Neural-Driven Control of RIS in 6G Networks: A GoSimRIS and xApp-Based Framework

Salim El Ghalbzouri, *Student Member, IEEE*, Karim Boutiba, *Member, IEEE*, Adlen Ksentini, *Senior Member, IEEE*, Mustapha Benjillali, *Senior Member, IEEE*[*†]

***Abstract*— In this letter, we propose an O-RAN-based framework for Reconfigurable Intelligent Surfaces (RIS) control in 6G. The key objective is to enable the development of RIS control algorithms as xApps running at the Real-time Intelligent Controller (RIC) of Open RAN (O-RAN). To validate the proposed framework, we developed a Golang-based RIS simulator, GoSimRIS, intended to mimic and examine RIS behavior in various environmental scenarios. The simulator is linked with the RIC via a specialized Service Model (SM) devised in this work, namely E2SM RIS, which allows the design of xApps that dynamically optimize RIS coefficients by computing the ideal phase shifts and applying them in real-time to maximize network performance using channel information that is retrieved from the GoSimRIS environment. Finally, we introduce an ML-based RIS control mechanism that runs as an xApp using only the positions of the transmitter (Tx) and receiver (Rx) and the presence of Line-of-Sight (LOS) conditions, which corresponds to a realistic indoor scenario in 6G such industry 4.0**

***Index Terms*—O-RAN, RIC, xApp, RIS, 6G Networks, ML.**

I. Introduction

The demand for extreme connectivity and ultra-low latency defines the shift toward 6G networks exploiting for the first time the Terahertz (THz) frequency spectrum ([1]). However, operating in these bands introduces significant challenges due to inherent propagation characteristics, such as high free-space path loss and extreme sensitivity to physical obstructions. Reconfigurable Intelligent Surfaces (RIS) have emerged as a transformative technology to address these issues by providing dynamic control over the propagation environment. By intelligently manipulating electromagnetic waves, RIS can enhance signal quality, mitigate interference, and optimize coverage, offering a powerful tool for next-generation wireless communication. However, to effectively implement RIS in real-world deployments, sophisticated control mechanisms capable of efficiently managing the complex interactions within a highly dynamic wireless environment are required. Although numerous research activities have been conducted to devise advanced control algorithms, whether based on optimization, control theory, or Machine Learning (ML) [2], most have been validated only theoretically. This highlights a gap in having a common and open framework to test and validate these algorithms using real RIS or high-fidelity simulated RIS.

In this work, we address these challenges by proposing an O-RAN-based framework that enables the development and deployment of RIS control algorithms as xApps running on the Real-time Intelligent Controller (RIC) of O-RAN. To validate the proposed framework, we developed a Golang-based RIS simulator, GoSimRIS, designed to mimic and analyze RIS behavior in various environmental scenarios. The simulator is connected to the RIC (in our implementation, we used OAI FlexRIC [3]) via a specialized Service Model (SM) devised in this work, namely E2SM RIS. This setup allows for the design of xApps that dynamically optimize RIS coefficients by calculating the ideal phase shifts and applying them in real-time to maximize network performance, using channel information (channel information between the nodes, RIS-UE-gNB) obtained from the GoSimRIS environment.

Acknowledging the shortcomings of traditional optimization techniques and the practical difficulty of achieving precise channel information in real-world situations, where the transmitter (Tx) and receiver (Rx) positions are usually the only information available, we have devised a novel xApp that relies on a ML model. This model, trained on a dataset generated by GoSimRIS, predicts optimal RIS coefficients based solely on Tx/Rx coordinates and the presence of Line-of-Sight (LOS) conditions. This approach provides a scalable and efficient solution to RIS control by eliminating the reliance on explicit channel measurements, making it particularly suited for scenarios where channel information is difficult or costly to obtain. It also holds the potential for future extrapolation into real-world applications

The contributions of this work are as follows: (i) an E2SM proxy that facilitates the integration of any RIS hardware into the O-RAN architecture, specifically allowing connection to the RIC; (ii) an E2SM RIS that conveys the essential information required by an xApp to control the RIS; (iii) GoSimRIS, a Golang-based high-fidelity RIS simulator that enables the simulation of RIS and Tx/Rx using the FR2 frequency band (millimeter waves), and generates the channel information necessary for an xApp to configure the RIS; (iv) an ML-based xApp that relies solely on the Tx/Rx positions and the presence of LOS conditions to compute the RIS coefficients.

