Graph Neural Network Based 77 GHz MIMO Radar Array Processor for Autonomous Robotics

Ransara Wijitharathna*, Pahan Mendis*, Rahal Perera*, Punsara Mahawela*, Nilan Udayanga†,
Chamira U. S. Edussooriya*†‡, Arjuna Madanayake†‡
*Department of Electronic and Telecommunication Engineering, University of Moratuwa, Moratuwa, Sri Lanka.
†Department of Electrical and Computer Engineering, Florida International University, Miami, USA.
‡Deep-Si Technologies, Colombo, Sri Lanka.
Emails: chamira@uom.lk, amadanay@fiu.edu

***Abstract*—Frequency-modulated continuous-wave (FMCW) multiple-input multiple-output (MIMO) long-range radars currently employed for autonomous robotics have limited maximum range. By employing transmit beamforming and beam scanning, the range can be increased. However, the beam scanning time reduces the achievable velocity resolution. In this paper, we propose an FMCW MIMO radar, operating at 77 GHz, with transmit (TX) beamforming with subarrays to increase the range. We employ four TX subarrays, each having three antennas, with analog beamforming. Compared to TX beamforming as one TX array, our approach provides sufficiently wide beams alleviating beam scanning, further, without substantially reducing the virtual elements of the MIMO radar. We implement the radar signal processing pipeline on a ZYNQ Ultrascale+ ZCU106 FPGA to achieve real-time processing. Furthermore, we employ a graph neural network to detect objects using the radar point cloud. Preliminary results are presented to confirm the operation of the proposed FMCW MIMO radar.**

***Index Terms*—FMCW radar, MIMO radar, beamforming, autonomous robotics, graph neural networks.**

I. Introduction

A robot can only act following sensing and perception of its environment. Autonomous robotics, therefore, require a variety of sensing modalities to ensure complete situational awareness [1], [2]. The expectations for situational awareness are especially pronounced in the case of autonomous vehicles, where life and limb are at stake. In fact, rapid automation of driving functions has stimulated research into advanced driver assistance systems. Reliable and resilient sensory perception dictates the adoption of multi-modal sensing. A robotic vehicle with advanced driver assistance systems needs to employ a clever mix of radars, lidars, ultrasonics, and cameras to sense, fuse, and then perceive its operational environment [3]–[5].

Multi-modal sensor technologies have their pros and cons. For example, the high pixel resolution of a camera provides the best representation of the surroundings for further processing through machine learning. However, the calculation of distances to the surrounding objects is crucially important in autonomous vehicles, which is quite tedious for video signals. Lidar, on the other hand, uses time of flight measurements of laser pulses, which in turn provides a reasonably high resolution and accuracy without demanding high computational complexity from the associated edge processor [6], [7]. The problem with both video and lidar technologies is the lack of robustness, as the optical sensing involved is prone to weather conditions and physical blockages. In contrast, radar can be designed to provide accurate measurements of distance, velocity, and angle of the objects regardless of the weather conditions and is not as easily blocked. For example, radar penetrates dielectric obstacles such as wooden sheets, vegetation, plastic signs better than lidar [8], [9]. Nevertheless, radar can be constrained in range and furnish unacceptably low resolution, especially at long sensing distances.

Current use cases of radar in automotive applications can be classified according to their ranging capability. Long-range radar is used for automotive cruise control; medium-range radar is used for lane-change assist, cross-traffic alert, blind-sport detection and rear-collision warning; and short-range radar is used for park assist, obstacle detection and pre-crash detection [10]. Continuous-wave radar has been the mostly used type of radar in automotive applications due to the high implementation complexity of classical pulsed radar. Frequency shift keying and linear frequency modulated waveforms are the most appropriate waveforms that can be employed for continuous-wave radars [11]. Linear frequency-modulated continuous-wave (FMCW) radar is the most frequently used radar [12] in autonomous robotics.

