A GNU Radio Framework for Real-Time Beam Steering with Reconfigurable Intelligent Surfaces

Parker Wilmoth PWILMOTH2023@FAU.EDU
George Sklivanitis GSKLIVANITIS@FAU.EDU
Dimitris A. Pados DPADOS@FAU.EDU
Center for Connected Autonomy and AI (CA-AI.fau.edu), Florida Atlantic University, Boca Raton, FL, USA

Abstract

This work presents a software framework for sustaining robust mmWave communication links in non-line-of-sight (NLoS) scenarios using a dynamically reconfigurable intelligent surface (RIS) operating at 28 GHz. The framework integrates GNU Radio with a python-based control backend to coordinate Tx/Rx signal processing, RIS beamsteering, and mmWave phased array beam selection in real time. The focus of this work is on the testbed implementation – specifically, software integration of dynamic RIS control with software-defined control of mmWave transceivers using GNU Radio. The objective of this work is to develop and demonstrate the capabilities of open-source software for rapidly prototyping and testing dynamic beam-control and Tx/Rx signal processing algorithms using GNU Radio and TMYTEK's mmWave transceivers and RIS. Experimental results demonstrate one of the capabilities of the developed framework i.e., to select the optimal – in receive signal strength– RIS configuration for various testbed topology configurations in an indoor laboratory environment.

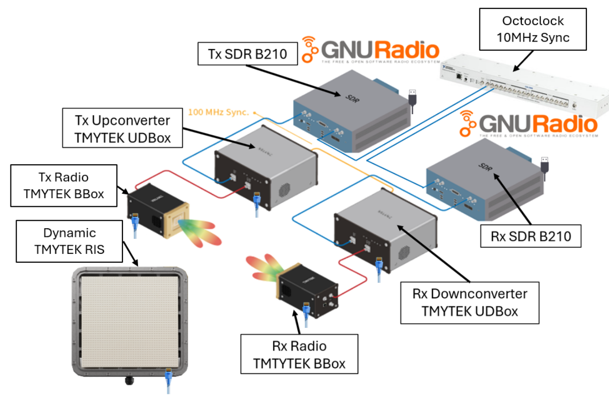

Figure 1. Testbed schematic diagram.

1. Introduction

As sub-6 GHz frequency bands become more congested and wireless data traffic continues to grow, millimeter wave (mmWave) is one of the key enablers for high-throughput and ultra low-latency wireless systems by allowing directive links over a large bandwidth. However, directionality has disadvantages in terms of tolerance to mobility and antenna size. mmWave faces significant propagation losses and can be blocked by obstacles, making it challenging to achieve optimal signal coverage, especially in a dense urban setting. How might a single NextG mmWave base station reach every user in such an environment? Reconfigurable intelligent surfaces (RIS) have emerged as an promising technology to enable users to customize EM signal distributions, cover dead zones, and create cold zones. Software-defined beamsteering jointly at each mmWave phased array transceiver and RIS may opportunistically solve mobility and line-of-sight (LoS) challenges of directionality. For example, when an obstacle hinders LoS between the transmitter and the receiver, a RIS device strategically deployed can alleviate this problem via suitably designing the re-configurable phase shift provided by each unit cell so as to effectively create a virtual LoS, which guarantees favorable signal propagation conditions.

In this work, we present a software framework that builds upon GNU Radio's control of sub-6 GHz software-defined radios (SDRs) and integrates to a python-based application programming interface for dynamic beamsteering control of TMYTEK's mmWave transceiver and RIS. Unlike prior sub-6 GHz RIS demonstrations or mmWave SDR testbed setups, our software framework integrates dynamic RIS control into the GNU Radio ecosystem of signal processing blocks to enable rapid prototyping and testing of adaptive beamsteering algorithms for applications in connected

Proceedings of the 15^{th} GNU Radio Conference, Copyright 2025 by the author(s).

