applied sciences

MDPI

Article

RL-Based Sim2Real Enhancements for Autonomous Beach-Cleaning Agents

Francisco Quiroga[1], Gabriel Hermosilla [1,*], German Varas [2], Francisco Alonso [1] and Karla Schröder [1]

[1] Escuela de Ingeniería Eléctrica, Pontificia Universidad Católica de Valparaíso (PUCV), Valparaíso 2340025, Chile; francisco.quiroga.a@mail.pucv.cl (F.Q.); francisco.alonso@pucv.cl (F.A.); karla.schroder@pucv.cl (K.S.)
[2] Instituto de Física, Pontificia Universidad Católica de Valparaíso (PUCV), Valparaíso 2340025, Chile; german.varas@pucv.cl
* Correspondence: gabriel.hermosilla@pucv.cl

Abstract: This paper explores the application of Deep Reinforcement Learning (DRL) and Sim2Real strategies to enhance the autonomy of beach-cleaning robots. Experiments demonstrate that DRL agents, initially refined in simulations, effectively transfer their navigation skills to real-world scenarios, achieving precise and efficient operation in complex natural environments. This method provides a scalable and effective solution for beach conservation, establishing a significant precedent for the use of autonomous robots in environmental management. The key advancements include the ability of robots to adhere to predefined routes and dynamically avoid obstacles. Additionally, a newly developed platform validates the Sim2Real strategy, proving its capability to bridge the gap between simulated training and practical application, thus offering a robust methodology for addressing real-life environmental challenges.

Keywords: deep reinforcement learning; mobile robotics; position control; obstacle avoidance; simulated environment; real laboratory validation; beach cleaning; Sim2Real

check for updates

Citation: Quiroga, F.; Hermosilla, G.; Varas, G.; Alonso, F.; Schröder, K. RL-Based Sim2Real Enhancements for Autonomous Beach-Cleaning Agents. *Appl. Sci.* **2024**, *14*, 4602. https://doi.org/10.3390/app14114602

Academic Editors: Nan Ma, Taohong Zhang and Yang Yang

Received: 26 March 2024
Revised: 21 May 2024
Accepted: 24 May 2024
Published: 27 May 2024

1. Introduction

The degradation of coastal ecosystems due to pollution is a pressing issue, particularly on beaches where waste accumulation adversely impacts both human and marine life [1,2]. In Chile, the severity of this issue is compounded by inadequate waste management practices, leading to significant shoreline pollution [3]. This paper introduces the deployment of autonomous beach-cleaning robots, equipped with advanced artificial intelligence, as a novel solution to mitigate these effects and contribute significantly to marine conservation.

The motivation for this work stems from the urgent need to preserve marine biodiversity and improve the health of coastal ecosystems. Previous efforts to tackle beach pollution have primarily focused on manual cleanup operations, which are often inefficient and unsustainable. Moreover, existing robotic technologies have not yet fully adapted to the dynamic and complex nature of beach environments, where fluctuating conditions present substantial operational challenges.

Building upon existing knowledge in the fields of robotics and artificial intelligence, this paper employs DRL, an advanced AI technique that integrates deep learning with reinforcement learning principles, to develop effective control systems for autonomous robots [4,5]. Through extensive training in simulated environments, these robots master essential skills for navigating complex terrains, avoiding obstacles, and maintaining precise position control, crucial for their effective operation in real-world scenarios [6].

Utilizing the Khepera IV robot within the CoppeliaSim environment, this paper pioneers a systematic approach to controller design. Robots are trained in a Gym-simulated environment, carefully designed to mirror real-world conditions. This methodology ensures

a seamless transition from simulation to actual deployment, with the robots demonstrating robustness and adaptability in their behaviors.

The contributions of this work are twofold: it establishes a robust framework for developing autonomous systems capable of efficiently operating in unpredictable beach environments, and it sets a benchmark in the methodology for transitioning from simulated to real-world applications, enhancing the practical applicability of reinforcement learning in environmental conservation efforts.

In conclusion, by implementing this innovative technology the issues of beach pollution are addressed effectively, establishing a precedent for the application of advanced AI in environmental protection and setting the stage for future initiatives in autonomous robotic systems for ecological restoration.

2. State of Art

The issue of beach cleaning represents a significant environmental challenge, where traditional solutions have demonstrated clear limitations [7,8]. Despite the prevalence of conventional methods such as the use of shovels and tractorized systems, their effectiveness is compromised in the face of increasingly complex challenges, such as escalating pollution and the urgent need to adopt more sustainable and efficient approaches [9].

The application of autonomous robotics in beach cleaning represents a developing area of research, still in its early stages but showing significant potential for advancement [10,11]. Traditional beach-cleaning robots, which are predominantly large-scale [12], often require human supervision and/or control, limiting their autonomy and operational efficiency [13–15]. However, recent advancements in DRL and simulation technologies have introduced new opportunities for improving these systems. Studies employing DRL agents, such as DDPG and DQN, have been conducted solely in simulation [16], demonstrating DRL's capability to train agents that can navigate and operate in complex [17,18] and dynamic [6,19] environments, typical of coastal settings. The success of DRL in autonomous vehicles [20,21] and drones [22–24] highlights its potential for enhancing autonomous navigation and real-time decision-making, which are crucial for efficient beach cleaning. Integrating Imitation Learning (IL) with RL could offer a robust strategy, where specific routes and cleaning tasks are predefined [25].

Simulating to real-world transfer (Sim2Real) has become a cornerstone in robotics, proving invaluable for training agents in simulated settings before their real-world application. This method's efficacy in translating simulated learning to practical scenarios is comprehensively analyzed in references [26,27], with specific applications to mobile robotics discussed in [28,29]. Extending beyond traditional robotics, Sim2Real has also facilitated advancements in diverse areas. For instance, deep reinforcement learning has been applied to optimize wind turbine energy output, demonstrating the versatility of Sim2Real techniques in energy sectors, as explored in [30]. Similarly, convolutional proximal policy optimization has been utilized for mapless navigation [31] and deep deterministic policy gradient methods have been employed for precise target tracking in [32].

In comparison to traditional AI or robotics control techniques, Sim2Real offers a distinctive advantage, especially in handling unpredictable and dynamic environments like beaches. Traditional methods often struggle to account for the complexities inherent in coastal landscapes, lacking the capability to incorporate factors such as variability in weather conditions, the intermittent presence of objects, and variations in terrain characteristics. Simulations also provides a safe and controllable space to test and refine algorithms before implementation in the field, as discussed in [33–35], where efforts are made to incorporate safety factors within simulated environments and the policies that the agents learn.

The Sim2Real technique presents significant challenges in robotics and other fields where real-world training is impractical or risky. This process involves training an RL model in a simulated environment and then deploying it in a real-world setting. However, this transition is not without complexities. One major hurdle is the reality gap [36], which refers to the substantial differences between simulated and real environments. These dis-

parities, including unmodeled dynamics and variabilities in the real world, can undermine the effectiveness of the learned policy. Additionally, ensuring the generalization and robustness of the model poses a challenge. Models trained in deterministic simulations may struggle to adapt to the complexities and variability of real-world environments, potentially leading to over-specialization in simulation-specific features. Moreover, the presence of uncertainty and variability in real-world scenarios further complicates matters. The model may encounter novel situations during deployment that were not present during training, highlighting the need for the model to handle uncertainties adeptly. Addressing these challenges is crucial for the successful application of Sim2Real techniques in real-world scenarios.

Despite progress in DRL and Sim2Real technologies, their application in beach cleaning is minimally addressed in the existing literature, highlighting a significant research and development opportunity. This paper addresses this gap by applying advanced DRL and Sim2Real techniques to develop an autonomous robotic system for beach cleaning. The research primarily focuses on adapting position control skills from simulated environments to real-world operations, thereby advancing the development of effective and sustainable robotic solutions for preserving coastal ecosystems.

3. Reinforcement Learning

Reinforcement learning is a computational paradigm where an agent interacts with its environment to learn optimal behaviors through trial and error, guided by a reward system. This learning process involves the agent making decisions, executing actions, and receiving feedback in the form of rewards aimed at maximizing long-term benefits.

The interaction between the agent and the environment occurs at each discrete time step, t. As depicted in Figure 1, the agent receives an observation or state from the environment (S_t), through which it chooses an action (A_t) from the set allowed by the environment. In return for this action, the agent receives a reward (R_t) and a new state.

