente teóricos, habiendo actualmente pocas materias que pidan la realización de proyectos concretos. Este enfoque puede traer aparejado una serie de inconvenientes: alumnos con poca capacidad para resolución de problemas reales, dificultad para relacionar los conceptos aprendidos con la vida real o para entender los mismos, falta de motivación en los alumnos, etc. Considerando esto y que actualmente la tasa de deserción en la carrera es de XXX %, es que se desea avanzar hacia las metodologías ABP mencionadas, particularmente en el área de robótica.
En el marco de esto, el presente trabajo busca implementar los conocimientos adquiridos en la materia en el diseño e implementación de distintas estrategias de control sobre un robot en la configuración de seguidor de línea.

3.2. Seguidor de línea

Los robots seguidores de línea son robots muy sencillos, que cumplen una única misión: seguir una línea marcada en el suelo (normalmente de color negro sobre un tablero blanco). Son considerados los "Hola mundo" de la robótica. La dificultad del recorrido determina la complejidad del robot y el requerimiento o no de una mayor cantidad de sensores, capacidad de procesamiento, mejoras de hardware, etc. Esta flexibilidad los hace particularmente interesantes como herramientas educativas, ya que se puede comenzar con diseños sencillos para luego avanzar en sucesivas mejoras del prototipo, pudiendo adaptarse a distintas materias de la carrera.
En Argentina se festejan distintas competencias relacionadas con la fabricación y puesta en marcha de los seguidores de líneas, también clasificada como la modalidad "carreras". Se destacan entre otros:

- 15^o Competencia Internacional de Robótica - GRS - UTN Bahía Blanca.
- Liga nacional de robótica - `http://www.lnr-argentina.com.ar`
- Competencia Nacional de Robótica 2017 - Instituto La Salle Florida – Bs. As.
- Grupo de Robotica Universidad de Mendoza (GRUM).
- Instituto Tecnológico del Comahue (ITC)- Neuquen.

A nivel internacional existe un proyecto llamado Open Lamborghuino [1], en el cual se detalla la construcción de un seguidor de línea con componentes relativamente baratos y de fácil acceso.

4. Objetivos de la propuesta

- Desarrollar e implementar un robot seguidor de linea.
- Identificación del sistema.
- Implementar un controlador PID discreto: efectos de la acción derivativa y algunas cuestiones practicas.

5. Estructura del Trabajo

El siguiente trabajo se divide en XXXX secciones. En la sección 6 se define el hardware y estructura mecánica a utilizar, con un detalle del material de trabajo disponible. En la sección 7 se analiza cómo modelar matemáticamente el comportamiento del sistema. En 8 se define la estrategia de control a implementar. En 9 se explican los sensores diseñados para el robot. En 10 se aborda la identificación de sistemas realizada para poder ajustar los controladores, lo cual se detalla en la sección 11. Finalmente, la sección 12 discute los resultados obtenidos al probar el sistema completo, mientras que en la sección 13 se dan las conclusiones del trabajo.

6. Implementación

Para facilitar la implementación del robot, se propone utilizar la configuración de hardware recomendada en open lamborghino mas un diseño propio. Se detallan a continuación:

- 2 motores N20 con reducción 1:10 [2].
- Driver para motores: TB6612FNG [3].
- Computadora de abordo: Arduino Nano [4].
- Batería de LiPo: 2S 1100mA.
- Comunicación inalámbrica: par de Módulos bluetooth HC-05 o similar.
- Estructura fabricada en la impresora 3D de la facultad.

6.1. Estructura Mecánica

Se definió utilizar una estructura con dos ruedas y un punto de apoyo. El frente del robot se encuentra en el lado opuesto al del punto de apoyo, con una estructura para soporte de los LEDs del sensor de línea colocada en esta zona. La estructura diseñada se muestra en la figura 1. El diseño de las piezas mecánicas fue realizado por el ahora ingeniero Carrascal Cliver. Las mismas se imprimieron con impresora 3D.

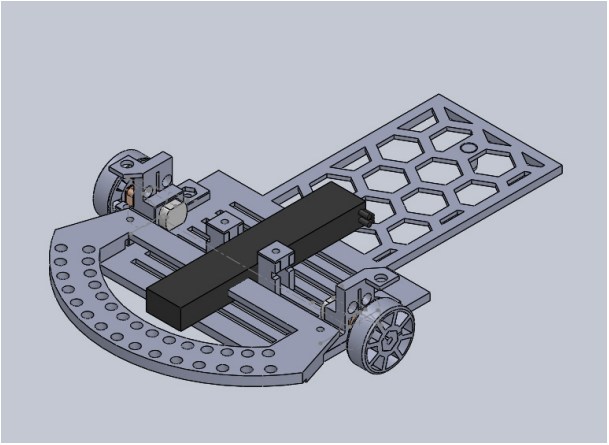


Figura 1: Estructura mecánica del robot.

7. Modelado del sistema

7.1. Modelo cinemático

7.1.1. Ecuaciones de Movimiento

Según [5], el robot que se desea modelar se lo conoce como robot diferencial ya que cuenta con 2 ruedas y un punto de apoyo. Esta clase de robots son no holonómicos ya que tiene restricciones a la hora de realizar movimientos sobre el plano; por ejemplo, no puede realizar movimientos puramente laterales. En la figura 2 se muestra un esquemático de este tipo de robots.

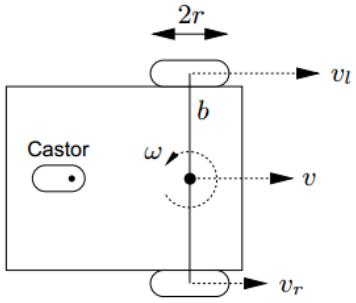


Figura 2: Robot diferencial.

Suponiendo que no hay deslizamiento, la posición y orientación del punto central del dispositivo, C, están dadas por las ecuaciones cinemáticas 1. El punto C es el punto medio del segmento que une las ruedas del robot.

$$\dot{x} = v\cos\theta \quad \dot{y} = v\sin\theta \quad \dot{\theta} = \omega \tag{1}$$

Donde (x, y) indica la posición del centro C en el espacio carteciano y θ, conocida como orientación, es el angulo entre el eje x del sistema de cordenadas y el eje homologo del robot, es decir, el eje de simetría del cuerpo. De esta manera $(x, y, \theta)^T$ definen la postura del sistema completo.

Si los motores se mueven a velocidades ω_R y ω_L, las velocidades v y ω son [6]:

$$v = \frac{r(\omega_R + \omega_L)}{2}$$

$$\omega_C = \frac{r(\omega_R - \omega_L)}{L} \tag{2}$$

Donde L es la distancia entre las ruedas.

7.1.2. Restricciones de Sensores Disponibles

Dado que el único sensor disponible es el sensor de línea, no se tiene la información completa para trabajar con las ecuaciones 1. En función de esto se decidió plantear una hipótesis simplificativa que permitiese estimar la orientación del vehículo respecto a la pista. Se decidió trabajar con esta variable porque la ecuación que

gobierna la orientación del robot es LTI y si la pista es una línea recta la orientación relativa a esta respondería a la misma ecuación.

Las hipótesis a utilizar son que la pista es una línea recta y que el centro C del vehículo se encuentra siempre sobre el centro de la linea a seguir. Así, la señal de error e se toma como la distancia entre el eje central del robot y el punto de intersección entre la línea de sensores y la recta a seguir. Esta situación se muestra en la figura 3.

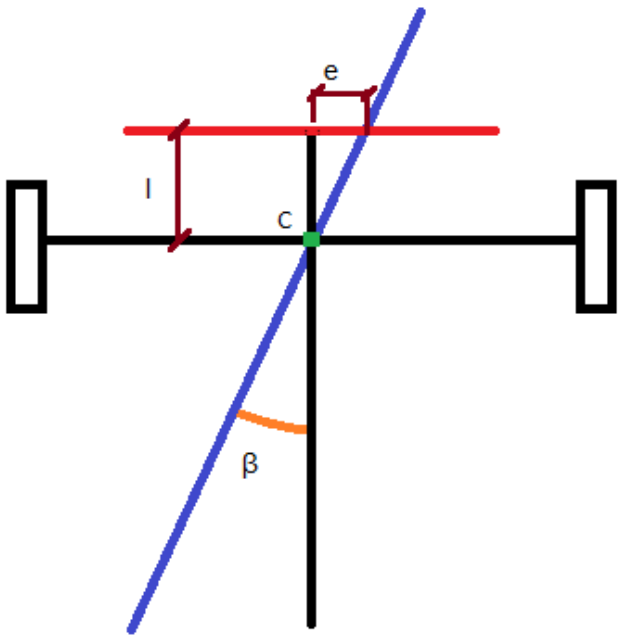


Figura 3: Esquema del robot sobre la línea a seguir (azul). La línea roja representa la línea de sensores, e es la distancia medida entre el eje central del robot y la intersección de la recta a seguir con la línea de los sensores, y β es el ángulo entre la línea a seguir y el eje central del robot.

