Semper, Sebastian; Döbereiner, Michael; Pawar, Sankalp; Landmann, Markus; Del Galdo, Giovanni:

eadf: representation of far-field antenna responses in Python

Original software publication

eadf: Representation of far-field antenna responses in Python[☆]

S. Semper [*,1], M. Döbereiner [2], S. Pawar [2], M. Landmann [2], G. Del Galdo [1]

EMS Research Group, TU Ilmenau, Germany

ABSTRACT

Understanding and analysis of antennas is crucial for the development of modern telecommunication technologies, like 5G or massive MIMO. Research in this area has the need for software to measure, analyze and test antenna systems. For instance, one needs a convenient way to transform large amounts of antenna measurement data into a concise representation. The herein described eadf package aims at providing this functionality. On top of that, it offers routines to interpolate the beam-pattern of an antenna as well as compressing it in spatial frequency domain. As such, it can be used for the development of high resolution parameter estimation algorithms and analysis of antenna beam-patterns. In summary, the package provides a flexible and extensible framework for dealing with antennas.

Code metadata

Current code version	v0.5
Permanent link to code/repository used for this code version	For example: https://github.com/ElsevierSoftwareX/SOFTX_2020_249
Code Ocean compute capsule	
Legal Code License	Apache License Version 2.0
Code versioning system used	git
Software code languages, tools, and services used	Python, NumPy, Travis, Readthedocs
Compilation requirements, operating environments & dependencies	
If available Link to developer documentation/manual	https://eadf.readthedocs.io/en/latest/
Support email for questions	sebastian.semper@tu-ilmenau.de

1. Motivation and significance

Antenna engineering plays a vital role in the application and advancement of telecommunications technologies. Several key areas of research and development like 5G, Massive MIMO, and radar benefit from antenna modeling and array signal processing techniques [1–4].

For the reliable application of antenna arrays in practical scenarios, one must account for the electromagnetic properties of the array in the form of its beam-pattern. The beam-pattern describes the complex and frequency-dependent radiation pattern in the far-field as a function of direction typically defined using the spherical coordinates (azimuth and co-elevation angles) [5]. The beam-pattern is dependent on polarization and frequency of the radiated electromagnetic wave. Although in principle the beam-pattern can be calculated using the solutions to Maxwell's equations [6], analytical expressions for describing the beam-pattern for every antenna design are usually not attainable. Hence, numerical simulations based on antenna models or even actual beam-pattern measurements are required for understanding the characteristics of an antenna system [6]. The precision of these simulations in depicting the antenna design depends on the accuracy of the simulated model, knowledge about the simulated materials, considerations of the radio environment, and so forth.

A concise representation of the beam-pattern is necessary for applications like high-performance antenna design, radio channel modeling [7] and high-resolution parameter estimation [8]. For

these applications one of the most important necessities is to be able to efficiently interpolate the beam-pattern at an arbitrary angular point, i.e. it is usually sampled or simulated for a discrete set of incident angles. Thus, we need to retrieve a continuous description of the so-called array manifold and its derivatives. For instance, using the latter, performance analysis based on metrics like the Cramèr–Rao Lower Bound (CRLB) [9,10] becomes possible, which allows us to quantify the minimum variance of an unbiased estimator based on an antenna array. Moreover, derivatives of the beam-pattern enable the implementation of gradient-based parameter estimation techniques, which usually represent an estimator that can be evaluated utilizing the CRLB.

To these ends, a robust framework for the effective generation, representation, and modification of the frequency-dependent polarimetric antenna beam-pattern referred to as the Effective Aperture Distribution Function (EADF) [11, Section 4.3.3], [12,13] is indispensable. The herein described Python-based `eadf` package provides the necessary means to handle antenna beam-patterns which are measured, simulated, or can be described using existing analytical expressions, to tackle the above-mentioned tasks.

