Annales Mathematicae et Informaticae
61 (2025) pp. 68–79
DOI: 10.33039/ami.2025.10.009
URL: https://ami.uni-eszterhazy.hu

Artificial Intelligence based robotic applications for higher education*

Isabela Drămnesc[a], Erika Ábrahám[b], László Antal[b], Péter Csóka[f], Nikolaos Fachantidis[c], Tudor Jebelean[a], Gábor Kusper[de], Konstantinos Papadopoulos[c]

[a]West University of Timişoara, Romania
{isabela.dramnesc,tudor.jebelean}@e-uvt.ro

[b]RWTH Aachen University, Germany
{abraham,antal}@cs.rwth-aachen.de

[c]University of Macedonia, Thessaloniki, Greece
{nfachantidis,kostaspap}@uom.edu.gr

[d]Eszterházy Károly Catholic University, Hungary
kusper.gabor@uni-eszterhazy.hu

[e]University of Debrecen, Hungary
kusper.gabor@inf.unideb.hu

[f]InnovITech Ltd., Hungary
peter.csoka@innovitech.hu

Abstract. This paper presents some key outcomes of the European project AiRobo, a collaborative initiative involving five universities from Romania, Germany, Greece, Hungary, and France. This is an educational project targeting higher education students, academic staff, and industry professionals. This paper presents four real-world AI-based robotic applications designed to serve as engaging teaching support materials, particularly for theoretical courses that students often find challenging to grasp. These practical applications are designed to bridge the gap between theory and practice, simplifying complex concepts and making them more accessible. They also aim to enhance the learning and teaching process, making it more engaging, motivating, and appealing for both students and academic staff. We present four real-world AI-based robotic applications: two applications integrating

*This work is co-funded by the EU through the Erasmus+ project *AiRobo: Artificial Intelligence-based Robotics*, 2023-1-RO01-KA220-HED-000152418.

Accepted: October 8, 2025
Published online: October 28, 2025

ChatGPT in Pepper and NAO robots are designed to serve as learning assistants for students, and two applications (one with the simulation of an agriculture robot in Unity and an AI-based underwater robot controller) are designed to serve as engaging teaching materials on AI-related subjects, but also for theoretical courses that students often find challenging to grasp. For each application, we outline its scope, model, and implementation code. In addition, the implementation code will be openly accessible in the final version of the paper, enabling academic staff and researchers to easily use and adapt these case studies for their own educational or research tasks.

Keywords: Artificial Intelligence, robotics, higher education

AMS Subject Classification: 68T40 Artificial intelligence for robotics

1. Introduction

Robots are indispensable across industries, from manufacturing and agriculture to research and healthcare, with AI further expanding their capabilities. Improving the quality of education in related areas is important for the development of these technologies and of the society in general. In education, applications as demonstrators and illustrators are of utmost importance, but are rarely available.

In this paper, we present four applications we developed that may be useful for researchers and teachers in subjects related to artificial intelligence, robotics, and formal verification (for example: computational and mathematical logic, automated theorem proving, algorithm synthesis and mathematical theory exploration, modeling and certifying algorithms, verification of hybrid systems, satisfiability checking, machine learning, etc.), as well as serving as learning assistants for students. The first two applications are AI-enabled robots to personalize learning and serve as teaching assistants, enhancing academic outcomes. Two further applications are AI-controlled robots that can serve as educative case studies, for example in machine learning, or for teaching formal methods to ensure safety and reliability.

Robots as student assistants. AI and robotics may enhance students' cognitive abilities through brain-computer interfaces (BCIs) [16–18]. Social robots [8] have the potential to actively engage students and enrich their learning experience. For instance, *chatbots* can interact through text or voice using AI [23, 30]. A comprehensive literature review for the use of AI in education is [32], which also reviews the use of robots [21].