Bajo la hipótesis de que la pista es una línea recta, la relación entre β y la orientación del robot θ está dada por:

$$\beta = \theta - \psi$$

Donde ψ es la orientación de la pista. Como ψ es constante, reemplazando en 1 obtendríamos:

$$\dot{\beta} = \omega_C \tag{3}$$

Dado que sólo se tiene la medición de e, se utiliza la segunda hipótesis propuesta: que el punto C se mantiene siempre sobre la línea a seguir. Bajo esta suposición, podemos calcular β por trigonometría:

$$\tan(\beta) = \frac{e}{l}$$

Como l es fijo, la medición de e puede traducirse en una medición del ángulo. De esta manera podríamos buscar minimizar el ángulo para alinear el robot con la pista. Como la medición se hace suponiendo que C está sobre la línea, si el robot no está centrado respecto a la pista el cálculo del ángulo va a indicar una medición errónea, por lo que al llevar a cero el β medido de esta manera se termina garantizando que el robot esté centrado respecto a la pista.

La hipótesis utilizada debería ser válida si el sistema controlado es lo suficientemente rápido para corregir desviaciones antes de que el centro se desplace fuera de la pista.

Como lo que se controla son las velocidades de cada motor de forma individual, resulta conveniente reemplazar la ecuación 2 en la ecuación 3:

$$\dot{\beta} = \omega_C = \frac{r(\omega_R - \omega_L)}{L}$$

Como sólo nos interesa la diferencia entre velocidades angulares del motor, se puede mantener v constante tomando:

$$\omega_R = \omega_{fijo} + \frac{\Delta\omega}{2}$$

$$\omega_L = \omega_{fijo} - \frac{\Delta\omega}{2}$$

Esto disminuiría el problema a una sola señal de control, $\Delta\omega$.

7.1.3. Radio de giro

En la figura 4 se muestra la disposicion del robot cuando esta siguiendo una curva. En este caso la velocidad

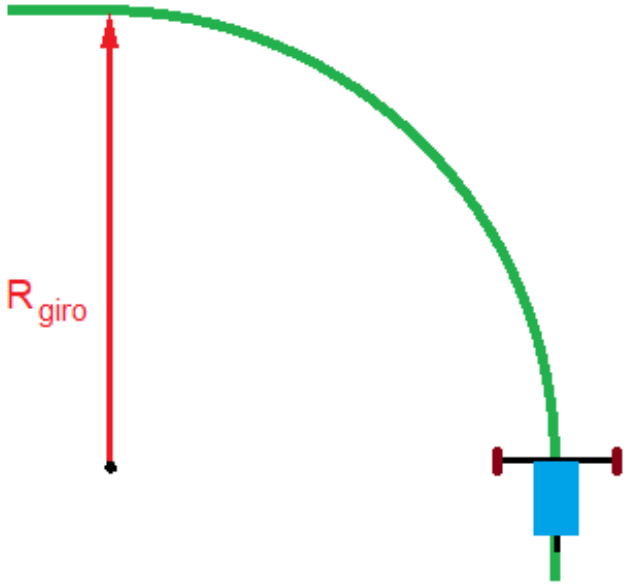


Figura 4: Esquema del robot girando.

del punto C tiene una componente tangencial, v, y una componente de giro, ω, dadas por:

$$v = \frac{r(\omega_R + \omega_L)}{2} = r\omega_{fijo} \tag{4}$$

$$\omega_C = \frac{r(\omega_R - \omega_L)}{L} = \frac{r\Delta\omega}{L} \tag{5}$$

Como la relación entre ambas magnitudes es:

$$v = R_{giro}\omega_C$$

reemplazando esto en las ecuaciones 4 y 5, se obtiene:

$$\Delta\omega = \frac{\omega_{fijo}L}{R_{giro}} \tag{6}$$

Así, mientras más pequeño sea el radio de giro, más grande debe ser $\Delta\omega$. Para una velocidad ω_{fijo} constante, el máximo valor admisible de $\Delta\omega$ determinará el menor radio de giro que podrá hacer el robot.

7.2. Modelado de los motores

Los motores a utilizar son motores de DC con escobillas. Según [8, pág. 58-59], si se desprecia el torque de la perturbación, la función de transferencia de la combinación motor-carga resulta

$$\frac{\theta(s)}{V_f(s)} = \frac{K_m}{s(Js+b)(L_f s + R_f)}$$

donde θ es la posición angular, V_f es la tension de exitacion, K_m es la constante que relaciona la corriente con el par del motor, J es el momento de inercia del eje, b es la constante de friccion del eje, L_f y R_f son la autoinductancia y resistencia del bobinado, respectivamente. Simplificando la función de transferencia resulta:

$$\frac{\theta(s)}{V_f(s)} = \frac{k}{s(s+a)(s+b)}$$

donde k, a y b son constantes a determinar mediante la identificación.

Por practicidad en vez de usar θ se usa $\dot{\theta}$, lo que produce la siguiente función de transferencia

$$\frac{\dot{\theta}(s)}{V_f(s)} = \frac{k}{(s+a)(s+b)} \tag{7}$$

8. Estrategia de Control a Utilizar

Para alcanzar los objetivos del proyecto se decidió implementar tres controladores PID en simultáneo: dos controladores para regularizar el comportamiento de los motores y el controlador para el sistema total, que se encargará de que el robot siga la pista. Los dos primeros se utilizan para que las ruedas se muevan a la velocidad requerida y para compensar asimetrías en la respuesta de los dos motores. Una vez implementados todos los controladores descritos el sistema total queda según se observa en el diagrama en bloque de la figura 5.

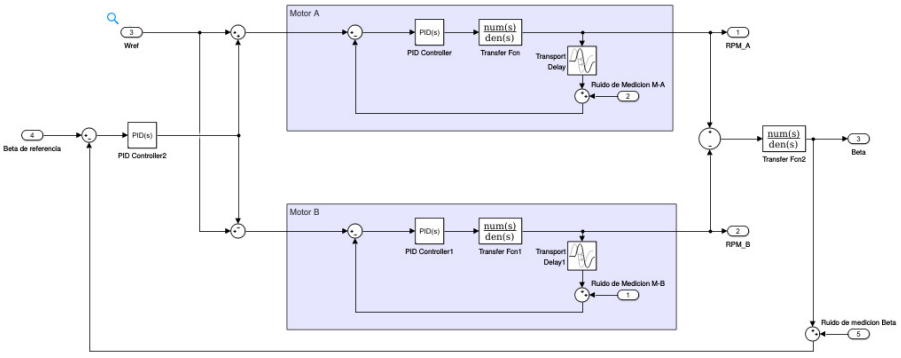


Figura 5: Diagrama en bloques del sistema total con los tres controladores PID implementados. MEJORAR ESTA IMAGEN

8.1. PID

Esta parte es un choreo MODIFICADO de [11, pág. 25].

En esta sección no se espera realizar un analisis profundo sobre el controlador PID, sino mas bien una intro cualitativa, con algunas referencias hacia [11]. Por otro lado se parte de la suposicion de que se conocen algunos principios fundamentales de control (realimentacion, lugar de las raices, polos, ceros, entre otros.)

Un controlador PID esta constituido por 3 controles facilmente identificables: Proporcional (P), Integral (I) y Derivativo (D). Si bien el nombre *PID* hace referencia a la estrategia de control, existen distintas maneras de implementarlo, esto es I-PD, PI-D, PI, PD, P, entre otros.

