Dynamic Indoor mmWave MIMO Radar Simulation: An Image Rendering-Based Approach

Yuan Liu, *Student Member, IEEE,* Moein Ahmadi, *Member, IEEE,* Johann Fuchs, Mohammad Alaee-Kerahroodi, *Senior Member, IEEE,* M. R. Bhavani Shankar, *Senior Member, IEEE*

***Abstract*—Radar simulation offers the potential to generate data cubes with effectiveness and accuracy. However, the radar simulator considering both dense multipaths of indoor environment and dynamic motions is rare. This paper develops a versatile channel simulator that can generate frequency modulation continuous wave (FMCW) waveform multiple inputs multiple outputs (MIMO) signals of indoor scenarios. In the proposed simulation framework, the open-source tool called Blender is utilized to model dynamic scenarios and render animations. Each frame of the animation is rendered into a picture with a defined number of pixels by the ray tracing (RT) engine embedded in Blender. Then, the distance, strength, angle of arrival (AoA), and motion speed of each pixel are calculated based on the traced rays. The sampled beat signal models of the commonly used orthogonal multiplexing MIMO modes are derived in terms of the Blender outputs. A virtual array generation method is utilized to improve the simulation time efficiency. To eliminate the velocity noise, a pixel filtering method is also introduced. For validation, the measurements of the time division multiplexing (TDM) FMCW MIMO sensor in both anechoic chamber and corridor scenario at the mmWave band are compared with the simulation using range, angle, Doppler, and micro-Doppler results. The comparisons show the merits and validation of the proposed simulation method.**

***Index Terms*—Blender, channel simulation, FMCW radar, indoor pedestrian, ray tracing.**

I. Introduction

RADIO frequency (RF) based sensing systems have long been used for aircraft monitoring, meteorological radar, and synthetic aperture radar (SAR) imaging [1]–[3]. With the use of millimeter wave (mmWave) and terahertz (THz) frequency bands, radar systems have started to receive significant attention in indoor applications, such as in-cabin monitoring, occupancy sensing, gesture recognition, smart factory, healthcare, and home robotics [4]–[8]. The reasons are twofold. Firstly, wide bandwidth can provide sufficient resolution for applications in indoor scenarios [9]–[11]. Secondly, radio-based sensors are believed to be better than camera-based ones for protecting privacy and robustness [12].

To design and optimize wide-band indoor RF systems, the knowledge of channel models is essential [13]. However, the channel characterizations of indoor radar systems are different from the outdoors for two main reasons. Firstly, because of the dense multipath in indoor environments, the influence of clutter is more severe than in outdoors [14]. Secondly, with the rise of data-driven-based applications, e.g., human motion detection and gesture classification, the modeling of Doppler and micro-Doppler in dynamic scenarios is becoming a new requirement [15]. Furthermore, the emerging integrated sensing and communication (ISAC) technology will be a key vertical in the coming B5G communications era. As a result, an understanding of the radar channel features, although not limited to indoor, is being sought by a wider audience [16].

Conventionally, the channel characteristics can be obtained by field measurements and simulations. However, measurements can be time-consuming and expensive, especially in high-frequency bands. Consequently, efficient channel simulation techniques offer a compromise [17]. The geometry-based simulations are popular for the balance of accuracy and time consumption. Some of them include the deterministic ray tracing (RT) [18], [19], stochastic propagation graph [20], [21], and the hybrid semi-deterministic methods [22], [23]. They make use of optical-ray-based geometrical principles to mimic electromagnetic wave propagation. Those channel simulation methods have long been used and validated in wireless communications [24], [25] and source localization [26], [27]. However, conventional simulators concentrate on wireless parameters and lack the emulation of radar characteristics like Doppler and micro-Doppler. Because communication systems have a preferential focus on large-scale coverage and channel coherent time, therefore wireless communication channel simulators usually simulate discrete positions of the users' track [28]–[30], while cannot satisfy the target-centric radar requirements in some emerging research trends, e.g., the abovementioned ISAC and radio-based AI applications.

In this paper, we aim to provide a radar simulator capable of generating both the multipath of indoor environments and the dynamic Doppler and micro-Doppler due to non-point target motion.

A. Related works of radar simulation

To achieve Doppler changes due to realistic motion, [34] and [35] utilize field measurement and computer vision, respectively, to record skeleton velocities of typical human motions and set up motion-capturing (MoCap) databases. Those MoCap databases can be used to generate synthetic radar signals. However, MoCap databases lack consideration of the environment, while the multipath due to the environment cannot be omitted for indoor applications, especially considering realistic radio propagation.

Y. Liu, M. Ahmadi, J. Fuchs, M. Alaee-Kerahroodi, and B. Shankar are with the Interdisciplinary Centre for Security, Reliability and Trust (SnT), University of Luxembourg, Luxembourg (e-mail: {yuan.liu, moein.ahmadi, johann.fuchs, mohammad.alaee, bhavani.shankar}@uni.lu). They are supported by the Luxembourg National Research Fund (FNR) through the BRIDGES project MASTERS under grant BRIDGES2020/IS/15407066.

TABLE I
A BRIEF COMPARISON OF THE PROPOSED WORK WITH THE CHANNEL MODELING METHODS IN LITERATURE

RT engine	Reference index	Indoor multipaths	Radar waveform	MIMO	Micro-Doppler	Validation via measurement
Conventional RT algorithm	Ref [18], [19], [24]	✓		✓		✓
Optix	Ref [31]		✓	✓		✓
Blender	Ref [32]	✓				
	Ref [33]					
	This work	✓	✓	✓	✓	✓

To address this, some recent studies try to utilize animation software, e.g., Blender [32], [36], [37] and Optix [31], to model the environment. As a follow-up, [33] enhances the Blender-based models by including the multi-bounce rays in the simulation, i.e., the multipaths via more than one-time reflections/ scattering by the target and environment. However, gaps still exist in the state of the art. Many works are motivated by the automatic driving industry and focus on outdoor scenes, e.g., the OptiX-based ones [31] and FEKO [38]. Thus, they simulate fewer multipath. Further, most of them fall short of generating micro-Doppler due to gesture motions of the targets [31]–[33], [36].

B. Contributions of this paper

In this paper, an image rendering-based mmWave multiple inputs and multiple outputs (MIMO) radar simulator is developed for indoor applications built on Blender. By importing the MoCap database and using the animation tools, we can capture even the slight changes in propagation paths due to the gestures and motions of the target as well as the interactions of the environment. A summary of the differences between the existing literature and the proposed work is summarized in Table I. The detailed contributions of this work include:

- Using Blender extension to import the realistic human motions in AMASS [39] and rendering propagation paths containing both velocity and environment effects.
- Identification of appropriate Blender outputs and subsequent mapping of the frame rates of Blender animation with the fast time and slow time of frequency modulation continuous wave (FMCW) waveform. The sampled beat-signal models of the commonly used orthogonal MIMO modes are derived based on the RT outputs of each frame, e.g., time division multiplexing (TDM), code division multiplexing (CDM), and frequency division multiplexing (FDM).
- Applying a virtual array generation method to accelerate MIMO antenna simulation based on the array geometry, the field of view (FoV) range settings, and the FoV resolution.
- Validating via field measurement of pedestrian scenarios in both anechoic chamber and office corridor using mmWave band TDM-MIMO sensors. The important radar images, i.e., the range-angle map (RAM), range-Doppler map (RDM), and the continuous time-Doppler velocity are measured, simulated, and compared. Besides, the clear observation of the micro-Doppler phenomenon due to the swings of arms and legs is presented in the simulation.
- This study can serve as a guide in generating dynamic digital maps for wireless channel simulations. For RT simulation frameworks like [36], the number of propagation paths may change in different frames, therefore it is challenging to identify the velocity of each path. The proposed method exploits the fixed number of pixels in each image to generate the same number of paths in each frame and calculate velocity due to target motions, i.e., the velocities are obtained by differentiating the distances of each pixel. Since the work does not assume identical paths, the calculated velocity is perturbed due to pixel migrations. Further, we use a velocity filtering method to reduce the noise due to pixel migrations.

The rest of the paper is organized as follows. Section II introduces the simulation chain and the outputs of Blender. Section III derives the sampled beat signal models and maps with Blender outputs. Section IV gives the measurement-based validation and Section V concludes the paper.

The following notations are used throughout this paper: lower-case (upper-case) bold characters denote vectors (matrices), particularly, $\mathbf{A} \in \mathbb{R}^{N_1 \times N_2}$ and $\mathbf{A} \in \mathbb{C}^{N_1 \times N_2}$ respectively denote the real and complex matrix of size $N_1 \times N_2$, and $[\mathbf{A}]_{n_1,n_2} \triangleq A_{n_1,n_2}$ denotes the (n_1, n_2)-th entry of the matrix, $\mathbf{A}[n_1, :]$ denotes the n_1th row of the matrix, and $\mathbf{A}[:, n_2]$ denotes the n_2th column of the matrix. $(\cdot)^T$ and $(\cdot)^H$ denote the transpose and the conjugate transpose of a matrix or vector, respectively. The symbol mod$(\cdot)$ refers to a remainder operation.

II. IMAGE RENDERING-BASED RADAR SIMULATION CHAIN

Fig. 1 illustrates the process considered in this paper to generate FMCW signals that will be detailed in the sequel. However, a cursory look indicates the two-step methodology wherein, an effective depiction of the scenario using the critical target parameters, i.e., range, angle, Doppler, and reflectivity is first carried out using image rendering and these parameters are subsequently used to generate FMCW signals using a model-based approach. The different steps in image rendering are elaborated below; the next section deals with the generation of radar signals.

A. Rendering methodology

1) Scenario modeling using Blender: The first step of scenario modeling requires emulating the objects in the real scene

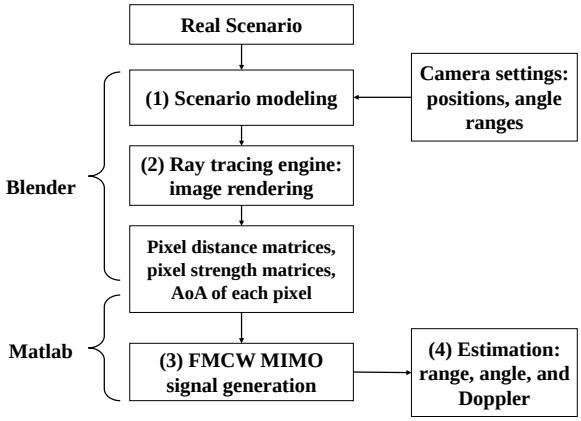


Fig. 1. FMCW Radar channel simulation outline.

in software by appropriately defining the material properties, their geometric size, orientation, and motion [40]. Blender is an open-source animation tool [41] offering flexibility in incorporating such scenes using native animation tools or through well-defined interfaces. In addition to the pre-set scene, the *view settings* define the perspective. The light source and camera of Blender are regarded as the transmitter (Tx) and receiver (Rx), respectively, where the FoV is regarded as the ideal Rx antenna pattern. Here, the Rx is used as an idealized directive antenna with unit gain throughout the FoV and no reception outside the FoV (i,e, without sidelobes outside the FoV). In this paper, we deal with monostatic radars, and hence the light source and camera are co-located. However, if the positions of the light source and camera are separated, the method can be extended to simulate bistatic systems.

2) Image rendering using RT: The operation of the second step is image rendering, where the dynamic 3D scene is converted to sequential 2D images. Each image consists of a certain number of pixels and each pixel can be regarded as a point object. The RT embedded in Blender is utilized to trace the propagation paths among Tx, Rx, and each point object, from which the desired parameters like the spatial, angular, and Doppler information can be extracted. Henceforth, in the paper, a rendered image is also called a frame; this forms the basic processing unit in Blender.

Each frame is rendered to be an image represented by $N_{\text{az}} \times N_{\text{el}}$ pixels, where N_{az} and N_{el} denote the number of pixels in the azimuth and elevation axes, respectively. Each pixel represents an object located in the pixel's angle of arrival (AoA) direction to the camera. As a result, any propagation path is denoted by propagation distance, signal attenuation due to scattering loss and propagation, and the AoA. Further, a dynamic scenario in Blender consists of N_{frame} frames at a particular frame rate of R_{frame} Hz. The velocity information is then calculated using the frame rate and the pixel strength across the frames. These parameters are now detailed.

B. Image rendering: parameters, their derivation, and visualization

1) Distance and strength matrices: In a dynamic scenario, the distance and strength of each pixel vary with the frames. To obtain these parameters, we begin by denoting $n_{\text{frame}} = 1, 2, ..., N_{\text{frame}}$ to be the index of the frame. The location of the object at the n_{el}th row and the n_{az}th column of the n_{frame}th frame is denoted as

$$\mathbf{l}_{n_{\text{frame}},n_{\text{el}},n_{\text{az}}} = [x_{n_{\text{frame}},n_{\text{el}},n_{\text{az}}}\ y_{n_{\text{frame}},n_{\text{el}},n_{\text{az}}}\ z_{n_{\text{frame}},n_{\text{el}},n_{\text{az}}}]^{\text{T}}. \quad (1)$$

The term $\mathbf{l}_{n_{\text{frame}},n_{\text{el}},n_{\text{az}}}$ is obtained from Blender after the modeling of the scenario.

