

0. INTRODUCCIÓN

Los progresos realizados en la parte de localización se han materializado en los siguientes cambios (no hay más):

1. La creación de un nuevo paquete, llamado “alpha_bot_odometry”, con dos nodos, ground_truth_translation_node y alpha_bot_odometry_node.

Se debe tener especial cuidado con los archivos cmake y package.xml de la carpeta alpha_bot_odometry. Se deben añadir las dependencias y librerías adecuadas, hasta llegar a estos ficheros.

2. Además son llamados desde el launch sim.launch (y por tanto también hay cambios en este launch).

1. CAMBIOS RELATIVOS AL PLUGIN DE GROUND TRUTH

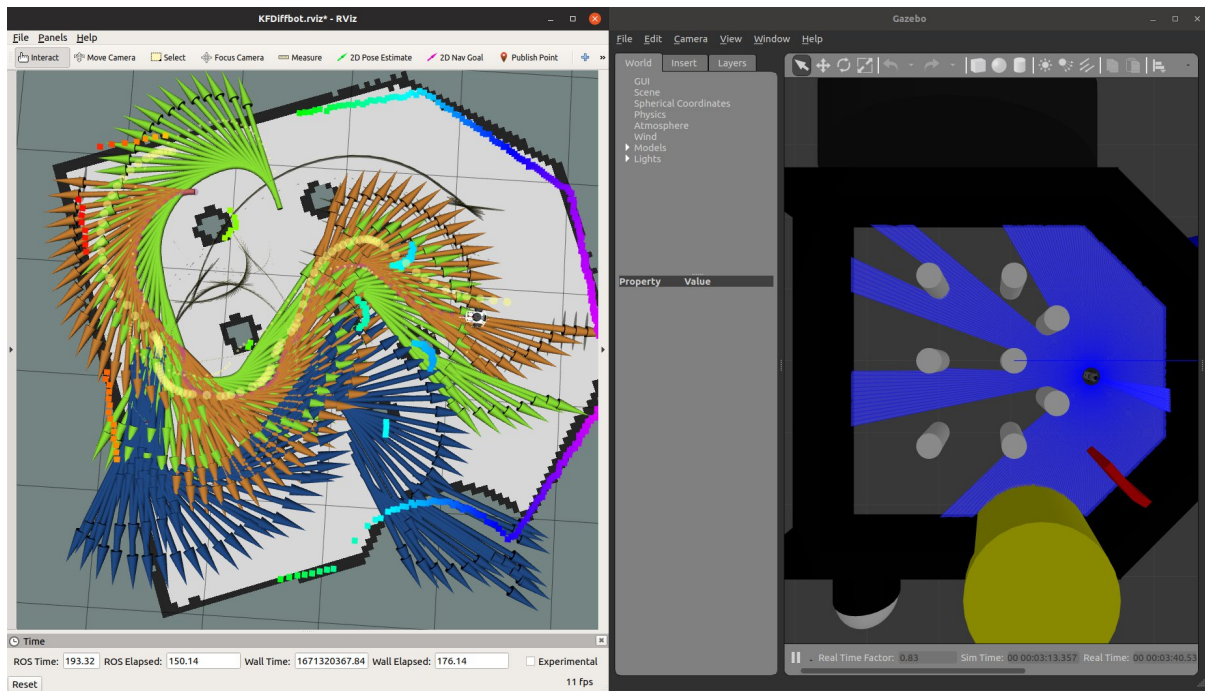
Se ha terminado de arreglar el ground truth. El problema era que el ground_truth no se asociaba al frame odom y por tanto no se publicaba información en su topic (/ground_truth/state). Lo que estaba ocurriendo es que, por algún motivo, gazebo lanza antes el plugin de ground_truth que la construcción del frame odom: para cuando el ground_truth quiere asociarse al frame odom, este frame todavía no existe, gazebo lanza un error, y ese topic no publica nada. En su lugar, se ha asociado al frame /map, origen del mapa generado por gmapping, aunque tendremos que cambiarlo a odom, pues es nuestro fixed frame en la simulación de Rviz.

```
<!-- Added Ground Truth plugin -->
<gazebo>
  <plugin name="p3d_base_controller" filename="libgazebo_ros_p3d.so">
    <alwaysOn>true</alwaysOn>
    <updateRate>50.0</updateRate>
    <bodyName>base_link</bodyName>
    <topicName>ground_truth/state</topicName>
    <gaussianNoise>0.01</gaussianNoise>
    <frameName>map</frameName>
    <xyzOffsets>0 0 0</xyzOffsets>
    <rpyOffsets>0 0 0</rpyOffsets>
  </plugin>
</gazebo>
```