Por cuestiones didacticas, se explicara sinteticamente lo mas relevante de algunos de estos controladores.

Considerando la figura 6, se puede observar el diagrama en bloques de un sistema generico realimentado, en donde l y n son ruido/perturbaciones del modelo y de la medición respectivamente, r es la señal de referencia, esto es, el valor deseado de la señal de salida y; e es la diferencia entre el valor deseado y el valor medido de la salida, u es la salida del controlador (o accion de control) y x es la salida de la planta sin que este afectada por ruidos / perturbaciones.

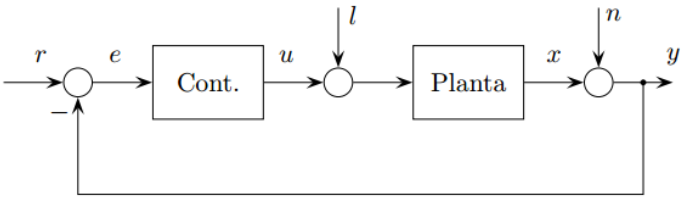


Figura 6: Diagrama en bloques de un sistema realimentado

8.1.1. Control Proporcional

Un control proporcional es aquel en donde la accion de control u es proporcional al error e, esto es

$$e = r - y \quad \rightarrow u = k(e) = k(r - y) \tag{8}$$

Para lograr un mejor entendimiento, vamos a hacer uso de un ejemplo:

Ejemplo: Se desea controlar la velocidad del eje de un motor de corriente continua (CC), para esto se cuenta con un generador de PWM y un sistema de medicion de las RPM del eje.

Supongamos que deseamos que el eje del motor gire a $1000RPM$, inicialmente el eje se encuentra en reposo, por lo que la medicion del sensor sera $y = 0RPM$. Dados estos datos el error sera

$$e = r - y \quad \rightarrow e = 1000RPM - 0RPM = 1000RPM$$

por lo que la accion de control sera $u = k1000RPM$.

En este caso el valor de k representa una constante que adecua las unidades de RPM a $\%$ *de duty* y ademas aplica un factor de ganancia, esto es:

$$k = k_{unidades} k_{ganancia}$$

Dicho esto, supongamos que $k_{unidades} = 1/10 \quad [\%duty/RPM]$ y que $k_{ganancia} = 1$, luego la salida del controlador sera $100\,\%duty$.

Si todo esta bien conectado, el eje se comenzara a mover, por lo que el sensor de velocidad medirá una velocidad no nula, lo cual producira una señal de control menor a la que inicialmente se obtuvo. Luego de un periodo prudente en donde el estado transitorio convergio, se observa que la velocidad final del motor resulta ser $y = 700RPM$ y la acción de control por lo tanto sera $u = k(r - y) = 30 \quad [\%duty]$.

Gracias al anterior ejemplo se pueden observar algunas características del control proporcional:

1. Cuando el error e es cero la salida es cero, por lo que siempre abra una diferencia entre el valor deseado (set point) y el valor medido, a menos claro que el set point sea cero.

2. Si el valor de $k_{ganancia}$ es mas grande, en el estado estacionario se puede observar que la salida del control sera mayor, por lo que el motor girara a mayor velocidad, produciendo asi, un valor mas cercano al deseado.

3. Si $k_{ganancia}$ es un valor elevado, cuando exista alguna perturbación en la medición, este error en la medida sera k veces amplificado en la acción de control.

8.1.2. Control Integral

Para resolver la problemática planteada en el item 1 del control proporcional, se plantea como solución la utilización de un control integral.

El control integral se puede interpretar como que la salida de control u es proporcional a la integral del error e, en símbolos

$$u = K_i \int_{t_0}^{t_1} e\, dt \tag{9}$$

en la cual se observa que si el error entre la salida y el set point perdura en el tiempo, la acción de control se incrementara hasta que el error sea nulo.

Problemática: En la realidad, la acción de control posee una cota, ya que no puede ser arbitrariamente grande. Suponga el ejemplo utilizado en el **Control Proporcional**, pero esta vez tiene implementado un controlador PI y ademas hay una falla en las conexiones lo que produce que el motor no funcione. El Set Point se establece en $1000RPM$, la acción de control es de $100\,\%duty$, pero como el motor esta mal conectado, el eje no gira produciendo que luego de unos instantes el error siga siendo de $1000RPM$ y el termino integral incremente a medida que siga pasando el tiempo. Cuando se corrija la falla eléctrica, el eje del motor girará a una velocidad superior a la del Set Point el tiempo necesario hasta que el error negativo "des-integre" todo el error acumulado anteriormente.

Como solución a esta problemática se implementan técnicas de ventaneo, tal que limiten la parte integral y no permita que diverja.

8.1.3. Control Derivativo

El control derivativo, o la acción derivativa, se puede interpretar como anticiparse al error en un estado posterior al actual, por medio del calculo de la tangente de la curva del error. Para obtener esta predicción, suponga la expansión en serie de Taylor

$$e(t_1) = e(t_0) + \frac{de(t_0)}{dt}(t_1 - t_0) + ...$$

$$e(t_1) \approx e(t_0) + \frac{de(t_0)}{dt}(t_1 - t_0) \tag{10}$$

en ella se puede apreciar que aparece un termino del error actual y luego otro termino con la derivada del error ponderada por la diferencia temporal. A este controlador se lo conoce comúnmente como PD, el cual su estructura básica se muestra a continuación

$$u(t) = K_p e(t) + K_d \frac{de(t)}{dt} \tag{11}$$

Según [11, pág. 40] la acción derivativa puede traer dificultades si existe ruido de medición de alta frecuencia. Un ruido de medición sinusoidal

$$n = a \sin(\omega t)$$

contribuye al término derivativo de la señal de control como

$$u_n = K_d \frac{dn}{dt} = aK_d\omega \cos(\omega t)$$

Podemos observar que la amplitud de la señal de control puede ser arbitrariamente grande si la frecuencia ω es suficientemente alta.

Por este motivo, es común que la ganancia del término derivativo sea limitada en alta frecuencia. Esto puede lograrse implementando el término derivativo como

$$D' = D_{idal}(s)D_{sat}(s) = sK_d \quad \frac{1}{1 + s\frac{K_d}{N}} = \frac{sK_d}{1 + s\frac{K_d}{N}} \tag{12}$$

Esta modificación puede interpretarse como la derivada del error del sistema concatenada con un filtro pasa bajos con un polo colocado N veces mas lejos. Quiere decir que para frecuencias menores a $\frac{N}{K_d}$ la derivada del error se amplificara como antes, pero cuando la frecuencia supere esta cota, el sistema saturara. Valores normales de N oscilan entre 8 y 20. (Pensar en el bode de este sistema, no se si hace falta poner una imagen)

8.1.4. PID total

Dado lo visto anteriormente, si se consideran todas las acciones de control en paralelo, el sistema para tiempo continuo resulta

$$u(t) = K_p e(t) + K_i \int_{t_0}^{t_1} e(t)\, dt + K_d \frac{de(t)}{dt} \tag{13}$$

en donde los parámetros K_p, K_i y K_d son las constantes con las que se realiza el ajuste para cada planta en particular.

La representación anterior es la forma clásica (o no interactuante) del controlador PID escrita para tiempo continuo. En forma de función de transferencia resulta:

$$G(s)e(s) = (K_p + K_i/s + sK_d)e(s) \tag{14}$$

considerando el ajuste de la ecuacion 12, la exprecion 14 resulta

$$G(s)e(s) = \left(K_p + K_i/s + \frac{sK_d}{1 + \frac{sK_d}{N}}\right) e(s) \tag{15}$$

8.1.5. Discretización PID

- Formas de discretizar
- discretizacion estandar, problemas
- Discretizacion del astrom

Según [15, pág. 214] se pueden representar una ecuación diferencial mediante su aproximación:

Método de Euler: La derivada de una variable se puede aproximar por una **diferencia adelantada**

$$\frac{dx(t)}{dt} \approx \frac{x(t+h) - x(t)}{h} \tag{16}$$

o por una **diferencia atrasada**

$$\frac{dx(t)}{dt} \approx \frac{x(t) - x(t-h)}{h} \tag{17}$$

por otro lado, según [11, pág. 46] la integral de una variable dada por la ecuación 18 se puede aproximar mediante distintas formas recursivas.

$$I(t) = \frac{K}{T_i} \int_0^t e(t)\, dt \tag{18}$$

Regla rectangular hacia adelante.

$$I(nh) = \frac{K}{T_i} h \sum_{0}^{n-1} e(nh) = \frac{K}{T_i} h \sum_{0}^{n-2} e(nh) + h\frac{K}{T_i} e((n-1)h) = I((n-1)h) + h\frac{K}{T_i} e((n-1)h) \tag{19}$$

Regla rectangular hacia atrás.

$$I(nh) = \frac{K}{T_i} h \sum_{0}^{n} e(nh) = I((n-1)h) + h\frac{K}{T_i} e(nh) \tag{20}$$

Esta aproximacion se diferencia de la anterior, en que utiliza la muestra actual de la señal $e(nh)$ en vez de la muestra anterior $e(h(n-1))$.