II. RIS Control Framework: An O-RAN Oriented Approach

A. Architecture design

The O-RAN Alliance is an industry collaboration aimed at transforming the traditional RAN by promoting open, disaggregated, and software-driven architectures [4]. The key objectives of O-RAN include fostering vendor interoperability, increasing flexibility, and reducing deployment costs by decoupling hardware and software components. By establishing a unified

[*]S. El-Ghalbzouri, K. Boutiba and A. Ksentini are with EURECOM, France (e-mail: first.last@eurecom.fr).

[†]M. Benjillali is with IMT Atlantique, Lab-STICC, 29238 Brest, France. He is also with INPT, Rabat 10100, Morocco. (e-mail: benjillali@ieee.org).

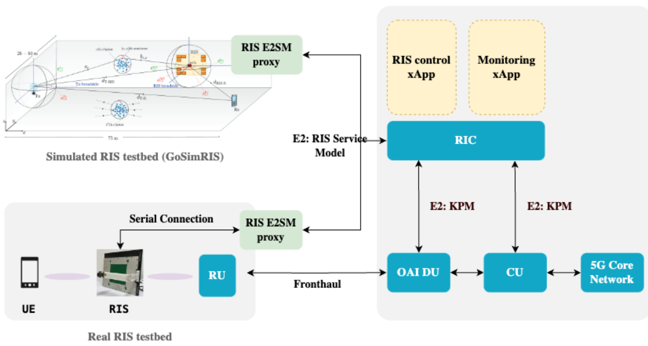

Figure 1: RIS Control Framework.

set of standards and interfaces, O-RAN enables network operators to integrate hardware and software from multiple vendors, thus reducing reliance on proprietary solutions and fostering innovation. O-RAN promotes the use of standardized, open interfaces, enabling more flexible and efficient deployment and management of RAN infrastructure. A key component of the O-RAN architecture is the RIC, which is designed to optimize the performance and flexibility of the RAN by enabling real-time, intelligent control over network functions. The RIC is divided into two layers: the near-real-time RIC and the non-real-time RIC. The near-real-time RIC operates with low latency (typically within 10 milliseconds) and manages dynamic functions such as handovers, interference management, and resource allocation, ensuring optimal performance and user experience. In contrast, the non-real-time RIC works at a higher latency, handling functions that require data aggregation and machine learning algorithms, such as long-term network optimization, traffic forecasting, and policy management. O-RAN defines the E2 interface, which allows the RIC to control the logic exposed by the E2 nodes (i.e., gNB, DU, and CU). The E2 interface is a logical interface composed of two protocols: (i) E2 Application Protocol (E2AP) and (ii) E2 Service Model (E2SM). The E2AP is a basic procedural protocol that coordinates how the near-real-time RIC and the E2 nodes communicate with each other and provides a basic set of services. E2AP messages can embed different E2SMs, which implement specific functionalities, such as the reporting of RAN metrics or the control of RAN parameters. Each E2 node exposes a number of RAN functions, i.e., the services or capabilities it supports. Last but not least, xApps are applications that run on top of the RIC, which acts as a relay between the xApps and the E2 nodes. Messages from xApps to E2 nodes are called E2 control messages, while those from E2 nodes to xApps are called E2 indication messages. These messages incorporate the SMs available at the RIC and the E2 nodes. Several SM models are standardized by O-RAN, including two key models: E2SM-KPM (Key Performance Measurement) and E2SM-RC (RAN Control). The E2SM-KPM model is used for monitoring and reporting KPIs, such as throughput, latency, and cell load, providing insights into network performance. Meanwhile, E2SM-RC allows the RIC to control specific RAN functions, including beamforming, scheduling, and handover management, to optimize resource utilization.