Forma final del plugin ground_truth dentro de alpha_bot_description/urdf/include/robot.gazebo.xacro

Se ha construido un nodo dentro del nuevo paquete alpha_bot_odometry. El nodo alpha_bot_odometry/src/ground_truth_translation.cpp se encarga de tomar el contenido del topic /ground_truth/state, trasladarlo en x,y (en la simulación aparece desplazado sin motivo aparente, porque encima del frame map no está, está mucho más lejos), y asociarlo al frame map.

Con todo, tras sucesivos experimentos, se ha podido comprobar que por algún motivo, los datos arrojados por el plugin dan lugar a un ground truth con una deriva totalmente exagerada. Por otro lado, hemos comprobado que el topic /alpha_bot/mobile_base_controller/odom, ese topic de odometría que en clase vimos que en realidad era ground truth, lo es efectivamente (tras varias pruebas siempre conserva la posición y orientación que el robot tiene realmente en gazebo):



- En marrón: /alpha_bot/mobile_base_controller/odom, el verdadero ground_truth.
- En verde: /ground_truth_final (ya trasladado). Se trata de una prueba intermedia, ambos han partido del mismo inicio. Al comienzo de esta prueba, este “ground truth” ya había sufrido una deriva fuerte, tanto en posición como en orientación, y por tanto naranja y marrón parten de puntos diferentes.
- En azul: odometría por velocidad de las ruedas. Lo vemos en el apartado siguiente.

Por tanto, usaremos el topic /alpha_bot/mobile_base_controller/odom como ground truth para la localización del robot.

2. ODOMETRÍA POR RUEDAS

Se ha construido un nodo dentro del nuevo paquete alpha_bot_odometry. El nodo alpha_bot_odometry/src/ground_truth_translation.cpp emula el comportamiento de unos encoder (no se han encontrado plugins de encoders, al menos no unos que no sean sencillos de desacoplar, puesto que se suele implementar como parte de algoritmos de control, no como sensores persé).