Despite wide usage, FMCW radars are constrained in several features. For example, a finer range resolution essentially requires a larger bandwidth, leading to the use of a more spacious spectrum such as 77 GHz Ka-band. Moreover, a higher range requires high transmission power; the limits of power amplifiers at such high frequencies has led to the exploration of novel directive antennas and beamforming techniques to reach high ranges that are required for fully autonomous driving applications [13]–[17]. The use of such techniques causes the field-of-view (FoV) to narrow down and the angular resolution of the radar to decrease. On one hand, to address the narrower FoV, beam-scanning can be employed. On the other hand, a long scanning time will degrade the velocity resolution of the radar. Furthermore, the angular resolution degradation is due to the decrease in the effective number of transmitting and receiving antenna elements resulting from beamforming. A key requirement for detection and identification of objects in autonomous robotics

is a dense point cloud of the radar's perceptive area. Multiple-input multiple-output (MIMO) processing techniques improve the angular resolution of the radar, creating a much larger virtual array of antennas. However, such a high-resolution radar has a significant memory and computational complexity, which makes real-time processing challenging.

We propose a long-range FMCW MIMO radar operating at 77 GHz for autonomous robotics in this paper. Instead of transmit (TX) uniform linear array (ULA), we employed four TX subarrays, each having three antenna elements and generating an independent beam using analog beamforming. Our approach increases the range of the radar without deteriorating the FoV due to sufficiently wide beams. Furthermore, our approach does not reduce the virtual elements of the MIMO radar substantially, leading to an angular resolution of 1.8° with a 16-element receive array. We implement the radar signal processing pipeline on a ZYNQ Ultrascale+ ZCU106 FPGA in order to achieve real-time processing. The angle of arrival (AoA) estimation is performed in a host computer after the processing at the FPGA. We then generate a radar point cloud and employ a graph neural network (GNN) to detect the objects. Preliminary simulation and experimental results confirm the operation of the proposed FMCW MIMO radar.

II. Proposed MIMO Radar Array Processor

We present the proposed FMCW MIMO radar processor in this section. The overall architecture consists of the radar front end, FPGA, and a host computer for visualization and post processing as shown in Fig. 1. We select the specifications of the FMCW MIMO radar as: maximum range = 300 m, range resolution = 3.5 m, maximum velocity = 100 km/h, velocity resolution= 1.6 km/h, and bandwidth = 86.77 MHz. The TX array is designed to generate $N_{tx} = 4$ transmit beams, each using a three-element subarray with analog beamforming whereas the receive (RX) array contains $N_{rx} = 16$ antenna elements as shown in Fig. 2. The RX array and four TX subarrays are configured as ULAs, and the antennas are designed as following the reference design of Texas Instruments series fed antenna [18]. Phase shift for all the three elements in a subarray is fixed so that each subarray acts as a single element with a narrower 3 dB beamwidth and higher gain compared to a single element. TX and RX arrays result a $N_{tx} \times N_{rx} = 64$ element virtual ULA, i.e., MIMO radar, when used with time division multiplexing. For the virtual ULA, the boresight angular resolution $\Delta\theta$, for a $\lambda/2$-spaced array, is

$$\Delta\theta = \frac{360^\circ}{\pi \times N_{tx} \times N_{rx}} = 1.8^\circ.$$

The radar signal processing chain is predominately implemented on an FPGA, and the object detection using a radar point cloud is achieved through a GNN. The FPGA architecture and object detection through a GNN (with a brief review) are presented in the next two subsections, respectively.

A. Radar Signal Processing on an FPGA

The architecture of the radar signal processing pipeline is shown in Fig. 3. The pipeline is designed to handle a continuous stream of data in real-time. After the analog-to-digital converters (ADCs), double buffers are placed at the first stage to handle the data rate mismatches between the incoming data stream and the pipeline. They can buffer data up to one

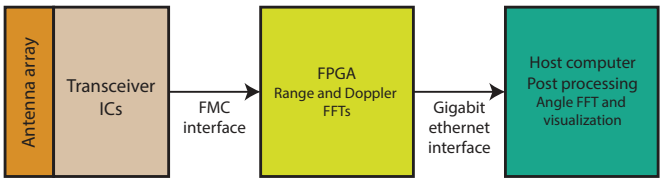

Fig. 1. High level overview of the architecture of the system : Antenna array and the transceiver ICs are on a single PCB as an FPGA Mezzanine Card. Range and Doppler FFTs are performed on the FPGA. A built-in Gigabit Ethernet interface is used to stream the data to a host computer for further processing.