Let $\mathbf{R}_{n_{\text{frame}}} \in \mathbb{R}^{N_{\text{el}} \times N_{\text{az}}}$ denote the output distance matrix of the n_{frame} frame, and $R_{n_{\text{frame}},n_{\text{el}},n_{\text{az}}}$ denote the distance of the received propagation path via the object at $\mathbf{l}_{n_{\text{frame}},n_{\text{el}},n_{\text{az}}}$. The path distance is calculated inside the RT engine. For example, the distance from Tx and scattered by the object at $\mathbf{l}_{n_{\text{frame}},n_{\text{el}},n_{\text{az}}}$ to Rx for the line of sight (LoS) path is determined as

$$R_{n_{\text{frame}},n_{\text{el}},n_{\text{az}}} = \frac{\|\mathbf{l}_{n_{\text{frame}},n_{\text{el}},n_{\text{az}}} - \mathbf{l}_{\text{tx}}\|_2}{2} + \frac{\|\mathbf{l}_{n_{\text{frame}},n_{\text{el}},n_{\text{az}}} - \mathbf{l}_{\text{rx}}\|_2}{2}, \quad (2)$$

where the $\mathbf{l}_{\text{tx}}$ and $\mathbf{l}_{\text{rx}}$ denote the location of the reference Tx and Rx, respectively. For non-LoS (NLoS) paths, the RT uses the geometry-based method, e.g., shooting-bouncing and image method, to find the interacting pixels and calculate the cascaded distance of multiple reflections.

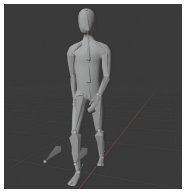
Let $\mathbf{P}_{\mathbf{r}n_{\text{frame}}} \in \mathbb{C}^{N_{\text{el}} \times N_{\text{az}}}$ denote the signal strength matrix of the n_{frame}th frame, and, in particular, $P_{r_{n_{\text{frame}},n_{\text{el}},n_{\text{az}}}}$ denotes the strength of the received propagation path via the object at $\mathbf{l}_{n_{\text{frame}},n_{\text{el}},n_{\text{az}}}$. This quantity is obtained from Blender. In principle, the LoS signal strength is calculated as

$$P_{r_{n_{\text{frame}},n_{\text{el}},n_{\text{az}}}} = \frac{P_t G_t \sigma_{n_{\text{frame}},n_{\text{el}},n_{\text{az}}}}{(4\pi)^2 (R_{n_{\text{frame}},n_{\text{el}},n_{\text{az}}}/2)^4} A_{\text{eff}}, \quad (3)$$

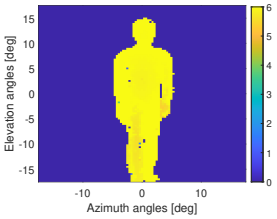
where $R_{n_{\text{frame}},n_{\text{el}},n_{\text{az}}}$ is the distance of the pixel obtained in (2), P_t, G_t, $A_{\text{eff}} = G_r \lambda^2/4\pi$, G_r, λ denote the transmitted power, transmitted antenna gain, the effective receive antenna aperture size, receiving antenna gain, wavelength, respectively. These quantities are either set in the Blender or calculated by the tool. Further, $\sigma_{n_{\text{frame}},n_{\text{el}},n_{\text{az}}}$, denotes the radar cross section (RCS) or the so-called scattering coefficient; its determination can be complicated and various scattering models are used in the literature, e.g., the Lambertian model used in optical waves [42], the single-lobe scattering pattern [43] and the back-scattering model [44] used in RF band. Generally, it is dependent on the material, frequency, incident angle, and even the Fresnel zone. In Blender, users can adjust parameters to obtain a relatively accurate replacement of (3) for different materials. For further studies, the field measurement-modified Blender parameters of more complex materials and frequencies can be adjusted based on the application requirement.

2) AoA of each pixel: Let $\mathbf{\Theta} \in \mathbb{R}^{N_{\text{el}} \times N_{\text{az}}}$ and $\mathbf{\Phi} \in \mathbb{R}^{N_{\text{el}} \times N_{\text{az}}}$ denote the azimuth and elevation AoA matrices, respectively with the assumption that the camera direction is the image center, i.e., both the azimuth and elevation angle are zero degrees. The entries in AoA matrices for the pixel at elevation index n_{el} and azimuth index n_{az} can be calculated as

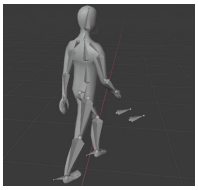
$$\begin{aligned} \Theta_{n_{\text{el}},n_{\text{az}}} &= \theta_{\text{bw}} \left(-\frac{1}{2} + \frac{n_{\text{az}} - 1}{N_{\text{az}} - 1} \right), \quad N_{\text{az}} > 1, \\ \Phi_{n_{\text{el}},n_{\text{az}}} &= \phi_{\text{bw}} \left(-\frac{1}{2} + \frac{n_{\text{el}} - 1}{N_{\text{el}} - 1} \right), \quad N_{\text{el}} > 1, \end{aligned} \quad (4)$$



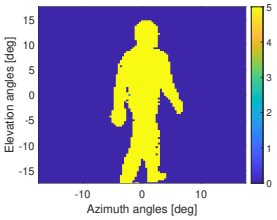
(a) Walking towards the camera



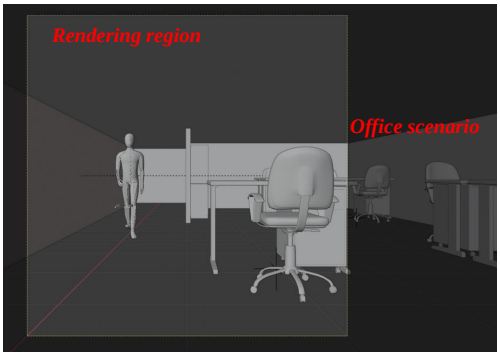
(b) Heat map of distance matrix



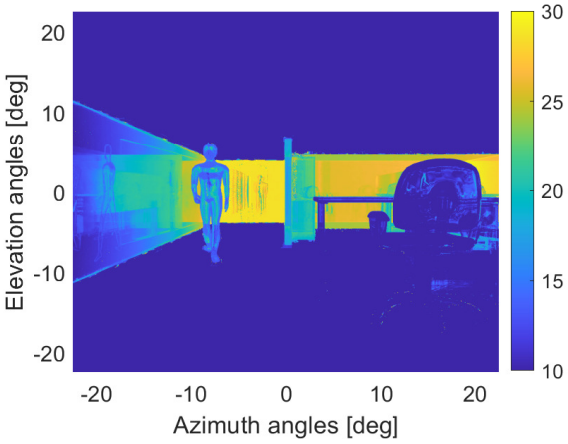
(c) Turning direction



(d) Heat map of distance matrix



(e) A human walking in an office scenario



(f) Heat map of rendering distance matrix after multiple bounces

Fig. 2. The modeling in Blender and rendering outputs are represented by the heat map as a function of elevation and azimuth angles, where the color bar denotes the distance in meters.

where $n_{\text{el}} = 1, 2, ..., N_{\text{el}}$, and $n_{\text{az}} = 1, 2, ..., N_{\text{az}}$ respectively, θ_{bw} and ϕ_{bw} are the azimuth and elevation FoV of the camera defined in Blender in degrees.

3) Visualization of image rendering: The rendering results of distance and strength matrices can be mapped with the AoA matrices using (2), (3), and (4), respectively. For example, Fig. 2 represents the heat map of the distance matrix[1] as a function of elevation and azimuth angles. Fig. 2 (a), (c), and (e) show the Blender models of pedestrian walking, turning, and an office scenario, respectively. The Fig. 2 (b), (d), and (f) illustrate the heat map of angle-distance matrices of each scenario, respectively, where the x, y and z axes denote the azimuth angle, elevation angle, and the distance of each pixel, respectively. Furthermore, multiple bounces can also be rendered and the user can define their maximum order in the rendering engine, e.g., in the indoor office scenario shown in Fig. 2 (f), the mirror images of the human and furniture due to multiple scattering can be observed clearly on the left and back walls. It is also worth mentioning that the one-bounce simulation is constrained by the FoV of the camera. However, the multiple bounces are not necessarily constrained by the FoV and the simulator is applicable in multiple target scenarios. This rendering offers a reference for the appropriate images subsequently created using radar signals.

In summary, we first define the FoV of the camera (i.e., the image view as seen by the camera) and the number of pixels in this image. This allows us to calculate the angle of arrival from each pixel to the camera using (4). The AoA of the multipath is obtained similarly using the pixel that is involved in that multipath.

C. Velocity calculation and mitigation of pixel migration

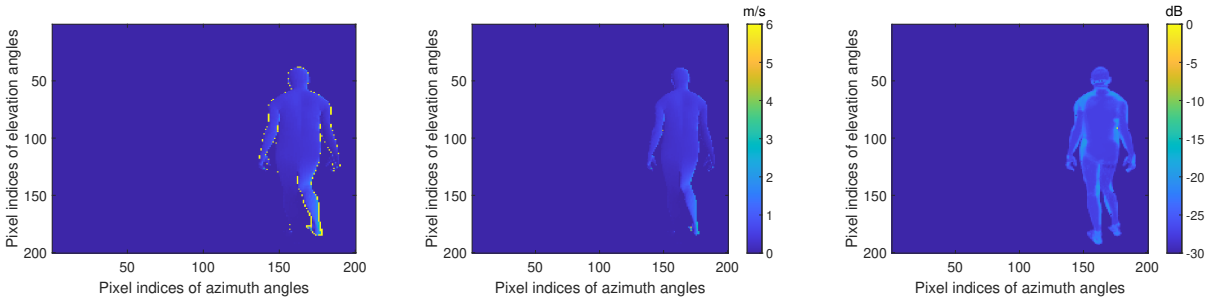
The difference in distance can be used for velocity calculation. Let $\mathbf{V}_{n_{\text{frame}}} \in \mathbb{R}^{N_{\text{el}} \times N_{\text{az}}}$ denote the radial velocity matrix, with the entries calculated as

$$V_{n_{\text{frame}}, n_{\text{el}}, n_{\text{az}}} = \frac{-2(R_{n_{\text{frame}}, n_{\text{el}}, n_{\text{az}}} - R_{n_{\text{frame}}-1, n_{\text{el}}, n_{\text{az}}})}{T_{\text{frame}}}, \quad (5)$$

where the range of each pixel $R_{n_{\text{frame}}, n_{\text{el}}, n_{\text{az}}}$ is the output of the Blender in (2), and $T_{\text{frame}} = 1/R_{\text{frame}}$, with R_{frame} being the frame rate of the imported MoCap databases or the manually set value of a rendered dynamic video in Blender. For example, R_{frame} is 30 Hz as mentioned in AMASS MoCap [39]. The Blender can also over-sample the R_{frame} of the imported motions, e.g., the case used in Fig. 3 is multiplied by two times the R_{frame} of AMASS MoCap databases to decrease pixel migration.

As objects move, the same pixel in subsequent frames tends to represent different scatterers; we refer to this as pixel migration. Setting short frame intervals, we assume a particular pixel in a frame continues to represent the same scatterer in the subsequent frame (i.e., we assume that the motion is confined within the pixel). Then the difference in distance for the same pixel in subsequent frames (i.e., pixels with identical AoA indices), obtained from Blender, can be used for velocity calculation. To avoid pixel migration based on this discussion, several settings in Blender need to be carefully chosen. As a case in point, a higher number of pixels for a given frame size leads to a lower size of each pixel thereby enhancing pixel migration; further, faster frame rates R_{frame} shorten the interval

[1]If there are no reflections, the distance in principle is infinite, for illustration and further processing convenience in calculating velocity, we set those to be zeros.



(a) Velocity image (b) Velocity image after filtering (c) Signal strength of pixels

Fig. 3. Velocity and strength heat maps with 200×200 pixels: the $R_{\text{frame}} = 60$ Hz, the threshold for filtering the pixel immigration is 6 m/s.

between two frames and reduce pixel migration. When setting those parameters, we have to consider the radar parameters in practice, the scenario, the relative velocities, and also the ultimate performance of the hardware and Blender software.

Nevertheless, pixel migration is inevitable and results in large velocity, especially at the edge of the target. Taking the human walking scenario in Fig. 3 (a) as an example, it can be observed that the velocities of some edge pixels are much greater than those corresponding to a human walking, which is typically less than 2 m/s. Considering that micro-Doppler velocity may be greater than the main body's velocity, a threshold of 6 m/s is applied. The velocity image after filtering is shown in Fig. 3 (b), where the extremely large velocities due to pixel migration are filtered out. It is also worth mentioning the signal strength image in Fig. 3 (c), where the signal strength of edge pixels is rather weak compared to the trunk, this feature prevents severe noise even if the pixel migration is not totally filtered.

