

## **Project Report**



## CELTIC-NEXT AIMM Project

## WP6: Testbed and Demonstration Development

## D6.2

Authors: Milan Zivkovic, Frank Schaich (Nokia Bell Labs Stuttgart), Minglei You, Gan Zheng (Loughborough University), Yangyishi Zhang, Adrian Sharples, Fraser Burton (BT), Wael Boukley Hasan, Mark Beach (University of Bristol), Norbert Schmidt (IMST), Daniel Martini (IMST), Cliff Ellement (ThinkRF)

Project Acronym:	AIMM
Project Full Title:	Al-enabled Massive MIMO
Project Coordinator:	Arman Shojaeifard (InterDigital)
Project Duration:	24 months (Oct. 2020 - Sep. 2022)
Submission Date:	29 September 2022 (M24)
Dissemination Level:	External

#### Abstract

This final delivery report provides a summary of activities and achieved results during the 24 months of the CELTIC-NEXT AIMM project Work-Package 6 (WP6) on "Testbed and Demonstration Development". These include updates and results on the WP6 tasks, "Building a cell-less testbed with distributed antennas and signal processing, "Building centralised standard compliant testbeds", "Capturing real-time dataset information", "Real-time evaluation and demonstration" and "AI based Radio Security Testbed". Future work plans and dissemination activities are also highlighted in this report.

This document contains material, which is copyright of certain PARTICIPANTS and may not be reproduced or copied without permission. The information contained in this document is the proprietary confidential information of certain PARTICIPANTS and may not be disclosed except in accordance with the regulations agreed in the Project Consortium Agreement (PCA).

All PARTICIPANTS have agreed to full publication of this document.

The commercial use of any information contained in this document may require a license from the proprietor of that information.

Neither the PARTICIPANTS nor CELTIC-Plus warrant that the information contained in the report is capable of use, or that use of the information is free from risk and accept no liability for loss or damage suffered by any person using this information.

## **Executive Summary**

This final report describes the activities that have taken place to date within the CELTIC-NEXT AIMM project Work-Package 6 (WP6) on "Testbed and Demonstration Development".

The focus of this work package is the design and implementation of testbeds for proof-of-concept, evaluation, and demonstration of algorithms and technologies developed in the other WPs. This includes both centralised and cell-free architectures. The overall objective is to verify and demonstrate the practicality of the methods proposed in AIMM, as well as to capture and analyse performance data.

This final report provides information on the progress and obtained results made against all these topics of work within WP6 a. This included future work plans and performed dissemination activities.

Within Task 6.1 on "Building centralised standard compliant testbeds", the progress on the development of centralised standard compliant MIMO testbeds is reported by University of Bristol, including the integration of the external control server driving ML algorithms. However, due to staffing issues, the University of Bristol focused on the activities in WP5, while keeping the current status of the previous work conducted in WP6. Further, within task 6.1, the status of DPD-testbed is reported by IMST. This comprises the implementation of a reference testbed with simple device-under-test (DUT) models for memory and non-linearity evaluation of the DUT. IMST was granted an extension of the project duration. Thus, the outstanding work will be processed in the extension period.

Within Task 6.2 on "Building a cell-less testbed with distributed antennas and signal processing", University of Loughborough and BT proposed the architecture of distributed cell-less MIMO Testbed, addressing the signalling overhead challenge via the cluster design and investigating the signal processing complexity of the AI algorithm. They developed a testbed with full integration of ML modules showing the performance improvements. Moreover, within task 6.1, Nokia Bell Labs Stuttgart reported the results on the progress, integration efforts and performance of the ML-enhanced GPU-based gNB.

Within Task 6.3 on "Capturing real-time data-set information", All of the developed testbeds did data collection campaign to train the corresponding ML components. The data sets are offered for sharing and exchange.

Within Task 6.4 on "Real-time evaluation and demonstration", the reported statuses of testbeds included the design of mechanisms for scalable real-time demonstration of ML solutions developed in WP3, WP4 and WP5.

Within Task 6.5 on "AI based Radio Security Testbed", ThinkRF developed the architecture of wireless security testbed showing the large progress of building such system and conducting experiments. Furthermore, It is reported that the focus was on investigating different interference scenarios and creating both synthetic and real data sets for initial training of ML models.

Regarding dissimination activities, Nokia Bell Labs Stuttgart, as WP6 lead organized AIMM dedicated workshop on "Testbeds and Platforms for AI-enabled Massive MIMO" in EuCNC & 6G Summit 2022.

The workshop session was well attended, both in presence and remotely.

Overall, WP6 deliverables are provided in accordance with the project plan.

## **Table of Contents**

Executive Summary	3
Table of Contents	4
	5
1 Introduction	1
2 Technical Work Progress	8
2.1 Building centralised standard compliant testbeds	8
2.1.1 Massive MIMO Testbed	8
2.1.2 Digital pre-distortion (DPD) testbed	3
2.2 Building a cell-less testbed with distributed antennas and signal processing	0
2.2.1 Distributed Cell-less MIMO Testbed Architecture1	0
2.2.2 ML for L1 in GPU-enabled gNB (NBLS) Addressing Distributed Real-time Signal	
Processing via FPGA Development of Baseband Modules1	3
2.3 ML for L1 in GPU-enabled gNB2	24
2.3.1 The platform architecture and Data collection	25
2.3.2 Data collection	27
2.3.3 ML-model conversion for optimized inference on GPU	29
2.3.4 An interactive graphic visualisation	0
2.3.5 Conclusion and Outlook	2
2.4 AI Based Interference Detection Testbed	2
2.4.1 Overview of Interference in Wireless Networks	3
2.4.2 Autoencoder-based anomaly detection	3
2.4.3 Recent Activities	5
2.4.4 Conclusions and Next Steps	5
3 Conclusions and Future Work	6
References 3	57

## Abbreviations

Abbreviation	Definition
1D	1-Dimensional
3G	Third generation cellular
3GPP	Third Generation Project Partnership
4G LTE/LTE-A	Fourth generation cellular Long Term Evolution/Long Term Evolution
	Advanced
5G NR	Fifth generation cellular New Radio
A1	O-RAN interface between Non-RT RIC and Near-RT RIC
AAS	Active antenna system
AARX	Antenna array as receiver
A/D	Analog to digital
AI/ML	Artificial Intelligence/Machine Learning
BS	Base station
CAPEX	Capital expenditure
СоМР	Coordinated multipoint
CPRI	Common Public Radio Interface
CPU	Central processing unit
CQI	Channel Quality Indicator
CSI	Channel State Information
CS-RS	Cell-Specific Reference Signal
CU	Centralised unit
D/A	Digital to analog
DCI	Downlink Control Indicator
DMRS	Demodulation Reference Signal
DNN	Dense Neural Network
DPB	Dynamic Point Blanking
DPC	Dirty-paper-coding
DPD	Digital Pre-Distortion
DPS	Dynamic Point Selection
DSP	Digital signal processing
DU	Distributed unit
DUT	Device under Test
E2	O-RAN interface between Near-RT RIC and CUs/DUs
eCPRI	Enhanced Common Public Radio Interface
EM	Electromagnetic
eNB	eNodeB (4G LTE/LTE-A base station)
EVM	Error vector magnitude
F1	3GPP interface between CU and DU
FD MIMO	3D/full-dimension MIMO
FPGA	Free/Field programmable gate array
FR1	Frequency range 1
FR2	Frequency range 2
	FIXED WIFEIESS ACCESS
gNB OBU	gNodeB (5G NR base station)
GPU	Graphics processing unit
HBF	Holographic Beamforming
HLS	Higher-layer-split
	Intellectual property rights
	Intelligent Reliecting Surface
	Ney-periormance-indicator
L#    C	Layer humber # on the protocol Stack
	Lower-rayer-spill
	Linear Minimum Mean Square Error
	Least Mean Squares
	Line-ol-signi Medium Access Central
	Minimization of drive text
	Willing input multiple output
	iviuitipie-input multipie-output

mMTC	Massive Machine Type Communications
MORTY	Mobile transmitter
MDT	Maximum ratio transmission
	Multi transmission/reception points
	Neural Natural
NON-KI	Non-real-time
OPEX	
O-RAN	O-RAN Alliance
Open RAN	Ecosystem for open standardised interfaces implementation
PA	Power amplifier
PBCH	Physical Broadcast Channel
PDCP	Packet Data Convergence Protocol
PDSCH	Physical Downlink Shared Channel
PHY	Physical Layer
PRACH	Physical Random Access Procedure
PSS	Primary Synchronisation Signal
PUSCH	Physical Uplink Shared Channel
QoE	Quality-of-experience
QoS	Quality-of-service
RAN	Radio access network
rApp	An application designed to run on the Non-RT RIC
REFTX	Reference transmitter
RF	Radio frequency
RIC	O-RAN RAN Intelligent Controller
RIS	Reconfigurable Intelligent Surfaces
RIT	Radio Interface Technology
RLC	Radio Link Control
RLS	Recursive Least Squares
RRC	Radio Resource Control
RSRP	Reference Signal Received Power
RSRQ	Reference Signal Received Quality
RT	Real-time
RU	Radio unit
SA	Stand alone
SDR	Software defined radio
SE	Spectral efficiency
SNR	Signal-to-noise ratio
SINR	Signal-to-interference-plus-noise ratio
SLNR	signal-to-leakage-plus-noise ratio
SISO	Single-input single-output
SON	Self-organising-network
SRIT	Set of Radio Interface Technologies
SSB	System synchronisation block
SSS	Secondary synchronisation Signal
TXRU	Transceiver chain
UE	User equipment
VRAN	Virtualised RAN
¥2	3GPP interface between eNBs
xAnn	An application designed to run on the Near-RT RIC
Yn	3GPP interface between aNRs
7F	Zero-forcing
<b>4</b> 1	

## 1 Introduction

MIMO is a key air-interface technology in nearly all modern communications systems [1]. MIMO, through utilization of multiple antennas at the radios, can provide several benefits including enhancing spectral efficiency and quality-of-service [2]. Despite significant performance improvements achieved through MIMO to date, there exists a significant gap between the theoretical versus practical performance of multi-antenna systems [3].