2. Software description

As a starting point, we introduce the so called polarimetric array manifold function a being defined as

$$a : [0, \pi] \times [0, 2\pi] \times \mathbb{R}^+ \to \mathbb{C}^{2 \times N_{\text{port}}}, (\theta, \varphi, f) \mapsto a(\theta, \varphi, f), \quad (1)$$

which models the response of an antenna array to a plane wave impinging from the direction (θ, φ). The response is obtained over $2 \times N_{\text{port}}$ outputs, accounting for the two polarizations (horizontal and vertical) and the number of ports N_{port}. Here, θ and φ denotes the co-elevation and azimuthal angles of the spherical coordinate system respectively. While, f denotes the frequency of the impinging wave. Additionally $\mathbb{R}^+$ denotes the non-negative real numbers and $\mathbb{C}$ denotes complex numbers. It can be shown that the function a suffices to describe the antenna response to a plane wave. One can usually attain samples of a in the form,

$$A_{i,j,k} = a(i \cdot \Delta_\theta, j \cdot \Delta_\varphi, f_k),$$

where $\Delta_\theta = \pi / N_{\text{coele}}$, $\Delta_\varphi = 2\pi / N_{\text{azim}}$ and $0 \leqslant i < N_{\text{coele}}$, $0 \leqslant j < N_{\text{azim}}$, $0 \leqslant k < N_{\text{freq}}$. Since, we wish to calculate a Discrete Fourier Transform (DFT) along the two angles as described below, we assume a uniform sampling in the 2D angular grid, but not in the excitation frequency dimension. So formally, the complete antenna array response is represented by means of a multi-dimensional array $A \in \mathbb{C}^{N_{\text{coele}} \times N_{\text{azim}} \times N_{\text{freq}} \times 2 \times N_{\text{port}}}$.

As physics dictates, the function a is naturally periodic in its second argument φ. If we extend a to a function $\hat{a} : [0, \pi) \times [0, 2\pi) \times \mathbb{R}^+ \to \mathbb{C}^{2 \times N_{\text{port}}}$, which is also periodic in θ it reads as,

$$\hat{a}(\theta, \varphi, f) = \begin{cases} a(\theta, \varphi, f), & \theta \leqslant \pi \\ -a(2\pi - \theta, \varphi + \pi, f), & \theta > \pi. \end{cases} \quad (2)$$

The domain of $\hat{a}$ covers the sphere twice, however, for $\theta > \pi$ the polarization of the impinging waves has to be altered, hence the minus $(-)$ sign and we have to move to the right corresponding point on the original sphere, hence the $\varphi + \pi$. This naturally extends the sampled data A to its periodified version $\hat{A} \in \mathbb{C}^{2 \cdot N_{\text{coele}} - 1 \times N_{\text{azim}} \times N_{\text{freq}} \times 2 \times N_{\text{port}}}$, such that it is periodic in its first two dimensions.

