Digital Twin-Enabled Domain Adaptation for Zero-Touch UAV Networks: Survey and Challenges

Maxwell McManus[a], Yuqing Cui[a], Josh (Zhaoxi) Zhang[a], Jiangqi Hu[a], Sabarish Krishna Moorthy[a], Nicholas Mastronarde[a], Elizabeth Serena Bentley[b], Michael Medley[b], and Zhangyu Guan[a*]
[a]Department of Electrical Engineering, University at Buffalo, Buffalo, NY 14260, USA
[b]Air Force Research Laboratory (AFRL), Rome, NY 13440, USA
Email: {memcmanu, yuqingcu, zhaoxizh, jiangqih, sk382, nmastron, guan}@buffalo.edu, {elizabeth.bentley.3, michael.medley}@us.af.mil

Abstract

In existing wireless networks, the control programs have been designed manually and for certain predefined scenarios. This process is complicated and error-prone, and the resulting control programs are not resilient to disruptive changes. Data-driven control based on Artificial Intelligence and Machine Learning (AI/ML) has been envisioned as a key technique to automate the modeling, optimization and control of complex wireless systems. However, existing AI/ML techniques rely on sufficient well-labeled data and may suffer from slow convergence and poor generalizability. In this article, focusing on digital twin-assisted wireless unmanned aerial vehicle (UAV) systems, we provide a survey of emerging techniques that can enable fast-converging data-driven control of wireless systems with enhanced generalization capability to new environments. These include simultaneous localization and sens-

*Corresponding author

[1]ACKNOWLEDGMENT OF SUPPORT AND DISCLAIMER: (a) Contractor acknowledges Government's support in the publication of this paper. This material is based upon work funded in part by AFRL under AFRL Contract No. #FA8750-20-C-1021 and #FA8750-21-F-1012 and in part by the NSF under Grant SWIFT-2229563. (b) Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of AFRL.

[2]Distribution A. Approved for public release: Distribution unlimited AFRL-2022-5944 on 16 Dec 2022.

ing (SLAM)-based sensing and network softwarization for digital twin construction, robust reinforcement learning and system identification for domain adaptation, and testing facility sharing and federation. The corresponding research opportunities are also discussed.

Keywords: UAV, Digital Twin, Domain Adaptation, Network softwarization, AI/ML

1. Introduction

Unmanned aerial vehicles (UAVs) have been envisioned as a key enabling technology for a wide set of new applications because of their unique characteristics such as fast deployment, high mobility, on-board processing capabilities, and reduced size. This has allowed significant progress in foundational research towards UAV-assisted communication networks, e.g., swarm UAV networks. Specifically, the high mobility of UAVs can be leveraged to enable dynamic network area coverage and maximize service capacity at mobile ground nodes [1]. Furthermore, UAV swarms can serve as MIMO-enabled self-organizing flying hotspots for terrestrial ad-hoc networks to improve spectral efficiency [2]. In IoT networks, UAVs can be leveraged as distributed relay nodes to expand coverage area and improve quality of service (QoS) [3]. To enable 5G and Beyond network capabilities, UAVs can provide additional computational resources for offloading and support in mobile edge computing (MEC) networks [4]. UAV swarms are also expected to enable 5G massive MIMO (MMIMO), serving as dynamic relays to enable high-throughput communications between MMIMO base stations and ground users and minimize inter-cell interference [5, 6]. Additionally, future network architectures, i.e., 6G, are projected to support hybrid aerial-ground communications, in which terrestrial networks, aerial UAV networks, and satellite communications are linked hierarchically to further enhance QoS and network flexibility [7].

However, while UAVs can certainly enable a new range of applications, the challenges are multi-fold. First, the management of UAV-assisted networks needs to consider the high mobility of all connected nodes, and this requires new resource orchestration and algorithm designs to anticipate the dynamics and requirements of each flying node and hybrid link in addition to those dynamics inherent to the networking environment [8]. The situation will get even worse when jointly considering the newly emerging sophisticated com-

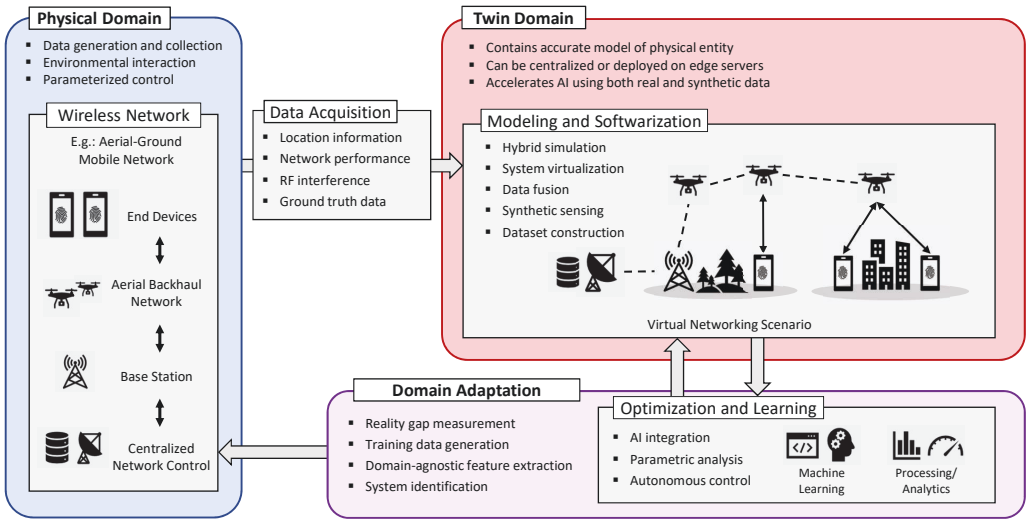


Figure 1: Top-level overview of a DT-enabled system.

munication techniques, such as heterogeneous multi-band communications [9], device-to-device communication links [10], integrated access and backhaul (IAB) 5G networks [11], non-orthogonal multiple access (NOMA) [12, 13] and spectrum coexistence [14, 15]. Moreover, in the current practice of wireless engineering, the networking environments are usually assumed to be known at design time, and the resulting control programs may fail when encountering unforeseen conditions. Traditional manual network management has hindered the adoption of new techniques and the evolution of wireless networks, motivating a new paradigm that can enable *zero-touch* management of UAV-enabled networks, including planning, design and deployment, service delivery, resource management, and end-to-end optimization [16].

Data-driven Approaches. Data-driven modeling and decision-making based on Artificial Intelligence and Machine Learning (AI/ML) are envisioned to be key enablers of zero-touch wireless network management. In recent years, data-driven approaches based on AI/ML have shown great potential for automating the modeling and control of complicated wireless systems. Examples of recent efforts include deep learning-based edge computing for Internet of Things (IoT) [17], multi-label classification for user association in mm-wave networks [18], trajectory and passive beamforming design in UAV-RIS wireless networks based on a decaying deep Q-network [19], and network slicing for industrial IoT based on deep federated Q-learning [20],

among others. Readers are referred to [21, 22, 23, 24] and the references therein for a good survey of the main results in this field.

However, the primary challenges with data-driven approaches are their slow convergence rates and the limited generalization capabilities of the learned policies when faced with new environments. Specifically, the performance of ML (especially deep learning) algorithms highly relies on the availability of a sufficient amount of well-labeled contextual data for model training, leading to slow convergence rates in online applications [21, 25]. Additionally, collecting training data can be too time costly and in some cases pose safety risks for hardware or network operators. Alternatively, the models can be trained in an offline manner using data previously collected or generated by simulators [26]. However, the trained models may suffer from poor robustness, i.e., it is hard for the models to generalize to new environments with different transition kernels.

Digital Twin-enabled Data-driven Control. Digital twins (DT) are envisioned as key enablers of fast-convergent and robust learning for next-generation intelligent cyber-physical systems, such as smart factories and manufacturing [27, 28, 29], smart cities [30, 31], construction, bio-engineering and automotive [32, 33], as well as wireless communication networks [34, 35].

With high-fidelity models in the virtual DT environment, the corresponding physical entity can be reconfigured, simulated and tested at a fraction of the cost and in a fraction of the time of deployment-based configuration testing. By simulating the behaviors of the physical entity in real-time, possible trajectories of a physical entity's life-cycle can be generated using physics-based simulation in order to predict events and conduct root-cause diagnosis. Further, the resulting data can be used to train AI/ML models to determine the optimal solutions for complex control problems, while the trained models can be fine-tuned through real-time feedback from the physical entity. This has been demonstrated in [28], in which a DT has been implemented to provide fault monitoring, scenario evaluation, and adaptive data-driven control for manufacturing systems to improve both safety and productivity. Furthermore, the authors of [30] show that using a custom protocol stack for real-time synchronization between physical and virtual entities, city-scale public services can be managed and optimized using data-driven algorithms deployed in a DT. While the great potential of DTs has been demonstrated in various areas such as manufacturing, smart city, and military applications [27, 28, 29], its adoption in wireless communication networks is still in its early phase.

In this article, we aim to provide a survey of the main results of DT-enabled machine learning in UAV-assisted wireless networks, and discuss the research challenges and possible solutions. In existing literature, there are already a number of surveys and tutorials focusing on DT-enabled wireless systems [33, 36, 37, 38, 39]. For example, in [33] Minerva et al. discuss the foundational properties, essential characteristics and business values of DTs focusing on IoT application domains such as digital patient, digital city and cultural heritage. The authors of [36] discuss DT-enabled 6G from an architectural perspective, including the key design requirements in decoupling, scalability, security and reliability as well as deployment. The authors emphasize the use of DT as an enabling technology for ML-enabled wireless networking via high-performance mobile edge computing and software-defined networking. The enabling technologies for DTs are discussed in [38] for cognizing and controlling the physical world, DT modeling, DT data management, DT services as well as connections in DTs. In [39], Nguyen et al. identify the potential benefits of DT for rolling out 5G networks, including interactive 5G emulation, 5G radio and channel emulation, and continuous validation and optimization. The design and implementation of a DT for optical communication systems are presented in [40], and results are presented to motivate the use of DT systems for hardware configuration, transmission simulation, and fault prediction tasks. The application of AI/ML techniques in wireless network modeling and control has also attracted significant research attention. Readers are referred to [41, 42, 43, 44] and references therein for a good survey and tutorial for the main results in this field. *Different from the above surveys and tutorials, in this article we discuss the challenges and enabling techniques for fast-convergent and robust learning in DT-assisted wireless UAV systems.*

2. Digital Twins for Wireless Systems: A Primer

Digital twins were first introduced in the NASA Apollo program as a *"multi-physics, multi-scale probabilistic simulation"* of an object, system, or process in the physical world, which uses physical parameters, historical data, and sensor updates to provide an accurate virtual "mirror" of the target system [45]. As depicted in Fig. 1, a DT system generally consists of three major components: a physical entity with observable behaviors, a logical (or virtual) object that represents the physical entity in a simulated environment, and a bidirectional feedback system between the two entities [27, 29, 46, 47].

Considering modern applications, DT systems can monitor and virtualize dynamically the behaviors of the physical systems at run-time and further aid in a zero-touch manner the decision-making in unforeseen situations based on data-driven modeling and optimization [48].

In order to provide accurate modeling and control decisions in spite of mathematical generalization, a DT system requires a bidirectional feedback loop capable of translating observed physical behaviors into a virtual model and vice versa. This behavioral translation process is termed *domain adaptation*. To broaden the scope of this investigation, we consider a general theoretical definition of a DT with three critical elements: the physical domain, the twin domain, and domain adaptation. We discuss each element in the context of wireless networks with flying base stations as an example to motivate the application of DTs for next-generation wireless network optimization. A survey of more general DT use cases envisioned to support 6G network capabilities such as high-density deployment configuration and reflective intelligent surface-enabled terahertz communications can be found in [49].

Physical Domain. The physical domain is also called the *target domain*. This domain encompasses all scenario- and application-specific aspects of the system, such as basic network functionalities, mobility controls, physical entities, and other features of the deployment environment. This data can be collected by the DT through a direct interface with each system component, as in [30] and [50], or aggregated by a central controller prior to collection [51]. In general, data acquisition is handled in the physical domain and uploaded to the twin domain in real-time or stored as a dataset for later use, which we will discuss further in Section 3. Considering the example of UAV-assisted networking, the physical domain would include UAV hardware, software, and communication systems used to realize the aerial base station capabilities, all comparable elements of network end-devices, as well as environmental and geographical features, such as wind speed, RF interference, and blockages, that constrain the UAVs' flight patterns and impact network coverage and performance.