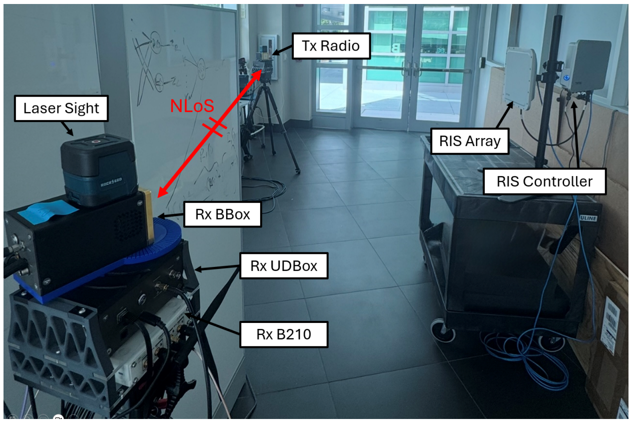

Figure 2. Testbed setup in the FAU CAAI lab.

robotics, urban mobility, to name a few. The testbed (as depicted in Fig. 1) comprises two USRP SDRs interfaced to TMYTEK's broadband frequency upconverters and FR2-compatible 28 GHz beamformers and a dynamic RIS with frequency range from 26 to 30 GHz. By leveraging the testbed's beamformer, frequency converter, SDR and RIS capabilities, our framework can unlock the potential of mmWave for applications such as 5G FR2, mmWave radar sensing, and beam management. In this work, we demonstrate the capabilities of the proposed software framework to rapidly test and evaluate mmWave beamformer and RIS configurations that maximize receive signal strength in various testbed topology configurations in an indoor laboratory environment.

2. Related Work

Prior research on RIS-assisted wireless communications and SDR-based testbeds can be broadly grouped into the following three categories.

2.1. Sub-6 GHz RIS Prototypes

A 5.8 GHz RIS with 1100 elements (Pei et al., 2021), demonstrates up to 27 dB power gain in outdoor trials and a 32 Mbps link with live video streaming. A 2-bit transmissive RIS at 27 GHz (Tang et al., 2022), achieves 22 dBi gain with beam steering up to $\pm 60°$. A vision-aided RIS (5.4 GHz, 20×20 array) (Ouyang et al., 2023) that dynamically adjusts reflection based on camera input, bypassing conventional feedback protocols. A 5.1 GHz multi-user SDR testbed is described in (Chen et al., 2025) where RIS-assisted co-channel interference mitigation was experimentally validated. Application of a reinforcement learning framework for blind beamforming improved the minimum signal-to-noise-interference-ratio (SINR) of two users by 13 dB compared to a system without RIS.

2.2. mmWave SDR Testbeds

The PAWR COSMOS (Chen et al., 2023) testbed incorporates programmable 28/60 GHz SDRs based on IBM's 28 GHz dual-polarized phased array module and Sivers IMA 60 GHz transceivers, enabling real-time channel sounding, beam tracking and dynamic beamforming in both indoor and outdoor settings. A portable 60 GHz SDR (Şahin et al., 2023) built on Xilinx RFSoC 2x2 and Sivers transceivers, introduces waveform-triggered reception and buffering to enable rapid beam sweeping across thousands of transmit–receive beam pairs, with open-source datasets for channel analysis. A low-cost mmWave SDR testbed that integrates USRPs, GNU Radio, and Python to enable programmable beam directionality is described in (Jean et al., 2023) and demonstrates proof-of-concept experiments such as angle-of-arrival detection with reinforcement learning. MiRa (Abari et al., 2016) is a USRP-based mmWave SDR platform with electronically steerable phased arrays at 24 GHz. MiRa supports dense modulation schemes up to 256-QAM and has been extended to multi-user MIMO operation, increasing throughput by 60% in office experiments.

2.3. Reconfigurable RF Front Ends with GNU Radio

An SDR system (Agasti et al., 2023) integrating GNU Radio flowgraphs to reconfigurable antennas enables dynamic joint configuration of antenna and baseband signal processing using a GNU Radio OOT project titled gr-recon. STAMINA (Santos et al., 2023) is a software-defined mmWave framework for customizable Initial Access (IA) procedures at 28 GHz. STAMINA exposes IA as a programmable control loop within GNU Radio, enabling experimenters to flexibly adjust beam sweep sequences, beam durations, and decision algorithms, and demonstrates measurable gains over static or manual beam selection.