III. Radar Signal Generation

Having obtained the key parameters from Blender, this section elaborates on the methodology to generate radar signals from the derived parameters. It begins by discussing the applicability of optical source-based results from Blender into RF-based radar systems, identifies the key missing component of MIMO, and provides a mechanism to generate MIMO radar signals based on FMCW waveforms.

A. Applicability of parameters

The RT renders the scenarios based on optical-ray principles, which can result in inaccurate powers of simulated paths. For purposes requiring accurate path loss predictions, e.g., wireless communication base station deployment, elaborated measurements are needed to calibrate RT parameters. However, for purposes of utilizing RT to identify the geometry parameters of propagation paths, e.g., sensing and environment mapping, accurate strength models are not necessary [45]. In this radar simulation, we are concerned about the radio path trajectory from RT, the geometry parameters, e.g., range, AoA, and the Doppler velocity due to target motions. Since the RT is based on the optical setting, the absolute reflected power would be different than mmWave propagation. However, users can define parameters like material and additional scattering loss in Blender. The simulation can obtain relative power strengths of targets and clutters based on the material settings and bouncing orders.

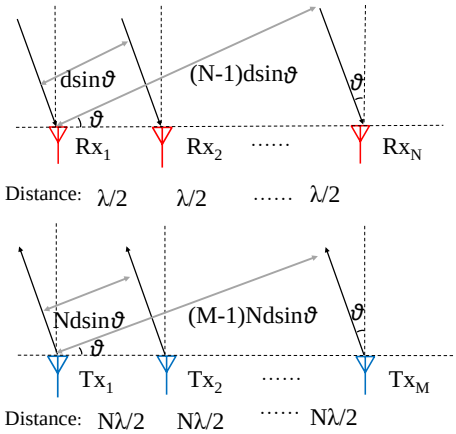


Fig. 4. An illustration of ULA with the AoA at Rxs and AoD at Txs denoted.

B. Virtual MIMO generation based on steering vector

The Blender-based rendering offers a single input single output (SISO) perspective. However, the paper is interested in generating radar signals beyond the SISO architecture. Towards this, and to enable MIMO, the rendering must be repeated for the position of each transmit-receive antenna pair. Such an exercise is resource-consuming and a simplified alternative is provided below.

Consider a colocated unified linear array (ULA) radar system with M transmitters (Tx) and N receivers (Rx) shown in Fig. 4, where the Rx elements are placed at a distance of $\lambda/2$ while the transmit elements are placed at a distance of $N\lambda/2$, Tx_m and Rx_n denote the mth Tx and the nth Rx element in the ULA, respectively, with $m = 1, 2, ..., M$ and $n = 1, 2, ..., N$ representing the index of the Tx and Rx elements, respectively. This resulting $M \times N$ channels generate a virtual MIMO, a more detailed illustration can refer to [46]. Kindly note that Tx_1 and Rx_1 are the reference antennas. Here we can not generate the $M \times N$ virtual channels simultaneously. Hence, a strategy for generating virtual MIMO channels based on the array geometry is utilized. We make use of only rendering results of one Tx-Rx pair, referring to them as the reference antennas in the sequel to generate virtual MIMO.

1) Distance matrix of virtual channel: Let $\mathbf{R}_{n_{\text{frame}},m,n} \in \mathbb{R}^{N_{\text{el}} \times N_{\text{az}}}$ denote the distance matrix of the channel between

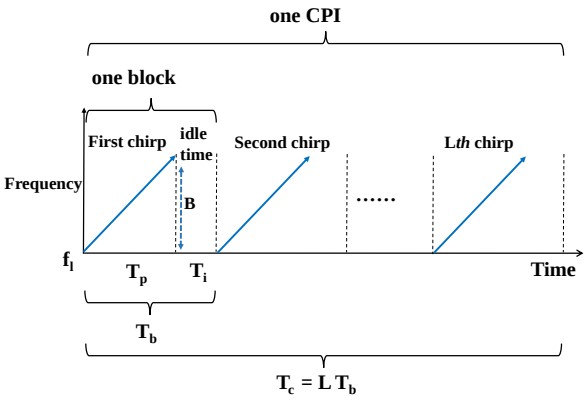


Fig. 5. Transmitted FMCW signal in one CPI.

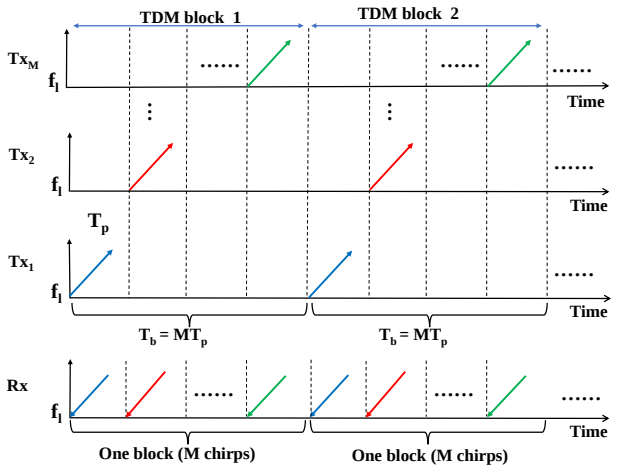


Fig. 6. An illustration of M-Tx TDM-MIMO FMCW signal.

Tx_m and Rx_n. Then, $\mathbf{R}_{n_{\text{frame}},m,n}$ can be generated based on the ULA geometry as

$$\begin{aligned}\mathbf{R}_{n_{\text{frame}}\,\mathrm{m,n}} =&\mathbf{R}_{n_{\text{frame}}} + (m-1)N\frac{\lambda}{2}\sin\boldsymbol{\Theta}_{n_{\text{frame}}}\\ &+(n-1)\frac{\lambda}{2}\sin\boldsymbol{\Theta}_{n_{\text{frame}}},\end{aligned} \tag{6}$$

where the $\mathbf{R}_{n_{\text{frame}}}$ is the reference range of the reference antennas obtained in (2) and $\boldsymbol{\Theta}_{n_{\text{frame}}}$ is the azimuth AoA matrix of the reference Rx obtained in (4). Here, we only consider the azimuth angles, it is also applicable to generate virtual rectangular MIMO by considering both azimuth and elevation angles.

2) Signal strength matrix of virtual channels: Considering the short spacing of the MIMO array in mmWave, the signal strengths of reference antennas are used for all the pairs of MIMO.

3) Doppler velocity matrix of virtual channels: Let $\mathbf{V}_{n_{\text{frame}},m,n} \in \mathbb{R}^{N_{\text{el}}\times N_{\text{az}}}$ denote the radial velocity matrix of the channel between Tx_m and Rx_n. Then, $\mathbf{V}_{n_{\text{frame}},m,n}$ can be obtained in a manner similar to the reference antenna in (5) as

$$\mathbf{V}_{n_{\text{frame}},m,n} = \frac{-2(\mathbf{R}_{n_{\text{frame}},m,n} - \mathbf{R}_{n_{\text{frame}-1},m,n})}{1/R_{\text{frame}}}. \tag{7}$$

Assuming the AoA angles are stationary within two frames and substituting (6) into (7), we have

$$\mathbf{V}_{n_{\text{frame}}} = \frac{-2(\mathbf{R}_{n_{\text{frame}}} - \mathbf{R}_{n_{\text{frame}-1}})}{1/R_{\text{frame}}}. \tag{8}$$

In this case, the Doppler is unchanged among MIMO channels in the simulation.

Finally, generating virtual MIMO channels in the aforementioned avoids significant processing delays caused by rendering all the paths in a multi-antenna system, one at a time.

Having obtained the relevant parameters, the sequel now discusses the generation of appropriate radar signals based on FMCW. Extensions to other waveforms are provided in Appendix B and Appendix C.

C. TDM-MIMO FMCW radar transmissions

FMCW is widely used in radar systems, due to the low-cost and efficient de-chirp techniques [8]. Fig. 5 shows the typical FMCW signal, the Tx transmits a linear frequency-modulated waveform sequentially. Each Coherent Processing Interval (CPI) consists of L chirps, with each chirp time T_p and idle time T_i consisting of a block duration T_b. Usually, in literature the idle time T_i is ignored, thereafter the block duration T_b can be replaced by the chirp time T_p. The T_b is the so-called pulse repetition interval (PRI). The indices of the chirps are indices of the so-called FMCW slow-time. In TDM-MIMO [47], [48], the orthogonality in time is exploited to separate the Tx signals at the Rx side. Recalling the colocated MIMO radar system with M Tx and N Rx elements as shown in Fig. 6, every TDM block consists of M time slots, each corresponding to transmission by one of the M Tx antennas.

The signal transmitted by the radar is a function of time t and chirp index l, and is obtained by the superposition of the transmitted signals of all the antennas as

$$s(t;l) = \sum_{m=1}^{M} s_m(t-(l-1)T_b-(m-1)T_p), \tag{9}$$

where $T_b = MT_p$ represents the TDM block time with the omission of the idle time, $l = 1, 2, ..., L$ is the index of TDM block, m is the index of Tx, and $s_m(t;l)$ is the transmitted signal of the mth Tx at the lth time block taking the form

$$s_m(t;l) = \sqrt{\frac{P_0}{2}}\exp(j\phi(\tilde{t})), \tag{10}$$

with

$$\tilde{t} = (t-(l-1)T_b-(m-1)T_p), \tag{11}$$

and P_0 is the transmitted power, mod denotes reminder operation, and the linear phase

$$\phi(\tilde{t}) = 2\pi(f_l\tilde{t} + \frac{1}{2}\mu\tilde{t}^2) - \phi_0, \tag{12}$$

where f_l is the starting frequency point, $\mu = B/T_p$ is the FMCW slope with B represents the bandwidth, T_p represents one chirp duration.

D. Beat signal model for TDM-MIMO FMCW radar

The received signal at the nth antenna is

$$r_n(t;l) = \sum_{m=1}^{M} \sigma_{m,n,l} s_m(\tilde{t} - \tau_{m,n,l}), \tag{13}$$

where $n = 1, ..., N$ is the index of Rx, $\sigma_{m,n,l}$ contains the RCS of the target and the channel attenuation between the mth Tx and the nth Rx at the lth CPI block, the propagation delay $\tau_{m,n,l}$ is time-varying and can be calculated as

$$\tau_{m,n,l} = 2\frac{R_{m,n,l} + v_r\tilde{t}}{c}, \tag{14}$$

where v_r is the speed of the target, $R_{m,n,l}$ is the range from the mth Tx antenna to the target and return to the nth Rx antenna at the lth chirp, for the ULA in Fig. 4 it can be calculated as

$$R_{m,n,l} = R_{0,l} + (m-1)Nd\sin\theta + (n-1)d\sin\theta, \tag{15}$$

where $R_{0,l}$ is the range between the target and the reference antennas, and θ is the AoA.

After the dechirp process on the receive side, the phase of the IF signal is

$$\Delta\psi_n(t;l) = \sum_{m=1}^{M} 2\pi(f_l\tau_{m,n,l} + \mu\tau_{m,n,l}\tilde{t} - \frac{1}{2}\mu\tau_{m,n,l}^2), \tag{16}$$

Substituting (14) into (16), we have

$$\Delta\psi_n(t;l) = \sum_{m=1}^{M} 2\pi((\frac{2\mu R_{m,n,l}}{c} - f_D)\tilde{t} + \phi_1), \tag{17}$$

where f_D represents the Doppler frequency shift, which can be represented as

$$f_D = -2\frac{v_r}{c}f_l, \tag{18}$$

and ϕ_1 is a constant

$$\phi_1 = \frac{2R_{m,n,l}f_l}{c}. \tag{19}$$

The detailed derivation of (17) is in Appendix A.

Every chirp block of the de-chirped signal is then sampled with the sampling frequency F_s thereby providing N_s number of samples. The sampling time $T_s = 1/F_s$ is the so-called fast time [49]. Let $\mathbf{Z}_n \in \mathbb{C}^{L\times N_s}$ be the beat signal of TDM-MIMO at the nth antenna, N_s denote the number of samples in one FMCW chirp. $n_s = 1, 2, ..., N_s$ and $l = 1, 2, ..., L$ denote the index of the samples and chirps, i.e., the fast time and slow time axis, respectively. The (l, n_s)-th entry can be expressed as

$$\mathbf{Z}_n(n_s,l) = \sum_{m=1}^{M} \sigma_{m,n,l}\exp(j2\pi(\frac{2\mu R_{m,n,l}}{c}\frac{n_s-1}{F_s} - f_D(l-1)T_b)). \tag{20}$$

With the beat signal model, the remaining task is to substitute relevant parameters from the Blender which is detailed in the sequel.