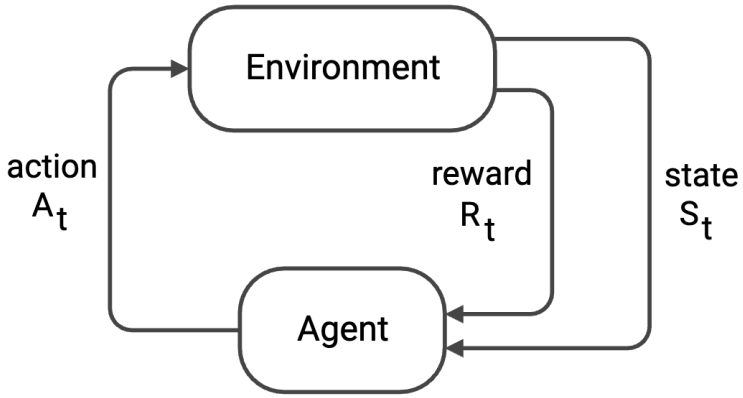


Figure 1. Interaction of the RL agent and environment.

This work utilizes two prominent RL algorithms: Proximal Policy Optimization (PPO) and Deep Deterministic Policy Gradient (DDPG). Both algorithms are implemented using the Stable Baselines3 library, known for its robust and efficient implementation of advanced RL algorithms [37].

The choice of Stable Baselines3 is based on its preference due to its extensive support and efficient implementation of RL algorithms, simplifying the experimentation and comparison of strategies. This library choice is crucial to ensure a robust and standardized foundation in the implementation of RL algorithms. The use of PPO and DDPG is motivated by the intention to explore and compare different approaches in solving specific tasks, such as position control and obstacle avoidance, leveraging the capabilities and facilities offered by Stable Baselines3.

3.1. Deep Deterministic Policy Gradient (DDPG)

DDPG is a model-free, off-policy actor-critic algorithm optimized for continuous action spaces, making it ideal for applications such as mobile robotics. It integrates the strengths of policy-based and value-based approaches, facilitating the development of complex control strategies that involve precise movements and real-time decision-making. The actor network proposes actions based on the current state, while the critic evaluates these actions by computing the value function as follows:

$$Q(s,a) \leftarrow Q(s,a) + \alpha\left(r + \gamma \max_{a'} Q(s',a') - Q(s,a)\right), \tag{1}$$

where α represents the learning rate and γ is the discount factor. The actor's policy is updated using the sampled policy gradient:

$$\nabla_{\theta^\mu} J \approx \frac{1}{N}\sum_i \nabla_a Q(s,a|\theta^Q)|_{s=s_i,a=\mu(s_i)} \nabla_{\theta^\mu}\mu(s|\theta^\mu)|_{s_i}. \tag{2}$$

For a more detailed understanding of the DDPG agent, readers are referred to the foundational paper [38], which provides comprehensive insights into the algorithm's architecture and performance metrics in various environments.

3.2. Proximal Policy Optimization (PPO)

PPO, known for its stability and efficiency, particularly in environments with high variability, implements a clipped surrogate objective function to manage policy updates [39]. This method prevents drastic policy changes that could destabilize the learning process. The primary equation of PPO reflects the limitation on policy updates:

$$L^{CLIP}(\theta) = \hat{\mathbb{E}}_t\left[\min(r_t(\theta)\hat{A}_t, \text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t)\right], \tag{3}$$

where $r_t(\theta)$ denotes the ratio of the new policy probability to the old policy probability and $(\hat{A}_t)$ represents the advantage estimate at time t.

For a comprehensive examination of the Proximal Policy Optimization algorithm and its application across diverse environments, readers are encouraged to refer to [40]. This paper provides an in-depth exploration of PPO's innovative clipped surrogate objective function, which is crucial for maintaining stability in policy updates by preventing excessive deviations in policy behavior.

Both DDPG and PPO have been effectively applied in various robotic tasks, providing a solid foundation for addressing the unique challenges of autonomous beach cleaning. The selected algorithms allow for the development of deterministic policies and flexible, safe policy adjustments, critical for autonomous systems operating in unpredictable environments.

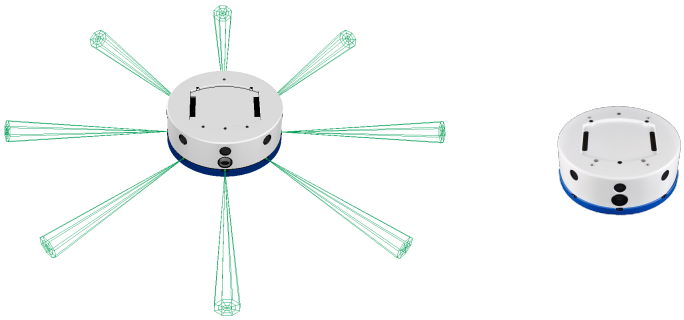
4. Simulation and Mobile Robot

The simulation of autonomous robotic systems is crucial for evaluating advanced RL agents, particularly those requiring sophisticated position control and obstacle avoidance capabilities. CoppeliaSim (V4.6.0), a versatile 3D robotic simulation software, has been employed to model the Khepera IV robot meticulously, enabling detailed examinations of autonomous navigation within a controlled environment.

The Khepera IV robot, developed by K-Team and renowned for its modular design and independent wheel motorization, is equipped with a comprehensive suite of sensors. These include eight infrared sensors for obstacle detection, additional sensors for fall prevention and line following, and ultrasonic sensors for long-range object detection, complemented by an accelerometer, gyroscope, encoders, and a color camera. This array facilitates the sophisticated perception and navigation capabilities crucial for RL applications in dynamic environments like beach cleaning [41].

CoppeliaSim supports this project by allowing precise replication of the Khepera IV's physical and sensory attributes through its KH4VREP library [42], which models the robot's rigid structure and non-deformable wheels, essential for accurate simulation outcomes. The robot model operates within CoppeliaSim under control via a remote Python API, fostering efficient programming and execution of RL algorithms. This setup ensures effective bidirectional communication between the simulation environment and the RL agents, facilitating the development and testing of algorithms in a risk-free setting [43].

Figure 2 showcases the Khepera IV robot in both simulated and real environments, highlighting the fidelity of the simulation in replicating real-world conditions.



(**a**) (**b**)

Figure 2. Comparison of Simulated Khepera IV with IR Sensors (**a**) and Real Khepera IV [44] (**b**).

Utilizing CoppeliaSim in this project offers significant advantages, including the capability to accurately simulate the physical and sensory attributes of the Khepera IV robot. This provides a reliable platform for developing and validating reinforcement learning algorithms in tasks such as navigation and obstacle avoidance before their real-world application, thereby reducing potential deployment risks and costs.

5. Environment Project Setup

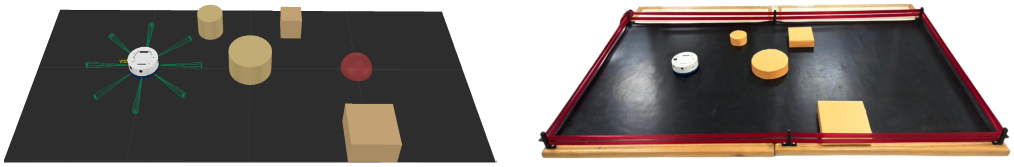
This section details the development and integration of the simulated and real testing environments used for the RL training of the Khepera IV robot, employing the OpenAI Gym library to facilitate effective communication between the RL agents and these environments.

5.1. Gym Environments

Two environments have been developed to bridge the gap between theoretical models and practical application. The first, a simulated environment crafted within CoppeliaSim, mirrors the intricate dynamics of robot–environment interactions with high fidelity. The second, its physical counterpart, is established in a laboratory setting, ensuring that RL agents receive consistent training experiences across both settings.

The primary task in these environments involves guiding the Khepera IV robot to reach a visually marked Target Point (TP) while navigating through scattered obstacles within a 2 m $\times$ 1 m area. Training sessions initiate with the robot in a central position, concluding upon successful TP navigation, a collision, or when exceeding a predefined step limit (Figure 3).

The observation space of the environments, which provides the current state to the agent, consists of a list that includes the distance between the robot and the target (d), the angular difference between the robot's orientation and the target position (O_c), ranging from π to $-\pi$, and a set of normalized measurements (between 0 and 1) from the robot's eight infrared sensors (as illustrated in Figure 4). These values are either derived from the simulation or calculated through requests made via the simulation API. In actual environments, the absolute position and orientation of the robot are determined through a tracking platform equipped with a camera.



(**a**) (**b**)

Figure 3. Visual comparison between the simulated (**a**) and the real environment (**b**), highlighting the consistency in layout and interactive elements.

