

# Channel Estimation for Massive MIMO systems using Tensor Cores in GPU

Bhargav Gokalgandhi\*, Ivan Seskar<sup>†</sup>

WINLAB, Rutgers University

671, US-1, North Brunswick Township, NJ 08902

Email: \*bgokal@winlab.rutgers.edu, <sup>†</sup>seskar@winlab.rutgers.edu

**Abstract**—For efficient use of Massive MIMO systems, fast and accurate channel estimation is very important. But the Large-scale antenna array presence requires high pilot overhead for high accuracy of estimation. Also, when used with software-based processing systems like CPUs and GPUs, high processing latency becomes a major issue. To reduce Pilot overhead, a Pilot transmission scheme in combination with PN Sequence correlation based channel estimation scheme is implemented. Then, to deal with the issue of high processing latency, Tensor Cores in Nvidia GPUs are used for computing the channel estimation. Experiments are performed by using Nvidia V100 GPU in the ORBIT Testbed to show the performance of the Pilot transmission scheme. By varying factors like PN sequence length, Channel Impulse Response length, number of multiplexed transmitters, and scale of MIMO, the accuracy and processing latency of Tensor Core implementation of the Channel Estimation is evaluated.

**Index Terms**—Massive MIMO, Channel Estimation, PN Sequence, Pilot Design, GPU, Tensor Cores.

## I. INTRODUCTION

To effectively use Massive MIMO systems, channel characterization and Channel State Information (CSI) acquisition are extremely crucial tasks. When performing real-time Channel Sounding and Channel Estimation in practical scenarios, issues related to high pilot overhead and pilot contamination need to be addressed [1]. The Large-scale antenna systems present at transmitter as well as the receiver make it difficult to balance the pilot overhead and data transmission time within the limited channel coherence time window. Furthermore, careful design and transmission of Pilot sequences to get reliable CSI is extremely important for precoding/decoding of data. Channel Sounding techniques and architectures for CSI acquisition in a variety of Massive MIMO and mmWave scenarios have been considered [2]–[9]. Also, techniques which address issues related to pilot contamination, pilot sequence overhead, antenna array design, etc. have been extensively studied for various Massive MIMO scenarios [10]–[14].

This project was funded by the NSF "COSMOS" Project under grant number CNS-1827923.

Recently, consideration has been given to combining Massive MIMO systems with high-performance servers consisting of multi-core CPUs and GPUs at the back-end for baseband processing. CPUs and GPUs can provide fast processing while giving high flexibility for implementation and testing of various baseband processing tasks. But the Large-Scale antennas arrays at the front-end of Massive MIMO systems result in high volume of data to be processed. This makes the processing for real-time Channel Estimation extremely difficult when using CPUs and GPUs for processing.

So, an investigation into the performance of channel estimation for Massive MIMO systems when implemented in software-based processing systems like GPU is required. Especially, consideration into the trade-off between processing latency and estimation accuracy is required when implementing algorithms in real-time Massive MIMO scenarios. Recently, investigation has been done related to implementation of various channel estimation algorithms. In [15], Least Squares Channel Estimation is implemented by using GPU and its performance in terms of processing latency is shown. [16] measures the performance of Least Squares Channel Estimation of Massive MIMO-OFDM systems through the error performance obtained after detection of M-QAM symbols. [17], [18] investigate the use of Deep Learning with GPUs to accelerate the channel estimation process for mmWave Massive MIMO systems. An issue with above channel estimation algorithms is that they have either bad error performance at low SNRs in case of LS, or in case of Deep Learning require training and high volume of initial data to get the required high performance. So, to solve the issue of low training overhead, and to maintain high estimation accuracy and low computational complexity, we look into sequence correlation based channel estimation. The advantages and limitations of correlation based channel estimation have been investigated [19], [20], but its implementation in a software-based processing system is yet to be completely explored. Specifically, exploration of newer architectures like Tensor Cores for factors like accuracy

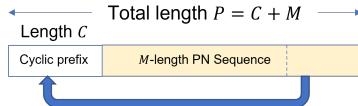


Fig. 1: Pilot Sequence after adding Cyclic Prefix to the PN sequence

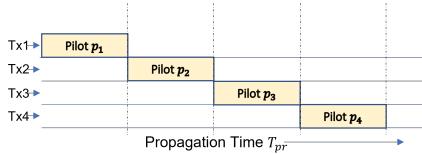


Fig. 2: Sequential Pilot transmission

and computational complexity of channel estimation need to be addressed.

