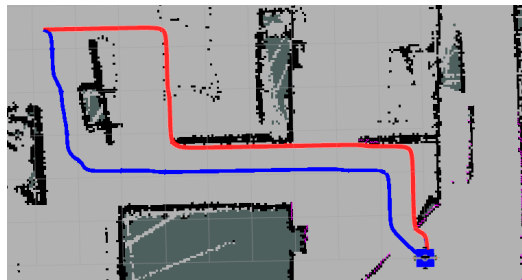


- Si se tiene un mapa basado en celdas de ocupación, las rutas se pueden planear mediante algoritmos de búsqueda en grafos.
- A\* es un algoritmo que explora la ruta con el menor costo esperado, calculado como:

$$f(n) = g(n) + h(n)$$

donde:

- $g(n)$  es el costo desde el nodo inicial hasta el nodo  $n$
- $h(n)$  es una heurística que subestima el costo desde  $n$  hasta el nodo meta.
- $g(n)$  debe diseñarse de acuerdo con lo que se quiera minimizar: distancia recorrida, energía consumida o, en el caso de este trabajo, el riesgo de colisionar con algún obstáculo.



**Se propone la función de costo:**

$$g(n) = d(n) + r(n)$$

donde:

- $d(n)$  es la distancia recorrida y
- $r(n)$  es inversamente proporcional a la distancia de  $n$  al objeto más cercano.

La figura muestra una comparación entre ambas rutas.

Dado que la ruta se calcula a partir de celdas, las vueltas siempre son “esquinas”. Esto no es deseable porque las discontinuidades producen señales de control discontinuas, lo que puede dañar los motores. Para suavizar la ruta, se minimiza una función de costo que pondera el suavizado y el parecido con la ruta original:

$$V = \frac{1}{2}\alpha \sum_{i=1}^n (p_i - q_i)^2 + \frac{1}{2}\beta \sum_{i=1}^{n-1} (p_i - p_{i+1})^2 \quad (1)$$

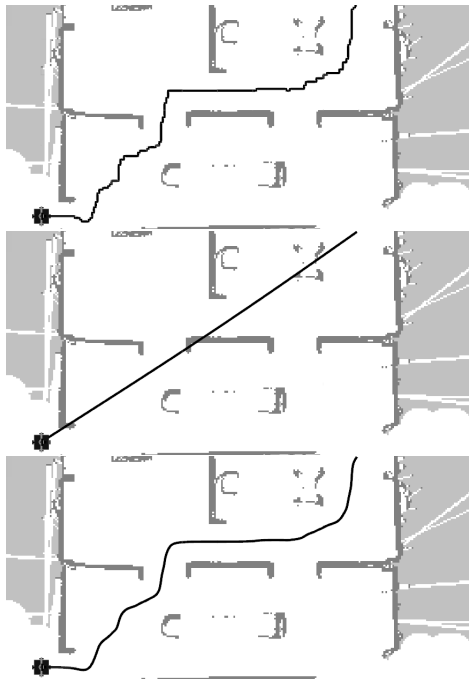
donde  $Q = \{q_1 \dots q_n\}$  son todos los puntos  $q = [x_q \ y_q]$  que forman la ruta original calculada con  $A^*$  y  $P = \{p_1 \dots p_n\}$  son los puntos  $p = [x_p \ y_p]$  de la ruta ya suavizada.

- $\alpha \geq 0$  indica el peso dado al parecido con la ruta original.
- $\beta \geq 0$  indica el peso de la distancia entre punto y punto.

El gradiente  $\nabla V$  queda de la forma:

$$\left[ \underbrace{\alpha(p_1 - q_1) + \beta(p_1 - p_2)}_{\frac{\partial V}{\partial p_1}}, \dots, \underbrace{\alpha(p_i - q_i) + \beta(2p_i - p_{i-1} - p_{i+1})}_{\frac{\partial V}{\partial p_i}}, \dots, \underbrace{\alpha(p_n - q_n) + \beta(p_n - p_{n-1})}_{\frac{\partial V}{\partial p_n}} \right] \quad (2)$$

Para minimizar la función  $V$  basta con variar  $p_i$  en sentido contrario a  $\nabla V$ .



### Data:

Puntos  $Q = \{q_1 \dots q_i \dots q_n\}$  de la ruta original

Parámetros  $\alpha$  y  $\beta$  Tolerancia  $\epsilon$

**Result:** Puntos  $P = \{p_1 \dots p_i \dots p_n\}$  de la ruta suavizada

**foreach**  $p_i \in P$  **do**

$p_i \leftarrow q_i$

**end**

**while**  $\|\nabla V(p_i)\| > \epsilon$  **do**

$x_1^p \leftarrow \alpha(x_1^p - x_1^q) + \beta(x_1^p - x_2^p)$

$y_1^p \leftarrow \alpha(y_1^p - y_1^q) + \beta(y_1^p - y_2^p)$

**foreach**  $i \in [2, n-1]$  **do**

$x_i^p \leftarrow \alpha(x_i^p - x_i^q) + \beta(2x_i^p - x_{i+1}^p - x_{i-1}^p)$

$y_i^p \leftarrow \alpha(y_i^p - y_i^q) + \beta(2y_i^p - y_{i+1}^p - y_{i-1}^p)$

**end**

$x_n^p \leftarrow \alpha(x_n^p - x_n^q) + \beta(x_n^p - x_{n-1}^p)$

$y_n^p \leftarrow \alpha(y_n^p - y_n^q) + \beta(y_n^p - y_{n-1}^p)$

**end**

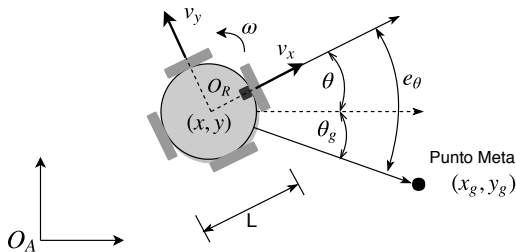
regresar  $P$

Considere una base omnidireccional cuyo modelo cinemático está dado por:

