

Coursework 1: Wall Following with a LEGO Robot

Thomas AE. Smith

March 12, 2014

1 Introduction

This paper describes the construction of a small, simple robot using the Lego NXT system, and the implementation of a basic subsumption architecture to govern its circumnavigation of an area; as described in Brooks (1991) and discussed in more detail in Brooks (1986).

2 Approach

I worked alone on this project, using one of the commercial NXT kits rather than the educational kit - supplemented with a set of wheels from my own supply that matched the ones provided with the educational kit.

2.1 Morphology

The design of the robot was intentionally simple in order to reduce the complexity of the systems needed to control it. A two-driving-wheels-one-follower differential steering approach was chosen to maximise manoeuvrability, with a fixed, forward-mounted ultrasonic sensor positioned close to the axis of rotation and inline with the axis of motion. The robot's forward profile and overall area were minimised in order to reduce collision risk and simplify detection, and a pair of bumpers mounted at the forward point covering the sides and top of the profile, linked to sensors at the lateral extremities. One issue encountered during development was the length of the sensor and motor cables - often these would protrude beyond the sensor profile and snag on obstacles, altering the path of the robot. These were stowed carefully in the final design.

2.2 Behaviour

The robot was controlled using a simple two-layer subsumption architecture as described in Brooks (1991). Layer 0 was initially configured to drive the robot forwards in the absence of any input, while Layer 1 monitored the ultrasonic sensor and the left and right bumpers, and initiated avoidance behaviour if an obstacle was detected. The object of the behaviour was to circumnavigate a typical room however, and it was found that this approach often resulted in a criss-cross behaviour that did not resemble circumnavigation. The Level 0 behaviour was therefore modified to introduce a gentle curve dependent on the location of the last obstacle detection - that is, the robot would attempt to 'go around' whatever it had detected, and if it transpired that this obstacle was wider than the radius of the curve (R), this would be discovered by means of a later detection and avoidance.

2.3 Experiment

A small range of values for R were tested for the circumnavigation of a typical kitchen. The robot was placed in the same position and orientation for each of five repetitions of each of three values of R — 60cm, 80cm and 100cm. The timer started at the same time as the robot, and was stopped once the robot had circumnavigated the area and detected an obstacle beyond its starting point. Each condition was repeated until five successful datapoints were collected - causes for unsuccessful attempts are described in Appendix B. Sources of variation in running times within values of R include the unreliability of the ultrasonic sensor when approaching objects at low angles of incidence or tea towels, non-deterministic outcomes if a collision occurs while the robot is moving backwards, and potential non-deterministic outcomes from bumper collisions due to wheel slippage or friction. In order to help discount the decrease in battery charge over the course of the experiment as a potential bias factor, the first experimental condition was repeated once after the other experiments had been performed, and the time obtained compared to the original mean.

3 Results

Larger values for R resulted in faster circumnavigations of the experimental area. $R = 100\text{cm}$ (mean. 83.2s, s.d. 2.482s) was around 40 seconds faster than $R = 60\text{cm}$ (mean 125.4s, s.d. 4.363s), with $R = 80\text{cm}$ falling in between (mean 98.2s, s.d. 5.231s). $R = 80\text{cm}$ was the first condition tested, and so an attempt with this value was repeated at the end of the experiment, resulting in a time of 94 seconds — within one standard deviation of the mean for that value. This provides preliminary evidence for the assumption that battery charge level was not an important factor across the experiments performed.

4 Discussion

Although the numerical results come out in favour of $R = 100\text{cm}$, other factors than time alone may potentially influence the choice of an ideal value for R . In this experimental setup, the centre of the area was clear of obstacles, allowing large arcs away from the wall to still arrive at their intended destination without interruption. An alternative environment (for example, a classroom — where only the areas near the walls might reasonably be expected to be clear) would favour smaller values for R , as these result in the robot following paths that remain closer to detected obstacles. An interesting avenue for further research would be the back-away radius r , where smaller angles might result in closer wall following but more attempts needed for high-angle-of-incidence collisions, and larger angles vice-versa. A possible extension would be to use ultrasonic readings from before and after obstacle avoidance re-orientation to hypothesise a continuation of a given obstacle and attempt to fine-tune a planned course to follow it.

5 Conclusion

I constructed a simple robot with differential steering and two forms of sensing, and implemented a basic subsumption architecture to govern its behaviour when circumnavigating a room. The approach taken (back-away-and-reacquire) was influenced by the simple morphology of the robot. Routes that minimise future collisions by giving detected obstacles a wide berth maximise circumnavigation times, but sacrifice proximity to the perimeter of the area. Off-the-shelf commercial robotics toys today can match or exceed state-of-the-art research equipment from 1986.

References

Brooks, R. A. (1986). A robust layered control system for a mobile robot. *Robotics and Automation, IEEE Journal of*, 2(1):14–23.

Brooks, R. A. (1991). Intelligence without representation. *Artificial Intelligence*, 47(1–3):139–159.

A Raw Data

B Experimental Outcomes

Experimental attempts fell into four categories:

Successful: The robot successfully circumnavigated the area and detected a feature beyond its original starting point. A datapoint was recorded with the number of seconds it took to do so.

Crash: The provided ultrasonic sensor API functions are not thread safe. Due to the use of both motor and sensor listeners, occasionally one detection is still being processed when another arrives, resulting in a `ConcurrentModificationException`.

Fail: Two main failure modes were observed during the course of experimentation. The robot has a small blind spot directly ahead of it, that the side bumpers do not cover. If an object sufficiently thin or angled to evade ultrasonic detection (such as a metal chair leg or corner of a unit) enters this blind spot, the robot may become stuck against it. Alternatively, due to the triangular wheelbase it is possible in some circumstances for the robot to tip up and become stuck on two wheels or collapse.

Stall: As the robot has no rearward-facing sensors, it is possible for it to collide with an object while moving backwards to avoid another collision. In most cases this is not a significant problem, though it can non-deterministically change the robot's orientation. In some cases it is possible for the power of the motors to drive the rear wheel up the object sufficiently for the ultrasonic sensor to detect the floor as a false positive, and attempt to back away from it. This leads to a stall in the motors.