



AARHUS UNIVERSITET

Department of Electrical and Computer Engineering

Assignment 1

Implementing robot circling with RL
and measuring individual vs. group
performance

Group 4

Alba Arcos I Ribas

Student no. 202503224

Birnir Þór Árnason

Student no. 202503220

Martin Skelde Krøjmand

Student no. 201710685

Wouter Jonathan Oudijk

Student no. 202109059

Zsombor Krisztián Szöke

Student no. 202411164

Teacher: Mirgita Frasher and Lukas Esterle

Number of characters: 1777 (1 standard pages)

Contents

| | |
|---|----------|
| 1 Implementation | 2 |
| 1.1 Hard-coded solution | 2 |
| 1.2 RL-learning | 2 |
| 1.2.1 Alba | 2 |
| 1.2.2 Krisztián | 2 |
| 1.2.3 Wouter Jonathan | 2 |
| 1.3 Subsumption Architecture and Obstacle Avoidance | 2 |
| 2 Results | 3 |
| 3 Conclusion | 4 |
| Bibliography | 5 |

List of Figures

List of Tables

1 Implementation

1.1 Hard-coded solution

To get familiar with the IR-Sim library, we first hard-coded a robot to move in a circle. To do this we first found the source code for the *Dash* behaviour when using *Diff* kinematics[1]. This basic behaviour simply makes the agent move directly towards the goal, slowing down to turn if necessary. It does so by calculating the difference in the angle of the agent and the angle between the agent and the goal, which is the angle the agent should have to move towards the goal, and the agent then simply turns to reduce this difference to zero.

To make the agent instead move along a circle, we simply add 90 degrees to the angle between the agent and the goal. This makes the agent move along the circle tangent and assuming infinitely small steps, this would result in a perfect circle. However since we don't have infinitely small steps, the agent will move further and further away from the circle with each step. To account for this, we simply take the distance from the circle center and add this to the equation, increasing the angle when the distance is less than the desired radius and decreasing the angle when the distance is more than the desired radius. This results in the code seen in listing 1.1.

```

1 # Get the distance and the angle to the goal (Circle center):
2 distance, radian = relative_position(state, goal)
3
4 # Calculate the distance correction amount:
5 distanceCorrection = (circle_radius - distance) * correction_multiplier
6
7 # Add 90 degrees + the distance correction to the angle between the agent and the goal
8 radian += np.pi/2 + distanceCorrection

```

Listing 1.1: Implementation of getting the desired angle for the agent

1.2 RL-learning

1.2.1 Alba

1.2.2 Krisztián

1.2.3 Wouter Jonathan

1.3 Subsumption Architecture and Obstacle Avoidance

2 Results

3 Conclusion

Bibliography

Websites

- [1] Hanruihua. *IR-SIM DiffDash documentation*. Last visited: 02-10-2024. URL: https://ir-sim.readthedocs.io/en/stable/_modules/irsim/lib/behavior/behavior_methods.html#DiffDash.