Regla trapezoidal.

$$I(nh) = \frac{K}{T_i} h \sum_{i=1}^{n} e((i-1)h) - e(ih) = I((n-1)h) + \frac{K}{T_i} \frac{e(nh) - e((n-1)h)}{h} \tag{21}$$

según [15] en el contexto del control digital la aproximación 21 suele denominarse *aproximación de Tustin, o transformación bilineal.* Utilizando estos métodos la funcion de transferencia discreta $H(z)$ se obtiene simplemente sustituyendo el argumento s de $G(s)$ por s', donde

$$s' = \frac{z-1}{h} \quad (Metodo\ de\ Euler) \tag{22}$$

$$s' = \frac{z-1}{zh} \quad (Diferencias\ en\ retroceso) \tag{23}$$

$$s' = \frac{2}{h}\frac{z-1}{z+1} \quad (Aproximacion\ de\ Tustin) \tag{24}$$

considerando la expresión 15 y teniendo en cuenta [15, pág. 219] el método más común para discretizar un PID es realizar una aproximación de Euler para la parte integral y una aproximación en diferencias en retroceso para la parte derivadora, lo cual origina el siguiente controlador discreto:

$$u(kh) = \left(K_p + \frac{K_i h}{z-1} + K_d \frac{z-1}{zh + \frac{K_d}{N}(z-1)} \right) e(kh) \tag{25}$$

el cual por medio de manipulación algebraica se puede transformar en la siguiente ecuación en diferencias

$$u(k+1) = Ae(k) + Be(k-1) + Ce(k-2) + Du(k-1) + Eu(k-2) \tag{26}$$

donde los parámetros A,B,C,D y E se obtienen a partir de las constantes h,Kp,Ki,Kd y N.

9. Sistemas de Medición

9.1. Medición de velocidad de los Motores

9.1.1. Estructura Mecánica

Para la medición de velocidad de los motores se utilizaron encoders ópticos. Estos consisten en un diodo LED infrarrojo enfrentado a un fotodiodo receptor en dirección vertical, con una estructura colocada entre ambos. Ésta última fue diseñada en SolidWorks e impresa con impresora 3D y su forma se asemeja a la de jaula de ardilla. La idea del diseño es que las varillas bloqueen o dejen pasar la luz del LED, de modo que el fotodiodo genere una señal que permita detectar el paso de cada una de ellas.
¿Poner que hay que verificar que la estructura sea opaca para la luz IR ?

9.1.2. Circuito Eléctrico

El fotodiodo receptor se encuentra conectado según el esquema mostrado en la figura 7. La señal de salida se conecta a un Schmitt trigger para discretizar la señal y luego la salida de ésta se conecta al microcontrolador. La sensibilidad de los diodos receptores se calibra manualmente por medio de R2.

Los diodos emisores, por su parte, se encuentran conectados en paralelo entre sí y en serie con una resitencia de 100Ω conectada a la salida de tensión de 5V del kit de desarrollo Arduino Nano. Inicialmente se había diseñado un circuito con espejo de corriente que permitía regular la intensidad de los LEDs desde el microcontrolador para poder adaptarse a cambios de luz en el ambiente, no obstante, las pruebas demostraron que se podía

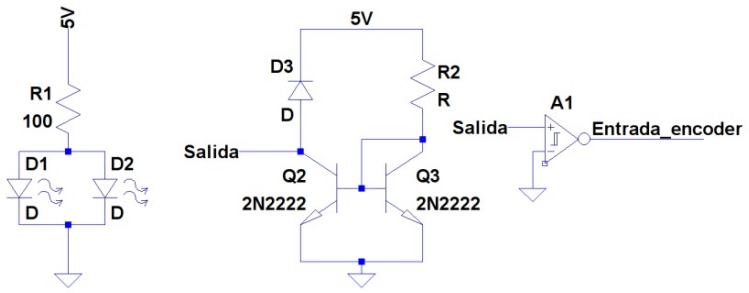


Figura 7: circuito del fotodiodo y de los diodos infrarrojos.

calibrar una intensidad que funcionase bajo todas las condiciones de luz del lugar de trabajo usado, por lo que se descartó el circuito por la simple versión actual.

Se midió con el osciloscopio la señal de salida de la compuerta y se verificó que la misma es una señal cuadrada, modulada por las barras de la jaula de ardilla. Esta modulación, al provenir de un sistema mecánico, copia las asimetrías del sistema.

9.1.3. Medición de Velocidad con el Microcontrolador

Las señales acondicionadas de los encoders (ver figura 7) se conectan a entradas del microcontrolador, donde se resuelve la medición de velocidad por medio de interrupciones. Para ello se utilizan pines con interrupción por cambio de estado y se mide el tiempo entre interrupciones sucesivas. Esta medición de tiempo presenta una serie de problemas explicados a continuación.

El microcontrolador ATmega328P posee un total de 3 timers. El timer1 se utilizó para generar los PWM de los motores, mientras que el timer2 se utilizó para la generación de un tiempo de muestreo constante para el sistema. Esto dejaba únicamente el timer0 disponible para realizar la medición de tiempos entre interrupciones, sin embargo, dado que el timer0 es el que utilizan las librerías de Arduino por defecto, se decidió buscar una alternativa que permitiese reservar este timer para realizar tareas de depuración. Para ello se decidió utilizar el timer2 de forma simultánea para las interrupciones constantes a Ts (periodo de muestreo) y para el conteo de tiempo.

Para entender cómo usar de forma conjunta el timer, consideremos lo que ocurre cuando se inicia el microcontrolador. Cada vez que el timer2 desborda se incrementa una variable auxiliar $cantOVerflow$. Cuando se detecta el primer cambio de estado o flanco (interrupción por pin), el tiempo transcurrido va a ser:

$$t = \frac{Prescaler}{f_{clk_{io}}}(TCNT2 + cantOVerflow\ OCR2A)$$

Ya que la cantidad de pulsos contados por el timer es igual al valor actual del timer (registro TCNT2 del microcontrolador) más $cantOVerflow$ veces el máximo valor que cuenta (dado por el registro $OCR2A$ del microcontrolador).

Una vez guardado el valor del timer actual, reseteamos $cantOVerflow$. Así, la siguiente vez que se genere una interrupción el valor de $cantOVerflow$ reflejará la cantidad de overflows que hubo desde la interrupción pasada. Si a la siguiente interrupción por cambio de estado de la señal del encoder hacemos el cálculo de tiempo como antes, tendremos un error de cálculo debido a que en realidad no transcurrieron $cantOVerflow$ ciclos de conteo completos del timer, sino que el primero arrancó a contar desde el valor que tenía el timer cuando se hizo la interrupción anterior. Para eso agregamos una variable auxiliar de corrección cuyo valor sea siempre el valor del timer al momento de la última interrupción:

$$t = \frac{Prescaler}{f_{clk_{io}}}(TCNT2 + cantOVerflow\ OCR2A - TCNT2_{anterior}) \tag{27}$$

Como este cálculo debe realizarse para cada motor, el código de interrupción por cambio de estado verifica cuál encoder generó la interrupción y almacena las variables necesarias para hacer el cálculo de velocidad del motor ($TCNT2$, $TCNT2anterior$ y $cantOverflow$ para cada motor) en el código principal, es decir, fuera de la interrupción para no consumir tiempo de ejecución con prioridad (interrupción) del microcontrolador.