3. Hardware Setup

We use two NI/Ettus USRP B210s as sub-6 GHz SDRs for the generation/reception of intermediate frequency (IF) signals, which are up-/down-converted by a TMYTEK UD-Box with a built-in local oscillator with 24-44 GHz control range. SDRs are interchangeable with most GNU Radio compatible FR1 and even FR3 SDRs, as the UDBox has a wide IF from 0.01 to 14 GHz capability. The UD-Box is connected to a BBox that comprises RF components such as transmit/receive (T/R) switch, power amplifier (PA), low noise amplifier (LNA) and phase shifters as well as TMYTEK's standard antenna for beamsteering from 26.5 to 29.5 GHz. We choose 28 GHz as the carrier center frequency for our experiments which is within the range of frequencies supported by the XRifle dynamic RIS that has a programmable center frequency range from 26 to 30 GHz, but also aligns with TMYTEK's fixed-beam RIS (called XRifle reflectors) that operate at 28 GHz. XRifle reflectors can therefore be used interchangeably with the dynamic RIS in our testbed. An Octoclock is used to provide a 10 MHz reference clock to the two USRP B210s to minimize timing and carrier frequency offsets. The UDBox up/down converters synchronize each other on a shared 100 MHz line. Both B210s are connected to the same host computer via USB 3.0. TMYTEK UDBox and BBox are connected to the same host computer via Gigabit ethernet. TMYTEK has a simple web-based GUI that allows the user to configure the settings of both the UDBox and BBox. Settings for TMYTEK's mmWave hardware can also be controlled via python API. For our experiments, the UDBox is configured to 28 GHz, 915 MHz IF with high side injection. The UDBox bandwidth should also be greater than or equal to that of the USRP. To measure the true inbound and outbound angles seen by the RIS from Tx and Rx, two different color laser sights are co-located with each BBox. The lasers shine a crosshair at the RIS. When the crosshair points directly at the center of the RIS, and a protractor is centered at the crosshair, the inbound and outbound angles are marked by the crosshair on the protractor. To measure the distance between the RIS and the Tx and Rx, we use a laser rangefinder tool. Figure 1 shows the testbed's data connections. Figure 2 depicts a snapshot of the physical setup of the hardware in the FAU CAAI lab.

4. Software Framework and GNU Radio Implementation

The proposed software framework is designed so that GNU Radio handles all IQ-level Tx and Rx functions, and separate python programs implement logic for beam steering control and additional signal processing. The software framework is comprised of three core programs which are described in detail below.

4.1. GNU Radio

GNU Radio is used to initialize the connections with the SDR hardware. At the Tx and Rx SDRs we use UHD drivers to interface with USRP sink and source blocks, respectively. Using SoapySDR -a vendor and platform neutral SDR support library- we could interface to other SDRs, too. Signal processing e.g., modulation, filtering etc. may take place in GNU Radio, or can be passed directly to another program. In this version of our software framework, samples from/to USRP source/sink are offloaded to a separate python controller using UDP source/sink blocks in GNU Radio. Figure 3 depicts the Tx and Rx GNU Radio flowgraphs. Figure 4 illustrates in a QT GUI time sink a snapshot of the over-the-air transmitted and received samples. The Tx repeatedly sends a Barker code sequence of length 13 followed by a Zadoff-chu (ZC) sequence with length $N = 499$ and root $r = 1$ for different Tx/Rx beamformer configurations. At the Rx, we use the known Barker sequence to identify the beginning of each transmitted sequence and calculate the received signal strength (RSS) by averaging the absolute squared values of the received IQ samples. In ongoing work with our software framework for testing mmWave radar and communications applications we use existing GNU Radio blocks for signal modulation, time and frequency synchronization, and channel equalization prior to moving bits/samples through UDP sockets for further processing to our python controller.

