

# Anhang zu Digitale Signalverarbeitung

Sebastian Semper – FG Elektrische Messtechnik und Signalverarbeitung – EMS

30. Januar 2024

# Inhaltsverzeichnis

<b>1 Theorie</b>	<b>3</b>
1.1 DFT-Interpolation . . . . .	3
1.2 B-Splines . . . . .	4
1.2.1 B-Splines als Polynome . . . . .	4
1.2.2 Kubische B-Spline Interpolation . . . . .	6
1.2.3 Verbindung zur Nyquist-Sampling-Theorie . . . . .	8
<b>2 Anwendungen</b>	<b>11</b>
2.1 DFT-Interpolation von Antennenantworten: EADF . . . . .	11
2.1.1 Motivation . . . . .	11
2.1.2 Messvorgang . . . . .	12
2.1.3 Ableitung der EADF . . . . .	14
<b>Akronyme</b>	<b>16</b>
<b>Literatur</b>	<b>17</b>

# 1 Theorie

## 1.1 DFT-Interpolation

Gegeben sei ein periodisches, analoges Signal  $x : \mathbb{R} \rightarrow \mathbb{C}$ , mit Periode  $T_p = 1/F_0$ . Wir beobachten dieses Signal auf einem uniformen Raster von Punkten, via  $x[n] = x(nT)$  und wollen eine Funktion  $y(t)$  herleiten, für welche die Interpolationsbedingung

$$y(nT) = x[n] = x(nT) \quad (1.1.1)$$

erfüllt ist. Hier zu entwickeln wir das Signal  $x$  in seine Fourier-Reihe via

$$x(t) = \sum_{k=-\infty}^{+\infty} c[k] \exp(j2\pi ktF_0). \quad (1.1.2)$$

Nun tasten wir dieses Signal uniform mit Samplerate  $F_s = N/T_p = 1/T$  (also passend zur Periodendauer) ab und erhalten die Folge

$$x[n] = x(nT) = \sum_{k=-\infty}^{+\infty} c[k] \exp(j2\pi knTF_0) = \sum_{k=-\infty}^{+\infty} c[k] \exp\left(j2\pi k \frac{n}{N}\right) \quad \text{für } n \in \mathbb{N} \quad (1.1.3)$$

bestehend aus den Samples von  $x$ . Mit der Periodizität von  $\exp(j2\pi t)$  und der Abtastung erhalten wir außerdem noch

$$x[n] = \sum_{k=0}^{N-1} \left[ \sum_{\ell=-\infty}^{+\infty} c[k - \ell N] \right] \exp\left(j2\pi k \frac{n}{N}\right) = \sum_{k=0}^{N-1} \tilde{c}[k] \exp\left(j2\pi k \frac{n}{N}\right), \quad (1.1.4)$$

wobei wir

$$\tilde{c}[k] = \sum_{\ell=-\infty}^{+\infty} c[k - \ell N] \quad (1.1.5)$$

als Abkürzung benutzt haben. Ist nun die Funktion  $x$  auch bandbegrenzt, d.h. ihre Fourier-Transformierte  $X : \mathbb{R} \rightarrow \mathbb{C}$  verschwindet außerhalb eines gewissen Bandes, also

$$X(F) = \int_{-\infty}^{+\infty} x(t) \exp(-j2\pi Ft) dt = 0 \quad \text{für } |F| > B. \quad (1.1.6)$$

Außerdem wissen wir, dass die Fourier-Transformation  $X$  und die Folge  $c[k]$  verknüpft sind via

$$c[k] = \frac{1}{T_p} X(kF_0), \quad (1.1.7)$$

was impliziert, dass die Folge  $c[k]$  verschwindet, also gilt

$$c[k] = 0 \quad \text{für } |k| > \frac{B}{F_0}. \quad (1.1.8)$$

Das heißt, dass wir nun  $F_s = N/T_p$  so groß wählen müssen, dass sich in (1.1.5) kein Aliasing für  $\tilde{c}[k]$  ergeben darf. Es muss also gelten

$$N > \lceil B/F_0 \rceil \quad \text{bzw. } F_s > \lceil B/(F_0 T_p) \rceil. \quad (1.1.9)$$