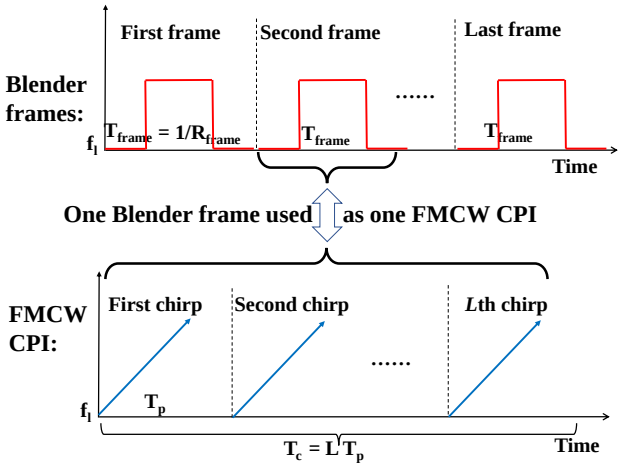


Fig. 7. The illustration of the mapping of Blender frames and FMCW CPIs.

E. Beat signal simulation for TDM-MIMO FMCW radar

In Blender-based radar simulation, the following assumptions are utilized:

- Each frame in Blender is regarded as one CPI, where there are L chirps within, hence the duration of CPI is $T_{\text{frame}} = 1/R_{\text{frame}}$, and the chirp duration is $T_p = T_{\text{frame}}/L$. An illustration of how Blender frames correspond to FMCW CPIs is provided in Fig. 7. The virtual slow time is generated by slicing one Blender frame to L chirps.
- In each chirp, the beat signal is obtained by sampling the IF signal with a sampling frequency F_s. Since F_s is much larger than f_D, the variations in the latter are ignored within sampling intervals.
- Each pixel of the Blender output image can be regarded as a target in radar detection, hence the received signal models used for simulation are the summary effects of $N_{\text{az}}N_{\text{el}}$ pixels.

Recalling the definition of $\mathbf{Z}_n$ from (20), the (l, n_s)-th entry of this matrix, after appropriate substitutions, can be expressed as

$$\mathbf{Z}_n(n_s,l) = \sum_{n_{\text{az}}=1}^{N_{\text{az}}}\sum_{n_{\text{el}}=1}^{N_{el}}\sum_{m=1}^{M} P_{r_{n_{\text{frame}},n_{\text{el}},n_{\text{az}}}}\exp(j2\pi(\frac{2\mu R_{n_{\text{frame}},n_{\text{el}},n_{\text{az}}}}{c}\frac{n_s-1}{F_s} - 2\frac{f_l V_{n_{\text{fame}},n_{\text{el}},n_{az}}}{c}(l-1)\frac{T_{\text{frame}}}{L} + \frac{((m-1)N+n-1)\Delta d}{\lambda}\sin\theta_{n_{\text{el}},n_{\text{az}}})), \tag{21}$$

where $\Delta d = \lambda/2$ is the interval between Rx antennas, λ is the wavelength, the $R_{n_{\text{frame}},n_{\text{el}},n_{\text{az}}}$, $P_{r_{n_{\text{el}},n_{\text{az}}}}$, $\Theta_{n_{\text{el}},n_{\text{az}}}$, and $V_{n_{\text{fame}},n_{\text{el}},n_{az}}$ are outputs of Blender obtained in (2), (3), (4), and (5), respectively.

In TDM-MIMO, the radar signal of different Txs can be separated based on the chirp index l at each Rx side, i.e., one TDM block contains M time orthogonal chirps. The signal from the mth Tx to the nth Rx $\mathbf{Z}_{m,n} \in \mathbb{C}^{(L/M)\times N_s}$ can be obtained as

$$\mathbf{Z}_{m,n}[\hat{l},:] = \mathbf{Z}_n[m + (\hat{l}-1)M, :], \tag{22}$$

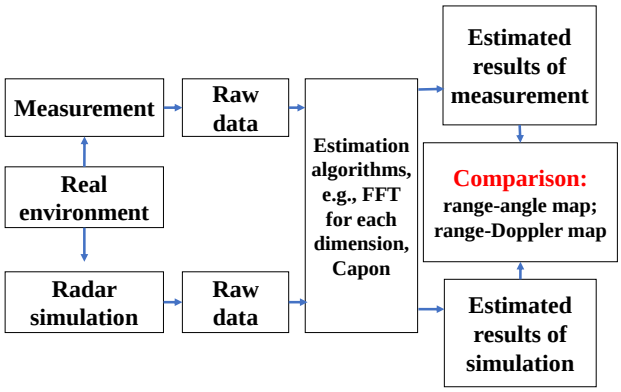


Fig. 8. The validation procedure of the radar simulator based on field measurement.

TABLE II
SENSOR CONFIGURATIONS

Configurations	Chamber	Corridor
Type of TI sensor	1642booster	6843ISK
Central frequency [GHz]	77	60
Bandwidth [GHz]	1.8	3.6
FMCW slope [MHz/us]	30	60
Frame/CPI duration [ms]	40	40
NO. of chirps per frame	128	128
Sampling rates [MHz]	10	10
NO. of IF samples	256	256
NO. of Tx	2	2
NO. of Rx	4	4
Sensor height [meter]	0.5	1.5
Walk speed [m/s]	1	[0.75, 1]

with

$$\hat{l} = 1, 2, ..., \frac{L}{M}. \tag{23}$$

In the preceding development, the beat signal for TDM mode radar with FMCW transmissions, denoted by $\mathbf{Z}_n$ has been obtained. Besides TDM mode, other commonly used multiplexing techniques include CDM-mode and FDM-mode [11], [50], [51]. However, there are no free lunches, and the utilization of MIMO for angle detection may sacrifice the Doppler ambiguity or the range ambiguity. Using the simulator as a tool to explore the features of MIMO configurations is possible. The beat signal models for other schemes can be developed in similar ways presented earlier for TDM-MIMO. Interested readers are kindly referred to Appendix B and Appendix C for details.

IV. VALIDATION

The developed radar simulator is validated by a human [2] walking in both anechoic chamber and office corridor scenarios, where the field measurements with mmWave Texas Instruments (TI) sensors are conducted. The validation procedure is outlined in Fig. 8. Firstly, a field measurement is carried out to collect raw data based on the TI mmWave TDM MIMO sensor. Important sensor configurations and measurement descriptions can be found in Table II, where these sensor configurations are applied in all subsequent measurements and simulations except those specifically mentioned. Subsequently, the environment and human motions are mimicked in Blender to obtain simulated data. Finally, applying the same estimation algorithms on both the measured and simulated data to compare the estimated range, angle, Doppler, and micro-Doppler results, and discuss the performance of the proposed simulator.

[2]All the testers are on the author list, hence informed consent was obtained.

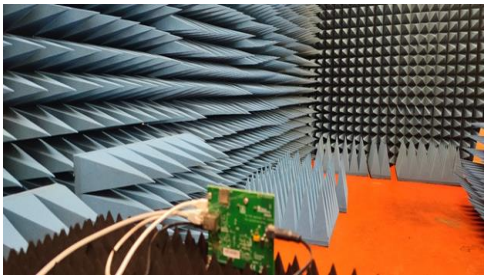


Fig. 9. A measurement picture in the anechoic chamber.

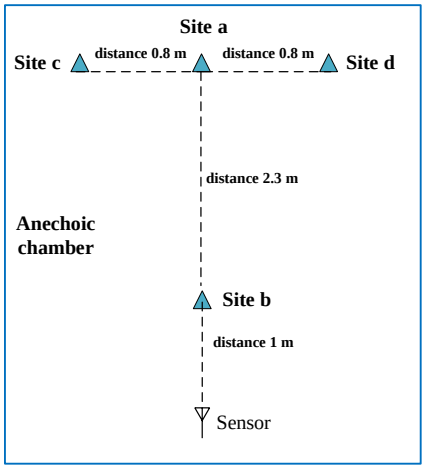
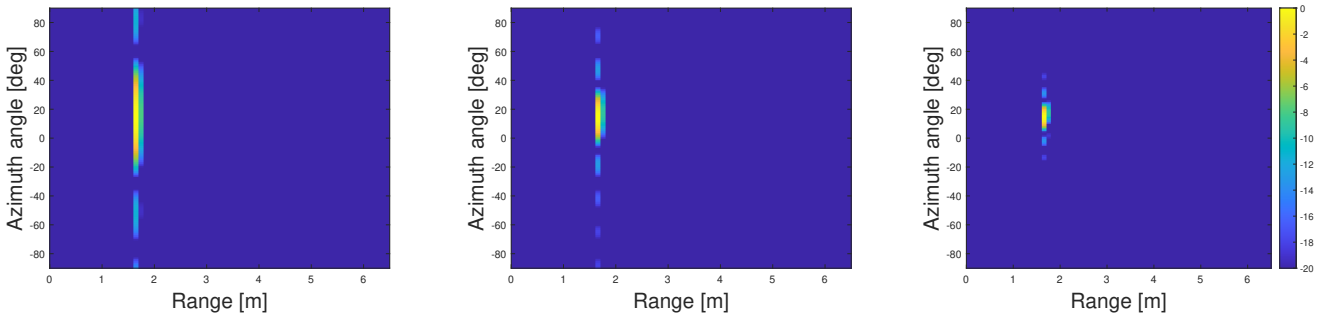


Fig. 10. Description of environment, sensor deployment, and walking routes in the anechoic chamber.

A. Measurement campaign of anechoic chamber

1) Measurement description: A picture of the measurement environment and sensor is shown in Fig. 9. A human walks in the predefined routes, which are illustrated in Fig. 10. A human walks from the **Site d** via **Site a** to **Site c**, and walks from **Site a** straight to **Site b**, then walks back to **Site a**.

2) Calibration in measurement: A static measurement is conducted to calibrate the power level, range, and angle offset of measurement, where a human is standing still at the **Site a** in Fig. 10. Then the RAM is obtained, where the estimated range and angle of the target are around 3.38 meters and -2.5 deg. In the ground truth as shown in Fig. 10, the range and angle should be 3.3 meters and 0 deg, hence concluding that the systematic error of measurement in range and angle are within 0.1 meters and 3 deg, respectively. To keep identical simulation and measurement power levels for a fair comparison, each radar image is normalized by the highest power of each image. Hence the highest power is 0 dB in all the figures shown later.



(a) Simulation of 1 by 4 antennas (b) Simulation of 2 by 4 antennas (c) Simulation of 4 by 4 antennas

Fig. 11. The simulated angular resolution using different numbers of MIMO antennas: the range-angle maps obtained via FFT.

3) Angular resolution of virtual MIMO in simulation: We simulate a human stand at **Site d** using different numbers of MIMO antennas to compare the simulator' angular resolutions. Fig. 11 shows the RAMs obtained via fast Fourier transform (FFT) of 1 by 4, 2 by 4, and 4 by 4 antennas in a ULA, respectively. We can observe that the angular resolution of 1 by 4, 2 by 4, and 4 by 4 ULA are around 40, 20, and 10 deg, respectively. As the number of antennas in the virtual array increases, the angular resolution increases, and the sidelobes are suppressed. It validates the simulator's capability of generating MIMO radar signals. The virtual MIMO array generation method keeps the properties of MIMO radars.

B. Comparison of the anechoic chamber scenario

1) Comparison of the range-angle maps: Considering the angular resolution of 2 by 4 ULA obtained by FFT is around 20 deg in Fig. 11. We apply Capon [52] to both simulated and measured data. The normalized RAMs at **Site d**, **Site a**, and **Site c** is shown in the first, second, and third column of Fig. 12, respectively, where the simulated and measurement results obtained via Capon [52] are shown in the sub-figures of the first and the second row, respectively. For each sub-figure, the point with the highest power is labeled with a red dot.

Generally, the simulation is in line with the measurement data, where both simulated angles and measured angles at **Site d**, **Site a**, and **Site c** are around $15\,\mathrm{deg}$, $0\,\mathrm{deg}$, and $-15\,\mathrm{deg}$, respectively. According to the trigonometric ranging formula, the theoretic absolute values of the angle of arrivals (AoAs) at **Site c** and **Site d** can be calculated as $\arctan\frac{0.8}{2.3+1}\frac{180}{\pi} = 13.63\,\mathrm{deg}$. The estimated ranges and angles corresponding to the maximum power in the RAM spectrum of both measurement and simulation are listed in Table III, where the difference between the ground truth and the simulation results in ranges and angles are less than 0.15 meters and 2.5 deg; the difference of measurement results in ranges and angles are less than 0.4 meters and 2.5 deg. Hence, both the simulated and measured RAMs match the ground truth.

2) Comparison of range-Doppler maps: The human walks from **Site a** to **Site b** and then walks back to **Site a** is used to compare the Doppler velocity. For the description convenience, the movement from **Site a** to **Site b** is called **Site ab** in the following discussion, and the movement from **Site b** to **Site a** is called **Site ba**. The estimated normalized RDM examples of **Site ab** of measurement, simulation with clutter, and simulation without clutter are shown with 30 dB dynamic range in Fig. 13 (a), (b), and (c), respectively.