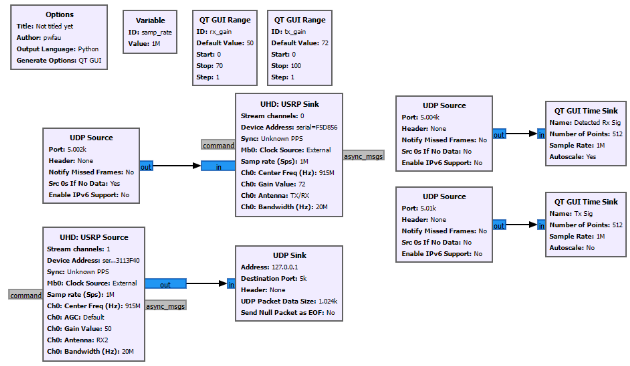

Figure 3. Transmit and receive GNU Radio flowgraphs.

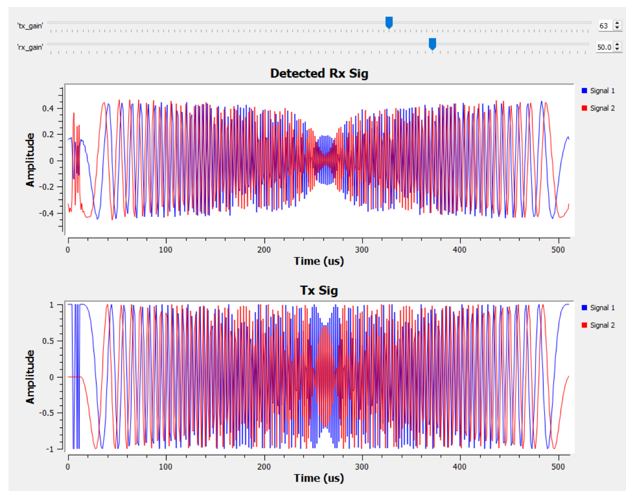

Figure 4. Snapshot of GNU Radio GUI during testbed operation.

4.2. Beamsteering Interface

To control the TMYTEK Tx/Rx and RIS, we use the python API provided by TMYTEK and build a custom software interface in python that creates an instance of the TMYTEK controller to communicate beamsteering instructions to the mmWave Tx/Rx beamformers and the RIS. Beamsteering instructions are exchanged via a UDP socket from the main python controller program (described in Section 4.3), similar to how IQ samples are exchanged between the python controller and GNU Radio. The python API is maintained up-to-date by TMYTEK. Latency due to our software interface running on a host PC and inter-process communication overhead, results in approximately 6 ms delay for beamsteering instructions to take effect. Beamsteering is a blocking function call i.e., every time the beam configuration is changed, the execution of the thread that controls Tx/Rx of IQ samples is halted until a Tx/Rx beam is set.

4.3. Python Controller

The python controller program communicates with GNU Radio and the beamsteering interface via UDP sockets. In this version of our software framework, we demonstrate Tx/Rx of Barker and ZC sequences, which are used to calculate RSS at the Rx while controlling the Tx/Rx beamformers and RIS. To enable coordinated control of the beamsteering interface and GNU Radio flowgraphs we choose to implement all signal processing in Python instead of GNU Radio. Next-generation versions of the software will move the controller and beamsteering interface in GNU Radio OOT blocks.

5. Testbed Experiments

We demonstrate the capabilities of the mmWave and dynamic RIS testbed by setting up the following experiment. We consider fixed Tx/Rx beamformer settings, RIS settings (i.e., angle-of-incidence, angle-of-reflection, distance to Tx, and distance to Rx) are our optimization variables and RSS measured at the Rx is our utility optimization function. In a NLoS scenario, RSS will be maximized when the RIS angle-of-incidence aligns with the direction of the Tx and the RIS angle-of-reflection aligns with the Rx. In a LoS scenario, the Tx beam should point directly to the Rx. The Rx should also point directly to the Tx. For an arbitrary testbed topology, our software framework implements Algorithm 1 to automate the calculation of the optimal RIS setting to maximize RSS.

To further simplify our setup, we choose to configure Tx/Rx and the RIS at the same elevation and therefore exclude elevation angle settings from our RIS optimization. TMYTEK BBoxes can steer in 81 unique directions. The steering functionality of the BBoxes by themselves has been demonstrated to work in previous experiments, so they are set to steer directly ahead ($0°$) for the experiments shown below. TMYTEK RIS can steer in 121 unique directions. The distances between RIS and BBox is much closer to analogue, as it allows for cm-level precision.

