CM30229 - Circumnavigation using Lejos

Dominic Hauton

February 28, 2017

0.1 Introduction

The goal of this research is to see the effect of changing the rate of sensing on the navigation capabilities of the rover while circumnavigating a room. The rover was built using parts from the LEGO Mindstorm NXJ kit. There were 3 sensors; a *Sonar*, *Light Sensor* and *Bump Sensor* (x3). Three motors were provided for movement and navigation.

To make informed decisions all of the sensors were used and the sonar was mounted on a pivot to allow for measurements in multiple directions, as shown in figure 1. To provide movement two driven wheels were used and a trolley wheel was placed at the back for stability.

During development a reactive approach was taken, using the subsumption architecture [Wooldridge, 2009] with emphasis on sensing using multiple layers of perception to allow for informed movement decisions. The use of the subsumption architecture was based on Brooks' thesis that Intelligence is an emergent property of certain complex systems. [Brooks, 1991] During development this subsumption approach was added into the perception mechanism, during which the rover had to to decide on it's current situation.

Figure 1: Rover Head Sensor

0.2 Approach

The initial design of the rover was a simple reflex agent as described by Russell et al. [1995]. This took the sensor readings, and using transduction converted them to a proximity reading in every direction. This allowed to rover to react to a crash but when recovering from a crash the rover had no context. As a result I settled on Russell et al. [1995]'s model-based reflex agent. This allows the sensors to modify the state at individual rates, in the final design there were two sensing thread loops, operating at different tick rates, as shown in figure 2. One for the sonar which provided slow but accurate readings and one for the other sensors which provided almost instantaneous measurements.

All of the sensor readings were mixed and the closest reading was taken. This was especially important for the front sensor where 3 separate readings were taken from the bump, light and sonar sensor. To improve sensor quality 15 readings were taken every measurement and any reading more than two standard deviations away from the mean were thrown out, and the remainder of the readings were averaged.

In order to control the rover a third thread was started. This read the state and converted the state into a meta-state for the rover. An example of such a state was *TOO_CLOSE_TO_WALL* or *WALL_NOT_FOUND*. To determine this

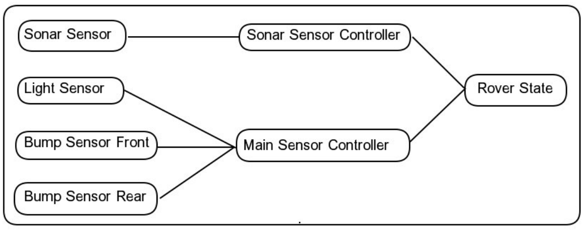

Figure 2: Final Perception Architecture

state a subsumption architecture used. For example, if the rover was *NEAR* something to the front, it had crashed and it did not matter that *NEAR* an object on the left too. This meta-state was then used in the action planner, which selected the best appropriate action given a meta-state and current states. If the rover was crashed and it was *NEAR* an object to the left. It shouldn't try to reverse to the left.

In addition the meta-state could direct the sonar. If it was following a wall on the left and had found the wall, it could just look left and forward for faster readings. There was no need to update the RHS. This basic system describes the platform for experimentation. The goal was to find if there was an optimal sensing rate.

The rover was run at a tick rate (measurements per second) of 5, 25 and 45. To rate the rover performance during each experiment I measured the number of impacts with the wall as well as marking the number of times it was stuck while navigating the test track shown in fig 3. The hypothesis is that the rover will perform better with a higher tick rate, but after a certain level, increasing it will no longer have an effect. The former measurement was recorded in software and the later was done by observation. If the rover was stuck in a corner,