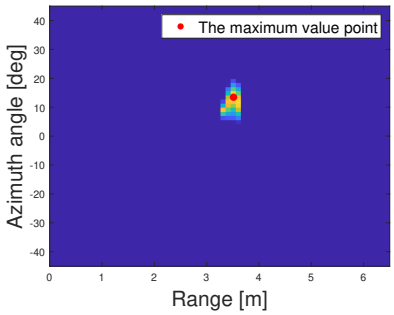
TABLE III
NUMERICAL COMPARISONS OF RANGE-ANGLE BETWEEN SIMULATIONS AND MEASUREMENTS

Contents	Site d	Site a	Site c
Measured range [meters]	3.74	3.37	3.51
Simulated range [meters]	3.51	3.41	3.5
Difference [meters]	0.23	0.04	0.01
Measured angle [deg]	15	-2.14	-13.57
Simulated angle [deg]	13.46	-2.12	-14.88
Difference [deg]	1.54	0.02	1.31

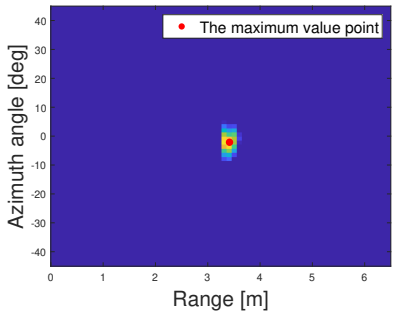
Typically in radar, we define the velocity of the target moving close to the radar to be negative, and moving away to be positive [53]. The estimated velocity of the simulation matches exactly with the measurement data of the movement **Site ab**, where the detected speed of the main body is around $-1\ m/s$. Besides, both the measurement and simulation results show dispersion in the ranges and micro-Doppler due to the walking motions. It gives a more clear observation of motion effects. Furthermore, we could also observe that the measurement results and simulation considering clutter contain some zero-Doppler returns, which are multipaths due to clutters in the measurement environment.

3) Comparison of continuous time-Doppler and micro-Doppler results: To have a further analysis of the Doppler and micro-Doppler results, we plot the continuous time-Doppler results based on each RDM, i.e., by choosing the velocity bins of the range index with the maximum value of the RDM and its neighbors to plot the continuous time-Doppler plots. This method is a commonly used way to observe the micro-Doppler [54]. The continuous time-Doppler plots of measurement, simulation with clutter, and without clutter are shown in Fig. 14 (a), (b), and (c), respectively.

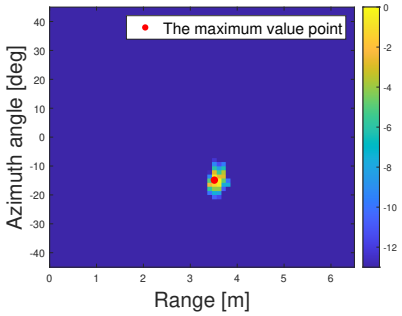
For both the measurement and simulation: (i) when the human walks toward the sensor, the Doppler velocity is around $-1\ m/s$; when the human walks away from the sensor, the Doppler velocity is around $1\ m/s$. (ii) The quasi-periodic micro-Doppler due to periodic arm and leg swings during



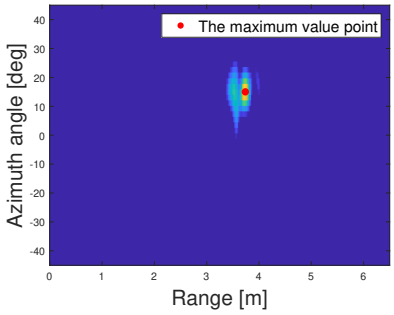
(a) Site d: Simulated RAM obtained via Capon



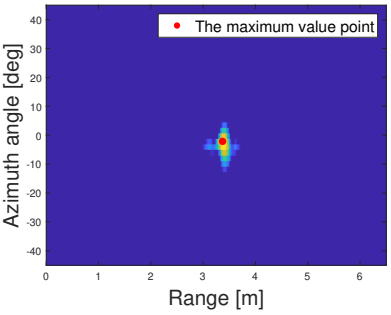
(b) Site a: Simulated RAM obtained via Capon



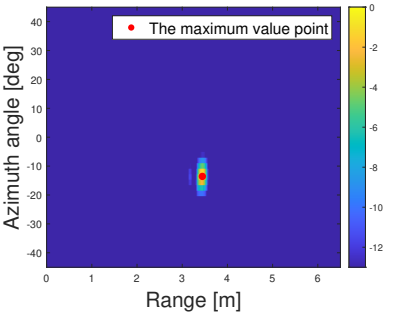
(c) Site c: Simulated RAM obtained via Capon



(d) Site d: Measured RAM obtained via Capon

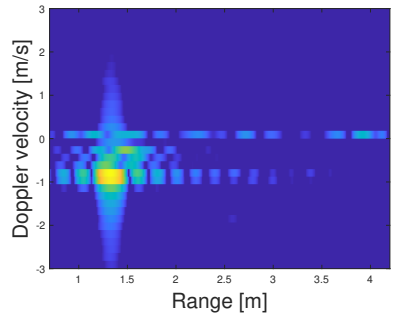


(e) Site a: Measured RAM obtained via Capon

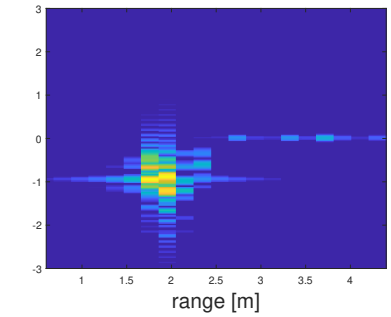


(f) Site c: Measured RAM obtained via Capon

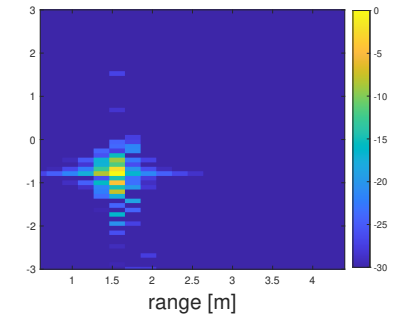
Fig. 12. Range-azimuth angle comparison of simulation and measurement via TDM-MIMO sensor.



(a) Measurement via TDM-MIMO sensor

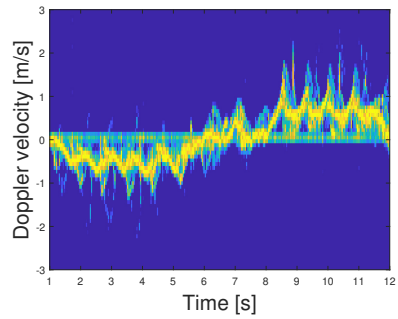


(b) Simulation considering clutter

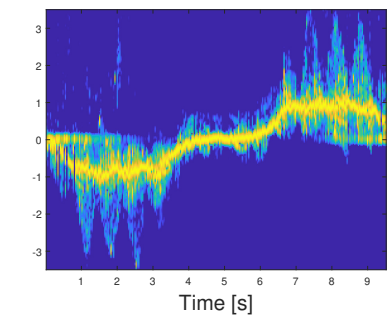


(c) Simulation without clutter

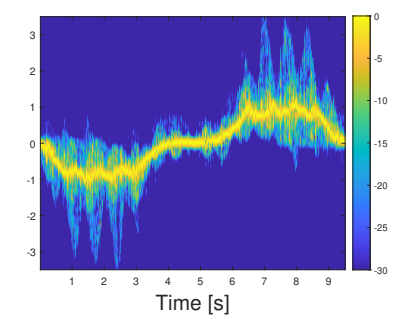
Fig. 13. Comparison of RDM obtained via 2D FFT: from Site a to Site b. simulation.



(a) Field measurement



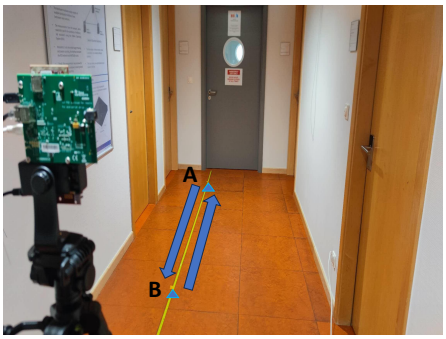
(b) Simulation considering clutter



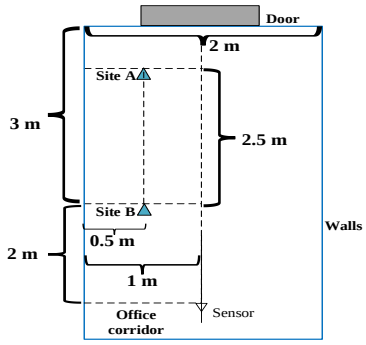
(c) Simulation without clutter

Fig. 14. Continuous time-Doppler plots comparison: the hamming window is applied to eliminate sidelobes for each RDM.

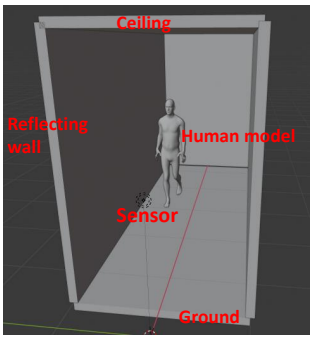
walking is observed. Simulation shows better micro-Doppler results than that obtained from measurement since Blender can capture any slight changes in motion, especially without clutter. Meanwhile, it could also be observed that the hardware has some limitations in the measurement result, e.g., the noise, however, the simulation approach shows good anti-interference



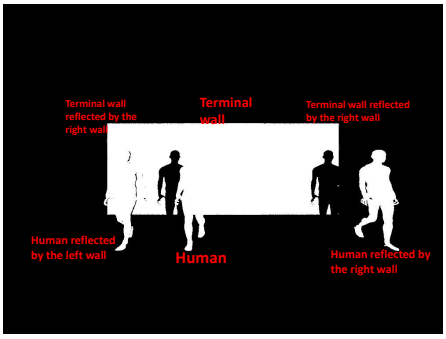
(a) Picture of the measurement



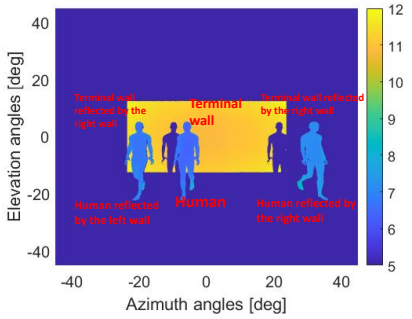
(b) Top view illustration



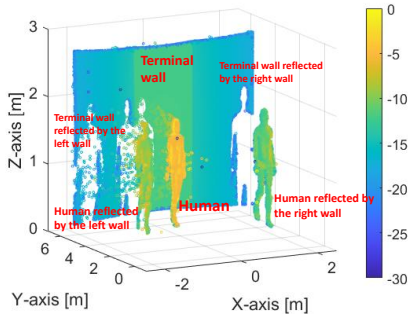
(c) 3D model of the office corridor scenario used in simulation



(d) Image rendering by considering two-bounce reflections: white pixels in the image denote a reflecting one-bounce or multi-bounce paths, the black pixels denote no multipaths in that direction



(e) Rendering outputs: each pixel is featured by distance, azimuth, and elevation angles, where the colorbar shows the total tracing distance



(f) Recovered point clouds of up-to two-bounce paths, where one point corresponds to one pixel that denotes one reflecting path, the color shows the normalized scattering strength of that path

Fig. 15. One frame description of the office corridor scenario in both measurement and simulation.

performance.