For a given excitation frequency, polarization, and antenna port, if we consider a 2D slice $S_{k,\ell,m} = \hat{A}_{\cdot,\cdot,k,\ell,m} \in \mathbb{C}^{2 \cdot N_{\text{coele}} - 1 \times N_{\text{azim}}}$, we can apply a 2D DFT to it to obtain $T_{k,\ell,m} \in \mathbb{C}^{2 \cdot N_{\text{coele}} - 1 \times N_{\text{azim}}}$, which we call the *effective aperture distribution function* of the antenna's beam-pattern.
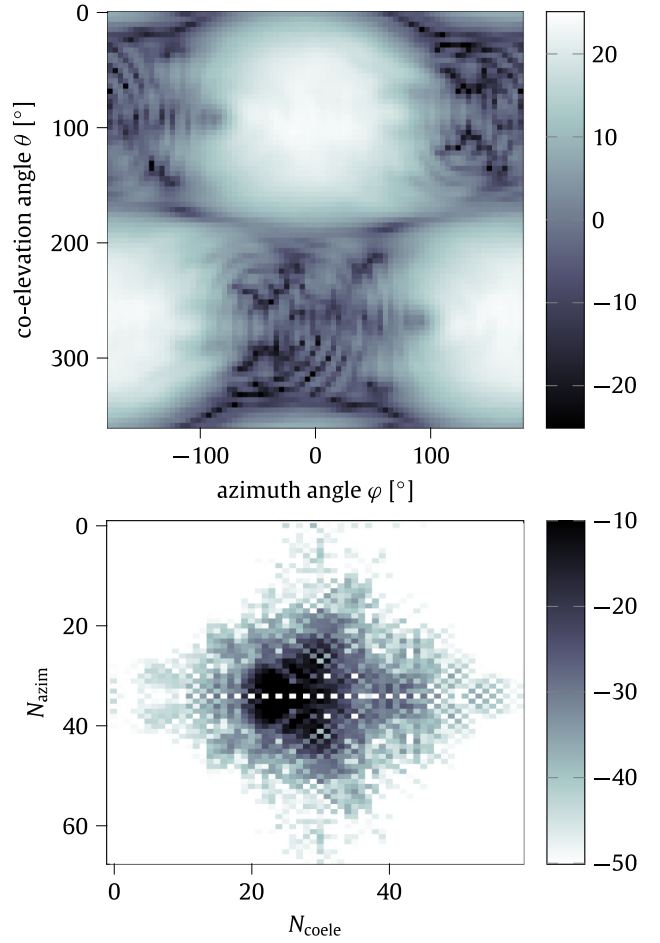
Fig. 1. Periodified beam-pattern slice $S_{k,\ell,m}$ for a certain polarization, frequency, and port of an antenna array (top), the corresponding 2D (60 × 68) Fourier co-efficients $T_{k,\ell,m}$ (bottom).

The remarkable property of $T_{k,\ell,m}$ now is that if N_{coele} and N_{azim} have been chosen large enough, we can apply the *continuous* 2D inverse DFT, such that,

$$\hat{a}(\theta, \varphi, f_k)_{\ell,m} = \exp(\jmath\theta\mu_\theta)^t \cdot T_{k,\ell,m} \cdot \exp(\jmath\varphi\mu_\varphi), \quad (3)$$

where the two vectors μ_θ and μ_φ are defined as,

$$\mu_\theta = [-N_{\text{coele}}, \ldots, +N_{\text{coele}}]$$

and

$$\mu_\varphi = [-N_{\text{azim}}/2, \ldots, +N_{\text{azim}}/2].$$

This means that we have derived an *interpolation* of the complete beam-pattern for arbitrary angles just by utilizing the finite number of elements in $S_{k,\ell,m}$ or $T_{k,\ell,m}$, and thus have derived an *analytical* description of the antenna's far-field beam-pattern. Examples for both $S_{k,\ell,m}$ and $T_{k,\ell,m}$ taken from measurement data can be found in Fig. 1.

This analytical representation has several advantages and applications. For instance, it allows us to calculate the derivatives of the beam-pattern with respect to the angles via

$$\frac{\partial}{\partial\theta} \hat{a}(\theta, \varphi, f_k)_{\ell,m} = (\text{diag}(\jmath\mu_\theta) \cdot \exp(\jmath\theta\mu_\theta))^t \cdot T_{k,\ell,m} \cdot \exp(\jmath\varphi\mu_\varphi),$$

$$\frac{\partial}{\partial\varphi} \hat{a}(\theta, \varphi, f_k)_{\ell,m} = \exp(\jmath\theta\mu_\theta)^t \cdot T_{k,\ell,m} \cdot (\text{diag}(\jmath\mu_\varphi) \cdot \exp(\jmath\varphi\mu_\varphi)).$$

Naturally, one can also compute higher order derivatives or mixed derivatives.

In summary, the EADF enables us to interpolate the beam-pattern and its derivatives. The herein described `eadf` package provides the capability to exploit the above-described theoretical findings and the associated preprocessing routines in an accessible, convenient yet computationally efficient framework.