Motivated by the above, the AIMM project targets radical performance improvements and efficiency dividends for 5G and beyond MIMO systems through adoption of AI/ML capabilities in both link-level and system-level RAN domains, as well as alternative deployment methods including radio intelligent surfaces and cell-less antenna systems. To achieve the set targets, the AIMM project work is divided between six tightly coupled work-packages, as illustrated in Figure 1 below.



Figure 1: AIMM project work-package structure.

WP6 will design and build four testbeds for proof-of- concept, evaluation, and demonstration of algorithms and technologies developed in the other WPs. This includes both centralised and cell-free architectures. The overall objective is to verify and demonstrate the practicality of the methods proposed in AIMM, as well as to capture and analyse performance data.

The developed testbeds will be leveraged to demonstrate and verify the practicality of the concepts of the AIMM project. There will be a total of five testbeds which focus on different use-cases (co-located versus distributed for antenna locations, centralized versus cell-less for operation) and technologies (real-time implementation using FPGAs versus non-real-time processing using CPUs and GPUs).

## 2 Technical Work Progress

This section provides progress reports by WP6 participants around the technical work that has been carried out to date within this work-package.

#### 2.1 Building centralised standard compliant testbeds

This task will focus on extending the existing massive MIMO SDR testbed at the University of Bristol into an AIMM SDR testbed. This will be achieved by connecting the massive MIMO SDR testbed to an external machine with AI capabilities. The algorithms proposed in WP4 and WP5 will be used in the external machine. Real-time information will be transferred from the massive MIMO SDR testbed to the external AI machine. The external AI machine will send commands to the massive MIMO SDR testbed based on the AIMM algorithms designed in WP4 and WP5.

The task will further include the DPD testbed to show the performance of the massive MIMO-based DPD as evaluated in WP3. Based on the specifications of WP3, an FPGA board will be selected including the necessary circuitry to connect it to a set of commercially available power amplifiers, which will be purchased as well as the FPGA board.

#### 2.1.1 Massive MIMO Testbed

The Massive MIMO testbed, depicted in Figure 2, comprises of a BS and up to 12 users. The BS is divided into 4 racks, providing 32 RF ends each, i.e. 128 in total. The testbed was built by the University of Bristol using the NI commercial off-the-shelf (COTS) products which are based on the PCIe platform known as PCIe eXtensions for instrumentation (PXIe). The RF ends are connected to a patch panel antenna array in a 4x32 configuration with vertical and horizontal polarisations operating at 3.51 GHz. The BS can serve simultaneously: a) up to 12 users with single antenna from 6 USRPs, or b) up to 6 users with two antennas. The system was designed and build to align closely with the TDD LTE air interface with a scalable sub-6 GHz carrier frequency. Table 1 shows the key specifications of the massive MIMO testbed. The 128 antennas are connected to 64 dual-channel USRPs divided equally into 4 racks as shown in Figure 2. These USRPs provide the RF front ends and perform the OFDM modulation/ demodulation of each complex subcarrier. The 16 USRPs in each rack linked via x4 PXIe links to a PXIe-1085 chassis with 18 slots acting as a switch. The PXIe-1085 chassis from each rack is then connected via a x8 PXIe link to the main chassis allocated on rack A (first rack from the left in Figure 2). The main chassis has 4 FlexRIOS 7976R co-processors performing channel estimation and MIMO detection in UL and reciprocity calibration and precoding in DL.



Figure 2: The UoB Massive MIMO Testbed: (left) Base station end, (right) users end.

Table 1: Specifications and Features for the UoB massive MIMO Testbed.			
Number of Antennas at BS	128		
Bandwidth	20MHz		
MCS	256-QAM, 64-QAM, 16-QAM, QPSK		
Duplexing Scheme	TDD		
MIMO Linear Decoder/Precoder	MMSE, ZF and MF		

#### 2.1.1.1 **Extending the UoB Massive MIMO Testbed**

The University of Bristol massive MIMO testbed was connected to an external machine controller as shown in Figure and Figure . This will enable extending the existing massive MIMO SDR testbed into AIMM SDR testbed. Data transfer connection was established between the massive MIMO testbed and external machine controller by using Ethernet connection. The massive MIMO testbed uses LabVIEW as a programming language. While the majority of ML and AI algorithms use Python as a programming language. Therefore, a LabVIEW to/from Python interface was created between the massive MIMO testbed and the external machine controller. This provides the external machine to control several of the massive MIMO testbed functionalities. Real time data can be transferred from the massive MIMO testbed to the external machine through an Ethernet cable. While the external machine can send commands to the massive MIMO testbed based on the results of ML&AI algorithms. This will allow flexible coordination between WP4, WP5 and WP6. It will also provide WP4 and WP5 with the flexibility to modify their algorithms without impacting the work progress on WP6. Some AIMM algorithms that require a very low latency will be implemented on the FPGAs of massive MIMO testbed.



Figure 3: The UoB Massive MIMO testbed connected to external machine.





#### 2.1.1.2 Enabling controllable TDD switching for external PA/LNA at the UE side

The universal software radio peripherals (USRPs) used at the UE side were modified to use external power amplifiers (PAs) and low noise amplifiers (LNAs). PAs and LNAs will be used in uplink and downlink respectively. By using the external PAs/LNAs, an Error Vector Magnitude (EVM) less than 0.5% can be achieved at 20 dBm. This will increase the channel state information (CSI) accuracy obtained by the testbed. Therefore, the performance of the massive MIMO testbed will be improved, and more accurate data can be captured to be used in AI/ML algorithms. In addition to increasing the CSI accuracy, the distance between user equipment (UE) and base station (BS) can be increased as well.

At the USRP side, TX1/RX1 port in RF0 and TX1/RX1 port in RF1 are connected to two external PAs/LNAs, as shown in Figure 5. The general purpose input output (GPIO) of the USRP was used to control the time division duplex (TDD) switching between uplink and downlink of the external PAs/LNAs.



Figure 5: USRP interfacing with external PA/LNA.

#### 2.1.1.3 Modifying data architecture

A rugged implementation of the peripheral component interconnect express (PCIe) (Gen 2) platform developed by NI for test and measurement applications known as PCIe eXtensions for instrumentation (PXIe) provides the interconnect fabric for the system. 64 dual-channel USRPs were required to create a 128 antenna implementation, requiring 4 PXIe-1085 chassis with 18 slots each. Using 12 bits for the in-phase and quadrature component of each complex subcarrier sample with OFDM modulation/demodulation performed on the USRP FPGA, and the bidirectional rate per USRP to the central processor is 100.8 MB/s. This results in a bidirectional rate of just over 800 MB/s per 1085 chassis switch and a 1.6 GB/s inter-chassis rate. Each chassis is linked back via x8 PXIe links to a fifth chassis where centralised 128×12 MIMO processing is performed across 4 FlexRIO 7976R co-processors split by bandwidth (300 subcarriers each). 3 co-processors would have been enough to accommodate the 6.5 GB/s bidirectional rate, but a fourth was added to reduce the low-latency design constraints. The Kintex-7 410T on-board each co-processor is also reportedly capable of 2.845 GMACS/s which satisfies the processing requirements. An overview of the chassis structure and the PCIe links is shown in Figure 6 and Figure 7. For distributed massive MIMO, four MXI-Express Cable, Gen 2 x8, Fiber Optic (Gen 2 x4 Speed) are used. The MXI-Express cable length is 100 m. These cables are used to connect the main chassis (centralised MIMO processing) with each of the four chassis.



Figure 6: 128-antenna massive MIMO testbed (BS side).



#### 2.1.1.4 Timing and synchronization

Whilst absolute phase calibration is not required for massive MIMO operation, all the USRP RF chains must be coherent with one another. To achieve this, a clock distribution network was implemented using nine pieces of Octoclock hardware paired with the NI 6674T PXIe timing card located in the central (master) chassis. The 6674T has an extremely stable 10MHz frequency source provided by an oven-controlled crystal oscillator (OCXO) that maintains an accuracy of <5 parts-per-billion (ppb). As each Octoclock module can amplify a 10MHz source and divide it between eight outputs, nine were required to distribute the 6674T OCXO source to all 64 USRPs as illustrated in Figure 8.

In addition to phase coherence, the digital clocking had to be appropriately triggered to ensure sample acquisition and generation was synchronised at each USRP. Conveniently, each Octoclock can also amplify a digital trigger input and divide it between a further eight outputs, allowing the same 9 modules to handle both frequency and sample synchronisation. One USRP sends a start trigger pulse to the 6674T card where it is reconditioned and routed to the first Octoclock module. The pulse is then fanned out by the Octoclock network through equal length cables as shown in Figure 9 to all of the USRPs, including the one that originally sent the start trigger. For distributed massive MIMO, 12 RG58 with 100 meter length cables are used for 10 MHz & PPS signals.



Figure 8: Distribution of frequency and trigger sources to all USRPs.

