Robótica Móvil

Segundo cuatrimestre de 2025

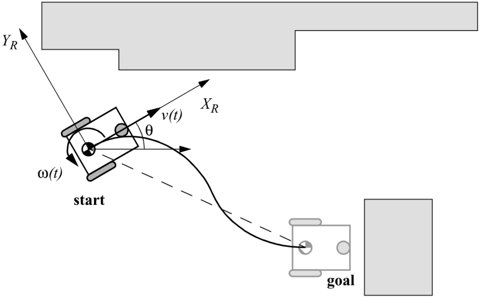
Departamento de Computación - FCEyN - UBA

Planificación de movimientos - clase 8

Seguimiento de trayectorias a lazo cerrado

Recordemos el problema

Queremos ir de una pose a otra. Estas poses van a ser puntos vía o puntos de paso de un camino más largo que me va a dar el path planner

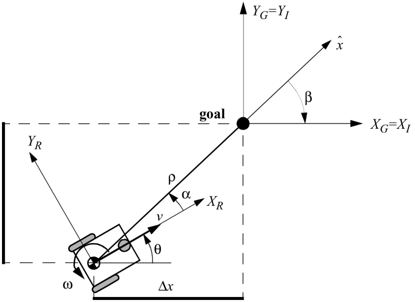


→ La clase pasada vimos cómo resolver este problema a lazo abierto, i.e. no tenemos un feedback de los movimientos, asumimos que el robot se mueve siguiendo perfectamente las consignas de velocidades.

→ Hoy vamos a resolver este problema a lazo cerrado, i.e. asumimos que tenemos un feedback en tiempo real del estado (la pose en este caso) del robot.

Definición del problema

- Vamos a considerar (por ahora!!!), sin perder generalidad que la pose objetivo (*goal*) coincide con el marco inercial (marco global).
- El error en cada instante entre la pose actual del robot y la pose objetivo es $(\Delta x, \Delta y, \theta)^\top$.
- Podemos calcular el error porque vamos a asumir que conocemos la pose del robot en todo momento (lazo cerrado).



Definición del problema

La forma del controlador que vamos a proponer consiste en hallar una matriz $\boldsymbol{K}$ que relacione el error del estado del robot (pose respecto del goal) y las variables de control:

$$\boldsymbol{u}(t) = \boldsymbol{K}\,\boldsymbol{e}(t)$$

$$\underbrace{\boldsymbol{u}(t)}_{\text{control}} = \underbrace{\boldsymbol{K}}_{\text{matriz de ganancia}} \underbrace{\boldsymbol{e}(t)}_{\text{error de estado}} \quad \text{tal que} \quad \lim_{t\to\infty} \boldsymbol{e}(t) = 0.$$

$$\boldsymbol{K} = \begin{pmatrix} k_{1,1} & k_{1,2} & k_{1,3} \\ k_{2,1} & k_{2,2} & k_{2,3} \end{pmatrix}$$

donde $k_{i,j} = k(t, \boldsymbol{e}(t))$, es decir, la ganancia se ajusta dinámicamente según el error o el instante de tiempo de forma tal que el control de $v(t)$ y $\omega(t)$ queda definido como:

$$\begin{pmatrix} v(t) \\ \omega(t) \end{pmatrix} = \boldsymbol{K}\,\boldsymbol{e}(t) = \boldsymbol{K}. \begin{pmatrix} x \\ y \\ \theta \end{pmatrix}^{I} \text{cumpla que} \lim_{t\to\infty} \boldsymbol{e}(t) = 0.$$

Importante: A esto se lo llama controlador no lineal y adaptativo, $\boldsymbol{K}$ puede variar en el tiempo o según el estado, a diferencial del controlador PID clásico que vimos antes que es lineal y estático.

