Formation control in ROS2

Prof. Ivano Notarnicola

Department of Electrical, Electronic, and Information Engineering
Alma Mater Studiorum Università di Bologna

ivano.notarnicola@unibo.it

Distributed Autonomous Systems M
A.A. 2024-2025

Workspace preparation

Activate ROS2

. /opt/ros/humble/setup.bash

Create a new directory that will contain the ROS2 workspace

mkdir -p formation_ros2_ws/src
cd formation_ros2_ws/src

Create a package called **formation_control** from the **src** directory using

ros2 pkg create --build-type ament_python formation_control

Package configuration

Add dependencies in **package.xml**

<exec_depend>rclpy</exec_depend>

<exec_depend>std_msgs</exec_depend>

<exec_depend>ros2launch</exec_depend>

Edit the **setup.py** to specify the launch file **max_launch.py** as

(i) include the header "**from** glob **import** glob" and to the **data_files** list:
("share/" + package_name, glob("launch_folder/formation_launch.py"))

(ii) specify the entry points, i.e., the name of the ROS2 node associated to the source file **the_agent.py**
"generic_agent = formation_launch.the_agent:main"

Package build and run

Include the (single) source file **the_agent.py** of the ROS2 node, which is to be located at
formation_ros2_ws/src/formation_control/formation_control

From the ROS2 workspace root **formation_ros2_ws** build the package
colcon build --symlink-install --packages-select formation_control

Then

- activate ROS2 (if needed)
  . /opt/ros/humble/setup.bash
- run
  . install/setup.bash
- execute the launch file
  ros2 launch formation_control formation_launch.py