DeepMIMOv2 Dataset Channel Generation

Notation: $\mathbf{A}$ is a matrix, $\mathbf{a}$ is a vector, a is a scalar, and $\mathcal{A}$ is a set. $\mathbf{A}^T$ is the transpose of $\mathbf{A}$. $[\mathbf{A}]_{r,:}$ and $[\mathbf{A}]_{:,c}$ are the rth row and c^{th} column of the matrix $\mathbf{A}$, respectively. $|\mathcal{A}|$ is the cardinality of the set $\mathcal{A}$. $\lfloor a \rfloor$ is the floor function of the scalar a. $\arg(z)$ is the phase angle of the complex number z.

I. DeepMIMO Parameters

TABLE I: DeepMIMO MATLAB Parameters

Symbol	Parameter
$M_{\text{BS},x}^b$	`params.num_ant_BS(b,1)`
$M_{\text{BS},y}^b$	`params.num_ant_BS(b,2)`
$M_{\text{BS},z}^b$	`params.num_ant_BS(b,3)`
$M_{\text{UE},x}$	`params.num_ant_UE(1)`
$M_{\text{UE},y}$	`params.num_ant_UE(2)`
$M_{\text{UE},z}$	`params.num_ant_UE(3)`
d_{BS}^b	`params.ant_spacing_BS(a)`
d_{UE}	`params.ant_spacing_UE`
γ_{BS}^b	`params.array_rotation_BS(b,1)`
β_{BS}^b	`params.array_rotation_BS(b,2)`
α_{BS}^b	`params.array_rotation_BS(b,3)`
γ_{UE}	`params.array_rotation_UE(1)`
β_{UE}	`params.array_rotation_UE(2)`
α_{UE}	`params.array_rotation_UE(3)`
BW	`params.bandwidth`
L	`params.num_paths`
K	`params.num_OFDM`

TABLE II: DeepMIMO Python Parameters

Symbol	Parameter
$M_{\text{BS},x}^b$	`params['bs_antenna'][b-1]['shape'][0]`
$M_{\text{BS},y}^b$	`params['bs_antenna'][b-1]['shape'][1]`
$M_{\text{BS},z}^b$	`params['bs_antenna'][b-1]['shape'][2]`
$M_{\text{UE},x}$	`params['ue_antenna']['shape'][0]`
$M_{\text{UE},y}$	`params['ue_antenna']['shape'][1]`
$M_{\text{UE},z}$	`params['ue_antenna']['shape'][2]`
d_{BS}^b	`params['bs_antenna'][b-1]['spacing']`
d_{UE}	`params['ue_antenna']['spacing']`
γ_{BS}^b	`params['bs_antenna'][b-1]['rotation'][0]`
β_{BS}^b	`params['bs_antenna'][b-1]['rotation'][1]`
α_{BS}^b	`params['bs_antenna'][b-1]['rotation'][2]`
γ_{UE}	`params['ue_antenna']['rotation'][0]`
β_{UE}	`params['ue_antenna']['rotation'][1]`
α_{UE}	`params['ue_antenna']['rotation'][2]`
BW	`params['OFDM']['bandwidth']`
L	`params['num_paths']`
K	`params['OFDM']['subcarriers']`

Tables I and II summarize the symbols adopted for the DeepMIMO dataset input parameters, $\mathcal{S}$. These parameters can be controlled by the DeepMIMO end-user and they can be defined for every active basestation (BS) b and every receive user u as follows:

- Number of BS antennas, $M_{\text{BS}}^b = M_{\text{BS},x}^b M_{\text{BS},y}^b M_{\text{BS},z}^b$;
- Number of user antennas, $M_{\text{UE}} = M_{\text{UE},x} M_{\text{UE},y} M_{\text{UE},z}$;
- BS antenna array spacing, d_{BS}^b;
- User antenna array spacing, d_{UE};
- BS rotation angles, $\left(\gamma_{\text{BS}}^b, \beta_{\text{BS}}^b, \alpha_{\text{BS}}^b\right)$;
- User rotation angles, $\left(\gamma_{\text{UE}}, \beta_{\text{UE}}, \alpha_{\text{UE}}\right)$;
- System bandwidth, BW;
- Number of channel paths, L;
- Number of OFDM subcarriers, K; *and*
- Set of OFDM subcarriers at which the channels need to be calculated, $\mathcal{K}$.

