

# Tarea 06

## Seguimiento de rutas\*

Robots Móviles (TSM I, TSM II, TSCR), FI, UNAM, 2025-1

Nombre: \_\_\_\_\_

### 1. Actividades

1. Abra el archivo `catkin_ws/src/navigation/simple.move/scripts/path_follower.py` y pegue el siguiente código en la línea 63 para implementar la máquina de estados para el seguimiento de rutas:

```
63 idx = 0
64 Pg = path[idx]
65 Pr, robot_a = get_robot_pose()
66 while numpy.linalg.norm(path[-1] - Pr) > 0.1 and not rospy.is_shutdown():
67     v,w = calculate_control(Pr[0], Pr[1], robot_a, Pg[0], Pg[1], alpha, beta, v_max,
68                             w_max)
69     publish_twist(v,w)
69     Pr, robot_a = get_robot_pose()
70     if numpy.linalg.norm(Pg - Pr) < 0.3:
71         idx = min(idx+1, len(path)-1)
72         Pg = path[idx]
73
```

2. En el mismo archivo, en la función `calculate_control`, implemente las siguientes leyes de control:

$$v = v_{max}e^{-e_a^2/\alpha}$$
$$\omega = \omega_{max} \left( \frac{2}{1 + e^{-e_a/\beta}} - 1 \right)$$

donde

$$e_a = \text{atan2}(y_g - y_r, x_g - x_r) - \theta_r$$

La pose del robot está dada por  $(x_r, y_r, \theta_r)$  y el punto meta por  $(x_g, y_g)$ . Las constantes  $\alpha$  y  $\beta$  son constantes de sintonización y  $v_{max}$ ,  $\omega_{max}$  representan las velocidades máximas que alcanzará el robot. La posición del robot, la posición deseada y las constantes de sintonización se pasan como argumento en la función.

3. Abra una terminal y corra la simulación con el comando:

```
1 roslaunch surge_et_ambula movement_planning.launch
2
```

4. Ejecute el inflado de mapas, el mapa de costo y la planeación de rutas con A\*, con los siguientes comandos, uno en cada terminal:

```
1 rosrn map_augmenter map_inflater.py _inflation_radius:=0.2
2
```

---

\*Material elaborado con apoyo del proyecto PAPIME PE105524

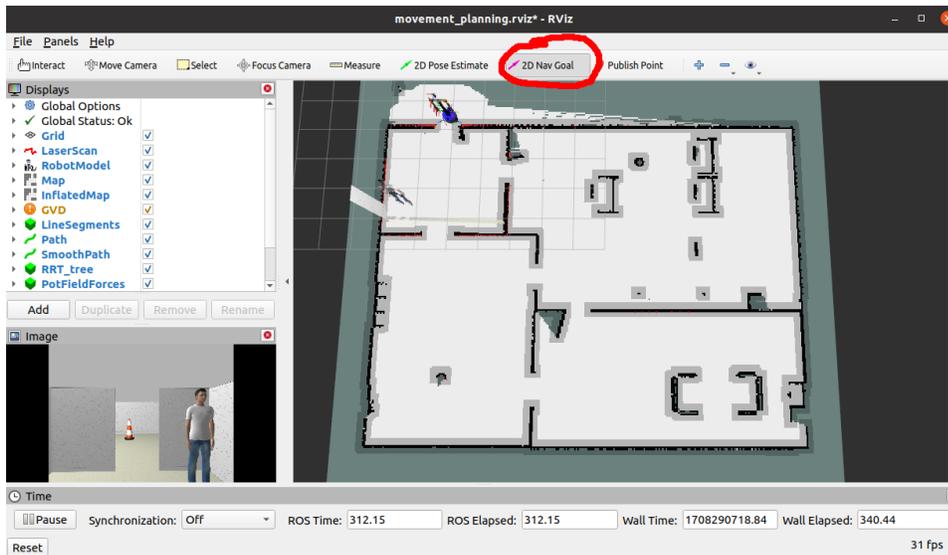
```
1  rosrun path_planner cost_map.py _cost_radius:=0.5
2
```

```
1  rosrun path_planner a_star.py
2
```

5. Ejecute el seguimiento de rutas (archivo modificado en los puntos 1 y 2) con el comando:

```
1  rosrun simple_move path_follower.py _v_max:=0.5 _w_max:=1.0 _alpha:=1.0 _beta:=1.0
2
```

6. Mueva al robot a una posición dentro espacio libre haciendo click en el botón *2D Nav Goal* y luego click en algún punto del mapa.



7. Detenga el seguimiento de rutas y ejecútelos con otras constantes  $\alpha$  y  $\beta$ . Sintonice las constantes hasta obtener un seguimiento satisfactorio.

8. Realice varios experimentos con distintas constantes de sintonización  $\alpha$  y  $\beta$  y registre la ruta deseada (la que es calculada por A\*), la ruta real que siguió el robot y las velocidades lineal y angular durante el movimiento.

9. Grafique la ruta deseada contra la ruta seguida por el robot y las velocidades lineal y angular para cada par de constantes de sintonización.

10. Analice los resultados.

11. Coloque las constantes con mejores resultados como valores por default en el archivo `path_follower.py`

## 2. Entregables

- Código modificado en la rama correspondiente
- Documento impreso con los siguientes puntos:
  - Capturas de pantalla donde se muestre al robot recorriendo la ruta calculada.
  - Gráficas de la ruta deseada contra la ruta real.
  - Gráficas de las velocidades lineal y angular.
  - Discusión de los resultados.

No es necesario pegar el código modificado en el documento escrito. Para eso está el repositorio en línea.

### 3. Evaluación

Para la evaluación se utilizará la siguiente lista de cotejo:

- **[0 puntos]** Los cambios al código se subieron a la rama correspondiente con las credenciales de cada alumno. Este punto es requisito para evaluar los demás.
- **[2 puntos]** El programa para el seguimiento de rutas funciona correctamente (el robot llega al punto meta).
- **[2 puntos]** El documento impreso incluye capturas de pantalla que muestran el correcto funcionamiento del seguimiento de rutas.
- **[2 puntos]** Se incluyen al menos dos gráficas de ruta deseada contra ruta seguida con diferentes constantes de sintonización.
- **[2 puntos]** Se incluyen al menos dos gráficas de las velocidades lineales y angulares, con respecto al tiempo, con diferentes constantes de sintonización.
- **[2 puntos]** Se incluye una discusión sobre la sintonización de las constantes  $\alpha$  y  $\beta$