Intelligent Robotics Coursework Report by Group 6

Baudouin Belpaire
Msc Robotics
H00445613

Luc Naoki Pichot
Msc Robotics
H00445072

Princy Mariam Jacob
Msc Robotics
H00450410

***Abstract*—This study utilises Behaviour-Based Robotics (BBR) and Evolutionary Robotics (ER) and is examined in the context of a Mars Rover simulation employing the e-puck robot. The research delineates a comparative analysis between the direct, sensor-driven responses characteristic of BBR and ER's complex, adaptive algorithms underpinned by Genetic Algorithms (GAs) and neural networks. The focus lies on the Rover's mission objectives, primarily navigation and soil sample collection, facilitated by a ground-based beacon system. Experimental results demonstrate BBR's effective execution of tasks, leveraging its immediate sensor-response capabilities. In contrast, ER shows a trajectory of performance enhancement, marked by a progressive learning curve, although with initial unpredictability. The study suggests potential improvements in ER through adaptive mutation and crossover in GAs and emphasizes the significance of hyperparameter tuning for neural network optimization. Comparative analysis between BBR, ER, and Proximal Policy Optimisation (PPO), the report offers insightful perspectives on their respective methodologies and advantages. This comparative study provides a foundation for future advancements in robotic control systems, particularly for challenging environments like Mars, offering a roadmap for developing more sophisticated, adaptive, and efficient robotic navigational systems.**

***Index Terms*—Robotic navigation, Mars Rover simulation, e-puck robot, Webots simulator, Behavioural-based Robotics, Evolutionary Robotics, Genetic Algorithm**

I. Introduction

Intelligent robotic controllers are essential for addressing diverse practical challenges in the physical environment. These controllers have experienced notable improvements in their processing and maritime capabilities. Several novel approaches have been utilized in the development of robotic controllers. Initial iterations were deficient in their ability to engage with and adjust to their surroundings. The advent of Behaviour Based Robotics (BBR) revolutionized this, allowing robots to dynamically adjust their behaviour by leveraging environmental feedback through a sensor-actuator control mechanism. Nevertheless, BBR may encounter difficulties in managing the intricacy of several sensor inputs, which is a frequent situation in complex real-world circumstances[1].

The discipline has witnessed the rise of Evolutionary Robotics (ER) to address these constraints. This methodology, derived from the tenets of natural evolution, centres on advancing and enhancing control systems through multiple iterations referred to as 'generations'[2]. It adapts robot control to unfamiliar surroundings through the utilization of sophisticated algorithms[3][4]. Although BBR is generally easy to use, it may not be suitable for complex situations due to its restricted flexibility. On the other hand, Evolutionary Robotics (ER) necessitates substantial processing resources and can exhibit unpredictability in its initial stages. However, it tends to enhance its performance with time, positioning it as a formidable candidate for real-world applications.

