

Previo de la Práctica 2

- **Modelo de un robot móvil diferencial (2,0).**

Un robot móvil diferencial (2,0) es una plataforma con dos ruedas motrices colineales y, típicamente, una rueda loca de apoyo; se considera no holonómico porque no puede desplazarse lateralmente, solo avanzar, retroceder y girar sobre su eje. Su estado de pose en el plano se describe por las coordenadas (x, y, θ) , donde x y y son la posición del punto medio entre las ruedas y θ es el ángulo de orientación respecto al eje X inercial. La cinemática del modelo (2,0) relaciona las velocidades lineal v y angular ω del robot con la evolución de la pose mediante el sistema: $\dot{x} = v \cos \theta$, $\dot{y} = v \sin \theta$, $\dot{\theta} = \omega$. Las velocidades de cada rueda, ω_r y ω_l , se vinculan con v y ω considerando el radio de rueda R y la mitad de la distancia entre ruedas L : $v = \frac{R}{2}(\omega_r + \omega_l)$ y $\omega = \frac{R}{2L}(\omega_r - \omega_l)$.

- **Principio de funcionamiento de un sensor LiDAR.**

LiDAR significa “Light Detection and Ranging” y es una tecnología de medición de distancias basada en pulsos de luz láser. El sensor emite un pulso láser hacia un objeto, detecta el eco reflejado y calcula la distancia midiendo el tiempo de vuelo del pulso y multiplicándolo por la velocidad de la luz, dividiendo entre dos para considerar el recorrido de ida y vuelta.

Al disparar muchos pulsos por segundo y combinar las distancias con la orientación del haz, el LiDAR genera una nube de puntos que representa la geometría 3D del entorno. Normalmente integra un láser, un sistema de escaneo y un detector, junto con sistemas de posicionamiento como GPS e INS para referenciar los puntos en un marco de coordenadas global.

- **Principio de funcionamiento de la navegación SLAM.**

SLAM (Simultaneous Localization and Mapping) es una técnica que permite a un robot estimar de forma simultánea su propia posición y construir un mapa del entorno sin disponer de un mapa previo. El algoritmo combina datos de sensores (LiDAR, cámaras, IMU, etc.) con modelos de movimiento del robot para actualizar recursivamente el estado del sistema, que incluye la pose del robot y los parámetros del mapa.

Para la estimación recursiva se emplean métodos probabilísticos como el filtro de Kalman extendido o filtros de partículas, que corrigen la predicción de la pose con base en nuevas observaciones del entorno. A medida que el robot se desplaza, el mapa se refina y la trayectoria se reajusta, lo que le permite planificar rutas, evitar obstáculos y navegar en entornos dinámicos.

- **Implementación del mapeo de un entorno utilizando RViz.**

En un sistema de navegación con ROS, el mapeo suele realizarse mediante paquetes como `gmapping`, `hector_slam` o `slam_toolbox`, que reciben datos del LiDAR y la odometría para producir un mapa de ocupación en 2D y publicarlo en un tema de ROS. Otros nodos consumen este mapa y la pose estimada del robot para tareas de navegación.

RViz se utiliza para visualizar en tiempo real el mapa que va construyendo el algoritmo SLAM, la trayectoria del robot y las lecturas de sensores como el LiDAR. El usuario puede configurar distintos *displays* (mapa, láser, TF, trayectoria, objetivos de navegación) y fijar metas en el mapa, que luego son enviadas al planificador para que el robot navegue hasta dichas posiciones.

- **¿Qué es RViz?**

RViz (ROS Visualization) es una herramienta de visualización 3D incluida en ROS que permite representar el modelo del robot, su entorno y los datos de sus sensores en un entorno gráfico interactivo. Carga la descripción del robot (por ejemplo en formato URDF) desde el servidor de parámetros de ROS y escucha los temas de sensores, transformaciones de referencia y mapas para mostrar su estado de forma intuitiva.

Entre sus funciones principales están la visualización de nubes de puntos LiDAR, imágenes de cámaras, mapas de ocupación y marcos de coordenadas, así como la posibilidad de enviar objetivos de navegación y depurar la correcta publicación de tópicos y transformaciones. Por su flexibilidad y sistema de plugins, RViz se ha convertido en una herramienta estándar para depuración, demostraciones y desarrollo de aplicaciones de robótica móvil en ROS.

1 Desarrollo

En esta práctica se amplió el modelo URDF de un robot tipo SCARA para incorporar un cuarto grado de libertad de tipo prismático, representando un actuador lineal que permite el movimiento vertical del efecto final. Para ello

se aprovechó la estructura ya existente del robot, compuesta por una base fija y tres eslabones unidos mediante articulaciones revolutas, sobre la cual se añadió un nuevo enlace y una junta prismática.