#### 2.1.1.5 Modifying antenna array

The University of Bristol massive MIMO testbed has a patch antenna array configured with alternate H & V polarisations for all 128 antennas. In order to distribute the massive MIMO testbed, the antenna array was split to four smaller patch antenna arrays (4×8), Each of the 4×8 antenna array is mounted at the top of one rack. The patch antenna array for distributed massive MIMO is shown in Figure 9, Figure 10, and Figure 11.



Figure 9: 128 distributed massive MIMO testbed (BS side).



Figure 10: 4×8 antenna array connected to 16 USRPs (backside).



Figure 11:  $4 \times 8$  antenna array (frontside).

#### 2.1.2 Digital pre-distortion (DPD) testbed

#### 2.1.2.1 Objective

Within this task, a testbed for DPD operation will be implemented. The DPD testbed will be used for two basic purposes:

- Generation of stimulus and training data for the algorithms to be developed in WP3;
- Implementation of the algorithms into real hardware to show the feasibility of real-time operation and to evaluate some performance metrics.

The complete development cycle of the algorithms within WP3 will be supported by the testbed of this WP6. Therefore, the first action will be the implementation of a reference testbed with simple DUT models for memory and non-linearity of the DUT.

The development starts with the classic signal processing solutions commonly known in the area of DPD implementations. For this testbed, the core system functionality is delivered by an RFSoC evaluation board from Xilinx together with an RF breakout board. This testbed is used for reference measurements and, later, for the characterization of the real power amplifier. The reference testbed has been completed and is under investigation now by WP3.

To incorporate the AI algorithms into implementation, there is the option for a second testbed based on Xilinx Versal technology. However, this testbed won't have the capability to drive RF inputs/outputs directly, so that the measurements still will be made on the RFSoC board and the Versal board will only use the measurement results for AI computation. It was not possible to purchase a Versal Board during project duration. Therefore we switched to an GPU Based approach for implementing AI-algorithms.

#### 2.1.2.2 Architecture & Implementation (reference testbed)

The core of the reference testbed is an RFSoC evaluation board as shown in Figure 12.

The main board consists of the core component FPGA from Xilinx [1]. together with peripheral devices interfaces. The interface to the user is ethernet. As the RFSoc FPGA has A/D and D/A conversion on board, the only other interfaces are two RF interfaces from the D/A converter and the A/D converter on the chip to transmit and receive RF signals at 3.6 GHz with a bandwidth of ~500MHz.

The RF chain consists of modular components which can be assembled in several different configuration by plugging them sequentially together. The following components have been provided for the reference chain:

- Wide band bandpass filter for image rejection;
- Narrow band bandpass filter for rejection of spurious components and image rejection;
- Balun component to change between symmetric/asymmetric RF signal flow;
- Pre-amplifier to compensate for the several attenuations caused by the other passive components;
- Limiter board to serve as a memoryless non-linearity for the reference chain.



Figure 12: RFSoC evaluation board with RF breakout board.

In Figure 13, Figure , and 15, all these components are shown.

The complete amplifier chain is a sequence of the following components:

Wide band bandpass->Balun1->PreAmp1->Narrow band bandpass->Balun2->PreAmp2->Limiter

The reference chain has to represent a moderate memory, which means a slight dip or ripple on the signal in the frequency domain. Additionally, at highest levels, nonlinear behaviour has to be seen.







Figure 14: Pre-amplifier and balun board.



Figure 15: Limiter board.

#### 2.1.2.3 Measurement results (reference chain)

First measurements have been conducted to characterize this reference chain and the wanted behaviour. All measurements have been made at a center frequency of 3.6 GHz and a bandwidth of 500 MHz.

The following figures show a test case with a 200 MHz band-limited white noise stimulus. Figure shows the transmit signal in the digital domain before it is sent to the D/A conversion. Figure 17 shows the response for a low signal level and a high signal level. It can be seen that the inband frequency response exhibits a slight ripple for all levels, which the first precondition for a valid reference chain. The achievable SNR is around 40—45 dB, which is sufficient for a valid analysis, to be seen at the figure with low signal level. At higher levels, out-of-band components of the signal appear due to the nonlinearity of the limiter. This can be seen in the right figure. The level of the distortion is somewhat higher than the noise level, which enables a good analysis.



Figure 16: 200 MHz TX spectrum.



Figure 17: 200 MHz RX spectrum at low and high signal levels.

Figure shows the signal in the time domain. The absolute values of the signal show a good match between TX and RX signal. Due to the group delay, which is unknown, the real and imaginary parts of the signal do not show a match because they have to be phase-rotated first.

The amplitude distribution shows the desired gaussian distribution for the real and imaginary part, and the resulting Rayleigh distribution for the absolute value without any clipping or distortion.



Figure 18: 200 MHz RX time domain and amplitude distribution.

The next figures show the two tone behaviour of the signal chain with two sine tones at 10 MHz and 15 MHz. Figure 19 shows the transmit signal in the digital domain before it is sent to the D/A conversion.



Figure 19: Two tone TX spectrum.

Figure 20 shows a signal with low level on the left side and a signal with high level on the right side. It can be clearly noted that a couple on nonlinear components are rising caused by the limiter in the signal chain.

At higher levels, the nonlinearity can be observed in the time domain, too. Figure shows the receive signal in time domain for large signal levels. The distortion of the received signal (blue) at higher levels can be seen with a significant damping compared to the original amplitudes.



Figure 20. Two tone RX spectrum.

abs, real, and imag



Figure 21: Two tone RX time domain analysis.

At a first glance, the conclusion out of these initial measurements is that the reference chain is feasible for the reference stimulus generation for the DPD algorithms of WP3. More analysis will be made during the further execution of the project.

#### 2.1.2.4 Architecture & Implementation (Power amplifier signal chain)

As a second step, a power amplifier has been procured and a power supply has been generated.



Figure 22: NXP 5W power amplifier, stand-alone and in signal chain.

The principle characteristics of the power amplifier have been measured. The amplifier shows a behaviour similar to the passive reference chain which has been previously used for measurements. With the potentiometers on the power supply board, different working characteristics can be configured and so behaviour of several slightly different power amplifiers can be emulated even with only one board.



Figure 23: Comparison of limiter and power amplifier gain curves parameterized with frequency.

In addition to pure noise stimulus, LTE test signals have been used to test the nonlinearity of the power amplifier as well. The test has been made at different modulation schemes and different levels. The metric which has been used is the EVM (error vector magnitude) of the LTE signal. Therefore, a complete LTE receiver has been implemented which is able to compensate for the influence of the measurement chain (latency, frequency synchronisation, time synchronisation, sub carrier amplitude and phase equalisation) and to measure the EVM of the received signal.

Levels have been adjusted at the output of the D/A converter for simplicity reasons (no need to change the hardware setup for each level). For this reason, low levels are noise-limited regarding the performance of the EVM. At higher levels, the EVM is dominated by the nonlinearity of the power amplifier.

The power amplifier testbed is now used continuously to take measurements within the WP3.

The following figures show the raw modulation constellations at the receiver for the pilot symbols and the data symbols for three different power levels of the power amplifier, at a bandwidth of 200MHz and 256QAM.



Figure 3: Pilot and data symbols, relative level=0.5 (Pout=1.25W).



Figure 25: Pilot and data symbols, relative level=0.8 (Pout=3.2W).



Figure 26: Pilot and data symbols, relative level=1.0 (Pout=5W).

The following figure shows the corresponding EVM values for 10 different LTE measurements, at 10 different relative levels. The frequency response of the signal chain (filter, pre-amplifier) has been compensated for these measurements.



Figure 27: EVM values for data symbols vs. relative level.

The EVM values at the lower levels are dominated by SNR due to the low signal levels at the input of the whole signal chain. For 256QAM modulation, the standard allows an EVM value of -26,94dBm. From level=0.5 on, the EVM is dominated by the power amplifier nonlinearity and the power levels violate the EVM requirements of the LTE standard, thus need a compensation by pre distortion. This compensation will be implemented within WP3.

#### 2.1.2.5 Further work:

Further work will be dedicated to implement the DPD pre distortion algorithms as well as AI based predistortion on the target instead of simulation. As it appears that a Xilinx Versal board cannot be delivered in time due to a shortage in Xilinx components (delivery time late in April 2022), the plan must be changed to implement the AI algorithms on a PC/Graphics card combination instead of using the Versal board. For this reason, an additional interface from the RFSoC board to the PC and back to the board must be implemented into the signal chain. IMST was granted an extension of the project duration. Thus, the outstanding work will be processed in the extension period.

# 2.2 Building a cell-less testbed with distributed antennas and signal processing

Pushing the AI algorithms to the network edges is one of the key concepts in further reducing communication latency, as well as improving the system throughput. The distributed AI algorithms for radio resource optimisation in WP4 and network operation and management in WP5 demand new testbeds for evaluations of the distributed antennas and signal processing, the focus of this task. Both real-time baseband signal processing and AI algorithms require high computing resources, therefore in this task solutions based on both CPU/GPU and FPGA will be explored.

#### 2.2.1 Distributed Cell-less MIMO Testbed Architecture

In the proposed scalable and distributed cell-less MIMO testbed, the APs and antennas will be distributed in the network, and the signal processing processes (e.g., channel estimation and beamforming) will be distributed at local or regional processing units, instead of all at one central processing unit.

With the distributed cell-less MIMO testbed, it is expected to provide a more uniform service across the network coverage, and improved end performance (e.g., throughput) via the coherent transmission among APs. To design and implement a practical cell-less testbed, it requires to address two key scalability challenges below [2]:

- 1. Signalling overhead challenge mainly due to CSI acquisition;
- 2. Signal processing complexity challenge.

