DESCRIPTION OF A SIMULATED ENVIRONMENT DEVELOPED FOR VALIDATING CFS AUTONOMY

Gerardo Fernandez-Lopez[1], Ruel Ellis[2] and Gerard Pounder[3*]

[1]Faculty of Engineering, Simón Bolivar University, Venezuela
[2,3]Faculty of Engineering, The University of the West Indies, Trinidad
[1]Email: gerardo.fernandez.lopez@gmail.com
[2]Email: ruel.ellis@sta.uwi.edu
[3]Email: pounderji@yahoo.co.uk *(Corresponding author)

Abstract: This paper describes the simulated environment that was prepared for validation of Cognitive Function Synthesis, or CFS, Autonomy. The environment was developed with the V-REP simulator and assembled around the 'Eric Rohmer' Pioneer robot. The simulated robot was configured to act as a Behaviour-Based Robot that is constrained to move about the test area, as defined by a state diagram, and based on pre-configured reflexes, provided by the Braitenberg algorithm. Vision sensors, i.e. simulated cameras, were found to be better suited to configure these behaviours than light sensor cues. Initial results suggest that the simulated environment may be used for CFS autonomy validation. This research marks the first simulated environment produced for investigating CFS Autonomy and consequently needs to be further validated.

Keywords: *Cognitive Function Synthesis, Artificial General Intelligence, Associative Memories, Autonomous Navigation, Biomimetic Navigation*