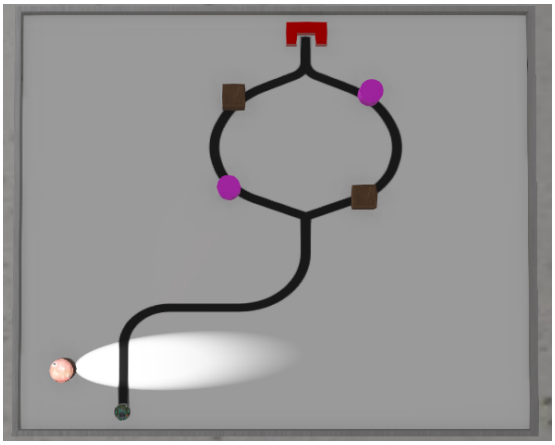

Fig. 1. Route A with Beacon light activated

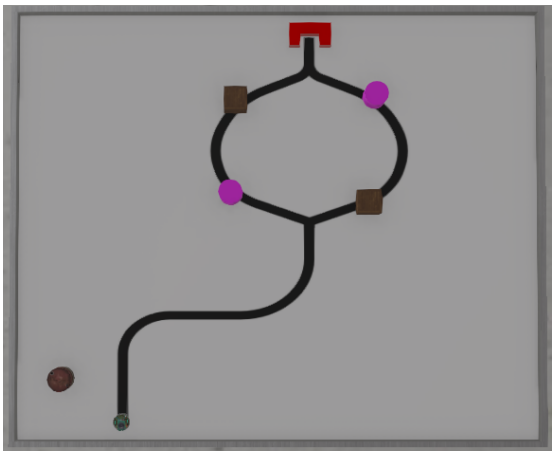

Fig. 2. Route B with Beacon light deactivated

Let's consider the development of a streamlined testing

environment for a new Mars rover navigation system. The robot should follow the line and navigate around any obstacles it encounters. It must accomplish this task efficiently. The rover's path options are marked on the ground of the testing arena. Two distinct routes, named Route A on the left part (refer figure [1]) and Route B on the right one (refer figure [2]), the reward zone. The rover's path selection depends on the status of a ground-based beacon. When the beacon is lit (activated), the rover follows Route A; if the beacon is unlit (deactivated), Route B is the correct path.

This study investigates the application of both BBR and ER in managing an e-puck robot created using Webots software within a simulated environment. It explores the straightforward nature of behavioural algorithms against evolutionary algorithms' nuanced, progressive learning curve. This comparative analysis scrutinizes the robot's performance under these varying algorithms and aims to benchmark these findings against other potential methodologies, setting the stage for future enhancements in robotic control.

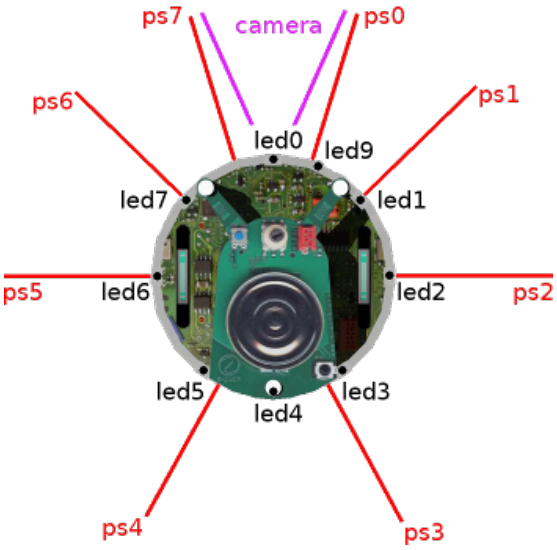

Fig. 3. E-puck robot with positions of camera and sensors[5]

In the experimental setup involving an e-puck robot (as in figure [3]), data acquisition and preprocessing play a critical role. The sensory inputs from the robot are systematically normalized and then aggregated into a comprehensive array for further analysis. This array is structured to categorize the sensory data based on their types and positions. Specifically, the array's initial three indices (0-2) are allocated for the readings from the ground sensors. Subsequently, indices 3 to 9 are designated for the data from the front proximity sensors, namely ps0, ps1, ps2, ps5, ps6, and ps7. The final index in the array (index 10) is reserved for the measurements obtained from the light sensor. This methodical organization of sensor data is essential for facilitating efficient data processing and analysis in robotic operations[5].

II. Task 1 using Behaviour-Based Robotics (BBR)

A. Overview

Behaviour-based robotics (BBR) is a methodology that uses concurrent processes to combine inputs from sensors and other behaviours in order to produce actions for a robot's actuators or new behaviours. The ultimate aim of BBR is to accomplish particular goals. This method is efficient for self-governing robots operating in ever-changing and uncertain surroundings. BBR exhibits a distinctive feature of having multiple modules spread out and working together to achieve overall system behaviour. BBR research primarily examines the emergent functionality of interactions between robots and their environment. It specifically emphasizes the coordination of opposing or orthogonal behaviours.[1][6]

Coordination methods in BBR exhibit a range of approaches, with competitive strategies emphasizing behaviours influenced by a network or action-selection mechanism and cooperative strategies combining outputs from many behaviours to reach a compromise. In addition, the paper examines coordinating approaches that prioritize higher-priority behaviours. The usefulness and versatility of BBR are demonstrated by its utilization in several types of robotic systems, including mobile robots, submersible vehicles, and humanoids.[1]

The guidebook of behavioural-based systems [1] offers an up-to-date summary of the latest advancements in BBR. A thorough comprehension of BBR is essential for the Mars Rover Robotics coursework. It guides the creation of intelligent, adaptable, and efficient behaviour-based control systems for the simulated Mars Rover. This ensures successful navigation and task execution in the Martian environment.