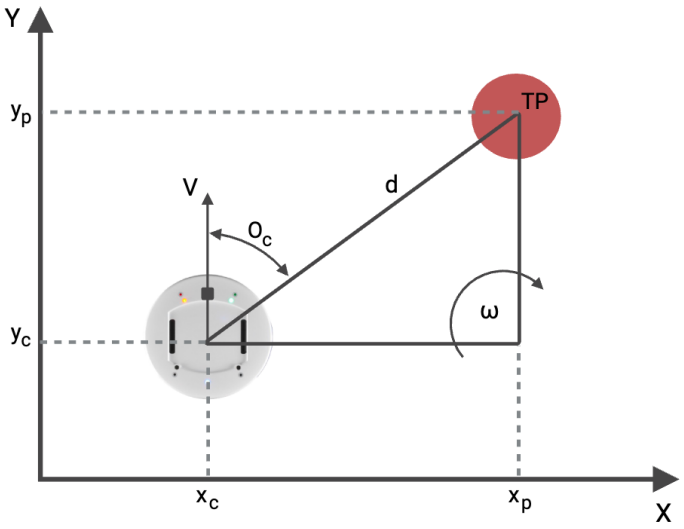


Figure 4. Key variables influencing the robot's perception and action within the environments. It illustrates measurements of distance (d) and angular orientation (O_c) relative to the TP, as well as the robot's linear (V) and angular (ω) velocities.

The action space of the environments is defined by the linear (V) and angular (ω) velocities of the mobile robot, reflecting the fundamental kinematic model of a differential drive robot. This framework allows for precise control over the robot's movements, as supported by the well-established kinematic equations cited in [45]. The linear velocity facilitates forward motion, while the angular velocity governs the rotation around the robot's central axis. The control signals for both linear and angular velocities are normalized to ranges of 0 to 1 and -1 to 1, respectively, ensuring compatibility with the simulation environment.

The reward function, depicted in Equation (4), is structured to reward proximity to the TP (distance d) and penalize collisions, thereby promoting a strategic and cautious approach. The reward increases as the robot nears the TP and adjusts negatively if sensor readings indicate proximity to obstacles. Parameters $R_{collision}$ and $R_{arrival}$ were empirically calibrated to enhance the learning and operational efficacy of the robot, with assigned values of -10 and 10, respectively.

$$Reward = \begin{cases} R_{arrival} & \text{if the robot reaches the TP} \\ R_{collison} & \text{if the robot collides} \\ -d^2 - \sum sensors & \text{in another case} \end{cases} \quad (4)$$

5.2. Laboratory Platform

Control over the Khepera IV within the lab is managed through a socket communication protocol, where the computer serves as a server, running a Python script, and the robot as a client, utilizing a script programmed in the C programming language. This arrangement employs the 'khepera4toolbox' [46] library, enabling efficient interaction with

the robot's components. The use of this library facilitates a reliable bidirectional exchange of sensor data and motor control commands between the server and the robot, ensuring precise control over the robot's movements and interactions with its environment, as depicted in Figure 5.

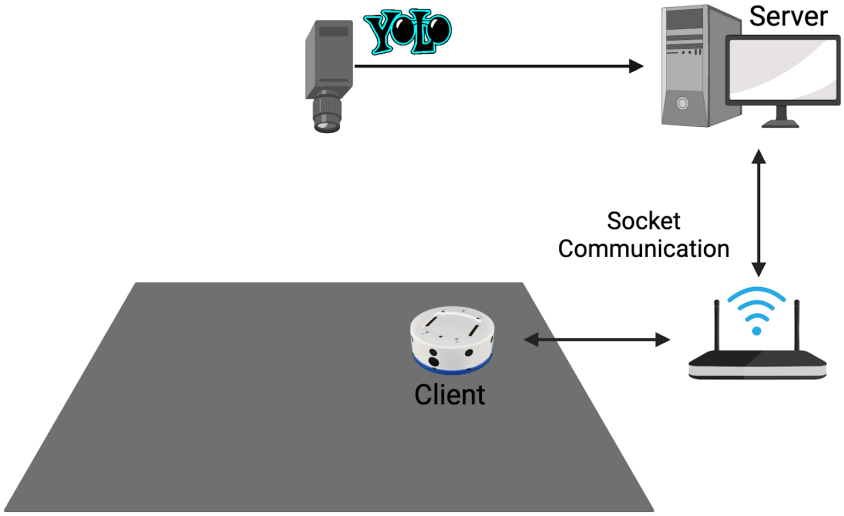


Figure 5. Schematic of the laboratory platform's component connections.

The robot's position within the platform is tracked using a webcam to provide input for a YOLO v8 [47] model trained on a dataset of 2000 manually labeled images of the Khepera on the platform. This model performs pose estimation tasks for the robot. Upon detection within an image, YOLO provides a bounding box from which the robot's (x, y) coordinates and three key points are extracted, as referenced in Figure 6. Through the application of trigonometric formulas, the relative positions of these points enable the calculation of the robot's angular orientation with respect to its surroundings. A black rectangle indicating the robot's front has been added to assist YOLO in determining the robot's orientation.

Figure 6. Keypoints the Khepera. Number 1 indicates the left marker, number 2 indicates the front marker, and number 3 indicates the right marker. These markers are used to calculate the robot's (x, y) coordinates and angular orientation through pose estimation tasks performed by the YOLO v8 model.

The model underwent training for 137 epochs, requiring 3.1 h to converge. It achieved a bounding box mAP50-95 of 0.936 and a pose estimation mAP50-95 of 0.993. These values reflect the model's proficiency in both object localization and pose estimation tasks for the robot within the images, as illustrated in Figure 7.

Given the infrared sensors' imprecisions on the Khepera and the simulation's assumption of linear measurement, detailed sensor calibrations were conducted. The results were plotted to form a sensor measurement curve, as illustrated in Figure 8. To maximize consistency between the simulated environment and the real platform, a process of linear interpolation and normalization was implemented, resulting in an adjusted measurement akin to that of the simulated infrared sensor.

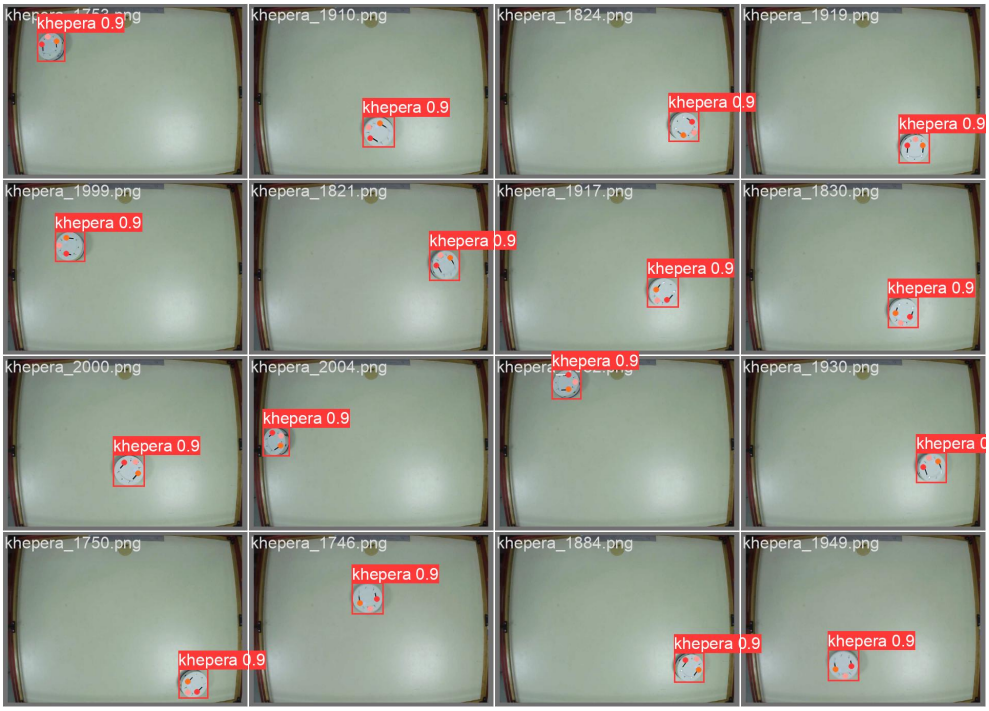


Figure 7. Model predictions for the validation set.

Integration and programming of all these interactions are carried out using the Gym library, identical to that used in the simulated environment. This ensures a seamless and coherent transition between virtual training and physical testing, establishing the necessary synergy for successful policy transfer to the Khepera IV in an operational context.