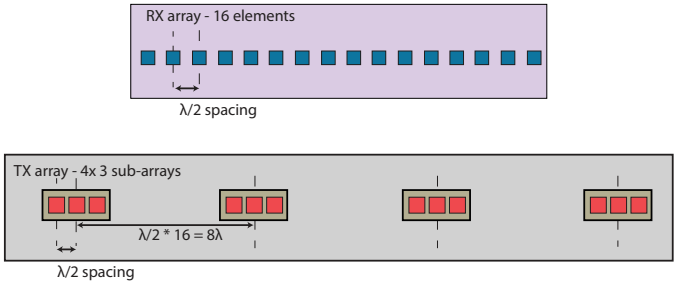

Fig. 2. Detailed antenna configuration: Each of the subarrays will create a single transmit beam that can be operated in time-division multiplexing. The resulting virtual array will have 4 x 16 virtual antenna elements spaced at 0.5λ at 77 GHz.

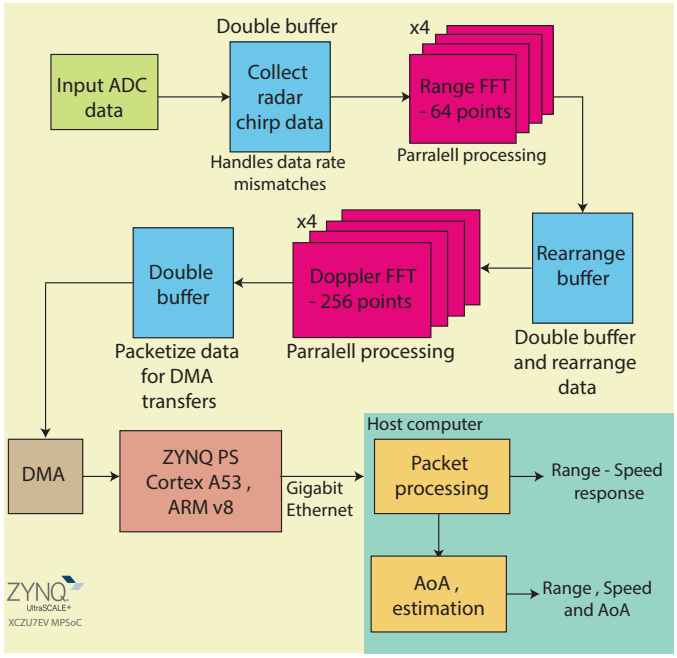

Fig. 3. The real-time radar processing pipeline: The double buffers are implemented on either BRAM or URAM, Xilinx LOGICORE^{TM} FFT IP in radix-4 butterfly architecture is used for FFTs. At the current stage, an inbuilt Gigabit Ethernet MAC is utilized to stream data to a host computer, and a Direct Memory Access controller is used to communicate data between PL and PS via DDR4 memory.

chirp in each receiver channel in each of the buffers. Next, the samples are streamed into the range discrete Fourier transform, implemented as a fast Fourier transform (FFT). An instance of Xilinx FFT IP core is utilized in 4-channel parallel Radix-4 butterfly configuration. The output samples are then buffered again. In contrast to the first buffer, this stores samples of chirps of a complete radar frame and outputs the frames in the dimension of chirps, which is required by the Doppler FFT processing. The Doppler FFT is also implemented in a manner identical to the range FFT with the exception of the FFT length. The next double buffer serves two purposes: rate handling between the processing chain and the direct memory access (DMA) engine and packetizing the data for DMA transfers. The DMA engine uses an AXI4-Stream interface to read data from the programmable logic (PL) to dynamic read access memory (DRAM). AXI4-Stream TLAST signal is used to indicate the end of a DMA packet transfer. The double buffer generates the TLAST signal according to a preset packet length. The samples of each of the four channels are organized in an interleaved manner to fit into the 128-bit AXI4-Stream bus.