Modelo cinemático de un robot diferencial

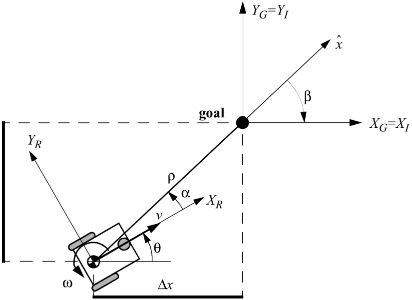
Recordemos que en el caso de un robot diferencial para pasar las velocidades del marco del robot al marco inercial teníamos que hacer:

$$^{I}\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{pmatrix} = {}^{I}\boldsymbol{R}_R(\theta) \; ^{R}\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{pmatrix} = \begin{pmatrix} cos(\theta) & 0 \\ sen(\theta) & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} v \\ \omega \end{pmatrix}$$

ya que $v = {}^{R}\dot{x}$, ${}^{R}\dot{y} = 0$ (restricccción no-holonómica) y que $\omega = {}^{R}\dot{\theta}$.

Control de movimiento a lazo cerrado

- Sea α el ángulo entre el eje X_R en el marco de referencia del robot y el vector $\hat{\boldsymbol{x}}$ que conecta el origen del marco del robot con el origen del marco de la pose objetivo (el ángulo entre la orientación actual del robot y "mirar hacia el goal").
- Sea θ es el ángulo entre la orientación del goal y la del robot. Es decir, la orientación del robot respecto al marco inercial.
- Si $\alpha \in (-\frac{\pi}{2}, \frac{\pi}{2}]$ podemos considerar la transformación de coordenadas euclídeas a coordenadas polares con su origen en el marco objetivo:



$$\rho = \sqrt{\Delta x^2 + \Delta y^2} \qquad \text{(distancia al goal)}$$
$$\alpha = \operatorname{atan2}(\Delta y, \Delta x) - \theta \qquad \text{(ángulo al goal)}$$
$$\beta = -\theta - \alpha \qquad \text{(ángulo entre el marco inercial o goal y el vector } \hat{\boldsymbol{x}}\text{)}$$

Derivación de $\dot{\rho}$

Recordemos:

$$\rho = \sqrt{(\Delta x)^2 + (\Delta y)^2}, \qquad \dot{x} = v\cos(\theta), \quad \dot{y} = v\sin(\theta)$$

Pero ni Δx ni Δy son constantes: ambas cambian en el tiempo, porque el robot se mueve. Por lo tanto, ρ depende del tiempo de forma indirecta, es decir:

$$\rho = \rho(\Delta x(t), \Delta y(t)).$$

Cuando una función depende del tiempo a través de otras variables, se usa la **regla de la cadena multivariable**:

$$\dot{\rho} = \frac{d\rho}{dt} = \frac{\partial \rho}{\partial \Delta x}\frac{d(\Delta x)}{dt} + \frac{\partial \rho}{\partial \Delta y}\frac{d(\Delta y)}{dt}.$$

Cuando hacemos una derivada parcial con respecto a Δx, consideramos que Δy es constante. Entonces:

$$\frac{\partial \rho}{\partial \Delta x} = \frac{\partial}{\partial \Delta x}\left[(\Delta x^2 + \Delta y^2)^{1/2}\right] = \frac{1}{2}(\Delta x^2 + \Delta y^2)^{-1/2} \cdot 2\Delta x = \frac{\Delta x}{\sqrt{(\Delta x)^2 + (\Delta y)^2}} = \frac{\Delta x}{\rho}.$$

Por otro lado, sabemos que $\Delta x = x_g - x$ y que $\Delta y = y_g - y$. Entonces, derivando nos queda:

$$\dot{\Delta x} = \frac{d(\Delta x)}{dt} = \frac{d(x_g - x)}{dt} = \dot{x_g} - \dot{x} = -\dot{x} = -v\cos(\theta).$$

ya que el objetivo está fijo ($\dot{x}_g = \dot{y}_g = 0$).
Luego, podemos reemplazar en lo que teníamos más arriba:

$$\dot{\rho} = \frac{d\rho}{dt} = \frac{\partial \rho}{\partial \Delta x}\frac{d(\Delta x)}{dt} + \frac{\partial \rho}{\partial \Delta y}\frac{d(\Delta y)}{dt} = \frac{\Delta x(-v\cos(\theta))}{\rho} + \frac{\Delta y(-v\sin(\theta))}{\rho} = \frac{-v(\Delta x\cos(\theta) + \Delta y\sin(\theta))}{\rho}$$