In diesem Falle gilt, dann dass  $c[k] = X[k]$ , wobei  $X[k]$  die **Discrete Fourier Transform (DFT)** der Folge  $x[n]$  darstellt. Das heißt, dass wir die Fourier-Koeffizienten der kontinuierlichen Funktion  $x$  durch die **DFT** der Abtastwerte  $x[n]$  bestimmen können. Mit (1.1.2) können wir also die Folge  $x[n]$  interpolieren, indem wir

$$y(t) = \frac{1}{T_p} \sum_{k=-\frac{B}{F_0}}^{+\frac{B}{F_0}} X[k] \exp(j2\pi ktF_0) \quad (1.1.10)$$

schreiben. Dieses  $y$  erfüllt die Interpolationsbedingung (1.1.1), weil wegen der Bandbegrenzung von  $x$  und der Periodizität sogar  $y(t) = x(t)$  für alle  $t$  gilt.

Man beachte hier, dass nun aus der Folge von diskreten Werten  $x[n]$  eine analytische Formel in Form einer *endlichen* Summation entstanden ist. Unter der Annahme der Bandlimitierung von  $x$  ist diese Interpolation *exakt* und kann effizient implementiert werden, durch die Vorberechnung der Folge  $X[k]$  durch die **Fast Fourier Transform (FFT)** [1] der Folge  $x[n]$ . Eine Anwendung der hier vorgestellten Methode zur Interpolation wird in Abschnitt 2.1 aufgezeigt.

## 1.2 B-Splines

Eine Grundvoraussetzung für eine praktisch nützliche digitale Signalverarbeitung ist die Möglichkeit zwischen dem analogen und digitalen Bereich wechseln zu können. Hierbei sollte man auch genau quantifizieren können, ob bei diesem Prozess Informationen verloren gehen, oder wie man garantieren kann, dass diese Umwandlung verlustfrei vonstatten geht. Meist nutzt man hierfür das Nyquist-Shannon Sampling Theorem [2, Kapitel 1.4.2]. Die hieraus resultierende sogenannte Nyquist-Sampling-Theorie fußt bekanntermaßen auf der Repräsentation von bandbegrenzten Signalen durch hinreichend dichte äquidistante Abtastwerte. Diese Theorie ist gut studiert, in Textbüchern aufbereitet und bildet die Grundlage für viele Messsysteme und Algorithmen im Digitalen, siehe beispielsweise Abschnitt 1.1.

Es gibt jedoch auch einige Nachteile von Nyquist-Sampling, die aus dessen Annahmen und der daraus folgenden Verarbeitung entstehen. Einerseits kann ein endliches Signal im Allgemeinen *nicht* bandbegrenzt sein. Weiterhin entstehen durch die Bandlimitierung von Signalen Gibbs-Artefakte, die besonders in Bildern nicht erwünscht sind. Geht es um die Auswertung  $x(t)$  eines Signals  $x$  zwischen den aufgenommenen Samples  $x[n]$ , also Interpolation, hat man das Problem, dass die sinc-Funktion nur sehr langsam mit Rate  $1/t$  abfällt. Diese Eigenschaft führt dazu, dass man für die Bestimmung eines Wertes  $x(t)$  mit einer Genauigkeit von 1% etwa 100 um  $t$  benachbarte Samples betrachten muss. Das heißt, vor allem bei 2D-Interpolation skaliert der resultierende Rechenaufwand nicht sehr günstig, falls hohe Genauigkeit benötigt wird.

Aus diesem Grund möchten wir uns eine alternative Sampling-Theorie genauer ansehen – die B-Splines [3]. Wir führen zunächst die auf Polynomen basierende Signalverarbeitung ein und vergleichen sie anschließend zur bereits bekannten Nyquist-Theorie.