2.1. Software architecture

The whole architecture of the package revolves around the EADF class, where each instance represents an antenna array. It handles the necessary preprocessing of the input data A and allows to specify some metadata about $\hat{A}$, like the sampled co-elevation and azimuth angles, the sampled frequencies f_k, and the number of antenna array elements. This EADF object can be completely initialized with the data A and minimal metadata. As such, it also provides the routines to evaluate the beam-pattern and its derivatives using the functions `EADF.pattern`, `EADF.gradient` for first-order derivatives and `EADF.hessian` for all second-order derivatives, see Section 3 for details.

For easy interfacing with other antenna modeling software (e.g. HFSS), various measurement systems (e.g. our research group's internal system), or just another computing language (e.g. Matlab), we collect different importers in the `importers` sub-module. Since the EADF object is simple to initialize, it is also very easy to write importers that convert given data to the format of the data array A. However, we also provide some theory-based importers. A non-trivial case for an importer occurs if the angles were not sampled on a uniform and thus full grid. This can easily happen because of mechanical constraints during the beam-pattern measurement of an antenna array. For this, we enable to approximate the non-uniformly sampled data by the so-called spherical harmonic functions, which then can be interpolated on the necessary uniform angular grid points.

Since our goal, among others, is to use the EADF object for high-resolution parameter estimators, it is of high priority to fully utilize available hardware. In some cases, the acquired measurement data has such a low dynamic range that calculations in single precision are sufficient since the accuracy is not limited by the datatype, but the data itself. In these cases, substantial speedups can be attained without sacrificing estimation accuracy. To this end, the `EADF_DTYPE` shell environment variable allows the choice between single or double floating-point precision.

Additionally, there are routines to generate the data array A for synthetic antenna arrays in the `arrays` sub-module, all of which produce an EADF object. The preprocessing routines, including periodification and DFT, are in a dedicated `preprocess` module.

2.2. Software functionalities

The architecture described above was designed with several goals for the functionality of the package. First, it is simple to import various kinds of measurement and simulation data that describe an antenna array with the provided importers. Second, the package allows to efficiently interpolate the complex polarimetric beam-pattern of an arbitrary antenna array from a finite number of samples. As such this represents the main feature of the package. This analytical representation also allows to calculate higher order derivatives of the beam-pattern.

Actually the transformation from $S_{k,\ell,m}$ to $T_{k,\ell,m}$ represents a form of compression, since usually the energy in $T_{k,\ell,m}$ is concentrated in a few entries. So, by removing rows or columns of the matrix $T_{k,\ell,m}$ in Eq. (3) with little or no signal energy one can achieve a form of de-noising of the beam-pattern, which also directly decreases the computational cost of the interpolation as well as the required storage for the fully preprocessed representation of the beam-pattern. This can be achieved by setting `EADF.compressionFactor` to some number $0 < \rho \leqslant 1$. If used in an high-resolution parameter estimator this can yield substantial runtime improvements.

Lastly, we provide routines to generate representations for idealized arrays, where each antenna element is either a dipole or a rectangular patch antenna. In both cases one can calculate the polarimetric and frequency dependent array response analytically. For instance, one can create an array where the elements are placed on a line with constant distance —- a so called Uniform Linear Array (ULA, `eadf.generateULA`), or a Uniform Circular Array (UCA, `eadf.generateUCA`), where elements are placed at uniform distance on a circle of given radius.

