Report on Turtle Navigating To Target

Robert Knowles Officer

# 1. Task Overview

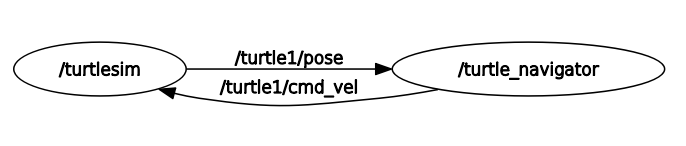
This assignment involved developing a ROS2 package that enables a turtle in pre-developed turtlesim package to be navigated automatically based on a user-specified target position. The user provides target coordinates (x, y) via terminal input, and the system ensures that the turtle reaches this position using ROS2 nodes, topics, and services. The primary goal is to implement an effective and efficient navigation system using ROS2 communication mechanisms.

# 2. System Design

## 2.1 Node Graph

The system makes use of only two nodes in order to generate the desired outcome: the turtlesim node and the turtle\_navigator node. The turtlesim node represents the ROS2 emulated robots movement in addition to providing real-time feedback with a visualisation of the turtles current position. The turtle\_navigator node was implemented for this assignment and is responsible for moving the turtle.

Shown here is the node graph that was generated using the ROS2 command *rqt\_graph* with *Nodes only:*



## 2.2 Services/Messages/Topics Implemented

### Topics

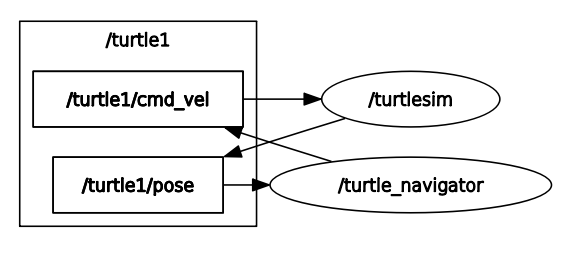
The system utilizes only previously implemented ROS2 topics, provided by the turtlesim package, in order to achieve desired outcomes. The /turtle1/cmd\_vel topic is used by the navigator node to publish velocity commands that move the turtle. The /turtle1/pose topic allows the turtle\_navigator node to read the current position of the turtle. The /target\_pose topic is used to transmit target coordinates, ensuring the navigation node receives the correct destinations.

### Services

In addition to basic movement, the package also offers additional commands to the user through the command terminal. These commands utilise the services offered by the turtlesim. The list is provided on startup as well as on the end of execution of the previous command. These are the commands that are included:

|  |  |  |
| --- | --- | --- |
| **Command** | **Description** | **Service used** |
| Clear | Clear the navigation panel | /clear |
| Home | Move the turtle to the centre of the page | /turtle1/teleport\_absolute |

Because only a single turtle is used, the /turtle1 namespace is used throughout the program.

Shown below is the final *rqt\_graph* generated:

## 2.3 Functionality Description

|  |  |
| --- | --- |
| **Function name** | **Function description** |
| direction\_callback | The callback function used in order to update the navigator node on the turtles current position. |
| navigate\_turtle | Used to move the turtle into a specific target location determined by the user. Continuous calculations occur in order to update both the angular and linear velocities, making use of trigonometric formulae, to align the turtle with the target position and ensure the turtle stops moving once the desired location is reached. |
| target\_reached | The function is used by the navigate\_turtle method in order to determine how far the turtle is away from the desired location. |

# 3. Results

The results of this assignment can be found in the video recording located in the uploaded directory “Report/videos”.  
  
The videos show represent the following user tests:

|  |  |  |
| --- | --- | --- |
| **Test** | **Start co-ordinates** | **End co-ordinates** |
| Middle 🡪 top right | {x: 5.5; y: 5.5} | {x: 10; y: 10} |
| Top right 🡪 bottom right | {x: 10; y: 10} | {x: 10; y: 0} |
| Bottom right 🡪 Top left | {x: 10; y: 0} | {x: 0; y: 10} |
| Middle 🡪 top left | {x: 0; x: 10} | {x: 5.5; y: 5.5} |

These movement commands where followed by a clearing of the screen, a final move to coordinates {x: 10; y: 10} and a teleportation to the center.

The command log can be found in the same directory in the txt file named “command-terminal-log.txt”.