

Navegación del Robot Móvil

Diana Marcela Ovalle Martínez



Proyecto Curricular de Ingeniería Electrónica
Facultad de Ingeniería
Universidad Distrital "Francisco José de Caldas"

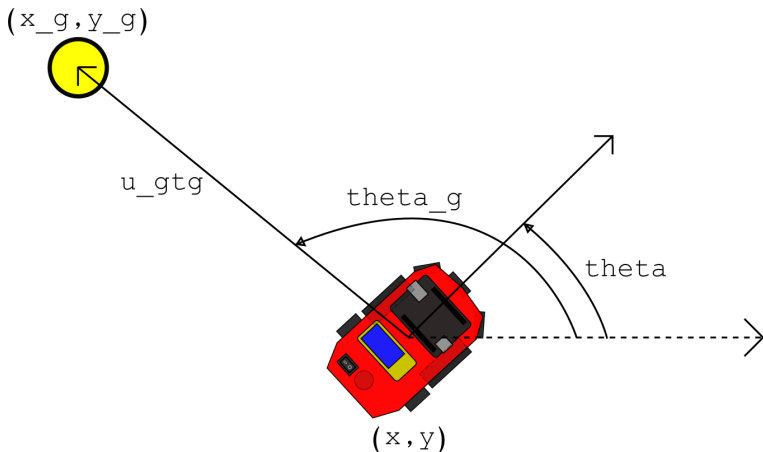
17 de marzo de 2020

Llegar a un punto deseado

El problema consiste en que estando en una posición y orientación determinada $[x_r, y_r, \theta_r]$, deseamos que el robot llegue a cierta posición deseada $[x_g, y_g]$.

Llegar a un punto deseado

El problema consiste en que estando en una posición y orientación determinada $[x_r, y_r, \theta_r]$, deseamos que el robot llegue a cierta posición deseada $[x_g, y_g]$. **Note que la posición deseada no incluye orientación, esta última se calcula para lograr llegar al punto deseado.**



Qué tenemos hasta ahora?

El modelo del vehículo

$$\dot{x}_r = u \cos(\theta_r)$$

$$\dot{y}_r = u \sin(\theta_r)$$

$$\dot{\theta}_r = r$$

$$\begin{aligned}\dot{u} &= \frac{1}{m} \left(\frac{1}{r_r} \overbrace{K_\tau \omega_d}^{\tau_d} + \frac{1}{r_r} \overbrace{K_\tau \omega_i}^{\tau_i} - X_f u - X_r u^2 + x_G r^2 \right) \\ \dot{r} &= \frac{1}{I_z} \left(\frac{L}{2 r_r} \overbrace{K_\tau \omega_d}^{\tau_d} - \frac{L}{2 r_r} \overbrace{K_\tau \omega_i}^{\tau_i} - N_f r - N_r r^2 - m x_G u r \right),\end{aligned}$$

Note que K_τ no es la constante de torque del motor que encuentra en los datos del fabricante del motor, sino un valor que asumimos constante, pero que en realidad depende del punto de operación del motor.

Cómo ajustamos el modelo del carro dadas las características de la dinámica asignadas?

Para la ecuación de \dot{u} , vamos a considerar un cambio en la fuerza de 10 N, dados K_τ y r_r para cada grupo y ninguna variación en el torque sobre el carro (movimiento en línea recta).

Note que una variación de 10 N implica que

$$10 = \frac{1}{r_r} K_\tau (\omega_d + \omega_i),$$

como el movimiento es en línea recta $\omega_d = \omega_i$, por lo que

$$\omega_d = \omega_i = \frac{10 r_r}{2 K_\tau}.$$

Para esos valores de ω_d y ω_i deben ser ajustados los parámetros X_f y X_r , recuerden que $x_G = 0$.

Cómo ajustamos el modelo del carro dadas las características de la dinámica asignadas? (cont. I)

Para la ecuación de \dot{r} , una vez ajustada la ecuación de \dot{u} , vamos a considerar un cambio en el torque de 0.05 Nm, dados K_τ , r_r y ω_d del movimiento en línea recta.

Note que una variación de 0.05 Nm implica que

$$0,05 = \frac{L}{2 r_r} K_\tau (\omega_d - \omega_i),$$

entonces

$$\omega_i = \omega_d - \frac{2 (0,05) r_r}{L K_\tau}.$$

Para esos valores de ω_d y ω_i deben ser ajustados los parámetros N_f y N_r , recuerden que $x_G = 0$.