- 1) We implement a Correlation based Channel Estimation algorithm which uses PN sequences as Pilot signals. We use the auto-correlation properties of the PN sequences to design multiplexed Pilot transmission which can reduce the Pilot overhead while maintaining low error for the Channel Estimation.
- 2) We then implement the Channel Estimation algorithm for Massive MIMO systems in a GPU and show the performance of Channel Estimation while varying factors such as PN sequence length, number of multiplexed pilots, Channel Impulse Response length, and Massive MIMO scale.

## II. SYSTEM MODEL

Consider an  $N_t \times N_r$  MIMO system where  $N_t$  and  $N_r$  are the transmitting and receiving antennas respectively. Each transmitting antenna  $t$  sends a Pilot sequence of length  $P$  which is represented by vector  $p_t$ . We consider an i.i.d. Gaussian channel with maximum  $L$  length CIR. Within the  $L$  length channel there are  $L_{nz} \leq L$  number of taps spread randomly with uniform distribution, and each tap is normalized such that maximum power of each tap is  $\frac{1}{N_t \sqrt{L_{nz}}}$ . Firstly, we consider the sequential transmission of pilots. To deal with the high pilot overhead for sequential transmissions, we then consider spatially multiplexed transmission case, where multiple transmitters send Pilot sequences simultaneously.

### A. Sequential Pilot transmission

We consider the use of an  $M$ -length PN sequence as Pilot signals. The circular auto-correlation properties of

$M$ -length PN sequence are well known and can be given as,

$$R[n] = \frac{1}{M} \sum_{m=0}^{M-1} x_t[m] x_t^*[m+n] = \begin{cases} 1, & \text{if } n = 0 \\ -\frac{1}{M}, & \text{if } 1 \leq n \leq M \end{cases} \quad (1)$$

Where  $R[n]$  is the circular auto-correlation, and  $*$  denotes the complex conjugate. The correlation in Eq. 1 approximates to the Delta function for long PN sequences. So, if we transmit PN sequences as pilots, we can then perform correlation of the received PN sequence with the transmitted PN sequence to get an estimated CIR.

To exploit this property, we create a Pilot sequence consisting of a  $M$ -length PN Sequence denoted by  $s = [s_0, s_1, \dots, s_{M-1}]$ , and a Cyclic Prefix (CP) of length  $C$ . This CP consists of the last  $C$  samples of the PN sequence, and  $L \leq C \leq M$  since we need at least  $L$  samples to absorb the ISI (Inter-Symbol Interference) effect of the multipath channel. More importantly, the CP gives a periodic nature to the Pilot sequence. So, when transmitted through the channel, circular correlation can be performed on the received PN sequence to get the CIR. An illustration of the Pilot sequence of length  $P = C + M$  is shown in Fig.1.

The Pilot sequences are passed through the  $L$  length channel. An illustration of the sequential transmission case is shown in Fig. 2. After sequential pilot transmissions by all transmitting antennas, the received PN sequence can be shown in the form of matrix multiplication. The received signal at antenna  $r$  for transmitting antenna  $t$  will be,