Twin Domain. The twin domain, aka *source domain*, encompasses all exogenous elements of the DT system that are designed to accelerate optimization of physical domain applications, facilitate machine learning applications without impact on real-time system operation, or otherwise improve system performance over what is achievable in a deployment that is solely in the physical domain. In general, this will include virtualization of

the target environment, synthetic data generation for policy convergence, and feedback with a domain adaptation process for effective policy transfer across the *sim-to-real gap*. The *sim-to-real gap* refers to the discrepancy in observable performance and behaviors between physical domain entities and their virtual counterparts. This discrepancy is typically caused by generalizations of unpredictable real-world phenomena present in the simulation. Experience collected by agents, especially in dynamic or time-varying environments, may only be valid temporarily, requiring continuous computation and re-optimization. Virtualization is a key technique for improving the flexibility and efficiency of zero-touch control for wireless networking systems [16]. This requires dedicated computational resources and infrastructure to support synthetic data generation and processing as well as bidirectional communication between twin and physical domain systems. We will discuss several methods of target system virtualization and softwarization for wireless networks Section 4.

A detailed example of source domain design for a coordinated UAV swarm network is outlined in [52]. This example includes a centralized intelligence center collecting periodic updates of environment state data, and returning control directives to the deployed hardware for optimal MAC-layer configuration. The intelligence center contains a simulation of the deployed hardware capable of generating synthetic data analogous to physical domain experience, which is in turn used to train a deep neural network to optimize protocol parameters based on a physical domain scenario. With reliable communication between the twin (i.e., source) and physical (i.e., the target) domains, offloaded computation can facilitate accelerated convergence and practical applications, removing prohibitive resource constraints on physical domain systems.

Domain Adaptation. This is the process by which experience collected or generated in one domain is translated for use in the complementary domain. While the addition of resources from the source domain can be incredibly useful, communication between twin and physical domains may not always be reliable. In the case of unreliable communication between domains, the sim-to-real gap may be increased due to the lack of synchronization between physical systems and their virtual counterparts. In such scenarios, learning conducted in the twin domain must be robust to the difference between dynamics in the physical domain and generalizations made in the source domain to enable effective *sim-to-real* policy transfer. The core focus of domain adaptation is to modify learning algorithms and source do-

main parameters to overcome these challenges, and is envisioned as the key to solving open research challenges associated with robustness and performance losses in transfer learning applications inherent to DT-enabled systems [53]. In existing literature, domain adaptation for RL applications can be achieved by modifying observations of the source domain [53], simulation parameters [54], or the reward function [55] of a well-defined Markov decision process (MDP). In each of these approaches, a twin domain is constructed for rapid and efficient training, with the goal of minimizing interaction with the physical domain hence maximizing communications efficiency while maintaining effective transfer learning performance between domains. We will discuss DT testbed development to experimentally evaluate domain adaptation techniques for *sim-to-real* policy transfer in wireless networks in Section 6.

3. Data Acquisition

In a DT system, the role of data acquisition is to generate and maintain a virtual environment using ground truth data from the physical domain. In addition to data required to build the virtual model, timely updates from the physical domain are necessary to maintain the accuracy of event prediction, trajectory modeling, and control capabilities in the twin domain. For a DT-enabled wireless network as outlined in Fig. 1, this time-sensitive information can include mobile base station and user locations, performance metrics, and changes to protocol specification such as modulation or bandwidth.

In existing work, especially for physics-based or high-fidelity models, construction of the virtual environment is done manually and prior to simulation events based on expert understanding of the target environment. The majority of works discussed in Sections 1 and 2 demonstrate the use of a virtual environment designed and deployed prior to execution time, and otherwise do not consider an explicit interactive construction of an environment model. For example, the authors of [50] propose a graph-based network topology portrait (NTP) for optical network which does not rely on environmental data to simulate network performance, instead relying on synchronization with hardware. However, especially for dynamic physical environments with low channel coherence time or mobile nodes such as UAV networks [52], the deployment environment may not be known ahead of time and the DT system must be able to generate blockage and boundary rules at execution time in order to provide accurate simulation capabilities.

The authors of [47] describe the virtual environment of a DT system as a repository of environmental and system signatures. Behavioral or physics-based modeling is of key importance to ensure accurate decision-making based on the virtual environment [39], which requires efficient, reliable collection of high-fidelity environmental data. New methods of collecting these signatures automatically are currently being investigated to accelerate the development and deployment of DT systems, especially with the help of robots, UAVs, or other technology to enable autonomous mapping and unassisted control.

In the following section, we discuss the enabling technologies for DT construction and deployment and discuss different methods of environmental data acquisition in this context.

3.1. Enabling Technologies and Techniques

We identify online environment virtualization using various data acquisition techniques as a key enabling technology for real-time and mission-critical applications of DT in unknown physical environments. These techniques include LiDAR [56, 57], millimeter-wave radar [58], and simultaneous localization and mapping (SLAM) [59, 60], to quickly and efficiently scan an environment and build an interactive virtual model. Once an environment is generated, contextual datasets can be generated using ray tracing or other simulation methodologies to accelerate optimization tasks as shown in Fig. 1 [25, 52]. However, the automation of data acquisition to enable online, on-the-fly DT construction is of critical importance to enabling DT for zero-touch networking. Online DT construction in general, to the best of our knowledge, is a challenge that remains unaddressed in existing DT literature.

LiDAR. In recent literature, LiDAR has demonstrated excellent promise for generating high-fidelity environmental models. LiDAR systems detect surface points in an environment by emitting light in the form of pulsed laser and calculating the time-of-flight based on reflections [56]. These points are aggregated in the form of a point cloud highlighting key features in an environment, which is then converted into a representative mesh by a central controller (typically via numerous filtering and reconstruction steps).

Currently, some visual sensor based autonomous vehicles use simplified LiDAR sensors to accurately measure the distance between the vehicle and obstacles, then fuse the LiDAR's data with other sensors' data to make a more robust map. Some LiDAR-based autonomous vehicles, especially UAV

systems, will use high-accuracy LiDAR as a primary sensor to generate the ambient environment's map. In addition, most vacuum robots use a simplified LiDAR system to build the map of the user's house for path planning, collision avoidance, and localization. A LiDAR-equipped UAV is used in [56] to construct an interactive virtual model of an environment in the Unity gaming engine. The Unity engine was selected to maintain the 3D environment model due to its high-quality visualization and integrated physics capabilities. While gaming engines such as Unity are typically optimized for rendering and visualization with user interactivity, they are not generally capable of rendering dynamic meshes from data acquired in real-time.

In general, LiDAR sensors, especially high-fidelity long-range sensors, can be very expensive, ranging from hundreds to tens of thousands of dollars [57]. The sensing range of high-accuracy LiDAR systems can reach more than 400 meters, with an accuracy of 1 cm. However, a high-accuracy LiDAR system is heavy (5 kg+) and expensive ($10,000+), while low-cost, simplified LiDAR systems' scanning frequencies are too low. As a result, high-end LiDAR systems may not be the best choice to enable large-scale aerial mapping, where the weight of sensors must be minimized. Instead, low-cost short-range LiDAR systems can take advantage of the high mobility of UAV systems to map larger areas more effectively, using the UAV flight path to improve area coverage without adding significant weight. For example, the authors of [61] demonstrate the capability of a lightweight (< 1 kg) LiDAR sensor with a range of 200 m deployed on a UAV at an altitude of 10-20 meters to perform large-scale mapping of a 1.7 km coastline.

Mm-Wave Radar. In addition to optical measurement methods, the use of millimeter-wave (mm-wave) radar has attracted attention in recent literature as a method of mapping a physical environment based on measured backscattering and time-of-flight of emitted high-frequency RF signals. Specifically, the use of Y-band (215 GHz) radar for indoor navigation and mapping is demonstrated in [58]. In this work, a portable mm-wave radar system is assembled using commercial-off-the-shelf RF components interfaced with a vector network analyzer to collect range information. The system was mounted on a turntable to enable full rotational scanning, and a LabView interface was developed to control the radar position and data collection. Data was collected by emitting RF signals at 215 GHz with 1 GHz bandwidth at four different types of polarization - HH, HV, VH, and VV - for performance comparison. The reflected signal response was measured at 220 GHz over 5 GHz bandwidth and processed using Hough transform, ghost image elimi-

nation, and false blockage elimination techniques to extract a 2D model of the local environment, providing up to 15 cm resolution when scanning in HH polarization. Since this method supports real-time map generation, it is considered a viable method for simultaneous localization and mapping of DT environments. Towards this goal, the authors of [62] present integrated sensing and communication (ISAC) techniques for UAV-based environment exploration using mm-wave massive MIMO antenna arrays.

The authors of [63] introduce the M-Cube, an experimental software-defined millimeter-wave radio system which is constructed using a low-cost 802.11ad radio and a programmable baseband module. This system can provide full control over MIMO beamforming, providing up to 256 antenna elements across 8 reconfigurable arrays, and has been experimentally validated for both mm-wave (60 GHz) communication at up to 325 Mbps as well as mm-wave radar based on AoA estimation for object detection at a range of 1 m with 8 cm resolution. This system represents a very interesting enabling technology that improves accessibility of experimental mm-wave approaches, reducing the overall cost and complexity of applications and providing support for new sensing-based virtualization techniques.

SLAM. SLAM is the method of creating a feature map of an unknown physical environment while tracking the location of an agent traversing through the environment at the same time, using monocular, stereo, RGB-D, or other visual sensing methods. SLAM requires sensors to detect the environment's features, track specific features then calculate the shape of the physical environment, the carrier's movements, and its relative location. While SLAM systems can leverage a variety of sensor input types, including LiDAR and mm-wave sensors, visual SLAM (V-SLAM) is very popular among different SLAM techniques because it only requires a camera as the input sensor.

SLAM is critical to the process of autonomous virtual environment construction. The authors of [64] propose ORB-SLAM3, an open-source, extensible visual SLAM framework library that supports monocular, stereo and RGB-D cameras for data collection. In general, most SLAM algorithms are executed following three steps: tracking, local mapping, and loop closure. In the tracking phase, points in the environment are used to generate representative images of the environment called keyframes. During local mapping, keyframes are inserted into a local model of the environment. Finally, the loop closure process detects and manages redundant points based on existing keyframes, integrates new information with the global model, and performs bundle adjustment (BA) to estimate camera trajectory. A more detailed

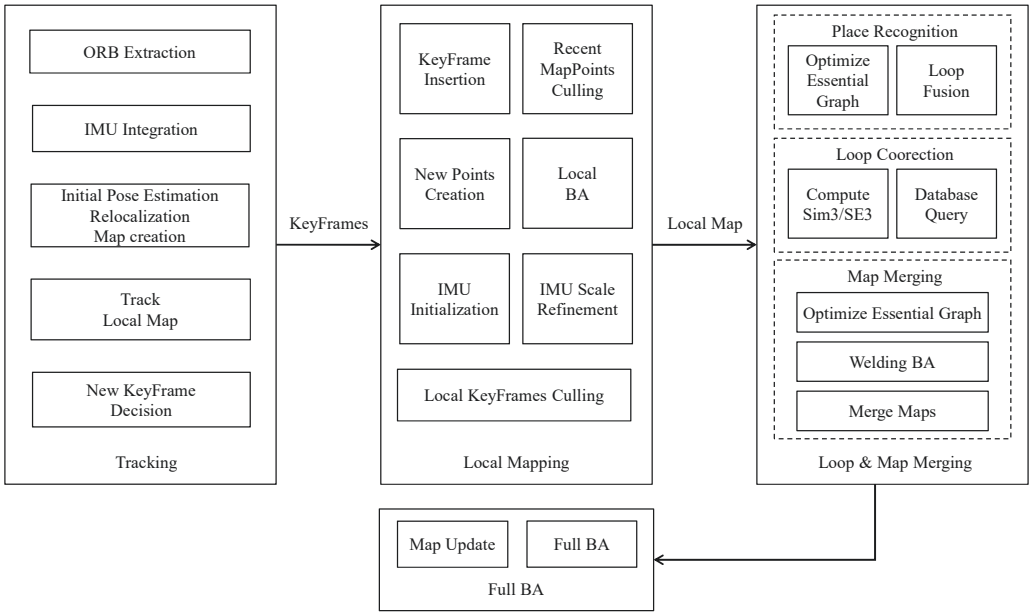


Figure 2: General diagram of ORB-SLAM3 mapping process.

overview of the processes involved in each step of this method is shown in Fig. 2. In ORB-SLAM3, the camera's image input will be fused with acceleration data from an inertial measurement unit (IMU) as shown in Fig. 2. This can significantly improve the mapping accuracy and make the tracking continuous even if visual tracking is lost. The IMU fusion of V-SLAM, deployed in the Tracking and Local Mapping modules in Fig. 2, has been shown to surpass other state-of-the-art SLAM methods on existing datasets collected using stereo and monocular cameras. Additionally, this framework library supports data collection and processing in real time, which is critical for online DT creation and can be leveraged for simultaneous virtualization and interaction.

