10

Robotics Corner

ROS for Professionals

URDF in ROS

Outline

01 Overview

02 What is URDF

03 Writing a simple URDF file

04 Building your robot urdf

05 Transformation TF

06 Simulation

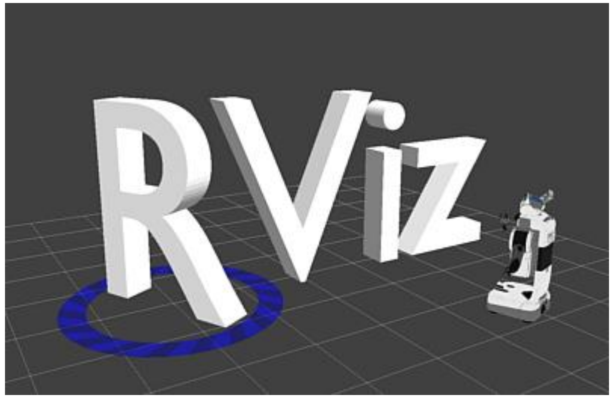
ROBOTICS CORNER

01 Overview

Definition

A **simulation environment** is a virtual space where robotic models interact with simulated physics, sensors, and environments to test control algorithms, perception systems, and navigation strategies.

A **visualization environment** is used to represent robot states, sensor data, and computed results graphically. It helps engineers analyze simulations, real-world sensor data, and decision-making processes.



blender

unity

GAZEBO

01 Overview

Definition

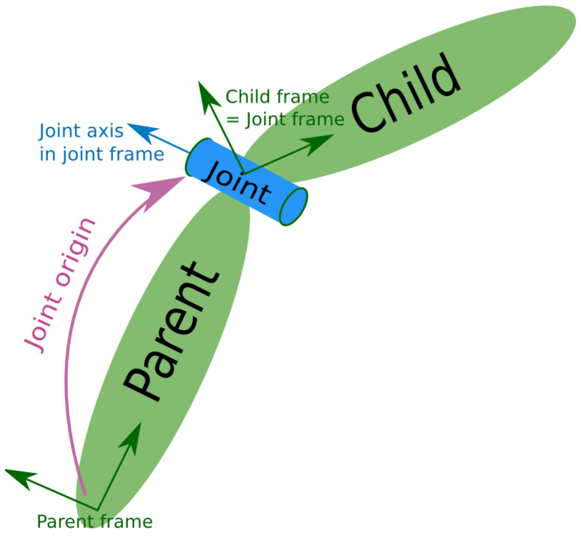
- SDF (**Simulation Description Format**) is an XML-based format used to describe objects, environments, and simulations in robotics, particularly in Gazebo, a popular robotics simulation tool.
- Model Description – Defines links, joints, sensors, and plugins for robots.
- Physics Properties – Includes mass, inertia, collision properties, and friction coefficients.
- World Description – Allows the creation of simulated environments with lighting, terrain, and objects.
- Plugins – Extends functionality with custom behaviors written in C++ or Python.

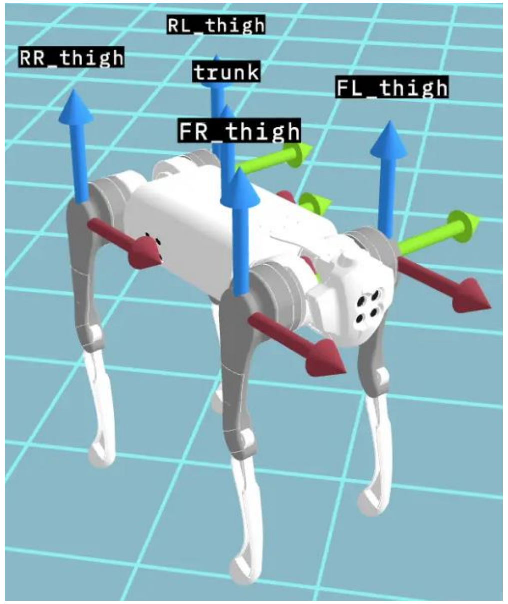
What is URDF

Definition

URDF, **Unified Robot Description Format** is an XML format for representing a robot model. URDF is commonly used in Robot Operating System tools such as rviz and Gazebo simulator. The model consists of links and joints motion.

- To define a robot we have to describe all the parts(aka **links**) and connections between them (aka **joints**).