As a first application, in Section 2 we describe the *Pepper robot* [28], which offers significant potential for improving student engagement, particularly in language acquisition [2, 9]. ChatGPT has been used in education [11], as well as in robotics applications [31]. The study [27] examines how integrating large language models (LLMs), such as Google PaLM2 and ChatGPT, into the Pepper robot, in conjunction with Reinforcement Learning with Human Feedback (RLHF), can enhance the robot's natural language processing (NLP) capabilities.

Our second application presented in Section 3 is a NAO storyteller. The NAO robot [29] can serve as an interactive teaching assistant [15], interactive storyteller [10], catalyst for teamwork [33], and others [7].

Robots as use cases. There is not much material available to support teaching in subjects related to AI-based robotics. Some loosely related research on the importance of logic in higher education [24] and on teaching applied formal methods [25] both address the verification of cyber-physical systems.

To provide further support, in Section 4 we present a plant-watering robot application. There are very few tutorials or educational material available in the field of agricultural robotics. For instance, [13] compares Gazebo and Unity for digital twin simulation in the field of agriculture, and [19] describes a robotic application using IoT (Internet of Things) for the intelligent watering of plants. Addressing agricultural harvesting, [12] describes a pneumatic gripper for robotic use inspired by gecko's foot and human finger.

Our last application, presented in Section 5, is an autonomous underwater vehicle. It can be used to illustrate, e.g., data gathering and processing, machine learning, simulation, testing, and formal verification.

2. Student assistance: Pepper with ChatGPT

Use case. The integration of AI assistants into classrooms has opened new possibilities for student engagement, comprehension, and skill development. However, many learners struggle with using AI-powered tools effectively due to a lack of context or familiarity. We programmed the Pepper robot as an interactive learning assistant, designed to support students in their academic journey, while fostering an understanding of AI-driven tools like ChatGPT.

Methodology. When Pepper detects human presence, it introduces itself and offers assistance to the user. It invites them to ask questions and provides relevant support. If the user provides sufficient context for ChatGPT to confidently generate a clear and accurate response, Pepper reads out the answer and congratulates the user. However, if the input lacks context, Pepper prompts the user to rephrase their question with more details to ensure a more precise and meaningful response. Thus Pepper assists by demonstrating how AI can be a valuable educational resource, explaining its strengths and limitations, while guiding students to frame their queries for better responses.

Implementation. Executing applications on Pepper requires certain workarounds due to the limitations of its underlying software. Pepper runs on NAOqi 2.5, an outdated framework that supports only Python 2.7, which introduces constraints, particularly in multi-lingual environments due to long-standing issues with Python

2.7. This creates challenges when integrating modern AI models such as ChatGPT. To enable ChatGPT functionality on Pepper, an external proxy for ChatGPT requests is required to handle requests and process responses efficiently for ChatGPT to function on Pepper's hardware. Pepper's Automatic Speech Recognition (ASR) system has inherent limitations, particularly with free-text input and multilingual speech, which led us to employ an external transcription service. In our implementation we used Google's Cloud Speech-to-Text API, selected for its higher recognition accuracy, stable latency (about 1-1.5 seconds in classroom use), and straightforward integration with the Python proxy service. An audio stream of student questions were transmitted to the API for transcription; to address privacy and data protection concerns, no personal identifiers were included in the recordings, access was limited to authorized users, and API usage was logged for auditing. This transcription is then forwarded to the ChatGPT API, where the response is generated externally and relayed back to Pepper for delivery to the student.

These adjustments ensure that Pepper can assist students despite its software constraints.

The source code for the robot is openly accessible at: `https://github.com/KostasPapadopoulosUOM/AiRobo/tree/main/PepperAIAssistant`. Our implementation requires a ChatGPT API Key. The end user will need to create an OpenAI account and then generate an API key. The API key is required to be placed at the following path: */data/home/nao/chatgpt.key*. An external ASR engine is also required due to the before mentioned limitation. In this case Google's Transcription API is used, which requires an additional Google API key with the *Cloud Speech-to-Text API* enabled. The end user will need to place the API Key in */data/home/nao/googleapi.key*.