The authors of [57] compare V-SLAM with LiDAR mapping using low-cost sensors for environmental mapping and mesh generation. The explored sensors include the Intel RealSense ZR300, which leverages stereo IR vision and visual-inertial odometry to perform 3D scanning and localization; the Microsoft Kinect V2, which leverages time of flight of emitted light for 3D sensing, similar to LiDAR; and the Asus ZenFone AR, which leverages a camera, a motion tracking camera, and an IR depth sensor to collect environmental signatures. These three approaches were compared to the ZEB-

	LiDAR	mm-Wave	Monocular Camera	Stereo Camera	RGBD Camera	Monocular Camera-IMU
Range	160m	300m	Relative	35m	5m	Need Further Research
Accuracy	1.5mm	20mm	Relative	20mm	3.7mm	Need Further Research
Cost	$10000	$600	$40	$75	$200	$40

Table 1: Mapping metrics for SLAM systems.

REVO handheld LiDAR system. Each sensor was mounted on an Intel Aero UAV, which navigated around a facility controlled by the native autopilot to collect environmental data. The environmental datasets collected by each sensor were loaded into the Unity game engine for offline visualization, observed in virtual reality using the Oculus Rift headset, and evaluated in terms of accuracy to the modeled environment and resolution of the selected hardware. It was shown that while the ZEB-REVO LiDAR sensor outperformed the other selected options, competent modeling performance can still be achieved for DT environment virtualization using cheaper visual sensor-based methods. Additionally, most modern passenger cars use visual SLAM for Lane Centering Control (LCC) and Adaptive Cruise Control (ACC).

3.2. Research Opportunities and Challenges

The adoption of these methods for DT construction provides the following key research opportunities towards enabling DT in the wireless domain.

Online DT Construction: The construction of a DT is broadly defined as the process by which spatial and temporal data is collected from the physical domain and used to generate a virtual environment in the twin or source domain. Online DT construction implies that data collection and virtual environment construction are parallel complementary processes: as the environmental data is collected, the virtual model is updated faster than the physical environment can transit to the next observable state. For example, the sensing system in [60] updates a local map of the environment roughly every 300 ms. In order for this system to support an accurate real-time representation of the environment for online DT, a physical entity observed by this system cannot generate more than one set of data points in a period shorter than 300 ms. This can be appropriate for wireless applications which observe networking performance over a period seconds or minutes, such as node location optimization [65] or delay-tolerant 5G network services including background file transfer or cloud synchronization [66], but cannot provide sufficient time granularity for tasks such as predictive channel modeling, in which channel coherence time can be $<$20 ms [67]. While [58] and [60] discuss

the potential for online environment construction and virtualization, the efficiency of simultaneous exploration and virtualization of an environment for use in an online DT-enabled wireless simulation remains an open problem.

Continuous SLAM is a special case of online DT construction in which 3-D environmental data is collected continuously via SLAM to update the virtual model in the twin domain. In general, this process will run in parallel with behavioral or analytical simulations in order to maximize spatial virtualization accuracy. An accurate DT simulation relies on the maintenance of the virtual model, and requires constant updates to track or model environmental dynamics in real-time. This is required for intelligence in the source domain to provide timely, adaptive support to agents in the physical domain. Continuous SLAM poses a unique challenge within the scope of online DT construction due to the amount of end-device resources, especially computational capacity and link bandwidth, required to simultaneously virtualize an environment and begin behavioral modeling. Furthermore, behavioral modeling, based on mobility models or historical data, may generalize too much or provide only temporarily valid solutions. Online, continuous generation of an interactive model presents a research opportunity which can significantly improve the state-of-the-art for next-generation wireless networks in dynamic, non-stationary environments.

Large Scale Sensing: Current approaches to environment virtualization pose several limitations when considering sensor accuracy range, especially above 100 m. The price, range, and accuracy of several sensor types are compared in Table 1. The authors of [57, 68] explore the use of UAVs in expanding sensing range for environment data collection, which provides clear advantages in terms of observable area and flexibility compared to manual measurement or static sensing approaches. However, this approach poses several tradeoffs of its own: UAVs are limited in battery life, which inherently limits functional range and on-board hardware; LiDAR systems capable of collecting data at long ranges without loss of fidelity can be prohibitively expensive; and cheaper optical/RGB-D cameras suffer at long ranges (typically > 100 m).

For LiDAR-based SLAM systems, LiDAR tracking is based on time-of-flight measurements, and LiDAR can only measure the distance between the carrier vehicle and landmarks. If there is a moving obstacle between the carrier vehicle and the landmark, LiDAR tracking may be lost. For optical-camera-based SLAM systems, camera tracking is based on angle changes. If the ambient light or viewing angle changes rapidly, then camera tracking

Platform	Fidelity	Accessibility	Type	Physics	Interface
NS-3	High	Free, Open-source	Simulation	Single (RF)	C++, Python
EMANE	Low	Free, Open-source	Emulation	Multi (RF, mobility)	C++, Python, XML
InSite	High	Paid, proprietary	Simulation	Single (RF)	Software GUI
Colosseum	High	Free, proprietary	Emulation	Multi (RF, mobility)	Linux VM
EXata	High	Paid, proprietary	Emulation	Single (RF)	Software GUI
UBSim	Low	Free, Open-source	Simulation	Multi (RF, mobility)	Python

Table 2: Wireless network virtualization platforms.

may be lost. In either case, the relocalization process is time consuming and significantly increases computational complexity of SLAM systems.

In order to prevent tracking loss, multi-sensor fusion can be leveraged to significantly increases the robustness and mapping accuracy of the SLAM system. During continuous tracking, the SLAM system could use movement data of the carrier vehicle to correct the motion-caused deviation and improve tracking accuracy. If the tracking is lost, the SLAM system will still be able to keep updating the map with movement data acquired from the carrier vehicle GPS data or IMU unit.

We envision one possible solution for large-scale DT construction is monocular-IMU data fusion. Traditional monocular camera mapping is a low-cost, low-complexity method for large-scale, low-resolution sensing. However, a monocular camera can only measure the relative distance between objects instead of the absolute distance. Additionally, the point cloud generated by a monocular system is far less dense than other visual methods. To address these challenges in a UAV-based monocular-SLAM system, the camera frame can be fused with the UAV's IMU data to improve monocular mapping accuracy without additional hardware. Furthermore, if the UAVs use accurate GPS service, such as RTK differential GPS, the camera frame can also be integrated with absolute coordinates to further improve mapping accuracy by integrating collected data with geographical information system (GIS) mapping in the DT. However, the application of this approach to enable large-scale environment sensing and virtualization remains an open research challenge in this area.

4. Network Softwarization and Virtualization

In the context of wireless networking research, the benefits of DT have attracted research attention as a key technology towards enabling highly anticipated intelligent networking tasks. The use of high-fidelity simulation

in DT leveraging full environmental modeling for system monitoring and control is the most widely-discussed implementation observed across several industries [29, 69].

Applications of machine learning, especially reinforcement learning and deep learning applications, require significant amounts of time and data to generate and employ optimal control parameters for a given scenario. A source domain containing both simulation and optimization in a centralized intelligence center or edge server allows the synthesis of training data in place of experience which may be otherwise challenging or costly to obtain in the physical domain [53]. This generation of contextual data by a virtual entity, termed "synthetic sensing" [47], is considered a key feature of high-fidelity DT. Specifically, synthetic sensing has been shown to reduce the time cost of dataset generation associated with ML applications to enable intelligent wireless network functionality, such as data collection and reliability, computational capability requirements of end-devices, among others [21]. The fidelity/accuracy of synthetic data available in a DT is directly correlated to the quality and quantity of available data for a physical context [70], as well as the capabilities of the DT platform to process this data. Due to the growing prevalence of software-defined networking (SDN) and virtual network control, virtualization fidelity can vary widely based on the requirements of the application and the tools leveraged to create a virtual environment [38]. We have identified several state-of-the-art network virtualization and softwarization platforms that have demonstrated promising synthetic sensing capabilities to address this challenge, which we will introduce later in this section. In addition to accurate network simulation, we identify several frameworks that have been developed for establishing accurate virtual models of physical scenarios. While supporting experimental literature using these tools for DT development in the wireless domain remains a key open challenge in this area, these tools offer support for the future value of DT for enabling ML applications in the wireless domain.

4.1. State of the Art

We have identified several network simulators that have immediate potential for advancing research into DT for the wireless domain, including NS-3 [71], Colosseum [72], EMANE [73], and Remcom Wireless InSite [74], among others. Refer to Table 2 for a comparison of several key aspects of these platforms in the context of DT system design.

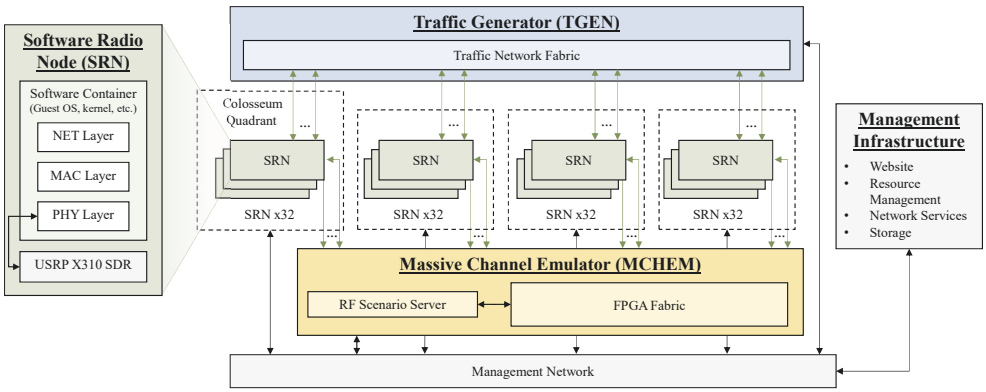


Figure 3: Architecture of Colosseum [78]

NS-3: NS-3 [71] is a popular open-access, open-source network modeling tool in both industry and academia, providing high-fidelity wireless network simulation. NS-3 simulation is built around three major elements: nodes, which serve as basic computing device abstractions on which to run applications and install network devices; packets, which provide data flow between applications; and channels, which are used to connect nodes via installed network devices [71]. All elements in a simulation are constructed from behavioral models written in C++ based on explicit protocol definition at each layer of the network stack [75], with simulation control provided via C++ and Python APIs. Simulated network traffic can be monitored and analyzed using standard network observation software, such as Wireshark [76]. This tool can be directly interfaced with radio hardware such as USRP to perform network emulation as well, improving accuracy of modeled networks.

While NS-3 can provide high-accuracy network simulation, this tool does not support explicit modeling of a physical networking environment. Additionally, NS-3 provides very limited native infrastructure for simulation visualization and data processing and analysis, which necessitates the use of 3rd-party software for these tasks. While its accuracy is still limited by generalizations inherent to model-based simulation [77], this tool is expected to play a significant role in the development of DT-enabled systems in the future due to its high fidelity, accessibility, and large community support.

Colosseum: Colosseum [72] is the world's largest network emulator, comprised of 256 software-defined radios (SDR) to provide a wide variety of emulated RF propagation scenarios. The architecture of Colosseum is shown

in Fig. 3 [78]. Each of the 256 SDR nodes (SRN) is made up of a software container that specifies physical, link, and network layer protocol, as well as a USRP X310 SDR which transmits data generated in the traffic generator (TGEN) based on this protocol stack. The generated signals are transmitted through an FPGA fabric in the massive channel emulator (MCHEM), which is configured to apply channel effects by emulating predefined scenarios stored on the RF scenario server. The platform is accessed, managed, and maintained using a management network connected to all constituent elements.

Similar to NS-3, it is considered an open-access tool, and provides support for a variety of different protocols including 4G/5G and IoT-type protocols with spectrum sharing. While there are many different predefined physical networking scenarios available, current support for custom scenarios is very limited, reducing its flexibility in the context of DT-enabled network deployments. We identify the need for an expansion of this framework to include user-definable networking scenarios with complete control over both network topology and communications protocol *and* agent mobility and behavioral modeling.

EMANE: The Extendable Mobile Ad-Hoc Network Emulator [73], or EMANE, is an open-source real-time framework for highly flexible simulation of mobile network systems. Modular network development allows for independent physical-layer modeling of each network element, providing accurate virtualization of system performance by considering signal propagation, antenna profile effects and interference sources between each emulated wireless link. In general, each emulator instance is comprised of a physical layer model instance paired with one or more radio waveform models, which are designed in C++ and configured using XML. Similar to the Colosseum MCHEM, emulation instances are linked to a shared multicast channel which generates over-the-air network behaviors such as signal propagation, antenna effects, and interference. Additionally, EMANE provides radio waveform model plugins compatible with SDR hardware to enable shared-code emulation, which is comparable to NS-3 emulation capabilities. While EMANE is limited to emulation of PHY and MAC layers, emulation of NET layer and above protocols is typically handled in practice through integration with the Common Open Research Emulator (CORE) [79].