The antenna array spacing is defined in terms of the operating wavelength, $\lambda = \frac{\varsigma}{f_c}$, where ς is the speed of light and f_c is the operating frequency; e.g., $d_{\text{BS}}^b = 0.5$ indicates a BS antenna array spacing of 0.5λ. The antenna array rotations are explained in more detail in Appendix A. The set $\mathcal{K}$ can be defined using the following Deep-MIMO parameters: (a) K, (b) `params.OFDM_limit`, and (c) `params.OFDM_sampling_factor`.

II. Ray-tracing outputs

Any BS-user or BS-BS link is defined as the wireless channel between (a) a BS transmitter (TX) and (b) a user receiver or a BS receiver (RX). For every transmitter t and every receiver r, and for every channel path ℓ, the ray-tracing outputs are as follows:

- Azimuth/zenith angles of departure (AoDs) from the TX, $(\psi_{\text{az}}^{t,r}, \psi_{\text{ze}}^{t,r})$;
- Azimuth/zenith angles of arrival (AoAs) at the RX, $(\omega_{\text{az}}^{t,r}, \omega_{\text{ze}}^{t,r})$;
- Receive channel power of the channel path, $P_\ell^{t,r}$;
- Receive channel phase of the channel path, $\vartheta_\ell^{t,r}$;
- Propagation delay of the channel path, $\tau_\ell^{t,r}$;
- Distance between the TX and the RX, $\text{dist}^{t,r}$;
- Path loss from the TX to the RX, $\text{PL}^{t,r}$;
- TX location, $\mathbf{p}^t = \left(p_x^t, p_y^t, p_z^t\right)$;
- RX location, $\mathbf{p}^r = \left(p_x^r, p_y^r, p_z^r\right)$; *and*
- Line-of-sight status, $\text{LOS}^{t,r} \in \{1, 0, -1\}$.

The locations are expressed in global rectangular coordinates, relative to the considered ray-tracing scenario 'R'. We have left the transmit power to be designed by the DeepMIMO end-user. For this reason, we define $\rho_\ell^{t,r}$ as the normalized version of the receive power values $P_\ell^{t,r}$ assuming $1\,\text{W}$ of transmit power. In addition, the TX/RX location represents the position of the first antenna element in the transmit/receive antenna array. Next, we show how these parameters can be used to construct the channel matrix between the transmit BS (TX) and the receive user/BS (RX).

III. DeepMIMO Channel Generation

Given the DeepMIMO dataset parameters in Section I and the ray-tracing outputs in Section II, the DeepMIMO dataset

generation code can construct the channels (a) between the specified active BSs (BS-BS link), and (b) between the specified active BS and active users (BS-user link). In this section, we formulate the adopted channel models for these links. Let the set of active users be denoted by $\mathcal{U} = \{u_1, \ldots, u_i, \ldots u_U\}$ and set of active BSs by $\mathcal{B} = \{b_1, \ldots, b_B\}$. The transmitters can only be BSs whereas the receivers can be either users or BSs. We define then the frequency-domain and time-domain channel models for any transmitter (TX) $t \in \mathcal{B}$, and any receiver (RX) $r \in \{\mathcal{B} \cup \mathcal{U}\}$, using a change of notation in the DeepMIMO parameters as follows:

- Number of TX antennas, $N^t = M_{\mathrm{BS}}^t$;
- TX array dimensions, $\mathbf{n}^t = \left[N_x^t, N_y^t, N_z^t\right]$;
- Number of RX antennas, $N^r \in \{M_{\mathrm{BS}}^r, M_{\mathrm{UE}}\}$;
- RX array dimensions, $\mathbf{n}^r = \left[N_x^r, N_y^r, N_z^r\right]$;
- TX antenna array spacing, $d^t = d_{\mathrm{BS}}^t$;
- RX antenna array spacing, $d^r \in \{d_{\mathrm{BS}}^r, d_{\mathrm{UE}}\}$;
- TX rotation angles, $(\gamma^t, \beta^t, \alpha^t) = (\gamma_{\mathrm{BS}}^t, \beta_{\mathrm{BS}}^t, \alpha_{\mathrm{BS}}^t)$; *and*
- RX rotation angles, $(\gamma^r, \beta^r, \alpha^r) \in \{(\gamma_{\mathrm{BS}}^r, \beta_{\mathrm{BS}}^r, \alpha_{\mathrm{BS}}^r),$ $(\gamma_{\mathrm{UE}}, \beta_{\mathrm{UE}}, \alpha_{\mathrm{UE}})\}$.