Algorithm 1 RIS Optimization

Initialize Tx and Rx beams at $0°$
Exclude elevation parameter (set to 0)
Initial search ranges:

$$\theta_{in}, \theta_{out} \in [-60°, 60°], \quad d_{in}, d_{out} \in [0, 3]\ m$$

Final target precision: $\Delta\theta_{\min} = 5°,\ \Delta d_{\min} = 0.2\ m$
Initialize coarse step sizes: $\Delta\theta = 20°,\ \Delta d = 1\ m$
while $\Delta\theta > \Delta\theta_{\min}$ **or** $\Delta d > \Delta d_{\min}$ **do**
  for $\theta_{in} = \theta_{in}^{rangestart}$ to $\theta_{in}^{rangeend}$ step $\Delta\theta$ **do**
    for $\theta_{out} = \theta_{out}^{rangestart}$ to $\theta_{out}^{rangeend}$ step $\Delta\theta$ **do**
      Measure RSSI for $(\theta_{in}, \theta_{out})$
    end for
  end for
  Select $(\theta_{in}^*, \theta_{out}^*)$ that maximize RSSI
  for $d_{in} = d_{in}^{rangestart}$ to $d_{in}^{rangeend}$ step Δd **do**
    for $d_{out} = d_{out}^{rangestart}$ to $d_{out}^{rangeend}$ step Δd **do**
      Measure RSSI for (d_{in}, d_{out})
    end for
  end for
  Select (d_{in}^*, d_{out}^*) that maximize RSSI
  Update search ranges to a neighborhood around $(\theta_{in}^*, \theta_{out}^*, d_{in}^*, d_{out}^*)$
  Refine step sizes: $\Delta\theta \leftarrow \Delta\theta/2$, $\Delta d \leftarrow \Delta d/2$
end while
Output:

RSS-optimal RIS configuration:$(\theta_{in}^*, \theta_{out}^*, d_{in}^*, d_{out}^*)$

Algorithm 1 describes our course-to-fine optimization algorithm to find the optimal RIS configuration (i.e., azimuth angle-of-incidence, angle-of-reflection, distance-to-Tx, distance-to-Rx). First, the angle-of-incidence and angle-of-reflection are scanned and the pair that maximizes RSS is selected. Then the distance-to-Tx and distance-to-Rx are scanned and the pair that maximizes RSS is selected. The whole process is repeated, each time with finer levels of precision (more steps between angles and distances) until the configuration with the best RSSI is chosen. Our algorithm is inspired by (Moghaddam & Moghaddam, 2025), but adapted for RIS hardware where element-level control is unavailable. To visualize the difference between RSSI at good and bad angles, an RSS heatmap is drawn in Fig. 5.

5.1. Experimental Results

We consider three testbed topology Tx/Rx/RIS configurations in the indoor lab environment of FAU CAAI. Figure 5 has three pairs of images, each illustrating the physical position of the equipment in the lab and the corresponding