2.3. Sample code snippets analysis

The code in Listing 1 demonstrates how one can create a ULA based on the far-field pattern of a rectangular patch [6, 727-752] neglecting coupling effects between the elements, set its compression factor and then sample the pattern on a uniform 30×60, co-elevation $\times$ azimuth angular grid.
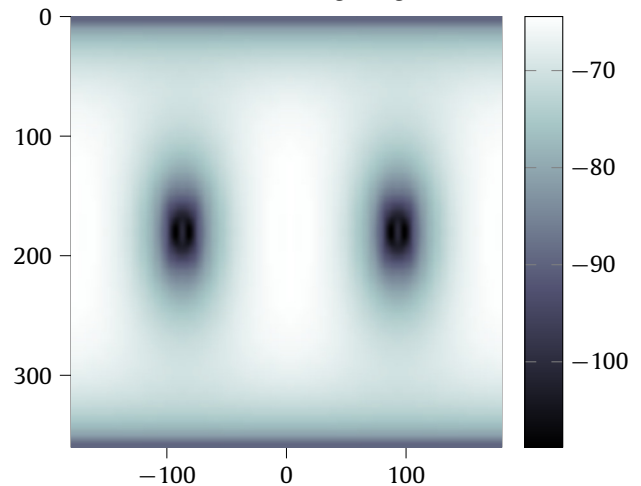


```python
import matplotlib.pyplot as plt
import numpy as np
import eadf


array = eadf.generateULA(
    3, 0.75,
    np.array([[30e3, 30e3, 5e3]]).T,
    np.linspace(5e9, 5e9, 1),
)
array.compressionFactor = 1  1e8


# pick out one port and one frequency
pattern = array.pattern(
    *eadf.toGrid(*eadf.sampleAngles(60, 60))
)[:, 0, 0, 0].reshape((60, 60), order="F")


plt.imshow(
    10 * np.log(np.abs(pattern)[1:]),
    extent=(180, 180, 180, +0), cmap="bone",
)
plt.colorbar()
plt.show()
```

Listing 1: The code (bottom) to produce the beam-pattern of a synthetic ULA antenna array. The plot (top) of the response of a single port.

3. Illustrative examples

The determination of the angle of arrival (AoA) of an electromagnetic wave is of importance for many present research
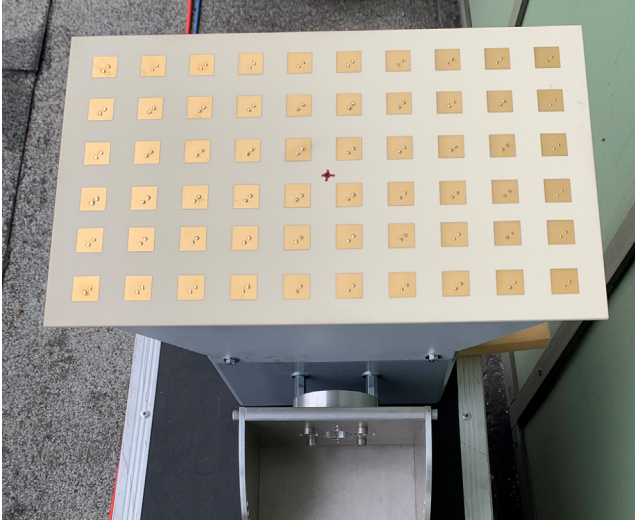
Fig. 2. 4 × 8 polarimetric uniform rectangular array (PURA) used for channel sounding measurements at 3.75 GHz to determine the AoA of various propagation paths. The outer ring of elements are dummy elements.

Fig. 3. Likelihood function for estimation of θ and φ, including the trace of the parameters during 10 gradient (red), and Newton's method (yellow) iterations respectively. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

topics related to wave propagation. For example, channel sounding measurement (see Fig. 2) is used to characterize the mobile radio channel for planning the distribution of new mobile base stations. Also, more and more radar applications use direction finding approaches to localize obstacles, e.g. in the automotive industry.

As described earlier, the EADF enables us to interpolate the beam-pattern of an antenna system for a wave impinging from a given angular point. Given this analytical representation of the antenna array, it is also possible to estimate the AoA of the impinging wave based on the data received by the antenna array.

For angle estimation at a given frequency f_0, the data model for a single impinging electromagnetic wave is given as, $s(\theta, \varphi) = \gamma \cdot \hat{a}(\theta, \varphi, f_0) \in \mathbb{C}^{N_{port}}$, with the polarimetric path weight $\gamma = [\gamma_H, \gamma_V] \in \mathbb{C}^{1 \times 2}$.