A. Frequency-Domain Channel Model

The DeepMIMO dataset generation code can construct the frequency-domain channel matrix, $\mathbf{H}_k^{t,r} \in \mathbb{C}^{N^r \times N^t}$, for any transmitter t and receiver r, and on each subcarrier $k \in \mathcal{K}$. Let the delay-d channel matrix between the transmitter t and receiver r be defined as $\boldsymbol{H}_d^{t,r} \in \mathbb{C}^{N^r \times N^t}$, where $d \in \{0, 1, \ldots, D-1\}$. D denotes the maximum delay-d channel length, such that $D = K$. The delay-d channel matrix can then be formulated as

$$
\boldsymbol{H}_d^{t,r} = \sum_{\ell=1}^{L} \left[\sqrt{\frac{\rho_\ell^{t,r}}{K}} \mathrm{e}^{j\vartheta_\ell^{t,r}} p\left(dT_{\mathrm{S}} - \tau_\ell^{t,r}\right) \times \right.
$$
$$
\left. \mathbf{a}_{\mathrm{R}}\left(\omega_{\mathrm{az}}^{t,r}, \omega_{\mathrm{ze}}^{t,r}\right) \mathbf{a}_{\mathrm{T}}\left(\psi_{\mathrm{az}}^{t,r}, \psi_{\mathrm{ze}}^{t,r}\right)^T \right], \quad (1)
$$

where $d \in \{0, 1, \ldots, D-1\}$. $p(\tau)$ represents the receive low-pass filtering function for T_{S}-spaced signaling evaluated at τ seconds. The choice of $p(\tau)$ function is controlled by the DeepMIMO end-user. The sampling time is designed such that $T_{\mathrm{S}} = 1/\mathrm{BW}$.

The transmit and receive array response vectors are denoted by $\mathbf{a}_{\mathrm{T}}\left(\psi_{\mathrm{az}}^{t,r}, \psi_{\mathrm{ze}}^{t,r}\right)$ and $\mathbf{a}_{\mathrm{R}}\left(\omega_{\mathrm{az}}^{t,r}, \omega_{\mathrm{ze}}^{t,r}\right)$, which are defined as

$$
\mathbf{a}_{\mathrm{T}}\left(\psi_{\mathrm{az}}^{t,r}, \psi_{\mathrm{ze}}^{t,r}\right) = \mathbf{a}\left(\psi_{\mathrm{az}}^{t,r}, \psi_{\mathrm{ze}}^{t,r}, d^t, \mathbf{n}^t\right), \quad (2)
$$
$$
\mathbf{a}_{\mathrm{R}}\left(\omega_{\mathrm{az}}^{t,r}, \omega_{\mathrm{ze}}^{t,r}\right) = \mathbf{a}\left(\omega_{\mathrm{az}}^{t,r}, \omega_{\mathrm{ze}}^{t,r}, d^r, \mathbf{n}^r\right). \quad (3)
$$

The generic array response vector $\mathbf{a}(\Phi, \Theta, \mathsf{d}, \mathbf{n})$ can be expressed as

$$
\mathbf{a}(\Phi, \Theta, \mathsf{d}, \mathbf{n}) = \mathbf{a}_z(\Theta, \mathsf{d}, N_z) \otimes \quad (4)
$$
$$
\mathbf{a}_y(\Phi, \Theta, \mathsf{d}, N_y) \otimes \mathbf{a}_x(\Phi, \Theta, \mathsf{d}, N_x). \quad (5)
$$

where $\mathbf{n} = [N_x, N_y, N_z]$. The functions $\mathbf{a}_x(.), \mathbf{a}_y(.)$, and $\mathbf{a}_z(.)$ represent the elemental array response vectors in the x, y, and z directions, and are expressed as

$$
\mathbf{a}_x(\Phi, \Theta, \mathsf{d}, N_x) = \left[1, \mathrm{e}^{jk\mathsf{d}\sin(\Theta)\cos(\Phi)}, \ldots \right.
$$
$$
\left. \ldots, \mathrm{e}^{jk\mathsf{d}(N_x-1)\sin(\Theta)\cos(\Phi)}\right]^T, \quad (6)
$$