Wireless InSite: Remcom Wireless Insite [74] is a proprietary electromagnetic (EM) propagation modeling tool for wireless networking, which can be leveraged for MIMO dataset generation [25] via ray tracing as discussed in

Section 1. The propagation behaviors are designed around several modeling theories such as Shooting Bouncing Ray (SBR), Adjacent Path Generation (APG), and Finite Difference Time Domain (FDTD), considering environmental reflection behaviors based on the Uniform Theory of Diffraction (UTD) [80]. From these models, this tool can recover significant receiver-side information such as received power, path loss, direction of arrival, delay spread, and interference estimates. Due to its high-fidelity modeling capabilities, this tool provides significant potential for accurate physics-based network event simulation based on signal behaviors within the propagation environment. However, this level of fidelity comes at a significant time cost. While APG with GPU can accelerate some scenarios, in general this tool will require several minutes to calculate network performance for a given deployment, preventing faster-than-real-time applications [80]. Additionally, each scenario will need to be fully re-calculated in the case of mobile transmitter, and partially re-calculated in the case of mobile receiver, further increasing this time cost in mobile networking scenarios.

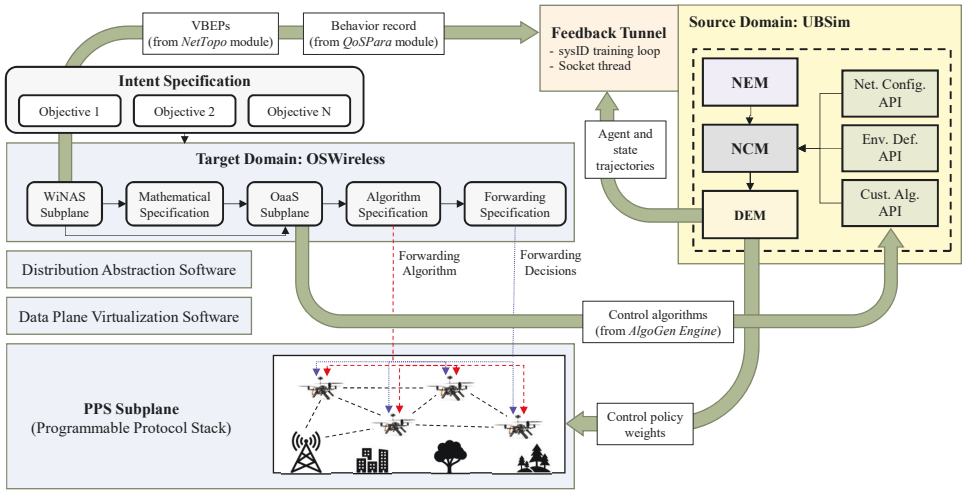


Figure 4: Expansion of UBSim to include OSWireless as a physical domain framework, considering accelerated control algorithm convergence and practical system identification.

ANSYS Twin Builder: The ANSYS Twin Builder [81] presents an open platform for DT development, with a set of built-in tools for physical and behavioral modeling of physical objects. This platform supports multiple modeling domains and languages, enabling multi-physics simulation and heterogeneous data fusion for operation [39]. Specifically, the core capabilities

of Twin Builder rely on two key elements: a multi-domain systems modeler, which can simulate interactions between synchronous modeled systems based on model libraries such as mechanical, hydraulic, and electronic components, logic blocks, and characterized manufacturer's components; and a multi-domain systems solver, which uses existing physics libraries for hydraulics, electronics, pneumatic systems, and thermodynamics to simulate model behaviors. While the platform itself is not immediately optimized for the wireless domain, its use for simulation of hardware within an IoT network, without explicit wireless network dynamics, is discussed in [38]. Twin Builder supports integration of third-party platforms as well, which implies compatibility with other tools capable of explicitly modeling wireless network behaviors.

Spirent 5G DT: The Spirent 5G Digital Twin [82] presents a very robust platform for emulating a full end-to-end 5G network. Various 5G network elements, including independent channel emulation, virtual EPC and gNB, and full-stack end-device emulation, are virtualized with very high fidelity in order to generate cost-effective accurate behavioral analysis, enable evaluation of new security protocols, as well as other "testing on demand" services. The authors of [39] outline several key functionalities of this platform, including wireless network automation and optimization, network slicing via SDR and network functions virtualization (NFV), and accelerated 5G network planning and validation, among others.

Pavatar: In the context of the Internet-of-Things (IoT), intelligent online monitoring systems envisioned for smart cities, Industrial IoT (IIoT), and other next-generation IoT systems are anticipated to play a key role in supporting virtual environments constructed to support a high degree of virtualization [83]. A key example of the capability of high-fidelity DT supported by distributed heterogeneous sensor networks is the Pavatar system [84]. Pavatar collects data at multiple system layers simultaneously in order to conduct comprehensive sensing of all system components and human activities in the operation environment. This heterogeneous data, which is in excess of 1 TB per day, is used to construct a VR representation of every system element for human interfacing, as well as conduct error prediction, anomaly detection, and root-cause diagnosis [84]. The use of different types of data sources (e.g. RF, optical, temporal) to construct a robust system virtualization, termed "data fusion", is necessary to provide accurate simulation of physical system behaviors [33]. In [85], emphasis is placed on detailed modeling of end-device behavior and dynamic agent-based interac-

tions in the virtual space as an integral component of DT for distributed or decentralized networks.

Keysight EXata: EXata [86] is a network digital twin development and analysis tool which uses network emulation and simulation for network virtualization. The platform is based on a software virtual network (SVN) to generate each protocol layer, antenna, and device in the twin domain. This SVN is stated to be interoperable with real radio hardware and capable of interacting with real applications. Additionally, the simulation kernel leverages parallel discrete-event drivers during runtime, which can enable faster-than real-time processing necessary for improved real-time ML algorithm training and deployment.

UBSim: UBSim is a custom hybrid network simulator designed for use in DT research. It is capable of simulating microwave, millimeter-wave, and terahertz-band communications in terrestrial, aerial, or hybrid aerial-ground networking scenarios deployed in a fully configurable physical networking area. It is fully open-source and open-access, written in Python for flexibility, and ease-of-use. It is comprised of three core elements: the network element module, which provides behavioral definitions of all available simulation elements; the network control module, which provides control over all deployed network elements; and the discrete event module, which schedules simulation events. To facilitate ease of use, three sets of APIs have been designed: the environment definition API, which coordinates all environmental features and blockages; the network configuration API, which specifies network topology and communications parameters; and the custom algorithm API, which provides templates for data-driven algorithm deployment. Each UBSim instance also provides a feedback tunnel, as indicated in Fig. 4, to enable socket communications with external software. In order to enable research into key technologies for comprehensive DT as outlined in Section 2, UBSim supports parallel learning across multiple simulation instances, configurable *sim-to-sim* policy transfer[3] for rapid evaluation of domain adaptation algorithms such as robust learning and system identification, and is currently being modified to enable online DT construction using SLAM. UBSim has been leveraged for experiments in domain adaptation [87], UAV network virtualization and optimization [88], and acceleration of machine learning for

[3]Sim-to-sim policy transfer is similar to sim-to-real policy transfer, but the policy is transferred to another simulation environment instead of the real environment.

wireless [65], among others. While the simulation fidelity of UBSim is low, its flexibility is intended to enable integration with high-fidelity platforms such as NS-3 or RF-SITL [89] to enable rapid design and evaluation of DT systems through multi-fidelity, multi-physics experimentation.

4.2. Research Opportunities

The tools discussed in this section provide an interesting scope of customizable, potentially interoperable network simulation at varying levels of fidelity for network virtualization as introduced in Section 1. However, very few tools have been accepted to be individually suitable for full DT implementation following the interdisciplinary feature set shown in Figure 1. Additionally, due to the lack of open-source and community support, they may not provide the level of accessibility required for rapid experimental development in this area. The contribution of a readily available, community-oriented DT platform for the purpose of ML-based wireless network experimentation remains a significant open challenge. It is expected that a combination of these tools, combined with data fusion [16] and platform integration [90], can be leveraged for a widely available, high-fidelity DT toolchain optimized for use in the wireless domain.

Multi-fidelity Simulation: While data-driven methods can provide significant improvements to network performance and services, they can be very time-consuming and data-expensive, and may require re-training if the target environment changes over time. To balance the tradeoff between optimization time and algorithm accuracy, we identify multi-fidelity simulation as an important element of future DT-enabled wireless networking systems. Low-fidelity simulation can be used for time-sensitive tasks, such as rapid control decisions, by leveraging an approximation of wireless network behaviors based on observable performance statistics, while high-fidelity models can be used to maximize network performance in stable environments, or leverage offline optimization algorithms for event prediction.

Standardization: Generating a standardized framework to facilitate high degrees of virtualization in the wireless domain presents many challenges, including simulation for highly complex and volatile networking environments [70], support for dynamic, heterogeneous, and distributed network architectures [39], and ready integration with machine learning and model-based network control [22]. Specifically, we identify the need for an open, accessible DT framework that supports configurable network virtualization at multiple levels of fidelity. The authors of [88] demonstrate preliminary work in this

area, by designing middleware between a high-fidelity UAV network virtualization platform for environmental definition with a low-fidelity network simulator for accelerated convergence of control algorithms. The continued development and distribution of such a framework would enable many contributions in this area, providing a stronger definition of the capabilities of DT technology for use in next-generation wireless networks [49].

Faster-than-real-time Optimization: As part of the envisioned model for 6G network architecture, DT is expected to play a large role in the real-time or faster-than-real-time optimization of network deployments. In order to provide high-fidelity simulation – hence accurate control directives – protocols designed for UAV-assisted communications will require support in virtual environments. Generalizable support for custom wireless protocols is possible on SDR hardware, and can be enabled based on integration of UBSim with OSWireless [91]. OSWireless is a wireless network operating system capable of decomposing operator intent to explicit network control algorithms in a zero-touch manner. Towards practical sim-to-real experimentation, we envision an expansion of UBSim to support integration with OSWireless, as detailed in Fig. 4. Specifically, OSWireless can serve as a physical domain control system to decompose operator intent into custom control algorithms. These control algorithms can be uploaded to UBSim, along with network state data from the Wireless Network Abstraction Specification (WiNAS) Subplane, for faster-than-real-time policy training based on low-fidelity simulation. UBSim will return the optimized control policy to OSWireless for deployment on hardware nodes. To address the *sim-to-real* gap, UBSim will leverage the WiNAS Subplane data to perform system identification, improving fidelity by tuning parameters to match simulation performance to physical domain observations.

5. Domain Adaptation Techniques

As discussed Section 4, synthetic sensing is a useful method to accelerate ML algorithm convergence for practical real-world applications [33, 92]. In general, synthetic data is unable to fully represent all system behaviors in the physical domain, and thus introduces some inaccuracy during algorithm training. Many works seek to minimize this by leveraging high-fidelity models to improve accuracy of synthetic data [93, 94]. These models achieve high fidelity through methods such as ray tracing [95], hardware-in-the-loop emulation [78, 96], or component-level modeling [75, 79]. However, processing

high-fidelity behavioral models can be quite time-consuming, taking possibly several minutes to render a single environment [80]. This trade-off between simulation fidelity and processing time may be problematic for time-critical applications. Domain adaptation seeks to minimize the need for this trade-off by improving the capability of simulation-accelerated learning frameworks to generalize from simulation in the source domain to real-world deployment in the physical domain.

5.1. State of the Art

Instead of seeking a tradeoff between simulation fidelity and operation time, domain adaptation seeks to improve the generalization capability of low fidelity simulation. The ideal domain adaptation system seeks to minimize the importance of simulation fidelity on the accuracy of the resulting control policy in the physical domain, instead focusing on solving the contextual mismatch between domains and overcoming inherent simulation generalization to make sure the resulting control policy works. We introduce three representative examples of this line of research as system identification [97, 98], domain-agnostic feature extraction [99, 100], and robust learning [87, 101, 102]. In system identification, source domain simulation parameters are adapted based on feedback from the physical domain to improve behavioral accuracy. In domain-agnostic feature extraction, contextual features are identified from low-level data to generate a shared observation space across domains. In robust learning, the gap between source and physical domains is estimated through feedback and considered during training in the source domain. Readers are referred to [103] and references therein for a survey of robust reinforcement learning specifically and [104, 105] for other general domain adaptation techniques.