One of the built-in Gigabit Ethernet controllers is used with Xilinx Light Weight Internet Protocol stack to relay the data to a host computer for further processing and visualization using Python. First, the received packets are processed to rearrange the samples from each virtual element and create the radar cube. The range-speed response of the first element in the RX array is employed to differentiate object areas using a threshold. Next, the range-speed responses of all the sixteen RX channels are coherently summed to calculate the angle FFT using the `numpy FFT` library function. We generate the point cloud by selecting the bins of the three-dimensional range-speed-angle FFT data cube that have magnitudes greater than the threshold. Here, we do not apply a constant false alarm rate for preliminary simulations and experiments. The point cloud consisting of the detected objects is then sent to a GNN running on the host computer.

B. Object Detection using a GNN

The approaches employed to detect objects within point clouds using deep neural networks can be divided into three major groups: point-based, grid-based, and graph-based methods. Point-based methods work directly with the input point clouds and do not require any prior data transformations. As a result, they retain all the structural information of the point cloud. However, these approaches do not account for the specific relationships between individual points, even though they can still consider the structure of local groups to some extent. Grid-based approaches involve converting point clouds into a structured grid representation using a process called discretization or voxelization. This allows the application of convolutional neural networks for various computer vision tasks. Nevertheless, this creates a sparse representation of data due to the loss of information during the initial data transformation. Graph-based techniques convert the input point cloud into a graph representation for processing and can be classified into three types: convolutional, attentional, and message-passing neural networks. In GNNs, each point serves as a node in the graph, without losing any structural information, and models the relationships among them as edges in the graph.

In radar point cloud processing, the sparsity of the point clouds remains challenging. Thus, using a GNN is beneficial as it uses not only the point features but also the relationships among them. In the graph construction, detected points are used as nodes of the graph, preserving structural information of the point cloud, and modeling the relationships among them as edges in the graph.

We employ RadarGNN [19] to perform multi-class object detection on the radar point cloud. The RadarGNN consists of four steps: data preparation, graph construction, the GNN, and the detection heads. The GNN used in RadarGNN is based on the pointGNN architecture [20], originally proposed for lidar point cloud processing. The point GNN architecture is comprised of two main parts and generates a comprehensive feature representation for the connected detection heads. The initial feature embedding creates a high-dimensional non-contextual feature representation from the low-dimensional node and edge features. To achieve this, a shared multi-layer perceptron with four layers is used for node feature embedding, while another multi-layer perceptron with three layers is used for edge feature embedding. The detection heads are constructed using a shared multi-layer perceptron that comprises two consecutive layers with a linear activation function, providing flexibility in the output space. This module is responsible for predicting bounding boxes for each point in a given point cloud.

We note that the nodes in the graph are characterized by their absolute spatial coordinates (x, y), velocity vectors (v_x, v_y), and radar cross-section. The edge generation process encodes the relationships between points by creating edges and incorporating edge features. The open-source RadarScenes dataset [21] is used for training the GNN.

III. Simulation and Experimental Results

In this section, we present the preliminary simulation results for the antenna arrays and experimental results for the proposed MIMO radar array processor.

A. Radiation Pattern of TX Subarrays

We first consider the beampattern generated by a three-element subarray with analog beamforming. Note that the antennas are designed following the reference design of the Texas Instruments series-fed antenna [18]. The three-element subarray is simulated on Ansys HFSS software, and the radiation pattern on the azimuth plane is shown in Fig. 4, where the beam is directed towards to the broadside direction. Note that the maximum gain is increased from 10.87 dB [18] to 14.92 dB due to beamforming, therefore increasing the range of the radar. Furthermore, we note that the beam width is wide enough to cover the full FoV of a long-range radar, typically $\pm 15^{\circ}$ [12].

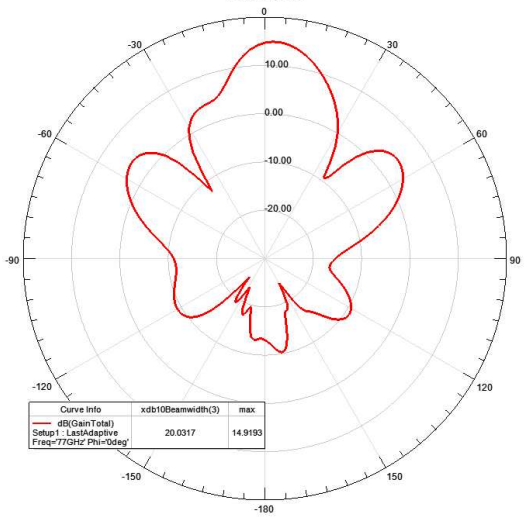

Fig. 4. Radiation pattern of a TX three-element subarray on the azimuth plane. The maximum gain is increased from 10.87 dB [18] to 14.92 dB due to beamforming; simulated at 77 GHz.