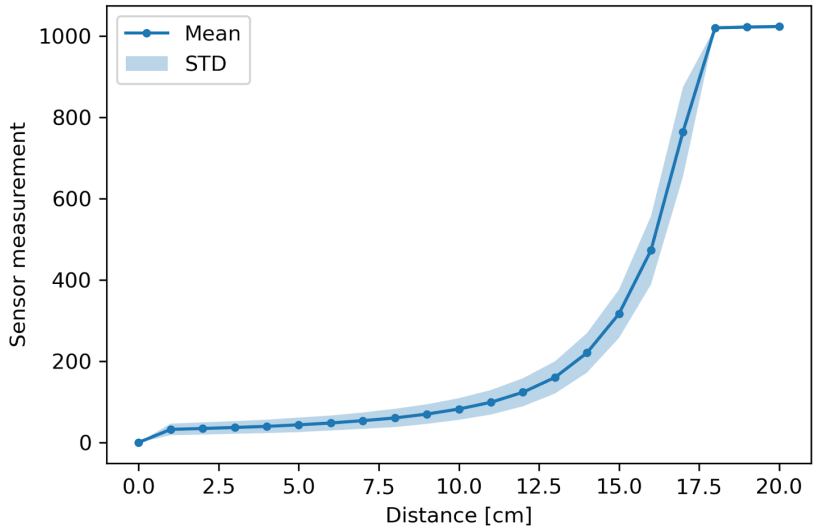


Figure 8. Mean measurement and standard deviation of the Khepera's infrared sensors.

5.3. Sim2Real: Bridging Platforms

The Sim2Real process ensures a seamless transition of control policies from the simulated environment to the physical platform. This approach involves training RL agents within the virtual CoppeliaSim environment, then transferring the learned weights to the real-world Khepera IV robot. By maintaining consistency in software and methodology across both platforms, the transition enables the effective application of sophisticated navigation and obstacle avoidance strategies developed in simulation (Figure 9).

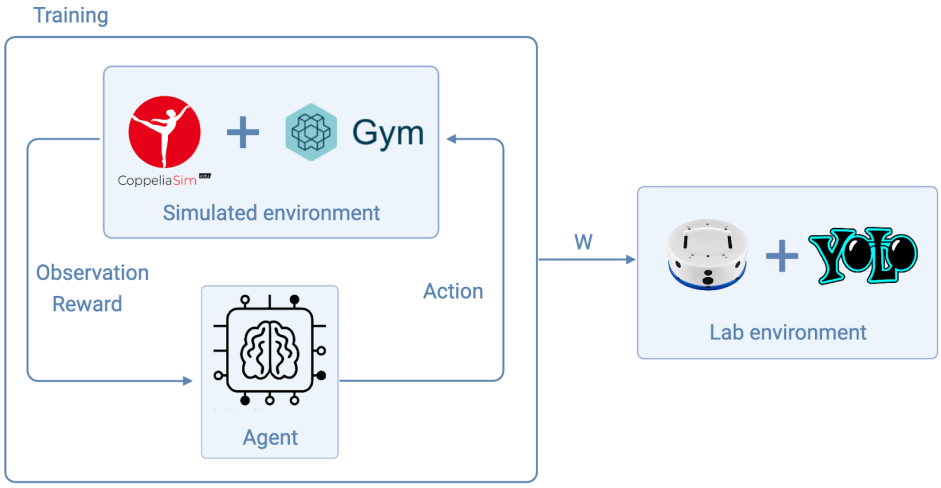


Figure 9. Diagram of the Sim2Real learning transfer process.

Strategies to Improve Sim2Real Transfer

Implementing Sim2Real transfer poses several challenges, including the reality gap between simulated and real environments and the need for robustness in handling uncertainties. To address these challenges and ensure a smoother transition of control policies from simulated environments to real-world applications, several techniques and strategies were implemented during the research. One key technique employed was domain randomization, aimed at bridging the reality gap and improving generalization. This involved systematically varying properties such as physics parameters and obstacle placement within the simulation environment. Random positions and orientations of the robot were introduced for each training episode, exposing the model to a diverse range of simulated scenarios and facilitating the learning of a more generalizable and adaptable policy.

Another crucial strategy utilized was the incorporation of uncertainty into the model. This was achieved through the integration of techniques for handling uncertainty, particularly focusing on model-based reinforcement learning approaches. By embedding these techniques into the model, it became more adept at adapting to uncertainty in the decision-making process, thereby enhancing its resilience to unforeseen situations encountered during deployment.

Furthermore, efforts were made to ensure consistency and compatibility between the simulated and real environments through normalization and standardization. Adjustments were made to the codebase to normalize variables and standardize sensor measurements, ensuring that the model's training data accurately reflected the conditions it would encounter during deployment. These techniques collectively contributed to improving the effectiveness and robustness of Sim2Real transfer, paving the way for more successful applications of reinforcement learning in real-world scenarios.

Applying Sim2Real to a model trained in a simulated environment and using it in a real environment presents significant challenges due to the reality gap and the need for generalization and adaptability. The effectiveness of the transfer will depend on how these challenges are addressed using techniques such as domain randomization, and by incorporating uncertainty into the model. These strategies can help mitigate the effects of real-world environment variability and uncertainty, improving the effectiveness and applicability of the model in real-world applications.

6. Experiments and Results

This section outlines the experiments conducted on position control and obstacle avoidance with the Khepera IV robot and provides a thorough discussion on the training process and subsequent performance analysis of the RL agents.

6.1. Simulated Robot Training Results

Two distinct RL models, PPO and DDPG, were trained over a span of 2 million timesteps, translating to a total training duration of approximately 27.3 h for PPO and 31.3 h for DDPG, respectively. For the DDPG agent, training was conducted using default parameters from the Stable Baselines3 library, with a learning rate of 0.001, buffer size of 1,000,000, learning initiation at the 100th iteration, batch size of 256, τ set to 0.005, and a γ value of 0.99. These parameters were optimized to balance exploration and exploitation, ensuring stable and efficient learning. Similarly, the PPO agent was configured with a learning rate of 0.0003, 2048 steps per update, batch size of 64, and training across 10 epochs. Additional settings included a γ of 0.99, Generalized Advantage Estimation (GAE) λ of 0.95, and a policy clipping range of 0.2, also tailored to foster both stability and efficiency in learning. The performance of each agent is graphically depicted in Figure 10, where the training graph illustrates the mean episode reward over the course of the training period. The primary objective of the training in simulation is to facilitate transfer learning (Sim2Real) for seamless deployment and adaptation in the real-world platform.

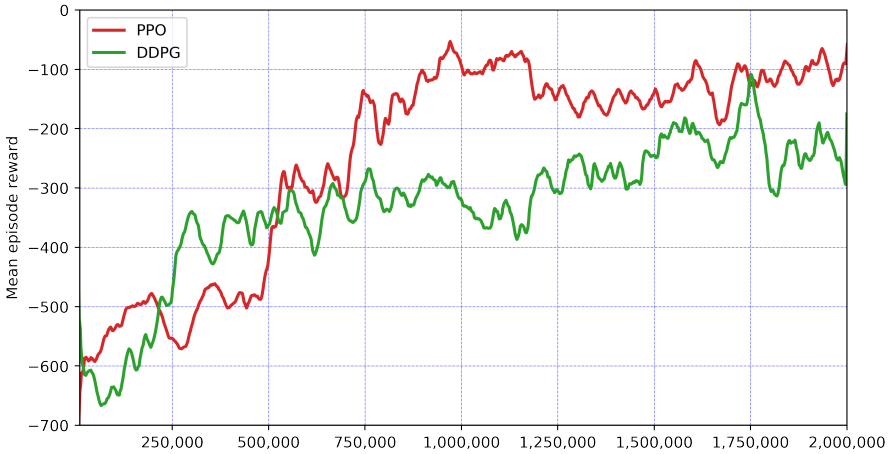


Figure 10. Training progression—mean episode reward graph.

From the graph, it is evident that both agents were capable of effectively learning from the environment, as demonstrated by the gradual increase in the reward over time. Notably, the PPO agent exhibited a particular enhancement in performance, achieving higher rewards in the initial stages of the training process. This suggests that the PPO agent was able to learn more quickly and efficiently compared to the DDPG agent, as can be inferred from the steeper slope of improvement in the corresponding curve.

The graphical analysis indicates a robust learning curve for both models, with the PPO agent demonstrating a consistent and superior performance throughout the training period. The DDPG agent, while showing a steady increase in rewards, lagged slightly behind the PPO, particularly in the early phases. However, both agents ultimately converged adequately, showcasing the efficacy of the RL models in navigating the simulated environment and addressing the complex task of autonomous movement and obstacle negotiation. It is noteworthy that the weights of the models used for the subsequent experiments correspond to those achieving the best rewards and embodying the desired action policy.