Veáse el programa `dinamica_car.m`

Cómo agregamos la dinámica de los motores?

Recuerden que el modelo del motor derecho está dado por

$$G_d(s) = \frac{K_d}{(\tau_{d1} s + 1)(\tau_{d11} s + 1)}.$$

De forma similar, el modelo para el motor izquierdo está dado por

$$G_i(s) = \frac{K_i}{(\tau_{i1} s + 1)(\tau_{i11} s + 1)}.$$

Noten que para cada motor deben diseñar un controlador PI con el código proporcionado en clase. Es importante tener en cuenta que el objetivo de control de los dos motores, además de que tengan cierto tiempo de establecimiento y sobrepaso máximo, es lograr que las repuestas de lazo cerrado sean lo más parecidas posibles entre ellas.

Cómo agregamos la dinámica de los motores? (cont. I)

Para el diseño del PID, lo primero que debemos hacer es discretizar los modelos de los motores con la función `c2d`. Si la función de transferencia del motor está definida en la variable `motor` y el periodo de muestreo $T=0.01$ s, el modelo del motor discretizado se encontrará como

$$\text{motord} = \text{c2d}(\text{motor}, T)$$

Para cada motor definiremos tando sobrepaso máximo deseado, M_{pd} , como tiempo de establecimiento deseado, t_{sd} , para encontrar

$$\zeta_d = \sqrt{\frac{(\ln(M_{pd}))^2}{(\ln(M_{pd}))^2 + \pi^2}} \quad \text{y} \quad \underbrace{\omega_{nd} = \frac{4}{\zeta_d t_{sd}}}_{0 < \zeta_d < 0,7} \quad \text{o} \quad \omega_{nd} = \underbrace{\frac{6,5}{\zeta_d t_{sd}}}_{0,7 \leq \zeta_d < 1}.$$

Los polos deseados de lazo cerrado en discreto quedarán en

$$z_d = \exp \left(T \left(-\zeta_d \omega_{nd} \pm \omega_{nd} \sqrt{1 - \zeta_d^2} \right) \right).$$

Cómo agregamos la dinámica de los motores? (cont. II)

El proceso de diseño es equivalente al de tiempo continuo:

- 1 Hallamos el ángulo que debe aportar el controlador:

$$\angle PID(z)|_{z=z_d} = 180^\circ - \angle motor_d(z)|_{z=z_d}.$$

- 2 Encontramos el cero del PID discreto: Dada la función de transferencia de un controlador PID discreto

$$PID(z) = \frac{K(z - a)}{(z - 1)},$$

encontramos que

$$\angle(z - a)|_{z=z_d} = \angle PID(z)|_{z=z_d} + \angle(z - 1)|_{z=z_d}.$$

Hacemos el diagrama de polos y ceros del modelo del motor en tiempo discreto con el PID y z_d y, a partir de trigonometría, encontramos la ubicación de a .

Cómo agregamos la dinámica de los motores? (cont. III)

- 3 Utilizamos la condición de ángulo para encontrar K

$$\left\| \frac{K(z-a)}{(z-1)} motor_d(z) \right\|_{z=z_d} = 1.$$

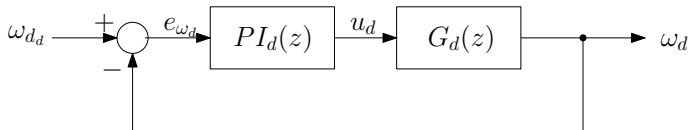
- 4 Verificamos las características de la respuesta en el tiempo y si no son satisfactorias, modificamos t_{sd} y/o M_{pd} y repetimos el diseño para el primer motor, hasta estar satisfechos. **Es importante recordar que la respuesta del segundo motor debe parecerse lo más posible a la del primero, con el fin de que el carro no sufra de desviaciones en los momentos iniciales.**

En este punto tendremos dos controladores, uno para el motor derecho, $Pl_d(z)$, y otro para el izquierdo, $Pl_i(z)$, con respuestas al escalón muy similares.

Veáse la primera sección del programa `control_motores.m` para detalles del diseño.