System Identification. System identification is the most established approach to domain adaptation in existing literature [55]. In practice, system identification seeks to iteratively improve behavioral parameters in a source domain simulation based on feedback from the physical domain. An outline of the general premise of system identification is depicted in Fig. 5. While there are many application-specific variants of system identification, this approach faces some challenges in general. Primarily, a significant amount of feedback is required from the target system to validate source domain performance. Additionally, some virtualization platforms such as Colosseum and InSite introduced in Section 4 may not be fully open-source or parameterized to support system identification. Such simulation platforms that are

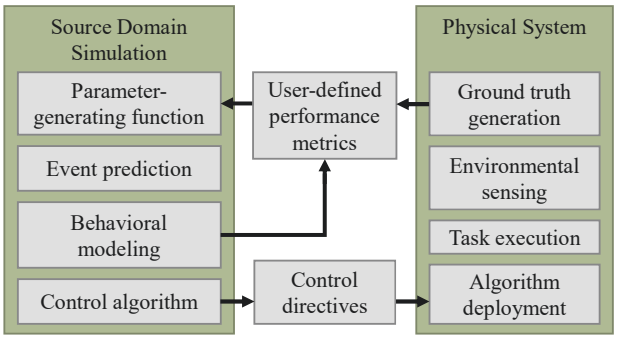


Figure 5: General outline of system identification.

configurable and offer full control of behavioral parameters are termed *hybrid simulators*, due to their analytical and behavioral modeling capabilities.

The authors of [97] explore sim-to-real transfer learning using robot navigation tasks, in which it was noticed that learners in the source domain are capable of exploiting a given simulation to perform tasks beyond capabilities in the real world. By modifying the simulator based on this feedback, the source-to-target gap was reduced and the similarity between simulated and real robot performance was improved. Similar to system identification, the use of domain-agnostic features can be used to enable domain adaptation. Instead of converging simulation parameters to maximize behavioral similarity, source domain observations can be adapted to extract high-level features common to both the source and physical domains, minimizing the impact of domain-specific dynamics on transfer learning performance. The authors of [99] and [106] demonstrate this practice using pixel-level observation adaptation for policy transfer in tasks related to computer vision.

Domain-Agnostic Feature Extraction. This method seeks to reduce the effect of the source-to-target gap by finding commonality between each domain. Specifically, this approach seeks to align behavioral inferences made in each domain through extraction of high- or low-level features that can minimize domain-specific phenomena. For example, the approach to semantic image segmentation outlined in [99] leverages pixel-level segment representation and classification to improve unsupervised adversarial domain adaptation for computer vision. The authors of [100] propose transfer component analysis, which seeks a set of features, termed transfer components, to minimize the difference in data distributions between domains. By projecting transfer components onto a shared latent space and applying standard ma-

chine learning models for classification or regression tasks.

Robustness Mechanisms. Robust learning aims to mitigate the effect of environmental perturbances, such as modeling errors, time-varying dynamics, or unreliable data, on the resulting control policy. This can be typically accomplished by applying a random or adversarial noise process to a system during policy training. Defined in the scope of the DT framework outlined in Fig. 1, this noise is generated by the source domain during policy training to mitigate the effect of the source-to-target gap during policy transfer [87, 101, 102, 107]. In many cases, the noise is added in the form of training samples manually selected from worst-case scenarios or an average of potential environmental anomalies. This is intended to generate a policy that will provide better generalization than non-robust policies when faced with unexpected, unknown, or adversarial physical domain dynamics, at the cost of reduced maximum achievable performance.

An effective approach to applying this policy noise to generate a robust policy is the R-contamination model [101]. Leveraged in [101] and [87] to implement model-free robust reinforcement learning, the R-contamination model is used to probabilistically alter, or "contaminate," observations made by the agent with random or worst-case dynamics to encourage conservative policy learning. Instead of an agent following a deterministic transition kernel p_s^a for a given state s and action a, this contamination probabilistically cause an arbitrary state transition q, selected from an uncertainty set $\mathcal{P}$, which is comprised of all possible transitions in an environment. In order to model contaminated agent trajectories, the R-contamination model generates a subset $\mathcal{P}_s^a$ of $\mathcal{P}$ for each s and a pair according to $\mathcal{P}_s^a = (1-R)p_s^a + Rq_s^a, q_s^a \in \Delta_{|\mathcal{S}|}$, where $\Delta_{|\mathcal{S}|}$ is the simplex of state space $\mathcal{S}$, and R represents the probability of state transition according to q.

It is shown in [87] that the selection of random parameters from the environment can improve policy transfer performance when the source and physical domains have different transition kernels due to differences in the environment dynamics. Both [87] and [101] demonstrate the requirement for careful parameter selection prior to training, highlighting a key limitation of robust learning in the context of domain adaptation. While robust learning can provide very conservative policies, the authors of [107] propose soft-robust learning, which takes an average over the uncertainty set instead of selecting worst-case scenarios to reduce the conservative nature of the resulting policy. This yields a model capable of generalization while limiting performance degradation.

5.2. Research Opportunities

Expertise Incorporated Learning: In order to advance the use of domain adaptation for wireless networks, we consider constraint sampling reinforcement learning (CSRL) [108] as a promising method to quantify the reality gap using domain expertise. In this way, expert knowledge of the networking environment can be integrated during the training process via sensing [56, 60] to enable effective policy transfer in unknown environments with minimal human interaction. Additionally, the authors of [109] underscore the importance of expert knowledge in guaranteeing both the overall accuracy as well as the safety of agents during state space exploration in RL scenarios. Sophisticated applications of this approach, especially in the wireless domain, remain an open challenge in this area.

Reality Gap: The mathematical generalizations present in all simulations make sim-to-real transfer a persistent challenge due to the inherent non-linearities of natural phenomena. Sim-to-sim experiments can be used to estimate the domain transfer performance of new algorithms for domain adaptation, but even high-fidelity models of physical domain hardware are incapable of predicting performance exactly. Additionally, while synthetic sensing can help accelerate convergence of data-driven models, synthetic data will introduce inaccuracies to the learning model based on generalization [98].

To overcome the reality gap between simulation results and physical system behaviors, the authors of [103] discuss different applications of robust learning to provide performance guarantees in the presence of environment uncertainties. While some contributions address challenges associated with the reality gap [53, 97], sim-to-real adaptation in the wireless domain remains an open research area.

Real-time Training: In ML applications, especially methods that leverage neural networks, the training process is generally time-consuming. Even considering faster-than-real-time training capabilities provided by a DT system, significant system or environmental changes may require re-training some or all of a learned policy. In time-critical applications, this may require interim behavioral models to be leveraged during policy re-training. As such, the complexity of ML algorithms require careful consideration of the long convergence times when deployed in an online DT system. The authors of [22] explore the integration of low-complexity inference models with ML algorithms to reduce the impact of long training times on system performance in such scenarios. Similarly, the authors of [110] demonstrate an example of real-time training of a DL-enabled routing algorithm, leveraging standard

non-DL routing algorithms such as round-robin during operation until the DL model can provide improved performance, and continually improving the DL model in the background during online operation. As mentioned in Section 3, implementing this method of real-time ML training in a DT-enabled system requires the update period of the DT to be shorter than the observation period (i.e. data generation rate) of the target environment. To address the case of incomplete data collection due to environmental variance, the authors of [109] investigate the combination of direct environmental observation with a convolutional long short-term memory (ConvLSTM) network for the prediction of missing values to guarantee online performance, with a fixed data collection period proportional to the estimated observation period of the environment.

When simultaneously optimizing multiple agents, as in multi-agent reinforcement learning (MARL), the computation complexity and communication overhead increase exponentially due to the additional problem dimensionality. This may further exaggerate the sim-to-real gap based on the aggregate generalizations made across multiple agent representations in the twin domain. In the example of a self-coordinating swarm UAV network, each agent may be required to relay significant state information including location, speed, height, and network status of itself and other agents to the twin domain in each training step to avoid collisions, reduce interference, and take actions without loss of information. Collecting this information from all agents in each timestep can cause significant communications overhead increasing latency and, as a result, algorithm convergence time [111]. Additionally, as the number of agents increase, the amount of information required by the twin domain to maintain an accurate model of the physical domain system increases as well, which further increases network resource consumption. The complexity of algorithm design and deployment can be further increased when considering a decentralized scenario, in which a twin domain model needs to be maintained at each agent instead of a central controller.

The Advantage Actor-Critic (A2C) algorithm [112] has been demonstrated as an effective tool to minimize policy training times considering high-dimensional state and action spaces. A2C uses the estimated optimal state-action value to update the policy, of which the gradient can be calculated as follows:

$$\nabla_\theta J(\theta) = \mathbb{E}_{\pi_\theta}[\nabla_\theta \log \pi_\theta(s,a) A_\pi(s,a)] \tag{1}$$

where $A_\pi(s,a) = Q_{\pi_\theta}(s,a) - V_{\pi_\theta}(s)$ is termed the *Advantage function*, in which $Q_{\pi_\theta}(s,a)$ is the action-value function and $V_{\pi_\theta}(s)$ is the state-value function. With this advantage function, variance of the gradient can be reduced which improves model training stability. It is very time consuming to find hyperparameters that stabilize the learning process, since A2C relies on the initial estimation of values, therefore A2C algorithms can be challenging to design or further time-consuming for real-time training scenarios.

To further accelerate policy convergence, an Asynchronous Advantage Actor-Critic (A3C) [113] can be used. Similar to A2C, A3C uses an advantage function to reduce variance and improve training stability. The only difference is that A3C allows agents to interact with the environment in parallel. In A3C, virtual agents work individually in multiple instances of the same environment to update a global policy asynchronously. This parallelism can significantly reduce policy convergence times.

To reduce the communication overhead of both centralized and decentralized MARL scenarios, the Lazily Aggregated Policy Gradient (LAPG) method [111] can be used to reduce the frequency of communication. Most information related to collision avoidance and task completion does not need to be exchanged in every time period, e.g., sensor malfunctions, low battery states, and obstacle detection. LAPG sets a trigger condition for this kind of information to reduce the exchange frequency, which can reduce the overall network communication overhead and computational complexity. The lower bound of the trigger condition for LAPG can be written as:

$$||\delta\hat{\nabla}_m^k||^2 \geq \frac{\xi}{\alpha^2 M^2}\sum_{d=1}^{D}||\theta^{k+1-d} - \theta^{k-d}||^2 + 6\sigma^2_{m,N,\delta/K}, \tag{2}$$

where $\delta\hat{\nabla}_m^k$ represents the importance of updating the information, which is calculated by the difference between previous and current policy parameters.

Another strategy that can be used to reduce the per-update communications overhead of a distributed network is by using a Partially Observable Markov Decision Process (POMDP) [114]. Instead of requiring the agents to fully observe the environment in each time step, action selection for each agent is based on a probability distribution given by the model instead of directly observing the underlying state. In this case, each agent has less information to maintain in the local twin domain model and the required information to be shared between agents per network update can be reduced.

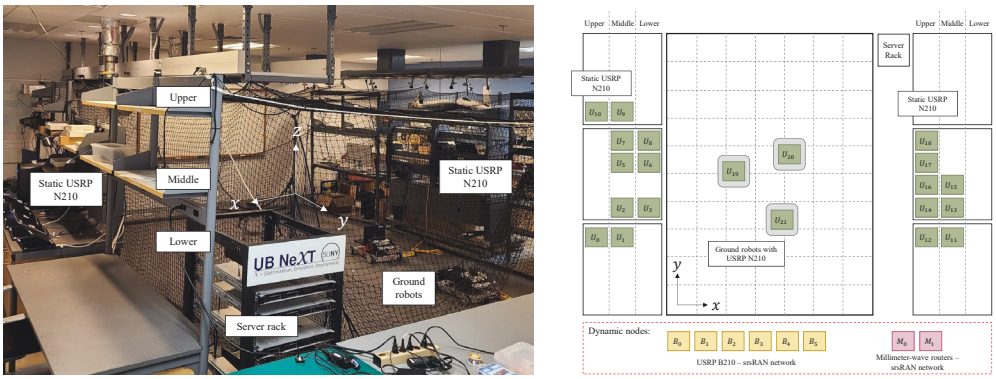
6. Physical Scenarios Development

While several works have proposed DT framework concepts to support adoption of DT-enabled technologies at scale [36, 39, 83], the state of the art in this area generally relies on performance inference based on sim-to-sim experimentation, inflexible virtualization of pre-defined physical scenarios, or small-scale sim-to-real experiments with numerous experimental constraints.