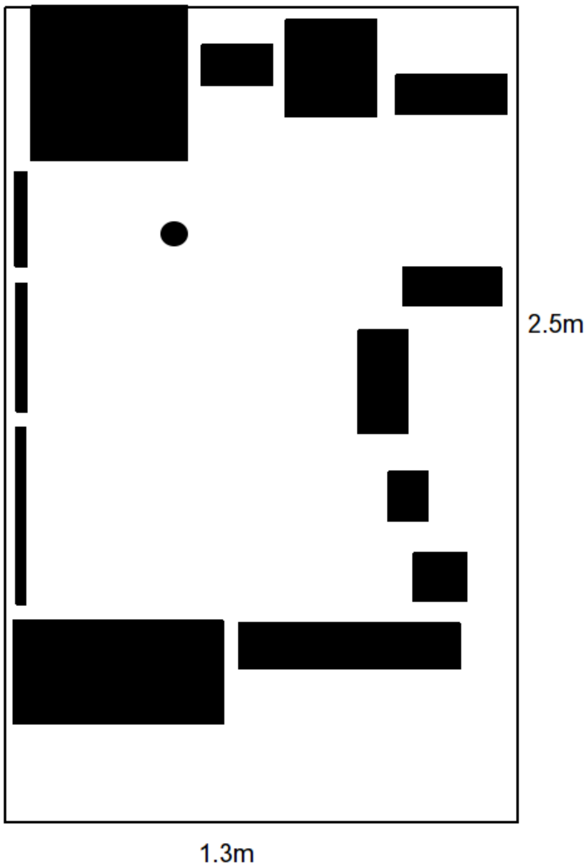

Figure 3: Test Area Layout

the experiment was restarted. During each run, the rover was started in the bottom left hand corner of the arena facing the left wall.

Throughout the research the rover was shared with Ryan Cullen. In the first part of the coursework the rover resided in my house. I built the rover itself, optimising it's design over several iterations and created the perception system of the code. The software continuously modified proximity readings in the state, indicating how close the rover was to an object in all four cardinal directions. At this point the code was then forked by Ryan and we each developed our

Table 1: Test Results

Tickrate	5	25	45
Bumps/lap @ Slow	23	14	16
Bumps/lap @ Fast	53	46	26

own action selection code based on the rover state.

0.3 Results

The hypothesis of the experiment was as expected. With this positive result I also ran the experiment after doubling the rover movement speed and found that a higher tick rate actually assisted the rover at higher speeds, leading to the conclusion that the faster the rover moves, the faster it needs to sense the environment around it.

The results for one lap were very noisy, but as the rover was required to do 10 laps, both 10 lap runs were within 5 bumps / lap of each other.

The video found at `https://youtu.be/LOPdOOw1Uec` depicts the start of a trail run.

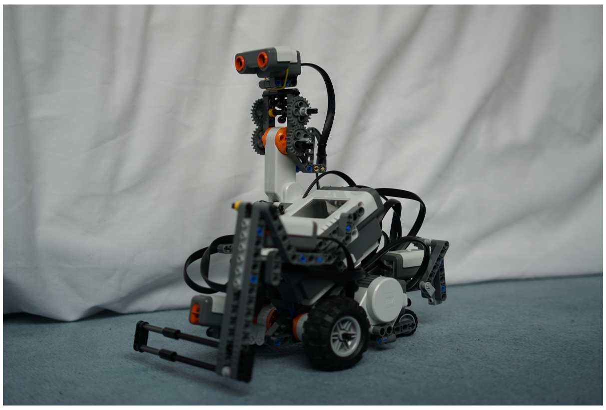

Figure 4: Stanley Rover

0.4 Discussion

The results I got closely matched the hypothesis, however, I was not able to increase the sensing speed of the sonar due to physical limitations, which is the key part of the navigation logic. I suspect that adding a second sonar that did not require physical movement for sensing would greatly increase the navigation capabilities of the rover.

During the experiments I tried to run the rover in as fixed of an environment as possible, as I found the light sensor was very susceptible to sunlight and found that closing the blinds made runs at different times of day much more consistent.

In general I think that to improve navigation, sensing and decision loops should always be run as fast as possible, and the faster the movement the more important this is. This extremely important in applications such as quadcopters where loop times affect flight characteristics [Betaflight, 2017]

0.5 Conclusion

This research set out to find the impact of sensing speed on the navigation of rovers. The results show that a faster sensing loop results in more accurate navigation, although, after a certain accuracy is reached returns diminish.

When deciding on sensing loop speeds, this level should be reached and not surpassed, as going too high might take compute cycles away from the planning loop.

Bibliography

Betaflight. Betaflight FAQ. `https://github.com/betaflight/betaflight/wiki/Frequently-Asked-Questions#what-is-2khz-mode-`, 2017. [Online; accessed 28-Feb-2017].

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

Stuart Russell, Peter Norvig, and Artificial Intelligence. A modern approach. *Artificial Intelligence. Prentice-Hall, Egnlewood Cliffs*, 25:27, 1995.

Michael Wooldridge. *An introduction to multiagent systems*. John Wiley & Sons, 2009.