IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received month day, year, accepted month day, year, date of publication month day, year,
date of current version September month day, year.

Digital Object Identifier 10.1109/ACCESS.2021.DOI

Sim-to-Real Transfer in Robotics Navigation: A Survey

Iure Rosa L. Oliveira[1], Ricardo S. Ferreira[1,2], (Member, IEEE), and Alexandre S. Brandão[1,3]
[1]Graduate Program on Computer Science, Federal University of Viçosa, Viçosa 36570-900, Brazil
[2]Department of Informatics, Federal University of Viçosa, Viçosa 36570-900, Brazil
[3]Department of Electrical Engineering, Federal University of Viçosa, Viçosa 36570-900, Brazil

Corresponding author: Alexandre S. Brandão (e-mail: iure.oliveira@ufv.br).

This work was supported by CAPES - Coordenação de Aperfeiçoamento de Pessoal de Nível Superior, CNPq – Conselho Nacional de Desenvolvimento Científico e Tecnológico, Brazilian Ministry of Science, Technology, Innovations and Communications, and FAPEMIG – Fundação de Amparo à Pesquisa de Minas Gerais, Brazil.

ABSTRACT Deep Reinforcement Learning is an effective way to train robots to adapt to the real-world as it overcomes the problem of data source sample inefficiency and the cost of collection. It provides a potentially infinite source of data as the agent explores the environment and exploits the knowledge learned from its exploration. However, there is a remarkable degradation in performance observed in transitioning from a simulated environment to the real world. This warrants a deeper look into the efficient policy transfer methods which closes the gap between the simulation and the real world. This survey article summarizes sim-to-real transfer fundamentals and gives an overview of the main methods applied in this area: domain randomization, domain adaptation, imitation learning, meta-learning, and knowledge distillation. It also highlights some of the most recent works and their application scenarios. In the end, we look at the challenges and areas of future research in the domain.

INDEX TERMS Sim-to-Real, Knowledge Transfer, Deep Reinforcement Learning, Robotics Navigation.

I. INTRODUCTION

OVER the past years, the robotics community has increasingly adopted reinforcement learning (RL) algorithms for controlling complex robots, multi-robot systems, and developing end-to-end policies for perception and control [1]–[3]. Inspired by human learning through trial-and-error processes, RL algorithms acquire knowledge based on the rewards agents receive when they take certain actions in different experiences. However, this learning process requires a large number of episodes, leading to limitations in terms of time and experience variability in real-world scenarios. Furthermore, learning with real robots raises concerns about potentially dangerous or unexpected behaviors, particularly in safety-critical applications [4], [5].

While deep reinforcement learning (DRL) algorithms have demonstrated success in various simulation environments, their effectiveness beyond simulated worlds has been limited, except for certain robotic tasks involving obstacle avoidance [6], [7]. This survey focuses on reviewing relevant works that address a key research question: how to leverage simulation-based training to transfer knowledge and adapt policies in real-world settings. Figure 1 illustrates the process of sim-to-real transfer. Simulation-based training provides low-cost data but introduces inherent mismatches with real-world settings. Bridging the gap between simulation and reality necessitates methods that can account for mismatches in both sensing and actuation [8]–[10].

Recent studies in the field of deep learning have extensively investigated the sensing aspect, such as adversarial attacks on computer vision algorithms, to address the mismatches between simulation and reality. Another aspect is minimizing the risks associated with actuation through more realistic simulation. Approaches to address these challenges include introducing perturbations in the environment or focusing on domain randomization [11]–[13]. Additionally, it is important to consider that agents deployed in the real world may encounter novel experiences not present in the simulations [10], requiring policy adaptation to encompass a wider range of tasks. Meta learning, continual learning, and other approaches have been proposed to bridge this gap.

While the aforementioned methods primarily focus on extracting knowledge from simulation-trained agents for deployment in real-life scenarios, other approaches aim to directly deploy robotic agents into the real world using sim-

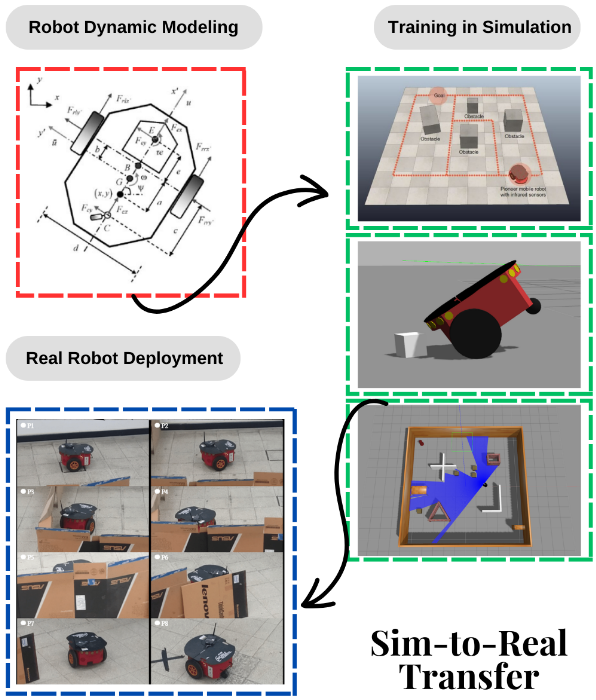


FIGURE 1. Conceptual view of a simulation-to-reality transfer process.

ulators that provide training data and experiences with minimal mismatches between real and simulated settings. Recent efforts have also focused on enhancing safety during training in real-world settings, as safety remains a major challenge in achieving online training of complex agents, including robot arms and self-driving cars. Some promising results have been achieved in safe DRL, ensuring convergence while reducing the exploration space [14]. This survey does not cover specific simulators or techniques for direct learning in real-world settings but rather concentrates on describing the main methods for transferring knowledge learned in simulation robotics platform to real-world.

This survey emphasizes the main application fields of current research efforts and takes a broader perspective by including related research directions in transfer learning, domain adaptation, knowledge distillation, and meta reinforcement learning. While previous surveys have focused on transfer learning techniques or safe reinforcement learning, this survey provides a distinct viewpoint with an emphasis on DRL policy transfer in the robotics domain. Although numerous publications have deployed DRL policies on real robots, this survey specifically focuses on works that tackle sim-to-real transfer issues. The emphasis is primarily on end-to-end approaches, but relevant research on applying sim-to-real transfer techniques to the sensing aspects of robotic operation, particularly the transfer of deep learning vision algorithms to real robots, is also described.

The remaining sections of this paper are organized as follows: Section II presents the search method employed in this study. Section III provides a brief introduction to the main approaches of DRL, along with relevant research directions encompassing knowledge distillation, transfer learning, adaptation, and meta-learning. Section IV explores various strategies adopted to address the simulation-to-reality gap, while Section V and VI concentrate on the significant application domains. Section VII presents a bibliometric analysis, and Section VIII discusses open issues. Finally, Section IX presents our conclusion.

II. THE SEARCH METHOD

A systematic search was implemented to identify and select related literature works. The search method for technical papers or surveys and reviews on the subject studied in this paper, Sim-to-Real Transfer in Robotics Navigation, follows the topics and criteria presented in Table 1. Main expressions such as “Sim-to-Real”, “DRL”, “autonomous navigation”, “RL”, “UGV”, and “UAV”, among others, were adopted, considering a 5-years period, from 2018 to 2023, to select papers. It is important to clarify that the short period of publication is due to the fact that the application of Sim-to-Real techniques in the robot navigation scenario is a recent topic, although continuously explored, and it presents various problems to design practical experiments and online learning.

As shown in Figure 2, we splitted up the search in three main topics: the robot type and configuration, the navigation related aspects, and transfer methods used in this scenario, in special reinforcement and deep reinforcement learning techniques. We also restricted the search results to peer-reviewed documents, such as journals and conference papers, to ensure the authenticity and quality of the outcomes. This premise is considered because the existing gap between simulation and real-world, which requires an accurate mathematical methods to describe their dynamics and to design their controllers. Modeling and control demands further increase depending on the environmental conditions and mission requirements. Thus, practical experiments or complete dynamic models to validate the simulations and the adopted control techniques were mandatory for a work to be included in this survey.