Figure 1 illustrates our approach to integrating RIS control within the O-RAN architecture. To achieve this objective, we devised a new component called the E2SM proxy, which serves as an interface between the RIS hardware and the RIC, onboarding the xApp that integrates the logic to control the RIS. The E2SM proxy, on one hand, uses the standardized E2 interface to communicate with the RIC, carrying the RIS SM messages used by the xApp to control the RIS. In this work, we propose a new SM model, namely E2SM RIS, dedicated to controlling the RIS. On the other hand, the E2SM proxy connects to the RIS via a dedicated interface to translate E2SM RIS messages into commands that follow the RIS hardware specifications, which are typically proprietary. The figure also depicts the proposed framework interfacing with both a real RIS and the simulated RIS. Although the proposed framework has been validated using GoSimRIS, it can interface with real RIS by adapting the E2SM proxy to the control logic of the hardware RIS.

B. *E2SM Proxy*

As indicated earlier, one of the key innovations of our work is the definition of the E2SM Proxy to integrate the RIS and its control into the O-RAN architecture. The E2SM RIS defines two types of information: (1) read information (or E2 indication messages in O-RAN terminology) and (2) write information (or E2 control messages in O-RAN terminology). Read information is twofold: (i) the RIS properties from each RIS E2 node, including the RIS dimensions (length and height), number of RIS elements, spacing between elements, number of RIS element states, position of the RIS, the supported frequency, and the list of available beams (angles). The xApp receives this information from each E2 node and forms a catalogue; (ii) UE's channel metrics: The channel between the RIS and the UE is a cascade channel (i.e., composed of a channel between the RIS and UE, a channel between the RIS and gNB, and a direct channel between the RIS and UE). The E2 indication message can contain one or many metrics. On the other hand, write information consists of the RIS configuration (i.e., the RIS elements' coefficients) sent from the xApp to the RIS's E2SM proxy. The configuration can have two abstraction levels: (i) High-level configuration: the angle of the beam in a 2D plane (an integer value) or two angles in a 3D plane (two integers); (ii) Low-level configuration: the phase distribution for all the RIS elements (list of bytes). Figure 2 illustrates the envisioned workflow for RIS control using the O-RAN architecture.

The RIS E2SM proxy reads the RIS configuration from a configuration file that contains the following RIS properties: RIS dimensions (length and height), number of RIS elements, spacing between elements, number of RIS element states, position of the RIS, supported frequency, and the list of available beams (angles). It then creates an E2 message, following the RIS SM description, and sends it to the RIC controller. The RIC controller forwards the message to all the xApps connected to the RIS. The RIS E2SM proxy listens for control messages from the RIC. Each time the proxy receives an E2 control message, it decodes the content using the RIS SM to obtain the RIS configuration. At this point, there are two options: if it is a low-level

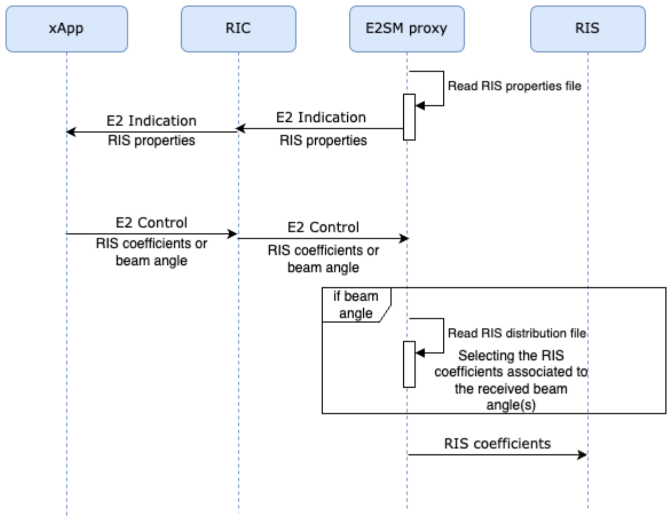

Figure 2: Systematic View of RIS Control Mechanism.