Teaching. A significant challenge in AI-assisted education is ensuring that students develop critical thinking skills and understand the importance of context in learning. Pepper facilitates discussions that encourage students to ask meaningful questions and refine their approach when using AI tools. By doing so, it ensures that students do not rely on AI blindly but instead use it as a complementary tool to their own reasoning and problem-solving abilities.

3. Student assistance: Nao storyteller

Use case. We further leverage the NAO robot's capabilities to create a dynamic learning experience that fosters curiosity and engagement among students. By integrating AI-driven storytelling and vision recognition, we aim to enhance students' understanding of both robotics and AI in an interactive manner.

Methodology. The NAO robot operates using its autonomous life mode, allowing it to exhibit human-like behaviors such as looking around, adjusting its posture, and responding naturally to stimuli. These features help to create a more immersive and interactive experience for students. The key components of our methodology include computer vision, AI-generated storytelling, and embodied interaction through gestures and speech. Additionally, we employ text-to-speech technology to ensure the robot delivers narratives in a clear and expressive manner, and automatic speech recognition to allow students to interact with the robot through spoken commands, further bolstering engagement.

Implementation. The NAO robot was programmed with the following functionalities. (i) Face Detection: the robot identifies and acknowledges the presence of students, creating a more personalized interaction. (ii) Vision Recognition: using its onboard camera, NAO recognizes images it has been trained on. In this study, we employed cubes featuring images of ancient Greek gods. (iii) Storytelling via AI: upon recognizing an image, the robot queries an AI system to generate a humorous story about the identified Greek god. It then reads the story aloud while performing subtle movements to maintain engagement. The source code for this application is publicly available at: `https://github.com/KostasPapadopoulosUOM/AiRobo/tree/main/NAO_Cube_Game`.

This implementation requires a ChatGPT API key. To obtain one, the end user must create an OpenAI account and generate an API key. The key should be stored at the following path: */data/home/nao/chatgpt.key*. This configuration allows the NAO robot to interact with ChatGPT for generating educational content and storytelling responses.

Teaching. The primary objective of this project was to introduce students to AI and robotics in an engaging and interactive manner. By incorporating storytelling and physical interaction, the students were encouraged to explore AI capabilities while learning about ancient mythology. Observations suggest that the robot's interactive features improved student participation and interest.

4. Robots as use cases: Agriculture robot

Use case. In this section we present our plant-watering robot, whose control is specified by the state machine depicted on Figure 1. The robot starts at its charging station, where it recharges and refills its water tank before navigating to water flowers at predefined locations. It encounters obstacles, classified as living (e.g., animals) and non-living (e.g., rocks); and attempts to navigate around them. If avoidance is impossible, it requests external assistance. When battery or water levels are low, the robot returns to the charging station; if it depletes its energy before reaching the station, it also requests help. To simplify the state machine, we have a special state, called ALL_STATES, which describes the common behavior of each state. We used the hardware platform SCOUT MINI [1], equipped with GPS, a compass, two LiDAR sensors (front and rear), and a velocity sensor.

Methodology. The robot's navigation relies on reinforcement learning (RL), where the agent optimizes movement through action rewards, using the Proxi-