6.2. First Experiment: Navigating an Obstacle-Free Environment

In the primary experiment, the capabilities of the RL models, PPO, and DDPG, were evaluated to guide a mobile robot towards a set target within a clear space. This foundational test of positional control was designed to simulate the basic task of beach cleaning, challenging the agents to execute precise navigational maneuvers without the complexity of obstacles. As depicted in Figure 11, the simulation environment places the Khepera IV robot at an opposing start point to the target, set one meter apart, demanding precision in

execution for successful task completion, analogous to directing a beach-cleaning robot to specific locations on the sand.

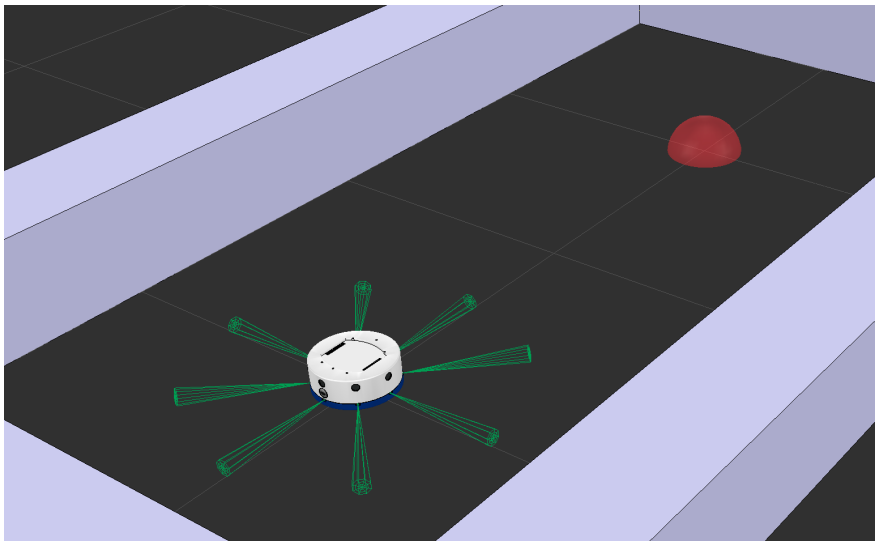


Figure 11. Simulated environment of the first experiment.

Figure 12 illustrates the trajectory paths traversed by the DDPG and PPO agents in simulated and actual environments. The paths showcase the strategic movements from the starting point at the origin (0,0), marked by the robot, to the target at (1,0), highlighted by a red circle. The robot's initial position, oriented away from the target, required a calculated turn, as evidenced by the early curvature of the paths, before proceeding directly towards the endpoint. This maneuvering efficiency is crucial for a beach-cleaning robot that needs to navigate to various locations on a beach while avoiding static and dynamic obstacles.

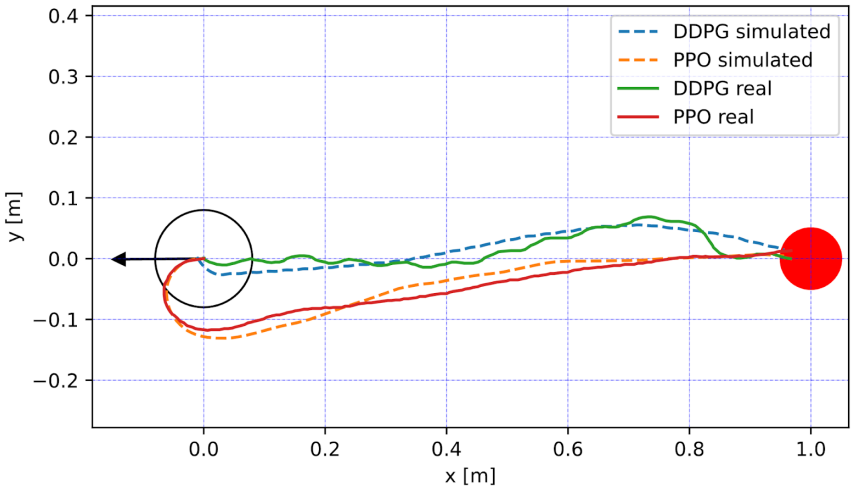


Figure 12. Trajectory plot of the first experiment.

The trajectory graph affirms the success of the Sim2Real transfer process, as both agents reliably achieved the core objective of reaching the target point. This successful transfer from simulation to real-world performance is promising for ongoing research and indicative of the potential applicability of the RL strategies to the domain of autonomous beach-cleaning robots. The ability to maintain consistent performance across both domains suggests that the refined control policies developed through simulation could effectively translate to a beach-cleaning robot navigating in a natural and less predictable environment.

In this work, the evaluation of the algorithms is structured around four specific performance indices, which are analyzed in scenarios both with and without obstacles. These

indices include the Integral of Absolute Error (IAE), which measures the total magnitude of error; the Integral of Squared Error (ISE), assessing the cumulative squared error; the Integral of Time Multiplied by Absolute Error (ITAE), reflecting the error magnitude over time; and the Integral of Time Multiplied by Squared Error (ITSE), which combines the error's square with its duration. For these performance indices, the error used is the distance (d) from the robot to the TP. These metrics provide a comprehensive framework for quantifying the effectiveness of the algorithms under varying conditions [48].

$$IAE = \int_0^\infty |e(t)|dt \tag{5}$$

$$ISE = \int_0^\infty e^2(t)dt \tag{6}$$

$$ITAE = \int_0^\infty te^2(t)dt \tag{7}$$

$$ITSE = \int_0^\infty t|e(t)|dt \tag{8}$$

Table 1 presents a comparison of performance metrics. The purpose of this comparative table is to establish a metric for evaluating the agent's adaptability to different environments. Specifically, when the performance indices exhibit similarity or equality between the agent in the real and simulated environments, it indicates that the agent's behavior remains consistent across both settings. This alignment in performance metrics signifies a successful Sim2Real transfer, suggesting that the agent is well-adjusted to the complexities of the real-world environment.

Table 1. Performance index comparison of the first experiment.

	Real		**Simulated**	
Index	**DDPG**	**PPO**	**DDPG**	**PPO**
ISE	**9.48**	11.16	**9.50**	11.00
IAE	**12.90**	14.25	**12.84**	14.15
ITSE	**57.64**	70.84	**59.03**	72.03
ITAE	**102.37**	116.41	**102.91**	118.99

Bold values indicate the best performance (lower index indicates better performance).

The data comparison indicates that both DDPG and PPO agents are competent in directing a robot towards a target, which is fundamental for a beach-cleaning robot's task of reaching specific areas for waste collection. The slightly superior performance of the DDPG agent, as indicated by lower error metrics in both environments, may suggest that this model could be more suitable for the precision required in real-world beach-cleaning operations.

These experimental outcomes validate the robustness of the simulation environment for training RL agents and suggest a promising avenue for deploying such models in the operational context of beach-cleaning robots. The agents' consistent performance across the simulated and actual environments underscores the potential of Sim2Real transfer techniques for future applications in environmental preservation tasks, where autonomous robots can provide a sustainable solution to beach waste challenges.

6.3. Second Experiment: Navigating with Obstacles in Simulated and Real Environments

The second experiment addresses the challenge of the positional control of a mobile robot in an environment enriched with the presence of obstacles. This scenario places a target point one meter away, with the robot oriented in the opposite direction, as depicted in Figure 13, representing the simulated environment for this experimental phase. Unlike the first experiment, which was designed to assess the agents' prowess in precise maneuvers

without external complexities, this second scenario tests the agents' abilities to navigate around obstacles while maintaining effective control of their position.

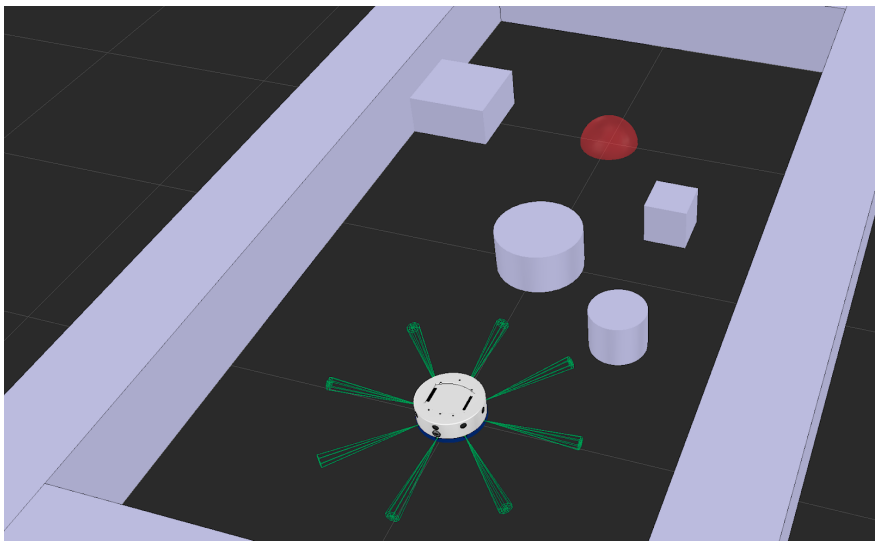


Figure 13. Simulated environment of the second experiment.