B. Design and Implementation Rationale

For Behaviour-Based Robotics (BBR), at the start of each simulation, the beacon flag is initially set to zero. The state of this condition may be updated depending on the data obtained from a light sensor. More specifically, when the reading from the sensor exceeds a predetermined threshold, the beacon flag is modified to a value of one, indicating the activation of the beacon. This system is essential for the robot's decision-making process at crossings. When the robot reaches the end of a queue, it is programmed to choose a direction - either turning right or left - based on the value of the beacon flag. When the flag is set to one, the robot is instructed to do a left turn. This instruction is consistently followed at two important points in the simulation[7].

The robot's aptitude in line following is evaluated using a systematic technique, where it is directed to keep a direct path when all three ground sensors detect a black surface. If the left ground sensor detects any deviation, the robot corrects its trajectory by making a little right turn to align itself with the desired path.

An essential component of this study is the robot's reactive mechanism to impediments. When an obstacle appears, a flag is raised to indicate the presence of the obstruction. This flag prompts the robot to temporarily stop using ground sensor data and start following a procedure to avoid the obstacle. The main driving force behind this approach is the data obtained from six proximity sensors. The reaction approach entails identifying the sensor with the most elevated reading and subsequently adapting the robot's manoeuvre accordingly. If the sensor with the greatest reading is located on the left side, the robot is designed to manoeuvre around the obstruction by going to the right side. The avoidance approach consists of two distinct responses: a clockwise rotation if prompted by the front left sensor, and a forward movement with a rightward shift if initiated by the side left sensor.

The successful navigation past an obstacle is decided by the ground sensors' ability to detect changes in data, namely the shift from white to black. After completing a successful journey around an obstacle, the flag indicating the presence of obstacles is set back to zero, ensuring that the robot is ready for future navigation difficulties.

Although not yet incorporated in the existing framework, a potential approach to indicate the completion of the robot's journey entails utilising elevated signals from all six proximity sensors, together with time measurement to evaluate the duration of the operation precisely.

This study highlights the significance of using sophisticated supervisory controls, along with adaptive decision-making algorithms and sensor-driven reactions, to improve a robot's ability to navigate and operate efficiently in intricate, simulated environments.

C. Results and Analysis

The Behavioral-Based Robot (BBR) work objectives were successfully achieved, with the preset endpoint reached within 5 minutes. The robot successfully travelled its path by accurately perceiving the best way through its effective response to the bright guiding light and skillfully avoiding any impediments it met. The data obtained from several sensors has been subjected to a normalisation process to improve its interpretability. This process has resulted in values that range from 0 to 1. This normalisation enables a more precise and organised analytical data display.

The light sensor data plot in Figure 4 reveals the robot's interaction with a light source at the beginning of its route, indicating a preference for path A (placed on the left). Figure 5 illustrates the findings obtained from the proximity sensor data, showing clear peaks corresponding to the barriers faced. The achievement of the ultimate objective is signalled by a prominent horizontal line in the data, occurring around 400 units of time on the index. Figure 6 combines information from ground sensors and wheel velocities. Analysing the middle ground sensor (shown by the green line) is crucial for detecting deviations from the desired course. Occurrences of the robot deviating from its intended path are particularly noticeable when it is trying to avoid obstacles. The robot's wheel movements provide insight into how it reacts to obstacles. Wheel values below 0.4 indicate significant and abrupt turns, especially at intersections and when manoeuvring around obstacles.

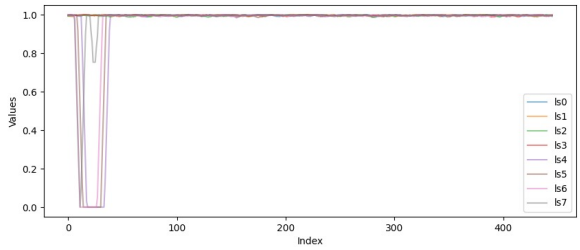

Fig. 4. Light sensor graph for BBR

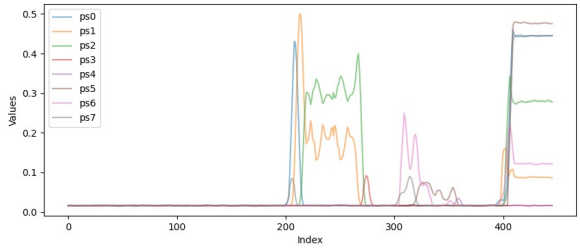

Fig. 5. Proximity sensor graph for BBR

The average runtime of the BBR was determined by measuring the duration of three simulations, both when the beacon was activated and deactivated (refer to Table 1).