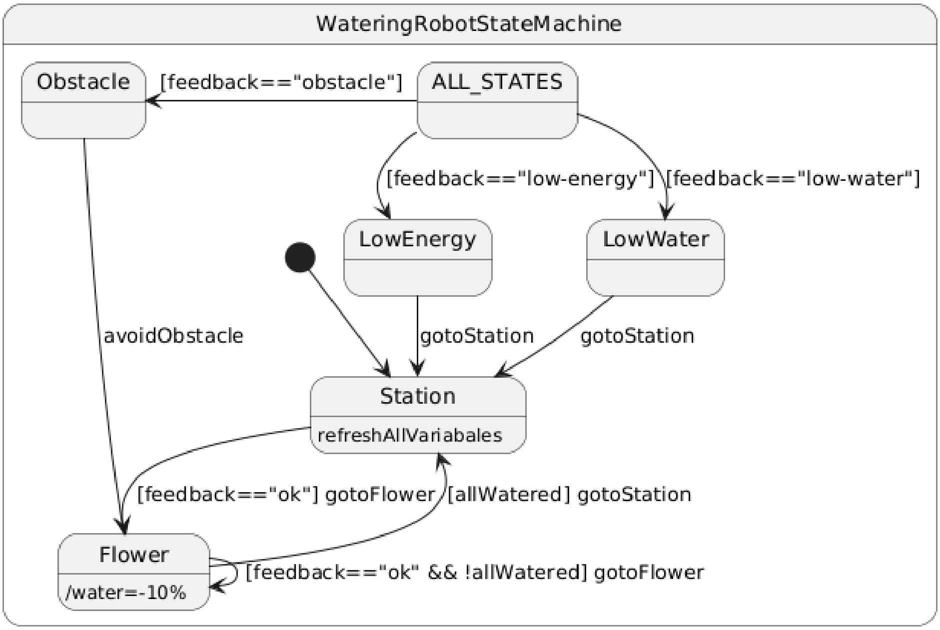

Figure 1. Finite state machine of the plant-watering robot.

mal Policy Optimization (PPO) algorithm. The hyperparameters used for the algorithm and training can be seen in Table 1. The observation space is an 11-dimensional continuous vector representing the robot's state. It includes the angular difference between the robot's orientation and the target vector, normalized to $[-1, 1]$. The linear velocity of the robot, also normalized to $[-1, 1]$. The distance between the robot and the target, normalized by the initial distance at the beginning of each episode to a range of $[0, 1]$. There are two LiDAR sensors on the robot. The forward facing sensor covers a 180°, from $-90°$ to $90°$ relative to the robot's heading. It is divided into 4 equal sections, and the minimum reading is taken in each section. The process is mirrored for the rear sensor, resulting in a total of eight distance readings, each normalized to $[0, 1]$. The action space is a 2-dimensional continuous vector, defining the rover's linear and angular velocity, each component normalized to $[-1, 1]$.

The reward function is designed to guide the agent towards its target while avoiding obstacles. More formally, the reward R_t at each timestep t is defined as:

$$R_t = \begin{cases} +200 & \text{if } d_t < \epsilon_{\text{target}} \quad \text{(goal reached)} \\ -200 & \text{if collision occurs} \\ -150 & \text{if } t \geq T_{\max} \quad \text{(timeout)} \\ 10 \cdot (d_{t-1} - d_t) & \text{otherwise} \end{cases}$$

Table 1. List of hyperparameters.

Hyperparameter	Value
Algorithm Parameters	
Number of epochs	8
Batch size	1024
Discount (γ)	0.99
GAE parameter (λ)	0.95
PPO clip range (ϵ)	0.2
Critic Loss Coefficient	0.5
Network Architecture	
Number of hidden layers	2
Hidden layer size	128
Hidden Layer Activation	ReLU
Actor Output Activation	Tanh
Optimizer algorithm	Adam
Training parameters	
Initial learning rate	3e-4
Learning rate decay	linear decay
Learning rate decay frequency (in episodes)	250
Minimum learning rate	1e-8
Standard deviation of selected actions	1

Here, d_t is the distance to the target at timestep t, ϵ_{target} is the success threshold distance set to 0.1 meters, and $T_{\max}$ is the maximum episode length, set to 1500 steps. A penalty is applied immediately upon any collision. The term $(d_{t-1} - d_t)$ provides a reward for moving closer to the target and a penalty for moving away in non-terminal states.

To enhance training, we apply curriculum learning [22], gradually increasing task complexity. RL agents often struggle with generalization. To address this, elements of the training environment are randomized.
The first stage involves point-to-point navigation. Here, the environment is a 10×10 meter plane. For each episode, the position of the target and the starting pose of the robot are randomly selected within this area. During this phase, the RL agent only uses the relative heading, velocity and distance to the target as observations. The training continues until 90 out of the last 100 episodes are successful.
The second stage introduces obstacles in the form of walls, to teach avoidance behavior. These walls are parallel to the direct path to the target and are positioned to form a 1.5-meter-wide corridor. The length of each wall is randomized for each episode, ranging from 0.1 to 15 meters. Here, the agent uses the full observation

vector including LiDAR data.
The final stage is a randomly generated but static garden resembling real-world conditions. The walls are replaced by randomly placed flowers. There is a moving dog to test the agent's robustness to changing conditions. It is used to evaluate the RL agent's behavior in a previously unseen and more realistic setting.