TABLE I
OBJECTS IN THE SYNTHETIC RADAR SCENE

Vehicle	Radar Cross Section (m^2)	Velocity (kmh^{-1})	Azimuth Angle	Distance (m)
1	10	50	30°	100
2	40	96	10°	50
3	20	-80	−10°	40
4	10	25	−30°	20

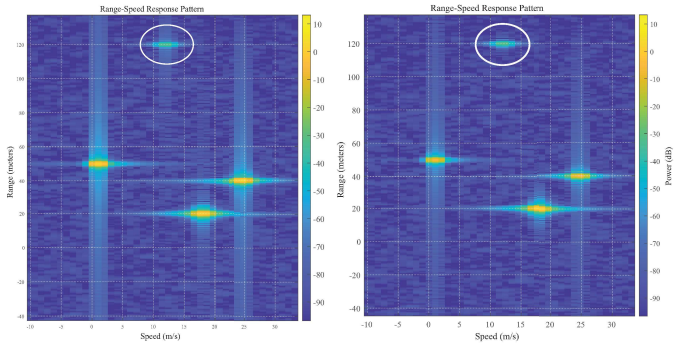

Fig. 5. On left - MATLAB simulation results of the radar consisting the new antenna configuration, Tx analog beamforming; on right same system without Tx analog beamforming, units are in dB. A Hanning window is used to limit spectral leakage.

B. Implementation of the Radar Signal Processing Pipeline

We partially implement the radar signal processing pipeline on a ZYNQ Ultrascale+ ZCU106 development kit only using the first four virtual elements. The FPGA design is set to operate at 100 MHz with a 64-point range and 256-point Doppler FFT lengths as presented in Sec II-A. In radix-4 butterfly architecture, the Xilinx FFT IP requires 231 and 837 clock cycles for 64- and 256-point FFTs, respectively. The IP core scaling settings are set to avoid any overflows during calculations. The proposed MIMO radar system has a maximum beat frequency of ∼17.4 MHz, which should be quadrature sampled at a frequency of around $17.4/0.8 = 19.33$ MSPS [22]. This will generate bursts of data up to 19.33 MSPS $\times 2 \times 16$ bits = 618.56 Mbps per receive channel. However, the available buffer sizes on transceiver ICs, number of multiplexing TX beams, intervals between bursts of chirps, buffer and processing latency in the digital design on the FPGA, and the software performance running on the ZYNQ PS (because we use an integrated Gigabit controller connected to the ZYNQ PS.) can affect this maximum data rate. Nevertheless, for the complete system with 4 TX beams and 16 receiving channels, an interface with a much larger throughput, such as PCIe, should be used. Currently, implementation is limited due to the 1 Gbps Gigabit Ethernet interface, and the implementation performs at around ∼766 Mbps and 3.79 fps.

TABLE II
FPGA RESOURCE UTILIZATION : ZYNQ ULTRASCALE+ XCZU7EV-2FFVC1156 MPSOC

Resource	Utilization	Utilization %
LUT	20029	8.69
LUTRAM	4406	4.33
FF	26584	5.77
BRAM	**62.50**	**20.03**
URAM	**16**	**16.67**
DSP	96	5.56
BUFG	4	0.74

The FPGA resource utilization of the system is presented in Table II. The main concern is on the Ultra RAM - URAM and Block RAM - BRAM utilization. The memory available as BRAM is not sufficient to implement all the buffers in BRAMs; the Ultrascale specific Ultra - RAM has to be used to store all the buffers on the PL with the minimum possible access latency. The buffer size requirement will grow significantly with the FFT sizes and the number of receive channels. External DDR memory connected to the PL needs to be utilized to store data to accommodate the buffers of the complete proposed system.