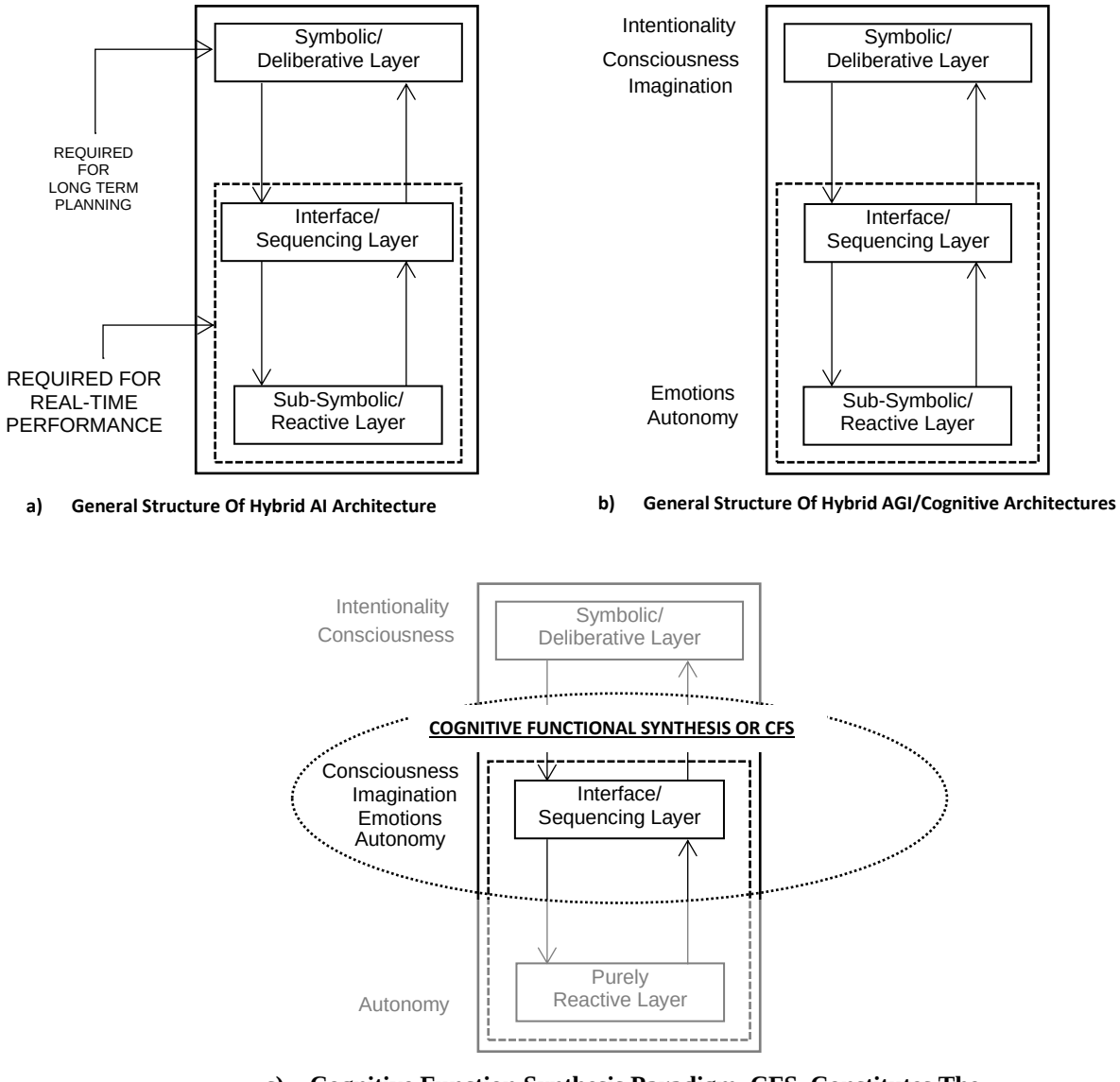
https://doi.org/10.47412/YTRH2545

1. Introduction

Pounder, Ellis, Fernandez-Lopez [1] introduced the Cognitive Function Synthesis (CFS) conceptual framework to artificial general intelligence. CFS postulates that at the "core" of intelligence in hybrid architectures, "interdependent" cognitive functions are synthesised through the interaction of various associative memory (AM)-based systems. The authors posit that this synthesis could form an interface layer between deliberative/symbolic and reactive/sub-symbolic layers in hybrid cognitive architectures. Much research is needed to determine the best way to successfully integrate CFS into each of the aforementioned layers and thus validate the framework. *Interfacing CFS with the reactive/sub-symbolic layer was determined to be the logical first-step.* The approach used was: determine the impact of CFS on the navigation of a simple Behaviour-Based Robot; and compare performance at a specified benchmark task with that of a Reactive/Behaviour-Based architecture. *This paper describes the simulated environment developed and the benchmark task used to investigate CFS navigation with a Behaviour-Based robot.* Environment validation was dealt with subsequently. The paper is structured as follows. Section 2 gives a brief overview of CFS within the field of Artificial Intelligence, or AI. The criteria used during the selection process of candidate robot platforms and simulation software are captured in Section 3. Section 4 describes the robot platform selected, the state diagram used to configure the Pioneer robot as a Behaviour-Based robot as well as some initial results and concludes with Section 5.

2. Cognitive Function Synthesis in the context of AI

AI typically decomposes intelligence into distinct properties or functions. These functions are developed into independently subsisting autonomous modules, which are finally integrated through interface mechanisms to form Multi-Agent Systems or MAS [2, 3]. This approach suggests that intelligence arises from the integration of a collection of independently subsisting modules and can be termed the Traditional AI Paradigm. This paradigm has given rise to Symbolic/Deliberative, Sub-Symbolic/Reactive and Hybrid AI architectures [4-6]. Nilsson [7] suggests that Hybrid architectures are Traditional AI's most likely candidates for achieving more sophisticated intelligence. Such architectures require an interface/ sequencing layer to integrate a Deliberative/Symbolic layer – for long-term planning – with a Reactive/Sub-Symbolic layer – for real-time performance [3, 4] – cf. Fig. 1a).



a) General Structure Of Hybrid AI Architecture

b) General Structure Of Hybrid AGI/Cognitive Architectures

c) Cognitive Function Synthesis Paradigm, CFS, Constitutes The Interface/Sequencing Layer For Hybrid Architectures.

Figure 34: The Introduction Of Cognitive Functions Into A Generic Hybrid Architecture

However, Pounder, Ellis, Fernandez-Lopez [1] note that, as a consequence of the Traditional AI Paradigm, integration challenges arise at all levels of AI, in general and in Artificial General Intelligence, or AGI, Hybrids in particular. In the latter architectures, cognitive functionality is introduced as independently subsisting cognitive modules at the symbolic and sub-symbolic levels, cf. Fig. 1b, but are usually subordinated to other intelligence modules. The authors claim that this may be the result of inadequate decomposition of intelligence and introduce the Cognitive Function Synthesis, or CFS, paradigm. CFS posits that cognitive functions might be implemented as 'interdependent' functions via Associative Memories, or AMs, to form an interface layer for hybrid cognitive architectures (c.f. Fig. 1c).

In light of the above, the CFS Conceptual Framework in Fig. 2 was developed from which the cognitive functions of: i) Perceptual Learning or Autonomous Extraction; ii) Consciousness; iii) Imagination; iv) Emotions; and v) Autonomous Action emerge. Pounder, Ellis, Fernandez-Lopez [1] performed experiments to investigate CFS display of functions i) to iii). Preliminary results suggest that the emergent properties, i.e. learned reflexes akin to those described by Vernon, Metta, Sandini [8] that arise on the basis of pointer-chain sequences, could prove useful to robot autonomy. *Interfacing CFS with the reactive/sub-symbolic layer was determined necessary to establish plausibility with respect to CFS Autonomy*. Whether CFS possesses the requisite characteristics of Deep Learning systems is yet to be investigated [9-11].