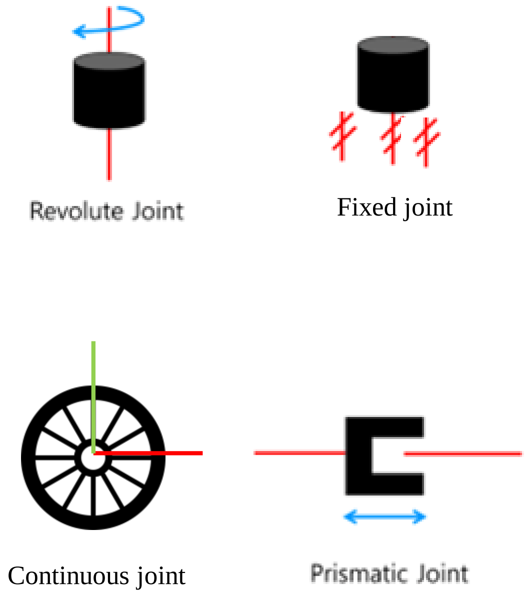


URDF File Components

- Links
  - Visual
    - Geometry
    - Origin
    - Material
  - Collision
  - Inertial
- Joins
  - Parent
  - Child
  - Origin

Common Joint Types in URDF

- Fixed: rigid connection
- Revolute: 1D rotation (Constrained Rotation Movement)
  - Rotational Limit.
  - Axis
- Continuous: Unlimited Rotation Movement. (From –inf to inf)
  - Axis
- Prismatic: 1D Translation Movement
  - Translational Limit.
  - Axis
- Planar: 2D translation
- Floating: unlimited 6D
- Note: we mostly use the first 4 types



Adding Physical and Collision Properties:

- In many cases, you’ll want the collision geometry and origin to be exactly the same as the visual geometry and origin. However, there are two main cases where you wouldn’t:
  - **Quicker Processing**. Doing collision detection for two meshes is a lot more computational complex than for two simple geometries. Hence, you may want to replace the meshes with simpler geometries in the collision element.
  - **Safe Zones**. You may want to restrict movement close to sensitive equipment. For instance, if we didn’t want anything to collide with R2D2’s head, we might define the collision geometry to be a cylinder encasing his head to prevent anything from getting too close to his head.

Adding Physical and Collision Properties:

- In order to get your model to simulate properly, you need to define several physical properties of your robot, i.e. the properties that a physics engine like Gazebo would need.
- This element is also a sub-element of the link object.
- The **mass** is defined in kilograms.
- The 3x3 rotational **inertia** matrix is specified with the inertia element. Since this is symmetrical, it can be represented by only 6 elements.

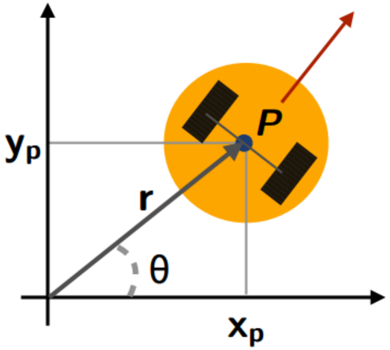
Adding Physical and Collision Properties:

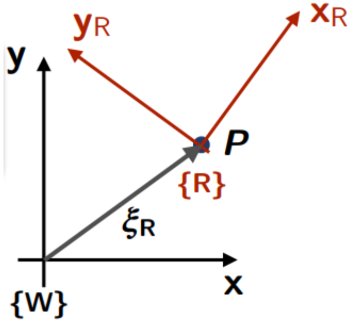
- **Contact Coefficients**: You can also define how the links behave when they are in contact with one another. This is done with a sub-element of the collision tag called contact_coefficients. There are three attributes to specify:
  - mu - Friction coefficient
  - kp - Stiffness coefficient
  - kd - Dampening coefficient

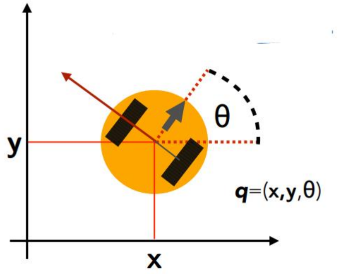
- **Joint Dynamics**: How the joint moves is defined by the dynamics tag for the joint. There are two attributes here:
  - friction - The physical static friction. For prismatic joints, the units are Newtons. For revolving joints, the units are Newton meters.
  - damping - The physical damping value. For prismatic joints, the units are Newton seconds per meter. For revolving joints, Newton meter seconds per radian.