$$Y_r = \mathbf{S}_t h_{r,t} + W \quad (2)$$

where  $Y_r$  is the  $M \times 1$  vector at receiver  $r$ ,  $x_t$  is the Pilot sequence for transmitter  $t$ ,  $h_{r,t} = [h_0, h_1, \dots, h_{L-1}, 0, \dots, 0]^T$  is the  $M$  length CIR. Here, within the  $L$  indices, only  $L_{nz}$  number of channel taps exist and rest of the values are zero.  $\mathbf{S}_t$  is the matrix representation of the PN sequence in  $M \times M$  circulant form i.e. first row of matrix is  $[s_0, s_{M-1}, s_{M-2}, \dots, s_1]$ , second row is  $[s_1, s_0, s_{M-1}, \dots, s_2]$  and so on, and  $W$  is the Additive White Gaussian Noise (AWGN). Here, CP is not considered in Eq. 2 as it represents the effect of CIR on the PN sequence.

We assume a real-valued PN sequence is transmitted. At the receiver, the CP is removed and only the PN sequence part of the received signal is taken for correlation. Due to the circular correlation property, the correlation based estimation is implemented as follows,

$$\hat{H} = \frac{1}{M} \mathbf{S}_t^T Y_r = \frac{1}{M} \mathbf{S}_t^T \mathbf{S}_t h_{r,t} + \frac{1}{M} \mathbf{S}_t^T W \quad (3)$$

For long PN sequences, due to noise mitigation and Eq. 1, the channel estimate can then be given as,

$$\hat{H} \approx h_{r,t} \quad (4)$$

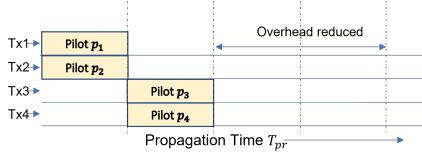


Fig. 3: Multiplexed pilot transmission with  $N_{batch} = 2$ .

Due to sequential transmissions, there will be no interference between transmitters. The total transmitted samples at each receiver are  $P \times N_t$ . If we denote the sampling rate by  $F_s$ , then the propagation time can be denoted as,

$$T_{pr} = \frac{P \times N_t}{F_s} \quad (5)$$

Intuitively, for the sequential transmission case, as  $N_t$  increases, due to the increase in Propagation time, the channel estimation overhead increases. Transmission time can be decreased by reducing PN sequence length. But this results in increase in the estimation error.

### B. Multiplexed Pilot transmission

To reduce the channel estimation overhead, simultaneous transmissions can be performed by multiple transmitters. We assume  $N_{batch}$  transmitters will be sending Pilot sequences simultaneously, where  $1 \leq N_{batch} \leq N_t$ . The received signal, without the CP, at each antenna  $r$  can be given as,

$$Y_r = \mathbf{S}_t h_{r,t} + \sum_{n_i=1}^{N_{batch}} \mathbf{S}_{n_i} h_{r,n_i} + W \quad (6)$$

Where  $\mathbf{S}_{n_i}$  is the circulant matrix representation of the PN sequence, and  $h_{r,n_i}$  is the CIR for transmitting antenna  $n_i$ .

Since multiple transmitters send Pilot sequences simultaneously, for a transmitter  $t$ , all other simultaneously transmitter sequences will be interference. To mitigate this, an option is to use multiple PN sequences. But using multiple PN sequences can become an issue since the number of PN sequences with minimal cross-correlation are very limited. Also, the cross-correlation between PN sequences will not approximate to Eq. 1, which means interference will not be mitigated completely. Instead, we use a single PN sequence with circular shifts for multiple antennas which are transmitting simultaneously.

The assumption made here is that CIR length  $L \ll M$ , which means correlating with only  $L$  circular shifts is required to get the CIR. Circular shifted row multiplications in Eq. 3 beyond the CIR length between matrix  $\mathbf{S}_t$  and  $Y_r$  become redundant as no new information is obtained. This assumption also holds true for the sequential transmission case since CIR estimate with low error can only be obtained with longer PN sequences. So, if we circularly shift the PN sequences beyond the CIR

length, they can be simultaneously transmitted without interference between the sequences.