In the following, the two challenges are detailed along with the solutions in the proposed testbed, addressing signalling overhead via the cluster design and addressing the signal processing complexity via the Al algorithm.

#### 2.2.1.1 Addressing signalling overhead challenge via the cluster design

Suppose the UL and DL are separated in a time-division manner. The UL channels are estimated via UL pilots, and the DL channels are estimated via UL channels by following the UL-DL reciprocity or AI algorithm. To achieve the coherent DL transmission among multiple APs, it requires the exchange of received CSI/pilots for UL and the beamforming vectors for DL. This exchange of CSI/pilots and beamforming vectors will cause significant signalling overheads.

The proposed testbed addresses this challenge by dividing multiple APs into 'clusters' as shown in Figure below. Within each cluster, one AP/processing unit is selected as the Master AP, which coordinates the CSI/pilots in UL and beamforming optimization in DL, as well as the involved signal processing. All UEs in each cluster are jointly served by all APs in that cluster. In this way, the signalling overhead regarding the beamforming in DL is limited by the number of APs and antennas in the cluster, which are determined and independent from the number of UEs. In practical deployment, the cluster size needs to be carefully chosen to balance the performance gain, interference and overhead.

There is obvious inter-cluster interference at the cluster edge. Specifically, the inter-cluster interference is addressed by introducing the 'border APs', which are the APs in the neighbour clusters and with strong interference power to the UEs in the focused cluster. To optimize the DL beamforming in each cluster, it only needs to collect: a). the CSI/pilots from all UEs to all APs in the cluster, and b). the CSI/pilots from border APs in neighbour clusters to all UEs in the cluster. By considering a limited number of border APs, the inter-cluster interference considers the strong interfering APs, and this limits the signalling overhead regarding CSI/pilots to the number UEs in the cluster and the number of APs in the cluster and border APs.

In the data plane, the UL user data are decoded at the Master AP, and then forwarded to the central data server. We assume that the DL user data are cached from the central data server to the Master AP, which are then sent from the Master AP to the APs through fronthaul links. We highlight that data plane is not considered in the current testbed design.



border APs

Figure 28: Distributed Cell-less MIMO Architecture.

#### 2.2.1.2 Addressing the signal processing complexity via the AI algorithm

The considered complexity challenge concerns the channel estimation and DL beamforming optimization algorithm, which needs to meet the time/computational requirement given a cluster. This challenge is addressed via the AI algorithm, which is trained in an offline manner with collected data and then implemented at the Master AP in each cluster. In this way, the signal processing functions can be realised in near real-time.



Figure 29: Distributed multicell massive MIMO.

We carried out a preliminary study of a multi-cell scenario shown in Figure in which users are associated with the AP and can be thought as a special case of our proposed architecture with only one AP in each cluster, and one cluster is equivalent to one cell.

In our proposed solution, the SLNR beamforming structure [3] is used in the training process, which considers the interference from the considered cluster to the users in other clusters, instead of the interference from the APs from other clusters to the considered user in this cluster as in SINR. The advantage is that it will remove the need to obtain the interference channels from neighbouring cells.

Based on the SLNR formulation, the required inputs of the trained neural networks NNs consist of the uplink CSI or pilots within each cluster, whose outputs are the beamforming vectors for the APs within that cluster. The AI algorithm will also include calibration/conversion of downlink CSI from uplink CSI if necessary. This will greatly simplify the channel estimation design. In this way, the proposed channel estimation and SLNR beamforming method can be fully distributed, with respect to both learning and implementing of the trained NNs.

The performance of the proposed SLNR beamforming solution is compared against the state-of-the-art solutions, including the WMMSE (the weighted minimum mean square error) solution, the solution that first learns the downlink channel and then applies zero-forcing beamforming (Learned Channel and ZF Beamforming solution) as illustrated in Figure. Results show that the proposed method is capable to achieve the sum rate performance close to the state-of-the-art solution based on WMMSE (centralised iterative solution). By extending this study with multiple APs in future works, the results are possible to be applicable for the distributed cell-less testbed.



Figure 30: Sum rate performance of multicell massive MIMO.

Settings in Figure:

- 1. FDD, 7 cells, UL 2.5GHz, DL 2.4GHz, total power = 10 dBm
- 2. ULA, small scale channel attenuations have square relation between uplink and downlink
- 3. Different coefficients in path loss.

#### 2.2.2 ML for L1 in GPU-enabled gNB (NBLS) Addressing Distributed Real-time Signal Processing via FPGA Development of Baseband Modules

In this task, the testbed is aiming at the distributed architecture, where antennas are distributed across the service coverage area. When the antennas are physically and spatially distributed in the network, the real-time data exchange with regard to the baseband signals, channel state information (CSI), and beamforming decisions will be a challenge. This is because unlike the case where massive antennas are collocated with the central processing units (CPUs), the fronthaul links between the remote antennas and CPUs could introduce not only latencies but also bandwidth concerns when the system scales up. Therefore, in this task, solutions to take advantage of the embedded FPGA computing resources will be explored to address the real-time signal processing challenges, where the FPGA based baseband modules are developed to estimate CSI at each antenna in a distributed manner. The bandwidth requirement for the fronthaul network will be much reduced by exchanging the estimated CSI instead of the raw baseband signals at each antenna.

#### 2.2.2.1 FPGA Development based on RFNoC Framework

In this task, the testbed design is implemented with the software defined radio (SDR) devices USRP N321 and USRP X310. The baseband signal processing modules are developed and implemented based on the open source framework. Specifically, the FPGA based baseband modules are developed using the RFNoC 4 framework, whose general structure is illustrated in Figure 31. The RFNoC 4 framework is developed by Ettus, which is a network-distributed heterogeneous processing tool with a focus on enabling FPGA processing in USRP devices [4]. The key reason for implementing the FPGA modules in this task with the RFNoC 4 is that the framework provides a good migration feature within the open source SDR devices, especially with the USRP series devices. Also, the developed modules can be running in a stand-alone manner that directly controlled or managed by the customized C++/Python programs, while it is capable to be further interfaced and used in the GNU Radio framework, which is the major open source framework for wireless communication applications based on CPU processing.



Figure 31: RFNoC Development Framework [5]

Specifically, in this task, the FPGA baseband signal processing modules are programmed in the Verilog HDL language, where 3 key modules are designed and implemented, including Synchronization Module, Channel State Information (CSI) Estimation Module, and BPSK Demodulation Module. The modules are tested/validated in both online and offline methods. For the offline method, the Block Test Bench via System Verilog is used, which provides waveform simulations with customized inputs. For the online method, the FPGA modules are integrated and compiled to bit streams that will burn onto the FPGA of USRP N321 for onboard evaluations. USRP Hardware Driver (UHD) is developed for each of the developed FPGA modules, while the interfaces for GNU Radio is also developed in C++ so that the implemented FPGA modules can be re-loaded and called in the GNU Radio based programmes implemented in the CPU processing.

To enable the distributed signal processing with the constrained embedded computing resources at each antenna, the data frame is following the design as Figure 32. Specifically, the data frame consists of a preamble for timing synchronization, K pilots for K users' CSI estimation, and payload part for data transmission.



Figure 32: The data frame is consisting of a preamble for timing synchronization, K pilots for K users' CSI estimation, and payload part for data transmission

#### 2.2.2.2 Channel State Information Estimation Module

The CSI is the key information required for the multi-user MIMO scenarios to optimize the beamforming designs, so that the spatial diversity can be exploited to improve the system throughput when multiple users are served at the same time and spectral bands. Although transmitting all baseband signals from remote antennas to the CPU could exploit the global information for better CSI estimation and demodulation performance, it will also be a challenge to the fronthaul links with raw real-time baseband signals. Therefore, in this design for the distributed testbed, the CSI will be estimated locally with the embedded computing resources at each antenna.

Since each antenna is considered to have equipped with constrained embedded FPGA computing resources, the pilot design is following the structure as shown in Figure 32. For a system with K users, each user is assigned with a specific time slot so that a pilot is transmitted in that time slot for the CSI estimation purpose.

For a precise estimation of the CSI, the Discrete Fourier Transform (DFT) sequence with a length of 64 is used. Since the pilot for each user can be distinguished by its allocated slot in the pilot part of each data frame, all users will reuse the same pilot sequence to reduce the CSI estimation computation complexity in the FPGA implementation. Since different DFT sequences are mutually orthogonal, while the correlation of the same DFT sequence will produce 1 as outputs when the sequences are aligned. Therefore, the CSI estimation module is implemented as a Matched Filter (MF), whole filter taps are the DFT sequences. In this way, the estimated CSI can be obtained via the outputs of the MF when the inputs of the MF are the whole DFT sequences, while different user's CSI can be distinguished by the time slot this DFT sequence is located to.



Figure 33: Waveform simulation results to validate the CSI estimation module.

The correctness of the CSI estimation module is validated using System Verilog scripts and waveform simulations, whose results are illustrated in Figure 33. In this test case, the CSI for two users are randomly generated in MATLAB as [-0.5175+0.66215j, 0.100705+0.31072j], and the CSI vector is used to synthetic the received signals at the receiver. The CSI vector can be represented as [-16957+21861j, 3300+10181j] with the USRP baseband signal format of sc32. As can be observed from the highlighted parts in Figure 3, the estimated CSI for user 1, with I phase and Q phase part given by "CSI\_I\_detected\_U1" and "CSI\_Q\_detected\_U1" respectively, the values are [-16957+21860j] which matches with the expected outputs. Note that the difference is due to the fixed point representations in the FPGA calculation, where the accuracy will be +/-1 if represented in the sc32 format, and this value corresponds to +/- 1/32768 in absolute values (i.e., the quantified error for the estimated CSI is within the range of +/- 1/32768, and any difference in this range against the true values is considered accurate as expected). Following the same procedure, the estimated CSI for user 2 is also matching the expected values, with I phase and Q phase part given by "CSI\_I\_detected\_U2" and "CSI\_Q\_detected\_U2", respectively.