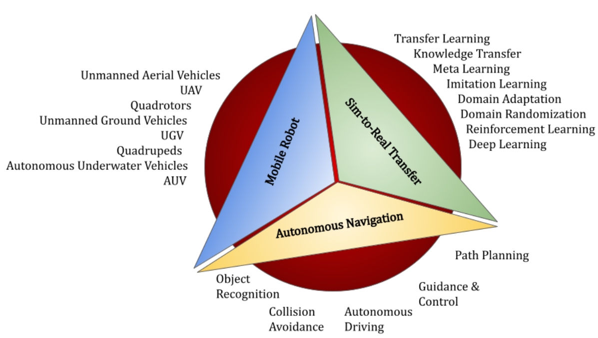


FIGURE 2. Main keywords used in the search method adopted in this survey.

Some exceptions were considered, such as works that present motion primitives such as "go forward" or "turn right", etc., as their goal was not to propose a control strategy considering the robot dynamics, but to evaluate the use of ML techniques for a specific application.

Finally, a manual search across the list of selected papers was performed, looking for relevant studies on the subject. Therefore, works whose results have been considered most interesting, using the criteria of relevance to the field, technical quality and originality, are here presented and critically analyzed.

III. THEORETICAL BACKGROUND

To have a better understanding of all the technical aspects faced by Sim-to-Real in DRL for robotics navigation discussed in this work, and to make the reading more comfortable, the following technical aspects are explained: first, what is Deep Reinforcement Learning; second, the concept of transfer learning and the Sim2Real problem; and, finally, a brief review about few methods intersect with this aim, including robust RL, and Meta Learning. The relationship between there concepts is illustrated in Figure 3.

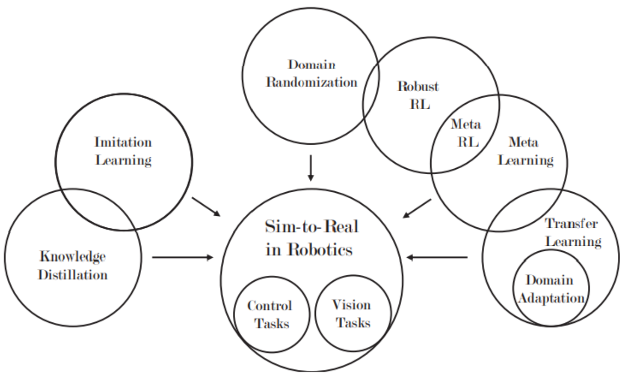


FIGURE 3. Illustration of the different methods related to sim-to-real transfer in deep reinforcement learning and their relationships.

A. REINFORCEMENT LEARNING

Reinforcement Learning is a kind of machine learning that enables an agent to learn how to act in an interactive environment to maximize sum of expected cumulative rewards. An agent (i.e., robot) can learn a policy itself using feedback on its actions. The RL can be applied to solve sequential decision-making under uncertainty, and this is formulated as the Markov Decision Process (MDP).

Different from supervised learning and unsupervised learning method, RL uses a feedback signal (reward) instead of a large number of labeled samples. Classical motion planning strategies (such as artificial potential field method, A^* algorithm, etc.) are based on the environment with certain information, and usually have certain limitations in the application process. MDP provides a unified model description for the decision-making and solving process in the environment with incomplete information, which can solve the motion planning problem in the unstructured environment well.

As shown in Figure 4. MDP includes the action, reward, and state [15]. The agent represents the mobile robot; the environment refers to the map information of the task; the state is the state space of the mobile robot; and the action is a set of actions of a mobile robot.

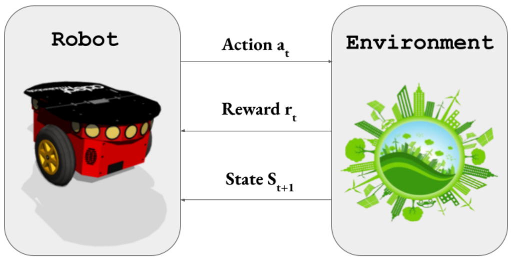


FIGURE 4. MDP for motion planning.

The goal of reinforcement learning is to find a strategy that maximizes cumulative rewards of the robot. Different from the immediate reward, the cumulative reward evaluates the long-term impact of a certain strategy on the performance of agents in the environment. The commonly used calculation methods include discount cumulative reward and step cumulative reward. When the reward value is not easy to set, the Inverse Reinforcement Learning can be used to learn the reward value function.

In contrast to conventional motion planning methods, the mobile robot is not told which action to choose, but instead tries to find the one that will get the most rewards in the long term. Motion planning can be viewed as a process of trial evaluation. Firstly, the agent selects an action for the environment and changes the state after the environment accepts the action. The action selection of the current state s is determined by the strategy $\pi(a|s) : \pi(a|s) = P(A_t = a|S_t = s)$, and it represents the conditional probability distribution of action selection, and the effect is evaluated by the value function $V_\pi(s)$ [16]:

$$V_\pi(s) = \mathbb{E}_\pi(R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \cdots |S_t = s), \quad (1)$$

where $\gamma \in [0, 1]$ is the discount factor, between $[0, 1]$.

Considering the influence of action A on the value function, the action value function $Q_\pi(s, a)$ is defined as [17].

$$Q_\pi(s, a) = \mathbb{E}_\pi(R_{t+1}+\gamma R_{t+2}+\gamma^2 R_{t+3}+\cdots |S_t = s, A_t = a). \quad (2)$$

According to the definition of the action-value function $Q_\pi(s, a)$ and the state value function $V_\pi(s)$, transformation relationship between them is:

$$v_\pi(s) = \sum_{a\in A} \pi(a|s)q_\pi(s, a). \quad (3)$$

TABLE 1. A summary of the search criteria adopted in this work.

Criteria	Information
Keywords	("sim-to-real" OR "DRL" OR "deep reinforcement learning" OR "transfer learning" OR "knowledge transfer" OR "meta learning" OR "imitation learning" OR "domain adaptation") **AND** ("Navigation" OR "Autonomous Flying" OR "Guidance" OR "Planning" OR "Control" OR "Situation Awareness" OR "Collision Avoidance" OR "Object Recognition") **AND** ("unmanned aerial vehicles" OR "UAV" OR "unmanned ground vehicles" OR "UGV" OR "quadrupeds" OR "quadrotors" "unmanned surface vehicles" OR "USV" OR "autonomous underwater vehicles" OR "AUV")
Scientific Database	IEEE Xplore, Google Scholar, Science Direct, Web of Science (WoS), and Engineering Village
Digital Libraries	IEEE, Elsevier, Springer, Taylor&Francis, Wiley, MDPI, AIAA, and ACM
Publication Period	May/2018 $\rightarrow$ May/2023

The purpose of robot motion planning is to find an optimal strategy for the mobile robot to consistently get more rewards during movement than any other strategy, and this optimal strategy, and this optimal strategy can be represented by $\pi^*(a|s)$:

$$\pi^*(a|s) = \begin{cases} 1, & a = argmax_{a \in A} Q_*(s,a) \\ 0, & otherwise \end{cases}, \tag{4}$$

where $Q_*(s,a) = R_s^a + \gamma \sum_{s' \in S} P_{ss'}^a max_{a'} Q_*(s',a')$.

In general, it is difficult to find an optimal strategy for motion planning, but a better strategy can be determined by comparing he advantages of several different strategies.

1) Deep Reinforcement Learning

DRL uses deep agents to learn the optimal policy where it combines Artificial Neural Networks (ANN) with Reinforcement Learning. The ANN type used in DRL varies from one application to another depending on the problem being solved, inputs type (state), and the number of inputs passed to the ANN. For example, the RL framework can be integrated with Convolutional Neural Network (CNN) to process images representing the environment states or combined with Recurrent Neural Network (RNN) to process inputs over different time steps.

The ANN loss function, also known as the Temporal Difference (TD), is generically computed by finding the difference between the output of the ANN $Q(s,a)$ and the optimal Q-value $Q_*(s,a)$ obtained from the Bellman equation as shown in (5).

$$\overbrace{E\left[R_{t+1} + \gamma \underset{a'}{max} Q_*(s',a')\right]}^{Target} - \overbrace{E\left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1}\right]}^{Predicted} \tag{5}$$

The architecture of the deep agent can be simple or complex based on the problem at hand, where a complex architecture combines multiple ANN. But what all deep agents have in common is that they receive the state as an input, then they output the optimal action and maximize the discounted return of rewards.

The application of Deep Neural Networks to the RL framework enabled the research community to solve more complex problems in autonomous systems that were hard to solve before and achieve better performance than previous state-of-the-art, such as robotics navigation and avoiding obstacles using images received from the monocular camera.