Derivación de $\dot{\rho}$

Ya sabemos que:

$$\dot{\rho} = \frac{-v(\Delta x \cos(\theta) + \Delta y \sin(\theta))}{\rho}.$$

Ahora queremos expresar esto en función del ángulo α. Recordemos que:

$$\alpha = \operatorname{atan2}(\Delta y, \Delta x) - \theta \quad \text{y} \quad \beta = -\theta - \alpha = -\operatorname{atan2}(\Delta y, \Delta x)$$

Entonces $-\beta = \operatorname{atan2}(\Delta y, \Delta x)$ y por trigonometría:

$$\cos(-\beta) = \frac{\Delta x}{\rho}, \qquad \sin(-\beta) = \frac{\Delta y}{\rho}.$$

Aplicando la identidad del coseno de una diferencia:

$$\cos((-\beta) - \theta) = \cos(-\beta)\cos(\theta) + \sin(-\beta)\sin(\theta)$$

y considerando que $\alpha = -\beta - \theta$, se obtiene:

$$\cos\alpha = \cos(-\beta - \theta) = \frac{\Delta x}{\rho}\cos(\theta) + \frac{\Delta y}{\rho}\sin(\theta) = \frac{\Delta x \cos(\theta) + \Delta y \sin(\theta)}{\rho}.$$

Sustituyendo en la expresión de $\dot{\rho}$:

$$\dot{\rho} = -v\frac{\Delta x \cos(\theta) + \Delta y \sin(\theta)}{\rho} = -v\cos(\alpha).$$

$$\boxed{\dot{\rho} = -v\cos(\alpha).}$$

Derivación de $\dot{\alpha}$

$$\alpha = \mathrm{atan2}(\Delta y, \Delta x) - \theta$$

Derivando con respecto al tiempo:

$$\dot{\alpha} = \frac{d}{dt}[\mathrm{atan2}(\Delta y, \Delta x)] - \dot{\theta}.$$

Sabemos que

$$\frac{d}{dt}[\mathrm{atan2}(y, x)] = \frac{x\dot{y} - y\dot{x}}{x^2 + y^2},$$

por lo tanto:

$$\dot{\alpha} = \frac{d}{dt}[\mathrm{atan2}(\Delta y, \Delta x)] - \dot{\theta} = \frac{\Delta x \dot{\Delta y} - \Delta y \dot{\Delta x}}{\rho^2} - \dot{\theta}.$$

Sustituyendo $\dot{\Delta x} = -v\cos(\theta)$, $\dot{\Delta y} = -v\sin(\theta)$, $\dot{\theta} = \omega$:

$$\dot{\alpha} = \frac{\Delta x(-v\sin(\theta)) - \Delta y(-v\cos(\theta))}{\rho^2} = \frac{v(\Delta y\cos(\theta) - \Delta x\sin(\theta))}{\rho^2} - \omega.$$

Aplicando la identidad del seno de una diferencia:

$$\sin(\alpha) = \sin(-\beta - \theta) = \sin(-\beta)\cos(\theta) - \cos(-\beta)\sin(\theta) = \frac{\Delta y}{\rho}\cos(\theta) - \frac{\Delta x}{\rho}\sin(\theta).$$

Multiplicando ambos lados por ρ:

$$\rho\sin(\alpha) = \Delta y\cos(\theta) - \Delta x\sin(\theta).$$

Reemplazamos eso en la ecuación anterior:

$$\dot{\alpha} = \frac{v(\Delta y\cos(\theta) - \Delta x\sin(\theta))}{\rho^2} - \omega = \frac{v}{\rho^2}(\rho\sin(\alpha)) - \omega = \frac{v}{\rho}\sin(\alpha) - \omega.$$

$$\boxed{\dot{\alpha} = \frac{v}{\rho}\sin(\alpha) - \omega.}$$

Derivación de $\dot{\beta}$

A partir de la definición:

$$\beta = -\theta - \alpha,$$

derivamos respecto del tiempo:

$$\dot{\beta} = -\dot{\theta} - \dot{\alpha}.$$

Como $\dot{\theta} = \omega$, resulta:

$$\dot{\beta} = -\omega - \dot{\alpha}.$$

Sustituyendo la expresión de $\dot{\alpha} = \frac{v}{\rho}\sin(\alpha) - \omega$:

$$\dot{\beta} = -\omega - \left(\frac{v}{\rho}\sin(\alpha) - \omega\right) = -\frac{v}{\rho}\sin(\alpha).$$

$$\boxed{\dot{\beta} = -\frac{v}{\rho}\sin(\alpha).}$$

Control de movimiento a lazo cerrado

Entonces, la descripción del modelo en el nuevo sistema de coordenadas polares nos queda como:

$$\dot{\rho} = -v\cos(\alpha) \qquad \dot{\alpha} = \frac{v}{\rho}\sin(\alpha) - \omega \qquad \dot{\beta} = -\frac{v}{\rho}\sin(\alpha).$$