TABLE I
RUNTIME FOR BBR TASK

Beacon	Simulation 1	Simulation 2	Simulation 3	Average
Deactivated(min)	03:34	03:24	03:24	03:27
Activated(min)	03:13	03:11	03:14	03:12

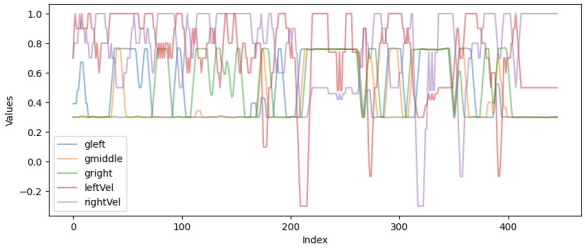

Fig. 6. Ground sensor graph for BBR

III. Task 2 using Evolutionary Robotics (ER)

A. Overview

Genetic Algorithms (GAs) are fundamental in Evolutionary Robotics (ER), especially in the Mars Rover project. Genetic algorithms (GAs), drawing inspiration from biological evolution, employ crossover and mutation operators to improve robot controllers and designs. Crossover operators, essential in genetic algorithms (GAs), merge genetic information from parent solutions to generate offspring with diverse navigation and decision-making techniques[4].

The integration of genetic algorithms (GAs) and neural networks in evolutionary robotics (ER) enables the development of complex robotic behaviours[8]. This methodology effectively uncovers unconventional solutions and strategies, frequently outperforming systems built by humans. Robots undergo evolution to enhance their skills and efficiency in ever-changing and uncertain surroundings, demonstrating the promise of ER as a potent tool for creating sophisticated autonomous systems with great adaptability and intelligence[2]. This is especially advantageous for managing the Martian landscape and assignments, promoting resilient solutions. Mutation operators, in contrast, introduce stochastic genetic modifications, which are crucial for preserving population diversity[4]. These modifications result in innovative strategies for avoiding obstacles and conserving energy, thus improving the efficiency of the Mars Rover.

A significant obstacle in the field of evolutionary robotics (ER) and genetic algorithms (GAs) as a whole is the need to find a balance between exploration (finding new regions of solutions) and exploitation (improving current solutions)[8]. This balance is crucial for the development of controllers that are both efficient and trustworthy[9].

Using simulations such as Webots by ER for testing and improving robotic controllers is crucial. It enables iterative testing in a simulated environment that mimics Mars, ensuring the development of well-suited final algorithms for the Mars Rover. The Rover's controllers should also be flexible enough to deal with unexpected hurdles, such as different terrains or impediments because ER stresses constant adaptation and development.

During the execution of tasks, the robots' performance is assessed, and the most proficient ones, determined by their capacity to adapt and complete tasks, are chosen for replication. This approach systematically refines both the physical characteristics of the robots and the parameters of their neural networks, thus improving their cognitive capacities and problem-solving skills.

B. Design and Implementation Rationale

A supervisory system and a control mechanism are used in this research to investigate the topic of evolutionary robotics. The supervisory element plays a crucial role in setting the standards for imposing penalties or providing rewards to the robotic unit. We initiated our strategy by establishing crucial characteristics, precisely the number of generations and the identification of elite members within our robotic population. We opted to utilize a sample of a hundred generations, with the population size set at eighty to cover the number of characteristics under consideration sufficiently. This choice aligns with the suggested range of 10 to 20 times the number of features. For the affluent demographic, we have determined a range including 10% to 20% of our whole population[4].

Subsequently, our attention was directed towards establishing the precise values for the rates of crossing and mutation, which were determined to be 90% and 3%, respectively (refer to Table 2). These percentages were selected based on insights from a comprehensive assessment of academic research. Expanding upon this, we devised a novel neural network architecture consisting of three successive layers of neurons, with ten neurons in each layer (refer to Table 2). The network's input layer was specifically developed to handle ten separate inputs: three from ground sensors, six from front-facing proximity sensors, and one from a light sensor. Prior to being inputted into the neural network, all signals are normalized

The primary focus of our network's output is to control and adjust the velocity of the robot's left and right motors. In order to assess the effectiveness of how we distribute weights in our network, we utilised a range of fitness functions. The measures encompassed line following, forward movement, collision avoidance, spinning, and a holistic fitness score amalgamating those above four. Greater values in these indicators signify superior performance of our model.