6.1. State of the Art

Scenario development is a key consideration for high-quality wireless experimentation platforms, which must support user-defined network topologies, protocols, and control problems in order to provide accurate validation for use in a practical DT system. The NSF PAWR platforms, including POWDER, COSMOS, AERPAW and ARA, represent the state-of-the-art for wireless network scenario development and experimentation, considering scale, accessibility, and capability. For UAV-enabled wireless networking research, AERPAW [115] provides a large-scale experimentation platform comprised of static nodes, mobile ground nodes, and UAV systems equipped with SDR hardware. The goal of AERPAW is to provide a general platform to develop and evaluate new capabilities for UAV-enabled wireless networks, and is envisioned to enable research into scalable zero-touch control systems for hybrid aerial-ground networks. POWDER [116] is a city-scale wireless networking research testbed in Salt Lake City, Utah, specializing in topics such as 5G O-RAN, massive MIMO, and spectrum sharing in the sub-6 Ghz band. COSMOS [117] is a testbed deployed in an ultra-dense area of New York City, specializing in research for ultra-high-bandwidth, low-latency wireless communications, millimeter-wave MIMO and beamforming, and advanced edge computing scenarios. Finally, ARA [118] is a wireless living laboratory focused on enabling research into rural broadband wireless connectivity by connecting an open-access software-defined virtual infrastructure with a heterogeneous mesh of terrestrial radio hardware and LEO satellite communication terminals, capable of providing 600 square miles of contiguous wireless coverage.

In addition to AERPAW, the UB NeXT testbed [65] provides a comprehensive framework for network virtualization and domain adaptation research in integrated aerial-ground wireless networking. We have included a picture and topology diagram of the UB NeXT testbed in Fig. 6. The NeXT testbed platform is part of the UB indoor autonomy research facility, and is



(a) (b)

Figure 6: (a) Snapshot of the UB NeXT testbed; (b) UB NeXT testbed topology.

comprised of 21 USRP N210 SDRs, 6 USRP B210 SDRs, and two millimeter-wave routers, with mobility support provided by three ground robots with 22 kg payload capacities and a netted UAV enclosure for safe aerial network testing. The testbed networking environment has been fully virtualized in UBSim, which has been demonstrated in [87]. This testbed can enable rapid, small-scale experimentation to address existing challenges in DT research such as evaluation of the *sim-to-real* gap, integrated optimization-learning algorithm design for efficient ML, and virtual network self-configuration via system identification.

6.2. Research Opportunities

Scenario development is at the core of validating DT-enabled experimental frameworks for the wireless domain. We identify several key research opportunities for expanding the scope and depth of continued research in this direction.

Sim-to-real gap Estimation: In general, domain adaptation methods seek to bridge the gap between physical and DT domains. However, especially in the case of robust learning, estimation of the *sim-to-real* gap may not guarantee optimal performance if the gap between physical and DT domain behaviors is large or unknown. Furthermore, since there is no unifying framework for *sim-to-real* gap measurement, methods that seek to reduce the *sim-to-real* gap may require manual tuning in the case of multi-physics optimization. System identification has shown promise in reducing the performance

gap between physical and DT domains for physical scenarios regarding mechanical or robotic systems [98], but there is insufficient investigation into how to quantify the *sim-to-real* gap between different physical scenarios for other methods of domain adaptation. Further research into the measurement or estimation of the *sim-to-real* gap induced by various physical networking scenarios is expected to accelerate design of domain adaptation schemes and advance the state-of-the-art of practical DT-enabled wireless networking.

Portable environments for UBSim: We demonstrate in [87, 119] the need for multiple environmental models to enable experimentation in domain adaptation, with specific attention to both sim-to-sim and sim-to-real gaps. Specifically, more virtual models will be made available for future work to build on the contributions in [87], enabling rapid and repeatable experimentation for domain adaptation in the wireless domain through sim-to-sim experimentation. By virtualizing real testbeds, as done in [87], this will provide preliminary benchmark results required to motivate continued research for sim-to-real transfer.

Building on the sim-to-sim framework outlined in [87], we identify the need for a flexible sim-to-real domain adaptation framework that can accommodate different environmental models based on the physical domain specification. This will enable the design of new domain adaptation algorithms for the wireless domain as well as adaptation of existing algorithms. Using the same simulation platform as [87] and [119], as well as the indoor autonomy research facility and UB SOAR facility at University at Buffalo, many network configurations can be observed, including heterogeneous aerial-ground networks and UAV-to-UAV networks. To achieve the short-term goal of sim-to-real experimentation, we plan direct integration with the UB NeXT testbed platform [65]. In preliminary sim-to-sim experiments, we have virtualized the NeXT testbed [87] and will use this DT environment to better understand the sim-to-real gap through rigorous sim-to-real experimentation and domain adaptation algorithm design.

Testbed Sharing and Remote Access: Considering the need for open, accessible DT experimentation platforms, we believe it is of critical importance to facilitate remote access and control for a fully realized DT-enabled wireless networking testbed. There remains a lack of testbeds and networking environments to enable validation of AI integration and further research into virtualization for network autonomy [16], especially to support advancement towards zero-touch networking. To address this challenge, we emphasize the contributions made in [90] as discussed in Sec. 4, and propose an expansion

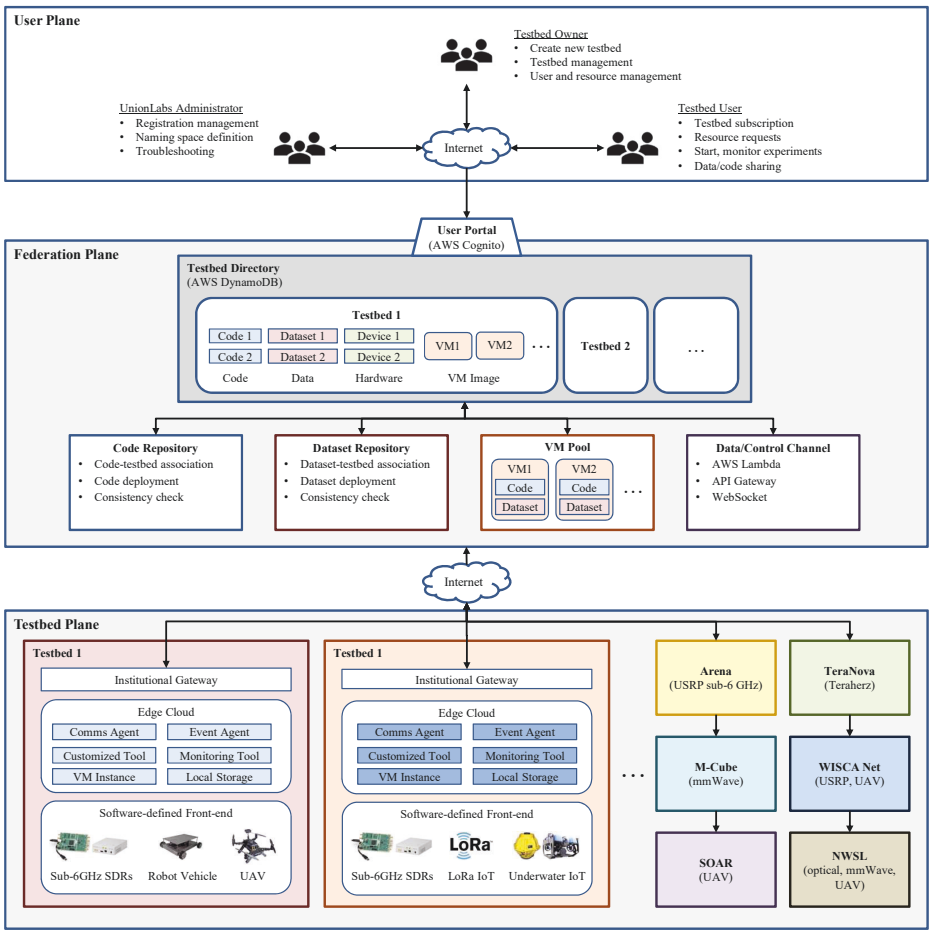


Figure 7: Overview of *UnionLabs* testbed federation.

of the supported framework to include simulation/emulation capabilities. We envision a new framework referred to as *UnionLabs* for testbed sharing and federation. As illustrated in Fig. 7, the architecture of UnionLabs consists of three planes, connected by the internet: the *User Plane*, which handles user/operator interactivity, registration, and management; the *Federation Plane*, which coordinates testbed access and stores experimental code, datasets, and virtual machines; and the *Testbed Plane*, which is comprised of all federated testbeds connected through institutional gateways. This initiative will provide a platform to share code, data, and software/hardware resources across a federation of cloud-enabled heterogeneous testbeds dis-

tributed throughout the country, with an emphasis on the advancement of research topics related to NextG wireless networks, zero-touch and network automation, and the wireless Internet of Things.

7. Conclusions

In this work, we reviewed existing literature regarding the use of DTs for ML-enabled wireless networks with an emphasis on UAV-enabled networking, and discussed the open research challenges in the area. DT for the wireless domain is a particularly important open research area, and can serve as an enabling technology for practical applications of data-driven network self-optimization such as UAV network self-coordination and autonomous network control. Domain adaptation, a key element to bridge the gap between simulations and real network deployments, requires further investigation in the wireless domain. Several methods of domain adaptation, including system identification, domain-agnostic feature extraction, and robust learning have been evaluated for use in a DT system, focusing on limiting interactions between domains to improve data efficiency. However, a comprehensive exploration of the reality gap present in DTs remains an open challenge in this area, as well as accelerating real-time training using domain adaptation in multi-agent systems such as UAV swarm networks. In order to further identify the reality gap across domains, the topic of physical scenario development also needs to be further explored especially for wireless UAV networks.

References

[1] Q. Wang, W. Zhang, Y. Liu, Y. Liu, Multi-UAV dynamic wireless networking with deep reinforcement learning, IEEE Communications Letters 23 (12) (2019) 2243–2246.

[2] Z. Guan, N. Cen, T. Melodia, S. Pudlewski, Self-Organizing Flying Drones with Massive MIMO Networking, in: Proc. of Mediterranean Ad Hoc Networking Workshop (Med-Hoc-Net), Capri, Italy, 2018.

[3] Q. Zhang, M. Jiang, Z. Feng, W. Li, W. Zhang, M. pan, IoT-enabled UAV: Network architecture and routing algorithm, IEEE Internet of Things Journal 6 (2) (2019) 3727–3742.

[4] X. Chen, T. Chen, Z. Zhao, H. Zhang, M. Bennis, Y. JI, Resource Awareness in Unmanned Aerial Vehicle-Assisted Mobile-Edge Computing Systems, in: Proc. of IEEE 91st Vehicular Technology Conference (VTC2020-Spring), Antwerp, Belgium, 2020.

[5] A. Garcia-Rodriguez, G. Geraci, D. López-Pérez, L. G. Giordano, M. Ding, E. Björnson, The Essential Guide to Realizing 5G-Connected UAVs with Massive MIMO, IEEE Communications Magazine 57 (12) (2019) 84–90.

[6] P. Chandhar, D. Danev, E. G. Larsson, Massive MIMO for Communications With Drone Swarms, IEEE Trans. on Wireless Communications 17 (3) (2018) 1604–1629.

[7] W. Sun, N. Xu, L. Wang, H. Zhang, Y. Zhang, Dynamic Digital Twin and Federated Learning with Incentives for Air-Ground Networks, IEEE Transactions on Network Science and Engineering 9 (1) (2020) 321–333.

[8] N. C. Luong, D. T. Hoang, S. Gong, D. Niyato, P. Wang, Y.-C. Liang, D. I. Kim, Applications of deep reinforcement learning in communications and networking: A survey, IEEE Commun. Surv. Tutor. 21 (4) (2019) 3133–3174.

[9] S. K. Moorthy, Z. Guan, Beam Learning in MmWave/THz-band Drone Networks Under In-Flight Mobility Uncertainties, IEEE Transactions on Mobile Computing 21 (6) (2022) 1945–1957.

[10] Z. Su, W. Feng, J. Tang, Z. Chen, Y. Fu, N. Zhao, K.-K. Wong, Energy Efficiency Optimization for D2D Communications Underlaying UAV-assisted Industrial IoT Networks with SWIPT, IEEE Internet of Things Journal (early access) (January 2022).

[11] H. Alghafari, M. SayadHaghighi, Decentralized Joint Resource Allocation and Path Selection in Multi-hop Integrated Access Backhaul 5G Networks, Computer Networks 207 (April 2022).

[12] J. Du, W. Liu, G. Lu, J. Jiang, D. Zhai, F. R. Yu, Z. Ding, When Mobile-Edge Computing (MEC) Meets Nonorthogonal Multiple Access (NOMA) for the Internet of Things (IoT): System Design and Optimization, IEEE Internet of Things Journal 8 (10) (2021) 7849–7862.

[13] Y. Cheng, K. H. Li, Y. Liu, K. C. Teh, G. K. Karagiannidis, Non-Orthogonal Multiple Access (NOMA) with Multiple Intelligent Reflecting Surfaces, IEEE Transactions on Wireless Communications 20 (11) (2021) 7184–7195.