Cómo agregamos la dinámica de los motores? (cont. IV)

El lazo de control del motor derecho se muestra en la figura a continuación.



Dado que ω_{dd} no tiene porqué ser constante (tipo escalón), será útil que podamos simular la dinámica del lazo para cualquier valor de la señal de entrada.

Para esto, sabemos que la función de transferencia del PI de cada motor se puede escribir como

$$PI_d(z) = \frac{U_d(z)}{E_{\omega_d}(z)} = \frac{K(z - a)}{z - 1},$$

reescribiendo en potencias negativas de z , tenemos

$$\frac{U_d(z)}{E_{\omega_d}(z)} = \frac{K(1 - az^{-1})}{1 - z^{-1}}.$$

Cómo agregamos la dinámica de los motores? (cont. V)

Eliminando denominadores

$$U_d(z)(1 - z^{-1}) = E_{\omega_d}(z)(K(1 - a z^{-1})),$$

sacando transformada z inversa, tenemos

$$u_d[k] - u_d[k-1] = K e_{\omega_d}[k] - K a e_{\omega_d}[k-1].$$

Por lo que $u_d[k] = u_d[k-1] + K e_{\omega_d}[k] - K a e_{\omega_d}[k-1]$.

Note que debemos inicializar $u_d[k-1]$ y $e_{\omega_d}[k-1] = 0$, y luego actualizar para cada k .

Algo similar procede para el modelo en tiempo discreto del motor, que puede escribirse de manera genérica como

$$G_d(z) = \frac{\Omega_d(z)}{U_d(z)} = \frac{c_1 z^{-1} + c_2 z^{-2}}{1 + d_1 z^{-1} + d_2 z^{-2}},$$

donde podremos llegar a que

$$\omega_d[k] = -d_1 \omega_d[k-1] - d_2 \omega_d[k-2] + c_1 u_d[k-1] + c_2 u_d[k-2].$$

Cómo agregamos la dinámica de los motores? (cont. VI)

Note que para el motor debemos inicializar $\omega_d[k-1] = 0$, $\omega_d[k-2] = 0$, $\omega_d[k-1] = 0$ y $\omega_d[k-2] = 0$, e ir las actualizando conforme se van haciendo calculos para diferentes valores de k .

Así las cosas, teniendo en cuenta que $\omega_d[k]$ no depende de $u_d[k]$, sino de valores retrasados de u_d , para cada valor de k tendremos:

while ON

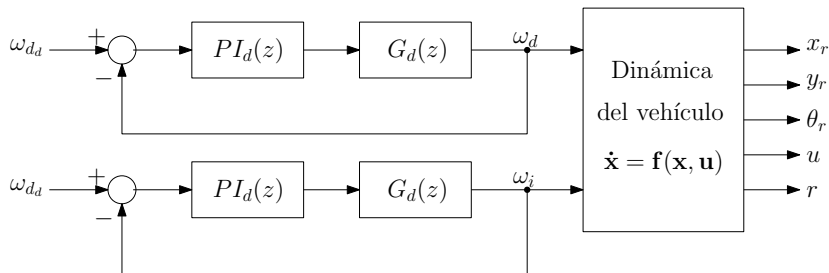
- ① $\omega_d[k] = -d_1 \omega_d[k-1] - d_2 \omega_d[k-2] + c_1 u_d[k-1] + c_2 u_d[k-2]$.
- ② $e_{\omega_d}[k] = \omega_{d_d}[k] - \omega_d[k]$.
- ③ $u_d[k] = u_d[k-1] + K e_{\omega_d}[k] - K a e_{\omega_d}[k-1]$.
- ④ Se actualizan los registros de los valores anteriores
 $\omega_d[k-2] \leq \omega_d[k-1]$, $\omega_d[k-1] \leq \omega_d[k]$, $u_d[k-2] \leq u_d[k-1]$,
 $u_d[k-1] \leq u_d[k]$, $e_{\omega_d}[k-1] \leq e_{\omega_d}[k]$.
- ⑤ $k = k + 1$ y volvemos al paso 1.

end

Veáse la segunda sección del programa `control_motores.m` para evidenciar que con el ciclo anterior se puede reproducir la respuesta al escalón de los motores.