$$
\mathbf{a}_y(\Phi, \Theta, \mathsf{d}, N_y) = \left[1, \mathrm{e}^{jk\mathsf{d}\sin(\Theta)\sin(\Phi)}, \ldots \right.
$$
$$
\left. \ldots, \mathrm{e}^{jk\mathsf{d}(N_y-1)\sin(\Theta)\sin(\Phi)}\right]^T, \quad (7)
$$

$$
\mathbf{a}_z(\Theta, \mathsf{d}, N_z) = \left[1, \mathrm{e}^{jk\mathsf{d}\cos(\Theta)}, \ldots \right.
$$
$$
\left. \ldots, \mathrm{e}^{jk\mathsf{d}(N_z-1)\cos(\Theta)}\right]^T. \quad (8)
$$

In case of applying any 3D rotation to the TX and/or RX antenna arrays, the azimuth/zenith angles of departure/arrival on the right hand sides of (2) and (3), $\psi_{\mathrm{az}}^{t,r}, \psi_{\mathrm{ze}}^{t,r}, \omega_{\mathrm{az}}^{t,r}, \omega_{\mathrm{ze}}^{t,r}$, are replaced by the angles $\dot{\psi}_{\mathrm{az}}^{t,r}, \dot{\psi}_{\mathrm{ze}}^{t,r}, \dot{\omega}_{\mathrm{az}}^{t,r}, \dot{\omega}_{\mathrm{ze}}^{t,r}$, which are formulated in detail in Appendix A.

Given the delay-d channel, the frequency-domain channel matrix, $\mathbf{H}_k^{t,r} \in \mathbb{C}^{N^r \times N^t}$, between the transmitter t and receiver r, can then be formulated as

$$
\mathbf{H}_k^{t,r} = \sum_{d=0}^{D-1} \boldsymbol{H}_d^{t,r} \mathrm{e}^{-j\frac{2\pi kd}{K}}, \ k \in \mathcal{K}. \quad (9)
$$

Special case: Assuming no receive low-pass filtering function such that $p(\tau) = \delta(\tau)$, the frequency-domain channel matrix $\mathbf{H}_k^{t,r} \in \mathbb{C}^{N^r \times N^t}$ can then be directly generated by

$$
\mathbf{H}_k^{t,r} = \sum_{\ell=1}^{L} \sqrt{\frac{\rho_\ell^{t,r}}{K}} \mathrm{e}^{j\varphi_{\ell,k}^{t,r}} \mathbf{a}_{\mathrm{R}}\left(\omega_{\mathrm{az}}^{t,r}, \omega_{\mathrm{ze}}^{t,r}\right) \mathbf{a}_{\mathrm{T}}\left(\psi_{\mathrm{az}}^{t,r}, \psi_{\mathrm{ze}}^{t,r}\right)^T, \quad (10)
$$

$$
\varphi_{\ell,k}^{t,r} = \vartheta_\ell^{t,r} - \frac{2\pi k}{K} \frac{\tau_\ell^{t,r}}{T_{\mathrm{S}}}. \quad (11)
$$

Generated frequency-domain channel matrix: Let $\mathbf{H}^{t,r} \in \mathbb{C}^{N^r \times N^t \times |\mathcal{K}|}$ be the 3D frequency-domain channel matrix between the transmitter t and receiver r. The 3D channel matrix $\mathbf{H}^{t,r}$ can then be constructed as

$$
\left[\mathbf{H}^{t,r}\right]_{:,:,k} = \mathbf{H}_k^{t,r}, \ \forall k \in \mathcal{K}. \quad (12)
$$

To access the frequency-domain channel matrix, the Deep-MIMO parameter `generate_OFDM_channels` needs to be set to unity. The channel matrix $\mathbf{H}^{t,r}$ between the active transmit BS b_i and the active user u_j can be accessed by using `DeepMIMO_dataset{i}.user{j}.channel` after running the DeepMIMO dataset generation code. Similarly, the channel between the active transmit BS b_i and the active receive BS b_j can be accessed by using `DeepMIMO_dataset{i}.basestation{j}.channel`.

