ROS2 Robot Control

Generated README

1 ROS2 and VSCode Extensions

Install the following extensions in VSCode:

- CMakeTools
- XML
- ROS

2 ROS2 Packages

The following ROS2 packages are required:

- ros-humble-ros2-controllers
- ros-humble-xacro
- ros-humble-gazebo-ros
- ros-humble-gazebo-ros-pkgs
- ros-humble-ros2-control
- ros-humble-gazebo-ros2-control
- ros-humble-joint-state-publisher-gui
- ros-humble-turtlesim
- ros-humble-robot-localization
- ros-humble-joy-teleop
- ros-humble-tf-transformations
- ros-humble-plotjuggler
- ros-humble-plotjuggler-ros

Linear and Angular Velocity

Linear Velocity

The total linear velocity is given by:

$$\text{total\_linear\_velo} = \frac{v_{\text{wheel}_1} + v_{\text{wheel}_2}}{2}$$

where the wheel velocities are calculated as:

$$v_{\text{wheel}_1} = \frac{w_{\text{right}} \cdot \dot{\phi}_{\text{right}} + w_{\text{left}} \cdot \dot{\phi}_{\text{left}}}{2}$$

Angular Velocity

The total angular velocity is given by:

$$\text{total\_angular\_velo} = \frac{w_{\text{left}} \cdot \dot{\phi}_{\text{left}}}{w_s} - \frac{w_{\text{right}} \cdot \dot{\phi}_{\text{right}}}{w_s}$$

where:

- w_{left} and w_{right} are the wheel speeds.
- $\dot{\phi}_{\text{left}}$ and $\dot{\phi}_{\text{right}}$ are the wheel angular velocities.
- w_s is a scaling factor or wheelbase.

3 Mobile Robots

There are 3 types of locomotion architectures for mobile robots:

- Differential Drive
- Ackerman Drive
- Omnidirectional Drive

For Differential Drive, the state space vector is: $[x, y, \theta]$.

4 Kinematics and Velocities

The velocity and kinematic equations are provided as follows:

$$v_{\text{total}} = \frac{v_{\text{wheel1}} + v_{\text{wheel2}}}{2}$$

4.1 Total velocity

$$\begin{bmatrix} v \\ w \end{bmatrix} = \begin{bmatrix} \frac{wheel_{\text{radius}}}{2} & \frac{wheel_{\text{radius}}}{2} \\ \frac{wheel_{\text{radius}}}{wheel_{\text{separation}}} & -\frac{wheel_{\text{radius}}}{wheel_{\text{separation}}} \end{bmatrix} \begin{bmatrix} \dot{\phi}_{\text{right}} \\ \dot{\phi}_{\text{left}} \end{bmatrix}$$

4.2 Velocity in the World Frame

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ 0 \\ w \end{bmatrix}$$

4.3 Forward Kinematics

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \frac{w_{\text{radius}} \cos(\theta)}{2} & \frac{w_{\text{radius}} \cos(\theta)}{2} \\ \frac{w_{\text{radius}} \sin(\theta)}{2} & \frac{w_{\text{radius}} \sin(\theta)}{2} \\ \frac{w_{\text{radius}}}{w_{\text{separation}}} & -\frac{w_{\text{radius}}}{w_{\text{separation}}} \end{bmatrix} \begin{bmatrix} \dot{\phi}_{\text{right}} \\ \dot{\phi}_{\text{left}} \end{bmatrix}$$

5 Angle Representations

There are two ways to represent angles: Euler Angles and Quaternions.

5.1 Euler Angles

Rotation matrices for the Z, Y, and X axes (Yaw, Pitch, Roll) are provided as:

$$\text{Yaw (Z):} \quad \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\text{Pitch (Y):} \quad \begin{bmatrix} \cos(\phi) & 0 & -\sin(\phi) \\ 0 & 1 & 0 \\ \sin(\phi) & 0 & \cos(\phi) \end{bmatrix}$$

$$\text{Roll (X):} \quad \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\psi) & -\sin(\psi) \\ 0 & \sin(\psi) & \cos(\psi) \end{bmatrix}$$

5.2 Quaternions

Quaternions are defined as $q = a + bi + cj + dk$ with the following properties:

- Unitary: $a^2 + b^2 + c^2 + d^2 = 1$
- Rotation Composition: $q_1 * q_2$
- Inverse: $q^{-1} = a - bi - cj - dk$

6 Localization

Localization methods include Wheel Odometry, Laser Odometry, and Visual Odometry.

