A Cognitive Approach for Reproducing the Homing Behaviour of Honey Bees

Xin Yuan, Michael John Liebelt and Braden J. Phillips
The School of Electrical and Electronic Engineering, The University of Adelaide, Adelaide, South Australia, Australia

Keywords: Artificial Intelligence, Honeybee, Agent-based system, Cognitive Computation.

Abstract: We describe the implementation if an agent-based controller for an autonomous robot with cognitive abilities that reproduce homing capability in the foraging behaviour of the honeybee. The agent is based on a symbolic representation of data and information and is written in a language designed to describe fine-grained large scale parallelism, the Street language (Frost et al. 2015). The objective of this approach is to enable the direct translation of agents written in Street into embedded hardware, to achieve compact, power efficient, autonomous cognitive processing capability.

1 INTRODUCTION

In the field of Artificial Intelligence research, the cognitive architecture approach has shown considerable promise in implementing autonomous cognitive agents (Laird 2012). However, a limitation of these systems is that they are constrained by the memory capacity and computational performance of the general-purpose computing platforms on which they run. As cognitive agents become more complex and have longer lifetimes, the number of matching conditions that must be evaluated concurrently increases and, in particular, the amount of long-term and working memory required increases rapidly. Much design effort has been invested in moderating memory requirements in these systems. Our focus is on engineering low power cognitive processors that can be embedded in autonomous devices so we have been investigating dedicated hardware systems that can implement cognitive architectures.

Our developing architecture broadly follows a production rule-based parallel processing method (Frost et al. 2015; Numan et al. 2015). It uses a potentially large array of dedicated production rule evaluation processors, which we call “productors”, to concurrently match elements in a distributed memory and to take actions by updating working memory. The production rules and actions are expressed in a customised language, loosely based on OPS-5 (Forgy 1981), which we call “Street”.

Currently, we are using a Java-based simulation and debugging environment to develop agents in Street, and we are concurrently pursuing agent development and hardware-mapping as part of our research. A key element of this research is verifying the capability of the Street language to capture useful levels of cognitive behaviour, and understanding how requirements for productors and working memory capacity scale up with agent complexity. We have identified the homing behaviour of honey bees as a suitable test case for validating our approach.

It is known from neuroethological research (Menzel & Giurfa 2001), that cognitive behaviour requires experience-dependent adaptation of neural networks, and researchers believe that these procedures require a more advanced and complex neuronal system than has been discovered in the insect brain. However, observations and experiments have shown that the honey bee (Apis mellifera) is an exception. (Menzel 2012). It is one of the most broadly researched eusocial insects in the field of animal ethology. Of particular interest is its ability to use magnetoreception (Wajnberg et al. 2010) and landmark recognition (Fry & Wehner 2002; Gillner, Weiß & Mallot 2008) with memorization support for homing navigation after long-distance foraging. (Menzel & Greggers 2015; Menzel et al. 2005)

Honey bees have a capability of memorising landmarks and magnetic field changes during a foraging trip, (Fry & Wehner 2002; Labhart & Meyer 2002; Menzel & Greggers 2015) and they use these memories as references to navigate back to their hive. (Menzel & Greggers 2015) This behaviour requires

Yuan X., Liebelt M. and Phillips B.
A Cognitive Approach for Reproducing the Homing Behaviour of Honey Bees.
DOI: 10.5220/0006195705430550
In *Proceedings of the 9th International Conference on Agents and Artificial Intelligence (ICAART 2017)*, pages 543-550
ISBN: 978-989-758-220-2

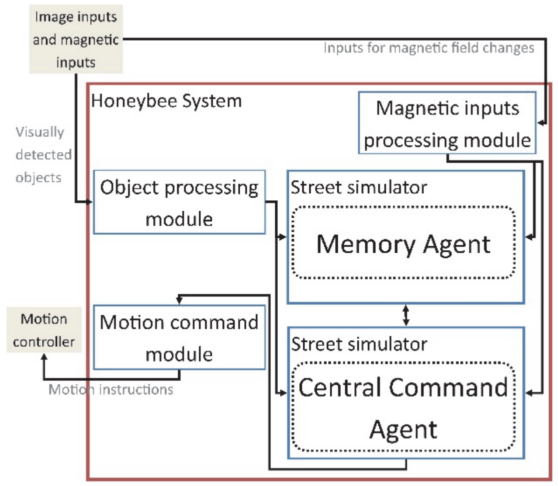


Figure 1: System Structure and Interaction.

memorization of characteristics of the experienced landmarks, such as colours and sizes, (Labhart & Meyer 2002) and magnetoreception (Wajnberg et al. 2010) at those memorised positions.