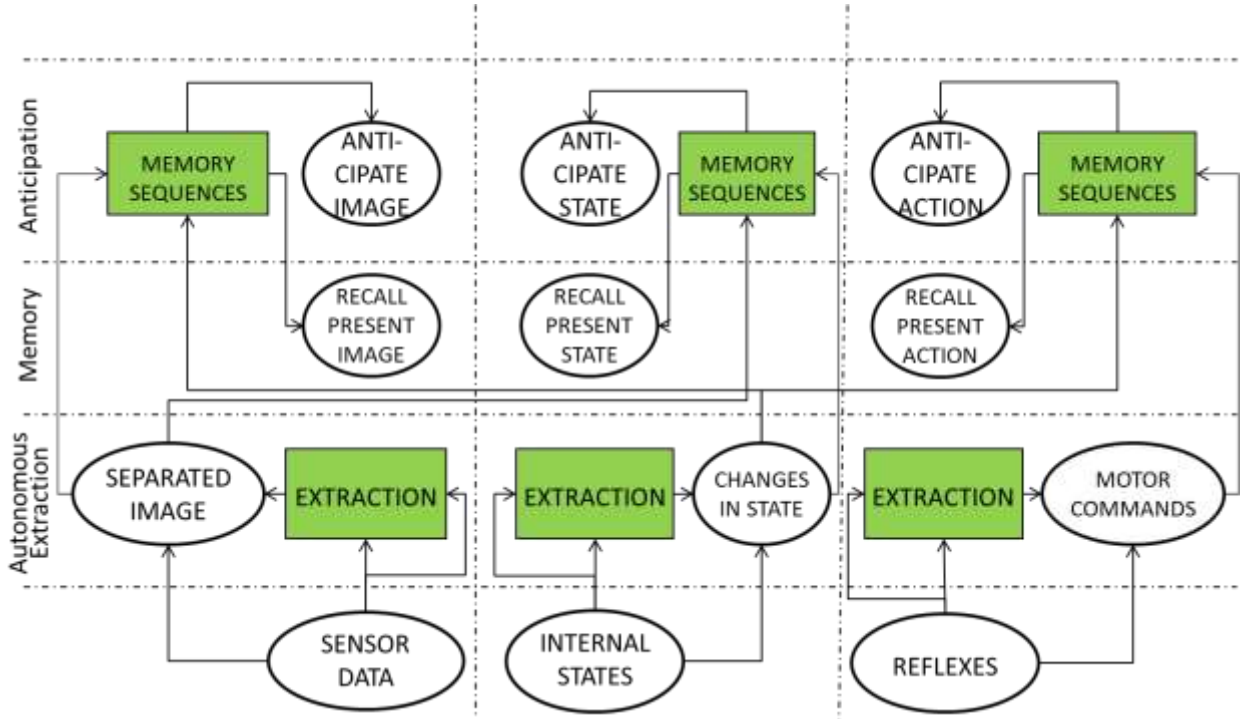


Figure 35: Cognitive Function Synthesis

3. Robot Platforms and Simulated Environments Considered

Prior surveys were used to identify candidate platforms on which to experiment with CFS autonomy. The survey by Hilal, Wagdy, Khamis [12] focused on commercially available robots suitable for use in tertiary education institutions. Kits were grouped into: Building Body Kits; Electronic Components Kits; Software Kits; Programmable Robots; and Complete Starter Kits. Subsequently, Ruzzenente, Koo, Nielsen [13] used a similar approach, but sub-divided non-versatile kits into: Manipulators; Household Robots; Robotic Aircrafts; Humanoid Robots. They limited their review to robots used in education while focussing on the

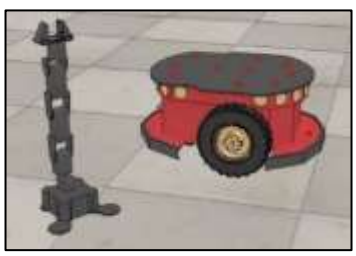
ease of reuse in the teaching of different subject areas such as electronics, programming or human-robot interaction.