[14] Z. Guan, T. Melodia, CU-LTE: Spectrally-Efficient and Fair Coexistence Between LTE and Wi-Fi in Unlicensed Bands, in: Proc. of IEEE Intl. Conference on Computer Communications (INFOCOM), San Francisco, CA, USA, 2016.

[15] J. Hu, S. K. Moorthy, A. Harindranath, Z. Guan, N. Mastronarde, E. S. Bentley, S. Pudlewski, SwarmShare: Mobility-Resilient Spectrum Sharing for Swarm UAV Networking in the 6 GHz Band,", in: Proc. of IEEE International Conference on Sensing, Communication and Networking (SECON), Virtual Conference, 2021.

[16] E. Coronado, R. Behravesh, T. Subramanya, A. Fernández-Fernández, S. Siddiqui, X. Costa-Pérez, R. Riggio, Zero Touch Management: A Survey of Network Automation Solutions for 5G and 6G Networks, IEEE Surveys & Tutorials (Early Access) (October 2022).

[17] H. Li, K. Ota, M. Dong, Learning IoT in Edge: Deep Learning for the Internet of Things with Edge Computing, IEEE Network 32 (1) (2018) 96–101.

[18] R. Liu, M. Lee, G. Yu, G. Y. Li, User Association for Millimeter-Wave Networks: A Machine Learning Approach, IEEE Transactions on Communications 68 (7) (2020) 4162–4174.

[19] X. Liu, Y. Liu, Y. Chen, Machine Learning Empowered Trajectory and Passive Beamforming Design in UAV-RIS Wireless Networks, IEEE Journal on Selected Areas in Communications 39 (7) (2021) 2042–2055.

[20] S. Messaoud, A. Bradai, O. B. Ahmed, P. T. A. Quang, M. Atri, M. S. Hossain, Deep Federated Q-Learning-Based Network Slicing for Industrial IoT, IEEE Transactions on Industrial Informatics 17 (8) (2021) 5572–5582.

[21] C. She, R. Dong, Z. Gu, Z. Hou, Y. Li, W. Hardjawana, C. Yang, L. Song, B. Vucetic, Deep Learning for Ultra-Reliable and Low-Latency

Communications in 6G Networks, IEEE Network 34 (5) (2020) 219–225.

[22] C. She, C. Sun, Z. Gu, Y. Li, C. Yang, H. V. Poor, B. Vucetic, A Tutorial on Ultra-Reliable and Low-Latency Communications in 6G: Integrating Domain Knowledge into Deep Learning, arxiv.org (Jan. 2020).
URL https://arxiv.org/abs/2009.06010

[23] S. AbdulRahman, H. Tout, H. Ould-Slimane, A. Mourad, C. Talhi, M. Guizani, A Survey on Federated Learning: The Journey From Centralized to Distributed On-Site Learning and Beyond, IEEE Internet of Things Journal 8 (7) (2021) 5476–5497.

[24] A. Feriani, E. Hossain, Single and Multi-Agent Deep Reinforcement Learning for AI-Enabled Wireless Networks: A Tutorial, IEEE Communications Surveys & Tutorials 23 (2) (2021) 1226–1252.

[25] A. Alkhateeb, Deepmimo: A generic deep learning dataset for millimeter wave and massive mimo applications, arXiv preprint arXiv:1902.06435 (February 2019).

[26] S. Levine, A. Kumar, G. Tucker, J. Fu, Offline Reinforcement Learning: Tutorial, Review, and Perspectives on Open Problems, arXiv.org (Nov. 2020).
URL https://arxiv.org/abs/2005.01643

[27] D. Jones, C. Snider, A. Nassehi, J. Yon, B. Hicks, Characterising the Digital Twin: A Systematic Literature Review, CIRP Journal of Manufacturing Science and Technology (Elsevier) 29 (2020) 36–52.

[28] K. Xia, C. Sacco, M. Kirkpatrick, C. Saidy, L. Nguyen, A. Kircaliali, R. Harik, A digital twin to train deep reinforcement learning agent for smart manufacturing plants: Environment, interfaces and intelligence, Journal of Manufacturing Systems 58 (B) (2021) 210–230.

[29] Q. Qi, F. Tao, Digital Twin and Big Data Towards Smart Manufacturing and Industry 4.0: 360 Degree Comparison, IEEE Access 6 (2018) 3585–3593.

[30] E. Ak, et al., T6CONF: Digital Twin Networking Framework for IPv6-Enabled Net-Zero Smart Cities, IEEE Communications Magazine 61 (3) (2023) 36–42.

[31] D. Ford, C. Wolf, Smart Cities with Digital Twin Systems for Disaster Management, Journal of Management in Engineering 36 (4) (Apr. 2020).

[32] S. Ivanov, K. Nikolskaya, G. Radchenko, L. Sokolinsky, M. Zymbler, Digital Twin of City: Concept Overview, in: Proc. of Global Smart Industry Conference (GloSIC), Chelyabinsk, Russia, 2020.

[33] R. Minerva, G. M. Lee, N. Crespi, Digital Twin in the IoT Context: A Survey on Technical Features, Scenarios, and Architectural Models, Proceedings of the IEEE 108 (10) (2020) 1785–1824.

[34] R. Dong, C. She, W. Hardjawana, Y. Li, B. Vucetic, Deep Learning for Hybrid 5G Services in Mobile Edge Computing Systems: Learn From a Digital Twin, IEEE Trans. on Wireless Communications 18 (10) (2019) 4692–4707.

[35] W. Sun, H. Zhang, R. Wang, Y. Zhang, Reducing Offloading Latency for Digital Twin Edge Networks in 6G, IEEE Transactions on Vehicular Technology 69 (10) (2020) 12240–12251.

[36] L. U. Khan, W. Saad, D. Niyato, Z. Han, C. S. Hong, Digital-Twin-Enabled 6G: Vision, Architectural Trends, and Future Directions, arXiv.org (February 2021).
URL `https://arxiv.org/abs/2102.12169`

[37] A. Rasheed, O. San, T. Kvamsdal, Digital Twin: Values, Challenges and Enablers From a Modeling Perspective, IEEE Access 8 (2020) 21980–22012.

[38] Q. Qi, F. Tao, T. Ho, N. Anwer, A. Liu, Y. Wei, L. Wang, A. Nee, Enabling Technologies and Tools for Digital Twin, Journal of Manufacturing Systems 58 (2021) 3–21.

[39] H. X. Nguyen, R. Trestian, D. To, M. Tatipamula, Digital Twin for 5G and Beyond, IEEE Communications Magazine 59 (2) (2021) 10–15.

[40] D. Wang, Z. Zhang, M. Zhang, M. Fu, J. Li, S. Cai, C. Zhang, X. Chen, The Role of Digital Twin in Optical Communication: Fault Management, Hardware Configuration, and Transmission Simulation, IEEE Communications Magazine 59 (1) (2021) 133–139.

[41] L. Lei, Y. Tan, K. Zheng, S. Liu, K. Zhang, X. Shen, Deep reinforcement learning for autonomous Internet of Things: model, applications and challenges, IEEE Communications Surveys & Tutorials 22 (3) (2020) 1722–1760.

[42] N. Luong, D. Hoang, S. Gong, D. Niyato, P. Wang, Y. Liang, D. Kim, Applications of Deep Reinforcement Learning in Communications and Networking: A Survey, IEEE Communications Surveys & Tutorials 21 (4) (2019) 3133–3174.

[43] A. Feriani, E. Hossain, Single and multi-agent deep reinforcement learning for AI-enabled wireless networks: a tutorial, IEEE Communications Surveys & Tutorials 23 (2) (2021) 1226–1252.

[44] Y. Qian, J. Wu, R. Wang, W. Zhu, W. Zhang, Survey on Reinforcement Learning Applications in Communication Networks, Journal of Communication and Information Networks 4 (2) (2019) 30–39.

[45] A. E. Campos-Ferreira, J. de J. Lozoya-Santos, A. Vargas-Martínez, R. R. Mendoza, R. Morales-Menéndez, Digital Twin Applications: A review, Memorias del Congreso Nacional de Control Automático (2019) 606–611.

[46] R. Saracco, Digital Twins: Bridging Physical Space and Cyberspace, IEEE Computer 52 (12) (2019) 58–64.

[47] R. Minerva, F. M. Awan, N. Crespi, Exploiting Digital Twin as Enablers for Synthetic Sensing, IEEE Internet Computing 26 (5) (2021) 61–67.

[48] J. M. Rozanec, L. Jinzhi, Towards Actionable Cognitive Digital Twins for Manufacturing, in: Proc. of the International Workshop on Semantic Digital Twins, Heraklion, Greece, 2020.

[49] N. Kuruvatti, M. Habibi, S. Partani, B. Han, A. Fellan, H. Schotten, Empowering 6G Communication Systems With Digital Twin Technology: A Comprehensive Survey, IEEE Access 10 (2022) 112158–112186.

[50] Y. Wu, M. Zhang, L. Zhang, J. Li, X. Chan, D. Wang, Dynamic Network Topology Portrait for Digital Twin Optical Network, Journal of Lightwave Technology 41 (10) (2023) 2953–2968.

[51] L. Lei, G. Shen, L. Zhang, Z. Li, Toward Intelligent Cooperation of UAV Swarms: When Machine Learning Meets Digital Twin, IEEE Network 35 (1) (2021) 386–392.

[52] L. Lei, G. Shen, L. Zhang, Z. Li, Toward Intelligent Cooperation of UAV Swarms: When Machine Learning Meets Digital Twin, IEEE Network 35 (1) (2021) 386–392.

[53] K. Bousmalis, A. Irpan, P. Wohlhart, Y. Bai, M. Kelcey, M. Kalakrishnan, L. Downs, J. Ibarz, P. Pastor, K. Konolige, S. Levine, V. Vanhoucke, Using Simulation and Domain Adaptation to Improve Efficiency of Deep Robotic Grasping, in: Proc. of IEEE International Conference on Robotics and Automation, Brisbane, Australia, 2018.

[54] Y. Chebotar, V. Makoviychuk, M. Macklin, J. Isaac, N. Ratliff, D. Fox, Closing the sim-to-real loop: Adapting simulation randomization with real world experience, in: Proc. of IEEE International Conference on Robotics and Automation, Montreal, Canada, 2019.

[55] B. Eysenbach, S. Chaudhari, S. Asawa, S. Levine, Off-Dynamics Reinforcement Learning: Training for Transfer with Domain Classifiers, in: Proc. of International Conference on Learning Representations, Vienna, Austria, 2021.

[56] E. Brock, C. Huang, D. Wu, Y. Liang, LiDAR-Based Real-Time Mapping for Digital Twin Development, in: Proc. of IEEE International Conference on Multimedia and Expo, Shenzhen, China, 2021.

[57] M. Minos-Stensrud, O. H. Haakstad, O. Sakseid, B. Westby, , A. Alcocer, Towards Automated 3D Reconstruction in SME Factories and Digital Twin Model Generation, in: Proc. of International Conference on Control, Automation and Systems, PyeongChang, GangWon, Korea, 2018.

[58] M. Moallem, K. Sarabandi, Polarimetric Study of MMW Imaging Radars for Indoor Navigation and Mapping, IEEE Transactions on Antennas and Propagation 62 (1) (2014) 500–504.

[59] R. Mur-Artal, J. D. Tardos, ORB-SLAM2: An Open-Source SLAM System for Monocular, Stereo, and RGB-D Cameras, IEEE Transactions on Robotics 33 (5) (2017) 1255–1262.

[60] A. Ali, Z. S. Hashemifar, K. Dantu, Edge-SLAM: Edge-Assisted Visual Simultaneous Localization and Mapping, in: Proc. of International Conference on Mobile Systems, Applications, and Services, Toronto, Ontario, Canada, 2020.

[61] Y. Lin, Y. Cheng, T. Zhou, R. Ravi, S. M. Hasheminasab, J. E. Flatt, C. Troy, A. Habib, Evaluation of UAV LiDAR for Mapping Coastal Environments, Remote Sensing 11 (24) (2019) 2893–2925.

[62] Z. Wan, Z. Gau, S. Tan, L. Fang, Joint Channel Estimation and Radar Sensing for UAV Networks with mmWave Massive MIMO, in: International Wireless Communications and Movile Computing (IWCMC, Dubrovnik, Croatia, 2022.

[63] R. Zhao, T. Woodford, T. Wei, K. Qian, X. Zhang, M-Cube: A Millimeter-Wave Massive MIMO Software Radio, in: Proc. of International Conference on Mobile Computing and Networking, London, United Kingdom, 2020.

[64] C. Campos, R. Elvira, J. J. Gomez, J. M. M. Montiel, J. D. Tardos, ORB-SLAM3: An accurate open-source library for visual, visual-inertial and multi-map SLAM, IEEE Transactions on Robotics 37 (6) (2021) 1874–1890.