C. Measurement campaign of the office corridor scenario

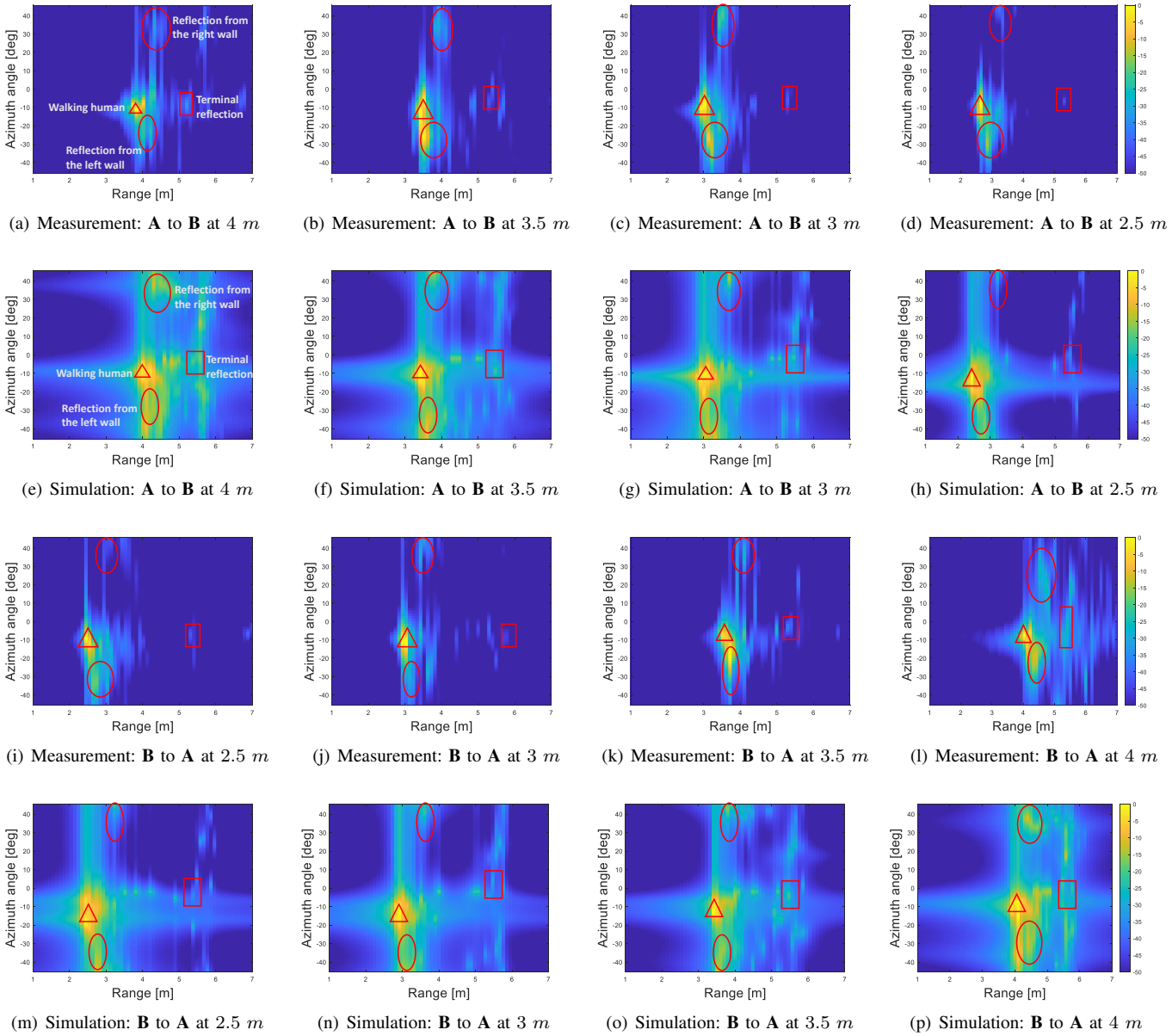
1) Measurement description: The measurement environment of the office corridor is shown in Fig. 15 (a). The width of the corridor is 2 meters, and the sensor is at the center facing a terminal located at a distance of more than 5 meters. A human walks along the predefined yellow line as shown in Fig. 15 (a) from **Site A** to **Site B** and returns. The sensor continuously measures the multipath signals from the moving human and the interaction with the environment. The detailed environment geometry and distance information are illustrated in Fig. 15 (b), and the sensor configuration is provided in Table II.

2) Simulation settings: The 3D environment is built in Blender according to the geometrical size of the measurement scenario, as shown in Fig. 15 (c). The left and right walls are set to be reflectors, and the terminal wall is set to be a diffuse scatter. Up to two bouncing paths are traced in this simulation setting. The human's walking velocity is around 1 m/s, where the whole motion process consists of 300-frame video with a frame rate of 30 Hz. This means the time difference between two images is $1/30$ seconds.

3) Rendering results: An example of the image rendering result is shown in Fig. 15 (d), where both the one-bounce and two-bounce paths are presented. Based on the strength of each pixel, the propagated distance corresponding to each pixel can be calculated in Blender as elaborated in [33], i.e., (i) the distance for one-bounce path: *light source* $\rightarrow$ *pixel* $\rightarrow$ *camera*; (ii) the distance for two-bounce path: *light source* $\rightarrow$ *pixel / (reflector)* $\rightarrow$ *pixel / (reflector)* $\rightarrow$ *camera*. As the view width of the camera is known, the AoAs of all pixels are known. The image rendering outputs can be illustrated by each pixel's AoA and its distance as shown in Fig. 15 (e). The pixels used in the simulation can be further illustrated as point clouds in Fig. 15 (f), using the distance, angles, and normalized strength calculated by RT in Blender. We can observe that the two-bounce paths are presented as virtual one-bounce paths with the exact distance and AoAs after two-bounce propagation. We can also notice from the color in Fig. 15 (f), the relative strength of paths diminishes after multiple bounces. Finally, each pixel is regarded as a one-bounce multipath and generates the radar signal according to (21).

D. Comparison of the office corridor scenario

1) Comparison of multi-bounce effects in range-angle maps: As the human walks from **Site A** to **Site B**, the distance between the human and the sensor changes from around 4.5 meters to 2 meters. We choose the measured RAMs at a distance of every 0.5 meter in the motion from **Site A** to **Site B**, i.e., the human at the ranges of 4, 3.5, 3, and 2.5 meters, to compare with the simulations. Those four measured RAMs are shown in Fig. 16 (a) to Fig. 16 (d), respectively. The return walk from **Site B** back to **Site A** at the range of



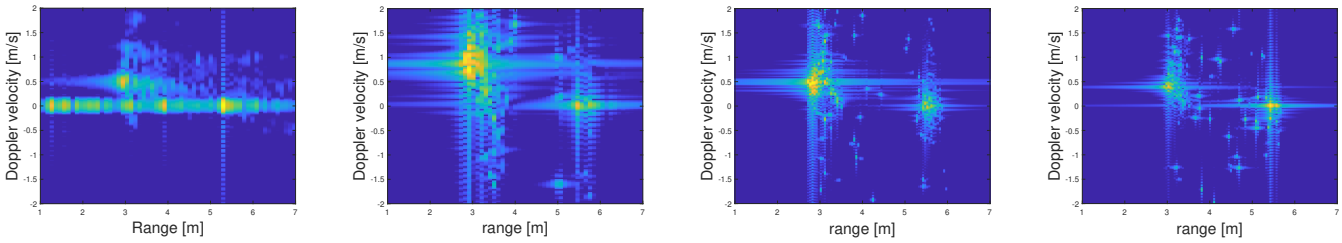
(a) Measurement: **A** to **B** at 4 m (b) Measurement: **A** to **B** at 3.5 m (c) Measurement: **A** to **B** at 3 m (d) Measurement: **A** to **B** at 2.5 m

(e) Simulation: **A** to **B** at 4 m (f) Simulation: **A** to **B** at 3.5 m (g) Simulation: **A** to **B** at 3 m (h) Simulation: **A** to **B** at 2.5 m

(i) Measurement: **B** to **A** at 2.5 m (j) Measurement: **B** to **A** at 3 m (k) Measurement: **B** to **A** at 3.5 m (l) Measurement: **B** to **A** at 4 m

(m) Simulation: **B** to **A** at 2.5 m (n) Simulation: **B** to **A** at 3 m (o) Simulation: **B** to **A** at 3.5 m (p) Simulation: **B** to **A** at 4 m

Fig. 16. Range-angle maps comparison between simulations and measurements.



(a) Measurement: **B** to **A** at 3 m (b) Simulation: **A** to **B** at 3 m (c) Simulation with velocity refined (d) Simulation with material refined

Fig. 17. Range-Doppler maps comparison between simulation and measurement.

2.5, 3, 3.5, and 4 meters are shown in Fig. 16 (i) to Fig. 16 (l), respectively. The simulated RAMs of walking from **Site A** to **Site B** at the range of 4, 3.5, 3, and 2.5 meters are shown in Fig. 16 (e) to Fig. 16 (h), respectively, and the return walk at the range of 2.5, 3, 3.5, and 4 meters are shown in Fig. 16 (m) to Fig. 16 (p), respectively.

Some RAMs are not identical and the following reason can be ascribed for the differences. The measurement and simulation are dynamic. The simulated human motions are imported from the MoCap database, which is recorded from

real human activities; hence the motion is not strictly symmetrical and contains random movements. Those dynamic factors also exist in measurements. Therefore, at a given distance, the gesture, orientation to the radar, and movement of part of the body may differ. The superposed signal contains the reflection of the walls, which can enlarge those differences. For comparison in the given dynamic process, the focus is laid on the macroscopic information concerning the target's location and the strong reflections from the left, right, and back walls. In the figures, these important and similar components are labeled as triangles, squares, and circles, respectively, in all the RAMs. Consider the measured Fig. 16 (a) and simulated Fig. 16 (e) as an example for comparison, we observe

- The one-bounce reflection of the human is highlighted at the range of 4 meters with an angle of -10 deg.
- At the range of around 4.2 meters, the reflection component by the left wall is around -25 deg.
- At the range of around 4.5 meters, the reflection component by the right wall is around 35 deg. The angle difference matches with the measurement environment because the human is closer to the left wall.
- The terminal wall reflection is around 5.5 meters with the angle 0 deg in all the RAMs.

2) Comparison of the range-Doppler maps: Using the common notion in radar that the velocity is defined to be negative when the distance between the target and sensor decreases, and positive when the distance increases, the measurement RDMs of the motion from **Site B** to **Site A** at the range of 3 meters is illustrated in Fig. 17 (a) and the corresponding simulation is illustrated in Fig. 17 (b).

There are two differences in Fig. 17 (a) and (b). Firstly, the range-Doppler map of measurement is more focused while the simulation is more spread. An identical walking velocity between the imported MoCap database and the measurements cannot be guaranteed and hence the velocity of the torso in the simulation and measurement are different. In Fig. 17 (a), the resulting human velocity is around $0.5\ m/s$, though the subject tried to walk at a velocity of $1\ m/s$ in the measurement, while in Fig. 17 (d), the imported simulation velocity is around $1\ m/s$. The superposition of reflection signals can enlarge those differences. These effects require calibration. In this context, we reduce the Blender frame rate in (5) by a factor of two, i.e., calibrated velocity in simulation is halved, and the shape of the simulated Doppler in Fig. 3 (c) becomes more focused and *more similar* to the measurement. This constant calibration is used for all frames of the scenario.

Secondly, the simulated reflection components are stronger than the measurement. By refining the additional $10\ dB$ loss in the target material, we can observe that the simulated RDM in Fig. 3 (d) matches better with the measurement.

- The non-zero velocities of RDMs are in the ranges of around 3 to 3.5 meters, where the human is at the range of around 3 meters with a velocity of around $0.5\ m/s$; the human reflections at the range of around 3.5 meters and with the velocity up-to $1.5\ m/s$. Another strong zero-velocity component of RDMs is mapping with the terminal wall at around 5.5 meters at RAMs.
- Compared with RDMs in anechoic chambers, the velocity becomes more dispersed, because of the superposition of multipath components.

3) Summary of the office corridor scenario: Generally, we can conclude that the simulated geometrical results, i.e., the distance, angle, and velocity of multipath components, are in line with the office corridor measurements. Besides, the strengths of some reflection components are not identical, thereby requiring material-specific calibration. We also find the interesting Doppler reflection phenomenon. Nevertheless, in this paper, we emphasize the general simulation framework for dynamic channels given the material reflectivity values. Calibrating the material loss parameters and quantifying the Doppler reflection are left for future modeling work.

V. Conclusion

This paper developed an FMCW MIMO radar channel simulation based on the Blender scenario animation. The simulator can generate time-varying radar signals with the consideration of MIMO modes. Field measurement of indoor pedestrians using mmWave sensors shows the validity and merits of the developed simulation tool, including the enhanced estimation of range, angle, Doppler, and micro-Doppler of the targets. Further, the simulation-based method also performs well for micro-Doppler assessment when compared with measurement in anechoic. This provides a general approach to generating radar channels, which can be useful for AI-based identification applications as well as the coming integrated sensing and communication era.

Appendix A
Derivation of (17)

Since $\tau_{m,n,l} \ll T_b$, we neglect the last term $\frac{1}{2}\mu\tau_{m,n,l}^2$ in (16), substitute (14) into (16),

$$\begin{aligned}\Delta\psi_n(t;l) &= \sum_{m=1}^{M} 2\pi(2(f_l+\mu\tilde{t})\frac{R_{m,n,l}+v_r\tilde{t}}{c}),\\ &= \sum_{m=1}^{M} 2\pi(\frac{2R_{m,n,l}f_l}{c}+\frac{2f_lv_r+2\mu R_{m,n,l}}{c}\tilde{t}+\frac{2\mu v_r\tilde{t}^2}{c}).\end{aligned} \tag{24}$$

Neglecting the range-Doppler-coupling term, i.e., $\frac{2\mu v_r\tilde{t}^2}{c}$, we obtain (17).