The system of incentives and sanctions is predicated on the robot's engagement with its surroundings. When the robot effectively meets a specific goal, such as travelling towards an illuminated light, it receives a reward. Penalties, however, are determined by the robot's proximity to barriers and its ultimate placement. There is no need to rewrite the user's text as it contains no meaningful content.

After establishing the link, the supervisory system initiates the process by transmitting the genotype of the genetic algorithm to the e-puck. Following every experimental iteration, the e-puck transmits its mean fitness value to the supervisor. Acknowledging that the supervisor resets the robot and box positions at the beginning of each run is crucial. This step is vital as they tend to relocate during lengthier simulations[4].

Within the field of robotics, the creation of a sophisticated fitness function for the e-puck robot controller is crucial for enhancing its ability to navigate and perform tasks efficiently. This study focuses on developing a comprehensive fitness function that carefully assesses all aspects of the robot's

operational dynamics from multiple perspectives (Eq. 1).

The core aspect of this function is assessing the progress made in moving ahead, which is measured by calculating the average velocity of the robot's two wheels. This statistic gives a vital gauge of the robot's linear advancement capabilities. To evaluate the robot's spinning movements, the function does a more intricate calculation: the square root of the absolute difference between the velocities of the right and left wheels. This metric provides a sophisticated comprehension of the robot's movement and positioning within the navigation area.

Including collision avoidance as a component of the fitness function is crucial for guaranteeing the operational safety and efficiency of the robot. It is determined by subtracting the highest measurement from a set of six proximity sensors from the value of one. This calculation demonstrates the robot's proficiency in securing operational distance from potential obstacles. In addition, the line-following component of the function, which is obtained by subtracting the average value of three ground sensors from one, offers valuable information on the robot's precision in following specified paths (Eq. 2). Various exploratory techniques were used to combine these components into a unified fitness score. This involved multiplying all measurements together and then normalising them to get a cumulative sum. The weighting mechanisms were used to refine this process further.

Moreover, the work builds a complex communication protocol between the supervisory system and the e-puck robot, which is crucial for the success of the applied fitness function. The supervisor utilises a Genetic Algorithm-based method to provide the genotype to the e-puck robot. In contrast, the robot transmits its mean fitness value to the supervisor at the end of every run. In order to ensure uniformity and dependability in the simulation scenario, the supervisor is responsible for resetting the robot's positions and any obstacles at the beginning of each trial.

A complex reward function is defined within the supervisory framework, intended to adjust the robot's fitness score according to predetermined performance criteria. This entails monitoring the robot's distance from strategically positioned obstacles and the specified objective. Considering their initial placements, the evaluation technique entails computing the least Euclidean distances between the robot and these elements. The distances are first averaged, then normalized according to the initial distance, and finally incorporated into the fitness score by multiplying them with the absolute value of the fitness function (Eq.3). This process produces a performance metric that varies dynamically. In addition, a bonus system is in place. If all three predefined objectives are achieved, the fitness score is increased by 25%. This encourages effective navigation and completion of tasks.

The neural network created for the genetic algorithm consists of separate layers, each with a specialised role in processing and generating output. The network starts with an input layer consisting of ten nodes. The function of these nodes is to receive and process the initial inputted data into the neural network, acting as the main entry point for information[4][10].

The network architecture consists of three hidden layers after the input layer. Each of these concealed levels comprises ten nodes. The hidden layers and their nodes play a vital function in the neural network, as they are responsible for most computation and data processing. The nodes inside these layers are interconnected and function collectively to detect patterns, analyse intricate relationships within the data, and transmit the knowledge further[4].

The output layer of the neural network is the final layer, comprising two nodes. The output layer is the final processing stage in a neural network, where it generates and provides the ultimate output. Within the framework of the genetic algorithm, this result generally signifies a determination, forecast, or categorization derived from the input data and the acquired knowledge within the concealed layers[10].