C. Implemented RADAR Processing Chain Results

We employ MATLAB Phased Array Toolbox to generate synthetic data to verify the operation of the array processor. The setup consists of four vehicles with the parameters described in Table I. For simplicity, the antenna elements were chosen to be Cosine antennas, a phased.ReplicatedSubarray object was used to create the transmit array as given in Fig. 2. In order to verify the performance, we implemented an FMCW MIMO radar without Tx beamforming with the same specifications [23]. The simulation results are presented in Fig. 5. We observe a noticeable improvement in the proposed radar with TX beamforming compared to the FMCW MIMO radar without TX beamforming.

The range speed response of the first channel of the hardware processed radar is shown in Fig. 6. The brighter areas correspond to detected targets, except for the spillover due to the spectral leakage. Furthermore, the estimated ranges and

TABLE III
ESTIMATED RANGE AND SPEED ACHIEVED WITH THE PROPOSED FMCW MIMO RADAR AND THE ACTUAL RANGE AND SPEED.

Vehicle	Actual range (m)	HW output range (m)	Actual speed (m/s)	HW output speed (m/s)
1	100	114.29	13.88	10.19
2	50	57.14	1.11	0
3	40	42.85	50	50.93
4	20	28.57	20.83	20.37

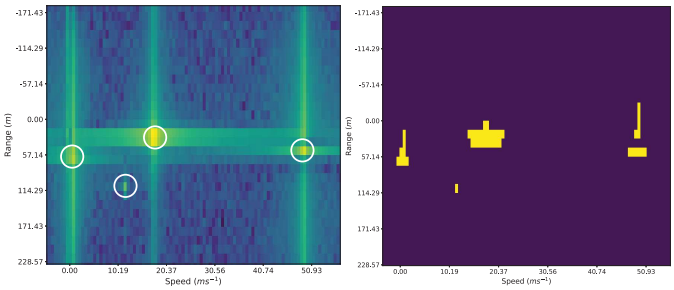

Fig. 6. On left : Range speed response of the first virtual element, hardware processed. On right : thresholded image to detect the object indices for angle FFT.

velocities are presented in Table III. The estimated ranges and speeds are fairly close to the actual values, and the error is mainly due to the fact that the resolution of the implemented radar on the FPGA is low ($= 28.64°$) because our preliminary experiments consider only four virtual elements out of 64 virtual elements.

IV. CONCLUSION AND FUTURE WORK

We propose a long-range FMCW MIMO radar operating at 77 GHz for autonomous robotics. We employ four TX subarrays, each having three antenna elements, to generate independent beams with analog beamforming. Our approach increases the range of the radar without deteriorating the FoV and does not reduce the virtual elements of the MIMO radar substantially. We partially implement the radar signal processing pipeline on a ZYNQ Ultrascale+ ZCU106 FPGA. The preliminary simulation and experimental results confirm the operation of the proposed FMCW MIMO radar. Future work includes the extension of the partial implementation to the full system implementation on an FPGA and experimental validation of the full system.

ACKNOWLEDGMENT

Arjuna Madanayake thanks NSF 2229471, 2052764, 2216332, 2329012 for financial support.