Conociendo el tiempo entre interrupciones, el cálculo de velocidad correspondiente en rpm se realiza según la fórmula:

$$\omega = 60\frac{seg}{min}\ \frac{1}{D\ t}$$

Donde D es la cantidad de varillas del encoder.

Para el primer modelo de encoder utilizado la señal cuadrada a la entrada del microcontrolador presentaba variaciones en el ancho del pulso debidas al movimiento de la rueda que se encontraba ligeramente descentrada. Para evitar este inconveniente y las variaciones introducidas por posibles asimetrías en la fabricación del encoder, se decidió calcular la velocidad con el tiempo demorado en realizar una vuelta entera. Para el modelo de encoder actual esto no debería ser tan necesario, sin embargo se ha mantenido en el código. El resultado de esta forma de medición equivale a agregar un filtro moving average variante en el tiempo. El efecto de esto se verá más adelante. Sabiendo que la mínima variación de tiempo está dada por la cantidad de conteos realizados por el timer, resulta evidente que mientras más rápido cambie el timer mayor precisión se tendrá en la medición de velocidad. Sabemos que el timer genera interrupciones cada:

$$T_{int} = \frac{Prescaler(1 + OCR2A)}{f_{clk_{io}}} \tag{28}$$

por lo que si se desea un T_{int} pequeño se debe utilizar un prescaler lo menor posible. Dado que este mismo T_{int} es el que se utiliza para generar la frecuencia de muestreo constante, se aumenta el prescaler efectivo para la determinación de Ts utilizando una variable auxiliar como contador. Así, la frecuencia de muestreo del sistema será un múltiplo de T_{int}. Para determinar el valor del prescaler a utilizar se debe tener en cuenta que el tiempo T_{int} sea suficiente para realizar las operaciones necesarias en la interrupción por overflow del timer y para que estas interrupciones no se generen con una frecuencia tan alta que impida el correcto funcionamiento de las restantes interrupciones.

9.2. Sensor de Línea

Los seguidores de línea utilizados regularmente consisten en una serie de fotodiodo y LEDs infrarrojos colocados en dos líneas paralelas. El mecanismo se detalla a continuación:

Suponga que el sensor se encuentra sobre una superficie reflectante (por ej. con color blanca). Luego la luz emitida por el LED IR se reflejará y llegará al fotodiodo con una intensidad I_{blanco}.
Ahora suponga que el sensor se encuentra sobre una superficie poco reflectante (por ej. con color negra). Luego la luz emitida por el LED IR se reflejará y llegará al fotodiodo con una intensidad I_{negro}.
Entonces, siempre y cuando la diferencia entre estas intensidades sea mayor que la sensibilidad del sensor $|I_{blanco} - I_{negro}| > I_{sens}$, se podrá discernir entre una superficie y la otra.
Generalmente cuando cambia el color de una superficie también cambia el material de la misma, por lo que no se puede asegurar que *siempre* una superficie de color negro generara una intensidad reflejada menor que una superficie de color blanco $I_{negro} < I_{blanco}$. La idea se muestra en la figura 8.

Poner imagen que muestre emisor-receptor con la luz rebotando

Figura 8: COMPLETAR.

9.2.1. Circuito Eléctrico

Para alimentar este arreglo de emisores y receptores se utilizó el circuito de la figura 9. En ella se observa que los diodos se alimentan a través de espejos de corriente que permiten ajustar la sensibilidad. Las salidas de cada fotodiodo se conectan a compuertas Schmitt trigger para acomodar la señal de salida a una que permita trabajar con lógica digital. Se corroboró con el osciloscopio que la señal de salida era apropiada para la lectura digital por parte del microcontrolador.

9.2.2. Estructura Mecánica

La disposición física de los LEDs estaba dada inicialmente por la placa PCB utilizada en el circuito. Esto hacía que los LEDs se encontraran a alturas ligeramente variables y con leves diferencias de inclinación, además de que la placa no se encontraba centrada respecto al cuerpo del robot por estar colocada en una posición casi arbitraria. A pesar de estos problemas, el sensor presentaba un desempeño básico aceptable que permitió

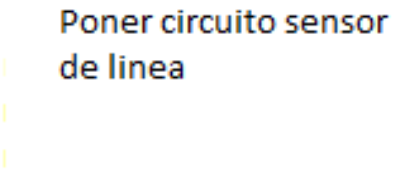
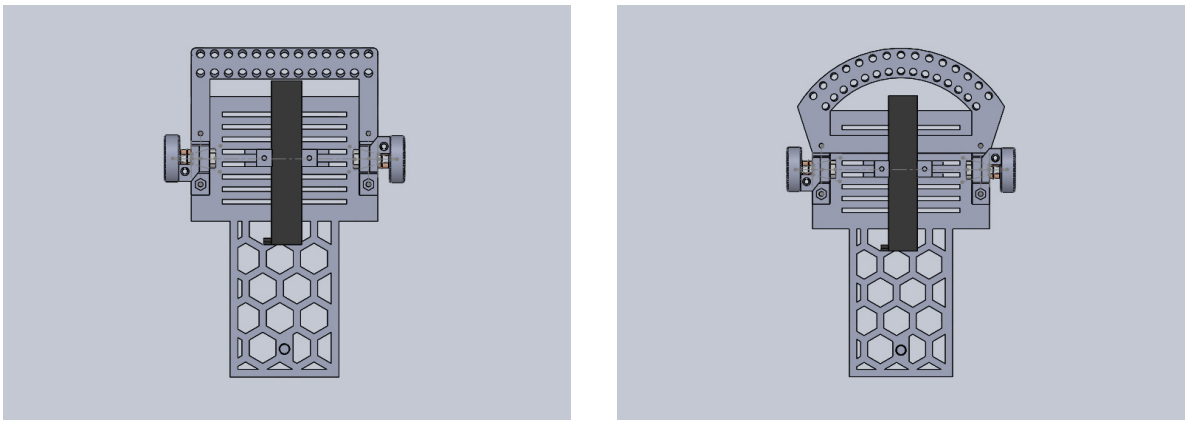


Figura 9: circuito utilizado para alimentar los diodos LED infrarrojos y los correspondientes fotodiodos que componen el sensor de línea.

realizar las pruebas necesarias hasta lograr que el robot realizara el seguimiento de una pista. No obstante, una vez finalizado el prototipo inicial, se rediseñó la estructura para solucionar estos problemas. En esta etapa del trabajo se diseñaron y fabricaron dos estructuras alternativas: la primera con una disposición de los LEDs tradicional lineal y la segunda con una disposición circular. Esta última se propuso en función del análisis realizado en la sección Modelado del Sistema [7], según el cual podíamos trabajar con un modelo LTI simple si considerábamos únicamente el ángulo de desviación de la orientación del robot respecto de la línea. Las estructuras obtenidas pueden observarse en la figura 10.



(a) (b)

Figura 10: esquema del robot con los dos tipos de estructuras diseñadas para fijar la disposición de los diodos según la configuración geométrica deseada: (a) lineal, y (b) circular.

Se optó por diseñar las dos variantes para poder probar a futuro distintas estrategias de control, no obstante, la utilizada actualmente funcionaría mejor con la disposición circular. Además, para permitir variar las pruebas en etapas de desarrollo, se diseñaron las estructuras con agujeros adicionales para tener la posibilidad de agregar LEDs o modificar la distancia entre los mismos. Para las pruebas mostradas en este informe se utilizó una configuración con 8 pares de LEDs ubicados desde los extremos al centro dejando un lugar vacío entre ellos.

9.2.3. Valores de Lectura

Considerando que se deseaba obtener un valor de ángulo como lectura del sensor, se podría haber realizado una deducción geométrica de los valores de ángulo a asignar para cada lectura posible del sensor. Sin embargo, dado que el diseño inicial tenía múltiples asimetrías, que resultaba difícil identificar correctamente el centro del robot y que los diodos utilizados pueden presentar imperfecciones o diferencias de sensibilidad, se optó por realizar un ensayo donde se midiera el rango de valores de ángulo de desviación para el cual se mantenía cada medición posible del sensor. Este ensayo se realizó marcando en un papel líneas de referencia y colocando cinta en línea recta sobre el mismo. Girando el robot sobre la pista con el centro siempre sobre un punto medio de la línea, se marcaron los rangos de desviación para los cuales se mantenía cada medición del sensor.