To design such Pilot sequences, we consider the  $P$  length Pilots used in Sec. II-A. We then batch transmitters depending on how many circularly shifted PN sequences can be obtained. This is determined by the  $C$  length CP. We assume to have information about the maximum CIR length  $L$  and consider  $C = L$ . So, the number of transmitters that can be batched are,

$$N_{batch} \leq \text{floor} \left( \frac{M}{C} \right) \quad (7)$$

The  $N_{batch}$  transmitters are then given a circularly shifted version of the same PN sequence where the circular shift for transmitter  $t$  is given by,

$$C_{shift} = \frac{M}{N_{batch}} \times (t \bmod N_{batch}) \quad (8)$$

After the Pilot sequences are circularly shifted, the CP is added by taking the last  $C$  samples of the circularly shifted PN sequence after which multiplexed transmissions are performed. An illustration of multiplexed transmissions is shown in Fig. 3. The Pilot sequences received can be given as,

$$Y_r^b = [(S_1^p)^T : (S_2^p)^T : \dots : (S_{N_{batch}-1}^p)^T]^T \\ [h_{r,1}^T : h_{r,2}^T : \dots : h_{r,N_{batch}-1}^T]^T + W \quad (9)$$

where  $S_i^p$  is the  $\frac{M}{N_{batch}} \times P$  partial circulant matrix representation of the circularly shifted Pilot sequence. Removing the CP and performing circular correlation, we get,

$$\hat{H}_{batch} = \frac{1}{N} S_t Y_r^b \approx [h_{r,0}^T : h_{r,1}^T : \dots : h_{r,N_{batch}-1}^T] \quad (10)$$

for PN sequence length  $\gg L$ . Here  $\hat{H}_{batch}$  is the CIR estimate consisting of CIRs of all transmitters whose Pilot sequences were sent in the same batch. So, the multiplexed transmission case has a propagation time of,

$$T_{pr}^b = \frac{P}{F_s} \times \frac{N_t}{N_{batch}} \quad (11)$$

which shows a reduction of the Pilot overhead by  $N_{batch}$  when compared with Eq. 5. While the multiplexed transmission case gives overhead reduction, an interesting thing to note is the potential to help mitigate the Pilot Contamination issue that occurs in Multi-User multi-cell Massive MIMO systems. Since only a single PN sequence is being used for all transmitters, other  $M$ -length PN sequences can be used by neighboring cells for channel estimation with mitigated interference.

### III. IMPLEMENTATION

#### A. Setup

Consider the  $N_t \times N_r$  MIMO system from Sec. II. For our implementation scenario we assume both  $N_t$  and  $N_r$  are  $\geq 16$  and an  $L$  length channel with  $L_{nz}$  taps. We simulate the Pilot sequence transmission through such the channel and then reception at the  $N_r$  receiving antennas. We consider the receiver side array connected to a server consisting of CPU and GPU via a switch. Here the CPU is used for control and data access whereas the GPU is used for the bulk of data processing. This type of setup has been used in various testbed deployments such as ORBIT [21] and COSMOS [22] due to the modular structure and flexibility that it provides for implementing various scenarios and testing multiple algorithms. We use this setup present in a Sandbox in the ORBIT Testbed to simulate and implement the channel estimation scenarios.

#### B. Implementation in Tensor Cores

Once the received sequences are sent to the GPUs for processing, we perform the correlation by using the Tensor Cores present in each GPU. For this implementation, we use the Nvidia V100 GPUs which consists of 5120 CUDA Cores and 640 Tensor Cores, 1530 MHz clock and 16 Gb of DRAM [23]. CUDA architecture abstraction in combination with the Tensor Cores is used for programming the correlation in the GPU [24].

Tensor Cores are being included in modern Nvidia (R) GPUs as an alternative micro-architecture to the CUDA cores. Tensor cores are specialized micro-architectures which will perform parallel  $4 \times 4$  matrix-multiply and accumulate operation for half-precision (16-bit) floating point numbers [24]. Since half-precision floats are used, the data which is usually in full-precision complex floats, is converted down to half-precision. This is done in combination with the CP removal before correlation is to be performed in the GPU.