B. TRANSFER LEARNING

Reinforcement Learning algorithms are very useful for solving a wide variety problems when the model is not known in advance, with many algorithms possessing guarantees of convergence to equilibrium [18], [19]. Unfortunately, the convergence of any RL algorithm may only be achieved after an extensive exploration of the state-action space, which is usually very time consuming.

One way to speed up the convergence of RL algorithms is by making use of a heuristic function in a manner similar to the use of heuristics in informed search algorithms. Heuristically Accelerated Reinforcement Learning (HARL) methods, which have been recently proposed 20, apply a conveniently chosen heuristc function for selecting the appropriate actions to perform in order to guide exploration during the learning process.

Transfer Learning (TL) is a paradigm of machine learning that reuses knowledge accumulate in a previous task to better learn a novel, but related, target task [21] and can be characterized as a gain or loss of proficiency in a task as a result of a practice in another task previously. For example, the abilities acquired while learning to drive a car can be applied when one learns to drive truck, making the second learning task easier. Although not yet comparable to the abilities of humans, transfer learning can be a very useful tool when faster learning is needed or when other learning techniques fail.

Transfer learning aims to enhance the performance of target learners in specific domains by leveraging the knowledge acquired from different but related source domains [22]. By doing so, transfer learning can reduce the reliance on target domain data when developing target learners.

Domain adaptation is a subset of transfer learning methods that addresses situations where there is an ample amount of labeled data available in the source domain, and the target task is the same as the source task. However, there is a scarcity or limited availability of data from the target domain. In the context of sim-to-real robotics, researchers often utilize simulators to train reinforcement learning (RL) models and subsequently deploy them in realistic environments. In such cases, employing domain adaptation techniques becomes crucial to facilitate effective transfer of the simulation-based models.

A special type of transfer learning that has gaining interest in the scientific community is Curriculum Learning. This is inspired in human learning, which is frequently supervised, but also is often accompanied by a curriculum [23]. When humans teach each other, the order of the contents is not randomly chosen. A task may be divided by the teacher in smaller sub-tasks, often called shaping [24]. A well proposed curriculum can improve convergence speed and overall performance of a model.

Specially in the context of reinforcement learning and Markov Decision Process, [25] highlight 3 key factors:

i. **Task Generation**: The quality of the curriculum is directly related to the quality of the tasks available to choose from. This key factor is the process of creating a good set of sub tasks from which to obtain experience samples. These tasks can be pre determined or dynamically generated during curriculum constructing by observing the learning agent.

ii. **Sequencing**: Deciding how to partially order the set of intermediate tasks is as a key challenge as choosing the set itself. Usually this is manually defined by a human supervisor,but automated curriculum sequencing have been more explored.

iii. **Transfer Learning**: The tasks contained in the curriculum may differ in state/action space, reward function or even transition function from each other and specially from the final task. Therefore the way that knowledge is transferred from one stage to the other may not be trivial and must be examined.

Figure 5 shows examples of curricula structures.

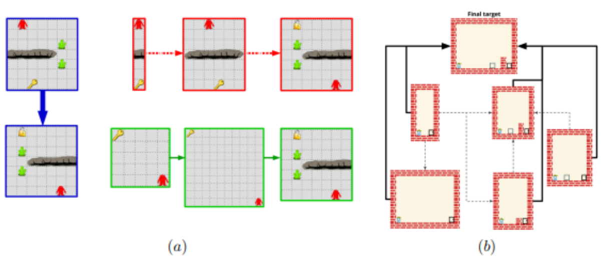


FIGURE 5. (a) Three examples of linear sequences in gridworld scenarios (b) Directed acyclic graphs in block dude, where there is more than one stage option for each stage in the curriculum. Source: [25].

In attempt to automatically generate the curriculum structure, [26] progressively initiate the learning agent further away from the predetermined destination, leading to a steady flow of more difficult tasks (longer path to goal) throughout training. In the same context, [27] use automatic generation of the curriculum in a robotic manipulator agent by employing a more loose accuracy requirement to consider a success in earlier stages, leading the agent to obtain more rewards for simple skills at first before being more severe with accuracy standards.

There is still room for improvement in the Transfer Learning field, such as better curriculum design, metrics or even automated episode generation. Also there are still challenges when it comes to integrating these techniques in reinforcement learning scenarios. This work proposes to contribute in this field as well.

1) The Sim-to-Real Problem

Transferring DRL policies from simulation environments to reality is an essential step in advancing complex robotic systems with deep learning-defined controllers. This challenge, however, is not unique to DRL algorithms but encompasses machine learning (ML) as a whole. Although many DRL algorithms provide end-to-end policies that directly process raw sensor data and generate actuation commands, these two aspects of robotics can be treated independently. Bridging the gap between simulation and reality, particularly in terms of actuation, necessitates more accurate simulators that consider the variability in agent dynamics. On the sensing side, the problem is broader, encompassing the general ML challenge of encountering real-world situations not seen in simulation. This paper primarily focuses on end-to-end models and provides an overview of research conducted on system modeling, dynamics randomization, and sensing-based randomization techniques.

2) Knowledge Distillation

DRL often involves large networks to handle high-dimensional input data, such as complex visual tasks. Policy distillation is a technique used to extract knowledge from a trained network, allowing for the training of a new network that can achieve a similar expert level while being significantly smaller and more efficient. In these scenarios, the two networks involved are commonly referred to as the teacher and the student. The student network is trained in a supervised manner using data generated by the teacher network. In a study by the authors, they introduced DisCoRL, a modular, effective, and scalable pipeline designed for continual DRL. DisCoRL has been successfully applied to multiple tasks, where knowledge from various teachers is distilled into a single student network.

C. META REINFORCEMENT LEARNING

Meta reinforcement learning (Meta-RL) is a subfield of reinforcement learning that focuses on enabling agents to learn how to learn. It explores the idea of agents acquiring the ability to adapt and generalize their learned policies to new tasks or environments. In Meta-RL, the goal is not just to optimize performance on a single task but to develop algorithms that can quickly and effectively adapt to a range of related tasks with minimal data or interaction.

Meta-RL involves training agents on a distribution of tasks and learning meta-policies that can efficiently infer and exploit underlying task structures. The key idea is to leverage prior knowledge and experience from previous tasks to improve the learning process on new tasks. This is achieved by learning representations or inductive biases that capture the underlying structure of the tasks, allowing for more efficient generalization and adaptation. Meta-RL has shown promising results in enabling agents to learn faster, achieve better performance, and exhibit more flexible behaviors, making it a promising approach for solving complex and dynamic real-world problems.

D. ROBUST RL AND IMITATION LEARNING

Robust RL is a research area within reinforcement learning that focuses on developing algorithms capable of handling uncertainties and disturbances in real-world scenarios. It aims to ensure stable and reliable performance by accounting for factors such as noisy observations, model inaccuracies, and environmental changes. Robust RL incorporates techniques such as robust optimization, robust control, and robust estimation to mitigate the impact of uncertainties and adapt to varying conditions. The ultimate goal is to enable RL agents to learn policies that are robust, adaptive, and capable of operating effectively in dynamic and uncertain environments.

Imitation learning, also known as learning from demonstrations, is a machine learning approach where a policy or behavior is learned by imitating expert demonstrations. Rather than relying solely on trial-and-error exploration, imitation learning leverages pre-existing expert demonstrations as training data. The learner aims to generalize from this dataset and acquire a policy that can reproduce similar actions or trajectories in new situations. Imitation learning is particularly useful when designing a reward function is challenging or when expert knowledge is readily available. It finds applications in domains such as robotics, autonomous driving, and game playing, allowing agents to quickly acquire expert-level behavior and serve as a foundation for further fine-tuning using reinforcement learning or other approaches.

IV. SIM-TO-REAL TRANSFER

Sim-to-real transfer research has witnessed a substantial increase in publications in recent years, spanning multiple research directions. In this section, we provide a summary of the most notable methods employed for sim-to-real transfer. Table 2 presents a compilation of relevant and recent works in this field. Domain randomization emerges as the predominant approach for learning transfer, accompanied by other notable techniques such as policy distillation, system identification, and metaRL. The realm of learning algorithms exhibits considerable diversity, including deep reinforcement learning algorithms such as Proximal Policy Optimization (PPO) [28], Trust Region Policy Optimization (TRPO) [29], Maximum a-posteriori Policy Optimization (MPO) [30], Asynchronous Actor-Critic (A3C) methods [31], Soft Actor-Critic (SAC) [32], Deep Deterministic Policy Gradient (DDPG) [33], Twin Delayed Deep Deterministic Policy Gradient (TD3) [34], and various others.