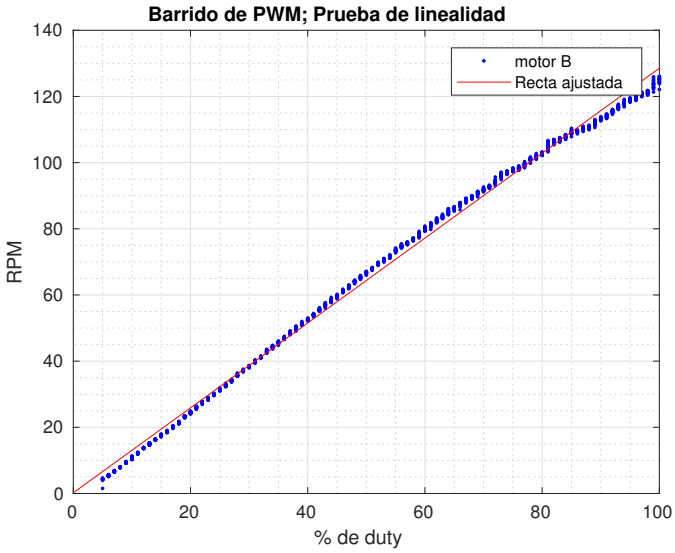


Figura 11: Curva de respuesta entrada-salida para el motor B.

9.2.4. Lectura del Sensor con el Microcontrolador

Las salidas de las compuertas se conectan a pines de entrada del microcontrolador. En el programa principal la medición de la línea se necesita cada vez que se calcula el PID del sistema total, por ello se programó una subrutina de medición que se llama en el main antes de calcular el controlador cada Ts segundos. Dentro de esta subrutina se lee cada valor digital de entrada y se almacenan los resultados en un vector de 8 componentes. Para anticiparse a valores de mediciones atípicos y atenuar el efecto de posibles errores de medición, se calcula una suma ponderada de las entradas para obtener un número que indica debajo de cuál de los fotodiodos se encuentra el centro de la línea, siendo las posibilidades para el caso de 8 LEDs: 1, 1.5, 2, 2.5,...,7,7.5 y 8, donde los números no enteros representan el caso en el que la línea se encuentre entre dos fotodiodos. Si a este número lo multiplicamos por dos y le restamos dos, nos queda uno de los siguientes resultados: 0,1,2,...14, que pueden usarse como índice de un vector. De esta forma lo que se hace es definir un vector constante con los resultados de ángulo obtenidos por ensayo cuando la línea estaba bajo cada una de las posibilidades y utilizar el número calculado para definir con cuál de los elementos del vector quedarse como medición actual. El caso en que todos las entradas estén en 0 se toma como valor particular y se la asigna el número 3 para que funcione como bandera e indique que se perdió la línea. Cuando esto ocurre el programa principal mantiene para los cálculos el último valor válido medido.

10. Identificación de Sistemas

Para sintonizar los controladores PID ha implementar se necesita tener algún tipo de información sobre el comportamiento de la planta a controlar. En función de lo definido en la sección 8, las plantas a identificar o caracterizar son tres: los dos motores, cuyo modelado matemático se analizó en la sección 7.2, y el sistema total, que incluye los motores con sus PID y la cinemática del robot analizada en la sección 7.1.

A continuación se detallan los ensayos realizados para la identificación de los sistemas a controlar.

10.1. Identificación de los Motores

El modelo de función de transferencia planteado matemáticamente en la sección 7.2 nos define como variable de salida la velocidad angular del motor y como variable de entrada la tensión promedio aplicada. Para la implementación con el microcontrolador se utiliza una señal de PWM de alta frecuencia (comparada con la frecuencia de corte del polo mecánico y el polo eléctrico) cuyo duty cycle modifica el valor de tensión continua aplicado al motor. Esto es factible debido a que el *sistema motor* actúa como un filtro pasabajos, manteniendo la componente de continua de la señal y atenuando las restantes. Dado que esta es la señal que podemos controlar realmente, se utiliza como señal de entrada el duty cycle utilizado en los PWM.

El modelo obtenido de forma teórica realiza varias simplificaciones. Una de ellas es ignorar el ciclo de histéresis de los motores y otra es no considera el efecto de una carga en el motor, que en este caso implicaría considerar la dinámica del robot sobre el suelo. Estos fenómenos hacen que el comportamiento real de los motores no sea realmente lineal. Para analizar la linealidad el sistema y el intervalo de trabajo a utilizar se diseñó un ensayo que barriera las distintas posibilidades de duty cycle para cada motor y almacenara varias muestras de las velocidades obtenidas en el estado estacionario. En la figura 11 se muestra la curva obtenida para el motor B. La del motor A es análoga.

Analizando la curva de respuesta del sistema observamos que los motores comienzan a moverse recién a partir de un 5 % de duty cycle. Esto se debe al par de fricción estática aplicada al eje. Por otra parte, la recta ajustada muestra que el comportamiento es aproximadamente lineal hasta un 80 % de duty cycle. Para garantizar un buen funcionamiento se trabajará de 10 a 80 para los ensayos de estimación del sistema.

En función de las consideraciones anteriores, se diseño un ensayo que permitiese obtener la respuesta al escalón de ambos motores. Dado que nos interesa conocer el comportamiento con carga y no en vacío, se buscó que el ensayo se realizara con el robot completo funcionando en condiciones normales. Para ello se programó que el robot anduviera en línea recta, primero a una velocidad baja y luego a una mayor dada por un cambio repentino del duty cycle de ambos motores. Las señales de interés se muestrearon a una frecuencia de 200Hz, se almacenaron en la EEPROM y luego fueron transmitidas a la computadora por puerto serie. La curva obtenida se muestra en la figura 12.

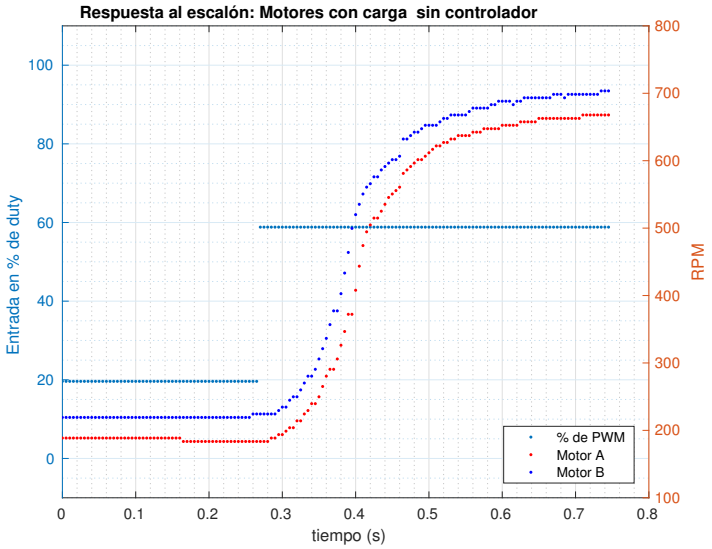


Figura 12: Respuesta al escalón de ambas motores a lazo abierto con carga.

Analizando los resultados obtenidos se detectó que existía una demora en el sistema no anticipada por el modelo matemático. Ésta se debe a la dinámica del sistema de medición de velocidad, que en los análisis teóricos no se suele considerar. Específicamente, el promedio realizado al momento de medir la velocidad de cada motor genera la demora observada. Dado que se almacenan los tiempos correspondientes a un giro completo de la rueda para realizar el cálculo de velocidad, cuando la velocidad real cambia de forma abrupta el sistema demora una vuelta en empezar a medir de forma correcta. Este comportamiento se corresponde con el de un sistema de media móvil, pero variante en el tiempo y alineal, ya que la demora depende de la velocidad de giro. Para poder continuar trabajando en el contexto de sistemas LTI se utilizará una estimación del sistema alrededor de cierto punto de trabajo.