Output channel structure: The generated frequency-domain channels in the DeepMIMO dataset are structured as follows:

- BS-user channel, $M_{\mathrm{UE}} \times M_{\mathrm{BS}}^t \times |\mathcal{K}|$; *and*
- BS-BS channel, $M_{\mathrm{BS}}^r \times M_{\mathrm{BS}}^t \times |\mathcal{K}|$;

For example, assume just one active transmit BS, employing a uniform planar array (UPA), is communicating with a receive

user adopting a uniform linear array (ULA). Let the Deep-MIMO parameters be: (a) $\mathbf{n}^t = [1, 6, 3]$; (b) $\mathbf{n}^r = [1, 3, 1]$; (c) $\mathcal{K} = \{0, 16, 32, 48, 64\}$. The dimensions of the generated 3D frequency-domain channel matrix are $3 \times 18 \times 5$.

B. Time-Domain Channel Model

The DeepMIMO dataset generation code can also construct the time-domain channel matrix, $\mathbf{H}_\ell^{t,r} \in \mathbb{C}^{N^r \times N^t}$ for every active transmitter t and receiver r, and for each channel path $\ell \in \{1, \ldots, L\}$. The time-domain channel matrix can then be formulated as

$$\mathbf{H}_\ell^{t,r} = \sqrt{\frac{\rho_\ell^{t,r}}{K}} \, \mathrm{e}^{j \vartheta_\ell^{t,r}} \mathbf{a}_{\mathrm{R}}\left(\omega_{\mathrm{az}}^{t,r}, \omega_{\mathrm{ze}}^{t,r}\right) \, \mathbf{a}_{\mathrm{T}}\left(\psi_{\mathrm{az}}^{t,r}, \psi_{\mathrm{ze}}^{t,r}\right)^T,$$
$$\ell \in \{1, \ldots, L\}. \qquad (13)$$

Generated time-domain channel matrix: Let $\mathbf{H}^{t,r} \in \mathbb{C}^{N^r \times N^t \times L}$ be the 3D time-domain channel matrix between the transmitter t and receiver r. The 3D channel matrix $\mathbf{H}^{t,r}$ can then be constructed as

$$\left[\mathbf{H}^{t,r}\right]_{:,:,\ell} = \mathbf{H}_\ell^{t,r}, \; \forall \ell \in \{1, \ldots, L\}. \qquad (14)$$

To access the time-domain channel matrix, the DeepMIMO parameter `generate_OFDM_channels` needs to be set to zero. The channel matrix $\mathbf{H}^{t,r}$ between the active transmit BS b_i and the active user u_j can be accessed by using `DeepMIMO_dataset{i}.user{j}.channel` after running the DeepMIMO dataset generation code. Similarly, the channel between the active transmit BS b_i and the active receive BS b_j can be accessed by using `DeepMIMO_dataset{i}.basestation{j}.channel`.

Output channel structure: The generated time-domain channels in the DeepMIMO dataset are structured as follows:

- BS-user channel, $M_{\mathrm{UE}} \times M_{\mathrm{BS}}^t \times L$; *and*
- BS-BS channel, $M_{\mathrm{BS}}^r \times M_{\mathrm{BS}}^t \times L$;

For example, assume just one active transmit BS, employing a uniform planar array (UPA), is communicating with a receive user adopting a uniform linear array (ULA). Let the Deep-MIMO parameters be: (a) $\mathbf{n}^t = [1, 6, 3]$; (b) $\mathbf{n}^r = [1, 3, 1]$; (c) $L = 10$. The dimensions of the generated 3D time-domain channel matrix are $3 \times 18 \times 10$. Notice that both the 3D frequency-domain and time-domain channel matrices, $\mathbf{H}^{t,r}$ and $\mathbf{H}^{t,r}$, have the same first two dimensions $N^r \times N^t$, which represents the various combinations of the receive and transmit antenna element pair. Next, we explain how these row/column indices map into the structure of the receive/transmit uniform antenna array.