Randolf (Menzel 2012; Menzel & Giurfa 2001) expressed the architecture of the honey bee's brain as a set of communicating modules. Modules are associated with specific external stimuli. We have the option of reproducing this modular functional organisation by using separate, intercommunicating Street-based agents and it is one of the objectives of this work to understand whether the modular organisation is effective and efficient when implemented in dedicated hardware.

We propose to demonstrate that an efficient real-time cognitive agent that is able to reproduce the homing behaviour of the honeybee can be implemented in compact electronic hardware, based on our Street parallel production rule language. Thus, we set an objective for the behaviour of our agent, which requires it to reach a target and track its way back to the initial position using its memorised landmarks and magnetic data. Our hardware platform for this work is a rover vehicle comprising a platform with two drive wheels, a camera and magnetic sensor.

There are many advanced technologies inspired by the behaviour of insects, and many of them have made significant progress with applications. However, our approach differs from prior work in that we are using an architecture, loosely based on the cognitive architecture of insects, which can be efficiently implemented in electronic hardware and massively scaled up in complexity.

2 THE AGENT-BASED SYSTEM

2.1 System Architecture

Our system reproduces the central decision-making processes in honey bee homing behaviour by imitating its processing model in our agent-based system. Our system has a structure that duplicates the modules in a bee's brain (Menzel & Giurfa 2001), with similar interactions between the modules. The system structure and relationships between the modules are shown in Figure 1. The system connects to input devices that provide information about objects detected within the visual field of the rover and an angle w.r.to geographical north. Outputs are sent to a motion controller, which receives mobility commands from the Central Command Agent and drives the robot wheels to produce the required movement. The visual object processing module and the magnetic input processing module both receive digital numeric inputs from the sensors, but translate these into a symbolic representation. These two modules represent the sensory organs of the bee, which sense external stimuli from the real environment and transmit them into its nervous system in the form of chemical elements and electronic impulses (Giurfa 2007; Kiya, Kunieda & Kubo 2007; Koch & Laurent 1999). The vision module passes elements, which contain symbolic information about objects detected in the field of view, to two different modules, which are both running our Street engine simulator but loaded with different agents. One memorises the information gathered in the foraging process, and the other determines the movement actions to be made. This multi-agent structure allows agent to develop their own Working Memory without interaction or interference, which we hypothesize will allow simpler implementation. The details of these two agents are explained in sections 2.3 and 2.4.

2.2 Input Format

Our rule-based processing methods require data to be stored in terms of symbolic *elements* comprising a tuple of symbols, e.g. (`infoA infoB infoC`). (Frost et al. 2015) The number of symbols in an element is, in principle, unlimited, but, in many cases, elements capture single attributes of symbols or binary relationships between symbols and will therefore contain two or three symbols. Symbols may take any form. For clarity we assume that symbols may comprise any printable characters, excluding white space. Therefore, in the input processing modules, numeric inputs are converted into symbolic representation using fuzzy concepts. We restrict ourselves to processing