The Robots considered included the: Pioneer Robot 3DX by Adept Mobilerobots; Khepera III and Hemisson from K-Team; Roomba Create by iRobot; MiaBot robots by Merlin Systems; Rovio by WowWee; and E-Puck by École Polytechnique Fédérale de Lausanne (EPFL); Boe-Bot; Stingray Robot; LEGO Mindstorm; VEX; FischerTechnik; and Qfix. Reviews by Mondada, Bonani, Raemy [14] and Junior, Neto, Hernandez [15] also suggested consideration of Robotino by Festo. More recent surveys, by Gyebi, Hanheide, Cielniak [16] and Takacs, Eigner, Kovács [17], identified additional robot platforms not considered in the initial analysis. These surveys were primarily conducted with a view to identify robot platforms useful as educational tools for Science, Technology, Engineering and Mathematics support, or STEM. These considerations did not substantially affect the decision to use the Pioneer robot platform for experiments in a simulated environment.

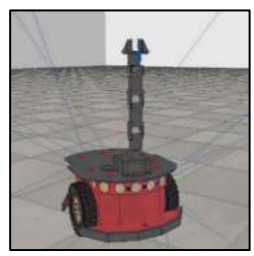
The Microsoft Robotics Developer Studio 4 and the V-REP 3.3.2 (rev 3) simulation packages were considered. V-REP was found to be superior and was installed on a Lenovo P50 64-bit laptop with an Intel Core i7-6820HQ, and a 2.7 GHz processor with 32GB RAM. Code written in C++ with Microsoft Visual Studio 10 was used to run the simulator. The robot simulator was configured to resemble, as much as possible, the environment described in experiments conducted with Darwin VII [18].

4. Description Of Platform And Behaviour-Based Configuration

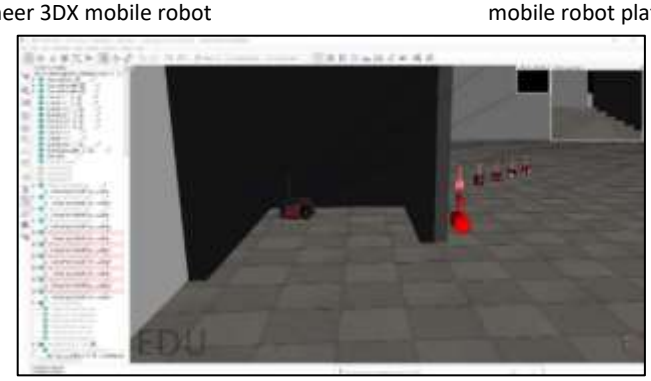
The V-Rep supplied 'Eric Rohmer' model of Pioneer came pre-configured: wheel joint in force/torque mode; motor enabled; and position control disabled. The Pioneer was configured biomimetically to act as a Behaviour-Based Robot. This was achieved through use of: the Braitenberg Algorithm in conjunction with the ultrasound sensors for reflexes; and a cube-targeting reflex in conjunction with vision sensors.



a) PhantomX Pincher manipulator (left) with Pioneer 3DX mobile robot



b) PhantomX Pincher fitted to Pioneer 3DX mobile robot platform with vision



c) Composite Robot Platform In Simulated Environment.

Figure 36: Screenshots of Simulated Composite Robot Platform

Figure 3a depicts a V-Rep simulator screenshot illustrating one such mobile platform to the right of a PhantomX Pincher manipulator. In these experiments, the manipulator was mounted onto the mobile platform as illustrated in Fig. 3b. Six vision sensors were also mounted onto the composite platform. Three were mounted to the front and two to the back of the mobile platform, while one was affixed onto the manipulator. During these experiments, only the main middle-front vision sensor on the mobile platform and the one on the manipulator were used, the latter effectively being utilized as a proximity sensor. Figure 3c shows the composite robot in the simulated environment. To ensure basic biomimetic reflexes, the sensory data from both the ultrasound and vision sensors were used in a Braitenberg sense.

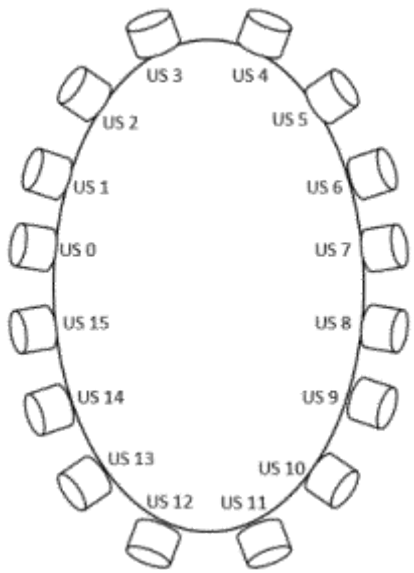


Figure 37: Ultrasound Sensor Configuration.