Catastrophic forgetting [14] is another challenge, where learning a new task degrades previously learned behaviors. This is mitigated by periodically revisiting earlier tasks, ensuring stable performance across different environments. Specifically, during the second stage, the environment alternates between the first and second stage every 100 episodes.

Implementation. The system consists of a Unity-based simulation and a Python module for RL training and testing. Running the simulation requires installing Unity. The final implementation is publicly available at `https://github.com/csokapeter/Agricultural-Robot`.

Teaching. This project serves as an educational tool for reinforcement learning and robotics. Students gain hands-on experience with RL-based navigation, digital sensor integration in Unity, and training methodologies. It allows them to experiment with reward functions, agent parameters, and real-world deployment challenges. Additionally, it highlights hardware limitations and safety considerations when transferring trained agents from simulation to physical robots.

5. Robots as use cases: Underwater robot controller

Use case. Autonomous underwater vehicles (AUVs) provide a promising technique to perform underwater exploration missions autonomously [4, 5]. To successfully operate in an uncertain, a priori unknown environment, a *rule-based* algorithm proposed in [20] relies on specific sensor measurements – such as depth, altitude and forward distance – to estimate the steepness of the seafloor (α), allowing the robot to adjust its pitch (β) accordingly (see Figure 2). Using these sensor values, *bottom tracking* maintains the distance to the seafloor as constant as possible, to increase the reliability of sensor data gathering, and *obstacle avoidance* recognizes rocks, walls etc. to avoid collisions.

While this method performs reasonably well in smooth environments, it is highly sensitive to noise, and it requires highly complex code, making it difficult to debug and maintain. To address these limitations, in [3] the authors propose an alternative *AI-based* controller to overcome these challenges.

Methodology. The AI-based controller [3] utilizes a neural network, which uses the observed sensor values within some time window to issue control commands within a long short-term memory (LSTM) architecture (see Figure 2). These control commands adjust the pitch of the AUV, facilitating bottom tracking with more efficient obstacle avoidance (less braking) and better robustness.

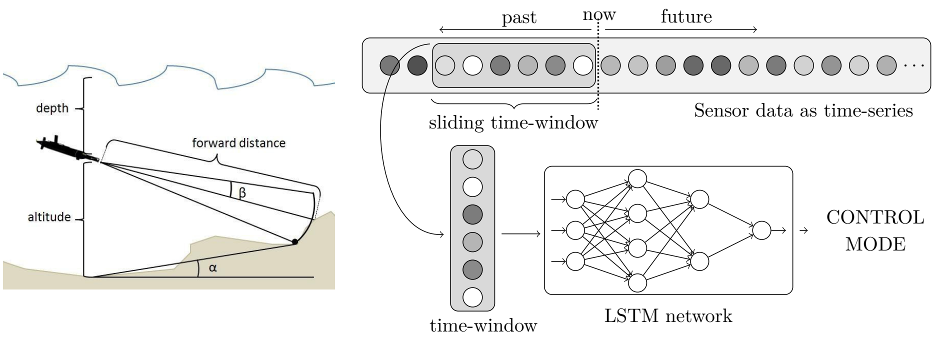

Figure 2. Left: Illustration from [20] for the *rule-based* obstacle avoidance method. Right: The LSTM controller system architecture.