De manera matricial podemos reescribirlo:

$$^{I}\begin{pmatrix} \dot{\rho} \\ \dot{\alpha} \\ \dot{\beta} \end{pmatrix} = \begin{pmatrix} -\cos(\alpha) & 0 \\ \sin(\alpha)/\rho & -1 \\ -\sin(\alpha)/\rho & 0 \end{pmatrix} \begin{pmatrix} v \\ \omega \end{pmatrix}$$

donde: $^{I}\dot{\rho}$ es la velocidad a la cual el robot se acerca al goal, $^{I}\dot{\alpha}$ es la velocidad a la cual el robot se orienta hacia el goal y $^{I}\dot{\beta}$ es la velocidad a la cual el robot se orienta hacia donde tiene que quedar orientado en el goal.

Importante: esto es válido en tanto y cuanto $\alpha \in (-\frac{\pi}{2}, \frac{\pi}{2}]$, que equivale a pensar que el goal está siempre hacia adelante del robot (lookahead). Vamos a asumir esto.
Problema: Este sistema tiene una singularidad en $\rho = 0$.

Control de movimiento a lazo cerrado

Las consignas de control v y ω deben ser definidas para guiar al robot desde su pose actual $(\rho_0, \alpha_0, \beta_0)$ hasta la pose final. Acá se presenta una discontinuidad para $\rho = 0$.

En este punto, tenemos que proponer una ley de control (la matriz $\boldsymbol{K}$). Una bien conocida para el modelo diferencial es la siguiente:

$$\begin{aligned} v &= k_\rho \rho \\ \omega &= k_\alpha \alpha + k_\beta \beta \end{aligned}$$

el sistema de control a lazo cerrado nos queda:

$$\begin{pmatrix} \dot{\rho} \\ \dot{\alpha} \\ \dot{\beta} \end{pmatrix} = \begin{pmatrix} -\cos(\alpha) & 0 \\ \sin(\alpha)/\rho & -1 \\ -\sin(\alpha)/\rho & 0 \end{pmatrix} \begin{pmatrix} v \\ \omega \end{pmatrix} = \begin{pmatrix} -k_\rho \rho \cos(\alpha) \\ k_\rho \sin(\alpha) - k_\alpha \alpha - k_\beta \beta \end{pmatrix}$$

- → Este sistema no tiene singularidad en $\rho = 0$ y tiene un punto único de equilibrio en $(\rho, \alpha, \beta) = (0, 0, 0)$ (donde las derivadas se anulan).
- → Para que el sistema sea estable debe valer que $k_\rho > 0, k_\beta < 0$ y $k_\rho < k_\alpha$.
- → Esto quiere decir, que con esas condiciones el sistema va a converger al punto de equilibrio (que es mi goal!).
- → Además, si $\alpha \in (-\frac{\pi}{2}, \frac{\pi}{2}]$ en el tiempo inicial, entonces se va a cumplir para todos los tiempos siguientes (se mantiene el lookahead).
- → Esto se demuestra en Teoría de Control no lineal por el Teorema de Lyapunov (no lo vamos a ver en la materia).