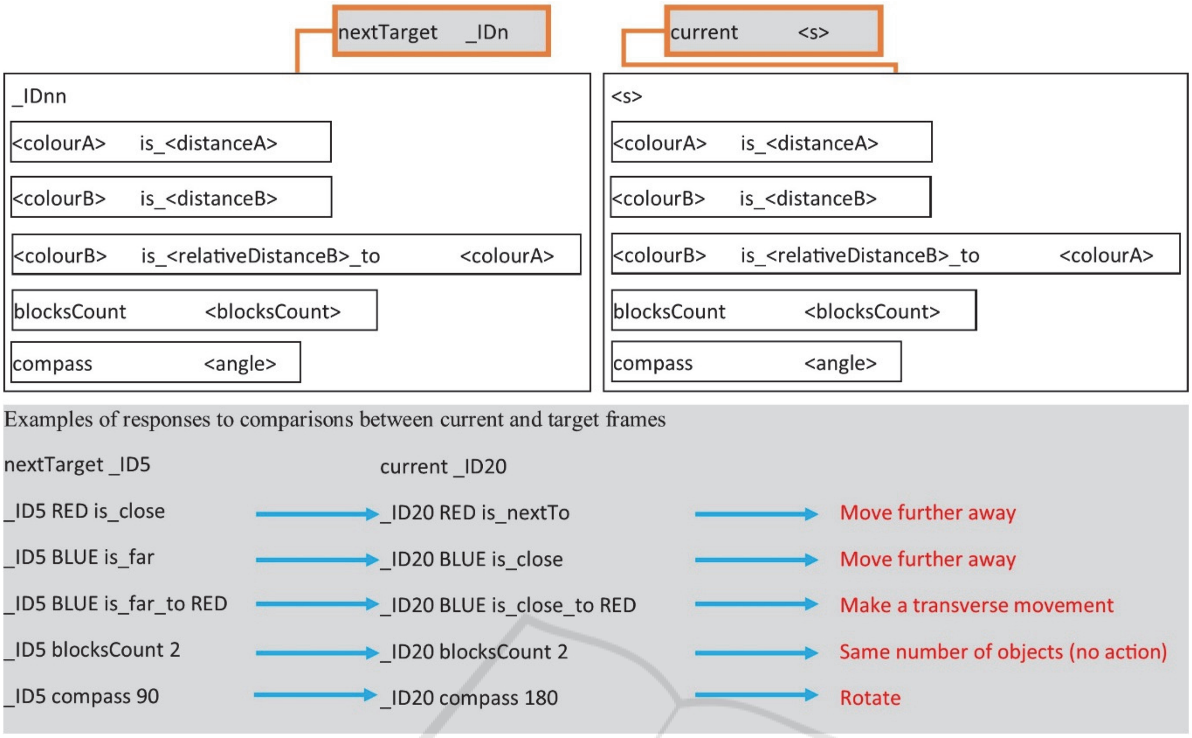


Figure 2.

imprecise representations of information because biological creatures, to the best of our knowledge, only recognise and process information in imprecise form. More specifically speaking, bees are not able to measure an exact distance to a visualised object, and they can only sense the surrounding magnetic field with limited precision. We define a format for sensory inputs based on this characteristic, such that an example of inputs describing a field of view in which there are two objects appearing is the following.

```
<s> RED is_close
<s> BLUE is_far
<s> BLUE is_far_to RED
<s> blocksCount 2
```

Each line is an individual element, in which `<s>` represents a unique reference ID for associating information contained in multiple elements. The number of elements is not fixed, and it depends on the number of objects appearing in the same frame. For example, if there are n different coloured balls detected, there will be n elements content the distance information to each ball and C_2^n elements describing the distance between each pair of balls. Also, the total number of the detected objects in the field is counted. We refer to a set of elements such as this, describing a single field of view and with a common ID, as a Master Frame.

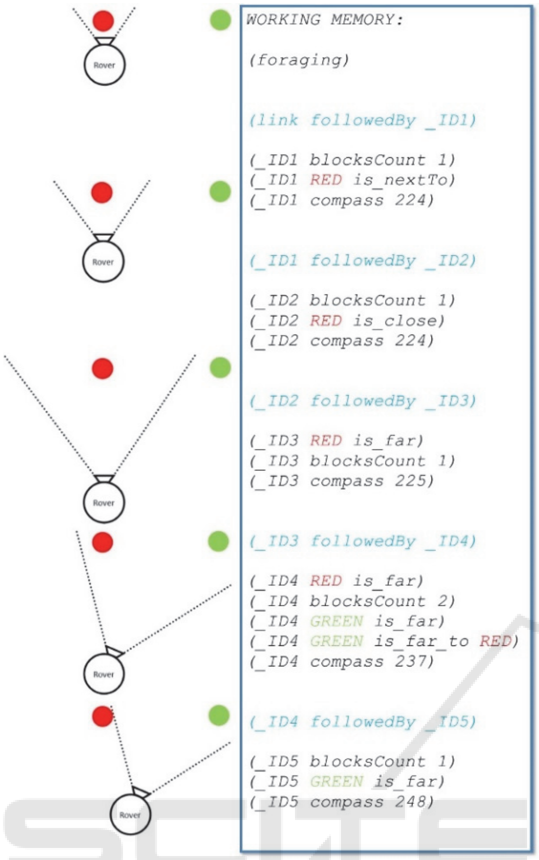
For the magnetic information, the format is as (`<s> compass <angle>`). However, this `<angle>` number is only used as a numerical symbol for the purposes of comparison between the current input and recorded angles to trigger rotation if they do not match. Section 2.4.4 introduces the details.

2.3 Memory Agent

This is an agent-based sub-system for information storage. This module imitates the episodic memory structure, which keeps a record of the bee's experiences, with some mechanism for tracing the order in which they occurred.