The correlation operation, as shown in Eqs. 3 and 10, can be considered as a matrix multiplication of the circulant matrix representation of the PN sequence and the received signal. So, we can utilize these Tensor Cores present in the GPUs to implement the correlation based channel estimation as a parallel matrix multiplication. Since multiple Tensor Cores exist in the GPUs, each core can take a smaller  $4 \times 4$  tile of the matrix multiplication to be performed, and gives a  $4 \times 4$  multiplication output which is stored in an output matrix.

While Tensor cores can provide fast processing, since they use half-precision arithmetic, the range of numeric representation can become very low. So, the final multiplication output can have increased error due to the rounding errors and early saturation of correlation output. We mitigate these errors occurring due to low

| Parameter                 | Value                                      |
|---------------------------|--|
| MIMO scale                | $16 \times 16, 32 \times 32, 64 \times 64$ |
| PN Sequence Length        | 511, 1023, 2047                            |
| CP length                 | 64 - 128                                   |
| No. of Channel Taps       | 1 - 128                                    |
| $N_{batch}$               | 1, 2, 4, 8                                 |
| Iterations per experiment | 50   |

TABLE I: Parameter values used for various experiments performed

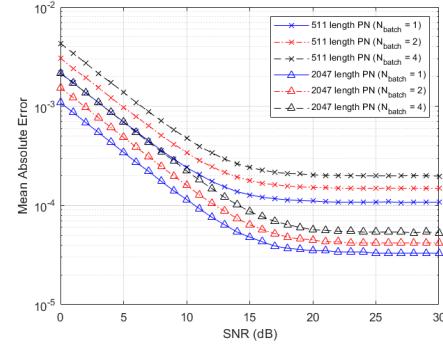


Fig. 4: MAE vs SNR for PN length  $M = 511, 2047$  and  $N_{batch} = 1, 2, 4$ .  $L = 64, L_{nz} = 64, N_r = N_t = 16$

precision computations by performing the normalizing operation i.e.  $(\frac{1}{M})$  in Eq. 10, of the correlation after partial multiplication instead of normalizing after the complete matrix multiplication. This keeps the multiplication output below the saturation point for half-precision computation.

### IV. EXPERIMENTS AND RESULTS

Table I shows parameter values that will be used when performing experiments. The crucial goal is to evaluate the performance of multiplexed PN sequence transmissions, and then investigate the processing latency and error performance when using Tensor Cores to perform channel estimation. We firstly look into the Mean Absolute Error (MAE) performance of the Channel Estimation. We define MAE as follows,

$$E = \frac{1}{LN_t N_r} \sum_{t=1}^{N_t} \sum_{r=1}^{N_r} \sum_{l=1}^L |\hat{h}_{r,t}[l] - h_{r,t}[l]| \quad (12)$$

where  $|\cdot|$  denotes the absolute value.

#### A. Mean Absolute Error vs SNR for different PN sequence lengths

We keep the channel length  $L = 64$  and channel taps  $L_{nz} = 64$ . Fig. 4 shows the MAE vs SNR graph for various PN sequence lengths and  $N_{batch}$  values. The PN sequence length affects the MAE performance significantly due to the correlation of longer PN sequences being close to the Delta function. While lower

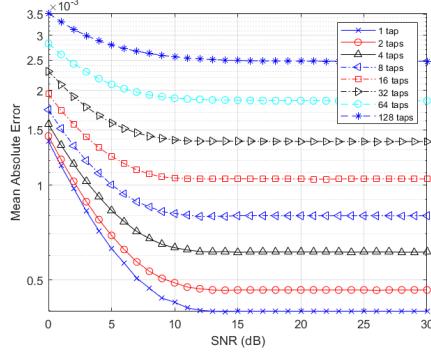


Fig. 5: MAE vs SNR for channel estimation while varying number of channel taps. PN length  $M = 2047$ ,  $N_{batch} = 4$ ,  $L = 128$ ,  $N_t = N_r = 16$

SNR regions affect the MAE, an error floor is seen at higher SNRs which reduces with increase in PN sequence length. The error floor mainly exists due to the  $-\frac{1}{M}$  in Eq. 1. But the Tensor Core implementation also introduces errors due to low precision. Also, variation in the pilot multiplexing can vary the MAE, but the MAE difference remains consistent throughout the SNR range.