configuration, the proxy sends it directly to the RIS hardware, which is connected via a serial port to the E2SM proxy; if it is a high-level configuration, the proxy first derives the phase distribution of the RIS associated with the received angle(s). The phase distribution for all beams has been precomputed by the RIS provider and listed in a file. The proxy reads this file to locate the correct phase distribution associated with the angle(s). Finally, it sends the phase distribution to the RIS hardware via the serial port.

C. *GoSimRIS: A Golang-Based RIS Simulator*

To validate the envisioned O-RAN-based RIS control architecture, we developed a RIS simulator called GoSimRIS. This Golang-based simulation environment mirrors the functionality of the well-established SimRIS toolbox originally developed in MATLAB [5]. The decision to mimic SimRIS in Golang was driven by several critical factors: First, Golang was chosen for its superior concurrency model, which is essential for handling the parallelism required in large-scale wireless simulations. Golang's goroutines and channels provide a lightweight and efficient means of managing multiple simultaneous operations, a capability that is particularly advantageous in simulating dynamic environments with mobile user equipment (UEs) and varying channel conditions. Additionally, Golang's performance is close to that of lower-level languages like C, while offering the productivity benefits of higher-level languages, making it ideal for real-time systems. GoSimRIS extends SimRIS's complex indoor scenarios by incorporating advanced features that are critical for realistic RIS simulations in 6G networks. Notably, our contributions with GoSimRIS include developing a RIS simulator in GO language to overcome Matlab's limitations in real-time communication with remote systems such as xApp/FlexRIC using an E2 agent enclosed in the simulator. Furthermore, GoSimRIS allows for the simulation of mobility of the Tx and Rx, which was not possible with the original SimRIS. The simulator is highly configurable, leveraging a JSON-based configuration file to define various aspects of the simulation environment, such as the positions and orientations of the Tx, Rx, and RIS elements, antenna array configurations, frequency settings, and environmental dimensions. GoSimRIS generates the necessary channel matrices—H (Tx-to-RIS path), G (RIS-to-Rx path), and D (direct path)—which are essential for RIS coefficient optimization. By simulating these channels, GoSimRIS can provide critical data needed to compute the phase shifts that maximize the achievable rate at the receiver. These shifts are essential for computing the optimal coefficients (done at the xApp in our approach). This capability allows for comprehensive testing and validation of RIS control algorithms under various environmental conditions and network configurations. We leverage GoSimRIS with the E2SM Proxy to interact in real-time with the RIC, allowing the xApp to control the RIS by deriving optimal coefficients based on the simulated wireless environment. The integration of the RIS with the RIC facilitates the building of a closed-loop feedback system that continuously updates the RIS coefficients based on live network conditions. Besides validating our approach that adapts the O-RAN architecture to control the RIS, we believe GoSimRIS can help the community simulate and validate different RIS control algorithms

III. ML-BASED XAPP FOR RIS CONTROL

In this section, we will detail the xApp that utilizes our framework to run a ML algorithm for periodically computing the optimal RIS configuration based on user positions and LOS conditions. First, we will introduce our optimization algorithm, which computes the optimal RIS coefficients using the channel matrices H, G, and D. However, obtaining these values in real-life deployments is highly challenging. Therefore, we introduce an ML-based approach that uses only the positions of user equipment (UEs) to determine the optimal configuration. For this approach, we focus on an indoor deployment of mmWave gNB and RIS, where the room dimensions are known in advance, such as in smart factories and Industry 4.0 environments.