In essence, this neural network configuration, with ten input nodes, three hidden layers with ten nodes each, and a two-node output layer, constitutes a comprehensive system for the processing and analysis of data within the framework of the evolutionary algorithm. This design enables a substantial degree of intricacy in data processing, which is crucial for the advanced duties anticipated in intelligent robotics applications. (refer to figure 7)[4].

$$\Phi = V(1 - \sqrt{\Delta v})(1 - i)\rho \tag{1}$$

$$\begin{aligned} -1 \leq V \leq 1 \\ 0 \leq \Delta v \leq 1 \\ 0 \leq i \leq 1 \end{aligned}$$

$$\rho = 1 - \sum_{i=0}^{n=2} \frac{g(i)}{n+1} \tag{2}$$

$$\textit{Penalty} = 0.25|\textit{fitness}| \frac{\left(\sum_{i=0}^{n=2} \frac{\min(\text{distance}[(x,y),(x,y)_{\text{object}}])}{\text{distance}[(x_0,y_0),(x,y)_{\text{object}}]} \right)}{n+1} \tag{3}$$

TABLE II
GENETIC ALGORITHM PARAMETERS

Generation number	100
Population size	80
Elite class	12
Crossover probability	90%
Mutation probability	3%
Action Duration	300 ms
Multilayer Perceptron	[10, 10, 10, 10, 2]

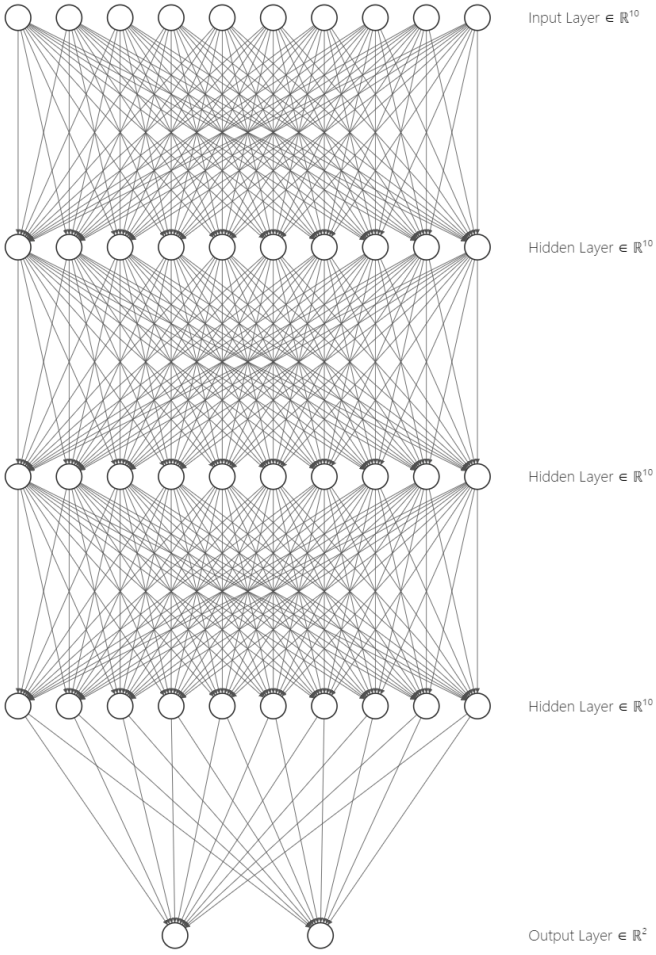

Fig. 7. Neural network for Genetic Algorithm

C. Results and Analysis

The Evolutionary Robotics (ER) test results demonstrate that the e-puck robot did not achieve the intended final goal. This deficiency is perceived through the use of both video recordings and graphical depictions. Despite a determined endeavour, the e-puck robot faces obstacles that prevent completing its intended path.

The light sensors, seen in Figure 8, effectively fulfil

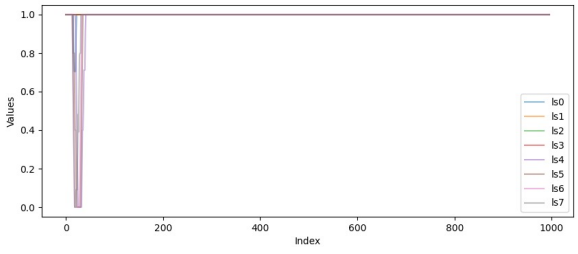

Fig. 8. Light sensor graph for ER

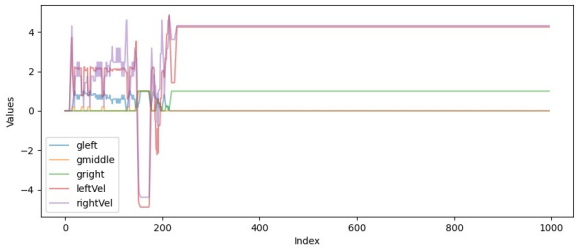

Fig. 9. Ground sensor graph for ER

their intended function of distinguishing the active status of the beacon. The data demonstrates noticeable variations, confirming the robot's ability to utilise this sensory input. Upon careful examination of the ground sensor data depicted in Figure 9, it is evident that the robot deviates from the designated ground trajectory on a single instance. This divergence may indicate the effective bypassing of obstacles that were faced.