Appendix B
Simulation of CDM-MIMO

In FMCW, CMD refers to slow-time CDM waveforms, where the commonly used ones are Doppler division multiplexing (DDM) [55] and binary phase multiplexing (BPM) [56]. The transmitted FMCW signals can be represented as

$$\begin{aligned}s_C(t;l) &= \sum_{m=1}^{M} w_{m,l}s_{C,m}(t;l)\\ &= \sum_{m=1}^{M} w_{m,l}\sqrt{\frac{P_0}{2}}\exp(j\phi(t-(l-1)T_p)),\end{aligned} \tag{25}$$

where $\phi(t)$ is the phase of FMCW chirp defined in (12), and $w_{m,l}$ is the orthogonal phase code. For M-Tx DDM [55],

$$w_{m,l}^{\text{DDM}} = \exp(j2\pi \frac{(m-1)(l-1)}{M}). \tag{26}$$

For M-Tx BPM, $M = 2^k$, integer $k \geq 1$, the orthogonality codes can be provided by $M \times M$ Hadamard matrix [57]

$$\mathbf{W}_{2^k}^{\text{BPM}} = \mathbf{W}_2^{\text{BPM}} \otimes \mathbf{W}_{2^{k-1}}^{\text{BPM}}, \tag{27}$$

with

$$\mathbf{W}_2^{\text{BPM}} = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}. \tag{28}$$

Following the procedures and the assumptions in Blender simulation to obtain (21) as in Section III-D. Let the beat signal model of the nth Rx be $\mathbf{Z}_{C_n} \in \mathbb{C}^{L \times N_s}$, the (l, n_s)-th entry is

$$\begin{aligned} Z_{C_n\, l,n_s} = \sum_{n_{\text{az}}=1}^{N_{\text{az}}} \sum_{n_{\text{el}}=1}^{N_{el}} \sum_{m=1}^{M} w_{m,l} P_{r_{n_{\text{frame}},n_{\text{el}},n_{\text{az}}}} \exp(j2\pi(\\ \frac{2\mu R_{n_{\text{frame}},n_{\text{el}},n_{\text{az}}}}{c} \frac{n_s-1}{F_s} - 2\frac{f_l V_{n_{\text{fame}},n_{\text{el}},n_{az}}}{c} \frac{l-1}{L} T_{\text{frame}} \\ + \frac{((m-1)N+n-1)\Delta d}{\lambda} \sin\Theta_{n_{\text{el}},n_{\text{az}}})), \end{aligned} \tag{29}$$

where the $R_{n_{\text{frame}},n_{\text{el}},n_{\text{az}}}$, $P_{r_{n_{\text{el}},n_{\text{az}}}}$, $\Theta_{n_{\text{el}},n_{\text{az}}}$, and $V_{n_{\text{fame}},n_{\text{el}},n_{az}}$ are outputs of Blender obtained in (2), (3), (4), and (5), respectively.

In CDM-MIMO, decoding strategies are simplified based on the transmitted codes, hence it would be difficult to obtain a general decoding strategy. Coincidentally, the two-Tx DDM and two-Tx BPM share the same code. Here, we derive the general formula of CDM-MIMO in Blender-based radar signal simulation and take two-Tx DDM/BPM as an example to explain the decoding procedure. Consider two-Tx DDM/BPM, $\mathbf{Z}_{C_n}$ is the sum of signals from Tx1 and Tx2 as

$$\mathbf{Z}_{C_n} = \mathbf{Z}_{C_{1,n}} + \mathbf{Z}_{C_{2,n}}, \tag{30}$$

where

$$\mathbf{Z}_{C_{1,n}} \approx \exp(j(l-1)\pi) \exp(\frac{\Delta d N \sin\theta}{\lambda}) \mathbf{h}_{C_{2,n}}. \tag{31}$$

The channel of Tx1 $\mathbf{Z}_{C_{1,n}} \in \mathbb{C}^{L/2 \times N_s}$ can be demodulated as

$$Z_{C_{1,n}\,\tilde{l},n_s} = \frac{1}{2}\left(Z_{C_n\,2\tilde{l}-1,n_s} + Z_{C_n\,2\tilde{l},n_s}\right), \tag{32}$$

where $\tilde{L} = L/2$, $\tilde{l} = 1, 2, ..., \tilde{L}$, and the channel of Tx2 $\mathbf{Z}_{C_{2,n}} \in \mathbb{C}^{L/2 \times N_s}$ can be demodulated as

$$Z_{C_{2,n}\,\tilde{l},n_s} = \frac{1}{2}\left(Z_{C_n\,2\tilde{l}-1,n_s} - Z_{C_n\,2\tilde{l},n_s}\right). \tag{33}$$

In BPM/DDM-MIMO, we could obtain individual channels to enable MIMO, however with some limits: (1) We only get $L/2$ chirps instead of L chirps for two-Tx systems. The unambiguous velocity decreases to $v_{\text{max}}/2$. (2) The demodulation approach in (31), (32), and (33) ignores the differences of phase difference term $\exp(\Delta d N \sin\theta/\lambda)$ in (31), which would result ambiguity in angle sidelobe.

APPENDIX C
SIMULATION OF FDM-MIMO

For FDM-MIMO, the orthogonality is in frequency, i.e., Txs simultaneously transmit signals with non-overlapping frequency bands. The transmitted signal of FDM-MIMO can be represented as

$$\begin{aligned} s_F(t;l) = \sum_{m=1}^{M} s_{F,m}(t;l) \\ \sum_{m=1}^{M} \sqrt{\frac{P_0}{2}} \exp(j\phi_{F,m}(t-(l-1)T_b)), \end{aligned} \tag{34}$$

with

$$\phi_{F,m}(t) = 2\pi((f_l + (m-1)B)t + \frac{1}{2}\mu t^2) - \phi_0, \tag{35}$$

where B denotes the bandwidth of one FMCW chirp.

The received signal at the nth antenna can be represented as

$$r_{F,n}(t;l) = \sum_{m=1}^{M} \sigma_{m,n,l} s_{F.m}(t - \tau_{m,n,l}; l). \tag{36}$$

The dechirp processing of FDM is more complex than TDM and CDM in the hardware and RF processing, the received signal $r_{F,n}(t;l)$ is alternatively multiplied by M transmitted signals $s_{F,m}(t;l)$, so as to separate the coupling Tx signals. However, for simulation, we can proceed directly to the beat signal of FDM-MIMO between the mth Tx and the nth Rx antenna as

$$\begin{aligned} z_{F_{m,n}}(n_s;l) = \\ \sigma_{m,n,l} \exp(j2\pi(\frac{2\mu R_{m,n}}{c}\frac{n_s-1}{F_s} - f_{D_F,m}(l-1)T_p)), \end{aligned} \tag{37}$$

where the Doppler shift $f_{D_F,m,l}$ of FDM-MIMO is distinguished from TDM and CDM by the carrier frequency of the Tx index m as

$$f_{D_F,m} = -2\frac{(f_l + (m-1)B)v_r}{c}. \tag{38}$$

Applying the assumptions of Blender-based radar simulation used in TDM-MIMO. Let the beat signal model between the mth Tx and the nth Rx antennas used for FDM-MIMO simulation be $\mathbf{Z}_{F_{m,n}} \in \mathbb{C}^{L \times N_s}$, the (l, n_s)-th entry can be expressed as

$$\begin{aligned} Z_{F_{m,n}\,l,n_s} = \sum_{n_{\text{az}}=1}^{N_{\text{az}}} \sum_{n_{\text{el}}=1}^{N_{el}} \sum_{m=1}^{M} P_{r_{n_{\text{frame}},n_{\text{el}},n_{\text{az}}}} \\ \exp(j2\pi(\frac{2\mu R_{n_{\text{frame}},n_{\text{el}},n_{\text{az}}}}{c}\frac{n_s-1}{F_s} \\ - 2\frac{(f_l+(m-1)B)V_{n_{\text{fame}},n_{\text{el}},n_{az}}}{c}\frac{l-1}{L}T_{\text{frame}} \\ + \frac{((m-1)N+n-1)\Delta d}{\lambda}\sin\Theta_{n_{\text{el}},n_{\text{az}}})), \end{aligned} \tag{39}$$

where the $R_{n_{\text{frame}},n_{\text{el}},n_{\text{az}}}$, $P_{r_{n_{\text{el}},n_{\text{az}}}}$, $\Theta_{n_{\text{el}},n_{\text{az}}}$, and $V_{n_{\text{fame}},n_{\text{el}},n_{az}}$ are outputs of Blender defined in (2), (3), (4), and (5), respectively.

For FDM-MIMO, the channels of different Txs have already been separated in (39), which is distinguished from the (21) in TDM and (29) in CDM. We do not need extra decoding approaches.

REFERENCES

[1] L. Wu, M. Alaee-Kerahroodi, and B. M. R. Shankar, "Improving pulse-compression weather radar via the joint design of subpulses and extended mismatch filter," in *Proc. Int. Geosci. Remote Sens. Symp. (IGARSS)*, July 2022, pp. 469–472.

[2] S. Brusch, S. Lehner, T. Fritz, M. Soccorsi, A. Soloviev, and B. van Schie, "Ship surveillance with TerraSAR-X," *IEEE Trans. Geosci. Remote Sens.*, vol. 49, no. 3, pp. 1092–1103, 2011.

[3] R. Chen, W. Long, Y. Gao, and J. Li, "Orbital angular momentum-based two-dimensional super-resolution targets imaging," in *Proc. IEEE Global Conf. Signal Inf. Process. (GlobalSIP)*, November 2018, pp. 1243–1246.

[4] J. Pegoraro, D. Solimini, F. Matteo, E. Bashirov, F. Meneghello, and M. Rossi, "Deep learning for accurate indoor human tracking with a mm-Wave radar," in *Proc. IEEE Radar Conf. (RadarConf)*, September 2020, pp. 1–6.

[5] A. Murtada, R. Hu, B. S. M. R. Rao, and U. Schroeder, "Widely distributed radar imaging: Unmediated ADMM based approach," *IEEE J. Sel. Top. Signal Process.*, vol. 17, no. 2, pp. 389–402, September 2023. [Online]. Available: https://doi.org/10.1109/JSTSP.2022.3210766

[6] G. Beltrão, R. Stutz, F. Hornberger, W. A. Martins, D. Tatarinov, M. Alaee-Kerahroodi, U. Lindner, L. Stock, E. Kaiser, S. Goedicke-Fritz *et al.*, "Contactless radar-based breathing monitoring of premature infants in the neonatal intensive care unit," *Scientific reports*, vol. 12, no. 1, pp. 1–15, 2022.

[7] Q. An, S. Wang, L. Yao, A. Hoorfar, W. Zhang, H. Lv, S. Li, and J. Wang, "Range-Max Enhanced Ultrawideband Micro-Doppler Signatures of Behind-the-Wall Indoor Human Motions," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, pp. 1–19, October 2022.

[8] R. Amar, M. Alaee-Kerahroodi, P. Babu, and B. S. M. R., "Designing interference-immune Doppler-tolerant waveforms for radar systems," *IEEE Trans. Aerosp. Electron. Syst.*, pp. 1–20, October 2022.

[9] Y. Rong, I. Lenz, and D. W. Bliss, "Vital Signs Detection Based on High-Resolution 3-D mmWave Radar Imaging," in *Proc. IEEE Int. Symp. Pha. Array Sys. Tech. (PAST)*, October 2022, pp. 1–6.

[10] Y. Lyu, P. Kyösti, and W. Fan, "Sub-terahertz channel sounder: Review and future challenges," *China Commun*, vol. 20, no. 6, pp. 26–48, June 2023.

[11] M. Alaee-Kerahroodi, P. Babu, M. Soltanalian, and M. B. Shankar, *Signal Design for Modern Radar Systems*. Artech House, 2022.

[12] K. Witrisal, P. Meissner, E. Leitinger, Y. Shen, C. Gustafson, F. Tufvesson, K. Haneda, D. Dardari, A. F. Molisch, A. Conti, and M. Z. Win, "High-accuracy localization for assisted living: 5G systems will turn multipath channels from foe to friend," *IEEE Signal Process. Mag.*, vol. 33, no. 2, pp. 59–70, March 2016.

[13] X. Yin and X. Cheng, *Propagation channel characterization, parameter estimation, and modeling for wireless communications*. John Wiley & Sons, 2016.

[14] A. Guerra, F. Guidi, D. Dardari, A. Clemente, and R. DErrico, "A Millimeter-Wave Indoor Backscattering Channel Model for Environment Mapping," *IEEE Trans. Antennas Propag.*, vol. 65, no. 9, pp. 4935–4940, July 2017.

[15] C. Tang, W. Li, S. Vishwakarma, F. Shi, S. Julier, and K. Chetty, "Mdpose: Human skeletal motion reconstruction using wifi micro-doppler signatures," *IEEE Trans. Aerosp. Electron. Syst.*, pp. 1–12, March 2023.

[16] W. Yang, Y. Chen, N. Cardona, Y. Zhang, Z. Yu, M. Zhang, J. Li, Y. Chen, and P. Zhu, "Integrated sensing and communication channel modeling and measurements: Requirements and methodologies toward 6G standardization," *IEEE Vehicular Technology Magazine*, pp. 2–10, 2024.

[17] Y. Liu, X. Yin, X. Ye, Y. He, and J. Lee, "Embedded propagation graph model for reflection and scattering and its millimeter-wave measurement-based evaluation," *IEEE Open J. Antennas Propag.*, vol. 2, pp. 191–202, January 2021.

[18] V. Degli-Esposti, F. Fuschini, E. M. Vitucci, M. Barbiroli, M. Zoli, L. Tian, X. Yin, D. A. Dupleich, R. Muller, C. Schneider, and R. S. Thoma, "Ray-tracing-based mm-Wave beamforming assessment," *IEEE Access*, vol. 2, pp. 1314–1325, October 2014.