$$\dot{x} = v_x \cos \theta - v_y \sin \theta \quad (8)$$

$$\dot{y} = v_x \sin \theta + v_y \cos \theta \quad (9)$$

$$\dot{\theta} = \omega, \quad (10)$$



Con  $v_x, v_y, \omega$ , las velocidades lineal y angular respectivamente, consideradas como señales de control.

Se desea alcanzar el punto  $(x_g, y_g)$ . Para ello se calcula un error de ángulo:

$$e_\theta = \text{atan2}(y_g - y, x_g - x) - \theta$$

Las leyes de control

$$v_x = v_{\max} e^{-\frac{e_\theta^2}{\alpha}} \quad (11)$$

$$v_y = 0 \quad (12)$$

$$\omega = \omega_{\max} \left( \frac{2}{1 + e^{-\frac{e_\theta}{\beta}}} - 1 \right) \quad (13)$$

Garantizan que el robot alcanzará la posición deseada.

$v_{\max}$  y  $\omega_{\max}$  se fijan de acuerdo con las capacidades del robot y  $\alpha$  y  $\beta$  se usan para evitar oscilaciones no deseadas.

$v_y = 0$  sirve para lograr un movimiento similar a una base diferencial.

Las leyes de control (11)-(13) tienen las siguientes desventajas:

- Sólo dependen del error  $e_\theta$ , por lo que el robot no desacelera conforme se acerca al punto meta.
- Lo anterior puede provocar oscilaciones cuando el robot está cerca de  $(x_g, y_g)$ .

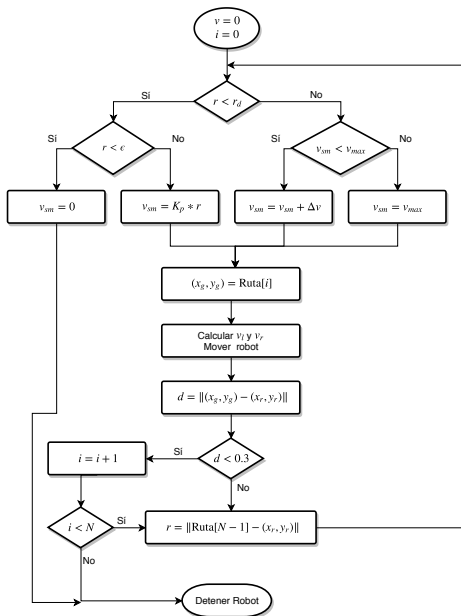
Es deseable tener un perfil de velocidad, y éste se puede lograr mediante una máquina de estados finita aumentada (AFSM por sus siglas en inglés).

Esta AFSM determinará los valores de  $v_{max}$  en las leyes de control (11)-(13).

Sea  $v_{sm}$  la nueva velocidad lineal máxima, de modo que ahora (11)-(13) quedan de la forma:

$$v = v_{sm} e^{-\frac{e_\theta^2}{\alpha}}, \quad (14)$$

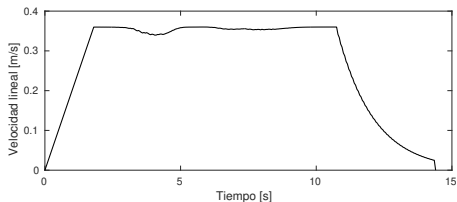
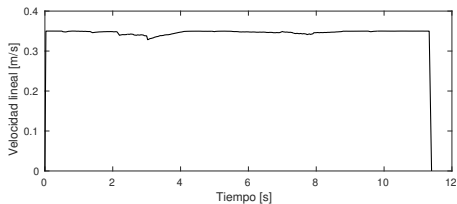
$$\omega = \omega_{max} \left( \frac{2}{1 + e^{-\frac{e_\theta}{\beta}}} - 1 \right), \quad (15)$$



En general, la AFSM realiza los siguientes pasos:

- Primero se incrementa  $v_{sm}$  hasta alcanzar el valor de  $v_{max}$
- Durante esta etapa, la velocidad lineal  $v$  se mantiene casi constante y disminuye sólo cuando el robot va a dar vuelta.
- Cuando el robot está a una distancia  $r$  del punto meta menor que  $r_d$ ,  $v_{sm}$  se asigna proporcional a  $r$ , lo que genera un movimiento similar a un control proporcional.
- Finalmente, cuando la distancia  $r$  es menor que una tolerancia  $\epsilon$ ,  $v_{sm}$  se hace cero.

Las constantes  $r_d$  y  $\Delta v$  son parámetros de diseño y determinan qué tan rápido se desea que el robot acelere o desacelere.



- **Arriba:** Velocidad lineal sin perfil.
- **Centro:** Velocidad lineal empleando la máquina de estados. Al inicio se observa una aceleración y al final, un movimiento similar a un control de posición.
- **Abajo:** Ruta seguida por el robot.