Figure 14 eloquently reveals the agents' skill in circumventing obstacles en route to the designated target. Here, a significant adaptation by the agents is observed, as they successfully navigate around the obstacles present in the scenario, a stark contrast to the obstacle-free environment of the first experiment.

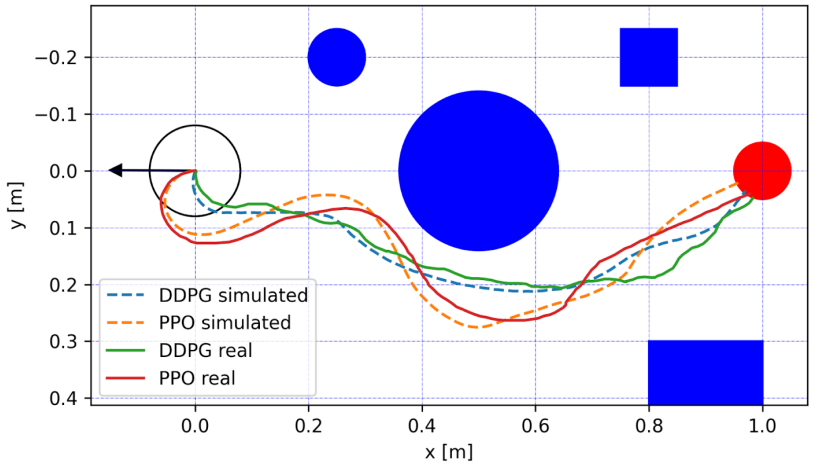


Figure 14. Trajectory graph of the second experiment.

A systematic analysis of the performance indices in Table 2 provides key insights into the agents' competencies in steering a mobile robot in an obstacle-laden environment. The DDPG agent consistently outperforms the PPO agent across all measured indices in both simulated and real-world scenarios. Lower values indicate a higher level of precision and effectiveness in guiding the robot toward the target compared to the PPO. This advantage is particularly notable in the real-world environment, where obstacles add complexity to the navigation task.

The performance indices not only affirm the efficacy of the DDPG algorithm but also highlight the reliability of the simulation environment for training RL agents in obstacle-involving situations. The superior performance of the DDPG agent suggests its potential for precise real-world robotic navigation applications, especially in environments where obstacles present navigational challenges. These findings further underscore the success

of the employed Sim2Real transfer methodologies, reaffirming their utility in tackling complex navigational tasks in practical and cluttered environments.

Table 2. Performance index comparison of the second experiment.

	Real		Simulated	
Index	**DDPG**	**PPO**	**DDPG**	**PPO**
ISE	**11.78**	12.45	**9.76**	12.16
IAE	16.71	**16.60**	**13.47**	16.27
ITSE	97.28	**95.04**	**63.66**	92.82
ITAE	178.29	**164.31**	**113.57**	159.50

Bold values indicate the best performance (lower index indicates better performance).

The outcomes of this experiment carry significant implications for the deployment of beach-cleaning robots. Unlike controlled laboratory settings, beach environments are dynamic and unpredictable, with variable elements such as wandering beachgoers, animals, and shifting terrain. The ability of the DDPG agent to adeptly handle these challenges in simulation is promising for the future application in beach-cleaning operations, where autonomous robots must operate effectively in the midst of such uncertainties. The experiment thus sets a precedent for the adaptability required in real-world beach-cleaning applications, illustrating the potential of RL-trained robots to contribute meaningfully to environmental preservation efforts.

6.4. Third Experiment: Path Following in Dynamic Beach Environments

The third experiment scrutinizes the robot's ability to precisely follow a pre-established route delineated by a sequence of control points, mimicking a beach-cleaning course. This test scenario is pivotal as it mirrors the practical application of the robot in real-life conditions, where navigational efficiency and precision are paramount.

Figures 15 and 16 visually demonstrate how the robot, under the guidance of reinforcement learning agents, adeptly maneuvers towards the control points, even amidst obstacles. Figure 15 depicts the robot's trajectory in an obstacle-free environment, underscoring the agents' ability to fluidly and accurately adhere to the desired path. In contrast, Figure 16 introduces impediments, simulating the dynamic and ever-changing beach environment where unexpected elements such as beachgoers, animals, or debris must be navigated around.

In the first two experiments, the models' ability to adapt effectively from simulation to reality was assessed. The robust convergence of performance metrics between simulated and real environments adequately supported the models' adaptability. This experiment focused exclusively on the execution of the task of following a predefined route. Given that the primary emphasis was on task execution rather than a detailed performance analysis, reward index tables were omitted to streamline result presentation and concentrate on the task execution capability.

This adaptability and precise navigation are crucial for the goals outlined in this article, which seeks to develop an autonomous robotic system for beach-cleaning tasks. The outcomes of the experiment affirm that the agents can adjust their behavior and make real-time decisions to circumvent obstacles and maintain the set route, promising indicators of their real-world applicability.

Reinforcing the Sim2Real methodology elaborated throughout the article, the experiment confirms that skills transitioned from simulated to actual environments are not only feasible but also effective. This success holds promising implications for scalability and the potential deployment of beach-cleaning robots across various beach environments.

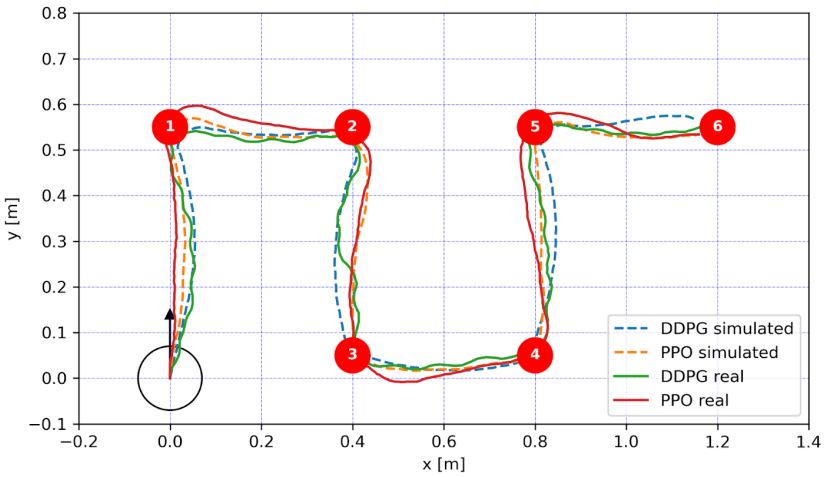


Figure 15. Trajectory graph of the third experiment without obstacles. The numbers 1 to 6 represent the key points along the trajectory followed by the robot, with 1 indicating the starting point and 6 indicating the final point.

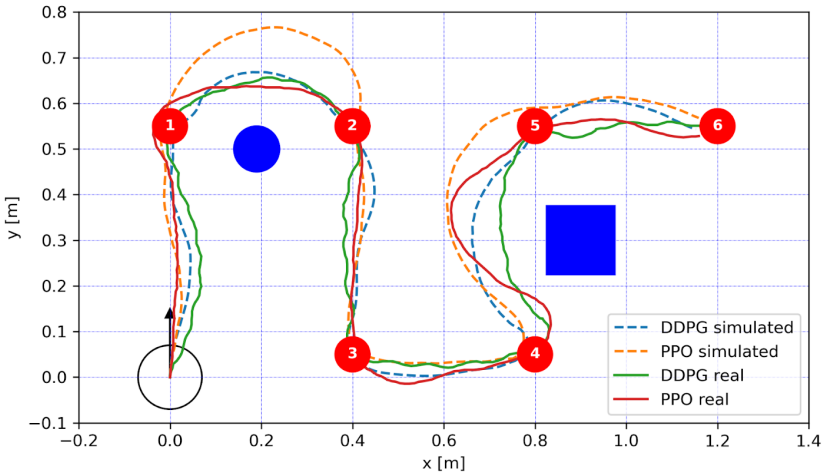


Figure 16. Trajectory graph of the third experiment with obstacles.