The memory agent contains rules, which, in foraging mode, take input elements from the image processing module and store these as a Master Frame. Each Master Frame that contains information that is recognised to be distinct from previous frames is given a unique reference ID `<s>`. These Master Frames are linked up by the memory agent using an element of the form `_ID1 followedBy _ID2`, and the reference ID of new inputs is stored in a new element with the last reference ID, as `_IDnn followedBy _IDNew`. As a result, a series of elements is constructed as a linked list of reference IDs in the agent's Working Memory.

Like the reaction when a honeybee starts to track a route back to its hive, when the system switches to homing mode, a command triggers the rules in the



```
WORKING MEMORY:

(foraging)

(link followedBy _ID1)

(_ID1 blocksCount 1)
(_ID1 RED is_nextTo)
(_ID1 compass 224)

(_ID1 followedBy _ID2)

(_ID2 blocksCount 1)
(_ID2 RED is_close)
(_ID2 compass 224)

(_ID2 followedBy _ID3)

(_ID3 RED is_far)
(_ID3 blocksCount 1)
(_ID3 compass 225)

(_ID3 followedBy _ID4)

(_ID4 RED is_far)
(_ID4 blocksCount 2)
(_ID4 GREEN is_far)
(_ID4 GREEN is_far_to RED)
(_ID4 compass 237)

(_ID4 followedBy _ID5)

(_ID5 blocksCount 1)
(_ID5 GREEN is_far)
(_ID5 compass 248)
```

Figure 3.

memory agent to output the last stored Master Frame, which provides the set of elements describing the last experienced episode to the Central Command Agent.

2.4 Central Command Agent

The Command Agent is the key controlling module of the entire system. Firstly, this agent decides on whether the system is in a free discovery foraging mode or in homing mode. The difference, or the trigger, between these two modes is whether a targeted object has been reached. If the information element representing the target has been received, the system will be switched into the homing mode and will start a route back to the original position.

This module receives information about detected objects from the input processing module, and it both transmits mode switching commands to the memory module and receives target information inputs from it while homing. In the homing mode, this module aims to find a match between the target objects, using a comparison process, as illustrated Figure 2. It attempts to determine the relationship between its current image frame and the next targeted Master Frame, and to infer what kind of movement the robot should make in order to make the two match. Based on the information received from the input module, there are multiple comparisons to be performed concurrently.

2.4.1 Distance to an Individual Object

Each image description in the memory agent contains elements that describe, in fuzzy terms, the perceived distance to each object in view. For example, the element

_ID5 RED is_close,

represents that a red object was detected to be close. While homing, if_*ID5* is the target Master Frame, the agent compares the current visual input with it to assess whether the same coloured object is being viewed at the specified distance. In the command agent, there are rules to react by moving to adjust the distance. If the target recorded is closer than an object currently visualised, the agent instructs the motion command module to move closer the object. Conversely, the agent outputs fall back instructions if the object is closer than the distance in the target Master Frame.

2.4.2 Distance between Objects

If more than one object is detected in the visual field, the visual processing module also generates information elements about the relative distance between the objects. Therefore, a distance between two objects is also a piece of information that can be used to trigger a movement command. For example, if the inputs about the contain the element