#### B. Mean Absolute Error vs SNR for different number of channel taps

To look closely at the error performance due to use of Tensor Cores, we perform an experiment to measure the MAE while varying the SNR and the channel taps for a 2047-length PN sequence. The goal for this experiment is to evaluate the Tensor Core implementation accuracy when performing correlation for varying channel taps. 4 transmitter are multiplexed, the channel length  $L = 128$ , but the number of channel taps within channel length vary from 1 to 128. Fig. 5 shows the effect of varying the channel taps.  $L$  length correlation is still performed by the Tensor Cores, as the number of channel taps change within the channel length, number of near non-zero correlated values will change. This affects the Tensor Core performance since the low precision computation could give increased errors for high number of channel taps. This is seen in Fig. 5, where PN sequence length is unchanged but the MAE is high for the  $L_{nz} = 128$ , and low for small number of  $L_{nz}$ . Here, since the channel length does not change which means correlation length is unchanged. So the processing latency will not be affected.

#### C. Frame Processing Latency vs number of multiplexed antennas for different PN sequence lengths

We then investigate the processing latency for the Tensor core implementation for a one frame. Here, a frame consists of Pilot transmissions for all transmitters

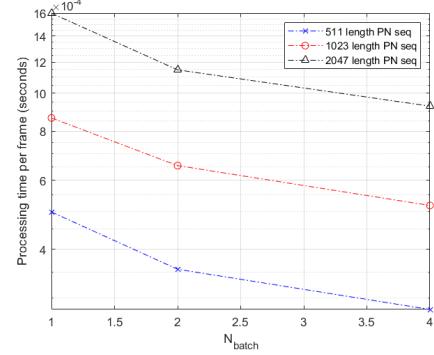


Fig. 6: Processing latency vs  $N_{batch}$  value for various PN sequence lengths.  $L = 64$ ,  $L_{nz} = 64$ ,  $N_t = N_r = 16$

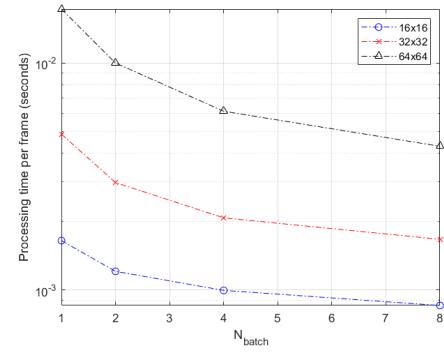


Fig. 7: Processing latency vs  $N_{batch}$  value for variation in Massive MIMO array size.  $M = 2047$ ,  $L = 128$ ,  $L_{nz} = 20$

and its length is  $\frac{P \cdot N_t}{N_{batch}}$ . Correlation is performed for various PN sequence lengths and various  $N_{batch}$  values. For calculation of the processing time, the correlation time as well as the memory transfer time between CPU and GPU is considered. From Fig. 6, it can be seen that the processing latency is heavily affected by the multiplexed pilot transmissions. This is mainly due to decrease in frame size when pilots are multiplexed, leading to decrease in memory transfer time between the CPU and GPU. Also, when we compare the processing latency with the frame propagation time from Eq. 11, the Tensor Core processing time can be lower than or equivalent to the propagation time for sampling rates up to 10 MHz. This makes it possible to use the implementation in real-time Massive MIMO deployments.