Carro en lazo abierto

En este punto tenemos un modelo en lazo abierto del carro al que le ingresan valores deseados de la velocidad angular de ambos motores, ω_{d_d} y ω_{i_d} , que se calcularon para la dinámica del carro, incluye la dinámica de los mismos en un lazo de control y también incluye la dinámica del carro, que nos da información de la evolución en el tiempo de las variables x_r , y_r , θ_r , u y r .

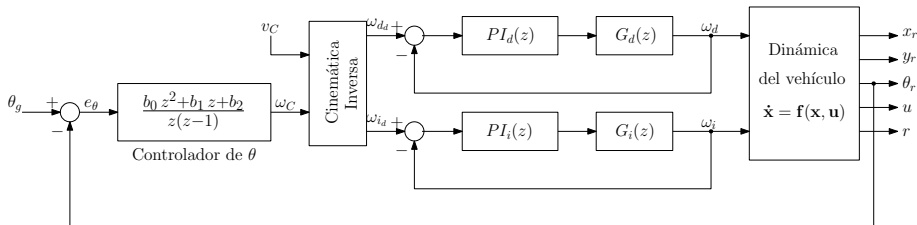


Veáse el programa `dinamica_motor_car.m`

Primera aproximación - Llegar a un punto deseado

Para esta aproximación vamos a asumir que si el vehículo tiene la orientación del vector que sale de su ubicación inicial (x_0, y_0) al punto deseado (x_g, y_g) , el vehículo puede llegar al punto deseado o a una vecindad del mismo.

Por lo que, $\theta_g = \tan^{-1} \left(\frac{y_g - y_0}{x_g - x_0} \right)$ y diseñaremos un lazo de control que nos permita tener la orientación deseada.



Primera aproximación - Llegar a un punto deseado (cont. I)

Noten que el controlador de θ es un PID digital, cuya función de transferencia original es de la forma

$$C_{\theta}(z) = K_p + K_i \frac{T_c z}{z - 1} + K_d \frac{z - 1}{T_c z},$$

que puede ser reescrita como

$$C_{\theta}(z) = \frac{K_p T_c z (z - 1) + K_i T_c^2 z^2 + K_d (z - 1)^2}{T_c z (z - 1)},$$

desarrollando los productos notables

$$C_{\theta}(z) = \frac{K_p T_c z^2 - K_p T_c z + K_i T_c^2 z^2 + K_d z^2 - 2 K_d z + K_d}{T_c z (z - 1)},$$

y agrupando términos

$$C_{\theta}(z) = \frac{(K_p T_c + K_i T_c^2 + K_d) z^2 - (K_p T_c + 2 K_d) z + K_d}{T_c z (z - 1)}.$$

Por lo que: $b_0 = \frac{K_p T_c + K_i T_c^2 + K_d}{T_c}$, $b_1 = \frac{-(K_p T_c + 2 K_d)}{T_c}$ y $b_2 = \frac{K_d}{T_c}$.

Primera aproximación - Llegar a un punto deseado (c. II)

Para implementar el controlador definido por

$$C_{\theta}(z) = \frac{\Omega_C(z)}{E_{\theta}(z)} = \frac{b_0 z^2 + b_1 z + b_2}{z(z-1)},$$

tomamos la segunda igualdad y escribimos la función de transferencia en potencias negativas de z

$$\frac{\Omega_C(z)}{E_{\theta}(z)} = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2}}{1 - z^{-1}},$$

eliminamos denominadores

$$\Omega_C(z)(1 - z^{-1}) = E_{\theta}(z)(b_0 + b_1 z^{-1} + b_2 z^{-2}),$$

y sacamos transformada z inversa

$$\omega_C[k] - \omega_C[k-1] = b_0 e_{\theta}[k] + b_1 e_{\theta}[k-1] + b_2 e_{\theta}[k-2].$$

De donde finalmente tenemos

$$\boxed{\omega_C[k] = \omega_C[k-1] + b_0 e_{\theta}[k] + b_1 e_{\theta}[k-1] + b_2 e_{\theta}[k-2]}.$$

Primera aproximación - Llegar a un punto deseado (c. III)

Noten que debemos inicializar los valores de $\omega_C[k-1] = 0$, $e_\theta[k-1] = 0$ y $e_\theta[k-2] = 0$, e irlos actualizando en la medida que se vayan haciendo cálculos.