C. DeepMIMO Antenna Map

In this subsection, we explain how the row/column indices, of the generated DeepMIMO channel matrices, map into the structure of the receive/transmit uniform antenna array. Given the DeepMIMO parameter defined in Section III, The first two dimensions of the generated 3D channel matrix are $N^r \times N^t$. For the transmit antenna array, the N^t column indices can be mapped into the TX antenna elements as follows.
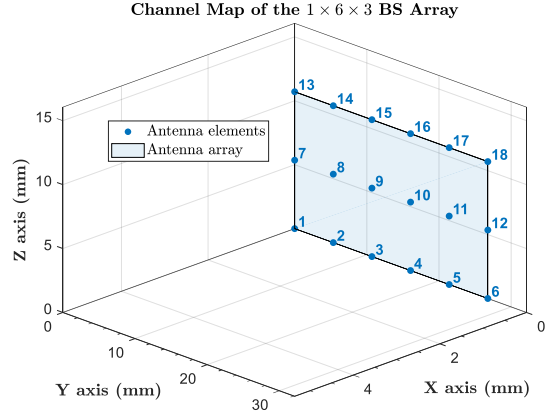

Fig. 1: An antenna map example of a 6×3 UPA BS is illustrated. The figure shows how the 18 column indices, of the generated 3D channel matrices, can be assigned to the TX antenna elements.

- The first index corresponds to the antenna element with the smallest (x, y, z) coordinates in the array.
- The last index corresponds to the antenna element with the largest (x, y, z) coordinates in the array.
- The order of the antenna elements follows increments based on the rectangular (x, y, z) coordinates.
- the column index number of the antenna element increases:
  1) by first moving along the positive x direction,
  2) then by moving along the positive y direction, and
  3) at last by moving along the positive z direction.

The row indices can be also mapped into the RX antenna element indices in a similar manner. Note that the coordinates of the first antenna element in the transmit/receive antenna array is the same as the TX/RX locations, $\mathbf{p}^t$ or $\mathbf{p}^r$. The antenna map indices of an antenna array can be generated using the function `antenna_channel_map` in MATLAB/Octave [1] or the function `DeepMIMO.ant_indices` in Python [2].

For example, assume just one active transmit BS, employing a uniform planar array (UPA), is communicating with a single-antenna receive user. Let the DeepMIMO parameters be: (a) $\mathbf{n}^t = [1, 6, 3]$, $d^t = 0.5$; (b) $\mathbf{n}^r = [1, 1, 1]$, $d^r = 0.5$. Let the operating frequency be $f_{\mathrm{c}} = 28 \, \mathrm{GHz}$. The first two dimensions of the generated 3D channel matrix are 1×18. Fig. 1 illustrates how the 18 column indices can be mapped into the TX antenna elements.

APPENDIX A
ANTENNA ARRAY 3D ROTATION MODEL

Generally, a 3D rotation can be applied to a transmit or receive antenna array to control its orientation. The array/panel orientation control can be useful for multiple reasons, which include among others: (a) changing the sector pointing direction for an antenna array, or (b) applying downward or upward tilting to an antenna array to face BSs or users located at a different elevation, or (c) arbitrary placement of the user equipment. In this section, we formulate the 3D rotation for an antenna array using three rotation angles around the $x, y,$
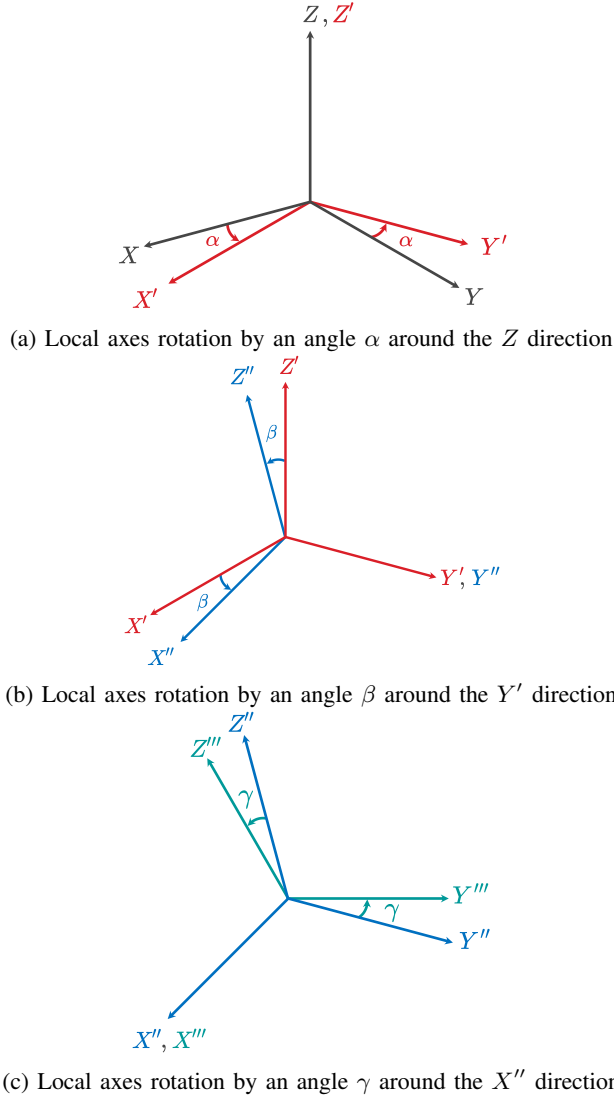
(a) Local axes rotation by an angle α around the Z direction