A key challenge of [3], is the lack of *labeled* data for the training. To address this, the authors used sensor data of logs from previously executed real world missions, and employed signal processing techniques for labeling each timestamp. Further improvements were achieved through online re-training using a simulator, and the employment of a simplex architecture [6, 26] to revert to the rule-based method in case the AI-based control would not prevent a collision. The controller was deployed on an AUV of OceanScan MST and two real world surveys were conducted. These demonstrated that the AI-based controller produced better efficiency (i.e., shorter mission time, less battery usage), while maintaining safety (no collisions) throughout the mission.

Implementation. The code implementation, the trained neural network in ONNX format, and further instructions on how to setup and use the simulator with the neural network controller is openly accessible at: `https://github.com/antallaszlo011/improved-AUV-obstacle-avoidance`.

Teaching. This application can be used to illustrate the *development of AI-based controllers*, including the preparation of training data, the training and re-training of neural networks, knowledge distillation, and the embedding of AI-based controllers in a simplex architecture for fall-safe functioning. Besides simulation and testing, *formal methods* can be applied to assess the reliability and safety of the AUV behavior.

6. Conclusions

In this paper we presented four real-world, AI-based robotic applications specifically designed to enhance the learning experience for students. These applications not only serve as interactive assistants, making the learning process more engaging, but also provide practical solutions for complex theoretical courses that students often struggle with, helping them grasp challenging concepts more effectively. A key contribution of this work is the free availability of the application implementations, empowering academic staff and researchers to seamlessly integrate or adapt these resources into their own teaching or research activities.

As future work, the authors consider: Integrating various generative AI engines, such as DeepSeek, which offers the advantage of self-hosting at no cost; Incorporating alternative transcription engines with a focus on self-hosted solutions, potentially leveraging OpenAI's Whisper model, as well as evaluating different deployment strategies and optimizing model performance for real-time transcription in educational contexts; Implementing additional sensors in the agricultural robot simulation to better reflect real-world conditions and allow students to experiment with different configurations of the robot.

References

[1] AGILE X: *SCOUT MINI*, https://global.agilex.ai/products/scout-mini.

[2] M. Alemi, A. Meghdari, M. Ghazisaedy: *The Impact of Social Robotics on L2 Learners' Anxiety and Attitude in English Vocabulary Acquisition*, eng, International Journal of Social Robotics 7 (2015), pp. 523–535, doi: 10.1007/s12369-015-0286-y.

[3] L. Antal, M. Aubard, E. Ábrahám, A. Madureira, L. Madureira, M. Costa, J. Pinto, R. Campos: *A Collision Avoidance Method for Autonomous Underwater Vehicles Based on Long Short-Term Memories*, in: Proc. of the 13th Innovations in Bio-Inspired Computing and Applications (IBICA'22), vol. 649, LNNS, Springer, 2023, pp. 448–457, isbn: 978-3-031-27499-2, doi: 10.1007/978-3-031-27499-2_42.

[4] M. Aubard, L. Antal, A. Madureira, E. Ábrahám: *Knowledge Distillation in YOLOX-ViT for Side-Scan Sonar Object Detection*, CoRR abs/2403.09313 (2024), doi: 10.48550/ARXIV.2403.09313, arXiv: 2403.09313.

[5] M. Aubard, L. Antal, A. Madureira, L. F. Teixeira, E. Ábrahám: *ROSAR: An Adversarial Re-Training Framework for Robust Side-Scan Sonar Object Detection* (2024), doi: 10.48550/ARXIV.2410.10554, arXiv: 2410.10554.

[6] S. Bak, D. K. Chivukula, O. Adekunle, M. Sun, M. Caccamo, L. Sha: *The system-level simplex architecture for improved real-time embedded system safety*, in: Proc. of the 15th IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS'09), IEEE, Institute of Electrical and Electronics Engineers, 2009, pp. 99–107, doi: 10.1109/rtas15501.2009.

[7] H. Banaeian, I. Gilanlioğlu: *Influence of the NAO robot as a teaching assistant on university students' vocabulary learning and attitudes*, Australasian Journal of Educational Technology (2021), pp. 71–87, doi: 10.14742/ajet.6130.