<new> BLUE is_far_to RED

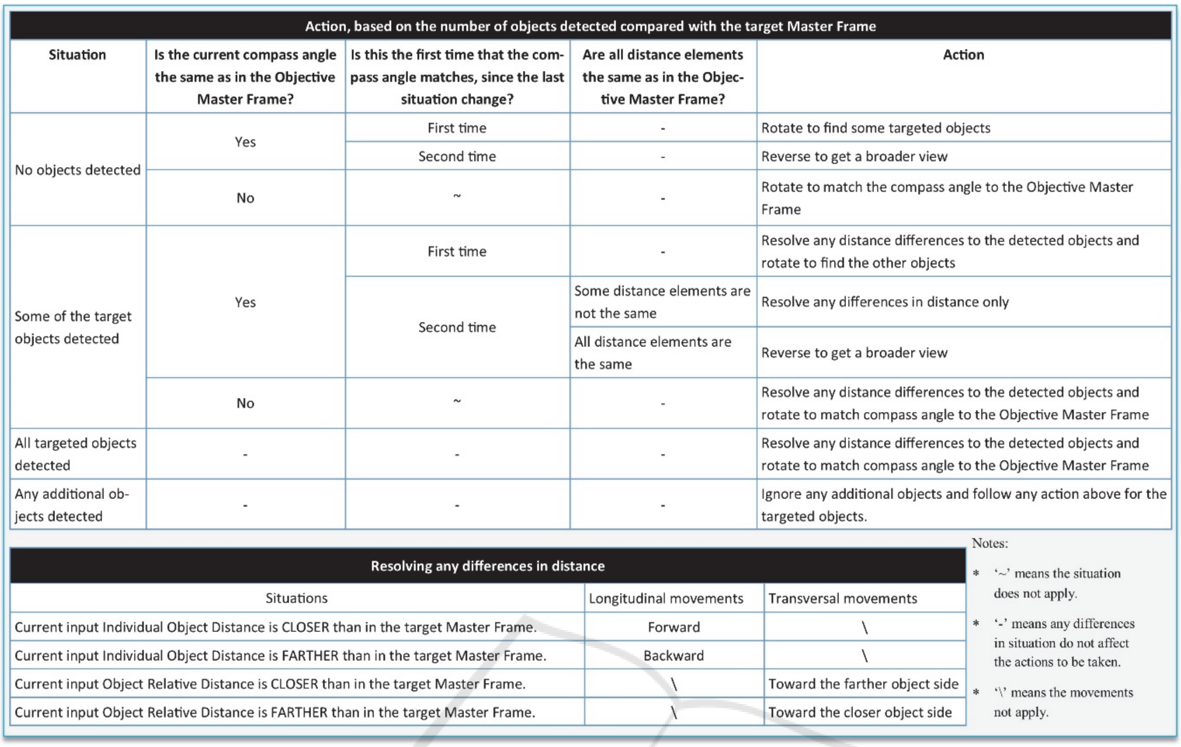
and the next image is sequence recorded in the memory has

_ID6 BLUE is_close_to RED

then, the agent will instruct the motion module to move in a direction that is closer to the two objects, to increase the perceived separation.

2.4.3 Number of Objects

Another index in a comparison is the number of objects detected. This is a significantly important feature for assessing the current position relative to a previous record. In order to assess either the current position matches that indicated by the target Master Frame, the central command agent counts the number of matched objects to ensure all objtcs in the targeted Master Frame are. In our architecture, production rules relating to distances distances are matched individually and concurrently, and it is necessary to also



Action, based on the number of objects detected compared with the target Master Frame				
Situation	**Is the current compass angle the same as in the Objective Master Frame?**	**Is this the first time that the compass angle matches, since the last situation change?**	**Are all distance elements the same as in the Objective Master Frame?**	**Action**
No objects detected	Yes	First time	-	Rotate to find some targeted objects
		Second time	-	Reverse to get a broader view
	No	~	-	Rotate to match the compass angle to the Objective Master Frame
Some of the target objects detected	Yes	First time	-	Resolve any distance differences to the detected objects and rotate to find the other objects
		Second time	Some distance elements are not the same	Resolve any differences in distance only
			All distance elements are the same	Reverse to get a broader view
	No	~	-	Resolve any distance differences to the detected objects and rotate to match compass angle to the Objective Master Frame
All targeted objects detected	-	-	-	Resolve any distance differences to the detected objects and rotate to match compass angle to the Objective Master Frame
Any additional objects detected	-	-	-	Ignore any additional objects and follow any action above for the targeted objects.

Resolving any differences in distance		
Situations	Longitudinal movements	Transversal movements
Current input Individual Object Distance is CLOSER than in the target Master Frame.	Forward	\
Current input Individual Object Distance is FARTHER than in the target Master Frame.	Backward	\
Current input Object Relative Distance is CLOSER than in the target Master Frame.	\	Toward the farther object side
Current input Object Relative Distance is FARTHER than in the target Master Frame.	\	Toward the closer object side

Notes:

- '~' means the situation does not apply.
- '-' means any differences in situation do not affect the actions to be taken.
- '\' means the movements not apply.

Figure 4.

match the total number of objects in frmae to avoid any false positives.

2.4.4 Compass Reference

It is known that honey bees possess a sensory system that is able to detect and remember the direction in which they are heading. In flight, they can yaw to match up their current heading with a remembered direction.

Our system includes a magnetic compass that is intended to reproduce this feature of honeybees. We use the number output by the compass (the angle w.r.to geographical North) as a symbolic indicator of direction, but do not perform any arithmetic processing on this number.

2.5 Motion

The motion activities depend on the mode of the system. While the system is in foraging mode, the motion command module generates random movement instructions to have a free exploration of the environment to discover the target object. When the system is in homing mode, all motion commands will be based on instructions from the central command agent.