El primer paso consistió en definir el nuevo enlace `link_4`, que modela el elemento móvil que se desplaza linealmente, asignándole su geometría de visualización, colisión e inercia mediante primitivas de caja para mantener un modelo ligero y adecuado para la simulación. Posteriormente se creó la articulación `link_4.joint` de tipo `prismatic`, especificando correctamente el enlace padre (`link_1`), el enlace hijo (`link_4`), el eje de traslación y los límites de recorrido en metros, de forma que el desplazamiento quedara restringido a un intervalo físicamente razonable.

La elección del eje de movimiento se realizó utilizando el vector `axis xyz="0 0 1"`, lo que sitúa la traslación a lo largo del eje `z` del marco de la junta, coherente con un movimiento de subida y bajada típico de un manipulador SCARA para tareas de *pick and place*. Además, se ajustó el origen de la junta mediante la etiqueta `<origin>` para posicionar correctamente el actuador respecto al extremo del primer eslabón, garantizando la continuidad geométrica del modelo y evitando solapes indeseados en la simulación.

Con estos cambios, el modelo URDF actualizado mantiene la jerarquía original de enlaces y continúa siendo compatible con herramientas de ROS como RViz y Gazebo, que pueden ahora simular tanto las rotaciones de las juntas revolutas como el desplazamiento lineal de la junta prismática. Este enfoque demuestra cómo la extensión modular de un URDF permite incrementar el número de grados de libertad del robot sin alterar la filosofía de descripción basada en árboles de enlaces y articulaciones.

2 Código URDF con junta prismática

A continuación se muestra el archivo URDF/Xacro completo del robot SCARA con la junta prismática incluida:

```
<?xml version="1.0" encoding="utf-8"?>
<robot name="robot_scara" xmlns:xacro="http://ros.org/wiki/xacro">

<link name="world"/>

<joint name="world_joint" type="fixed">
  <parent link="world"/>
  <child link="base_link"/>
  <origin xyz="0 0 0" rpy="0 0 0"/>
</joint>

<link name="base_link">
  <visual>
    <origin xyz="0.0 0.0 0.1" rpy="0 0 0"/>
    <geometry>
      <box size="0.1 0.15 0.2"/>
    </geometry>
    <material name="grey">
      <color rgba="0.1 0.1 0.1 1.0"/>
    </material>
  </visual>
  <collision>
    <origin xyz="0.0 0.0 0.1" rpy="0 0 0"/>
    <geometry>
      <box size="0.1 0.15 0.2"/>
    </geometry>
  </collision>
  <inertial>
    <origin xyz="0 0 0" rpy="0 0 0"/>
    <mass value="1"/>
    <inertia ixx="1.0" ixy="0.0" ixz="0.0"
             iyy="1.0" iyz="0.0" izz="1.0"/>
  </inertial>
```

```

</link>

<link name="link_1">
  <visual>
    <origin xyz="0.25 0.0 0.0" rpy="0 0 0"/>
    <geometry>
      <box size="0.5 0.05 0.05"/>
    </geometry>
    <material name="red">
      <color rgba="1.0 0.0 0.0 1.0"/>
    </material>
  </visual>
  <collision>
    <origin xyz="0.25 0.0 0.0" rpy="0 0 0"/>
    <geometry>
      <box size="0.5 0.05 0.05"/>
    </geometry>
  </collision>
  <inertial>
    <origin xyz="0 0 0" rpy="0 0 0"/>
    <mass value="1"/>
    <inertia ixx="1.0" ixy="0.0" ixz="0.0"
              iyy="1.0" iyz="0.0" izz="1.0"/>
  </inertial>
</link>

<joint name="link_1_joint" type="revolute">
  <parent link="base_link"/>
  <child link="link_1"/>
  <axis xyz="0 0 1"/>
  <origin xyz="0 0 0.225" rpy="0 0 0"/>
  <limit lower="-3.14159" upper="3.14159"
        velocity="50.0" effort="1000.0"/>
</joint>

<link name="link_2">
  <visual>
    <origin xyz="0.25 0.0 0.0" rpy="0 0 0"/>
    <geometry>
      <box size="0.5 0.05 0.05"/>
    </geometry>
    <material name="green">
      <color rgba="0.0 1.0 0.0 1.0"/>
    </material>
  </visual>
  <collision>
    <origin xyz="0.25 0.0 0.0" rpy="0 0 0"/>
    <geometry>
      <box size="0.5 0.05 0.05"/>
    </geometry>
  </collision>
  <inertial>
    <origin xyz="0 0 0" rpy="0 0 0"/>
    <mass value="1"/>
    <inertia ixx="1.0" ixy="0.0" ixz="0.0"
              iyy="1.0" iyz="0.0" izz="1.0"/>
  </inertial>
</link>
```

```

        </inertial>
    </link>

<joint name="link_2_joint" type="revolute">
    <parent link="link_1"/>
    <child link="link_2"/>
    <axis xyz="0 0 1"/>
    <origin xyz="0.45 0 0.05" rpy="0 0 0"/>
    <limit lower="-3.14159" upper="3.14159"
          velocity="50.0" effort="1000.0"/>
</joint>

<link name="link_3">
    <visual>
        <origin xyz="0.15 0.0 0.0" rpy="0 0 0"/>
        <geometry>
            <box size="0.3 0.05 0.05"/>
        </geometry>
        <material name="blue">
            <color rgba="0.0 0.0 1.0 1.0"/>
        </material>
    </visual>
    <collision>
        <origin xyz="0.15 0.0 0.0" rpy="0 0 0"/>
        <geometry>
            <box size="0.3 0.05 0.05"/>
        </geometry>
    </collision>
    <inertial>
        <origin xyz="0 0 0" rpy="0 0 0"/>
        <mass value="1"/>
        <inertia ixx="1.0" ixy="0.0" ixz="0.0"
                  iyy="1.0" iyz="0.0" izz="1.0"/>
    </inertial>
</link>

<joint name="link_3_joint" type="revolute">
    <parent link="link_2"/>
    <child link="link_3"/>
    <axis xyz="0 0 1"/>
    <origin xyz="0.45 0 -0.05" rpy="0 0 0"/>
    <limit lower="-3.14159" upper="3.14159"
          velocity="50.0" effort="1000.0"/>
</joint>

<link name="link_4">
    <visual>
        <origin xyz="0.0 0.0 0.0" rpy="0 0 0"/>
        <geometry>
            <box size="0.1 0.05 0.2"/>
        </geometry>
        <material name="yellow">
            <color rgba="1.0 1.0 0.0 1.0"/>
        </material>
    </visual>
    <collision>