[19] P. Koivumäki, G. Steinböck, and K. Haneda, "Impacts of point cloud modeling on the accuracy of ray-based multipath propagation simulations," *IEEE Trans. Antennas Propag.*, vol. 69, no. 8, pp. 4737–4747, January 2021.

[20] T. Pedersen, G. Steinbock, and B. H. Fleury, "Modeling of reverberant radio channels using propagation graphs," *IEEE Trans. Antennas Propag.*, vol. 60, no. 12, pp. 5978–5988, August 2012.

[21] A. Bharti, R. Adeogun, and T. Pedersen, "Learning parameters of stochastic radio channel models from summaries," *IEEE Open J. Antennas Propag*, vol. 1, pp. 175–188, April 2020.

[22] L. Tian, V. Degli-Esposti, E. M. Vitucci, and X. Yin, "Semi-deterministic radio channel modeling based on graph theory and ray-tracing," *IEEE Trans. Antennas Propag.*, vol. 64, no. 6, pp. 2475–2486, March 2016.

[23] Y. Miao, T. Pedersen, M. Gan, E. Vinogradov, and C. Oestges, "Reverberant room-to-room radio channel prediction by using rays and graphs," *IEEE Trans. Antennas Propag.*, vol. 67, no. 1, pp. 484–494, October 2019.

[24] D. He, B. Ai, K. Guan, L. Wang, Z. Zhong, and T. Kurner, "The design and applications of high-performance ray-tracing simulation platform for 5G and beyond wireless communications: A tutorial," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 1, pp. 10–27, Firstquarter 2019.

[25] J. Chen, X. Yin, L. Tian, and M.-D. Kim, "Millimeter-wave channel modeling based on a unified propagation graph theory," *IEEE Commun. Lett.*, vol. 21, no. 2, pp. 246–249, February 2017.

[26] S. Zhao, Y. Liu, L. Wu, J. Rodrjguez-Pineiro, X. Yin, and J. Hong, "Recursive UE localization for a multi-RIS-assisted wireless system in an obstacle-dense environment," in *Proc. 17th Eur. Conf. Ant. Prop. (EuCAP)*, March 2023, pp. 1–5.

[27] X. Ye, J. Rodriguez-Pineiro, Y. Liu, X. Yin, and A. Perez Yuste, "A novel experiment-free site-specific TDoA localization performance-evaluation approach," *Sensors*, vol. 20, no. 4, 2020. [Online]. Available: https://www.mdpi.com/1424-8220/20/4/1035

[28] K. Guan, X. Lin, D. He, B. Ai, Z. Zhong, Z. Zhao, D. Miao, H. Guan, and T. Kurner, "Scenario modules and ray-tracing simulations of millimeter wave and terahertz channels for smart rail mobility," in *Proc. 11th Eur. Conf. Ant. Prop. (EuCAP)*, March 2017, pp. 113–117.

[29] X. Cai, J. Rodriguez-Pineiro, X. Yin, N. Wang, B. Ai, G. F. Pedersen, and A. P. Yuste, "An empirical air-to-ground channel model based on passive measurements in LTE," *IEEE Trans. Veh. Technol.*, vol. 68, no. 2, pp. 1140–1154, February 2019.

[30] A. F. Molisch, *Wireless communications*. John Wiley & Sons, 2012.

[31] C. Schubler, M. Hoffmann, J. Braunig, I. Ullmann, R. Ebelt, and M. Vossiek, "A realistic radar ray tracing simulator for large MIMO-arrays in automotive environments," *IEEE J. Microw.*, vol. 1, no. 4, pp. 962–974, October 2021.

[32] M. Ouza, M. Ulrich, and B. Yang, "A simple radar simulation tool for 3D objects based on blender," in *Proc. 18th Int. Radar Symp. (IRS)*, June 2017, pp. 1–10.

[33] R. F. da Costa, D. d. S. de Medeiros, R. Andrade, O. Saotome, and R. Machado, "General purpose radar simulator based on Blender Cycles path tracer," in *XXXVIII Simposio Brasileiro de Telecomunicacoes de Processamento de Sinais (SBrT2020)*, 2020.

[34] F. Krebs, A. Meixner, I. Patzer, and T. Asfour, "The KIT bimanual manipulation dataset," in *20th IEEE-RAS Int. Conf. on Humanoid Robots, (Humanoids)*, July 2021, pp. 499–506. [Online]. Available: https://doi.org/10.1109/HUMANOIDS47582.2021.9555788

[35] G. Pavlakos, V. Choutas, N. Ghorbani, T. Bolkart, A. A. Osman, D. Tzionas, and M. J. Black, "Expressive body capture: 3D hands, face, and body from a single image," in *IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR)*, June 2019, pp. 10 967–10 977.

[36] J. Hoydis, F. A. Aoudia, S. Cammerer, M. Nimier-David, N. Binder, G. Marcus, and A. Keller, "Sionna RT: Differentiable ray tracing for radio propagation modeling," 2023.

[37] Y. Liu, M. Ahmadi, J. Fuchs, M. Alaee-Kerahroodi, and M. Shankar, "A blender-based channel simulator for FMCW radar," *arXiv preprint arXiv:2307.09226*, 2023.

[38] A. F. Applications. Altair only forward. [Online]. Available: https://www.altair.com/

[39] N. Mahmood, N. Ghorbani, N. F. Troje, G. Pons-Moll, and M. J. Black, "AMASS: Archive of motion capture as surface shapes," in *Int. Conf. Computer Vision (ICCV)*, Oct. 2019, pp. 5442–5451.

[40] G. Richichi and G. Junkin, "An optimization method for tuning a helical beam antenna using Blender-Python and CFDTD," in *Proc. 2nd URSI Atlantic Radio Science Meeting (AT-RASC)*, May 2018, pp. 1–1.

[41] B. Foundation. Blender is the free and open source 3D creation suite. [Online]. Available: https://www.blender.org/

[42] D. Wu, Z. Ghassemlooy, S. Rajbhandari, H. Le Minh, and A. C. Boucouvalas, "Improvement of transmission bandwidth for indoor optical wireless communication systems using an elliptical lambertian beam," *IEEE Photon. Technol. Lett.*, vol. 25, no. 2, pp. 107–110, January 2013.

[43] V. Degli-Esposti, F. Fuschini, E. M. Vitucci, and G. Falciasecca, "Measurement and modelling of scattering from buildings," *IEEE Trans. Antennas Propag.*, vol. 55, no. 1, pp. 143–153, January 2007.

[44] T.-Q. Fan, L.-X. Guo, B. Lv, and W. Liu, "An improved backward SBR-PO/PTD hybrid method for the backward scattering prediction of an electrically large target," *IEEE Antennas Wireless Propag. Lett.*, vol. 15, pp. 512–515, 2016.

[45] H. Mi, B. Ai, R. He, R. Caromi, J. Wang, A. Bodi, C. Gentile, and Y. Miao, "Cluster association for 3D environment based on 60 GHz indoor channel measurements," in *Proc. 17th Eur. Conf. Ant. Prop. (EuCAP)*, March 2023, pp. 1–5.

[46] Y. Liu, M. Ahmadi, J. Fuchs, and M. R. B. Shankar, "Time-code-spatial modulated IRS-aided radar localization in NLoS scenario," in *Proc. IEEE Radar Conf. (RadarConf23)*, 2023, pp. 1–5.

[47] V. Winkler, "Range Doppler detection for automotive FMCW radars," in *Proc. Eur. Radar Conf. (EuRAD)*, October 2007, pp. 166–169.

[48] J. Lin Jr, Y.-P. Li, W.-C. Hsu, and T.-S. Lee, "Design of an FMCW radar baseband signal processing system for automotive application," *SpringerPlus*, vol. 5, no. 1, pp. 1–16, 2016.

[49] T. Instruments. Introduction to mmwave radar sensing: FMCW radars. [Online]. Available: https://training.ti.com/node/1139153

[50] H. Sun, F. Brigui, and M. Lesturgie, "Analysis and comparison of MIMO radar waveforms," in *Proc. Int. Radar Conf. (RADAR)*, October 2014, pp. 1–6.

[51] E. Raei, M. Alaee-Kerahroodi, B. S. M. R., and B. Ottersten, "Transmit beampattern shaping via waveform design in cognitive Mimo radar," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2020, pp. 4582–4586.

[52] X. Zhang and D. Xu, "Angle estimation in bistatic MIMO radar using improved reduced dimension Capon algorithm," *J. Syst. Eng. Electron.*, vol. 24, no. 1, pp. 84–89, February 2013.

[53] V. Chen, F. Li, S.-S. Ho, and H. Wechsler, "Micro-doppler effect in radar: phenomenon, model, and simulation study," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 42, no. 1, pp. 2–21, 2006.

[54] F. Fioranelli, H. Griffiths, M. Ritchie, and A. Balleri, Eds., *Micro-Doppler Radar and its Applications*. The Institute of Engineering and Technology (IET), 2020.

[55] Y. Sun, M. Bauduin, and A. Bourdoux, "Enhancing unambiguous velocity in Doppler-division multiplexing MIMO radar," in *Proc. 18th Eur. Radar Conf. (EuRAD)*, April 2022, pp. 493–496.

[56] H. A. Gonzalez, C. Liu, B. Vogginger, and C. G. Mayr, "Doppler ambiguity resolution for binary-phase-modulated MIMO FMCW radars," in *Proc. Int. Radar Conf. (RADAR)*, September 2019, pp. 1–6.

[57] F. Xu, S. A. Vorobyov, and F. Yang, "Transmit beamspace DDMA based automotive MIMO radar," *IEEE Trans. Veh. Technol.*, vol. 71, no. 2, pp. 1669–1684, February 2022.

Yuan Liu received his Bachelor's degree from Central China Normal University (CCNU) (Wuhan, China) in 2017 and Master's degree from Tongji University (Shanghai, China) in 2020, where he was a joint student at Aalto University (Helsinki, Finland) from 2018 to 2019. He was a wireless algorithm engineer at Guangdong Communications & Networks Institute (Guangzhou, China) from 2020 to 2021. He is now pursuing a Ph.D. degree at the Signal Processing Applications in Radar and Communications research group, SPARC, SnT, University of Luxembourg (Luxembourg), working on mmWave/THz channel models for sensing systems.

Moein Ahmadi received the M.Sc. and the Ph.D. degree in information engineering from the K.N. Toosi University of Technology, Tehran, Iran, in 2009, 2014, respectively. He is currently a postdoctoral researcher at the University of Luxembourg. His general research interests include statistical signal processing, detection and estimation, and radars.

Johann Fuchs received his Bachelor's and Master's degrees from Trier University of Applied Sciences (Germany) in 2009 and 2013 respectively. He has a long working experience as software developer in industrial companies with focus on user interfaces, signal processing and communication protocols. His research interests are radar signal processing and free open source software. Johann joined the Signal Processing Applications in Radar and Communications, SPARC, research group headed by Prof. Bhavani Shankar in 2020. Since then, he has been working on applying radar technologies to areas such as robotics, simulation, and monitoring vital signs.

Mohammad Alaee-Kerahroodi (SM'23) received his Ph.D. degree in telecommunication engineering from the Department of Electrical and Computer Engineering, Isfahan University of Technology, Iran, in 2017. After obtaining his Ph.D. degree, he joined the Interdisciplinary Centre for Security, Reliability, and Trust (SnT), University of Luxembourg, Luxembourg, as a Research Associate. He is currently a Research Scientist with SnT, University of Luxembourg, where he leads the prototyping and laboratory activities for the Signal Processing Applications in Radar and Communications (SPARC) Research Group. His research interest is radar waveform design and optimization with a particular interest in signal processing for 4D imaging millimeter wave MIMO radar sensors.

Bhavani Shankar M. R (SM'16) received Masters and Ph. D in Electrical Communication Engineering from Indian Institute of Science, Bangalore in 2000 and 2007 respectively. He was a Post Doc at the ACCESS Linnaeus Centre, Signal Processing Lab, Royal Institute of Technology (KTH), Sweden from 2007 to September 2009. He joined SnT in October 2009 as a Research Associate and is currently a Senior Research Scientist/ Assistant Professor at SnT leading the Radar Signal Processing activities. He was with Beceem Communications, Bangalore from 2006 to 2007 as a Staff Design Engineer working on Physical Layer algorithms for WiMAX compliant chipsets. He was a visiting student at the Communication Theory Group, ETH Zurich, headed by Prof. Helmut Bölcskei during 2004. Prior to joining Ph. D, he worked on Audio Coding algorithms in Sasken Communications, Bangalore as a Design Engineer from 2000 to 2001. He is currently the Chair of the IEEE Benelux joint chapter on communications and vehicular technology, serves as handling editor for EURASIP Journal on Advances in Signal Processing and is a member of the IEEE SPS SAM Technical Committee. He was a co-recipient of the 2014 Distinguished Contributions to Satellite Communications Award, from the Satellite and Space Communications Technical Committee of the IEEE Communications Society and co-author of the paper that received Special Mention for the Barry Carlton best paper award by IEEE TAES in 2023.