TF:

- A robot is embedded in a physical environment, its actions and their effects strictly depend on where the robot is precisely located in the environment in terms of **position + orientation.**
- **Rigid body assumption:** The object is rigid, such that its constituent points maintain a constant relative position with respect to each other and to the object’s coordinate frame.
- A robot of any single-body shape can be “reduced” to a point, selected as reference point.
- A multi-body robot, with n moving parts (not rigidly connected) can be “reduced” to n points, selected as reference points.
- Define a fixed world reference coordinate frame {W}.
- Center a (local) coordinate frame {R} in the robot’s reference point P, (possibly) oriented according to robot’s natural orientation.
- **The pose/configuration of the object/robot in {W} is described by the position and orientation of the (local) coordinate frame {R} wrt {W}**
- The configuration of a non-omnidirectional mobile robot in a two-dimensional coordinate systems is fully defined by:
  - position (x,y) 2.
  - orientation angle θ, with respect to the coordinate axes

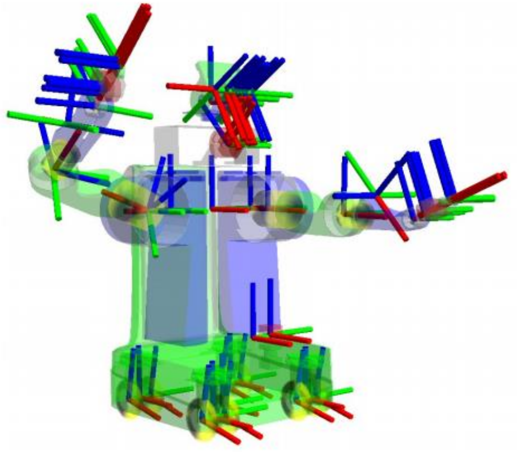


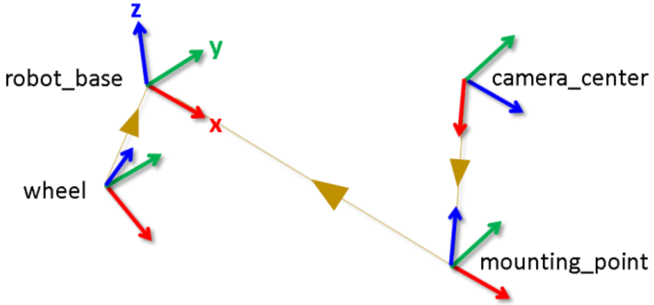


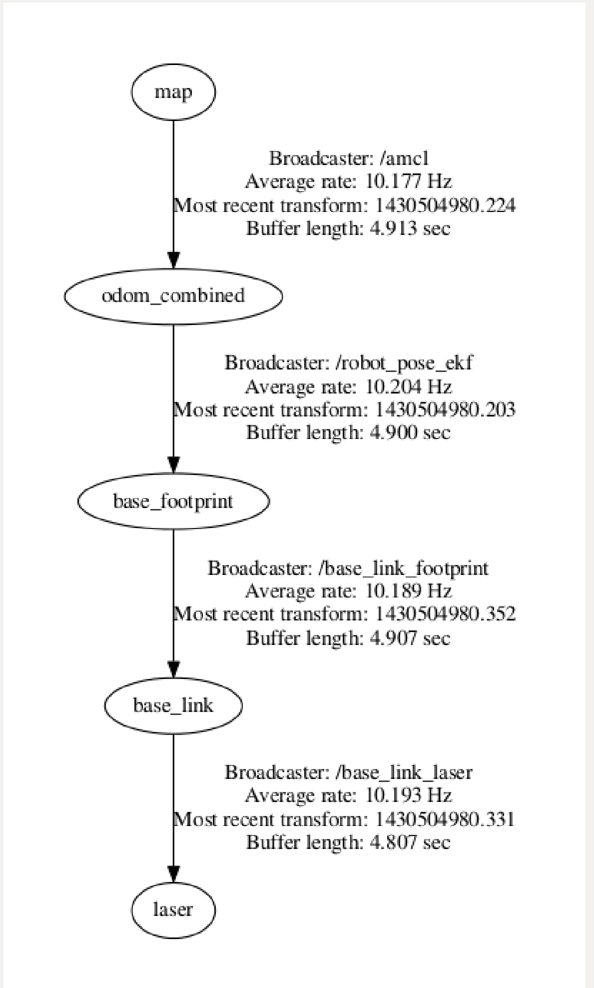


TF:

- Tool for keeping track of coordinate frames over time
- Maintains relationship between coordinate frames in a tree structure buffered in time
- Lets the user transform points, vectors, etc. between coordinate frames at desired time.
- Implemented as publisher/subscriber model on the topics /tf and /tf_static







Resources for learning

- https://docs.ros.org/en/galactic/Tutorials/Intermediate/URDF/URDF-Main.html
- https://docs.ros.org/en/foxy/Tutorials/Intermediate/URDF/Adding-Physical-and-Collision-Properties-to-a-URDF-Model.html
- https://automaticaddison.com/how-to-load-a-urdf-file-into-gazebo-ros-2/

What to expect?

- Understanding Sensors.
- Getting everything together (Connecting Gazebo, worlds, URDF)
- Understanding ROS controllers

Thank You

Do you have any questions?

ROBOTICS CORNER

ROBOTICS CORNER

01211626904

www.roboticscorner.tech

Robotics Corner

Robotics Corner

Robotics Corner

Robotics Corner