Clearly, on a two wheeled robotic platform movement is confined to planar surfaces and we do not attempt to model the three dimensional freedom of movement that flying insects have. Therefore, movements only simulate three types of mobility actions, which are linear movements in two orthogonal directions and yaw rotations.

3 EXPERIMENT ENVIRONMENT

We were inspired by ethological experimental techniques (Chittka & Tautz 2003; Giurfa 2003; GIURFA et al. 1999; Horridge 2006; Srinivasan, Zhang & Lehrer 1998), which typically use shapes and colour symbols in an artificial environment to discover and verify the ability of honey bees to memorise route information. Therefore, we build a physical environment with a pure colour background and coloured plastic balls to represent landmarks, and, with the intention of real-life autonomous system application, a small two-wheeled robot serving as the experimenting artificial bee. Our target object is a simple yellow ball.

Our Java honeybee system runs on a single-board computer (SBC), the LattePanda. The LattePanda runs the Windows 10 operating system, which we have chosen for ease of portability of our Street simulation environment. Ultimately the Street simulation on the Windows platform will be replaced by a dedicated Street engine implemented in custom hardware.

Motion commands are transmitted to an on-board Arduino microcomputer, which controls two stepper motors through two motor-controllers, the A4899. A Pixy Cam5, which is a camera with in-built detection of coloured objects, is used as the visual sensor. It provides position, colour and size information about any objects detected. An LSM303 compass module provides information about the magnetic field. The device includes both an accelerometer and a magnetometer, but we don't use the accelerometer. Both input devices connect to a Micro Arduino Board, and inputs are provided to the LattePanda through a serial communication port.

During our experiments, the Honeybee System illustrated in Figure 1, including the Street simulators running the two agents, runs in a Java environment on the LattePanda SBC.

The experiment process starts with placing the honeybee simulating rover at an initial position, representing a hive, on our testing ground. Then, the rover starts with its foraging phase to find the target object, and as it does so the system records any landmark changes. Once the rover reaches the target, it starts the homing phase based on the experienced pathway to track back to the original position.

4 SYSTEM BEHAVIOUR

4.1 Foraging

In this phase, the mobility module can generate any uncoordinated motion instructions, which allow the rover to explore the surrounding environment until the target object is located serendipitously. During this phase of free discovery of the environment, a list of any objects encountered are recorded in sequence in the episodic memory of the Memory Agent, in the previously described.

Figure 3 shows an example of as sequence of Master Frames after a series of movements. The rover starts at a position, which is right next to a red ball, and then it reverses. On its way, two Master Frames are recorded. MF `_ID2` records that the rover is `close` to the red object, and `_ID3` records that the rover is `far` from the red ball. Our rover has only captured an image of a red ball so far, and each set of these stored elements are triggered by the change of perceived distance to that red object. Then, the rover rotates clockwise, and a new Master Frame, `_ID4`, records that two coloured blocks are viewed at the same time. Both of the objects are at a `far` distance from the rover, and the relative distance between them is `far`. After that, the rover moves forwards, and Master Frame `_ID5` is created when the rover sees only the green object, with a `far` distance.

A new Master Frame is created only if any visual information changes, such as changes of perceived distances and the number of detected objects. Putting it another way, the system is able to recognise changes in image properties during the movement, and it only creates a new Master Record when the situation changes.

When new elements are generated for a new Master Frame in the Working Memory of the Memory Agent and only when that it happening, an element containing magnetic information is also created. As the example in Figure 3 shows, the compass angles with `_ID4` and `_ID5` are different because `(_ID4 compass 237)` is created at the moment when the green ball just gets into the frame, and `(_ID5 compass 248)` is recorded when the edge of the red ball cannot be viewed any more.

4.2 Homing

4.2.1 Homing Strategy

This homing activity is designed to perform in much the same way as it has been observed to happen in the ethological experiments. This is to say, our rover is expected to track back to the starting point, the hive, by revisiting each recorded Master Frame, the landmarks, one after another.

Our homing mode is triggered after the target object is reached. The first step is to request the last Master Frame from the memory module as the first target. Then, the command agent compares this information with the current visual inputs. If any elements does not match, the command module generates motion activities to resolve any differs in distance.