6.1 Wheel Odometry

- Encoder sensor is use to get the number of the rotation
- Light reciever provides a binary information 1 or 0, depending on whether it recieves the light beam from the light source or not.
- More advanced encoders: absolute encoders

6.2 Differential Inverse Kinematics

$$\begin{bmatrix} v \\ w \end{bmatrix} = \begin{bmatrix} \frac{\text{wheel\_radius}}{2} & \frac{\text{wheel\_radius}}{2} \\ \frac{\text{wheel\_radius}}{\text{wheel\_separation}} & -\frac{\text{wheel\_radius}}{\text{wheel\_separation}} \end{bmatrix} \begin{bmatrix} \dot{\phi}_{\text{right}} \\ \dot{\phi}_{\text{left}} \end{bmatrix}$$

- $v = s \cdot t$
- $\dot{\phi}_{\text{right}} = \frac{\phi_{1_{\text{right}}} - \phi_{0_{\text{right}}}}{t_1 - t_0} = \frac{\Delta\phi_{\text{right}}}{\Delta t}$
- $\dot{\phi}_{\text{left}} = \frac{\phi_{1_{\text{left}}} - \phi_{0_{\text{left}}}}{t_1 - t_0} = \frac{\Delta\phi_{\text{left}}}{\Delta t}$
- $v = \frac{\text{wheel\_radius}}{2} \cdot \frac{\Delta\phi_{\text{right}}}{\Delta t} + \frac{\text{wheel\_radius}}{2} \cdot \frac{\Delta\phi_{\text{left}}}{\Delta t}$
- $w = \frac{\text{wheel\_radius}}{\text{wheel\_separation}} \cdot \frac{\Delta\phi_{\text{right}}}{\Delta t} - \frac{\text{wheel\_radius}}{\text{wheel\_separation}} \cdot \frac{\Delta\phi_{\text{left}}}{\Delta t}$

6.3 Wheel Odometry

$$\begin{aligned} \text{position} = \int v\,dt &= \int \left(\frac{\text{wheel\_radius} \cdot \dot{\phi}_{\text{right}}}{2} + \frac{\text{wheel\_radius} \cdot \dot{\phi}_{\text{left}}}{2} \right) dt \\ &= \frac{\text{wheel\_radius} \cdot \Delta\phi_{\text{right}}}{2} + \frac{\text{wheel\_radius} \cdot \Delta\phi_{\text{left}}}{2} \end{aligned}$$

$$\begin{aligned} \text{orientation} = \int w\,dt &= \int \left(\frac{\text{wheel\_radius} \cdot \dot{\phi}_{\text{right}}}{\text{wheel\_separation}} - \frac{\text{wheel\_radius} \cdot \dot{\phi}_{\text{left}}}{\text{wheel\_separation}} \right) dt \\ &= \frac{\text{wheel\_radius} \cdot \Delta\phi_{\text{right}}}{\text{wheel\_separation}} - \frac{\text{wheel\_radius} \cdot \Delta\phi_{\text{left}}}{\text{wheel\_separation}} \end{aligned}$$

7 Probability for Robotics

7.1 Bayes Rule

1. The intersection of two events A and B:

$$P(A \cap B) = P(A \mid B) \cdot P(B) = P(B \mid A) \cdot P(A)$$

2. The probability of the intersection of A and B is equal to the probability of the intersection of B and A:

$$P(A \cap B) = P(B \cap A)$$

3. Bayes' Rule:

$$P(A \mid B) = \frac{P(B \mid A) \cdot P(A)}{P(B)}$$

where:

- $P(A \mid B)$: Posterior probability, represents the updated probability of the event A given that we observed the occurrence of event B.
- $P(A)$: Prior probability, represents the initial probability we estimated for the event A.
- $P(B)$: Marginal probability, represents the overall probability of observing the event B.
- $P(B \mid A)$: Likelihood, indicates the probability of observing the event B assuming that the event A occurred.

8 Kalman Filter

Kalman Filter is used for sensor fusion in robotics.

Update

Given:

- **Odometry:** μ_1, σ_1^2
- **IMU:** μ_2, σ_2^2

The updated mean μ_3 and variance σ_3^2 are calculated as follows:

$$\mu_3 = \frac{\sigma_2^2 \mu_1 + \sigma_1^2 \mu_2}{\sigma_2^2 + \sigma_1^2}$$

$$\sigma_3^2 = \frac{1}{\frac{1}{\sigma_2^2} + \frac{1}{\sigma_1^2}}$$

Prediction

Given:

- **Prior Belief:** μ_1, σ_1^2
- **Motion:** μ_2, σ_2^2

The predicted mean μ_3 and variance σ_3^2 are calculated as follows:

$$\mu_3 = \mu_1 + \mu_2$$

$$\sigma_3^2 = \sigma_1^2 + \sigma_2^2$$