(b) Local axes rotation by an angle β around the Y' direction

(c) Local axes rotation by an angle γ around the X'' direction

Fig. 2: These figures illustrates the sequence of intrinsic elemental rotations adopted in the DeepMIMO channel generation process.

and z axes. For more details on the derivations, we refer the reader to [3].

In DeepMIMO, we adopt the rotation definition derived in the 3GPP TR 38.901 version 16.1.0 Release 16 specifications [3]. These rotations are defined as intrinsic rotations, which occur around the three axes of a local coordinate system attached to a rotation object. This means that the local axes change their orientation after each elemental rotation to the object. In such case of intrinsic rotations, the rotation angles are commonly referred to as the *Tait–Bryan angles*. More specifically, let x, y, z denote the fixed global axes used in the rectangular coordinates. Let X, Y, Z denote the local axes attached to a rotation antenna array. Before applying any intrinsic rotation to the antenna array, the local X, Y, Z axes coincide with the fixed global x, y, z axes. After applying an intrinsic rotation, the pointing directions of the local axes would change whereas the pointing directions of the global x, y, z axes remain fixed.

In the DeepMIMO channel generation process, the following sequence of intrinsic elemental rotations is illustrated in Fig. 2 and is applied as follows:

1) Apply a rotation by an angle α around the Z direction, as shown in Fig. 2a;
2) Apply a rotation by an angle β around the new Y' direction (after the first rotation by α), as shown in Fig. 2b;
3) Apply a rotation by an angle γ around the new X'' direction (after the first rotation by α and the second rotation by β), as shown in Fig. 2c.

Before applying any rotation, we refer to the local axes as the X, Y, Z axes. After the first elemental rotation, the local axes are then referred to as the X', Y', Z' axes, where the Z' axis coincides with the Z axis. After the second elemental rotation, the local axes are then referred to as the X'', Y'', Z'' axes, where the Y'' axis coincides with the Y' axis. After the third elemental rotation, the local axes are then referred to as the X''', Y''', Z''' axes, where the X''' axis coincides with the X'' axis. Note that the curl right-hand rule is followed in defining the rotation around the positive direction of an axis.

Now, we can formulate the effect of these rotations on the azimuth/zenith angles, which are used in constructing the transmit and receive array response vectors in (2) and (3). After applying all three elemental rotations, let the angles $\overset{\bullet}{\Phi}, \overset{\bullet}{\Theta}$ represent the azimuth and zenith angles relative to the X''', Y''', Z''' axes and let the angles Φ, Θ represent the azimuth and zenith angles relative to the global x, y, z axes. These updated angles $\overset{\bullet}{\Phi}, \overset{\bullet}{\Theta}$ can be formulated in terms of the original angles Φ, Θ and the rotation angles α, β, γ as

$$\overset{\bullet}{\Theta}(\Phi, \Theta, \alpha, \beta, \gamma) = \mathrm{acos}\,(\cos\beta\cos\gamma\cos\Theta + \sin\Theta\,(\sin\beta\cos\gamma\cos(\Phi - \alpha) - \sin\gamma\sin(\Phi - \alpha))), \quad (15)$$

$$\overset{\bullet}{\Phi}(\Phi, \Theta, \alpha, \beta, \gamma) = \arg(\chi + \jmath\kappa), \quad (16)$$

$$\chi = \cos\beta\sin\Theta\cos(\Phi - \alpha) - \sin\beta\cos\Theta, \quad (17)$$

$$\kappa = \cos\beta\sin\gamma\cos\Theta + \sin\Theta\,(\sin\beta\sin\gamma\cos(\Phi - \alpha) + \cos\gamma\sin(\Phi - \alpha)). \quad (18)$$