The demonstrated effectiveness of transferring learned skills underscores the scalability of the proposed methodology. As the robots showcase adaptability and efficiency in real-world beach landscapes, the potential for deployment in diverse environments becomes evident. The scalability of the approach positions it as a versatile solution with the capability to address the challenges posed by different beach environments, ultimately contributing to more widespread and effective deployment of autonomous beach-cleaning systems.

7. Discussion

The comprehensive experiments conducted using the PPO and DDPG agents with the Khepera IV robot platform have provided insightful data and affirmed the capabilities of these agents in both simulated and real-world settings. During the simulation phase, a commendable learning curve was observed for both agents. The PPO agent displayed an impressive rate of early learning, suggesting potential for rapid adaptation in dynamic environments. Conversely, the DDPG agent, while slower to start, demonstrated high precision in task execution, particularly when transferred to the real-world scenario. This indicates promising applications for tasks where precision is paramount.

In environments devoid of obstacles, the findings showed that both PPO and DDPG agents could navigate efficiently to reach predefined targets. The DDPG agent, however, edged out slightly in terms of precision. This advantage became more pronounced in

real-world settings, suggesting that DDPG could be particularly useful in scenarios where navigational accuracy is critical to the task at hand. When obstacles were introduced, the DDPG agent's performance remained consistent, surpassing the PPO in maneuvering through the challenges presented in both simulated and real terrains. This consistency is indicative of DDPG's robustness and makes it an optimal candidate for deployment in environments with variable and unpredictable obstacles.

The third experiment's success in path following, with both agents efficiently navigating a predefined route, solidifies confidence in the RL agents' ability to perform complex navigation tasks. This outcome is particularly pertinent to the ultimate goal of developing autonomous robots for beach cleaning, where precise route adherence is necessary.

8. Conclusions

The outcomes of this research firmly position reinforcement learning as a transformative force in the evolution of autonomous beach-cleaning robots. The rigorous experimental design has enabled the PPO and DDPG agents to learn effectively in a simulated environment and successfully transfer and adapt these skills to real-world scenarios. This seamless transition from simulation to reality, known as Sim2Real, underscores the practicality of the approach and the agents' robust performance under varying conditions. The DDPG agent's consistent outperformance in environments rich with obstacles showcases its potential as an asset in precision-demanding beach-cleaning tasks. It highlights the agent's ability to navigate through complex terrains, reflecting its readiness for real-world applications where unpredictability is a norm.

Looking ahead, the Sim2Real methodology will continue to play a central role in future developments. It facilitates efficient utilization of computational resources during the training phase and mitigates risks associated with real-world testing. By bridging the gap between digital and physical realms, Sim2Real has established itself as a cornerstone of innovation in deploying RL agents for ecological tasks. These advancements encourage further exploration into the intricacies of beach ecosystems, fostering a collaborative synergy between artificial intelligence and environmental conservation. The vision is to harness the power of Sim2Real to deploy autonomous beach-cleaning robots that are not only effective in maintaining coastal cleanliness but are also adept at preserving the natural dynamics of beach environments.

Building on the successful outcomes of this research, the next phase of the investigation will focus on scaling up the results to a larger robot equipped with advanced sensors, designed for real beach-cleaning tasks. This step aims to test and validate the effectiveness and practicality of the learned control policies and Sim2Real strategies in a more challenging and dynamic natural beach environment. This progression marks a significant move towards deploying autonomous robots for environmental conservation and beach maintenance, leveraging the full potential of reinforcement learning and Sim2Real methodologies in real-world applications.

Author Contributions: Conceptualization, F.Q. and G.H.; methodology, G.H.; software, F.Q.; validation, G.H. and F.Q.; formal analysis, F.Q.; investigation, F.Q.; resources, G.V.; data curation, F.Q. and K.S.; writing—original draft preparation, F.Q. and G.H; writing—review and editing, F.A. and G.H.; visualization, F.Q.; supervision, G.H. and G.V.; project administration, G.H.; funding acquisition, G.V. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by FONDEF under Grant ID23I10249 and FONDECYT under Grant 1240573.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data supporting the reported results can be found at the following GitHub repository: https://github.com/Fco-Quiroga/gym-kheperaposition.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Zielinski, S.; Botero, C.M.; Yanes, A. To clean or not to clean? A critical review of beach cleaning methods and impacts. *Mar. Pollut. Bull.* **2019**, *139*, 390–401. [CrossRef] [PubMed]
2. Deshpande, P.; Milkhe, O.; Kamble, A.; Kudu, N. Beach cleaning robots a comprehensive survey of technologies challenges, and future directions. *Int. Res. J. Mod. Eng. Technol. Sci.* **2023**, *5*, 7182–7188. [CrossRef]
3. Kiessling, T.; Salas, S.; Mutafoglu, K.; Thiel, M. Who cares about dirty beaches? Evaluating environmental awareness and action on coastal litter in Chile. *Ocean. Coast. Manag.* **2017**, *137*, 82–95. [CrossRef]
4. Sutton, R.S.; Barto, A.G. *Reinforcement Learning: An Introduction*; MIT Press: Cambridge, MA, USA, 2018.
5. Mnih, V.; Kavukcuoglu, K.; Silver, D.; Graves, A.; Antonoglou, I.; Wierstra, D.; Riedmiller, M. Playing Atari with Deep Reinforcement Learning. *arXiv* **2013**, arXiv:1312.5602. [CrossRef]
6. Lu, Z.; Huang, R. Autonomous mobile robot navigation in uncertain dynamic environments based on deep reinforcement learning. In Proceedings of the 2021 IEEE International Conference on Real-Time Computing and Robotics (RCAR), Xining, China, 15–19 July 2021; pp. 423–428. [CrossRef]
7. Davenport, J.; Davenport, J.L. The impact of tourism and personal leisure transport on coastal environments: A review. *Estuar. Coast. Shelf Sci.* **2006**, *67*, 280–292. [CrossRef]
8. Vieira, J.V.; Ruiz-Delgado, M.C.; Reyes-Martínez, M.J.; Borzone, C.A.; Asenjo, A.; Sánchez-Moyano, J.E.; García-García, F.J. Assessment the short-term effects of wrack removal on supralittoral arthropods using the M-BACI design on Atlantic sandy beaches of Brazil and Spain. *Mar. Environ. Res.* **2016**, *119*, 222–237. [CrossRef] [PubMed]
9. Naik, A.V.; Raj, E.V.; Chaitra, C.T.; Harshitha, K.S.; Komal. AI Based Robot for Beach Cleaning. In Proceedings of the 2023 International Conference on Applied Intelligence and Sustainable Computing (ICAISC), Dharwad, India, 16–17 June 2023; pp. 1–5. [CrossRef]
10. Schmoeller da Roza, F.; Ghizoni da Silva, V.; Pereira, P.J.; Wildgrube Bertol, D. Modular robot used as a beach cleaner. *Ingeniare Rev. Chil. Ing.* **2016**, *24*, 643–653. [CrossRef]
11. Ichimura, T.; Nakajima, S.I. Development of an autonomous beach cleaning robot "Hirottaro". In Proceedings of the 2016 IEEE International Conference on Mechatronics and Automation, Harbin, China, 7–10 August 2016; pp. 868–872. [CrossRef]
12. Thompson, R. Turtle Friendly Beach Cleaning Device. U.S. Patent 2015/014.4362 A1, 28 May 2015.
13. Praveen, R.; Prabhu, L.; Premjith, P.; Mohan, A.K.; Ajayraj. Design experimental of RF controlled beach cleaner robotic vehicle. *IOP Conf. Ser. Mater. Sci. Eng.* **2020**, *993*, 012030. [CrossRef]
14. Bano, N.; Amin, A.; Boghani, H.; Tariq, H.; Bakhtawar, S.; Waggan, I.; Younas, T. Radio Controlled Beach Cleaning Bot. In Proceedings of the 2019 IEEE 6th International Conference on Engineering Technologies and Applied Sciences (ICETAS), Kuala Lumpur, Malaysia, 20–21 December 2019; pp. 1–6. [CrossRef]
15. Deshpande, P.; Milkhe, O.; Kamble, A. Autonomous beach cleaning robot controlled by mobile application with real-time video feed and object detection. *Int. Res. J. Mod. Eng. Technol. Sci.* **2023**, *5*, 7169–7176. [CrossRef]
16. Quiroga, F.; Hermosilla, G.; Farias, G.; Fabregas, E.; Montenegro, G. Position Control of a Mobile Robot through Deep Reinforcement Learning. *Appl. Sci.* **2022**, *12*, 7194. [CrossRef]
17. Montero, E.E.; Mutahira, H.; Pico, N.; Muhammad, M.S. Dynamic warning zone and a short-distance goal for autonomous robot navigation using deep reinforcement learning. *Complex Intell. Syst.* **2024**, *10*, 1149–1166. [CrossRef]
18. Hu, H.; Wang, Y.; Tong, W.; Zhao, J.; Gu, Y. Path Planning for Autonomous Vehicles in Unknown Dynamic Environment Based on Deep Reinforcement Learning. *Appl. Sci.* **2023**, *13*, 56. [CrossRef]
19. Qiu, X.; Wan, K.; Li, F. Autonomous Robot Navigation in Dynamic Environment Using Deep Reinforcement Learning. In Proceedings of the 2019 IEEE 2nd International Conference on Automation, Electronics and Electrical Engineering (AUTEEE), Shenyang, China, 22–24 November 2019; pp. 338–342. [CrossRef]
20. Tammewar, A.; Chaudhari, N.; Saini, B.; Venkatesh, D.; Dharahas, G.; Vora, D.; Patil, S.; Kotecha, K.; Alfarhood, S. Improving the Performance of Autonomous Driving through Deep Reinforcement Learning. *Sustainability* **2023**, *15*, 3799. [CrossRef]
21. Kiran, B.R.; Sobh, I.; Talpaert, V.; Mannion, P.; Sallab, A.A.A.; Yogamani, S.K.; Pérez, P. Deep Reinforcement Learning for Autonomous Driving: A Survey. *arXiv* **2020**, arXiv:2002.00444. [CrossRef]
22. Çetin, E.; Barrado, C.; Muñoz, G.; Macias, M.; Pastor, E. Drone Navigation and Avoidance of Obstacles Through Deep Reinforcement Learning. In Proceedings of the 2019 IEEE/AIAA 38th Digital Avionics Systems Conference (DASC), San Diego, CA, USA, 8–12 September 2019; pp. 1–7. [CrossRef]
23. Tan, Z.; Karaköse, M. A new approach for drone tracking with drone using Proximal Policy Optimization based distributed deep reinforcement learning. *SoftwareX* **2023**, *23*, 101497. [CrossRef]
24. Azar, A.T.; Koubâa, A.; Mohamed, N.A.; Ibrahim, H.A.; Ibrahim, Z.F.; Kazim, M.; Ammar, A.; Benjdira, B.; Khamis, A.M.; Hameed, I.A.; et al. Drone Deep Reinforcement Learning: A Review. *Electronics* **2021**, *10*, 999. [CrossRef]
25. Fu, Z.; Zhao, T.Z.; Finn, C. Mobile ALOHA: Learning Bimanual Mobile Manipulation with Low-Cost Whole-Body Teleoperation. *arXiv* **2024**, arXiv:2401.02117. [CrossRef]
26. Petrovic, O.; Schäper, L.; Roggendorf, S.; Storms, S.; Brecher, C. Sim2Real Deep Reinforcement Learning of Compliance-based Robotic Assembly Operations. In Proceedings of the 2022 26th International Conference on Methods and Models in Automation and Robotics (MMAR), Międzyzdroje, Poland, 22–25 August 2022; pp. 300–305. [CrossRef]