[8] T. Belpaeme, J. Kennedy, A. Ramachandran, B. Scassellati, F. Tanaka: *Social robots for education: A review*, Journal of Science Robotics Vol. 3.Issue 21 (2018), doi: 10.1126/scirobotics.aat5954.

[9] T. Belpaeme, F. Tanaka: *Social robots as educators*, in: OECD Digital Education Outlook 2021 Pushing the Frontiers with Artificial Intelligence, Blockchain and Robots: Pushing the Frontiers with Artificial Intelligence, Blockchain and Robots, OECD Publishing Paris, 2021, p. 143.

[10] H. Cao, R. Simut, N. Krepel, B. Vanderborght, J. Vanderfaeillie: *Could NAO robot function as model demonstrating joint attention skills for children with autism spectrum disorder? An exploratory study*, International Journal of Humanoid Robotics 19 (04 2022), doi: 10.1142/s0219843622400060.

[11] O. E. Chinonso, A. M.-E. Theresa, T. C. Aduke: *ChatGPT for Teaching, Learning and Research: Prospects and Challenges*, Global Academic Journal of Humanities and Social Sciences (2023), url: https://api.semanticscholar.org/CorpusID:257476234.

[12] A. Clark, L. Goodsell-Carpenter, P. Buckow, D. Hewett, F. White, A. Imam, N. Naz, B. Robinson, S. K. Manna, A. Ahmed: *Bio-Inspired Soft Pneumatic Gripper for Agriculture Harvesting*, in: Proc. of the 25th Annual Conference on Towards Autonomous Robotic Systems (TAROS'24), vol. 15052, LNCS, Springer, 2025, pp. 266–277, isbn: 978-3-031-72062-8, doi: 10.1007/978-3-031-72062-8_24.

[13] J. P. Espejel Flores, A. Yilmaz, L. A. Soriano Avendaño, G. Cielniak: *Comparative Analysis of Unity and Gazebo Simulators for Digital Twins of Robotic Tomato Harvesting Scenarios*, in: Proc. of the 25th Conference Towards Autonomous Robotic Systems (TAROS'24), vol. 15051, LNAI, Springer, 2025, pp. 15–27, isbn: 978-3-031-72059-8, doi: 10.1007/978-3-031-72059-8_2.

[14] R. M. French: *Catastrophic forgetting in connectionist networks*, Journal of Trends in Cognitive Sciences Vol. 3.Issue 4 (1999), pp. 128–135, doi: 10.1016/s1364-6613(99)01294-2.

[15] S. Gouraguine, M. Qbadou, M. Rafik, M. Riad, K. Mansouri: *A new knowledge primitive of digits recognition for NAO robot using MNIST dataset and CNN algorithm for children's visual learning enhancement*, Journal of Information Technology Education: Research 22 (2023), pp. 389–408, doi: 10.28945/5194.

[16] N. Jamil, O. S. Alawa, S. M. Manar, S. A. Alaidarous, A. S. Adam, S. A. Eldemerdash, A. N. Belkacem: *On Neurodevelopmental Disorder based on Brain Computer Interface for Enhancing the Learning Process*, in: Proc. of the 2023 IEEE Frontiers in Education Conference (FIE'23), vol. 1, Institute of Electrical and Electronics Engineers, 2023, pp. 1–9, doi: 10.1109/FIE58773.2023.10343434.

[17] N. Jamil, A. Belkacem, A. Lakas: *On enhancing students' cognitive abilities in online learning using brain activity and eye movements*, English, Jorunal of Education and Information Technologies Vol. 28.Issue 4 (Apr. 2023), pp. 4363–4397, issn: 1360-2357, doi: 10.1007/s10639-022-11372-2.

[18] N. Jamil, A. N. Belkacem, S. Ouhbi, C. Guger: *Cognitive and Affective Brain–Computer Interfaces for Improving Learning Strategies and Enhancing Student Capabilities: A Systematic Literature Review*, IEEE Access 9 (2021), pp. 134122–134147, doi: 10.1109/ACCESS.2021.3115263.