#### D. Frame Processing Latency vs number of multiplexed antennas for different MIMO array sizes

We increase the scale of antenna arrays and evaluate the processing latency of the correlation. Fig. 7 shows the variation in Processing time per frame vs the  $N_{batch}$  value for various Massive MIMO scales. We consider

the  $32 \times 32$  and  $64 \times 64$  MIMO systems and compare the Frame processing latency. With no multiplexing, the processing latency varies highly with the antenna array size. But when high multiplexing is performed, the effect reduces. This effect is similarly seen in the PN sequence variation case in Fig. 6, where the decrease in memory transfer time decreases the overall processing latency. This becomes more apparent for larger MIMO scale where the frame size also increases considerably.

## V. CONCLUSION

We implemented a fast and low-overhead channel estimation scheme for Massive MIMO systems in a GPU by exploiting the PN sequence correlation properties and using multiplexed pilot transmissions. To deal with the high complexity of correlation based channel estimation, we implemented the channel estimation by using Tensor Cores in Nvidia GPUs. We then performed experiments for various scenarios to evaluate the error performance and estimation accuracy of the Tensor Core based implementation.

While Tensor Core implementation shows potential, using only single GPU for processing limits the scale at which Massive MIMO systems can be used. To deal with this issue, further investigation into distributed multi-GPU processing is required. Heterogeneous processing architectures which combine CPUs and GPUs for processing should also be investigated.