US (Ultrasound Sensor Readings)
= {US0, US1, US2, US3, US4, US5, …, US9, US10, US11, US12, US13, US14, US15} (13)

Left Motor Coefficient (8 Sensors)
= {-0.2, -0.4, -0.6, -0.8, -1.0, -1.2, -1.4, -1.6, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0} (14)

Right Motor Coefficient (8 Sensors)
= {-1.6, -1.4, -1.2, -1.0, -0.8, -0.6, -0.4, -0.2, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0} (15)

Left Motor Coefficient (16 Sensors)
= {-0.2, -0.4, -0.6, -0.8, -1.0, -1.2, -1.4, -1.6, 0.2, 0.4, 0.6, 0.8, 1.0, 1.2, 1.4, 1.6} (16)

Right Motor Coefficient (16 Sensors)
= {-1.6, -1.4, -1.2, -1.0, -0.8, -0.6, -0.4, -0.2, 1.6, 1.4, 1.2, 1.0, 0.8, 0.6, 0.4, 0.2} (17)

Ultrasound, or US, Sensor Data from the 16 sensors around the periphery of the Pioneer platform, c.f. Fig. 4, is given by Eq. (1). This permitted an "8 sensor" US configuration, given by Eqs. (2) and (3) and a "16 sensor" US configuration, given by Eqs. (4) and (5). Now, the Braitenberg Obstacle Avoidance algorithm, readily available with the 'Eric Rohmer' sample model of the Pioneer robot in the V-Rep simulator, used Eqs. (6) to (9) with: sensor data from the front "8 sensors" i.e. Eqs. (2) and (3); and coefficients from the 'Obstacle Avoidance' column of Table 1. However, the simulated Pioneer robot provided was not equipped

with either bumpers or proximity sensors on which to base a 'Contact/Collision' reflex system that would be engaged when the robot was in close proximity to enclosure walls/borders. Additionally, the Pioneer robot, being a differential drive platform, would experience constrained manoeuvrability when making direct contact with the walls of the enclosure, often limiting robot ability to extricate itself from 'tight spots'. Being the most essential preconfigured motor primitive for the robot platform, a 'Contact/Collision' reflex system was obtained by adjusting the coefficients of the Braitenberg Obstacle Avoidance Algorithm, which came originally with the Pioneer simulation, to that specified in the 'Contact/Collision' column of Table 1. Initial trials using the 'Contact/Collision' reflex system with the "8 sensor" US configuration produced a 'sluggish' response that made the robot seem insensitive to its proximity to the enclosure, from an observer's perspective, and was due to insufficient sensor coverage of the rear of the platform. Consequently, as outlined later in this paper, consideration was then given to the "16 sensor" US reflex-configuration.

Table 1: Parameters for Detection Value used in Braitenberg algorithm

	Obstacle Avoidance	Contact/Collision
No Detect Limit (No Detection Limit)	0.5	0.2
Max Limit (Maximum Detection Limit)	0.2	0.1

$$\text{Braitenberg (US)}_{\text{Left Motor}} = \sum_{k=0}^{\text{Num Sensors}} \text{Left Motor Coefficient}_k \times \text{Detection}_k \tag{18}$$

$$\text{Braitenberg (US)}_{\text{Right Motor}} = \sum_{k=0}^{\text{Num Sensors}} \text{Right Motor Coefficient}_k \times \text{Detection}_k \tag{19}$$

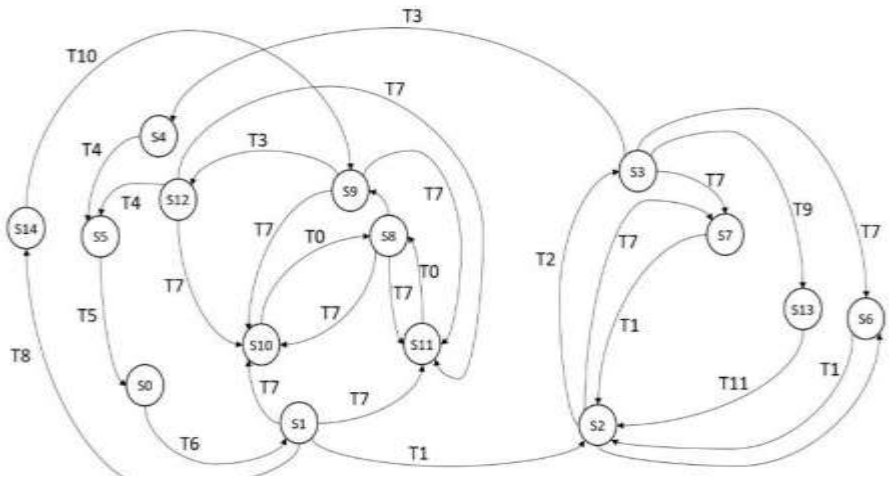
$$\text{Detection}_k = (1 - \text{Detection Factor}_k) \tag{20}$$