Actions and instructions are generated as described in Figure 4. For example, in some cases, the rover does not detect in its current visual field any objects of the target Master Frame. In this case, the rover rotates to match its magnetic orientation to that recorded for the target Master Frame, and then generates commands to adjust the dirtance, once objects in the target MF are visualised. If no targeted objects can be detected even after matching its magnetic orientation to the MF, the rover continues to rotate. If it still does not find the target objects after a full circle of rotation,

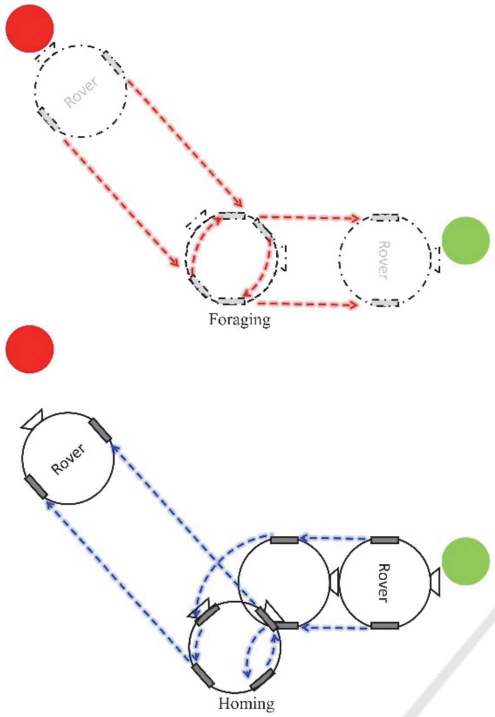


Figure 5.

the agent requests the rover to reverse to gain a broader view. In some circumstances, the agent instructs multiple movements, such as both longitudinal and transversal movements, when more than two elements are required to be changed in the current situation. After moving to a position such that the number of matched objects is same as the number in the target Master Frame, the command agent determines that this targeted Master Frame has been reached and requests the next one from the memory module. This process repeats until the memory agent detects that there are no more stored Master Frames, which means that the original position has been reached.

4.2.2 Test Method

In order to test homing behaviour, we use a manual foraging mode, in which we manually move the rover along a planned route in order to produce a predictable homing pathway. Once the rover has completed the planned route, we command the system to enter the homing mode and start to observe the rover's homing performance.

Our initial test has been with single object in the field of view but we have also tested multiple object views with linear movements only. We have found that if objects are at different distances from the rover the homing navigation movements are more accurate. This is because there are more references for distances along the same line.

In the next stage of testing, we allow the rover to rotate. With this additional freedom of mobility, the rover can resolve any magnetic orientation differences recorded between its current position and the target Master Frame, to find objects that are not in the current field of view.

4.2.3 Example: Linear Movements with Rotations

To give an illustration, Figure 5 shows an example of the process. We manually move the rover through a path such that there are six Master Frames captured and stacked. While the rover tracks back to the original position, it refers to each Master Frame one after another.

Firstly, the Memory Agent outputs all of the elements of the last Master Frame, as the first target, to the Command Agent. It immediately matches up that the characteristics of the target MF with its current view, which is a `nextTo` distance with a green ball and a same magnetic angle at the end of. The Command Agent determines t has been found and requests the next target MF. This causes the Command Agent to direct to the rover to reverse to make the distance to the green object `Close`. After that, the Command Agent receives the third recorded Master Frame, and the agent directs the rover to perform a reverse right turn to achieve a far view of the green ball with a different magnetically-determined orientation. This action occurs because, when we manually rotate the rover during the foraging stage, the system records a new Master Frame at the moment that the green ball just comes into the field of view, and it is not facing directly to the green ball yet. Therefore, on the way to reach this target position, a combined movement is performed to justify the magnetic orientation and increase in the linear distance to the green ball. The next target MF contains (`_ID3 RED is_far`) and another change in magnetic orientation, such that the rover faces the initial direction. Because it can't initially see the red ball the rover rotates towards the targeted orientation. During this anti-clockwise rotation, the red ball first appears in frame with a `far` distance, so no motion is required to adjust the distance. The next MF indicates that the red ball should be `close` so the Command Agent instructs the rover to move forward. It stops moving at the moment when the final Master Frame is matched and the Memory Agent indicates that that this is the end of the route. This homing trajectory route is shown in the lower part of Figure 5.

5 CONCLUSIONS

Our agent-based system shows behaviour with similar characteristics to the homing behaviour of bees, and it serves as a proof of concept for our hardware-based cognitive agent approach. Our system structure shows the feasibility of a hardware efficient system that is able to reproduce the cognitive behaviour of a simple biological system.

This research is extendable in terms of the density and complexity of the behaviour we reproduce. Insects, like honey bees, use many intelligent features, which have been studied and explained in some detail, such as the random foraging activity of ants (Traniello 1989). Our future work will involve modelling more advanced cognitive behaviours of creatures such as the ant, and developing the technology for implementing Street agents directly in low-power hardware.