[65] J. Hu, M. McManus, S. K. Moorthy, Y. Cui, Z. Guan, N. Mastronarde, E. S. Bentley, M. Medley, NeXT: A Software-Defined Testbed with Integrated Optimization, Simulation and Experiment, in: Proc. of 2022 IEEE Future Networks World Forum (FNWF), Montreal, Canada, 2022.

[66] A. Prasad, P. Lunden, M. Moisio, M. Uusitalo, Z. Li, Efficient mobility and traffic management for delay tolerant cloud data in 5G networks,

in: IEEE 26th Annual Symposium on Personal, Indoor, and Mobile Radio Communications, Hong Kong, China, 2015.

[67] N. Simmons, et al., AI-based Channel Prediction in D2D Links: An Empirical Validation, IEEE Access 10 (2022) 65459–65472.

[68] K. Chiang, G. Tsai, Y. Li, N. El-Sheimy, Development of LIDAR-based UAV system for environment reconstruction, IEEE Geoscience and Remote Sensing Letters 14 (10) (2017) 1790–1794.

[69] E. H. Glaessgen, D. S. Stargel, The Digital Twin Paradigm for Future NASA and U.S. Air Force Vehicles, in: Proc. of Structures, Structural Dynamics and Materials Conference - Special Session on the Digital Twin, Honolulu, HI, 2012.

[70] T. Yang, J. Chen, N. Zhang, AI-Empowered Maritime Internet of Things: A Parallel-Network-Driven Approach, IEEE Network 34 (5) (2020) 54–59.

[71] NSNAM, NS-3 Network Simulator, (2011-2021).
URL https://www.nsnam.org/

[72] Colosseum, NSF Colosseum: The World′s Most Powerful Wireless Network Emulator.
URL https://www.northeastern.edu/colosseum/

[73] AdjacentLink, EMANE: Extendable Mobile Ad-Hoc Networking Emulator.
URL https://adjacentlink.com/documentation/emane/v1.2.1/

[74] Remcom, Wireless InSite 3D Wireless Prediction Software, (2021).
URL https://www.remcom.com/wireless-insite-em-propagation-software

[75] R. Chaudhary, S. Sethi, R. Kechari, S. Goel, A study of comparison of Network Simulator -3 and Network Simulator -2, International Journal of COmputer Science and Information Technologies 3 (1) (2012) 3085–3092.

[76] G. C. et al, Wireshark.
URL https://www.wireshark.org/

[77] P. Fuxjaeger, Validation of the NS-3 interference model for IEEE802.11 networks, in: Proc. of IFIP Wireless and Mobile Networking Conference (WMNC), Munich, Germany, 2015.

[78] L. Bonati, et al., Colosseum: large-scale wireless experimentation through hardware-in-the-loop network emulation, in: Proc. of IEEE International Symposium on Dynamic Spectrum Access Networks, Virtual Conference, 2021.

[79] J. Ahrenholz, T. Goff, B. Adamson, Integration of the CORE and EMANE network emulators, in: Proc. of Military Communications Conference (MILCOM), Baltimore, MD, USA, 2011.

[80] M. Schmiedekamp, A. Kuhlman, R. Ohs, S. Buscemi, High fidelity modeling of spatio-temporally dense multi-radio scenarios, in: Proc. of Annual conference of the Applied Computational Electromagnetics Society (ACES), Denver, CO, USA, 2013.

[81] ANSYS, ANSYS Twin Builder.
URL `https://www.ansys.com/products/digital-twin/ansys-twin-builder`

[82] Spirent, 5G Network Digital Twin Whitepaper.
URL `https://www.spirent.com/assets/wp_simplifying-5g-with-the-network-digital-twin`

[83] K. M. Alam, A. E. Saddik, C2PS: A Digital Twin Architecture Reference Model for the Cloud-Based Cyber-Physical Systems, IEEE Access 5 (2017) 2050–2062.

[84] Y. He, J. Guo, X. Zheng, From Surveillance to Digital Twin: Challenges and Recent Advances of Signal Processing for the Industrial Internet of Things, IEEE Signal Processing Magazine 35 (5) (2018) 120–129.

[85] T. Jung, N. Jazdi, M. Weyrich, A Survey on Dynamic Simulation of Automation Systems and Components in the Internet of Things, in: Proc. of IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), Limassol, Cyprus, 2017.

[86] Keysight, EXata Network Modeling.
URL https://www.keysight.com/us/en/product/SN100EXBA/exata-network-modeling.html

[87] M. McManus, Z. Guan, N. Mastronarde, S. Zou, On the Source-to-Target Gap of Robust Double Deep Q Learning in Digital Twin Enabled Wireless Networks, in: Proc. of SPIE Big Data IV: Learning, Analytics, and Applications, Orlando, Florida, United States, 2022.

[88] S. K. Moorhty, A. Harindranath, M. McManus, Z. Guan, N. Mastronarde, E. S. Bentley, M. Medley, A Middleware for Digital Twin-Enabled Flying Network Simulations Using UBSim and UB-ANC, in: Proc. of International Conference on Distributed Computing in Sensor Systems (DCOSS), Los Angeles, CA, USA, 2022.

[89] N. Mastronarde, D. Russell, Z. Guan, G. Sklivanitis, D. Pados, E. S. Bentley, M. Medley, RF-SITL: a software-in-the-loop channel emulator for UAV swarm networks, in: Proc. of International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM), Belfast, United Kingdom, 2022.

[90] S. K. Moorthy, C. Lu, Z. Guan, N. Mastronarde, G. Sklivanitis, D. Pados, E. S. Bentley, M. Medley, CloudRAFT: A Cloud-based Framework for Remote Experimentation for Mobile Networks, in: Proc. of CCNC 2022 WKSHPS: 2nd International Workshop on Communication and Networking for Swarms Robotics (RoboCom 2022), Virtual Conference, 2022.

[91] S. K. Moorthy, Z. Guan, N. M. E. S. Bentley, M. Medley, OSWireless: Enhancing Automation for Optimizing Intent-Driven Software-Defined Wireless Networks, in: Proc. of IEEE International Conference on Mobile Ad-Hoc and Smart Systems (MASS), Denver, CO, USA, 2022.

[92] H. Viswanathan, P. E. Mogensen, Communications in the 6G Era, IEEE Access 8 (2020).

[93] M. Tehrani-Moayyed, L. Bonati, P. Johari, T. Melodia, S. Basagni, Creating RF Scenarios for Large-scale, Real-time Wireless Channel Emulators, in: Proc. of Mediterranean Communication and Computer Networking Conference (MedComNet), Ibiza, Spain, 2021.

[94] B. Sheen, J. Yang, X. Feng, , M. M. U. Chowdhury, A Digital Twin for Reconfigurable Intelligent Surface Assisted Wireless Communication, arxiv.org (Sept. 2020).
URL https://arxiv.org/abs/2009.00454

[95] A. Liu, V. Lau, Optimization of Multi-UAV-Aided Wireless Networking Over a Ray-Tracing Channel Model, IEEE Transactions on Wireless Communications 19 (9) (2019) 4518–4530.

[96] L. Baumgartner, M. Bauer, B. Bloessl, SUN: A Simulated UAV Network Testbed with Hardware-in-the-Loop SDR Support, in: IEEE Wireless Communications and Networking Conference, Glasgow, United Kingdom, 2023.

[97] A. Kadian, J. Truong, A. Gokaslan, A. Clegg, E. Wijmans, S. Lee, M. Savva, S. Chernova, D. Batra, Sim2Real predictivity: does evaluation in simulation predict real-world performance?, IEEE Robotics and Automation Letters 5 (4) (2020) 6670–6677.

[98] Y. Jiang, T. Zhang, D. Ho, Y. Bai, C. K. Liu, S. Levine, J. Tan, SimGAN: Hybrid simulator identification for domain adaptation via adversarial reinforcement learning, in: Proc. of IEEE International Conference on Robotics and Automation, Xi'an, China, 2021.

[99] R. Romijnders, P. Meletis, G. Dubbelman, A domain agnostic normalization layer for unsupervised adversarial domain adaptation, in: Proc. of IEEE Winter Conference on Applications of Computer Vision, Hawaii, HI, United States, 2019.

[100] S. J. Pan, I. W. Tsang, J. T. Kwok, Q. Yang, Domain adaptation via transfer component analysis, IEEE Transactions on Neural Networks 22 (2) (2011) 199–210.

[101] Y. Wang, S. Zou, Online robust reinforcement learning with model uncertainty, in: Proc. of Advances in Neural Information Processing Systems, Virtual Conference, 2021.

[102] M. Khodabandeh, A. Vahdat, M. Ranjbar, W. Macready, A robust learning approach to domain adaptive object detection, in: Proc. of International Conference on Computer Vision, Seoul, Korea, 2019.

[103] S. Chen, Y. Li, An overview of robust reinforcement learning, in: Proc. of IEEE International Conference on Networking, Sensing, and Control, Nanjing, China, 2020.

[104] W. M. Kouw, M. Loog, A review of domain adaptation without target labels, IEEE Transactions on Pattern Analysis and Machine Intelligence 43 (3) (2021) 766–785.

[105] G. Wilson, D. J. Cook, A survey of unsupervised deep domain adaptation, ACM Transactions on Intelligent System Technology 11 (5) (2020) 51:1–46.

[106] K. Bousmalis, A. Irpan, P. Wohlhart, Y. Bai, M. Kelcey, M. Kalakrishnan, L. Downs, J. Ibarz, P. Pastor, K. Konolige, S. Levine, V. Vanhoucke, Using simulation and domain adaptation to improve efficiency of deep robotic grasping, in: Proc. of IEEE International Conference on Robotics and Automation, Brisbane, Australia, 2018.

[107] E. Derman, D. J. Mankowitz, T. A. Mann, S. Mannor, Soft-robust actor-critic policy-gradient, arxiv.org (2018).
URL `https://arxiv.org/pdf/1803.04848.pdf`

[108] T. Mu, G. Theocharous, D. Arbour, E. Brunskill, Constraint Sampling Reinforcement Learning: Incorporating Expertise for Faster Learning, in: Proc. of AAAI Conference on Artificial Intelligence, Virtual Conference, 2022.

[109] J. Deng, et al., A Digital Twin Approach for Self-optimization of Mobile Networks, in: Proc. of IEEE Wireless Communications and Networking Conference Workshops, Nanjing, China, 2021.

[110] F. Tang, et al., On Removing Routing Protocol from Future Wireless Networks: A Real-time Deep Learning Approach for Intelligent Traffic Control, IEEE Wireless Communications 25 (1) (2017) 154–160.

[111] T. Chen, K. Zhang, G. B. Giannakis, T. Başar, Communication-efficient policy gradient methods for distributed reinforcement learning, IEEE Transactions on Control of Network Systems 9 (2) (2022) 917–929.

[112] X. Han, J. Wang, Q. Zhang, X. Qin, M. Sun, Multi-uav automatic dynamic obstacle avoidance with experience-shared a2c, in: Proc. of 2019 International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob), 2019, pp. 330–335. doi:10.1109/WiMOB.2019.8923344.

[113] L. Liu, J. Feng, Q. Pei, C. Chen, Y. Ming, B. Shang, M. Dong, Blockchain-enabled secure data sharing scheme in mobile-edge computing: An asynchronous advantage actorcritic learning approach, IEEE Internet of Things Journal 8 (4) (2021) 2342–2353.

[114] M. Lauri, D. Hsu, J. Pajarinen, Partially observable markov decision processes in robotics: A survey, IEEE Transactions on Robotics (early access) (2022) 1–20.

[115] V. Marojevic, I. Guvenc, R. Dutta, M. L. Sichitiu, B. Floyd, Advanced Wireless for Unmanned Aerial Systems: 5G Standardization, Research Challenges, and AERPAW Architecture, IEEE Vehicular Technology Magazine 15 (2) (2020) 22–30.

[116] J. Breen, et al., Powder: Platform for Open Wireless Data-driven Experimental Research, in: Proc. of International Workshop on Wireless Network Testbeds, Experimental evaluation and Characterization (WiNTECH, London, United Kingdom, 2020.

[117] D. Raychaudhari, et al., Challenge: COSMOS: A city-scale programmable testbed for experimentation with advanced wireless, in: Proc. of International Conference on Mobile Computing and Networking (MobiCom), London, United Kingdom, 2020.

[118] H. Zhang, et al., ARA: A wireless living lab vision for smart and connected rural communities, in: Proc. of 15th ACM Workshop on Wireless Network Testbeds, Experimental Evaluation and CHaracterization (WiNTECH), New Orleans, Louisiana, USA, 2022.

[119] S. K. Moorthy, M. McManus, Z. Guan, ESN Reinforcement Learning for Spectrum and Flight Control in THz-Enabled Drone Networks, IEEE/ACM Transactions on Networking 30 (2) (2022) 782–795.