#### 2.2.2.3 Time Synchronization Module

Although the primary intention of this testbed is to study the use of AI in the distributed cell-less MIMO testbed, it is important to implement all necessary functions in the baseband signal processing, so that the testbed can support the real-world and real-time experiments. Therefore, in this part, the time synchronization module is implemented as a FPGA module in the RFNoC framework, which provides the fine timing recovery during the real-time reception for the later real-time signal processing modules, including the CSI estimation module and the PSK demodulation module.

The time synchronization is critical to the successful reception and decode the transmitted signals. After Digital Down Converter (DDC), the transmitted signals are among the real-time baseband data streams at the receiver. The time (change the test) synchronization module is to exploit the preamble sequences to determine the start of the received symbols.



Figure 34: A comparison between the desired waveform generated in MATLAB and the output of the FPGA based Timing Synchronization module.

To reduce the implementation complexity of this Timing Synchronization module via the embedded FPGA resources at the antennas, the preamble is reusing the same DFT sequence for the CSI estiamtion purpose. In the Timing Synchronization module, the timing synchronization is achieved via the peak detection based on the MF outputs for preamble part. The comparison shows that the results match the expectation between the desired waveform generated in Matlab and the output of the FPGA based Timing Synchronization module, which is illustrated in Figure 34. Note that the design here is simplified for the proof of concept purpose, the preamble sequence can be easily reconfigured with other advanced sequences, due to the reconfigurable design exploited by this task.

#### 2.2.2.4 PSK Demodulation Module

The payload part contains the transmitted symbols for individual users, which have been modulated and coded for transmissions and need to be demodulated and decoded into information bits. The system performance is also evaluated via this part, e.g., the bit error rate (BER) and data rates. For the proof-of-concept purpose, the testbed design exploits the following scheme to form the payload part, where the information bits is modulated into symbols via PSK, and then PSK symbols are pulse shaped via Squared Root Raised Cosine (SRRC) Filter as the baseband signals. Specifically, the SRRC filter is configured with a roll-off factor of 0.2, 6 output samples per symbol, and a filter span in 10 symbols.

To demodulate and decode the payload, the PSK demodulation module consists of a SRRC filter and PSK demodulator, where the filter taps of the SRRC filter use the same parameters as specified above, and the PSK demodulator is implemented via the Finite State Machine (FSM).



Figure 35: A comparison between the desired outputs of the SRRC outputs in MATLAB and the SRRC outputs from the implemented PSK Demodulation module.

For demonstration purpose, the BPSK scheme is implemented, where the correctness has been validated via the comparison between the desired outputs of the SRRC outputs in MATLAB and the SRRC outputs from the implemented PSK Demodulation module, as illustrated in Figure 35. In addition, the transmitted 32 bits 0xd9da7bea have been decoded correctly, which is shown in the results below the waveforms in Figure 35.

#### 2.2.2.5 Validation via Real-world Experiments

The designed baseband modules detailed in Section 2.2.2.2-2.2.2.4 are integrated under a single top module, which is then further integrated with the other N321 basic modules via the RFNoC 4 framework, e.g., the Radio Front module for RF interfaces, DDC for baseband to frequency band signal processing. The UHD and interfaces to GNURadio, are developed via C++.

The designed baseband modules are compiled via Xilinx Vivado, and integrated as part of the N321 FPGA images, which is then burned to the N321 for onboard tests, as illustrated in Figure 36. Post processing with recorded signals validate that the outputs of the onboard test match with the expected results as detailed in Section 2.2.2.2-2.2.2.4.



*Figure 36: Real-world tests for the designed modules* 

#### 2.2.2.6 FPGA Image Implementation for USRP X310 and N321

The proposed solution is exploiting the open source framework, i.e., RFNoC 4, for the FPGA development and integration with the existing software defined radio devices, especially the USRP devices. Therefore, the developed modules are implemented as an out-of-tree (OOT) module in the framework of RFNoC 4, which is then integrated as an additional Computing Engine (CE) with all in-tree modules provided by Ettus or any third parties from the open source world.

A block design is given in Figure 37, whose corresponding implemented blocks in GNU Radio for User 1 (U1) has been illustrated in Figure 38. Here are two feasible connections for the customized AIMM module: a) For debugging purposes, one specific Stream Endpoint can be created for each AIMM module, so that the AIMM module can be independently accessed and assessed, and b) To reserve more FPGA resources on USRP devices for CE, the AIMM module can be statically connected with the Digital Down Converter (DDC) module, which is achieved by configuring the Static Routing table fulfilled by the RFNoC image core yaml file. In this method, any received signals from the Radio will go through DDC and the AIMM module and save the FPGA resources from the extra Stream Endpoint, but the AIMM module can no longer be independently accessed and assessed without Radio and DDC module.



Figure 37: Block design for the customized FPGA core architecture for the USRP N321 and X310.



Figure 38: Implemented blocks for the customized FPGA core architecture for the USRP N321, where the module RFNoC srrcni is the designed and implemented AIMM module for User 1 (U1).

In the design of this cell-less testbed, each AP with a single antenna will be equipped with individual computing resources. Since each USRP N321 is equipped with two independent RX/TX channels, one USRP N321 can be implemented as two independent APs. In the testbed, this is achieved by duplicating the same AIMM module on the USRP N321 FPGA with different user index parameters. Then the two APs are implemented in parallel on the same USRP N321 device, and this method is exploited in the testbed as illustrated in Figure 39. The antennas for the different APs on the same USRP N321 can be physically separated by using extending cables for lab-scale proof-of-concept experiments. The design is backcompatible to host only one AIMM module on one USRP N321, which supports the larger scale experiment where APs are required to be largely separated in space.



Figure 39: Screenshot of the implemented FPGA modules on USRP N321

#### 2.2.2.7 USRP Hardware Driver Development for AIMM FPGA Module

The FPGA modules developed under RFNoC 4 framework also require the design and implementation of the block controller for usage, and this block controller is following the standard of USRP Hardware Driver (UHD). Since the designed AIMM module is an OOT module, its UHD must be designed and implemented where existing UHD for in-tree modules cannot be used.

The UHD block controller design mainly involves 3 key tasks as follows:

a) configuring and exposing the FPGA registers for user access, e.g., re-configurable parameters such as user index, pilot sequence and PSK mode selection.

b) configuring properties and their associated functions, e.g., setting/getting registers in a).

c) registering the block controller with its unique NoC-ID along with existing UHD block controllers.

The UHD block controller is for the access and control of the customised AIMM FPGA module in the CPU based programmes. This allows the flexibility of the FPGA design, where certain parameters can be re-loaded during runtime, e.g., different pilot sequences can be written to FPGA registers on demand via the UHD block controller.

With the UHD block controller, CPU based programmes can be implemented via GNU Radio or in C++. The C++ approach is ideal to further optimise the efficiency and performance of CPU based programmes, but this demands extra efforts in C++ programming and is very time consuming. In this testbed, the CPU based programmes are implemented via GNU Radio in Python. This approach is also favourable because the testbed can exploit existing signal processing modules in GNU Radio frameworks.

It is noticed that the UHD block controller development is for CPU usage, where there are differences between CPU architectures. In this testbed, there are two different CPU architectures involved, which are a) x86\_64 architecture for PC hosts, and b) Arm cortex-a9 for N321 embedded systems. The UHD source files, including the C++ headers files and implementation files, are the same for both architectures, while the key difference is on the compiling environment. To address this challenge, there are two common solutions, a) compiling in the local environment for both x86\_64 architecture and Arm cortex-a9 environment, i.e. to compile on PC and N321, respectively.

b) both compiling in the x86\_64 architecture, where the Arm cortex-a9 is cross-compiled with a dedicated cross-compiling environment in the x86\_64 architecture.

The first approach is seemingly a straightforward solution but it has several practical challenges:

- a) the embedded environment (e.g. Arm cortex-a9) has significantly less resources than the PC (e.g. x86\_64 architecture), including memories and CPU power for compiling purpose. Especially for open source environment, there are dependent toolboxes or libraries that are very demanding, and will fail to compile locally.
- b) the compiling is very time consuming due to the lack of computing resources in the embedded environment, which is not favorable for the developing stage.

Therefore, in this testbed, the cross-compiling solution is used for UHD development in the Arm cortex-a9 architecture. The development and compiling are achieved using the cross-compiling SDK provided by Ettus, where Ettus provides the corresponding SDK along with each N321 operating system image [6]. For this testbed, the developing environment is with UHD-4.0.0-104-g8f273305 version with GNU Radio v3.8.2.0-111-g6aad98a6 (Python 3.6.9). By compiling in the same PC via cross-compiling, the CPU based scripts developed and tested with x86\_64 architecture can be migrated to the Arm cortex-a9 environment, where it should be noticed that all GUI related features are to be avoided for embedded executions.

#### 2.2.2.8 Data Frame Structure Design for the Multi-user CSI Estimation

The CSI is the key information required for the multi-user MIMO scenarios to optimize the beamforming designs, so that the spatial diversity can be exploited to improve the system throughput when multiple users are served at the same time and spectral bands. Although transmitting all baseband signals from remote antennas to the CPU could exploit the global information for better CSI estimation and demodulation performance, it will also be a challenge to the fronthaul links with raw real-time baseband signals. Therefore

in this design for the distributed testbed, the CSI will be estimated locally with the embedded FPGA computing resources at each BS.