In the longer term we aim to demonstrate that with the very high speed processing and high levels of parallelism available in custom microelectronic hardware, it will possible to use a Street-based architecture to implement significantly higher levels of cognitive behaviour than insect foraging.

REFERENCES

Chittka, L & Tautz, J 2003, 'The spectral input to honeybee visual odometry', *Journal of Experimental Biology*, vol. 206, no. 14, pp. 2393-2397.

Forgy, CL 1981, *OPS5 user's manual*, DTIC Document.

Frost, J, Numan, MW, Liebelt, M & Phillips, BJ 2015, 'A new computer for cognitive computing', *Cognitive Informatics & Cognitive Computing (ICCI* CC), 2015 IEEE 14th International Conference on,* IEEE, pp. 33-38.

Fry, SN & Wehner, R 2002, 'Honey bees store landmarks in an egocentric frame of reference', *Journal of Comparative Physiology A*, vol. 187, no. 12, pp. 1009-1016.

Gillner, S, Weiß, AM & Mallot, HA 2008, 'Visual homing in the absence of feature-based landmark information', *Cognition*, vol. 109, no. 1, pp. 105-122.

Giurfa, M 2003, 'Cognitive neuroethology: dissecting non-elemental learning in a honeybee brain', *Current opinion in neurobiology*, vol. 13, no. 6, pp. 726-735.

Giurfa, M 2007, 'Behavioral and neural analysis of associative learning in the honeybee: a taste from the magic well', *Journal of Comparative Physiology A*, vol. 193, no. 8, pp. 801-824.

GIURFA, M, HAMMER, M, STACH, S, STOLLHOFF, N, MÜLLER-DEISIG, N & MIZYRYCKI, C 1999, 'Pattern learning by honeybees: conditioning procedure and recognition strategy', *Animal behaviour*, vol. 57, no. 2, pp. 315-324.

Horridge, A 2006, 'Some labels that are recognized on landmarks by the honeybee (Apis mellifera)', *Journal of insect physiology*, vol. 52, no. 11, pp. 1254-1271.

Kiya, T, Kunieda, T & Kubo, T 2007, 'Increased neural activity of a mushroom body neuron subtype in the brains of forager honeybees', *PLoS One*, vol. 2, no. 4, p. e371.

Koch, C & Laurent, G 1999, 'Complexity and the nervous system', *Science*, vol. 284, no. 5411, pp. 96-98.

Labhart, T & Meyer, EP 2002, 'Neural mechanisms in insect navigation: polarization compass and odometer', *Current opinion in neurobiology*, vol. 12, no. 6, pp. 707-714.

Laird, JE 2012, *The Soar cognitive architecture*, MIT Press.

Menzel, R 2012, 'The honeybee as a model for understanding the basis of cognition', *Nature Reviews Neuroscience*, vol. 13, no. 11, pp. 758-768.

Menzel, R & Giurfa, M 2001, 'Cognitive architecture of a mini-brain: the honeybee', *Trends in cognitive sciences*, vol. 5, no. 2, pp. 62-71.

Menzel, R & Greggers, U 2015, 'The memory structure of navigation in honeybees', *Journal of Comparative Physiology A*, vol. 201, no. 6, pp. 547-561.

Menzel, R, Greggers, U, Smith, A, Berger, S, Brandt, R, Brunke, S, Bundrock, G, Hülse, S, Plümpe, T & Schaupp, F 2005, 'Honey bees navigate according to a map-like spatial memory', *Proceedings of the National Academy of Sciences of the United States of America*, vol. 102, no. 8, pp. 3040-3045.

Numan, MW, Frost, J, Phillips, BJ & Liebelt, M 2015, 'A Network-based Communication Platform for a Cognitive Computer', *AIC 2015*, p. 94.

Srinivasan, M, Zhang, S & Lehrer, M 1998, 'Honeybee navigation: odometry with monocular input', *Animal behaviour*, vol. 56, no. 5, pp. 1245-1259.

Traniello, JF 1989, 'Foraging strategies of ants', *Annual review of entomology*, vol. 34, no. 1, pp. 191-210.

Wajnberg, E, Acosta-Avalos, D, Alves, OC, de Oliveira, JF, Srygley, RB & Esquivel, DM 2010, 'Magnetoreception in eusocial insects: an update', *Journal of the Royal Society Interface*, vol. 7, no. Suppl 2, pp. S207-S225.