However, a more detailed examination of wheel velocities demonstrates a synchronised negative movement, suggesting a backward motion. This behaviour can be understood by analysing the data from the proximity sensor, as illustrated in Figure 10. When the robot comes across a barrier, it has acquired the ability to respond by moving backwards to avoid the obstruction. After initially failing to overcome an obstacle, the robot becomes stuck against it and continues to try to circumvent the impediment while staying on the intended path. This may be observed by simultaneously activating both wheels at maximum velocity, demonstrating the robot's coordinated attempt to overcome the impediment.

The graphical representation of the genetic algorithms (refer to Figure 11) displays clear patterns in each individual's fitness over the simulation duration. At first, there is a rapid improvement in the fitness of the top individual, followed by a long period of no progress, and finally, a renewed growth towards the end of the simulation. The mean fitness roughly mirrors the performance of the top individual, with intermittent time lags. In the optimal simulation, the most exceptional individual has a consistent and fluctuating enhancement rate, displaying oscillating patterns with high points indicating possible difficulties caused by insufficiently adjusted mutation values, resulting in sudden alterations and missed solutions.

Discussion and Conclusion

Evolutionary robotics performance can be improved by enhancing the adaptability of the genetic algorithm (GA) by improving the mutation and crossover procedures. This involves integrating a variable sensitive to changing fitness results, allowing for the dynamic adjustment of mutation

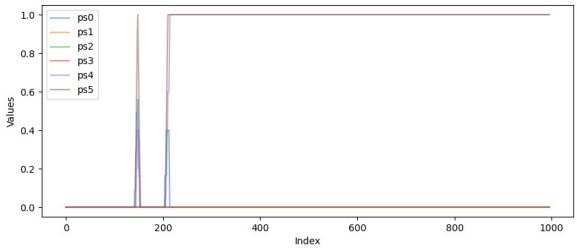

Fig. 10. Proximity sensor graph for ER

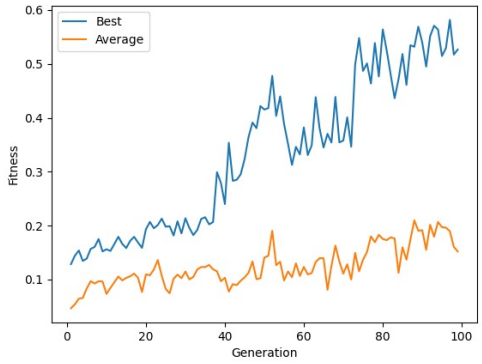

Fig. 11. Graph of Best and Average Fitness Values Across Generations

and crossover levels. This framework aims to enhance performance by adapting to the complex dynamics of the fitness environment. Considering this adaptable framework leads to examining possible consequences, such as more extended simulation periods, larger population sizes, and more generational iterations, to investigate adaptive dynamics thoroughly. Nevertheless, it is recognised that the quest for these improvements may need increased computational power and time.

The strategic approach of hyperparameter tweaking to determine the most effective setup of neural networks inside the GA framework. This systematic research thoroughly examines factors to determine their influence on performance, discovering the most efficient neural network structure. Simultaneously, the issue of choosing the necessary quantity of sensors as inputs in the neural network arises as a prominent feature, requiring thoughtful deliberation to achieve an ideal equilibrium between the amount of information provided and the computing efficiency. This thorough investigation highlights the complexities involved in improving the flexibility and effectiveness of the GA for better performance in the given situation.

The comparison of different techniques in the field of bio-inspired robotics, explicitly focusing on Behavior-Based Robotics (BBR) and evolutionary Robotics (ER), both of which we have implemented in this project, are compared with another method used in Bio-inspired called Proximal Policy Optimisation (PPO). This inquiry highlights the unique characteristics and practical uses of each methodology. Behaviour-based robotics (BBR) combines basic, responsive actions that directly react to sensor inputs, affecting the robot's actuators. This strategy is highly efficient in dynamic, real-time settings, especially in autonomous robots assigned with navigation and obstacle avoidance tasks. BBR excels in its resilience, straightforwardness, and adaptability in uncertain settings, yet it may struggle in intricate tasks that demand advanced planning or learning capabilities[1][3].