A common approach for the parameter estimation is to design a maximum-likelihood estimator, e.g. for i.i.d. Gaussian noise the estimator maximizes the log-likelihood function,

$$\mathcal{L}(x, \theta, \varphi) = -\frac{1}{\sigma^2} \cdot (x - \gamma \cdot \hat{a}(\theta, \varphi, f_0))^H \cdot (x - \gamma \cdot \hat{a}(\theta, \varphi, f_0)), \quad (4)$$

with respect to the desired parameters $[\theta, \varphi] \in \mathbb{R}^{2 \times 1}$. The measured data $x \in \mathbb{C}^{N_{port}}$ is affected by additive noise of variance σ^2.

The gradient ascent algorithm is a popular way for finding the maximum of a differentiable function. To find a local maximum of a function using gradient ascent, we take steps proportional to the gradient of the function at the current point. To ensure reaching the global maximum, a starting point close to the maximum is necessary, since the log-likelihood is usually non-convex and just locally convex around the global maximum. The iterative update of the parameters is obtained as,
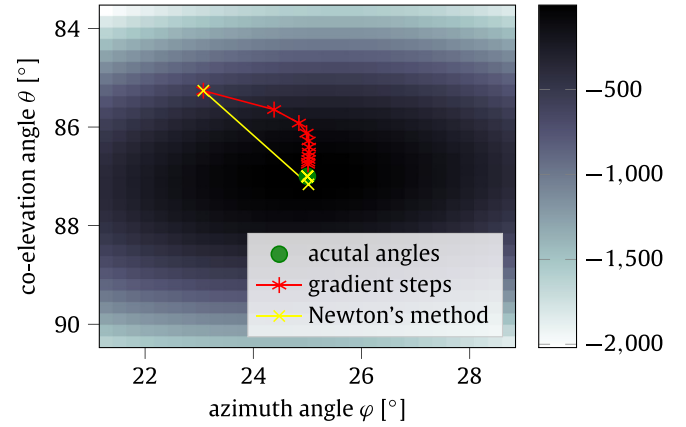
$$[\theta, \varphi]_{k+1} = [\theta, \varphi]_k + \varepsilon J_{\mathcal{L}}([\theta, \varphi]_k), \quad (5)$$

where $\varepsilon > 0$ is a step size and $J_{\mathcal{L}}([\theta, \varphi]_k) \in \mathbb{C}^{N_{port} \times 2}$ denote the first-order derivatives of the log-likelihood function

$$J_{\mathcal{L}}([\theta, \varphi]) = \left[\frac{\partial}{\partial \theta} \mathcal{L}(x, \theta, \varphi), \frac{\partial}{\partial \varphi} \mathcal{L}(x, \theta, \varphi) \right]. \quad (6)$$

Now, calculating the first-order derivative of the log-likelihood function with respect to θ we get,

$$\frac{\partial}{\partial \theta} \mathcal{L}(x, \theta, \varphi) = 2 \cdot \mathcal{R} \left\{ \gamma \cdot \frac{\partial}{\partial \theta} \hat{a}^H(\theta, \varphi, f_0) \cdot (x - \hat{a}(\theta, \varphi, f_0)) \right\} \quad (7)$$

and we see the influence of the derivative of the beam-pattern $\hat{a}(\theta, \varphi, f_0)$.

```
theta = theta0

for ii in range(numSteps):
    theta += eps * np.real(
        gamma
        * array.gradient(theta).conj()
        * (x  array.pattern(theta))
    )
```

Listing 2: Simplified gradient iteration

One can refine this idea, such that, instead of maximizing the log-likelihood function, one searches for a zero over the derivative of the function. Starting with a rough initial estimate of the actual AoA this task is suitable for using an iterative gradient-based optimization approach, e.g. the Newton's method. Based on the initial estimate of the parameters $[\theta, \varphi]_0$, the updated parameters are obtained as

$$[\theta, \varphi]_{k+1} = [\theta, \varphi]_k + H_{\mathcal{L}}^{-1}([\theta, \varphi]_k) \cdot J_{\mathcal{L}}([\theta, \varphi]_k). \quad (8)$$

The matrix with the second-order derivatives of the likelihood-function $H_{\mathcal{L}}([\theta, \varphi]) \in \mathbb{C}^{N_{port} \times 2}$ is calculated as a further derivative of $J_{\mathcal{L}}([\theta, \varphi])$, which also depends on the second-order derivatives of the beam-pattern.