The pilot design is following the structure, whose structure can be further adapted to popular frame designs including IEEE 802.11. For a system with K users, each user is assigned with a specific time slot so that a pilot is transmitted in that time slot for the CSI estimation purpose. For a precise estimation of the CSI, the pilot is using the discrete Fourier transform (DFT) sequence with a length of 64. The CSI estimation is fulfilled by the matched filter (MF), whole filter taps are the DFT sequences. In this way, the estimated CSI can be obtained via the outputs of the MF when the inputs of the MF are the whole DFT sequences, while different user's CSI can be distinguished by the time slot this DFT sequence is located to. To reduce the hardware resource complexity, all users are reusing the same pilot sequence.

#### 2.2.2.9 Baseband Module Experiment Hardware Setup

An initial testbed has been prototyped with the designed baseband modules in Figure 40, where two independent single-antenna BSs and two single-antenna users are implemented. For evaluation purpose, the testbed contains only essential modules including timing recovery, CSI estimation and PSK demodulation modules. These modules provide the minimum setup for a complete link layer communication system, where key metrics can be evaluated including bit error rate (BER), data rate and spectrum efficiency.



Figure 40. An initial testbed implementation with 2 single-antenna BSs and 2 single-antenna users.

Each antenna has been separated with 0.5m apart, where the transmit power has been set as 32dB, and the central frequency has been set as 915 MHz.

#### 2.2.2.10 Baseband Module Experiment Software Setup

Since the designed testbed is software-defined, the hardware part needs to work together with the software so that the functionalities can be defined and exploited by the high-level scripts. To facilitate the test of the baseband module experiment, the developed AIMM modules are connected in the *GNURadio grc* files as illustrated in Figure 41.



Figure 41. GNURadio configurations for the baseband module experiment.

Also to facilitate the observation of the BER performance and the estimated CSI, several GUI interfaces have been configured and arranged as illustrated in Figure 41. Specifically, the N321 has been configured in RFNoC mode so that the on-board modules can work with the developed AIMM baseband modules, where the sample rate has been set as 245.76MHz for all N321 fulfilling BSs and Users. The receiver side is using RX2 channel, while the transmitter side is using TX/RX channel. The baseband sample rate is set as 1MHz, and all N321s have been set to use external reference clocks and Pulse Per Second (PPS) sources, which is to synchronize the local oscillator and to mitigate the phase noise. It is noted that there is a bug with the GNURadio 3.8 with the RFNoC development, where the external synchronisation is not properly implemented. Although in the GRC interface there are indeed these options can be selected and configured in the RFNoC Graph (Device) module as illustrated in Figure 41, actually they are not working when generated to python scripts. The solution is to generate the python script using the GNURadio and then manually add the external synchronisation. But after this operation, the script needs to be executed as a stand-alone python script from the command window, instead of directly run in the GNURadio companion (the grc graph interface). Everytime the file is re-generated in the GNURadio companion, this manual configuration should be re-done, as the re-generation will remove any manually added contents.

#### 2.2.2.11 Baseband Module Experiment Results

The initial testbed has been evaluated in the lab environment, whose results are demonstrated in Figure 42. This result and the baseband module experiment has also been presented in EuCNC & 6G Summit 2022 Workshop with the theme "Testbeds and Platforms for AI-enabled Massive MIMO". Note that in the experiment, only the uplink from the user to the BS has been evaluated.



Figure 42. Baseband Module Experiment Results.

By separating the antennas largely enough without using beamforming algorithms, the BSs can successfully estimate the corresponding CSIs between the antennas. This is illustrated in Figure 42, where the estimated CSI between BS1 and user 1 is represented by the real and image parts as h\_11\_real and h\_11\_imag and the other parameters are following the same notation method. During the experiment, both users can achieve a spectrum efficiency above 0.629 bps/Hz, which is expected for the minimal system implementation with a low bandwidth of 200 kHz and BPSK modulation schemes.

It is noticed the baseband module experiment results met with the expected results, which demonstrates the developed AIMM baseband modules can work in the real-world scenarios. However, it should be noticed that since there are no beamforming algorithms implemented while the transmission is in the same time and same frequency band, it will suffer from sever interference between users. This will be addressed using the closed-loop experiment with beamforming algorithms as detailed in the following subsections.

#### 2.2.2.12 Closed-loop Experiment Setup

The closed-loop experiment is designed and conducted to get a more in-depth test of the designed modules and the AI beamforming algorithms. The scheme of the closed-loop experiment setup is illustrated in Figure 43, where the aggregation of the CSI feedbacks, the illustration of the BERs for each user and the beamforming is conducted on a PC. Note that since the baseband signals have been pre-processed with the customised AIMM baseband modules in FPGA onboard the N321, the feedback link can directly collect the estimated CSI. This will much reduce the bandwidth for the feedback links especially when system scales up, comparing to the traditional solution like remote head to transmit raw signals.



Figure 43. Closed-loop experiment setup for 2 single-antenna users and 2 single-antenna BSs.

Also to facilitate the user interaction, GUIs have been used to illustrate the real-time BER on the same PC along with the debugging information printed in the command window executing the GNURadio python scripts.

#### 2.2.2.13 Closed-loop Experiment Results

The hardware implementation of the designed closed-loop experiment is illustrated in Figure 44. The parameter configurations are similar to the open-loop experiment as detailed in Section 2.2.2.10-2.2.2.12.



Figure 44. The real-world implementation of the closed-loop experiment.

The experiment has been setup in the lab environment in the Communication Lab at Loughborough University, where both zero-forcing algorithm and the AI-based algorithm have been evaluated. The obtained results are given in Table 2.

# Table 2. Results comparison between AI-based and ZF-based beamforming algorithms in the closed-loop experiment,where the transmit power has been set as 32dB, and the central frequency has been set as 915 MHz.

BER Performance	User 1	User 2
AI-Based Beamforming	0.0004	0.0006
Zero Forcing Beamforming	0.0097	0.0121

It can be seen from Table 2 that under the same experiment scenarios, the AI-based beamforming algorithm can outperform ZF based beamforming algorithm. It also demonstrated that the designed cell-free testbed has fulfilled its primary objective, where it can serve as the testbed and reconfigured to evaluate and compare different algorithms. Specifically, the integration of new algorithms and to test it in the real-world scenario has been much facilitated thanks to the software-defined architecture of this testbed, while the real-time applications and services like beamforming has been enabled due to the much-reduced computational complexity via FPGA.

### 2.3 ML for L1 in GPU-enabled gNB

The hybrid architecture of general-purpose CPU/GPU-based gNB architecture is shown in Figure 45. It implements the baseband functionalities of the DU of the radio system. It exploits CPUs for bit-level processing, while the GPUs are used to accelerate computationally exhaustive functionalities with a high level of parallelism [7]. The RU is based on an USRP X310 SDR board [8], interfaced over split 8 (time-domain IQ samples) with the DU.



Figure 45: The overall architecture of GPU-enabled gNB.

#### 2.3.1 The platform architecture and Data collection

The processing architecture follows 5G NR release 15 SA and will be extended according to the needs of the project, e.g., by upscaling to a higher number of antennas. The functionalities will be enhanced to support the AI algorithms for radio resource optimisation, developed in WP4, i.e., AI-enhanced Physical Uplink Shared Channel (PUSCH) channel estimation [9] and possible extension to equalization/demapping.

Since the original focus has been on optimizing the UL receiver chain for the shared data channel, i.e., PUSCH channel, of the gNB, the existing platform needs to be extended such that it supports necessary tools for performance evaluation and data generation. This includes the following efforts:

The implementation of a "mirrored" gNB, i.e., the existing PDSCH chain is modified in a way to serve as PUSCH transmitter, i.e., emulating the respective UL functionalities of a device. By doing so, the PUSCH receiver performance can be evaluated for various system parameters and different use cases.

Performance evaluation of the current implementation using conventional signal processing blocks, in two modes:

- Software (SW) loopback mode, shown in Figure 46, where RF transmission chain is bypassed and short-circuited such that Tx IQ samples are fed directly into the PUSCH receiver over a simulated channel without being transmitted over the air.
- RF loopback mode, shown in Figure 47a, with the RF chain being included, such that Tx and Rx communicate over a channel emulator. In this way, the impact of real RF imperfection and various channels on system performance can be easily assessed.

The preparation of a framework for performance testing of either individual blocks or grouped conventional signal processing units and ML modules. The KPIs to be evaluated within the project are link performance (in terms of improved NMSE – normalized mean squared error - and BLER), and the latency of the RX chain. For this the existing virtualized testbed had to be further extended. This includes the implementation of a test environment for each module under consideration (both blocks including conventional and ML-based processing).



Figure 46: SW loopback mode.



Figure 47: Setup for data generation and testing: a) RF loopback mode; b) E2E connection with MTP UE.

Defining and implementing set of data collection points for the training of AI models. The training of ML modules/models initially starts by using synthetic data (i.e. data artificially generated by simulated channel models and hardware impairments). However, to enable expected KPI improvements in the considered testbed (with real RF chain in an OTA environment), it is of crucial interest to collect real data within the TRX chain. This real data (transfer to WP4) will be used to further train the ML models, bringing them to a more realistic operational environment. This data collection points, and corresponding acquisition methods are implemented on the CPU/GPU platform.