Para ajustar el controlador, el periodo de muestreo del lazo del carro será $T_c = 0,1$, se propone empezar con $K_p \neq 0$ y $K_i = K_d = 0$, variar K_p y empezar a aumentar levemente los valores de K_i y K_d , hasta lograr la respuesta deseada.

Note que esta primera aproximación tiene el inconveniente de que, aunque garanticemos la orientación final del vehículo, siempre vamos a tener un error, dependiendo de en qué punto se logre tener constante la orientación y si salimos o no del origen.

Tarea: Para el lunes 23 de marzo a más tardar a las 11:59 pm, deben enviar al correo por grupos la implementación en Matlab del control en lazo cerrado del vehículo para llegar a un punto deseado (por la primera aproximación), partiendo de los programas proporcionados.

Segunda aproximación - Llegar a un punto deseado

Para superar el problema de la primera aproximación, la orientación deseada del vehículo no va a ser constante, sino variable y dependerá del vector $\mathbf{u} = \mathbf{p}_g - \mathbf{p}_r$. Siendo \mathbf{p}_g la posición a la que deseamos ir y \mathbf{p}_r la posición actual del vehículo.

- Para calcular el ángulo deseado, θ_g , debemos

$$\begin{aligned}u_x[k] &= x_g - x_r[k], \\u_y[k] &= y_g - y_r[k], \\ \theta_g[k] &= \tan^{-1} \left(\frac{u_y[k]}{u_x[k]} \right), \quad -\pi < \theta < \pi.\end{aligned}$$

A partir del θ_g se calcula el error en el ángulo

$$e_\theta[k] = \theta_g - \theta_r[k],$$

y sobre e_θ actúa el controlador que genera la velocidad angular del vehículo.

Tarea: A la entrega del lunes 23 de marzo se debe sumar la implementación de la segunda aproximación, verificando que se soluciona el problema de la primera aproximación.

Llegar a un punto deseado controlando v_C

Lo último que hace falta por considerar es el control de v_C . Como lo mencionamos en clase, lo ideal sería que el carro fuera más rápido entre más lejos del punto de destino se encuentre y más despacio entre más cerca del punto de destino se encuentre. Así las cosas, definimos la distancia de donde estamos (x_r, y_r) a donde queremos ir (x_g, y_g) , como

$$d[k] = \sqrt{(x_g - x_r[k])^2 + (y_g - y_r[k])^2},$$

esa la distancia, será la entrada a otro controlador PID, que nos generará v_C . El controlador lo definimos, similar a como definimos de forma genérica $C_\theta(z)$ en la diapositiva 16,

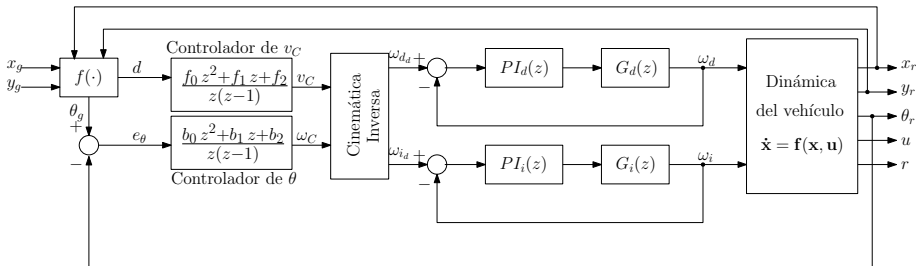
$$C_v(z) = \frac{v_C(z)}{D(z)} = \frac{f_0 z^2 + f_1 z + f_2}{z(z-1)},$$

que se puede implementar como

$$v_C[k] = v_C[k-1] + f_0 d[k] + f_1 d[k-1] + f_2 d[k-2].$$

Llegar a un punto deseado controlando v_C (cont. I)

El diagrama de bloques resultante del sistema tanto con control de ω_C como de v_C se muestra a continuación.



donde $f(\cdot)$ representa las funciones para obtener $\theta_g[k]$ y $d[k]$ a partir de las coordenadas de la ubicación actual (x_r, y_r) y el punto al que queremos llegar (x_g, y_g).

Tarea: A la entrega del lunes 23 de marzo, se debe incluir la implementación de la segunda aproximación, con el control de $v_C \Rightarrow$ **enviar 3 programas distintos en .m (no Live Script).**