A. ZERO-SHOT TRANSFER

Transferring knowledge from simulation to reality can be achieved through two main strategies: zero-shot or direct transfer and one-shot transfer. In the zero-shot approach, a realistic simulator is constructed or sufficient simulated experience is accumulated, allowing the learned model to be directly applied in real-world settings. This strategy requires a high level of fidelity in the simulation to accurately represent the real-world dynamics.

On the other hand, one-shot transfer techniques aim to bridge the gap between simulation and reality through system identification and domain randomization. We discuss both of these separately in Sections IV-B and IV-C.

B. SYSTEM IDENTIFICATION

System identification involves building precise models of the real world, capturing its dynamics and characteristics. This approach allows for a more accurate representation of the target system, enhancing the transferability of the learned policies.

It is important to acknowledge that simulators do not provide a completely faithful representation of the real world. While system identification aims to construct accurate mathematical models for physical systems, achieving a realistic simulator requires careful calibration. However, challenges persist in obtaining a simulator that accurately captures the complexities of the real world. One such challenge is the difficulty in creating high-quality rendered images that simulate real vision. Simulating realistic visual perception in a simulated environment remains a complex task. Additionally, various physical parameters of a robot, such as temperature, humidity, positioning, and wear-and-tear over time, can significantly vary, posing additional difficulties for system identification. These variations introduce uncertainties that must be accounted for to improve the realism of the simulator.

Despite these challenges, researchers continue to strive for more realistic simulators, exploring advanced techniques in system identification and calibration. Overcoming these limitations is crucial to bridge the gap between simulation and reality and enable effective sim-to-real transfer in robotics and other domains.

C. DOMAIN RANDOMIZATION METHODS

Domain randomization is a concept that involves highly randomizing the simulation environment instead of precisely modeling all parameters of the real world. The goal is to encompass the distribution of real-world data despite the discrepancies between the simulation and reality. Figure IV-D-a illustrates the domain randomization paradigm.

There are two main categories of domain randomization methods based on the components randomized within the simulator: visual randomization and dynamics randomization. In robotic vision tasks like object localization, object detection, pose estimation, and semantic segmentation, the training data from the simulator exhibit variations in textures,

TABLE 2. The most relevant publications in Sim2Real Transfer.

Ref.	Year	Application	Algorithm	Simulator	Real Exp.	Summary
[35]	2018	Quadruped Locomotion	PPO	Bullet	Yes	A system to automate the locomotion process by leveraging DRL techniques.
[36]	2019	Visual Navigation	MPC	GibsonEnv	Yes	Perform a visual navigation while avoiding collisions with unseen objects on the path.
[37]	2019	Navigation	MLP	Duckietown	Yes	A robust framework that plans in simulation and transfers well to the real environment.
[38]	2020	Object Grasping	CycleGan + Q-Learning	Bullet	Yes	A method to automatically transfer vision-based policies from simulation with an objective (task-aware).
[39]	2020	Obstacle Avoidance	SAC	Gazebo	Yes	Using SAC algorithm to drastically reduce the need for additional training in the real world.
[40]	2020	Vision-and-Language Navigation	Not Specific	Coda	No	A model to identify nearby waypoints, and use domain randomization to mitigate visual domain differences.
[41]	2020	Manipulation, Navigation, and Control	Not Specific	MuJoCo	Yes	A novel real-sim-real transfer-method that includes a real-to-sim training phase and a sim-to-real inference phase.
[42]	2020	Quadrotor Landing	Sequencial DQN	Not Specific	Yes	The use of DRL as an end-to-end learn paradigm to find a policy for UAVs autonomous landing.
[43]	2020	Visual Navigation	Not Specific	Not Specific	Yes	A method that learns on an observation space constructed by point clouds and environment randomization.
[44]	2021	Obstacle Avoidance	DDPG+Demo+PER	Gazebo	Yes	A algorithm for use in mobile robots that map raw sensor data to linear and angular velocities and unknown scenarios.
[45]	2021	Local Navigation	PPO	Not Specific	Yes	A learning-based pipeline to realise local navigation with a quadruped robot in cluttered and dynamic environment.
[46]	2021	Cluttered Rough Terrain Navigation	Not Specific	Gazebo	Yes	The development of a novel sim-to-real pipeline for a mobile robot to effectively learn how to navigate in rough terrain.
[47]	2021	Autonomous Navigation	Dynamic-PPO-CMA	Gazebo	No	A dynamic proximal meta policy optimization to avoid obstacles and realize autonomous navigation.
[48]	2022	Indoor Navigation	Not Specific	Gazebo	Yes	Is proposed a visual information pyramid model to investigate a practical environment representation systematically.
[49]	2022	Wireless Communication	SAC	Not Specific	Yes	Focuses in solve the optical beam alignment problem by maintaining the relative position and orientation of two AUVs.
[50]	2022	Outdoor Navigation	Not Specific	CARLA	Yes	Develop a quadruped robot that follows a route plan generated by public maps services, while remains sidewalksand avoid pedestrians.
[51]	2022	Mapless Navigation	Actor Critic	Gazebo	Yes	Design a DRL framework for a navigation policy that allows the USV to reach the destination collision-free using local sensors.
[52]	2022	Path Planning	PPO	Webots	Yes	A novel end-to-end path planning based DRL for aerial robots deployed in dense environments.
[53]	2022	Autonomous Drone Racing	Not Specific	Unreal Engine	No	A novel DNN-based perception method for racing gate detection which relies on a lightweight NN backbone on top of a pencil filter.
[54]	2023	Not Specific	SAC (Modified)	Not Specific	Yes	A Sim-to-Lab-to-Real to bridge the reality gap with a probabilistically guaranteed safety-aware policy distribuiton.
[55]	2023	Outdoor Local Navigation	PPO	RaiSim	Yes	A novel outdoor navigation algorithm to generate stable and efficient actions to navigate a robot to reach a goal.

lighting conditions, and camera positions compared to real-world environments. Visual domain randomization aims to provide sufficient simulated variability in visual parameters during training to enable the model to generalize to real-world data during testing. Additionally, dynamics randomization introduces randomness in physical parameters to facilitate the learning of robust policies, especially in scenarios where precise control is required.

An example of the power of domain randomization is demonstrated in the work of, where various physical parameters such as object dimensions, masses, friction coefficients, joint damping coefficients, and actuator force gains are randomized in the simulator to train dexterous in-hand manipulation policies for a physical hand. Their successful sim-to-real transfer experiments highlight the effectiveness of domain randomization.

In addition to randomizing the simulated data to match the real-world data distribution, presents an alternative perspective on applying domain randomization. They propose translating the randomized simulated images and real-world images into canonical simulated images, demonstrating the effectiveness of this sim-to-real approach by training a vision-based closed-loop grasping reinforcement learning agent in simulation.

D. DOMAIN ADAPTATION METHODS

Domain adaptation methods aim to enhance the performance of a learned model on a target domain, where data availability is limited, by utilizing data from a different source domain. Typically, the feature spaces of the source and target domains differ, making it necessary to unify these feature spaces for effective knowledge transfer from the source data. This concept is illustrated in Figure IV-D-b. While domain adaptation research has primarily focused on vision-based tasks such as image classification and semantic segmentation [59][60], this paper specifically concentrates on reinforcement learning tasks and their application in robotics.

In the context of reinforcement learning and robotics, domain adaptation techniques serve as priors for subsequent tasks involving building reinforcement learning agents or other control-related tasks. These techniques are particularly useful in scenarios where vision-related tasks play a crucial role in informing the learning process. For example, domain adaptation has been applied to generalize policies learned from synthetic data or accelerate learning on real-world robots. Additionally, domain adaptation has been used to directly transfer policies between agents.

Specifically, we now formalize the domain adaptation scenarios in a RL setting []. Based on the definition of MDP, we denote the source domain as $D_S \equiv (\mathscr{S}_S, \mathscr{A}_S, \mathscr{P}_S, \mathscr{R}_S)$ and target domain as $D_T \equiv (\mathscr{S}_T, \mathscr{A}_T, \mathscr{P}_T, \mathscr{R}_T)$, respectively. In RL scenarios, the state $\mathscr{S}$ of the source and target domain can be quite different ($\mathscr{S}_S \neq \mathscr{S}_T$) due to the perceptual-reality gap [], while both domains share the action spaces and the transitions $\mathscr{P}(\mathscr{A}_S \approx \mathscr{A}_T, \mathscr{P}_S \approx \mathscr{P}_T)$ and their reward functions $\mathscr{R}$ have structural similarity ($\mathscr{R}_S \approx \mathscr{R}_T$).