Finally, once the the ML models are trained using the collected data sets, they will be integrated into the Rx chain. Using the implemented framework, their performance and latency will be evaluated.



Figure 48. ML PoC setup in the Lab

#### 2.3.2 Data collection

The setup, shown in Figure 47a, is used to collect the data (signal received over the RF chain) for the training of the ML-model (in WP4). The lab setup is shown in Figure 48. The baseband IQ samples from PUSCH Tx chain (PDSCH Tx chain configured to work in "mirrored" mode facilitated by configuration similarity between Tx and Rx processing chain) are derived to transmit RF board (X310), up-converted to 3.5 GHz carrier frequency (in general X310 operates in 20MHz – 6GHz frequency range, thus allowing to experiment in sub-6GHz frequency range) and further delivered to the Spirent VR5 RF channel emulator [10]. The output of the real-time RF channel emulator is fed to the receive X310 board, down-converted to received IQ samples and further derived to the Rx PUSCH chain. The setup operates in real-time, i.e., both Tx and Rx chain processing satisfy the latency constraint of 500 us, which is the Transmit Time Interval (TTI), or slot duration, of the 5G configuration with numerology  $\mu = 1$  (subcarrier spacing of 30 kHz).

Channel model	TDLA30, TDLB100, TDLC300, Rayleigh Flat
User velocity [kmph]	3, 30, 70, 120
Received SNR [dB]	0, 5, 10, 15, 20, 25, 30

Table 3: Set of channel parameters used in data collection.

Table 4: System Parameters.		
Carrier frequency	3.5 GHz	
Subcarrier spacing/numerology	30 kHz/ μ = 1	
Number of PRBs	273	
Number of PUSCH symbols per slot	14	
Number of DMRS/Data symbols per slot	1/13, 2/12, 3/11	
MCS	10 (16 QAM)	



Figure 49: One realization of TDLA30 in RF emulator.

The real-time RF channel emulator is used to implement a set of predefined channel models for producing the data set that includes the effects of real RF chain in an RF environment. The lab setup, with static locations of Tx/Rx antennas, provides channel scenarios with very low variations of time and frequency selectivity. Therefore, it cannot be used to fully demonstrate the full range of performance improvements offered by ML models, as well as to showcase their generalization for the wide range of channel parameters.

The predefined set of channel models and their parameters (time selectivity and received SNR), chosen to cover the wide range of potential scenarios, is used to create parameter grid for the data collection campaign, as shown in Table 3. The Time Delay Line (TDL) channels are statistical channel models that 3GPP recommends for evaluation of 5G systems in multipath fading propagation conditions [11]. The initial set of channels contains TDLA30, TDLB100, TDLC 300, each characterised with different delay spreads (frequency selectivity) of 30 nm, 100 ns, and 300 ns, respectively. Additionally, the Rayleigh flat channel is also emulated to involve flat channel conditions necessary to improve the generalization of the ML-model. Furthermore, each channel is emulated for different SNR conditions, to provide sufficient diversity of noise scenarios because ML-based wireless modules are shown to be very sensitive to SNR range. To further generalize the ML-model, several user velocities are emulated to cover the wide range of user mobility scenarios, from very-low (3 kmph) to high (120 kmph).



Figure 50: PRB time-frequency structure containing: (a) one DMRS symbol, (b) three DMRS symbols.



Figure 51: ML integrationn concept and data collection points

A single measurement/data collection of the duration of 10 seconds, i.e., 1000 5G frames, corresponds to one of 112 (= 4 channel models x 4 user velocities x 7 SNR values) combinations of considered channel parameters. In this way, the collected data set contains all possible combinations of channel variations. Figure 43 shows a single realization of TDLA 30 channel shown in Spirent VR5 Graphical User Interface (GUI).

The (5G) system parameters are given in Table 4. The carrier frequency of 3.5 GHz belongs to 5G frequency band n78 and a subcarrier spacing of 30 kHz is used, corresponding to 5G numerology  $\mu$  = 1, such that the full bandwidth occupancy of 100 MHz contains 273 Physical Resource Blocks (PRBs). To fully investigate the PUSCH channel estimation impact on system performance, all 14 OFDM symbols in a slot are used for PUSCH symbols, using three different DMRS configurations. For the measurement campaign, the data is collected for 1, 2, and 3 DMRS symbols per PUSCH slot. The time frequency structure of one PRB with 1 and 3 DMRS symbols is shown in Figure 50a and Figure 50b, respectively. The single MCS=10 is chosen because channel

estimation/ML-model operates only on DMRS inputs. Therefore, the model training and its influence on the system performance do not depend on data symbols, thus is transparent to applied MCS in the Transmission Block (TB).

The data collection dump points, necessary for the training of the channel estimation ML module, e.g., as solution proposed in WP4, implemented in PUSCH receiver are shown in Figure 51. It is also depicted which part of the channel estimation chain will be replaced with ML-based module.

The previously described set of channel models and system parameters will be extended in the future for more advanced scenarios and system configurations to further train the ML-models, thus allowing for better generalization.

#### 2.3.3 ML-model conversion for optimized inference on GPU

After the ML-block is trained in a Tensorflow or PyTorch framework, to further improve the inference performance, it must be converted to to GPU-optimized inference engine, named TensorRT [12], as shown in Figure 52. TensorRT enables for higher throughput and lower inference latency. The developed framework is used to further optimize TensorRT engine inference performance and hardware utilization. An example of latency evaluation (y-axis) of one ML-model with different batch sizes (individual curves) for the different data loads (x-axis) is shown in Figure 53. It is shown that models with low batch size perform in general better for full range of realistic data loads.



Figure 53: Latency evaluation of the AI model

Furthermore, the advanced NVIDIA tools are used to further profile the inference execution and dissect the ML-model to measure the performance of individual layers. A screenshot of Nsight Systems tool, depicted in Figure 54, shows the execution timeline of individual layers for a single inference execution. This visualisation can help to further optimize the ML-model for more efficient inference.

By observing the latency of each individual layer, shown in Figure 46, the "slow" kernels can be identified and optimized during model refinement process, e.g., by changing some of the hyperparameters (number of channels in a layer, stride, dilation, ...).



Figure 54: An example of latency evaluation



Figure 55: An example of latency evaluation

#### 2.3.4 An interactive graphic visualisation

The real-time interactive GUI is implemented to allow for visual demonstration of KPI improvements, such as SNR, BLER, as well as waveform visualization of the specific stage within the Rx chain, as shown in Figure 56.



Figure 56: A screenshot of the real-time interactive GUI (initial version) ot of the real-time interactive GUI (initial version)

The raw (least-square) channel estimation (depicted by yellow curve) visualizes the amount of noise in the received signal, while the red curve demonstrates the noise-filtered raw channel estimate that is fed to the equalizer. The equalization widget on GUI visualizes the signal quality at the input to the soft-demapper, while Demapper Log Likelihood Ratio (LLR) values depict the soft bits that are fed to the LDPC decoder. The more LLR curves are separated, the better performance.

As an example, as shown in Figure 57, by setting the TDLB100 channel and inserting the noise such that received signal is set to SNR = 10 dB, the corresponding GUI in Fig. 50 depicts the signal quality and corresponding measurements in the receiver chain.



Figure 57: VR5 control interface: an example of setting the channel parameters (channel model and SNR selection).

The button, shown in Figure 56 interactively controls the receiver operation from performing the legacy channel estimation to AI-based channel estimation. By switching it on/off, the ML-acceleration of the channel estimation processing block is activated/deactivated, such that performance improvements can be shown. The intuitive visual experience provided by the real-time interactive GUI, should showcase the potential of ML-enhanced PHY. The GUI also visualizes the latency of the transmit and receive chain.

#### 2.3.5 Conclusion and Outlook

While working on the the project, we made large progress on the integration of the L1 ML-component into the GPU-based gNB receiver. The testbed platform is designed to provide high flexibility for prototyping and testing of ML-enhanced PHY functionalities. The presented methodology, although conducted on the experimental platform, has provided us a deep insight into the implementation issues potential relevant for the product platforms. That includes the required efforts to obtain low latency (required by strict L1 timing requirements), optimal selection of batch sizes, model size selection, ML-model optimization, model training generalization, and conducting of data collection and measuring campaign.

The development progress on platform development, and conducted integration methodology are presented in EuCNC & 6G Summit 2022 Workshop on "Testbeds and Platforms for AI-enabled Massive MIMO", organized by Nokia Bell Labs Stuttgart as WP6 leader.

While channel estimation presents a small, but very relevant processing sub-component in the receiver chain, the learned lessons allowed us to expand this expertise for performing more complex integration of ML components into the transceiver chain.

The further plans involve implementation of RF real-time end-to-end communication (both transmitter and receiver chains having ML-enhanced components) model as an important milestone towards native AI/ML air interface (fully AI-defined L1 interface).

### 2.4 AI Based Interference Detection Testbed

ThinkRF has made significant progress over the last 12 months on the AI based Interference Detection capabilities with detection high-accuracy across a number of Interference sources including Narrow band (jamming), adjacent Channel and Co-Channel interference. The initial Machine Learning design was trained using Simulated MatLab 4G/5G signal samples that comprised of clean signal as well as samples with Physical RF channel impairments of various degrees. Later stages of testing included real-world 4G/5G signals from local Base-stations transmitters with lab based injection of interference sources. A significant amount of assessment in both feature engineering design and AI algorithm hyper parameter modification was required. In addition, the latter testing scenario require the integration of ThinkRF's real-time Spectrum Analyser, embedded processor (for real-time detection inference), live antenna for spectrum signal capture as well as Interference lab set-up.