Utilizando los datos obtenidos se estimó un sistema LTI continuo que se ajustara a cada curva utilizando la herramienta de Matlab System Identification [13]. Las funciones de transferencias obtenidas para cada motor fueron:

$$M_a(s) = \frac{7075.3e^{-0.013s}}{(s+9.631)(s+67.5)} \tag{29}$$

$$M_b(s) = \frac{5042.1e^{-0.013s}}{(s+9.988)(s+45.48)} \tag{30}$$

10.2. Identificación de la Cinemática el Robot

En función de lo analizado en la sección 7.1, podemos pensar que si el robot arranca alineado con una línea recta, y manteniendo la restricción de v constante, entonces la planta se comporta como un sistema cuya entrada es $\Delta\omega$ y su salida β, gobernado por la ecuación:

$$\dot{\beta} = \frac{r\Delta\omega}{L} \tag{31}$$

Para realizar el ensayo al escalón se debería empezar con $\Delta\omega = \beta = 0$ y luego cambiar $\Delta\omega$ abruptamente a un valor final, midiendo el valor de β obtenido. Físicamente esto implica que el robot debe empezar andando a velocidad constante, perfectamente alineado con la línea, y luego girar hasta perder la línea (más allá de este punto no tiene sentido seguir porque se pierde la medición). Si el cambio de velocidad de los motores fuese automático, el robot giraría siguiendo una circunferencia perfecta como se ve a en la figura 13.

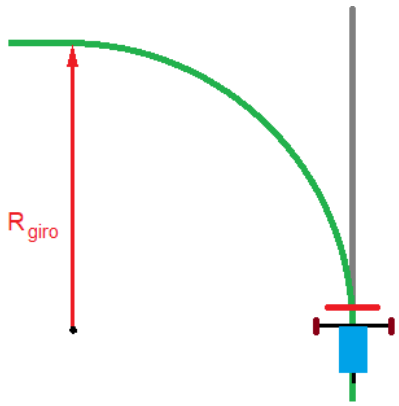


Figura 13: Trayectoria seguida por el robot si empieza a girar a $\Delta\omega$ constante (verde) arrancando inicialmente alineado con la línea (gris).

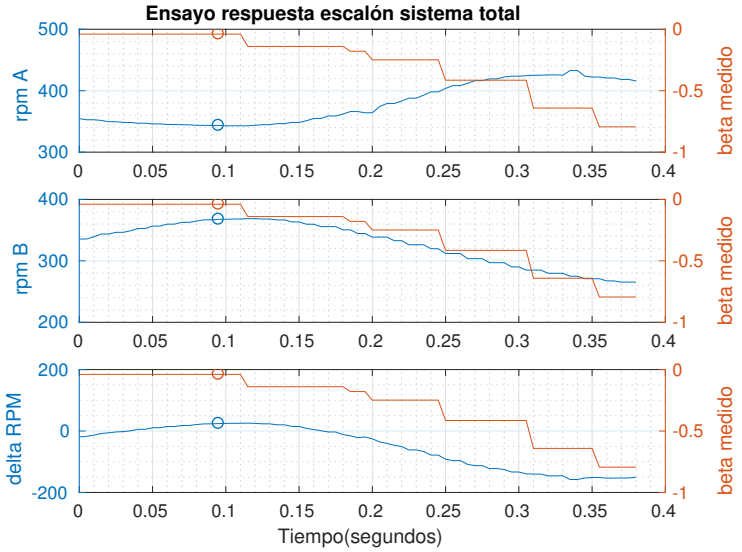


Figura 14: Resultado del ensayo de respuesta al escalón del sistema total. El circulo marca cuando se desactiva el control y el sistema comienza a trabajar en lazo abierto. En esta caso $\Delta\omega = -50$ RPM

Sin embargo, la implementación perfecta de un ensayo al escalón resulta muy difícil por dos inconvenientes. El primero es que se requiere que durante un tiempo el robot siga la línea perfectamente antes de girar. Para lograr esto se implementó un controlador proporcional ajustado a mano que permitía que el robot siga la pista. Así, el código implementado sigue la línea usando este controlador durante un breve lapso de tiempo que permita la estabilización del robot y luego desactiva el controlador y cambia abruptamente la diferencia de velocidad de los motores (manteniendo v constante), siendo los valores del ensayo los tomados a partir de este instante de tiempo. El segundo conflicto es que la señal que realmente se puede controlar no es la diferencia real de velocidad de los motores, sino la diferencia de velocidad deseada. La diferencia entre ambas señales está dada por la dinámica de los motores con sus controladores correspondientes. Como sólo se desea estimar la sección correspondiente a la cinemática, se optó por medir las señales de entrada y salida de esta sección, es decir, $\Delta\omega$ real y β, y utilizar esas señales para estimación con el System Identification de Matlab [13] . De esta manera, no se tiene exactamente la respuesta al escalón, pero la señal es lo suficientemente parecida como para resultar de utilidad en la estimación.

En la figura 14 se muestra uno de los resultados de este ensayo. Utilizando un conjunto de estos ensayos se estimo el siguiente sistema:

$$Sys_{tot} = \frac{0.028934}{s} \tag{32}$$

De la ecuación 31 y de 32 se concluye que $\frac{r}{L} = 0.028934$. Los parámetros medidos, si embargo, son $r = 1.5cm$ y $L = 16cm$, lo cual daría $\frac{r}{L} = 0.09375$. Esta discrepancia no se pudo explicar aún.

11. Ajuste de los Controladores

Existen diversas estrategias para realizar la sintonización de un controlador PID. Una de las posibilidades estudiadas en la carrera es la utilización de tablas empíricas, como son los métodos de Ziegler Nichols. Inicialmente se quiso implementar el ajuste con uno de estos métodos por considerárselo más didáctico y porque no requiere un modelo completo de la planta. Más adelante se detalla el procedimiento seguido, no obstante los controladores obtenidos no presentaban un desempeño satisfactorio.

Otras opciones estudiadas en la carrera requieren disponer de un modelo completo del sistema a controlar. Es por esto que se realizaron las estimaciones de funciones de transferencia explicadas en la sección 10. Conocidos los modelos de los sistemas, se puede realizar un ajuste con fundamentos matemáticos, pero dado que resulta más rápido y más intuitivo se optó por realizar un ajuste por software. Para ello se usó la herramienta de PID tuner de Matlab [14] que permite un ajuste muy fácil observando la curva de salida. Esto resulta de gran utilidad ya que se puede ver en la gráfica el impacto en el comportamiento efectivo, mientras que los cálculos teóricos estudiados en la carrera utilizan restricciones sobre los parámetros matemáticos que se relacionan de forma indirecta, aproximada y menos tangible con la respuesta temporal que realmente tendrá el sistema a lazo cerrado.

Para el ajuste de los controladores de los motores se buscó que la respuesta al escalón fuese rápida y con poco sobreimpulso, ya que el impacto se vería incrementado al considerar todo el robot en su conjunto. Además, se buscó un ajuste que diera un comportamiento muy similar para ambos motores, ya que las asimetrías se traducirían en trayectorias erráticas. Los controladores obtenidos para cada motor fueron los siguientes:

$$C_a(s) = \frac{19.243(s - 11.74)}{s}$$

$$C_b(s) = \frac{18.68(s - 10.93)}{s}$$

El ajuste se realizó en tiempo continuo, con lo cual resulta necesario realizar una discretización del controlador para poder implementarlo en el microcontrolador. La discretización utilizada fue la de Backward Euler, la cual realiza la siguiente sustitución:

$$s = \frac{z - 1}{zh}$$

Una vez implementados los controladores, se repitió el ensayo al escalón con el sistema a lazo cerrado. El resultado es el mostrado en la figura 15. Se observa que se pudo alcanzar el objetivo propuesto de reducir la asimetría de los motores.

11.1. Ajuste fallido por medio de Z-N

Según [10] como casi todos los controladores PID se ajustan en el sitio, en la literatura se han propuesto muchos tipos diferentes de reglas de sintonización, que permiten llevar a cabo una sintonización delicada y fina de los controladores PID en el sitio. Asimismo, se han desarrollado métodos automáticos de sintonización y algunos de los controladores PID poseen capacidad de sintonización automática en línea. Actualmente se usan en la industria formas modificadas del control PID, tales como el control I-PD y el control PID con dos grados de libertad.