From the existing literature, three common methods for domain adaptation can be identified: discrepancy-based, adversarial-based, and reconstruction-based methods. These methods are applicable across various tasks. Discrepancy-based methods involve measuring the feature distance between the source and target domains using predefined statistical metrics, with the goal of aligning their feature spaces. Adversarial-based methods, on the other hand, employ a domain classifier that distinguishes features originating from the source domain from those of the target domain. Through training, the feature extractor learns to produce invariant features that are consistent across both domains.

Reconstruction-based methods also aim to identify invariant or shared features between domains. They achieve this by introducing an auxiliary reconstruction task, where the shared feature is used to reconstruct the original input. The objective is to ensure that the shared feature is independent of the specific domains. These three methods offer different approaches to unify features from diverse domains and can be applied in both vision tasks and control tasks based on reinforcement learning (RL).

E. LEARNING WITH DISTURBANCES

Domain randomization and dynamics randomization methods are designed to mitigate the discrepancies between simulation and reality by introducing perturbations in the simulation environments. These perturbations aim to enhance the robustness of agents and make them less susceptible to the mismatches encountered during sim-to-real transfer.

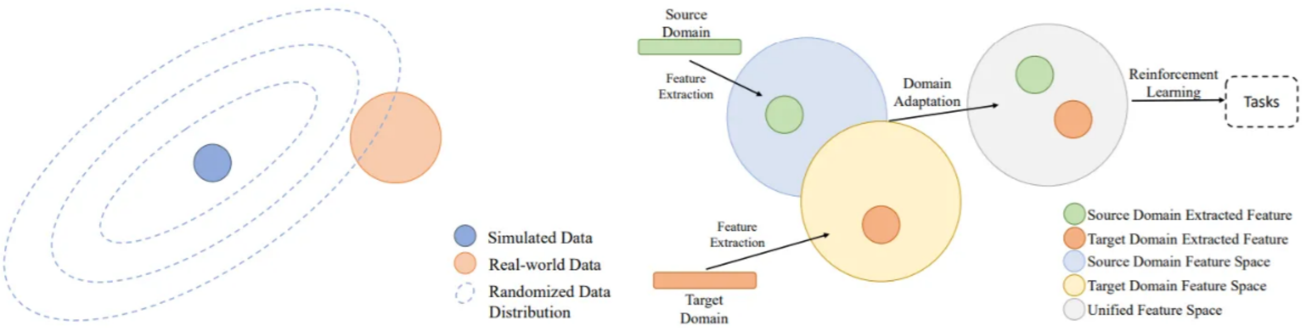
The concept of introducing perturbations to improve robustness has been extended to other works as well. For instance, explores the impact of noisy rewards on agent training. Although not directly related to sim-to-real transfer, noisy rewards can better emulate the challenges faced during real-world agent training.

In our own recent works, we have also considered environmental perturbations that affect different agents differently in parallel learning scenarios. This aspect becomes crucial when multiple real agents are being deployed or trained with a common policy. Accounting for such environmental perturbations can help ensure that the trained agents are capable of handling the inherent variabilities and uncertainties present in real-world settings.

Overall, the introduction of perturbations, whether through domain randomization, dynamics randomization, noisy rewards, or environmental variations, plays a crucial role in enhancing the robustness and adaptability of agents during sim-to-real transfer, ultimately improving their performance in real-world scenarios.

F. SIMULATION ENVIRONMENTS

The choice of simulation plays a crucial role in sim-to-real transfer. Regardless of the techniques employed for efficient knowledge transfer to real robots, a more realistic simulation tends to yield better results. Several simulators have been



(a) Intuition behind the domain randomization paradigm.

(b) Intuition behind the domain adaptation paradigm.

widely used in the literature, including Gazebo, Unity3D, PyBullet, and MuJoCo.

Gazebo offers the advantage of seamless integration with the Robot Operating System (ROS) middleware, making it compatible with various components of the robotics stack used in real robots. On the other hand, PyBullet and MuJoCo exhibit broader integration with deep learning and reinforcement learning libraries, as well as gym environments. In general, Gazebo is well-suited for complex scenarios, while PyBullet and MuJoCo provide faster training capabilities.

In cases where system identification is the objective for one-shot transfer, researchers often resort to building or customizing specific simulations tailored to meet problem-specific requirements and constraints. These custom simulations are designed to capture the intricacies and dynamics of the target system, enabling more accurate modeling and facilitating the transfer of knowledge to real robots.

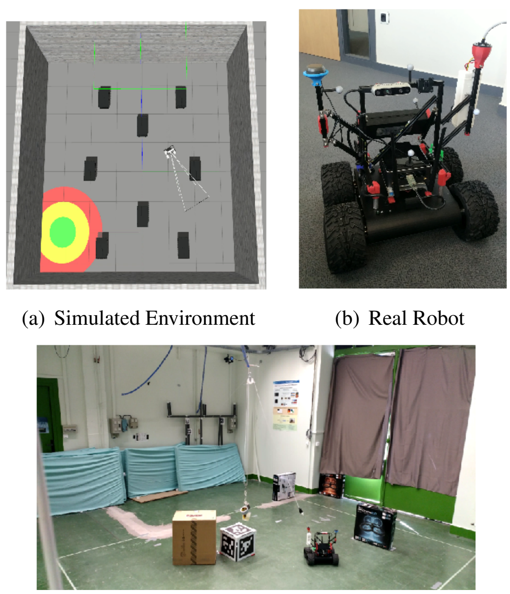
V. SIM-TO-REAL IN ROBOTICS NAVIGATION

Robotics navigation is a critical application scenario where sim-to-real transfer techniques are highly relevant. In simulated environments, robots can be trained to navigate complex environments, map their surroundings, and plan efficient paths. However, transferring these navigation skills to real-world settings poses significant challenges due to differences in perception, sensor noise, and environmental dynamics. Domain adaptation methods play a crucial role in bridging this gap by aligning the sensory inputs, adapting the mapping and localization algorithms, and fine-tuning the navigation policies. By leveraging sim-to-real transfer, robots can effectively navigate in real-world scenarios, including indoor environments, outdoor terrains, and dynamic environments with moving obstacles. This enables applications such as warehouse automation, autonomous exploration, and search and rescue missions, where robots need to navigate safely and efficiently in diverse and unpredictable environments. In this section, we explore the applications and challenges of sim-to-real transfer in robotics navigation, highlighting key techniques and advancements.

Figure 6 depicts a comparison between navigation in a simulated environment and navigation in a real-world setting. Simulations provide a controlled and cost-effective platform for training navigation algorithms, but they often fail to capture the intricacies and uncertainties of the real world. Sim-to-real transfer techniques aim to overcome these limitations by adapting the learned policies from simulations to real-world scenarios.

Over the past five years, there has been a remarkable surge in the number of research articles focusing on sim-to-real transfer in robotics navigation. This growing interest reflects the increasing recognition of the challenges faced by autonomous robots when transitioning from simulation to real-world environments. The availability of powerful computational resources, advancements in machine learning algorithms, and the growing accessibility of robotic platforms have contributed to the accelerated progress in this field.

The evolution of research articles in sim-to-real transfer for robotics navigation can be attributed to several factors.



(a) Simulated Environment (b) Real Robot

(c) Real Experiment

FIGURE 6. Simulated vs. Real-world Navigation.

TABLE 3. Comparison of Most used Sim-to-Real Techniques in Robotics Navigation

Technique	Description	Advantages	Limitations
Domain Randomization	Introduces variations in simulation to increase robustness	Easy implementation and generalization	Limited ability to handle complex real-world scenarios
Discrepancy-based	Measures feature distance between source and target domains	Quantitative alignment of feature spaces	Relies on predefined statistical metrics
Adversarial-based	Builds a domain classifier to produce invariant features	Produces domain-invariant features	Requires additional training of the domain classifier
Reconstruction-based	Constructs an auxiliary reconstruction task to find shared features	Encourages shared and invariant feature representation	Requires additional computational resources for training

Firstly, the exponential growth in computational capabilities has enabled researchers to train and evaluate complex navigation algorithms efficiently. High-performance computing resources, such as GPUs and cloud computing, have empowered researchers to tackle larger and more realistic simulation environments, leading to improved policy generalization and transferability. Secondly, the advancements in machine learning algorithms, particularly in the field of deep learning, have played a pivotal role in enhancing the effectiveness of sim-to-real transfer. Deep reinforcement learning models, coupled with techniques like domain adaptation and domain randomization, have demonstrated remarkable capabilities in transferring policies from simulations to real-world robotic platforms. These advancements have opened up new possibilities for developing robust and adaptive navigation systems.