$$\text{Detection Factor } k = \begin{cases} \dfrac{\text{US}_k - \text{Max Limit}}{\text{No Detect Limit} - \text{Max Limit}} & \text{Max Limit} < \text{Sensor Value} < \text{No Detect Limit} \\ 0 & \text{Sensor Value} < \text{Max Limit} \text{ or } \text{Sensor Value} > \text{No Detect Limit} \end{cases} \tag{21}$$

Early attempts to configure robot behaviour based on light sensors were abandoned due to difficulty in using diffused light as cues. The images captured from the main vision sensor were subjected to segmentation such that dark-ranged colours and a bright range of colours were used to develop cues in a Biomimetic sense. The state diagram governing the platform's 'Reactive' behaviour is presented in Fig. 5. In states 2, 4, 8 and 12, the vision sensor output was partitioned such that the number of dark-ranged pixels on the left side of the image determined the output of the right motor and vice-versa as follows. In the "go to dark" states 2 and 8 the sum of dark coloured pixels just to the left and right of the centreline of the image was used to determine output of the right and left motor respectively. In the "dark on periphery" states 4 and 12 the sum of dark coloured pixels on the extreme left and extreme right of the image centre was used to determine output of the right and left motor respectively.

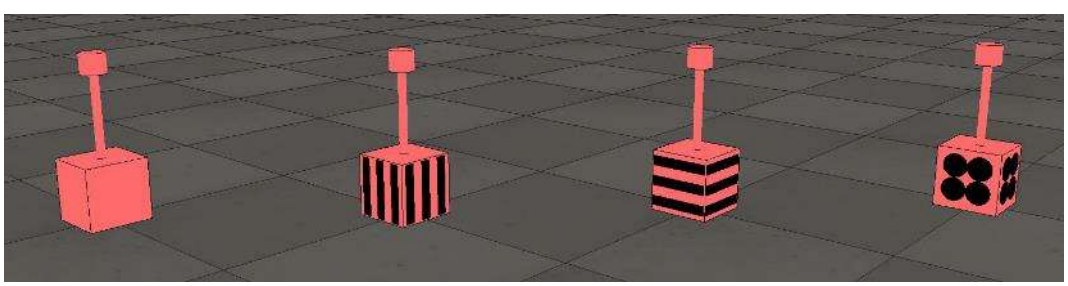
Similarly, in Roaming State 1, if a cube were within the robot's Field of Vision, FOV, the number of bright-ranged pixels on the right side of the image determined the output of the left motor and vice-versa. This

allowed the robot to move toward the cube while the manipulator would descend to a standby position to grab the cube. When the cube is within the pincers of the manipulator, the vision-sensor based proximity switch triggers it to enclose the cube and place it on the platform's back. Cubes may be: Vertically-striped; Horizontally-striped; dotted or without a pattern, c.f. Fig. 6. In the absence of a cube, in state 1, both motors are equally powered pushing the robot forward. States 6, 7, 10 and 11 ensured that



STATES		TRANSITIONS	
S1	Roaming	T0	No Cube
S2 & S8	Go Towards Dark Area	T1	Grab(bed) Cube
S3 & S9	Turn Away From Very Close Wall	T2	Facing Walls Of Den
S4 & S12	Seeks Darkness On Periphery Of FOV	T3	Bright On Periphery of FOV
S5	Stop In Comfort Zone	T4	Darkness On Periphery Of FOV
S0	Rest In Comfort Zone	T5	Zero Robot Arm
S6 & S10	Turn Anti-clockwise When Outside	T6	Counter Flag 1
S7 & S11	Turn Clockwise When Outside	T7	Flag Indicating Exit From Enclosure
S13	Transition From S3 To S2	T8	Transition Flag (Object Already Dropped)
S14	Transition From S1 To S9	T9	Transition Flag (Close To Exit)
		T10	Counter Flag 2
		T11	Counter Flag 3

Figure 38: State Diagram Describing Preconfigured Reflexes For BBR



a) Blank b) Vertical Stripes c) Horizontal Stripes d) Dots

Figure 39: Patterns on cubes used in this experiment.

the robot never ventured too far beyond the enclosed area. Once beyond the extremities of the enclosure, a pre-set value is applied to one of the motors to ensure that it turned to face the entrance to the enclosure. State 5 and 0 were used to facilitate the manipulator's return to its default position and remain immobile for a 10-cycle rest period. In states 3 and 9, on the robot's return to the recesses of the enclosure, the motors are powered equally to provide a constant speed to the robot. Once the 'Obstacle Avoidance' algorithm is activated, depending on its effectiveness, the robot should navigate within its environment without coming into contact with the enclosure walls. In the absence of this algorithm, the 'Contact/Collision' reflex system would enable the platform to operate autonomously, despite contact with enclosure.

