Cognitive Robotics
Final Report

Serge Thill

Released: November 26, 2020

1 Task

Use Nengo to create a cognitive agent that can navigate a simple grid world. In particular, this grid world contains colour-coded locations (Red, Green, Blue, Yellow, and Magenta). Your agent needs to be able to perform the following behaviours:

1. The agent should randomly explore the environment, remembering the colour of all coloured squares it encounters
2. Once the agent has encountered n different colours, it should stop moving. n is to be provided by the user of your simulation; you decide how (e.g. a simple variable in code, via an input node, or via a spa buffer).

You will be provided with the basic code for the gridworld and the agent. The agent code contains a simple variable that assigns a numerical value to each of the possible colour and that can be read out to know the colour of the cell the agent is currently occupying. An example map is also included; note however, that the agent may be evaluated on one or more *different* maps. Your agent should be able to function even if the map is initially unknown. *Do not modify the provided grid.py file.* If you choose to modify the agent's abilities, document those.

Some points you should consider and/or decide upon:

- Should this be a SPA model, a simple network, or both?
- The agent comes with a basic distance sensors and the ability to detect the colour of the cell it currently visits. It cannot in principle see colours at a distance. What kind of robot platform does this mimic?
- The function that detects colours encodes them as simple integers, which is fine for disambiguating between them but might not be particularly realistic. You can use this as is, or you can decide to transform that input into something that appears more plausible.
- How will the agent navigate the world? There are many possible strategies to choose from, some more algorithmic, some more cognitively plausible.
- How does the agent know it has finished the task?

2 Report

Your report should consist of two parts: the first part documents and comments on your solution to the task above; the second part is a reflection on cognitive robotics based on this task as well as the course as a while.

2.1 A simple cognitive agent

Provide a description of the functionality of the agent, including all design choices you made and the strategy it implements. The report should also provide a reflection on the behaviour of the agent, and any mistakes it might make. Focus on the agent's cognitive abilities (for example: what kind of information can it process? What kind of information can it keep in memory and how are these memories implemented? How adaptive is the agent to unknown environments and situations?) Example discussion points include:

- How are the pieces of information the agent needs represented internally? Is this efficient/plausible? What alternatives might one have considered?
- Are there fundamental limitations on the agent's behaviour? For example, does it rely on specific features in the environment? Is there a limit to the length of the colour sequence it needs to navigate?
- What behaviour does the agent show if things go wrong (say, it cannot find a specific colour, or it forgets the target sequence)?
- What limitations arise from the simple nature of the agent's embodiment? In particular, consider the default color detection (possibly against any changes you might have introduced).

2.2 Reflection

Throughout the course we have seen many facets of cognitive robotics and the practicals have given you some initial insights on how we can produce implementations based on this, even if the agent we have developed here is really quite simple, both in terms of its sensorimotor abilities and in terms of the environment it exists in. In this part of the report, you are asked to reflect on the possibilities and limitations within cognitive robotics today: what can we do, what can we not do, where should research be headed next? For example, you could consider:

- To what degree are simple agents like we have built here useful for cognitive models (think about minimal cognition approaches)
- What would be concrete implementational issues with theories we have seen throughout the course. Be concrete with examples.
- What features should a cognitive architecture contain? Should we favour emergent or cognitivist approaches? What are the advantages and disadvantages of each?
- What might fundamentally have been different if the task in this assignment had used Dynamic Field Theory instead?

Note that these are just examples; feel free to come up with your own discussion points *ensuring that what you discuss is a reflection on the current possibilities and limitations of Cognitive Robotics and motivated insights of where research should move next.* You can focus on any topic within Cognitive Robotics that is of interest to you, but you should motivate *why* you consider your chosen topic to be of particular interest in a discussion of the field as a whole. What does it highlight that other topics could not?

You should make use of what you learned while implementing the agent. This does not mean you need to restrict yourself to the abilities of this agent: you can reflect on what it would take to build a more complex agents with additional abilities based on your experience.

3 Format and submission

There is no fixed length for this report, but you are not required to produce exessive amounts of text. Do refer to the literature where appropriate. Submit both your code (.py and the visualisation configuration .py.cfg) and the report (pdf) as a single zip file on Brightspace. The deadline is as specified on Brightspace.

4 Assessment (25p total)

4.1 Agent implementation (10p)

- How is the ability to recognise and remember colours implemented? (3p)
- How is the ability to count different colours and stopping after a number implemented? (3p)
- Does the report accurately describe the functionality and limitations of the submitted agent? (1p)
- Does the discussion of the implementation demonstrate an understanding of cognitive robotics in that it is capable of reflecting upon the implemented solution and evaluating its effectiveness and design from a cognitive perspective? (3p)

4.2 Reflection (10p)

- Does the reflection build on both content from the course and the agent implemented in this assignment? (2)
- Are the arguments well motivated and based on the current state of the art, as demonstrated by references to the literature? (3)
- Is the reflection insightful and critical? Does it discuss a relevant topic in Cognitive Robotics? (5)

4.3 Presentation (5p)

- Is the report well written stylistically and grammatically? Is it nicely presented and consistently formatted? Is it proofread and free of typos and other mistakes? (5p)

5 Bonus assignment (+2.5p)

For a bonus on the grade, take your solution to this exercise and adapt it to control a robot in CoppeliaSim (see later practical). You'll have to decide on an appropriate robot and environment design. The intention is to show that your main architecture is independent of the specific agent; the bonus points are *not* awarded for a separate implementation. Additional submit your scene file and all code if you intend to claim the bonus.

6 Collaboration and group work

You will self-enrol into groups of 2-3 persons. Within these groups, you are encouraged to collaborate on this assignment, in particular to decide on strategies for solving the assignment and implementing them.

You are also permitted to discuss between groups, in particular when it comes to solving bugs or how to approach specific functionalities in Nengo code. The solution to the assignment should be your own (as a group) though.

The final report is individual and should contain your own reflections, not the reflection of a group. **The report must mention all collaborations that took place**.

You will assess your own groups using the standard peer review form. Information provided here can be taken into account in the marking.

7 Hints for the implementation of the cognitive agent

- Do test different maps. Note that some maps can be harder than others (a circular corridor can be solved by just circling it until all colours have been visited, but other maps need more elaborate exploration strategies).
- Consider what happens if the agent is put in an environment that does not correspond to the initial brief. For example, what happens if the map contains multiple squares of the same colour, not all colours are used, or new colours appear?
- It is often helpful to include screenshots in the report.
- If you start Nengo from the directory where you have the code for the assignment, the files will be directly selectable in the left pane.
- Remember to take regular backups, do not forget to save, and protect yourself against crashes (e.g. regularly copy the code, or edit in an editor that does not depend on Nengo and the Browser both behaving).
- Comment your code. Describe what you intended to do with a particular code even if it does not work.
- Submit a report even if the code does not work; discuss where it fails and why; discuss your chosen strategies and what elements are necessary to implement it and focus the reflection on this hypothetical design.