Table 3 provides a comparison of different sim-to-real techniques used in robotics navigation. It highlights the advantages and limitations of each technique, aiding researchers in selecting the most suitable approach for their specific navigation problem. Furthermore, the availability of diverse and affordable robotic platforms has facilitated the validation and evaluation of sim-to-real techniques in real-world scenarios. Researchers now have access to a wide range of robots, including ground-based and aerial vehicles, which can be deployed to test the efficacy of sim-to-real transfer approaches. This has enabled researchers to investigate the practical challenges and limitations that arise when deploying sim-to-real policies in complex, dynamic, and unpredictable environments.

The next sections shows other application scenarios of the sim-to-real techniques addressed in this work, as well as a bibliometric analysis related to this research area. This analysis will highlight the progress achieved in recent years, showcasing the growing interest and publication volume. Furthermore, open problems will be discussed, offering opportunities for future research in this rapidly evolving field.

VI. OTHER APPLICATION SCENARIOS

Sim-to-real transfer and the application of domain adaptation techniques have significant implications in various domains, including robotic manipulation and control, robotics navigation, autonomous driving, robotics research, and computer vision. By bridging the gap between simulation and reality, these techniques enable the transfer of knowledge, policies, and models from simulated environments to real-world applications, leading to safer, more efficient, and more reliable AI systems and robotics solutions. In this section, we explore some application scenarios where sim-to-real transfer plays a crucial role in robotics.

A. ROBOTIC MANIPULATION AND CONTROL

Sim-to-real transfer is particularly valuable in the field of robotic manipulation and control. Simulations provide a controlled and reproducible environment for training robotic agents, allowing them to learn complex manipulation tasks before deploying them in real-world scenarios. By applying domain adaptation methods, the gap between the simulated and real-world dynamics can be addressed. This includes matching the sensory inputs, such as vision or depth perception, and adapting the control policies to account for the differences in physical properties. Domain adaptation techniques enable the robotic agents to generalize their learned behaviors and effectively manipulate objects, navigate obstacles, and perform various control tasks in real-world environments.

B. AUTONOMOUS DRIVING

The application of sim-to-real transfer in autonomous driving is crucial for the development and deployment of safe and reliable self-driving systems. Simulations provide a controlled and scalable platform to train and test autonomous vehicles. By utilizing domain adaptation techniques, models trained in simulated environments can be adapted to real-world driving scenarios. This involves aligning the perception models to handle variations in lighting conditions, weather, and other environmental factors. Additionally, domain adaptation ensures that the control policies learned in simulation can be effectively transferred to real-world driving tasks, enabling autonomous vehicles to navigate complex traffic scenarios and respond appropriately to dynamic road conditions.

C. ROBOTICS RESEARCH AND DEVELOPMENT

Simulations are widely used in robotics research and development as they offer a cost-effective and efficient way to iterate and evaluate algorithms and systems. Domain adaptation methods play a vital role in transferring knowledge and policies learned in simulated environments to real robotic platforms. This allows researchers to accelerate the development and deployment of advanced robotic capabilities in real-world applications. For example, object recognition models trained in simulation can be adapted to recognize and manipulate objects in real-world scenarios. Similarly, path planning algorithms can be fine-tuned using domain adaptation techniques to account for differences in terrain, obstacles, and sensor noise, enabling robots to navigate in challenging environments.

D. TRANSFER LEARNING IN COMPUTER VISION

Sim-to-real transfer is not limited to robotics but also extends to transfer learning in computer vision tasks. Simulated environments provide a rich source of labeled training data, allowing deep learning models to be trained efficiently. However, these models often face challenges when applied to real-world scenarios due to domain differences. Domain adaptation methods facilitate the alignment of visual features and representations between the simulated and real domains. By adapting the learned models using real-world data, the performance and generalization capabilities of computer vision models can be significantly improved. This enables applications such as object recognition, semantic segmentation, and image classification to be more robust and accurate in real-world settings.

VII. BIBLIOMETRIC ANALYSIS

For the bibliographic analysis, the number of publications, and the publications by techniques over the last five years will be taken into account. The factors, along with their relationships, provide quite a bit of information on how UAV sim-to-real applications are doing for robotics navigation problems.

Figure VII illustrate the evolution of the number of publications in the last 5 years, shows the interest in the subject and the evolution of the field based on the State of the Art. In Figure VII it is possible to notice that the Domain Adaptation and Domain Randomization outperforms the others techniques, indicating their prominence and effectiveness in addressing sim-to-real challenges in robotics navigation. This suggests that researchers are actively exploring and developing methods that focus on adapting simulated models to real-world scenarios and reducing the reality gap.

Furthermore, the analysis reveals an increasing trend in the number of publications related to sim-to-real transfer in robotics navigation over the past five years. This indicates the growing interest and recognition of the importance of sim-to-real techniques in advancing the field. The increasing number of publications also signifies the continuous efforts of the research community to overcome the challenges associated with sim-to-real transfer and improve the performance and applicability of robotic navigation systems in real-world environments.

Overall, the bibliometric analysis provides valuable insights into the current state of research in sim-to-real transfer for robotics navigation. It highlights the dominant techniques and trends, emphasizing the need for effective domain adaptation and domain randomization strategies. The analysis

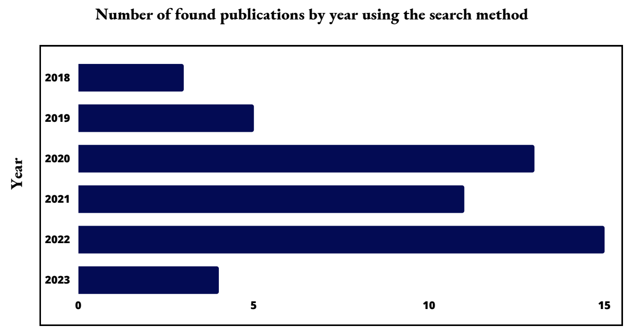


FIGURE 7. Relevant and novel publications found per year. The bar of the year 2023 is the number of papers of the first quarter

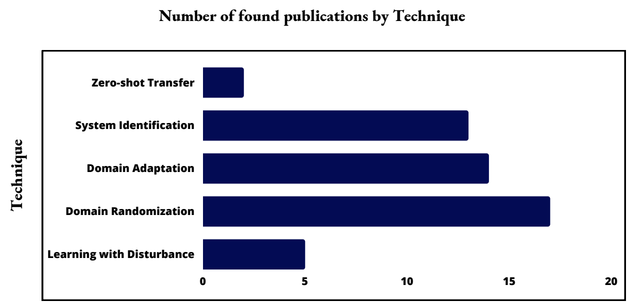


FIGURE 8. Relevant and novel publications found per technique.

also demonstrates the growing interest and dedication of researchers in addressing sim-to-real challenges, paving the way for future advancements and innovations in the field of robotics navigation.

VIII. OPEN ISSUES

In this work we summarize and compare multiple strategies used in the literature to improve different aspects, using DRL methods, involved in the Sim-to-Real Problem. Based on the analysis performed in the previous sections, the following paragraphs discuss and highlight possible future research issues.