## REFERENCES

- [1] P. Xu, J. Wang, and J. Wang, "Effect of pilot contamination on channel estimation in massive mimo systems," in *2013 International conference on wireless communications and signal processing*, pp. 1–6, IEEE, 2013.
- [2] C. U. Bas, R. Wang, S. Sangodoyin, D. Psychoudakis, T. Henige, R. Monroe, J. Park, C. J. Zhang, and A. F. Molisch, "Real-time millimeter-wave mimo channel sounder for dynamic directional measurements," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 9, pp. 8775–8789, 2019.
- [3] P. Laly, D. P. Gaillot, G. Delbarre, M. Van den Bossche, G. Vermeeren, F. Challita, E. Tanghe, E. P. Simon, W. Joseph, L. Martens, *et al.*, "Massive radio channel sounder architecture for 5g mobility scenarios: Mamimosa," in *2020 14th European Conference on Antennas and Propagation (EuCAP)*, pp. 1–5, IEEE, 2020.
- [4] Y. Yang, Y. Gui, H. Wang, W. Zhang, Y. Li, X. Yin, and C.-X. Wang, "Parallel channel sounder for mimo channel measurements," *IEEE Wireless Communications*, vol. 25, no. 5, pp. 16–22, 2018.
- [5] Z. Wen, H. Kong, Q. Wang, S. Li, X. Zhao, M. Wang, and S. Sun, "mmwave channel sounder based on cots instruments for 5g and indoor channel measurement," in *2016 IEEE Wireless Communications and Networking Conference*, pp. 1–7, IEEE, 2016.
- [6] W. Fan, A. W. Mbugua, X. Cai, and K. Olesen, "Development and experimental validation of an ultra-wideband channel sounder," in *2019 13th European Conference on Antennas and Propagation (EuCAP)*, pp. 1–5, IEEE, 2019.
- [7] Y. Zhao, C. Tao, B. Ai, *et al.*, "System design and calibration for wideband channel sounding with multiple frequency bands," *IEEE Access*, vol. 5, pp. 781–793, 2017.
- [8] R. Wang, C. U. Bas, Z. Cheng, T. Choi, H. Feng, Z. Li, X. Ye, P. Tang, S. Sangodoyin, J. Gómez-Ponce, *et al.*, "Enabling super-resolution parameter estimation for mm-wave channel sounding," *IEEE Transactions on Wireless Communications*, vol. 19, no. 5, pp. 3077–3090, 2020.
- [9] B. Gokalgandhi, P. Maddala, and I. Seskar, "Implementation of fpga based channel sounder for large scale antenna systems using rfnoe on usrp platform," *Proceedings of the GNU Radio Conference*, vol. 2, no. 1, p. 9, 2017.
- [10] K. Hassan, M. Masarwa, M. Zwingelstein, and I. Dayoub, "Channel estimation techniques for millimeter-wave communication systems: Achievements and challenges," *IEEE Open Journal of the Communications Society*, vol. 1, pp. 1336–1363, 2020.
- [11] M. Benzaghta and K. M. Rabie, "Massive mimo systems for 5g: A systematic mapping study on antenna design challenges and channel estimation open issues," *IET Communications*, 2021.
- [12] B. Hassan, S. Baig, H. M. Asif, S. Mumtaz, and S. Muhamadat, "A survey of fdd-based channel estimation schemes with coordinated multipoint," *IEEE Systems Journal*, pp. 1–11, 2021.
- [13] N. Shaik and P. K. Malik, "A comprehensive survey 5g wireless communication systems: open issues, research challenges, channel estimation, multi carrier modulation and 5g applications," *Multimedia Tools and Applications*, pp. 1–39, 2021.
- [14] S. Chen, J. Zhang, J. Zhang, E. Björnson, and B. Ai, "A survey on user-centric cell-free massive mimo systems," *arXiv preprint arXiv:2104.13667*, 2021.
- [15] B. Gokalgandhi, C. Segerholm, N. Paul, and I. Seskar, "Accelerating channel estimation and demodulation of uplink ofdm symbols for large scale antenna systems using gpu," in *2019 International Conference on Computing, Networking and Communications (ICNC)*, pp. 955–959, IEEE, 2019.
- [16] A. Riadi, M. Boulouird, and M. M. Hassani, "Performance of massive-mimo ofdm system with m-qam modulation based on ls channel estimation," in *2019 International Conference on Advanced Systems and Emergent Technologies (ICASET)*, pp. 74–78, 2019.
- [17] M. Belgiovine, K. Sankhe, C. Bocanegra, D. Roy, and K. R. Chowdhury, "Deep learning at the edge for channel estimation in beyond-5g massive mimo," *IEEE Wireless Communications*, vol. 28, no. 2, pp. 19–25, 2021.
- [18] Y. Jin, J. Zhang, B. Ai, and X. Zhang, "Channel estimation for mmwave massive mimo with convolutional blind denoising network," *IEEE Communications Letters*, vol. 24, no. 1, pp. 95–98, 2019.
- [19] M. Liu, M. Crussiere, and J.-F. Hélar, "Improved channel estimation methods based on pn sequence for tds-ofdm," in *2012 19th International Conference on Telecommunications (ICT)*, pp. 1–5, IEEE, 2012.
- [20] S. Yu and J. W. Lee, "Channel sounding for multi-user massive mimo in distributed antenna system environment," *Electronics*, vol. 8, no. 1, p. 36, 2019.
- [21] D. Raychaudhuri, I. Seskar, M. Ott, S. Ganu, K. Ramachandran, H. Kremo, R. Siracusa, H. Liu, and M. Singh, "Overview of the orbit radio grid testbed for evaluation of next-generation wireless network protocols," in *IEEE Wireless Communications and Networking Conference, 2005*, vol. 3, pp. 1664–1669, IEEE, 2005.
- [22] D. Raychaudhuri, I. Seskar, G. Zussman, T. Korakis, D. Kilper, T. Chen, J. Kolodziejski, M. Sherman, Z. Kostic, X. Gu, *et al.*, "Challenge: Cosmos: A city-scale programmable testbed for experimentation with advanced wireless," in *Proceedings of the 26th Annual International Conference on Mobile Computing and Networking*, pp. 1–13, 2020.
- [23] Nvidia, "Nvidia tesla v100 gpu architecture." <https://images.nvidia.com/content/volta-architecture/pdf/volta-architecture-whitepaper.pdf>.
- [24] S. Markidis, S. W. Der Chien, E. Laure, I. B. Peng, and J. S. Vetter, "Nvidia tensor core programmability, performance & precision," in *2018 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)*, pp. 522–531, IEEE, 2018.