A. *Optimization Algorithms for RIS Coefficient Calculation*

The core functionality of the xApp lies in its ability to compute optimal RIS coefficients through advanced optimization algorithms. We envision a system model that considers a single-input, single-output (SISO) communication system enhanced by a RIS. We assume that the RIS is composed of N reconfigurable elements. The primary objective is to maximize the achievable rate R at the user equipment (UE) by optimizing the phase shifts θ_i of the RIS elements.

1) System Model: The received signal y at the UE can be expressed as:

$$y = \left(h_d + \mathbf{h}_r^T \boldsymbol{\Theta} \mathbf{g}\right) x + n, \tag{1}$$

where h_d represents the direct channel between the base station (BS) and the UE, $\mathbf{h}_r \in \mathbb{C}^N$ and $\mathbf{g} \in \mathbb{C}^N$ denote the channel vectors from the RIS to the UE and from the BS to the RIS, respectively. The matrix $\boldsymbol{\Theta} = \text{diag}(\theta_1, \theta_2, \ldots, \theta_N)$ is a diagonal matrix whose elements θ_i are the phase shifts introduced by the RIS elements. The term x is the transmitted signal from the BS, and n is the additive white Gaussian noise (AWGN) at the UE.

1) Optimization Objective: The goal of the optimization is to maximize the achievable rate R at the UE, given by:

$$R = \log_2 \left(1 + \frac{P_t |h_d + \mathbf{h}_r^T \mathbf{\Theta} \mathbf{g}|^2}{N_0} \right), \quad (2)$$

where P_t denotes the transmit power of the BS and N_0 represents the noise power. The optimization problem can be formulated as:

$$\max_{\mathbf{\Theta}} R \quad \text{subject to} \quad |\theta_i| = 1, \forall i \in \{1, 2, \ldots, N\}. \quad (3)$$

This non-convex optimization problem is challenging due to the unit-modulus constraint on each θ_i.

1) Optimization Algorithm: To address the optimization problem, we propose a hybrid approach that combines Simulated Annealing (SA) with Gradient Descent (GD). Simulated Annealing is utilized for the global exploration of the solution space, enabling the algorithm to avoid local minima. Gradient Descent is then applied for fine-tuning, allowing for precise adjustments to the phase shifts. The operation of this hybrid algorithm is detailed in Algorithm 1. This hybrid approach ensures a balance between exploration and exploitation, providing an effective solution for the optimization problem inherent in RIS configuration.

Algorithm 1 Simulated Annealing with Gradient Descent for RIS Optimization

```
Initialize θ randomly
Set initial temperature T
while T > T_final do
    for iteration = 1 to max_iterations do
        Compute gradient ∇R(θ)
        Propose new θ′ = θ + α∇R(θ)
        Perturb θ′ with noise N(0, T)
        Compute new achievable rate R(θ′)
        if R(θ′) > R(θ) or exp((R(θ′) − R(θ))/T) > rand() then
            θ ← θ′
        end if
    end for
    Decrease temperature T ← T · cooling_rate
end while
```

