default

January 8, 2023

1 TP1 Mission Coordination ROS-Gazebo | SENDIL EFE-M2SAAS

1.1 PART 1 and 2:

Q1: In this new terminal (the last one), you should see some data. What do this text represent?

[ ]:

```
roslaunch evry_project_strategy agent.launch nbr_robot:=1
```

This is the robot's distance to flag. But we didn't reach so we need to build a strategy to reach the flag.

Q2: What is list command used for and what is the result?

[ ]:

```
rostopic list
```

This command is used to list current topics published or subscribed to the robots.

At first we see gazebo topics as the simulation environment is still running.

Also we have see cmd_vel, odom and sensor/sonar_front topics for each of 3 robots. cmd_vel is used to move our robot. Also we can get the robots pose(position+velocity) information from odom.

Q3:According to you, who are the publisher and the subscriber? rostopic info /robot_1/odom

/gazebo is the publisher

/agent_1 is the subscriber

Q4: According to you, what type of messages are published on this topic? nav_msgs/Odometry messages are published

We can use ”rostopic type /topic_name | rosmsg show” to get more information about the messages that we are interested

For example :

[ ]:

```
rostopic type /robot_1/odom | rosmsg show
```

Gives us more information about geometry_msgs/Pose and geometry_msgs/Twist

Q5: What is echo command used for and what is the result? rostopic echo /robot_1/odom

This command used to get pose+twist information of the robot in each step. pose is robot's position and orientation, twist is robot's linear and angular velocity.

1.2 PART 3 : Move on robot

Q6 Q7 and Q8 : Simple approach to go the goal and the timing strategy to avoid robots collision

[ ]:

```
def run_demo():
    """Main loop"""
    robot_name = rospy.get_param("~robot_name")
    robot = Robot(robot_name)
    print(f"Robot : {robot_name} is starting..")

    # Timing
    rospy.sleep(5 + int(robot_name[-1]))
    while not rospy.is_shutdown():
        # Strategy
        Kp = 0.5
        distance = float(robot.getDistanceToFlag())
        print(f"{robot_name} distance to flag = ", distance)

        # Write here your strategy..
        velocity = Kp*distance
        angle = 0
        if(distance < 1):
            velocity = 0
        # Finishing by publishing the desired speed.
        # DO NOT TOUCH.
        robot.set_speed_angle(velocity, angle)
        rospy.sleep(0.5)
```

As I defined a P gain, the velocity is changing regarding to the distance from the flag, then the robot slows down when it gets closer to the goal, and if the distance is less than 1, I set the velocity to 0

By runing :

[ ]:

```
roslaunch evry_project_strategy agent.launch nbr_robot:=3
```

All the robots will start to move respected to their delay