As described in Section 2, the EADF-package provides functionality for the calculation of derivatives of the beam-pattern. A gradient-based optimization technique can be easily implemented as shown in simplified Pseudo-Code in Listing 2 for the first-order derivative case.

As an example, we created some data representing a wave impinging an 32-port antenna array from $\Theta = 87°$ and $\varphi = 25°$ angles. Starting with some initial estimates of the angles near the simulated ones, Newton's method yields the correct estimation of the angles after considerably few iterations. Fig. 3 shows the evolution of the estimation process over 10 iterations.

4. Impact

4.1. Research questions

The analytical representation of an antenna array allows us to directly quantify its performance through simulation of the estimation process as sketched out in Section 3 or by directly

studying the CRLB in various settings. Not only is it possible to evaluate the properties of a particular antenna array, but it is also possible to directly compare different arrays with each other. This can also greatly facilitate the design process of antenna measurement systems that are composed of one or more antenna arrays. Finally, the fact that the EADF object can represent the beam-pattern for a large band of operating frequencies, it is possible to develop high-resolution parameter estimation routines, which can operate on measurement data covering a wide frequency range.

The packages routines can directly be used in high-resolution parameter estimation routines [14–16], where one has to account for the beam-pattern of the involved antennas, while also allowing for high computation speed and high dimensional data to be handled. This will allow treating (for instance) channel estimation problems where data is sampled at increasingly higher rates, higher bandwidths, and with larger arrays. In channel estimation the beam-pattern models the influence of the antenna to the whole channel impulse response. For instance in a double directional model [17] the channel can be described as

$$H(f, t) = \sum_i \gamma_i \cdot a_{RX}(\theta_{RX,i}, \varphi_{RX,i}, f) \cdot U(f, t) \cdot a_{TX}(\theta_{TX,i}, \varphi_{TX,i}, f)$$

such that the functions a_{RX} and a_{TX} describe, the receive and transmit antennas contribution to the channel impulse response.

Additionally, the initial phase of data acquisition through a measurement system is greatly facilitated by the fact that there is no need to directly sample on the full uniform grid, but we rather offer methods to circumvent this by estimating missing values optimally in a least-squares sense. This can speed up certain measurement processes tremendously.

4.2. Daily practice of the users

The primary user base currently consists of scientists at our research lab that work on different topics associated with antenna engineering. For instance, when designing new antenna arrays that operate over a certain frequency band, one must study the associated complex impulse response of the array. Another example is antenna beam-pattern measurements which allow us to characterize the beam-pattern of a manufactured array in practical scenarios that can be validated through an evaluation routine that utilizes the EADF representation.

Additionally, for over-the-air testing [18,19] via the radiated two-stage method of some device under test (DUT), one has to embed the antenna array of the DUT into the calculations, which is carried out by this package.

Finally, we have researchers who are developing high-resolution parameter estimation routines. These are applied to a wide range of measurement data, acquired by the different antenna arrays, with high dimensional parameter spaces and large amounts of data. To this end, the package provides a simple interface to easily incorporate the antennas in the algorithms as well as in the analysis of their performance.

5. Conclusions

In addition to the described functionality, the package allows the interpolation of the input data along the frequency dimension via splines, thereby introducing a comprehensive framework for the analysis of modern antenna arrays. It provides the necessary tools to interpolate arbitrary complex vector-valued functions on a sphere, allowing its application to problems beyond the scope of antennas and wave propagation. Future versions of the package will provide features like the capability to rotate the beam-pattern along a designated axis by using vector spherical harmonics [20].

In summary, the `eadf` package has provided extensive utilization for antenna array signal processing applications. We illustrated how the concise representation of the antenna beam-pattern is a central building block in developing high-resolution parameter estimation algorithms and outlined how it can help researchers involved in topics with antennas.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

References

[1] Shafi M, Molisch AF, Smith PJ, Haustein T, Zhu P, De Silva P, et al. 5G: A tutorial overview of standards, trials, challenges, deployment, and practice. IEEE J Sel Areas Commun 2017;35(6):1201–21.