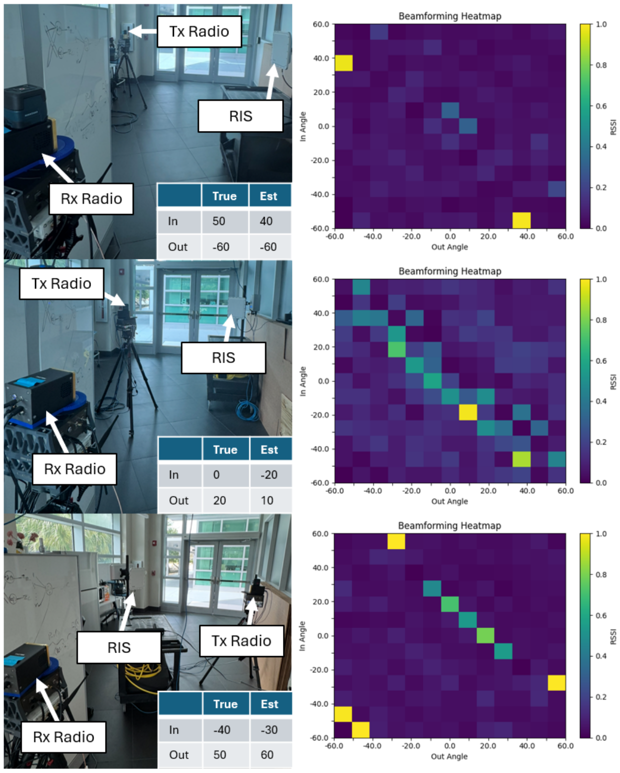

Figure 5. Experimental setups and RSS results for fixed beamsteering configurations of mmWave SDRs and adaptive beamsteering based on Algorithm 1 at the RIS.

RSS heatmap after implementing Algorithm 1. Included in each pair is also the true (physically measured) angle-of-incidence and angle-of-reflection at the RIS as well as the ones estimated by Algorithm 1 to be the ones that maximize RSS. The RSS heatmap produced by different angle configurations was selected over the one produced by different distance configuration as it demonstrates better the effect of misalignment between the Tx/Rx and the RIS. Some of the plots show similar strength RSS at opposite angles, as expected. in many cases the beam pattern generated by the RIS is the same whether the angle is positive or negative. Another important note is that measuring the true angle-of-incidence and angle-of-reflection is difficult to do with degree-level precision. Thus, the true angles are approximated to $\pm 5^\circ$.

From our experiments we observe that even when the LoS path was blocked, the system was able to choose the RIS configuration that maximizes RSS. The three topology configurations for the Tx/Rx and RIS have been selected such that we test the impact of drastically different locations for NLoS. As shown, there are large improvements in RSS when the selected angles by the RIS are close to the true angles (or possibly their negative counterpart). Independently of the physical setup of the hardware, the system is able to quickly control the RIS and find the optimal beam configuration. When the RIS was turned off the measured RSS was close to 0 (similar to when the RIS is misaligned with the Tx/Rx), thus, our experiments demonstrate the effectiveness of the RIS for NLoS operation. GNU Radio allows us to control SDR settings (such as software Tx/Rx gains) in real-time to adapt operation in any environment.

6. Challenges

The prevailing issue during the design, test and evaluation of the testbed was communication latency between software and hardware. It is necessary to have full-control of the beamsteering configurations during Tx/Rx of any signal. To address this problem, we have introduced artificial wait times between Tx/Rx of any signal and switching beamforming configurations. This one-out-one-in approach isolates signals from each other and allows each signal to be appropriately labeled with the associated beam configuration. However, this leads to a low transmit duty cycle, as most of the time is spent waiting for a transmitted signal to be received, detected and labeled appropriately. The Tx/Rx round-trip time of a signal (as seen by GNU Radio) using this setup is roughly between 50 and 100 ms. Considering $\approx$100 ms round-trip, the maximum beam scan rate is 10 Hz, which is an order of magnitude below the practical requirement for mobile mmWave links. The issue is becoming more pronounced when the number of beam-states that need to be tested increases. Another source of delay in the current system stems from the settle time of the hardware. Using the Gigabit ethernet interface the settle time is about 2 ms. In the next version of our software framework, we plan to address latency issues by transitioning physical layer functionalities to the FPGA and leveraging TMYTEK's BBox SPI control that satisfies the fast beam control requirements of 3GPP.

7. Conclusion and Future Work

In this work, we demonstrated a GNU Radio–based software framework for real-time control of a dynamic RIS and mmWave Tx/Rx beamformers. By integrating sub-6 GHz SDRs, mmWave Tx/Rx beamformers with a python-based backend for RIS beamsteering, we showed that the testbed can dynamically maintain high-quality links even under NLoS conditions. Experiments in an indoor laboratory confirmed that RIS beam configurations with maximum RSS can be identified and sustained across various testbed topology configurations.

Future work will focus on intelligent algorithms for joint Tx/Rx and RIS beamsteering to enable resilient mmWave connectivity in mobile scenarios. To significantly increase the effective bandwidth for joint communications and sensing applications, we will experiment with USRP X410 and RFSoC-based SDRs and explore how we could use RFNoC to transition physical layer processing to the FPGA of the SDR.