This project is the first step in developing a commercial product offer. The ongoing project task aims at developing an edge AI-based solution to detect the presence of anomaly or interference in the 5G PHY downlink, with the minimum false positive and negative rates. To this end, we are exploring digital signal processing and machine learning (ML) tools to analyse 5G spectrum for robust interference detection in a completely unsupervised way. The objective behind employing unsupervised learning for interference detection is to detect previously unseen interference events without any prior knowledge about these.

The developed interference detection approach is initially trained and tested on synthetic data generated using MATLAB 5G/LTE toolboxes. These 5G waveforms are contaminated with background noise and two sources of interference at different SNR and SINR levels, namely co-channel and adjacent channel interferences. SNR and SINR stand for signal to interference ratio and signal to interference plus noise ratio, respectively.



Fig.63. Interference types and causes in wireless networks.

#### 2.4.1 Overview of Interference in Wireless Networks

Interference is one of the most performance-limiting factors in wireless networks, which isoften used to refer to the addition of unwanted signals to a signal of interest. There are several determinants of interference; one can mention (i) the network geometry or problems related to spatial distribution of concurrently transmitting nodes, (ii) the path loss law or signal attenuation with distance, and (iii) equipment malfunctions. As shown in Figure 63, the most common interferences are due to unintentional adjacent or co-channel emissions, or any other sources of unwanted emissions. There are also so-called intentional interferences, also known as jamming, which are potentially threatening public safety.

#### 2.4.2 Autoencoder-based anomaly detection

Autoencoders are a specific type of feedforward neural networks where the input is the same as the output. They comprise of the input into a lower-dimensional code and then reconstruct the output from this representation. The code is a compact "summary" or "compression" of the input, also called the latent-space representation. An autoencoder consists of 3 components: encoder, code and decoder. The encoder compresses the input and produces the code, the decoder then reconstructs the input only using this code. To build an autoencoder we need 3 things: an encoding method, decoding method, and a loss function to compare the output with the target.

The autoencoder's dimensionality reduction technique can be applied to many problems include image denoising and anomaly detection. The latter is achieved by training the autoencoder in a "normal" condition environment. Once that is achieved, the trained autoencoder is now used to reconstruct the operation RF signal environment, where it is unable to do so, may indicate an RF anomaly is present. The anomaly detection process is generally based on the analysis of the MSE error (see Fig. 64).



Figure 64. Autoencoder-based architecture for anomaly detection.

The ThinkRF testbed, shown in Figure 65, comprises of several components, including embedded processor (GPU) to run the advanced AI machine Learning algorithms and techniques to assess and evaluate Training parameters, ThinkRF Spectrum Analyzer with Omni-directional antenna. Network access and PC to control and monitor the set-up and provide remote access to users.

The testbed set-up will provide the ability to Detect and locate RF sources for interference from both unintended and nefarious sources. Although, in the short-term only interference detection is support as geolocation capabilities are currently under development.



Figure 65: ThinkRF Wireless Security Testbed.

#### 2.4.3 Recent Activities

The goal behind the current project task is to explore AI and signal processing software and hardware powerful capabilities for robust interference detection in 5G/LTE PHY. The work completed during the last five months led to invaluable learnings that should be thoroughly considered for the next steps of this project. It has been revealed the importance of data extracted from 5G frequency domain for robust interference detection.

Overall, we achieved promising results in detecting two common interference types in 5G networks, which are adjacent- and co-channel interference. The results presented in the current report are related to experiments conducted on synthetic data generated using MATLAB 5G toolbox, at different SNR and SINR levels. As such, an in-depth analysis has shown the importance of feature engineering, in frequency domain, prior to machine learning-based interference detection. This is achieved by exploiting signal processing tools to amplify 5G parts contaminated by interference.

Based on the current results of this project task, we identified several open research issues that need to be investigated as well as planned main actions for the next project steps, such as:

- make the approach more scalable and robust, more complex scenarios will be investigated, such as

   (i) testing and generalizing the approach to data with different bandwidth and channel Spacing, and
   to multiuser 5G downlink cases and (ii) considering other types of interference, particularly those
   seen in time division duplex (TDD) mode.
- Capture real-world data, using ThinkRF receivers, from different locations and at different points in time. Because more diverse training data generally leads to more accurate AI models. The captured data will be compared with MATLAB synthetic data though a statistical analysis. This will help addressing problems related to distributed shift which occurs when the training data and test distributions are different.
- Study effects related to I/Q imbalance, phase offset, and other receiver' impairments.

The improvements that are in progress will be introduced to the ThinkRF Interference Detection testbed periodically and as results show good promise.

#### 2.4.4 Conclusions and Next Steps

The goal behind the current project task is to explore AI and signal processing software and hardware powerful capabilities for robust interference detection in 5G/LTE PHY. The work completed during the last five months led to invaluable learnings that should be thoroughly considered for the next steps of this project. It has been revealed the importance of data extracted from 5G frequency domain for robust interference detection. The following conclusions are observed:

Overall, we achieved promising results in detecting two common interference types in 5G networks, which are adjacent- and co-channel interference. The results presented in the current report are related to experiments conducted on synthetic data generated using MATLAB 5G toolbox, at different SNR and SINR levels. An in-depth analysis has shown the importance of feature engineering, in frequency domain, prior to machine learning-based interference detection. This is achieved by exploiting signal processing tools to amplify 5G parts contaminated by interference.

Based on results and conclusions of this project task, we identified several open research issues that need to be investigated as well as planned main actions for the next project steps, such as:

To make the approach more scalable and robust, more complex scenarios will be investigated, such as (i) testing and generalizing the approach to data with different bandwidth and SCS, and to multiuser 5G downlink cases and (ii) considering other types of interference, particularly those seen in time division duplex (TDD) mode. Capture real-world data, using ThinkRF receivers, from different locations and at different points in time. Because more diverse training data generally leads to more accurate AI models. The captured data will be compared with MATLAB synthetic data though a statistical analysis. This will help addressing problems related to distributed shift which occurs when the training data and test distributions are different. Study effects related to I/Q imbalance, phase offset, and other receiver' impairments. Train and test the proposed approach on real-world data as well as implementation on embedded devices (e.g., Jetson series).

## **3** Conclusions and Future Work

The performed work in WP6 has delivered the expected testbeds and Proof-of-Concepts according the AIMM project plan. Some of the partners, such as IMST, got six months extension to finalize the project. Thus far, the project partners have mostly accomplished the final versions of the proposed testbeds, containing the key AI/ML-based capabilities and data collection frameworks. This is achieved whether by modifying their existing testbed setups or by designing completely new testbed architectures. The testbeds and frameworks are equipped with GUI to visualize and demonstrate the performance enhancements achieved by implementing the proposed methods. The insight obtained from the implementation efforts allow for further investigations in the relevant research areas.

More specifically, the experience obtained during designing and building the testbeds, collecting the real data sets, their pre-processing, training of ML modules, and, finally, evaluation of system performance, resulted in novel methodologies and skillsets for system development, that differ from the standard approaches for development of classical communication systems.

#### References

- X. E. B. a. Kits, "https://www.xilinx.com/products/boards-and-kits/see-all-evaluationboards.html," Xilinx, [Online]. Available: https://www.xilinx.com/products/boards-andkits/see-all-evaluation-boards.html.
- [2] E. Björnson and L. Sanguinetti, "A New Look at Cell-Free Massive MIMO: Making It Practical With Dynamic Cooperation," in 2019 IEEE 30th Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC), Istanbul, Turkey, 2019.
- [3] A. Sadek, A. Tarighat and A. H. Sayed, "A Leakage-Based Precoding Scheme for Downlink Multi-User MIMO Channels," *IEEE Transactions on Wireless Communications*, vol. 6, no. 5, pp. 1711-1721, May 2007.
- [4] "https://kb.ettus.com/RFNoC," RFNoC. [Online].
- [5] "https://www.gnuradio.org/grcon/grcon20/grcon20\_RFNoC\_4\_Part2.pdf," RFNoC 4 Workshop, [online].
- [6] "https://files.ettus.com/binaries/cache/n3xx/meta-ettusv4.0.0.0/n3xx\_common\_sdk\_default-v4.0.0.0.zip," [Online].
- [7] "https://developer.nvidia.com/cuda-toolkit," NVIDIA Cuda Toolkit, [Online]. Available: https://developer.nvidia.com/cuda-toolkit.
- [8] "https://www.ettus.com/all-products/x310-kit/," Ettus X310 Kit, [Online]. Available: https://www.ettus.com/all-products/x310-kit/.
- [9] Y. Chen, J. Mohammadi, S. Wesemann and T. Wild, "Turbo-AI: Iterative Machine Learning Based Channel Estimation for 2D Massive Arrays," November 2020. [Online]. Available: https://arxiv.org/abs/2011.03521. [Accessed Preprint arXiv: 2011.03521 [eess.SP]].
- [1 "https://support-
- 0] kb.spirent.com/resources/sites/SPIRENT/content/live/DOCUMENTATION/10000/DOC10394/ en\_US/UM\_VR5\_v1\_20\_A2.pdf," Spirent VR5 Channel Emulator. [Online].
- [1 3GPP, "TS 38.104, Annex G".
- 1]
- [1 "https://docs.nvidia.com/deeplearning/tensorrt/quick-start-guide/index.html," NVIDIA
- 2] TensorRT Quick Guide. [Online].
- [1 "https://www.xilinx.com/products/boards-and-kits/see-all-evaluation-boards.html," Xilinx
- 3] Evaluation Boards and Kits. [Online].