Relative to the DeepMIMO channel generation process, the definitions of the various angles are as follows:

- Original azimuth/zenith angles, $(\Phi, \Theta) \in \{(\psi_{\mathrm{az}}^{t,r}, \psi_{\mathrm{ze}}^{t,r}), (\omega_{\mathrm{az}}^{t,r}, \omega_{\mathrm{ze}}^{t,r})\}$;
- Updated azimuth/zenith angles, $(\overset{\bullet}{\Phi}, \overset{\bullet}{\Theta}) \in \{(\dot{\psi}_{\mathrm{az}}^{t,r}, \dot{\psi}_{\mathrm{ze}}^{t,r}), (\dot{\omega}_{\mathrm{az}}^{t,r}, \dot{\omega}_{\mathrm{ze}}^{t,r})\}$;
- Origin coordinates, $\mathbf{p} \in \{\mathbf{p}^t, \mathbf{p}^r\}$; *and*
- Rotation angles, $(\gamma, \beta, \alpha) \in \{(\gamma^t, \beta^t, \alpha^t), (\gamma^r, \beta^r, \alpha^r)\}$.

Finally, these antenna array rotations can be incorporated into the channel generation process by first computing the updated azimuth/zenith angles of departure/arrival, $\dot{\psi}_{\mathrm{az}}^{t,r}, \dot{\psi}_{\mathrm{ze}}^{t,r}, \dot{\omega}_{\mathrm{az}}^{t,r}, \dot{\omega}_{\mathrm{ze}}^{t,r}$. These updated angles then replace the original azimuth/zenith angles of departure/arrival, $\psi_{\mathrm{az}}^{t,r}, \psi_{\mathrm{ze}}^{t,r}, \omega_{\mathrm{az}}^{t,r}, \omega_{\mathrm{ze}}^{t,r}$, in the transmit and receive array response vector definitions, (2) and (3).

Array orientation visualization: To visualize the orientation of an antenna array using the rotation angles
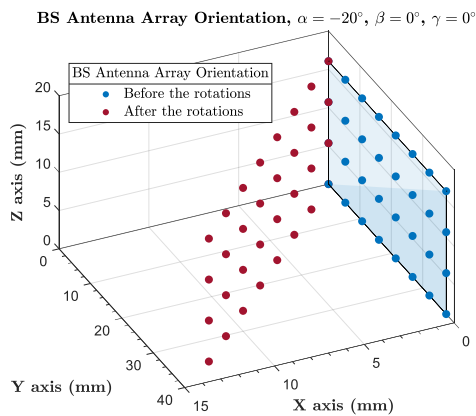
Fig. 3: An array orientation example of an 8×4 UPA BS is illustrated. The figure shows how the BS antenna array is positioned before and after applying a rotation by $\alpha = -20°$ around the Z direction.

α, β, γ, we refer the reader to Example 2 in the documentation page of the DeepMIMO v2 generator [1]. The antenna array orientation can be visualized using the function `Visualize_ant_array_orientation`. For example, assume an active transmit BS is equipped with an 8×4 UPA, where $\mathbf{n}^t = [1, 8, 4]$ and $d^t = 0.5$. Let the operating frequency be $f_c = 28\,\text{GHz}$. Assume the rotation angles are $(\gamma^t, \beta^t, \alpha^t) = (0°, 0°, -20°)$. Fig. 3 illustrates how the BS antenna array is positioned before and after applying a rotation by $\alpha = -20°$ around the Z direction. In addition, although the antenna elements located on the Z axis do not change positions after the rotation, as shown in the figure, these elements are now facing a different direction because of the rotation; i.e., their section facing directions has changed.

REFERENCES

[1] Wireless Intelligence Lab, "Documentation of the DeepMIMO v2 MATLAB Generator." [Online]. Available: https://deepmimo.net/versions/v2-matlab/

[2] ——, "Documentation of the DeepMIMO v2 Python Generator." [Online]. Available: https://deepmimo.net/versions/v2-python/

[3] 3GPP, "Technical Specification Group Radio Access Network; Study on channel model for frequencies from 0.5 to 100 GHz," 3rd Generation Partnership Project (3GPP), TR 38.901, Dec. 2019, version 16.1.0, Release 16. [Online]. Available: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3173