References

Abari, Omid, Hassanieh, Haitham, Rodreguiz, Michael, and Katabi, Dina. Poster: A millimeter wave software defined radio platform with phased arrays. In *Proceedings of the 22nd Annual International Conference on Mobile Computing and Networking*, MobiCom '16, pp. 419–420, New York, NY, USA, 2016. Association for Computing Machinery. ISBN 9781450342261. doi: 10.1145/2973750.2985258.

Agasti, Rosalind, Kim, Elliot, Ruyle, Jessica, Onwuchekwa, Chukwunodebem, Azizi, Shabnam, and Rahaim, Michael. Implementation of software-defined antenna and radio test system for congested spectral environments. In *Proceedings of the GNU Radio Conference*, volume 8, 2023.

Chen, Tianrui, You, Minglei, Zhang, Yangyishi, De Paolis, Fabrizio, Lerosey, Geoffroy, Nasser, Youssef, and Gradoni, Gabriele. Reconfigurable intelligent surface (ris)-assisted co-channel interference mitigation: Experimental validation with multi-user testbed. In *2025 IEEE 26th International Workshop on Signal Processing and Artificial Intelligence for Wireless Communications (SPAWC)*, pp. 1–5, 2025. doi: 10.1109/SPAWC66079.2025.11143447.

Chen, Tingjun, Maddala, Prasanthi, Skrimponis, Panagiotis, Kolodziejski, Jakub, Adhikari, Abhishek, Hu, Hang, Gao, Zhihui, Paidimarri, Arun, Valdes-Garcia, Alberto, Lee, Myung, Rangan, Sundeep, Zussman, Gil, and Seskar, Ivan. Open-access millimeter-wave software-defined radios in the pawr cosmos testbed: Design, deployment, and experimentation. *Computer Networks*, 234:109922, 2023. ISSN 1389-1286. doi: https://doi.org/10.1016/j.comnet.2023.109922.

Jean, Marc, Yuksel, Murat, and Gong, Xun. Millimeter-wave software-defined radio testbed with programmable directionality. In *IEEE INFOCOM 2023 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pp. 1–8, 2023. doi: 10.1109/INFOCOMWKSHPS57453.2023.10226092.

Moghaddam, Shahriar and Moghaddam, Kiaksar. A group-based coarse-fine algorithm for intelligent reflecting surface beamforming. *Physical Communication*, 71: 102668, 2025. ISSN 1874-4907.

Ouyang, Ming, Gao, Feifei, Wang, Yucong, Zhang, Shun, Li, Puchu, and Ren, Jian. Computer vision-aided reconfigurable intelligent surface-based beam tracking: Prototyping and experimental results. *IEEE Transactions on Wireless Communications*, 22(12):8681–8693, 2023. doi: 10.1109/TWC.2023.3264752.

Pei, Xilong, Yin, Haifan, Tan, Li, Cao, Lin, Li, Zhanpeng, Wang, Kai, Zhang, Kun, and Björnson, Emil. Ris-aided wireless communications: Prototyping, adaptive beamforming, and indoor/outdoor field trials. *IEEE Transactions on Communications*, 69(12):8627–8640, 2021. doi: 10.1109/TCOMM.2021.3116151.

Santos, Joao F., Fathalla, Efat, Da Silva, Aloizio Pereira, Da Silva, Luiz A., and Kibilda, Jacek. Stamina: Implementation and evaluation of software-defined millimeter wave initial access. In *ICC 2023 - IEEE International Conference on Communications*, pp. 346–352, 2023. doi: 10.1109/ICC45041.2023.10278868.

Tang, Junwen, Cui, Mingyao, Xu, Shenheng, Dai, Linglong, Yang, Fan, and Li, Maokun. Transmissive ris for 6g communications: Design, prototyping, and experimental demonstrations. 2022.

Şahin, Alphan, Sichitiu, Mihail L., and Güvenç, İsmail. A millimeter-wave software-defined radio for wireless experimentation. In *IEEE INFOCOM 2023 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pp. 1–6, 2023. doi: 10.1109/INFOCOMWKSHPS57453.2023.10226082.