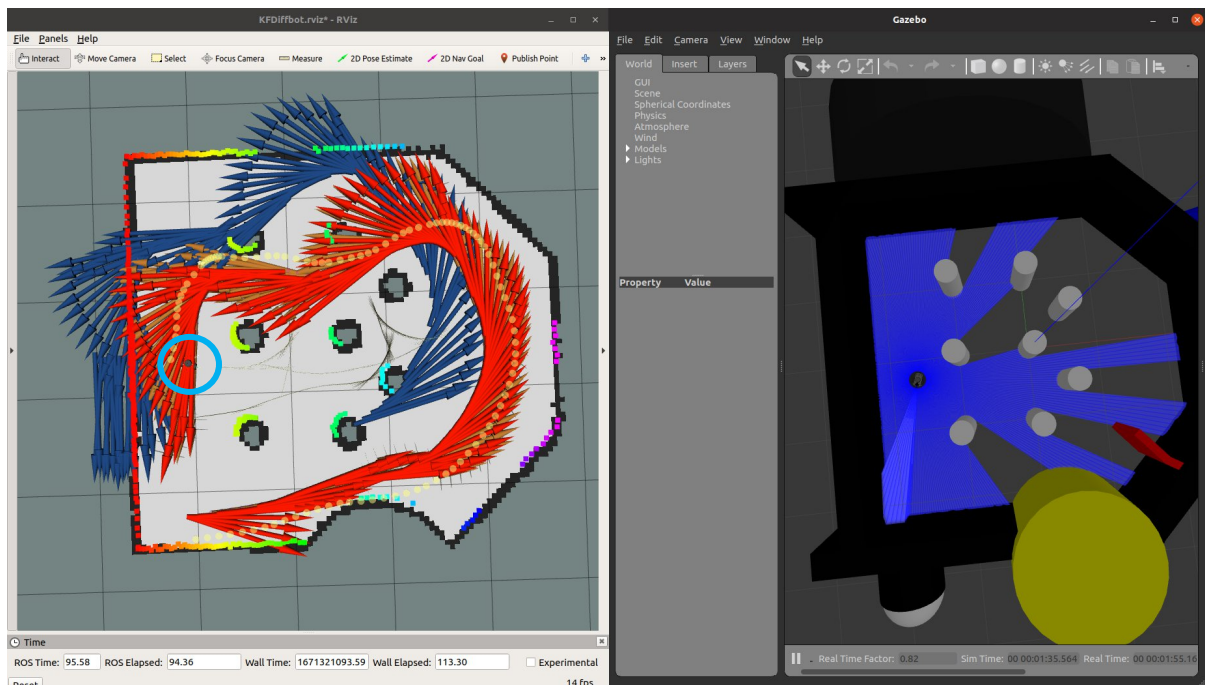
Para emular la lectura de velocidad de los encoder, el nodo se suscribe al ground truth (/alpha_bot_odometry/mobile_base_controller/odom), toma las velocidades en el instante k , y les suma un ruido gaussiano de media 0 y desviación típica 0.2.

Por otro lado, este nodo también calcula la odometría a mano del robot, con las fórmulas vistas en clase. Calcula también unas transformaciones entre frames, requeridas por rviz, pero finalmente no se publican, dado que hacen que el mapa tiemble y se vuelva inestable, puesto que la vista de rviz está centrada sobre el frame odom, y estas transformaciones afectan directamente a dicho frame.

Los resultados de la odometría "real" del robot se publican en el nuevo topic `/wheel_odometry`. Por su parte, la odometría del EKF se ha recableado a dicho topic (antes estaba usando el ground truth, lo cual no tiene mucho sentido pues es lo que queremos estimar).

Sobre las pruebas realizadas, cabe destacar que, aunque el filtro de Kalman es óptimo eliminando el ruido gaussiano, como el que hemos introducido, NO LO ELIMINA POR COMPLETO, SER ÓPTIMO QUIERE DECIR QUE ES EL QUE MÁS LO REDUCE, PERO NO LO HACE 0.

Lo cierto es que, si a un filtro le metemos "basura", sacaremos "basura". Por tanto, no tiene mucho sentido añadir una desviación típica mayor a la comentada anteriormente. Aún así, tal y como podemos comprobar en las imágenes inferiores, el EKF consigue aproximar bastante mejor el ground truth que la odometría (a falta de cuantificar dicha mejora de manera formal con varias simulaciones).



- En marrón: `/alpha_bot/mobile_base_controller/odom`, el verdadero ground_truth.
- En rojo: `/odometry/filtered` (EKF). Se trata de una prueba intermedia, ambos han partido del mismo inicio. Al comienzo de esta prueba, este EKF ya había sufrido una cierta deriva, tanto en posición como en orientación, y aunque parten de puntos algo diferentes, suelen ir casi a la par.
- En azul: odometría por velocidad de las ruedas (`/wheel_odometry`). Mientras en esta prueba intermedia EKF y ground truth han ido muy próximos, él ha comenzado la prueba en otra posición.

Por último, en los .cpp se dejan los enlaces consultados como bibliografía.