B. ML-based RIS Control

The ML-based approach for RIS coefficient prediction builds upon the foundation of the optimization-based control strategy discussed earlier. In controlled simulation environments, achieving high performance with optimization algorithms relies on the availability of detailed channel information (i.e., H, G and D), a luxury rarely feasible in practical deployments. In real-world scenarios, acquiring comprehensive channel information is not only costly and time-consuming but also often impractical due to rapid environmental changes and the high mobility expected in 6G networks. Our ML-based approach addresses this critical gap by leveraging spatial data—specifically, the positions of the transmitter and receiver—rather than relying on explicit channel information. This shift allows for a scalable and adaptable solution that can predict effective RIS configurations in real-time, even when only positional information is available. By developing this neural model, we provide a novel framework that extrapolates the insights gained from simulation-based optimization to real-world conditions where dynamic and unpredictable factors come into play. To construct a high-fidelity model capable of generalizing across diverse network conditions, particularly in an indoor environment, we generated an extensive dataset using GoSimRIS. The dataset spans a wide range of Tx and Rx positions, constrained within an x-axis of 75 meters and a y-axis of 50 meters. Each Tx and Rx position was incremented by 2.5 meters for each fixed point corresponding to the scene dimensions modeled in GoSimRIS. To capture the multifaceted nature of wireless environments, two LOS conditions—clear and obstructed—were simulated for each Tx/Rx pair. This deliberate variation ensures that the dataset reflects a wide range of realistic propagation conditions, providing a solid foundation for training the neural network. For each combination of Tx/Rx positions and LOS conditions, the optimized RIS coefficients were computed using the advanced optimization algorithms presented in Algorithm 1, representing the best achievable configuration for given scenarios, and were then meticulously included in the dataset. In total, the dataset contains nearly 850,000 samples, each entry correlating spatial positions, LOS conditions, and their corresponding optimized coefficients. The neural network architecture was carefully designed to balance depth and regularization, allowing it to model the complex non-linear relationships between input positions and optimal RIS coefficients. For the ML model, we adopted a deep neural networks (DNN) approach. After extensive hyperparameter tuning, the final architecture consists of six hidden layers with varying units and dropout rates, optimized for accurate and reliable predictions. The trained model was rigorously evaluated on a hold-out test set to quantify its predictive accuracy. The evaluation metrics—Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and R-squared (R^2)—are reported in Table 1. These metrics reflect the model's capability to generalize well beyond the training data, which is crucial for practical deployment scenarios.

Table 1: Model Performance Metrics on Test Set

Metric	MAE	RMSE	R-squared
Value	0.4448	0.8391	0.5803

IV. Performance Evaluation of Neural-Driven Configuration

A. Analysis of Achievable Rates for Dynamic LOS Conditions

Building on the dataset generation and rigorous model evaluation presented earlier, this section focuses on the practical application of the neural-driven configuration under dynamic network conditions. Traditional optimization methods are already validated in scenarios where full channel information is accessible, allowing this study to concentrate on the neural-driven approach, which aims to achieve superior performance over base-

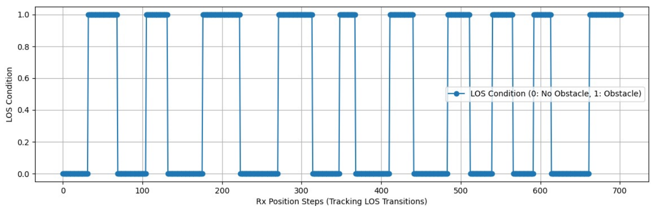

Figure 3: Dynamic LOS condition transitions over Rx position steps.

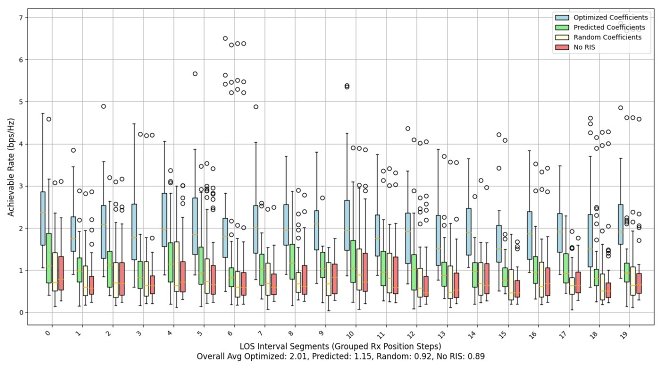

Figure 4: Achievable rate distribution per LOS interval (aggregating Rx position steps) for different configurations (see LOS intervals in Fig. 3).