The discussion introduces some problems and challenges, including the sim-to-real transfer limitations and the environment dynamic level. The open research issues are:

i. **Generalization to complex real-world environments**: Simulators often struggle to capture the full complexity and diversity of real-world environments, which poses a challenge for sim-to-real transfer in robot navigation. Developing simulators that can accurately represent the complexities of real-world scenarios remains an open issue.

ii. **Handling perception and sensor discrepancies**: Simulated sensor data may not perfectly match real-world sensor data due to differences in noise, resolution, and other factors. Bridging the gap between simulated and real-world perception is a key challenge in sim-to-real transfer for robot navigation.

iii. **Dynamic and changing environments**: Real-world environments are dynamic, with moving objects, changing lighting conditions, and varying obstacles. Simulators need to account for these dynamic elements to enable effective sim-to-real transfer in robot navigation tasks.

iv. **Efficient training and adaptation**: Training RL-based navigation policies in simulators can be time-consuming and computationally intensive. Finding methods to improve training efficiency and adapt policies trained in simulation to real-world settings is an ongoing challenge.

v. **Calibration and system identification**: Accurate system identification and calibration of simulators to match real-world dynamics are crucial for effective sim-to-real transfer in robot navigation. Developing robust and efficient calibration techniques is an open research area.

vi. **Uncertainty and risk-awareness**: Real-world navigation tasks involve inherent uncertainties and risks. Incorporating uncertainty estimation and risk-awareness into sim-to-real transfer approaches is an important direction for future research in robot navigation.

vii. **Long-term and multi-modal navigation**: Sim-to-real transfer methods often focus on short-term navigation tasks or rely on simplified representations of the environment. Developing techniques that enable long-term and multi-modal navigation in complex real-world environments is an open issue.

vii. **Transfer learning across different domains**: Extending sim-to-real transfer beyond a single domain or environment remains challenging. Finding methods to transfer learned policies across different domains or adapt them to new environments is an important research direction in sim-to-real transfer for robot navigation.

vii. **Real-time adaptation and robustness**: Real-world navigation scenarios require robots to adapt and respond to changing conditions in real-time. Developing real-time adaptation techniques and ensuring the robustness of sim-to-real transferred policies in dynamic environments are key open issues in robot navigation.

These open issues reflect ongoing research efforts and highlight the complexity and breadth of challenges in sim-to-real transfer for robot navigation. Interested readers are referred to [?] for more information on practical challenges that may arise in the real-world application of the RL, such as high-dimensional continuous and actions spaces, real-time inference, system delays, cases with limited samples, just to name a few. Furthermore, practical experiments to validate the simulations and the proposed solutions are practically mandatory to conclude about some approach, specially considering the robots limitations and the high computational power required by DRL techniques.

IX. CONCLUSION

This survey has examined various aspects of sim-to-real transfer in robotics and highlighted its growing importance in bridging the gap between simulated environments and real-world applications. Key topics explored include meta reinforcement learning, robust RL, and imitation learning, which have proven crucial in achieving successful sim-to-real transfer.

Domain randomization and dynamics randomization methods have emerged as effective strategies for enhancing agent robustness and addressing simulation-reality discrepancies. The choice of a realistic simulator, such as Gazebo, PyBullet, or MuJoCo, is critical for effective sim-to-real transfer. System identification, perception handling, and calibration have been identified as vital elements for building accurate simulators that closely replicate real-world dynamics. However, open issues remain, including generalization to complex environments, handling dynamic and changing environments, and ensuring transferability across robot platforms. Further research is needed to address challenges in training efficiency, adaptation, uncertainty and risk-awareness, and long-term and multi-modal navigation.

Overall, sim-to-real transfer in robotics presents a multi-dimensional problem with significant implications for real-world applications. This survey provides valuable insights into the current state of the field, showcasing achievements and identifying areas for future exploration. By tackling these challenges and advancing sim-to-real transfer techniques, researchers can pave the way for more reliable, adaptable, and robust robotic systems capable of seamless transitions from simulation to real-world environments.

REFERENCES

[1] B. Singh, R. Kumar, and V. P. Singh, “Reinforcement learning in robotic applications: a comprehensive survey,” *Artificial Intelligence Review*, pp. 1–46, 2022.

[2] J. Orr and A. Dutta, “Multi-agent deep reinforcement learning for multi-robot applications: A survey,” *Sensors*, vol. 23, no. 7, p. 3625, 2023.

[3] M. Jaritz, R. De Charette, M. Toromanoff, E. Perot, and F. Nashashibi, “End-to-end race driving with deep reinforcement learning,” in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 2070–2075.

[4] W. Zhao, J. P. Queralta, and T. Westerlund, “Sim-to-real transfer in deep reinforcement learning for robotics: a survey,” in *2020 IEEE symposium series on computational intelligence (SSCI)*. IEEE, 2020, pp. 737–744.

[5] E. Salvato, G. Fenu, E. Medvet, and F. A. Pellegrino, “Crossing the reality gap: A survey on sim-to-real transferability of robot controllers in reinforcement learning,” *IEEE Access*, vol. 9, pp. 153 171–153 187, 2021.

[6] T. Zhang, K. Zhang, J. Lin, W.-Y. G. Louie, and H. Huang, “Sim2real learning of obstacle avoidance for robotic manipulators in uncertain environments,” *IEEE Robotics and Automation Letters*, vol. 7, no. 1, pp. 65–72, 2021.

[7] Y. Liu, Z. Cao, H. Xiong, J. Du, H. Cao, and L. Zhang, “Dynamic obstacle avoidance for cable-driven parallel robots with mobile bases via sim-to-real reinforcement learning,” *IEEE Robotics and Automation Letters*, vol. 8, no. 3, pp. 1683–1690, 2023.

[8] X. B. Peng, M. Andrychowicz, W. Zaremba, and P. Abbeel, “Sim-to-real transfer of robotic control with dynamics randomization,” in *2018 IEEE*

international conference on robotics and automation (ICRA). IEEE, 2018, pp. 3803–3810.

[9] M. Hwasser, D. Kragic, and R. Antonova, "Variational auto-regularized alignment for sim-to-real control," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 2732–2738.

[10] Y. Chebotar, A. Handa, V. Makoviychuk, M. Macklin, J. Issac, N. Ratliff, and D. Fox, "Closing the sim-to-real loop: Adapting simulation randomization with real world experience," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 8973–8979.

[11] F. Zhang, J. Leitner, Z. Ge, M. Milford, and P. Corke, "Adversarial discriminative sim-to-real transfer of visuo-motor policies," *The International Journal of Robotics Research*, vol. 38, no. 10-11, pp. 1229–1245, 2019.

[12] X. Jing, K. Qian, T. Jianu, and S. Luo, "Unsupervised adversarial domain adaptation for sim-to-real transfer of tactile images," *IEEE Transactions on Instrumentation and Measurement*, 2023.

[13] F. Muratore, C. Eilers, M. Gienger, and J. Peters, "Data-efficient domain randomization with bayesian optimization," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 911–918, 2021.

[14] R. Cheng, G. Orosz, R. M. Murray, and J. W. Burdick, "End-to-end safe reinforcement learning through barrier functions for safety-critical continuous control tasks," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 33, no. 01, 2019, pp. 3387–3395.

[15] H. Ali, D. Gong, M. Wang, and X. Dai, "Path planning of mobile robot with improved ant colony algorithm and mdp to produce smooth trajectory in grid-based environment," *Frontiers in neurorobotics*, vol. 14, p. 44, 2020.

[16] B. Cunha, A. M. Madureira, B. Fonseca, and D. Coelho, "Deep reinforcement learning as a job shop scheduling solver: A literature review," in *International Conference on Hybrid Intelligent Systems*. Springer, 2018, pp. 350–359.

[17] H. Nguyen and H. La, "Review of deep reinforcement learning for robot manipulation," in *2019 Third IEEE International Conference on Robotic Computing (IRC)*. IEEE, 2019, pp. 590–595.

[18] R. S. Sutton, A. G. Barto *et al.*, *Introduction to reinforcement learning*. MIT press Cambridge, 1998, vol. 135.

[19] C. J. C. H. Watkins, "Learning from delayed rewards," 1989.

[20] R. A. Bianchi, C. H. Ribeiro, and A. H. Costa, "Accelerating autonomous learning by using heuristic selection of actions," *Journal of Heuristics*, vol. 14, no. 2, pp. 135–168, 2008.

[21] M. E. Taylor and P. Stone, "Transfer learning for reinforcement learning domains: A survey." *Journal of Machine Learning Research*, vol. 10, no. 7, 2009.

[22] F. Zhuang, Z. Qi, K. Duan, D. Xi, Y. Zhu, H. Zhu, H. Xiong, and Q. He, "A comprehensive survey on transfer learning," *Proceedings of the IEEE*, vol. 109, no. 1, pp. 43–76, 2020.

[23] D. Weinshall, G. Cohen, and D. Amir, "Curriculum learning by transfer learning: Theory and experiments with deep networks," in *International Conference on Machine Learning*. PMLR, 2018, pp. 5238–5246.

[24] K. A. Krueger and P. Dayan, "Flexible shaping: How learning in small steps helps," *Cognition*, vol. 110, no. 3, pp. 380–394, 2009.

[25] S. Narvekar, B. Peng, M. Leonetti, J. Sinapov, M. E. Taylor, and P. Stone, "Curriculum learning for reinforcement learning domains: A framework and survey," *arXiv preprint arXiv:2003.04960*, 2020.