Para tener en cuenta

Dijimos que ibamos a considerar que la pose objetivo (goal) coincide con el marco inercial para hacer más fáciles las cuentas. Pero, ¿qué pasa si no es así? (en el Taller por ejemplo, el marco inercial está fijo y no es el goal)

Tenemos que hacer un cambio de coordenas (ya sabemos cómo hacerlo!):

$$
\begin{aligned}
{}^{I}\Delta x &= {}^{I}x_g - {}^{I}x_r \\
{}^{I}\Delta y &= {}^{I}y_g - {}^{I}y_r
\end{aligned}
$$

y luego podemos expresar ese vector en el marco del **goal**:

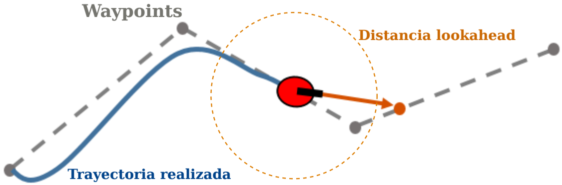
$$
\begin{pmatrix} {}^{G}\Delta x \\ {}^{G}\Delta y \end{pmatrix} = {}^{G}\mathrm{R}_I({}^{G}\theta_I) \begin{pmatrix} {}^{I}\Delta x \\ {}^{I}\Delta y \end{pmatrix} = \begin{pmatrix} cos({}^{G}\theta_I) & sen({}^{G}\theta_I) \\ -sen({}^{G}\theta_I) & cos({}^{G}\theta_I) \end{pmatrix} \begin{pmatrix} {}^{I}\Delta x \\ {}^{I}\Delta y \end{pmatrix} \quad \text{con } {}^{G}\theta_I = -{}^{I}\theta_G.
$$

Además, el error angular entre el robot y el goal puede escribirse como:

$$
{}^{G}\theta_R = {}^{I}\theta_R - {}^{I}\theta_G.
$$

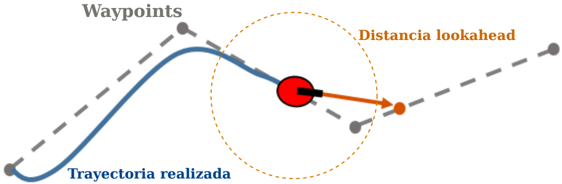
Algoritmo de siga la zanahoria (pure pursuit)

1. Determinar la pose actual del robot.
2. Encontrar el punto del camino más cercano al robot.
3. Encontrar un punto objetivo (goal) delante del robot (lookahead).
4. Transformar el goal a las coordenadas del robot.
5. Realizar el control a lazo cerrado hacia el goal.
6. Volver al paso 1.

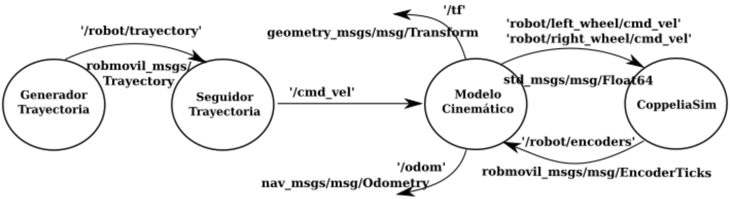


Cómo encontrar el siguiente goal

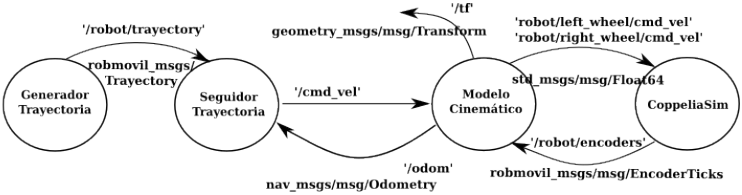
- → Verifico si estoy cerca del waypoint final del camino. Si es así, lo seteo como goal. Sino:
- → Busco el waypoint más cercano (tengo que recorrer todos los waypoints y medir la distancia euclídea).
- → A partir del más cercano busco el siguiente waypoint que se distancie al menos el lookahead definido.
- → Este es el waypoint que seteamos como goal.



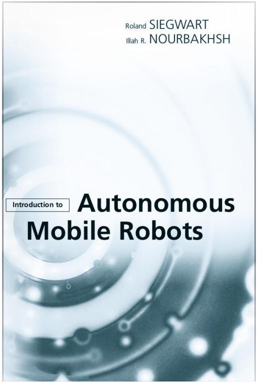
Para el Taller, ¿de dónde venimos?



Para el Taller, ¿qué tenemos que hacer?



Más sobre seguimiento de trayectorias



"Introduction to autonomous mobile robots", Siegwart, Roland, Illah Reza Nourbakhsh, Davide Scaramuzza. MIT press, 2011. **Capítulo 3**