line configurations even without explicit channel information. For this analysis, we extracted a subset of nearly 700 distinct Rx positions from the broader dataset of 850,000 samples, keeping the Tx position fixed. The selected positions provide a representative overview of changing line-of-sight (LOS) scenarios and spatial configurations, offering a focused validation of the neural-driven model's adaptability within this subset, while also demonstrating its scalability across the larger dataset. The subset used in this analysis encompasses a variety of LOS intervals, each representing continuous Rx positions where LOS conditions remain constant. Figure 3 illustrates these dynamic LOS transitions, simulating real-world scenarios where obstacles intermittently obstruct the communication path. In Figure 4, the achievable rates are averaged over each LOS interval (20 as illustrated in Figure 3), providing a comprehensive evaluation of the model's performance under both clear and obstructed conditions. The neural-driven configuration demonstrates a decisive performance advantage over baseline methods (Random and No RIS), achieving a 25% higher achievable rate than Random and a 29% improvement over No RIS. Additionally, it consistently maintains higher median achievable rates and exhibits a narrower interquartile range, indicating greater reliability across LOS intervals. By leveraging spatial and LOS data without explicit channel information, the neural model dynamically adjusts phase shifts, enhancing achievable rates across varying propagation environments. This adaptability underscores the robustness of the neural-driven approach in handling the complex dynamics of real-world 6G networks. From these results, we can conclude that the ability of the neural network to predict RIS coefficients using only Tx/Rx positions and LOS conditions represents a significant advancement. By eschewing the need for explicit channel information, the model offers a scalable solution that can be deployed in real-world scenarios where acquiring comprehensive channel information is either costly or infeasible. The generalization capability of the model is further underscored by its consistent performance across a broad range of Tx/Rx positions and LOS conditions. This robustness suggests that the model can adapt to various environmental contexts, making it a versatile tool for future 6G networks.

V. Conclusion

This paper presented a dual-component framework for the control of RIS in 6G networks. The first component, GoSimRIS, provided a scalable and efficient simulation environment for generating realistic channel data, which was then leveraged by an xApp within FlexRIC for real-time RIS control using advanced optimization algorithms. Using xApp for RIS control is a groundbreaking achievement, marking the first instance of such an integration, and lays the foundation for future research in RIS control. The second component introduced a deep learning model designed to predict RIS coefficients based on transmitter and receiver positions and LOS conditions. The evaluation demonstrated that the neural model delivered near-optimal performance with reduced complexity, highlighting its potential for scalable and adaptive RIS control. Future works will include testing with hardware prototypes and field experiments, which will be essential for assessing its adaptability and reliability in diverse scenarios and extend its applicability beyond controlled settings.

VI. Acknowledgement

This work is supported by the European Union's Horizon Program under 6G-Bricks project (Grant No. 101096954) and national ANR PEPR 5G NF-NAI project.

References

[1] A. Shafie, N. Yang, C. Han, J. M. Jornet, M. Juntti, and T. Kürner, "Terahertz Communications for 6G and Beyond Wireless Networks: Challenges, Key Advancements, and Opportunities," *IEEE Network*, vol. 37, May/June 2023.

[2] Y. Liu, X. Liu, X. Mu, T. Hou, J. Xu, M. Di Renzo, and N. Al-Dhahir, "Reconfigurable intelligent surfaces: Principles and opportunities," *IEEE Communications Surveys & Tutorials*, vol. 23, no. 3, 2021.

[3] R. Schmidt, M. Irazabal, and N. Nikaein, "FlexRIC: An SDK for next-generation SD-RANs," in *Proc. 17th International Conference on Emerging Networking EXperiments and Technologies (CoNEXT '21)*, 2021.

[4] K. Boutiba, M. Bagaa, and A. Ksentini, "Multi-agent deep reinforcement learning to enable dynamic TDD in a multi-cell environment," *IEEE Trans. Mob. Comput.*, vol. 23, no. 5, pp. 6163–6177, 2024.

[5] E. Basar and I. Yildirim, "SimRIS channel simulator for reconfigurable intelligent surface-empowered communication systems," in *Proc. IEEE Latin-American Conference on Communications (LATINCOM'20)*, pp. 1–6, 2020.