Note that the Braitenberg Algorithm has a tendency to eventually get stuck in local minima. Two measures were used to resolve this problem. In the first instance, should the robot grind to a near halt, a 'shock' treatment in which an instantaneous boost of high power activating the wheel motors in opposing senses is applied to disengage the robot. Additionally, states 13 and 14 were introduced to transition the robot out of states 3 and 1, respectively, should they enter into these minima. Once navigation autonomy can be achieved, performance measures could then be used to compare Obstacle Avoidance algorithms.

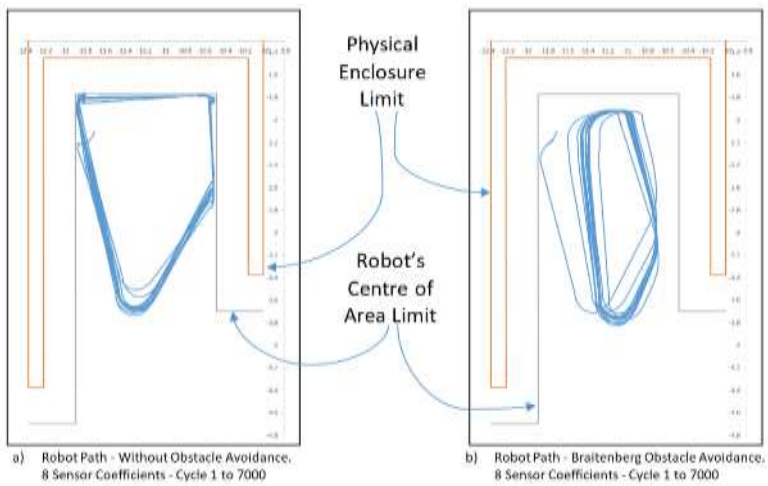


Figure 40: Robot Path a) Without Obstacle Avoidance (i.e. Basic Reflexes Only) and b) with Braitenberg Obstacle Avoidance

Figure 7a shows sample results illustrating the robot's path when operated: in 'Contact/Collision' reflex mode only, i.e. without Obstacle Avoidance. This demonstrates the platform's navigation autonomy in the absence of the Obstacle Avoidance behaviour. Notice the 'sluggish' "8 sensor"-configuration response that prompted consideration of employing the "16 sensor" US configuration. Figure 7b shows a sample result with Braitenberg 'Obstacle Avoidance'. *These preliminary results suggest that the simulated environment has potential for being used as a tool for comparing Obstacle Avoidance algorithms in robot navigation.* An obvious gauge to measure the success of an Obstacle Avoidance algorithm is to equate lower frequency of contact/collision with improved performance. Nevertheless, other metrics may also be used.

One of CFS's claims is its ability to learn to avoid obstacles without being equipped at manufacture with a readily designed behaviour to this end. As such, validation requires comparison of CFS performance against platforms operating with and without Obstacle Avoidance algorithms. *A simulated environment operating autonomously in all scenarios, such as the one presented here, is ideal for this purpose.* However, to be used in this manner, it needs to be established whether the difference in observed performance in Fig. 7 was statistically significant. This will be considered in a subsequent paper.

5. Conclusions

After an initial summary of the CFS framework, the criteria followed in the selection of a robot platform and simulation software was presented. The V-Rep simulator was chosen and the main features of the simulated environment, which was developed for validation experiments with CFS autonomy and assembled around the 'Eric Rohmer' Pioneer robot, were described.

In the process of configuring the robot to navigate autonomously in the simulated environment as a Behaviour-Based robot, the use of diffused light to trigger light sensors to effectively coordinate behaviours was found to be challenging. Instead, vision sensors, or virtual cameras, were used in conjunction with ultrasound sensors, and the data captured was subjected to the Braitenberg algorithm. Reflex behaviours designed included: 'Contact/Collision' reflex; Return to enclosure reflex; Go toward target reflex; and 'Obstacle Avoidance' reflex. A state diagram was developed to map the robot's movement about the test area based on these pre-configured reflexes.