[26] C. Florensa, D. Held, M. Wulfmeier, M. Zhang, and P. Abbeel, "Reverse curriculum generation for reinforcement learning," in *Conference on robot learning*. PMLR, 2017, pp. 482–495.

[27] S. Luo, H. Kasaei, and L. Schomaker, "Accelerating reinforcement learning for reaching using continuous curriculum learning," in *2020 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2020, pp. 1–8.

[28] Y. Wang, H. He, and X. Tan, "Truly proximal policy optimization," in *Uncertainty in Artificial Intelligence*. PMLR, 2020, pp. 113–122.

[29] H. Li and H. He, "Multiagent trust region policy optimization," *IEEE Transactions on Neural Networks and Learning Systems*, 2023.

[30] A. Abdolmaleki, J. T. Springenberg, Y. Tassa, R. Munos, N. Heess, and M. Riedmiller, "Maximum a posteriori policy optimisation," *arXiv preprint arXiv:1806.06920*, 2018.

[31] M. Sewak and M. Sewak, "Actor-critic models and the a3c: The asynchronous advantage actor-critic model," *Deep reinforcement learning: frontiers of artificial intelligence*, pp. 141–152, 2019.

[32] J. C. de Jesus, V. A. Kich, A. H. Kolling, R. B. Grando, M. A. d. S. L. Cuadros, and D. F. T. Gamarra, "Soft actor-critic for navigation of mobile robots," *Journal of Intelligent & Robotic Systems*, vol. 102, no. 2, p. 31, 2021.

[33] J. C. Jesus, J. A. Bottega, M. A. Cuadros, and D. F. Gamarra, "Deep deterministic policy gradient for navigation of mobile robots in simulated environments," in *2019 19th International Conference on Advanced Robotics (ICAR)*. IEEE, 2019, pp. 362–367.

[34] N. A. Mosali, S. S. Shamsudin, O. Alfandi, R. Omar, and N. Al-Fadhali, "Twin delayed deep deterministic policy gradient-based target tracking for unmanned aerial vehicle with achievement rewarding and multistage training," *IEEE Access*, vol. 10, pp. 23 545–23 559, 2022.

[35] J. Tan, T. Zhang, E. Coumans, A. Iscen, Y. Bai, D. Hafner, S. Bohez, and V. Vanhoucke, "Sim-to-real: Learning agile locomotion for quadruped robots," *arXiv preprint arXiv:1804.10332*, 2018.

[36] N. Hirose, F. Xia, R. Martín-Martín, A. Sadeghian, and S. Savarese, "Deep visual mpc-policy learning for navigation," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 3184–3191, 2019.

[37] H. Bharadhwaj, Z. Wang, Y. Bengio, and L. Paull, "A data-efficient framework for training and sim-to-real transfer of navigation policies," in *2019 International Conference on Robotics and Automation (ICRA)*, 2019, pp. 782–788.

[38] K. Rao, C. Harris, A. Irpan, S. Levine, J. Ibarz, and M. Khansari, "Rl-cyclegan: Reinforcement learning aware simulation-to-real," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[39] T. Chaffre, J. Moras, A. Chan-Hon-Tong, and J. Marzat, "Sim-to-Real Transfer with Incremental Environment Complexity for Reinforcement Learning of Depth-Based Robot Navigation," in *17th International Conference on Informatics, Automation and Robotics, ICINCO 2020*, ser. Proceedings of the 17th International Conference on Informatics, Automation and Robotics, ICINCO 2020, Virtual, Online, France, Jul. 2020, pp. 314–323. [Online]. Available: https://ensta-bretagne.hal.science/hal-02958155

[40] P. Anderson, A. Shrivastava, J. Truong, A. Majumdar, D. Parikh, D. Batra, and S. Lee, "Sim-to-real transfer for vision-and-language navigation," in *Proceedings of the 2020 Conference on Robot Learning*, ser. Proceedings of Machine Learning Research, J. Kober, F. Ramos, and C. Tomlin, Eds., vol. 155. PMLR, 16–18 Nov 2021, pp. 671–681. [Online]. Available: https://proceedings.mlr.press/v155/anderson21a.html

[41] N. Liu, Y. Cai, T. Lu, R. Wang, and S. Wang, "Real–sim–real transfer for real-world robot control policy learning with deep reinforcement learning," *Applied Sciences*, vol. 10, no. 5, 2020. [Online]. Available: https://www.mdpi.com/2076-3417/10/5/1555

[42] R. Polvara, M. Patacchiola, M. Hanheide, and G. Neumann, "Sim-to-real quadrotor landing via sequential deep q-networks and domain randomization," *Robotics*, vol. 9, no. 1, 2020. [Online]. Available: https://www.mdpi.com/2218-6581/9/1/8

[43] K. Lobos-Tsunekawa and T. Harada, "Point cloud based reinforcement learning for sim-to-real and partial observability in visual navigation," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, pp. 5871–5878.

[44] H. Niu, Z. Ji, F. Arvin, B. Lennox, H. Yin, and J. Carrasco, "Accelerated sim-to-real deep reinforcement learning: Learning collision avoidance from human player," in *2021 IEEE/SICE International Symposium on System Integration (SII)*, 2021, pp. 144–149.

[45] D. Hoeller, L. Wellhausen, F. Farshidian, and M. Hutter, "Learning a state representation and navigation in cluttered and dynamic environments," *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 5081–5088, 2021.

[46] H. Hu, K. Zhang, A. H. Tan, M. Ruan, C. Agia, and G. Nejat, "A sim-to-real pipeline for deep reinforcement learning for autonomous robot navigation in cluttered rough terrain," *IEEE Robotics and Automation Letters*, vol. 6, no. 4, pp. 6569–6576, 2021.

[47] S. Wen, Z. Wen, D. Zhang, H. Zhang, and T. Wang, "A multi-robot path-planning algorithm for autonomous navigation using meta-reinforcement learning based on transfer learning," *Applied Soft Computing*, vol. 110, p. 107605, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1568494621005263

[48] G. Chen, H. Yu, W. Dong, X. Sheng, X. Zhu, and H. Ding, "What should be the input: Investigating the environment representations in sim-to-real transfer for navigation tasks," *Robotics and Autonomous Systems*, vol. 153, p. 104081, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0921889022000409

[49] Y. Weng, T. Matsuda, Y. Sekimori, J. Pajarinen, J. Peters, and T. Maki, "Sim-to-real transfer for underwater wireless optical communication alignment policy between auvs," in *OCEANS 2022 - Chennai*, 2022, pp. 1–7.

[50] M. Sorokin, J. Tan, C. K. Liu, and S. Ha, “Learning to navigate sidewalks in outdoor environments,” *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 3906–3913, 2022.

[51] N. Wang, Y. Wang, Y. Zhao, Y. Wang, and Z. Li, “Sim-to-real: Mapless navigation for usvs using deep reinforcement learning,” *Journal of Marine Science and Engineering*, vol. 10, no. 7, 2022. [Online]. Available: https://www.mdpi.com/2077-1312/10/7/895

[52] H. I. Ugurlu, X. H. Pham, and E. Kayacan, “Sim-to-real deep reinforcement learning for safe end-to-end planning of aerial robots,” *Robotics*, vol. 11, no. 5, 2022. [Online]. Available: https://www.mdpi.com/2218-6581/11/5/109

[53] H. X. Pham, A. Sarabakha, M. Odnoshyvkin, and E. Kayacan, “Pencilnet: Zero-shot sim-to-real transfer learning for robust gate perception in autonomous drone racing,” *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 11 847–11 854, 2022.

[54] K.-C. Hsu, A. Z. Ren, D. P. Nguyen, A. Majumdar, and J. F. Fisac, “Sim-to-lab-to-real: Safe reinforcement learning with shielding and generalization guarantees,” *Artificial Intelligence*, vol. 314, p. 103811, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0004370222001515

[55] J. Liang, K. Weerakoon, T. Guan, N. Karapetyan, and D. Manocha, “Adaptiveon: Adaptive outdoor local navigation method for stable and reliable actions,” *IEEE Robotics and Automation Letters*, vol. 8, no. 2, pp. 648–655, 2023.

IURE ROSA L. OLIVEIRA, received the B.S. degree in electrical engineering from the Federal University of Viçosa, Brazil, in 2022. He is currently pursuing the M.Sc. degree in computer science with the Federal University of Viçosa. His research interest includes control of multi-robot systems applied to aerial robots navigation and communication systems.