[19] KA, Jayasheel Kumar, Akhtar, Mannaan, Talha, C. Mohammed, VG, Gokul: *Design of Hybrid Sprinkler: The IoT-Powered Robot for Watering Plants*, MATEC Web Conf. 393 (2024), p. 04004, doi: 10.1051/matecconf/202439304004.

[20] L. Madureira, A. Sousa, J. Braga, P. Calado, P. Dias, R. Martins, J. Pinto, J. Sousa: *The light autonomous underwater vehicle: Evolutions and networking*, in: Proc. of 2013 MTS/IEEE OCEANS Conference - Bergen (OCEANS'13), vol. 1, Institute of Electrical and Electronics Engineers, 2013, pp. 1–6, doi: 10.1109/OCEANS-Bergen.2013.6608189.

[21] R. R. Murphy: *"Competing" for a robotics education*, IEEE Robotics & Automation Magazine 8.2 (2001), pp. 44–55.

[22] S. Narvekar, B. Peng, M. Leonetti, J. Sinapov, M. E. Taylor, P. Stone: *Curriculum learning for reinforcement learning domains: A framework and survey*, Journal of Machine Learning Research 21.181 (2020), pp. 1–50.

[23] J. Q. Pérez, T. Daradoumis, J. M. M. Puig: *Rediscovering the use of chatbots in education: A systematic literature review*, Computer Applications in Engineering Education 28.6 (2020), pp. 1549–1565.

[24] A. Platzer: *The Significance of Symbolic Logic for Scientific Education* (2024), doi: 10.48550/ARXIV.2407.09959, arXiv: 2407.09959.

[25] K. Y. Rozier: *On Teaching Applied Formal Methods in Aerospace Engineering*, in: Formal Methods Teaching, Cham: Springer International Publishing, 2019, pp. 111–131, isbn: 978-3-030-32441-4.

[26] D. Seto, B. Krogh, L. Sha, A. Chutinan: *The Simplex architecture for safe online control system upgrades*, in: Proc. of the 1998 American Control Conference. (ACC'98), vol. 6, IEEE, Institute of Electrical and Electronics Engineers, 1998, pp. 3504–3508, doi: 10.1109/ACC.1998.703255.

[27] S. Shibi, S. Zaidi: *STEM Approach to Enhance Robot-Human Interaction Through AI Large Language Models and Reinforcement Learning*, in: 2024 IEEE Integrated STEM Education Conference (ISEC), 2024, pp. 1–2, doi: 10.1109/ISEC61299.2024.10665163.

[28] SoftBank Robotics: *Meet Pepper: The Robot Built for People*, https://us.softbankrobotics.com/pepper.

[29] SoftBank Robotics: *NAO: Personal Robot Teaching Assistant*, https://us.softbankrobotics.com/nao.

[30] E. Vázquez-Cano, S. Mengual-Andrés, E. López-Meneses: *Chatbot to improve learning punctuation in Spanish and to enhance open and flexible learning environments*, International Journal of Educational Technology in Higher Education 18 (2021), pp. 1–20.

[31] S. H. Vemprala, R. Bonatti, A. F. C. Bucker, A. Kapoor: *ChatGPT for Robotics: Design Principles and Model Abilities*, IEEE Access 12 (2023), pp. 55682–55696, url: https://api.semanticscholar.org/CorpusID:259141622.

[32] S. Wang, F. Wang, Z. Zhu, J. Wang, T. Tran, Z. Du: *Artificial intelligence in education: A systematic literature review*, Expert Systems with Applications 252 (2024), p. 124167, issn: 0957-4174, doi: 10.1016/j.eswa.2024.124167, url: https://www.sciencedirect.com/science/article/pii/S0957417424010339.

[33] Z. Warren, Z. Zheng, A. Swanson, E. Bekele, L. Zhang, J. A. Crittendon, A. Weitlauf, N. Sarkar: *Can robotic interaction improve joint attention skills?*, Journal of Autism and Developmental Disorders 45 (11 2013), pp. 3726–3734, doi: 10.1007/s10803-013-1918-4.