



Tecnológico de Monterrey

Reporte Act 8.1

Emiliano Olguin Ortega A01737561

22 de Mayo del 2025

Implementación de robótica inteligente (Gpo 501)

Parámetro**Justificación**

`LookaheadDistance` Reduce oscilaciones y mejora la precisión en curvas

`DesiredLinearVelocity` Velocidad lenta para evitar colisiones y permitir reacción

`MaxAngularVelocity` Controla qué tan rápido puede girar el robot

`lidar.scanAngles` Mayor resolución para detectar obstáculos con precisión

`lidar.maxRange` Alcance extendido para anticiparse a paredes o esquinas

Resultados del Punto 2:

- El robot lograba seguir trayectorias simples.
- En algunos casos, atravesaba paredes debido a una mala configuración del VFH.
- Se comprobó que velocidades altas y anticipaciones largas causaban errores en mapas complejos.

Ajustes aplicados para mejorar:

- Aumento del número de sectores en el histograma (`NumAngularSectors`).
- Reducción del `MinTurningRadius` para permitir giros cerrados.
- Uso de `isInside(map, pose)` para prevenir que el robot saliera del mapa.

Punto 3: Implementación Final Óptima

Se diseñaron dos soluciones finales, cada una optimizada para su entorno.

Mapa `exampleMap`

Configuración final:

- `LookaheadDistance = 0.60`: equilibra suavidad con precisión.
- `DesiredLinearVelocity = 0.6`: velocidad adecuada para trayectorias sin mucho estrechamiento.
- `MaxAngularVelocity = 1.5`: evita giros bruscos.
- `lidar.scanAngles = 400` y `maxRange = 2.0`: permite anticipar paredes.
- `MinTurningRadius = 0.8`: apropiado para curvas amplias del mapa.

Resultado:

- Trayectoria cerrada completa sin colisiones.

- Buen desempeño en detección temprana de obstáculos.
- Movimiento estable, fluido y realista.

Mapa complexMap

Configuración final:

- `LookaheadDistance = 0.4`: mayor precisión en pasillos angostos.
- `DesiredLinearVelocity = 0.5`: permite al robot responder con mayor tiempo.
- `MaxAngularVelocity = 20`: habilita giros agresivos en curvas cerradas.
- `NumAngularSectors = 900`: mejora el análisis del entorno para espacios estrechos.
- `SafetyDistance = axleLength = 0.18`: asegura separación segura de las paredes.

Resultado:

- El robot navega correctamente por los pasillos sin atravesar paredes.
- Ajustes finos de `MinTurningRadius` y `MaxAngularVelocity` permitieron giros cerrados.
- Se detiene correctamente al llegar a la meta (`if norm(...) < 0.2`).

Conclusiones Generales

1. El desempeño óptimo depende completamente del entorno: mapas con pasillos estrechos requieren giros más agresivos y mayor resolución de detección.
2. Parámetros como velocidad y anticipación deben adaptarse al mapa para balancear precisión y estabilidad.
3. La combinación de Pure Pursuit y VFH es efectiva, siempre que se afinen cuidadosamente los parámetros.
4. Evitar paredes no se logra solo con VFH: es necesario controlar también la velocidad y dirección de avance para respetar los márgenes del mapa.