Initial results suggest that the simulated environment may be useful as a tool for comparing robot navigation performance with respect to Obstacle Avoidance algorithms. As one of the claims of CFS is its ability to learn to navigate, i.e. avoid obstacles, without being specifically equipped with a predesigned Obstacle Avoidance behaviour, it is hoped that the simulated environment presented here may be used to investigate and validate this aspect of CFS functionality. Consequently, validation of the simulated environment is paramount. Therefore, the robot's Obstacle Avoidance performance, with and without the algorithm, needs to be compared to determine whether these differences are statistically significant.

References

[1] G. A. J. Pounder, R. L. A. Ellis, G. Fernandez-Lopez. Cognitive Function Synthesis: Preliminary Results. *Kybernetes* 46 no. 2, (2017) 272-290.

[2] F. Klügl. Beyond Distributed Artificial Intelligence. *KI - Künstliche Intelligenz* 28 no. 3, (2014) 139-141.

[3] L. Leahu, P. Sengers, M. Mateas. 2008. Interactionist AI and the promise of ubicomp, or, how to put your box in the world without putting the world in your box. In *Proceedings of the 10th International Conference on Ubiquitous Computing* (pp. 134-143). ACM.

[4] R. C. Arkin, 1998. *Behavior-Based Robotics*. 1998: MIT Press,.

[5] R. A. Brooks. A robust layered control system for a mobile robot. *IEEE Journal of Robotics and Automation* 2 no. 1, (1986) 14-23.

[6] N. J. Nilsson. 1969. A mobile automaton: an application of artificial intelligence techniques. In *First International Joint Conference on Artificial Intelligence* (pp. 509-520).

[7] N. J. Nilsson, 2007. The physical symbol system hypothesis: status and prospects. 2007, Springer: Berlin, Heidelberg. (pp. 9-17).

[8] D. Vernon, G. Metta, G. Sandini. 2007. A Survey Of Artificial Cognitive Systems: Implications For The Autonomous Development Of Mental Capabilities In Computational Agents. In *Evolutionary Computation, IEEE Transactions On* (pp. 151-180). IEEE.

[9] I. Arel, D. C. Rose, T. P. Karnowski. Deep machine learning-a new frontier in artificial intelligence research. *IEEE computational intelligence magazine* 5 no. 4, (2010) 13-18.

[10] Y. LeCun, Y. Bengio, G. Hinton. Deep learning. *Nature* 521 no. 7553, (2015) 436.

[11] O. Walker, F. Vanegas, F. Gonzalez, S. Koenig. 2019. A Deep Reinforcement Learning Framework for UAV Navigation in Indoor Environments. In *2019 IEEE Aerospace Conference* (pp. 1-14). IEEE.

[12] A. R. Hilal, K. M. Wagdy, A. M. Khamis. 2007. A Survey on Commercial Starter Kits for Building Real Robots. In *Proceedings of the International Conference on Electrical Engineering* (pp.

[13] M. Ruzzenente, M. Koo, K. Nielsen, L. Grespan, P. Fiorini. 2012. A review of robotics kits for tertiary education. In *Proceedings of International Workshop Teaching Robotics Teaching with Robotics: Integrating Robotics in School Curriculum* (pp. 153-162).

[14] F. Mondada, M. Bonani, X. Raemy, J. Pugh, C. Cianci, A. Klaptocz, S. Magnenat, J.-C. Zufferey, D. Floreano, A. Martinoli. 2009. The e-puck, a robot designed for education in engineering. In *Proceedings of the 9th conference on autonomous robot systems and competitions* (pp. 59-65).

[15] L. A. Junior, O. T. Neto, M. F. Hernandez, P. S. Martins, L. L. Roger, F. A. Guerra. A Low-Cost and Simple Arduino-Based Educational Robotics Kit. *Journal of Selected Areas in Robotics and Control (JSRC)* 3 no. 12, (2013) 1-7.

[16] E. Gyebi, M. Hanheide, G. Cielniak. 2015. Affordable mobile robotic platforms for teaching computer science at African universities. In *Proceedings on the 6th International Conference on Robotics in Education* (pp. 44-49).

[17] A. Takacs, G. Eigner, L. Kovács, I. J. Rudas, T. Haidegger. Teacher's Kit: Development, Usability, and Communities of Modular Robotic Kits for Classroom Education. *IEEE Robotics & Automation Magazine* 23 no. 2, (2016) 30-39.

[18] J. L. Krichmar, J. A. Snook. 2002. A neural approach to adaptive behavior and multi-sensor action selection in a mobile device. In *Robotics and Automation, 2002. Proceedings. ICRA'02. IEEE International Conference on* (pp. 3864-3869). IEEE.