REFERENCES

[1] F. Fahimi, *Autonomous Robots*. New York, NY: Springer, 2009.
[2] W. Schwarting, J. Alonso-Mora, and D. Rus, "Planning and decision-making for autonomous vehicles," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 1, pp. 187–210, Jan. 2018.
[3] M. Mahlisch, R. Hering, W. Ritter, and K. Dietmayer, "Heterogeneous fusion of video, lidar and ESP data for automotive ACC vehicle tracking," in *IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems*, 2006, pp. 139–144.
[4] B. Steux, C. Laurgeau, L. Salesse, and D. Wautier, "Fade: a vehicle detection and tracking system featuring monocular color vision and radar data fusion," in *Intelligent Vehicle Symposium*, vol. 2, 2002, pp. 632–639 vol.2.
[5] J. Steinbaeck, C. Steger, G. Holweg, and N. Druml, "Next generation radar sensors in automotive sensor fusion systems," in *Sensor Data Fusion: Trends, Solutions, Applications*, 2017, pp. 1–6.
[6] Y. Li and J. Ibanez-Guzman, "Lidar for autonomous driving: The principles, challenges, and trends for automotive lidar and perception systems," *IEEE Signal Processing Magazine*, vol. 37, no. 4, pp. 50–61, Jul. 2020.
[7] R. Roriz, J. Cabral, and T. Gomes, "Automotive lidar technology: A survey," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 7, pp. 6282–6297, Jul. 2022.
[8] W. Jones, "Keeping cars from crashing," *IEEE Spectrum*, vol. 38, no. 9, pp. 40–45, Sep. 2001.
[9] R. H. Rasshofer and K. Gresser, "Automotive radar and lidar systems for next generation driver assistance functions," *Advances in Radio Science*, vol. 3, no. 10, pp. 205–209, May 2005.
[10] J. Hasch, E. Topak, R. Schnabel, T. Zwick, R. Weigel, and C. Waldschmidt, "Millimeter-wave technology for automotive radar sensors in the 77 GHz frequency band," *IEEE Transactions on Microwave Theory and Techniques*, vol. 60, no. 3, pp. 845–860, Mar. 2012.
[11] H. Rohling and C. Moller, "Radar waveform for automotive radar systems and applications," in *IEEE Radar Conference*, 2008, pp. 1–4.
[12] S. M. Patole, M. Torlak, D. Wang, and M. Ali, "Automotive radars: A review of signal processing techniques," *IEEE Signal Processing Magazine*, vol. 34, no. 2, pp. 22–35, Mar. 2017.
[13] B.-H. Ku, P. Schmalenberg, O. Inac, O. D. Gurbuz, J. S. Lee, K. Shiozaki, and G. M. Rebeiz, "A 77–81-GHz 16-element phased-array receiver with $\pm 50°$ beam scanning for advanced automotive radars," *IEEE Transactions on Microwave Theory and Techniques*, vol. 62, no. 11, pp. 2823–2832, Nov. 2014.
[14] Y. Liu, G. Bai, and M. C. Yagoub, "A 79 GHz series fed microstrip patch antenna array with bandwidth enhancement and sidelobe suppression," in *International Conference on Radar, Antenna, Microwave, Electronics, and Telecommunications*, 2020, pp. 155–158.
[15] I. E. Lager, C. Trampuz, M. Simeoni, and L. P. Ligthart, "Interleaved array antennas for FMCW radar applications," *IEEE Transactions on Antennas and Propagation*, vol. 57, no. 8, pp. 2486–2490, Aug. 2009.
[16] K. Wu, Y. Yao, X. Cheng, J. Yu, and X. Chen, "Design of high efficiency linearly polarized 8 × 8 millimeter-wave antenna array," in *IEEE Asia-Pacific Microwave Conference*, 2019, pp. 735–737.
[17] J. Zhang, Y. Su, R. Cao, X. Tao, Y. Zhang, and X. Qi, "Design of 94 GHz series-fed microstrip antenna array," in *International Conference on Electromagnetics in Advanced Applications*, 2022, pp. 225–227.
[18] Texas Instruments, "AWRx cascaded radar RF evaluation module (MMWCAS-RF-EVM)," Sep. 2019.
[19] F. Fent, P. Bauerschmidt, and M. Lienkamp, "RadarGNN: Transformation invariant graph neural network for radar-based perception," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 182–191.
[20] W. Shi and R. R. Rajkumar, "Point-GNN: Graph neural network for 3D object detection in a point cloud," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 1711–1719.
[21] O. Schumann, M. Hahn, N. Scheiner, F. Weishaupt, J. F. Tilly, J. Dickmann, and C. Wöhler, "RadarScenes: A real-world radar point cloud data set for automotive applications," in *IEEE International Conference on Information Fusion*. IEEE, 2021, pp. 1–8.
[22] V. Dham, "Programming chirp parameters in TI radar devices," Texas Instruments Incorporated, Tech. Rep., May 2017.
[23] MathWorks Inc., "Increasing angular resolution with virtual arrays." [Online]. Available: https://in.mathworks.com/help/phased/ug/increasing-angular-resolution-with-mimo-radars.html