```

```

<origin xyz="0.0 0.0 0.0" rpy="0 0 0"/>
<geometry>
  <box size="0.1 0.05 0.2"/>
</geometry>
</collision>
<inertial>
  <origin xyz="0 0 0" rpy="0 0 0"/>
  <mass value="1"/>
  <inertia ixx="1.0" ixy="0.0" ixz="0.0"
            iyy="1.0" iyz="0.0" izz="1.0"/>
</inertial>
</link>

<joint name="link_4_joint" type="prismatic">
  <parent link="link_1"/>
  <child link="link_4"/>
  <origin xyz="0.5 0 0" rpy="0 0 0"/>
  <axis xyz="0 0 1"/>
  <limit lower="0.0" upper="0.1"
        effort="100.0" velocity="0.2"/>
</joint>

<link name="P"/>

<joint name="link_P_joint" type="fixed">
  <parent link="link_3"/>
  <child link="P"/>
  <origin xyz="0.25 0 0" rpy="0 0 0"/>
</joint>

<link name="lidar_link">
  <visual>
    <origin xyz="0 0 0.15" rpy="0 0 0"/>
    <geometry>
      <cylinder radius="0.03" length="0.05"/>
    </geometry>
    <material name="lidar_black">
      <color rgba="0.0 0.0 0.0 1.0"/>
    </material>
  </visual>
  <collision>
    <origin xyz="0 0 0.15" rpy="0 0 0"/>
    <geometry>
      <cylinder radius="0.03" length="0.05"/>
    </geometry>
  </collision>
  <inertial>
    <origin xyz="0 0 0" rpy="0 0 0"/>
    <mass value="0.2"/>
    <inertia ixx="1e-3" ixy="0.0" ixz="0.0"
              iyy="1e-3" iyz="0.0" izz="1e-3"/>
  </inertial>
</link>

<joint name="lidar_joint" type="fixed">
  <parent link="base_link"/>

```

```

<child link="lidar_link"/>
<origin xyz="0 0 0.25" rpy="0 0 0"/>
</joint>

<gazebo reference="lidar_link">
  <sensor name="lidar" type="ray">
    <always_on>true</always_on>
    <update_rate>10</update_rate>
    <visualize>true</visualize>
    <pose>0 0 0 0 0 0</pose>
    <ray>
      <scan>
        <horizontal>
          <samples>360</samples>
          <resolution>1</resolution>
          <min_angle>-3.14159</min_angle>
          <max_angle>3.14159</max_angle>
        </horizontal>
      </scan>
      <range>
        <min>0.1</min>
        <max>5.0</max>
        <resolution>0.01</resolution>
      </range>
    </ray>
    <topic>/lidar</topic>
  </sensor>
</gazebo>

</robot>

```

3 Conclusiones

La incorporación de la junta prismática permitió extender el modelo SCARA inicial añadiendo un grado de libertad lineal que representa de forma más realista el comportamiento del eje vertical de muchos manipuladores industriales. Esta modificación mostró que el formato URDF facilita la ampliación progresiva de la cinemática del robot simplemente añadiendo nuevos enlaces y articulaciones bien definidos.

Durante el desarrollo se comprobó la importancia de elegir correctamente el tipo de junta, su eje de movimiento y los límites de recorrido, ya que estos parámetros impactan directamente en el espacio de trabajo y en la estabilidad de la simulación. Asimismo, el uso de primitivas geométricas sencillas en las secciones de visualización y colisión ayudó a mantener un modelo computacionalmente eficiente, adecuado para herramientas como RViz y Gazebo.