IAR - Task 1 Report

Jakob Calero - s0948339
Samuel Neugber - s0821562
October 1, 2012

1 Abstract

In a known environment and with a simpel task one can find all cases and reduce them to simple if statements, doesn't look smooth though. Likely to not work that nice for most other cases.

2 Introduction

We approached the problem of simple robot navigation using two separate approaches: Discrete, functional behaviour and more continuous, differential control, as described in the first few chapters of Vehicles (FOOT). The Khepera II robots are circular, two-wheeled robots, which come equipped with 8 infra-red sensors arranged as seen in diagram 1. (DIAGRAM) The task was to move a Khepera II robot around in a relatively static (enclosed) testing arena (DIAGRAM) while following walls, if possible, and avoiding obstacles, if necessary. In order to achieve this behaviour we used discrete conditional cases to determine the current state and act accordingly on the one hand, as well as a more direct mapping between sensor values and motor speeds on the other hand.

3 Methods

3.1 Gathering Data

For either approach, the control program has to know which values of the IR sensors represent certain thresholds for the distance of the robot to an object. The IR sensors have a range of values between 0 and 1020, where the base level is usually around 50 (indoors. daylight and artificial light) and putting the robot right next to a white wall gives readings of 1000+. Since the relation between sensor values and distance of the robot to an object describes a logarithmic function (DIAGRAM) we found that sensor values of 200+ describe a sensible value representing the robot being near an object, regardless of surface. Additionally, we found that values of 500+ mean that we are too

close to an object and values ranging between 180 and 250 are what we are looking for when trying to follow a wall without getting too close or too far away from it (again, regardless of colour/material of the wall). From these values we then extrapolated if necessary.

3.2 Functional Control

The first approach we took was to identify cases in which the robot needs to react, capture them as conditionals and then identify appropriate behaviour for these cases. As seen in (DIAGRAM) our final control code for this methodology uses just four if-statements, representing seven conditions:

- There is an object to the front-left or too close directly to the left, so turn right.
- There is an object to the left, but slightly too far away, so turn left.
- There is an object to the right, but slightly too far away, so turn right.
- In any other case, move forward.

The first two cases use the IR sensors in the front as well as the furthest out sensors on the sides to avoid obstacles in the front of the robot as well as ones which are to the side of the robot that would get in the way of a direct path ahead. The next two cases attempt to capture the fact that the robot may turn away from a wall when avoiding collision so it should turn back towards the wall, if it moves too far away.

3.3 Differential Control

The second approach we took to solve the task ahead was one described in the first few chapters of Vehicles. (FOOT) Our goal was that, ideally, the robot should always be moving and do so as smoothly as possible (if only to look appealing). The control code in (DIAGRAM) shows that in order to do so, each iteration of the control loop resets the wheel speed to a maximum value and then reduces the speed of the wheel on the opposite side of the corresponding IR sensor if certain thresholds are met. The higher the value of the sensor, the more the speed of the corresponding wheel is reduced.

4 Results

5 Discussion

6 Appendix