[2] Rappaport TS, Xing Y, MacCartney GR, Molisch AF, Mellios E, Zhang J. Overview of millimeter wave communications for fifth-generation (5G) wireless networks—With a focus on propagation models. IEEE Trans Antennas and Propagation 2017;65(12):6213–30.

[3] Zhang P, Chen J, Yang X, Ma N, Zhang Z. Recent research on massive MIMO propagation channels: A survey. IEEE Commun Mag 2018;56(12):22–9.

[4] Li J, Zhu S, Chen X, Lv L, Liao G, Yi M. Sparse recovery for bistatic MIMO radar imaging in the presence of array gain uncertainties. Int J Antennas Propag 2014.

[5] Van Trees HL. Optimum array processing (detection, estimation, and modulation theory, part IV). 2002.

[6] Balanis CA. Antenna theory: analysis and design. 2005.

[7] Steinbauer M, Molisch AF, Bonek E. The double-directional radio channel. IEEE Antennas Propag Mag 2001;43(4):51–63.

[8] Thomä RS, Landmann M, Richter A. RIMAX - A maximum likelihood framework for parameter estimation in multidimensional channel sounding. In: Proceedings of the ISAP 2004. Sendai, Japan; 2004.

[9] Stoica P, Larsson EG, Gershman AB. The stochastic CRB for array processing: a textbook derivation. IEEE Signal Process Lett 2001;8(5):148–50.

[10] Landmann M, Richter A, Thomä RS. DOA resolution limits in MIMO channel sounding. In: IEEE antennas and propagation society international symposium, vol. 2; 2004. p. 1708–11.

[11] Hansen JE, editor. Spherical near-field antenna measurements. Electromagnetic Waves, Institution of Engineering and Technology; 1988.

[12] Landmann M, Galdo GD. Efficient antenna description for MIMO channel modelling and estimation. In: 7th European conference on wireless technology, 2004; 2004. p. 217–20.

[13] Landmann M, Thomä RS. Common pitfalls in multidimensional high resolution channel parameter estimation. In: IEEE digital signal processing workshop; 2009. p. 314–9.

[14] Wang R, Renaudin O, Bas CU, Sangodoyin S, Molisch AF. High-resolution parameter estimation for time-varying double directional V2V channel. IEEE Trans Wireless Commun 2017;16(11):7264–75.

[15] Fleury BH, Yin X, Jourdan P, Stucki A. High-resolution channel parameter estimation for communication systems equipped with antenna arrays. IFAC Proc Vol 2003;36(16):97–102, 13th IFAC Symposium on System Identification (SYSID 2003), Rotterdam, The Netherlands, 27-29 August, 2003.

[16] Fleury BH, Jourdan P, Stucki A. High-resolution channel parameter estimation for MIMO applications using the SAGE algorithm. In: 2002 International zurich seminar on broadband communications access - transmission - networking (Cat. No.02TH8599); 2002. p. 30.

[17] Steinbauer M, Molisch AF, Bonek E. The double-directional radio channel. IEEE Antennas Propag Mag 2001;43(4):51–63.

[18] Schirmer C, Lorenz M, Kotterman WAT, Perthold R, Landmann MH, Del Galdo G. MIMO over-the-air testing for electrically large objects in non-anechoic environments. In: 2016 10th European conference on antennas and propagation; 2016. p. 1–6.

[19] Schirmer C, Landmann MH, Kotterman WAT, Hein M, Thomä RS, Del Galdo G, et al. 3D wave-field synthesis for testing of radio devices. In: The 8th European conference on antennas and propagation; 2014. p. 3394–3398.

[20] Carrascal B, Estevez GA, Lee P, Lorenzo V. Vector spherical harmonics and their application to classical electrodynamics. Eur J Phys 1991;12(4):184–91.