La utilidad de los controles PID estriba en que se aplican en forma casi general a la mayoría de los sistemas de control. En particular, cuando el modelo matemático de la planta no se conoce y, por lo tanto, no se pueden emplear métodos de diseño analíticos, es cuando los controles PID resultan más útiles. En el campo de los sistemas para control de procesos, es un hecho bien conocido que los esquemas de control PID básicos y modificados han demostrado su utilidad para aportar un control satisfactorio, aunque tal vez en muchas situaciones específicas no aporten un control óptimo.

Las reglas de Ziegler-Nichols son muy convenientes cuando no se conocen los modelos matemáticos de las plantas. Tales reglas sugieren un conjunto de valores de K_p, T_i y T_d que darán una operación estable del sistema. No obstante, el sistema resultante puede presentar una gran sobreelongación en su respuesta escalón de forma que resulte no aceptable. En tales casos se necesitará una serie de ajustes finos hasta que se obtenga el resultado deseado. De hecho, las reglas de sintonía de Ziegler-Nichols dan una estimación razonable de los parámetros del controlador y proporcionan un punto de partida para una sintonía fina, en lugar de dar los parámetros K_p, T_i y T_d en un único intento.

Tal determinación de los parámetros de los controladores PID o sintonía de controladores PID se pueden realizar mediante experimentos sobre la planta. Hay dos métodos denominados reglas de sintonía de Ziegler-Nichols: método de la respuesta al esalón y método del $K_{inestable}$. En este trabajo se implemento el primero, por lo que sólo se detallara ese. En caso de querer profundizar mas en el tema, puede consultar [11] [12].

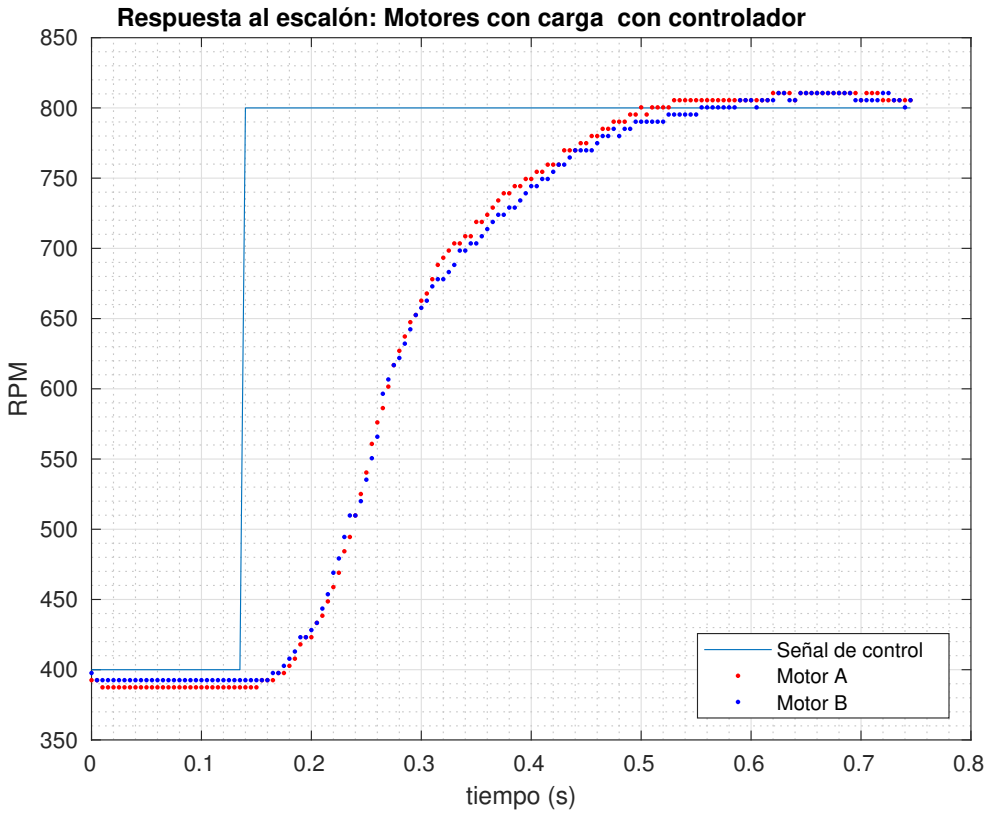


Figura 15: Resultado del ensayo de respuesta al escalón para ambos motores funcionando con los controladores PID calculados.

11.1.1. Metodo de Z-N: respuesta al escalón

Según [11] este método se basa en registrar la respuesta a un escalón unitario del sistema a lazo abierto, para luego ajustar los tres parámetros de un modelo de primer orden con retardo dado por la ecuación 33.

$$G_s(s) = \frac{K_p}{Ts+1}e^{-sL} \tag{33}$$

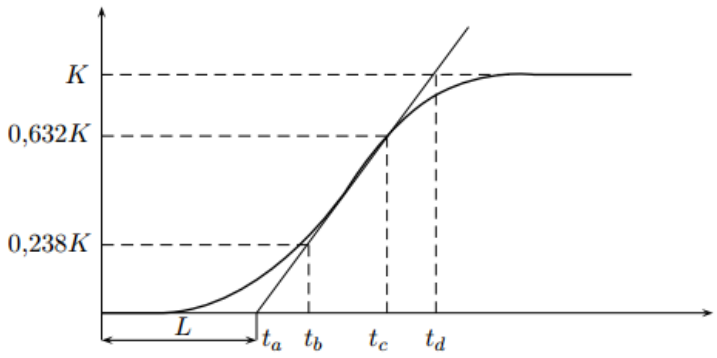


Figura 16: Determinación gráfica de los parámetros de un modelo a partir de la respuesta escalón en forma de S

Los parámetros del modelo de la ecuación 33 pueden ser determinados por medio de la figura 16. La ganancia estática K la obtenemos del valor final de la salida del proceso dividido el valor máximo de la entrada.

Para hallar los otros dos parámetros hay varias alternativas. Una de ellas consiste en trazar una recta tangente al punto donde de la respuesta al escalón presenta la mayor pendiente. El valor de la demora L lo obtenemos de la intersección de esta recta con el eje horizontal. El valor de la constante de tiempo T lo podemos obtener a partir de la distancia $t_a - t_c$, donde t_c es el tiempo donde la respuesta al escalón es $0.632K$. Otro método, un poco más preciso, consiste en determinar los puntos t_b y t_c donde $s(t)$ toma los valores $0.283K$ y

$0.632K$, respectivamente. Luego, la demora y la constante de tiempo la hallamos a partir de la ecuación 34

$$T = 1.5(t_c - t_b) \quad L = t_c - T \tag{34}$$

En como se menciono en la sección 11 este método no genero resultados satisfactorios en lo que el control de los motores respecta. Una de las causas reside en que este método es valido dentro de un rango de validez [11, pág. 60] dado por $0.1 < L/T < 1$. Para el caso que nos acomete, dado por el delay de medición, el parámetro L/T oscilaba entre 10 y 6, es decir $L/T \in (6, 10)$.

11.2. Sistema total y control

Si se parte de la suposición de que los motores se comportan de manera semejante, el diagrama del sistema total resulta:

A la hora de implementar el control se encontraron varios inconvenientes.... Poner Z-N, problemas de rozamiento y de inestabilidad; falta de linealidad a bajas RPM

12. Funcionamiento del Sistema Total

13. Conclusiones

Referencias

[1] `http://lamborghino.com`.

[2] `https://www.pololu.com/product/3061`

[3] `https://www.pololu.com/file/0J86/TB6612FNG.pdf`

[4] `https://store.arduino.cc/usa/arduino-nano`

[5] *Path Following Mobile Robot in the Presence of Velocity Constraints*, Bak, Poulsen y Ravn.

[6] *Robotics: Modelling, Planning and Control*, Siciliano, Sciavicco, Villani y Oriolo.

[7] Tesis de alexei.

[8] Dorf 10ma edicion; pagina 58.

[9] `https://sites.google.com/site/picuino/ziegler-nichols`.

[10] Ogata 5ta edicion capitulo 5

[11] PID que nos paso colon.

[12] `https://sites.google.com/site/picuino/ziegler-nichols`

[13] `https://la.mathworks.com/products/sysid.html`

[14] `https://la.mathworks.com/discovery/pid-tuning.html`

[15] Aström, Karl J. Sistemas controlados por computador. No. 04; TJ213, A8.. 1988.