27. Huang, J.; Zhang, Y.; Giardina, F.; Rosendo, A. Trade-off on Sim2Real Learning: Real-world Learning Faster than Simulations. *arXiv* **2022**, arXiv:2007.10675. [CrossRef]
28. Li, D.; Okhrin, O. A Platform-Agnostic Deep Reinforcement Learning Framework for Effective Sim2Real Transfer in Autonomous Driving. *arXiv* **2023**, arXiv:2304.08235. [CrossRef]
29. Li, D.; Okhrin, O. Vision-based DRL Autonomous Driving Agent with Sim2Real Transfer. *arXiv* **2023**, arXiv:2305.11589. [CrossRef]
30. Bawo, B. Optimizing The Output Energy of a Vertical Axis Wind Turbine Using Deep Deterministic Policy Gradient and Proximal Policy Optimization. Master's Thesis, Sabanci University, Tuzla, Turkey, 2023.
31. Toan, N.D.; Woo, K.G. Mapless Navigation with Deep Reinforcement Learning based on The Convolutional Proximal Policy Optimization Network. In Proceedings of the 2021 IEEE International Conference on Big Data and Smart Computing (BigComp), Jeju Island, Republic of Korea, 17–20 January 2021; pp. 298–301. [CrossRef]
32. You, S.; Diao, M.; Gao, L.; Zhang, F.; Wang, H. Target tracking strategy using deep deterministic policy gradient. *Appl. Soft Comput.* **2020**, *95*, 106490. [CrossRef]
33. Yuan, Z.; Hall, A.W.; Zhou, S.; Brunke, L.; Greeff, M.; Panerati, J.; Schoellig, A.P. Safe-control-gym: A Unified Benchmark Suite for Safe Learning-based Control and Reinforcement Learning in Robotics. *arXiv* **2022**, arXiv:2109.06325. [CrossRef]
34. Gu, S.; Yang, L.; Du, Y.; Chen, G.; Walter, F.; Wang, J.; Yang, Y.; Knoll, A. A Review of Safe Reinforcement Learning: Methods, Theory and Applications. *arXiv* **2023**, arXiv:2205.10330. [CrossRef]
35. Yang, W.C.; Marra, G.; Rens, G.; Raedt, L.D. Safe Reinforcement Learning via Probabilistic Logic Shields. *arXiv* **2023**, arXiv:2303.03226. [CrossRef]
36. Chen, J., Zhang, K., Wang, J., Shen, W. Closing the Simulation-to-Reality Gap for Digital Twin-Assisted Fault Diagnosis: Sim2real Knowledge Transfer with Contrastive Learning. 2024. Available online: https://ssrn.com/abstract=4699149 (accessed on 23 May 2024).
37. Raffin, A.; Hill, A.; Gleave, A.; Kanervisto, A.; Ernestus, M.; Dormann, N. Stable-Baselines3: Reliable Reinforcement Learning Implementations. *J. Mach. Learn. Res.* **2021**, *22*, 1–8.
38. Lillicrap, T.P.; Hunt, J.J.; Pritzel, A.; Heess, N.; Erez, T.; Tassa, Y.; Silver, D.; Wierstra, D. Continuous control with deep reinforcement learning. *arXiv* **2019**, arXiv:1509.02971. [CrossRef]
39. OpenAI Spinning Up. Proximal Policy Optimization. 2020. Available online: https://spinningup.openai.com/en/latest/algorithms/ppo.html (accessed on 1 March 2024).
40. Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; Klimov, O. Proximal Policy Optimization Algorithms. *arXiv* **2017**, arXiv:1707.06347. [CrossRef]
41. Tharin, J.; Lambercy, F.; Carron, T. *Khepera IV User Manual*; K-Team S.A.: Vallorbe, Switzerland, 2019. Available online: https://ftp.k-team.com/KheperaIV/software/Gumstix%20COM%20Y/UserManual/Khepera%20IV%20User%20Manual%204.x.pdf (accessed on 1 March 2024).
42. Farias, G.; Fabregas, E.; Peralta, E.; Torres, E.; Dormido, S. A Khepera IV library for robotic control education using V-REP. *IFAC-PapersOnLine* **2017**, *50*, 9150–9155. [CrossRef]
43. Coppelia Robotics. *CoppeliaSim User Manual*. 2023. Available online: https://manual.coppeliarobotics.com/index.html (accessed on 1 March 2024).
44. K-Team. KHEPERA IV. Available online: https://www.k-team.com/khepera-iv (accessed on 1 March 2024).
45. Peralta, E.; Fabregas, E.; Farias, G.; Vargas, H.; Dormido, S. Development of a Khepera IV Library for the V-REP Simulator. *IFAC-PapersOnLine* **2016**, *49*, 81–86. [CrossRef]
46. Soares, J. Khepera IV Toolbox. 2018. Available online: https://github.com/jsoares/khepera4toolbox (accessed on 1 March 2024).
47. Reis, D.; Kupec, J.; Hong, J.; Daoudi, A. Real-Time Flying Object Detection with YOLOv8. *arXiv* **2023**, arXiv:2305.09972. [CrossRef].
48. Shinners, S.M. *Modern Control System Theory and Design*, 2nd ed.; John Wiley & Sons: Hoboken, NJ, USA, 1998.