Evolutionary Robotics (ER) use evolutionary algorithms to create and enhance robots' controls and physical structures. This methodology, which emulates a process similar to natural selection, is especially valuable in intricate systems where creating explicit behaviours or control algorithms is intrinsically difficult. The capacity of ER to develop innovative solutions and adjust to dynamic surroundings is a notable benefit. However, it requires substantial computational resources and frequently demands numerous iterations for optimal solutions[4][3].

Proximal Policy Optimisation (PPO) is a recent development in reinforcement learning that utilises policy gradient techniques to improve a policy by making modest, gradual adjustments. This approach is especially appropriate for situations when it is possible to accurately replicate the conditions, such as activities involving controlling robots or playing games. PPO is recognised for its optimal equilibrium between efficiency and stability and its comparatively straightforward implementation in contrast to alternative policy gradient methods. Nevertheless, the reliance on a simulator or environment for training and the inherent limits in applying knowledge to other jobs are significant disadvantages[11][12].

The choice amongst BBR, ER, and PPO for a particular robotic task should be determined by the work's demands, the surrounding conditions, and the intended trade-off between computing efficiency and solution optimality. BBR demonstrates exceptional performance in jobs that require real-time processing and are driven by sensor data. ER is very skilled in developing solutions for intricate and ever-changing situations. PPO offers significant benefits in environments that are suitable for simulation-based training. Every methodology has distinct benefits and drawbacks, emphasising the significance of a customised approach in designing and implementing robotic systems.

IV. Acknowledgement

The authors would like to express their heartfelt appreciation to everyone who helped make this study a reality. Dr. Patricia A. Vargas and the teaching assistants deserve special recognition for their excellent guidance, insights, and experience,

which considerably improved the quality of this study. The authors also thank Heriot-Watt University for providing the essential facilities and resources.

REFERENCES

[1] F. Michaud and M. Nicolescu, "Behavior-based systems," *Springer handbook of robotics*, pp. 307–328, 2016.

[2] X.-S. Yang, S. F. Chien, and T. O. Ting, "Bio-inspired computation and optimization: An overview," *Bio-inspired computation in telecommunications*, pp. 1–21, 2015.

[3] D. Floreano, P. Husbands, and S. Nolfi, "Evolutionary robotics," *Handbook of robotics*, 2008.

[4] S. M. Lim, A. B. M. Sultan, M. N. Sulaiman, A. Mustapha, and K. Y. Leong, "Crossover and mutation operators of genetic algorithms," *International journal of machine learning and computing*, vol. 7, no. 1, pp. 9–12, 2017.

[5] F. Mondada, M. Bonani, X. Raemy, J. Pugh, C. Cianci, A. Klaptocz, S. Magnenat, J.-C. Zufferey, D. Floreano, and A. Martinoli, "The e-puck, a robot designed for education in engineering," in *Proceedings of the 9th conference on autonomous robot systems and competitions*, vol. 1, no. CONF. IPCB: Instituto Politécnico de Castelo Branco, 2009, pp. 59–65.

[6] Z. Wang, E. Nakano, and T. Matsukawa, "Realizing cooperative object manipulation using multiple behaviour-based robots," in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems. IROS'96*, vol. 1. IEEE, 1996, pp. 310–317.

[7] X. Zhang and H. Lin, "Stochastic hybrid systems modeling and performance verification of behavior-based robots," in *2015 American Control Conference (ACC)*. IEEE, 2015, pp. 781–786.

[8] K. Deep and M. Thakur, "A new mutation operator for real coded genetic algorithms," *Appl. Math. Comput.*, vol. 193, pp. 211–230, 2007. [Online]. Available: https://api.semanticscholar.org/CorpusID:2529570

[9] J. Savage, S. Munoz, M. Matamoros, and R. Osorio, "Obstacle avoidance behaviors for mobile robots using genetic algorithms and recurrent neural networks," *IFAC Proceedings Volumes*, vol. 46, no. 24, pp. 141–146, 2013.

[10] G. F. Miller, P. M. Todd, and S. U. Hegde, "Designing neural networks using genetic algorithms." in *ICGA*, vol. 89, 1989, pp. 379–384.

[11] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[12] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. P. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu, "Asynchronous methods for deep reinforcement learning," in *International Conference on Machine Learning*, 2016. [Online]. Available: https://api.semanticscholar.org/CorpusID:6875312