### 1.2.1 B-Splines als Polynome

Allgemein bezeichnet man stückweise definierte und stetig differenzierbare Polynome als Splines. Man bezeichnet die Stellen an denen zwei unterschiedliche Polynome zusammenstoßen als Knoten. Ein Spline der Ordnung  $\ell \in \mathbb{N}$  ist ein Polynom vom Grad  $\ell$ , ist also von der Form

$$p(x) = a_\ell t^\ell + a_{\ell-1} t^{\ell-1} + \cdots + a_1 t + a_0. \quad (1.2.1)$$

Ein Spline ist nun eine Funktion  $s(t)$ , welche für Knoten  $n = 1, 2, \dots$  definiert ist durch

$$s(t) = \begin{cases} p_1(t) & \text{für } x \in [1, 2], \\ p_2(t) & \text{für } x \in [2, 3], \\ \vdots & \end{cases} \quad (1.2.2)$$

wobei sich die Glattheit durch die Forderung ergibt, dass die Funktion und ihre Ableitungen an den Knoten stetig sei, also

$$\lim_{t \rightarrow n^-} s^{(m)}(t) = \lim_{t \rightarrow n^+} s^{(m)}(t) \quad (1.2.3)$$

erfüllt ist, wobei  $s^{(m)}$  für  $m \geq 0$  die  $m$ -te Ableitung des Splines  $s$  repräsentiert. In einer Arbeit [4], die sogar dem berühmten Paper von Shannon vorausgeht, beschreibt Schoenberg, dass sich diese Splines der Ordnung  $\ell$  via

$$s(t) = \sum_{k \in \mathbb{Z}} c[k] \beta^\ell(t - k) \quad (1.2.4)$$

darstellen lassen. Hierbei ist die Funktion  $\beta^\ell : \mathbb{R} \rightarrow \mathbb{R}$  definiert als eine iterierte Faltung einer Rechteck-Funktion via

$$\beta^\ell = \underbrace{\beta^0 * \dots * \beta^0}_{(\ell+1) \text{ mal}}, \quad \text{wobei } \beta^0(t) = \begin{cases} 1, & |t| < \frac{1}{2} \\ \frac{1}{2}, & |t| = \frac{1}{2} \\ 0, & \text{sonst.} \end{cases} \quad (1.2.5)$$

In Abb. 1 sind die Funktionen  $\beta^\ell$  für  $\ell = 0, \dots, 3$  dargestellt. Man erkennt sehr gut, dass der Grad der Glattheit von der Ordnung des Splines abhängt und dass die Funktionswerte  $\beta^\ell(t)$  für  $|t| > \ell + 1/2$  verschwinden. Man spricht von Funktionen mit kompaktem Träger. Das heißt die Summation in (1.2.4) ist für fixes  $t \in \mathbb{R}$  *endlich* und ist auf  $\ell + 1$  Summanden beschränkt! Wir wollen nun eine explizite Formel für  $\beta^\ell$  entwickeln. Hierzu betrachten wir die Fourier-Transformation

$$B^\ell(\omega) = \left( \frac{\sin(\omega/2)}{(\omega/2)} \right)^{\ell+1} = \frac{(\exp(j\omega/2) - \exp(-j\omega/2))^{\ell+1}}{(j\omega)^{\ell+1}} \quad (1.2.6)$$

mit einigen Rechentricks (siehe [3, Box 1.]) kann man dies so lange umformen, bis man

$$\beta^\ell(t) = \frac{1}{\ell!} \sum_{p=0}^{\ell+1} \binom{\ell+1}{p} (-1)^p \left( t - p + \frac{\ell+1}{2} \right)_+^\ell \quad \text{mit } (x)_+ = \begin{cases} x, & \text{für } x \geq 0, \\ 0, & \text{sonst,} \end{cases} \quad (1.2.7)$$

erhält. Damit ist  $\beta^\ell$  wirklich ein Polynom  $\ell$ -ten Grades. Die Stetig- und Differenzierbarkeit muss man sich aber noch separat überlegen.

Weiterhin kann man zeigen, dass folgende Formeln für Differentiation und Integration von B-Splines gelten:

$$(\beta^\ell)'(t) = \beta^{\ell-1}(x + 1/2) - \beta^{\ell-1}(x - 1/2), \quad \int_{-\infty}^t \beta^\ell(s) ds = \sum_{p=0}^{+\infty} \beta^{\ell+1}(t - 1/2 - p). \quad (1.2.8)$$

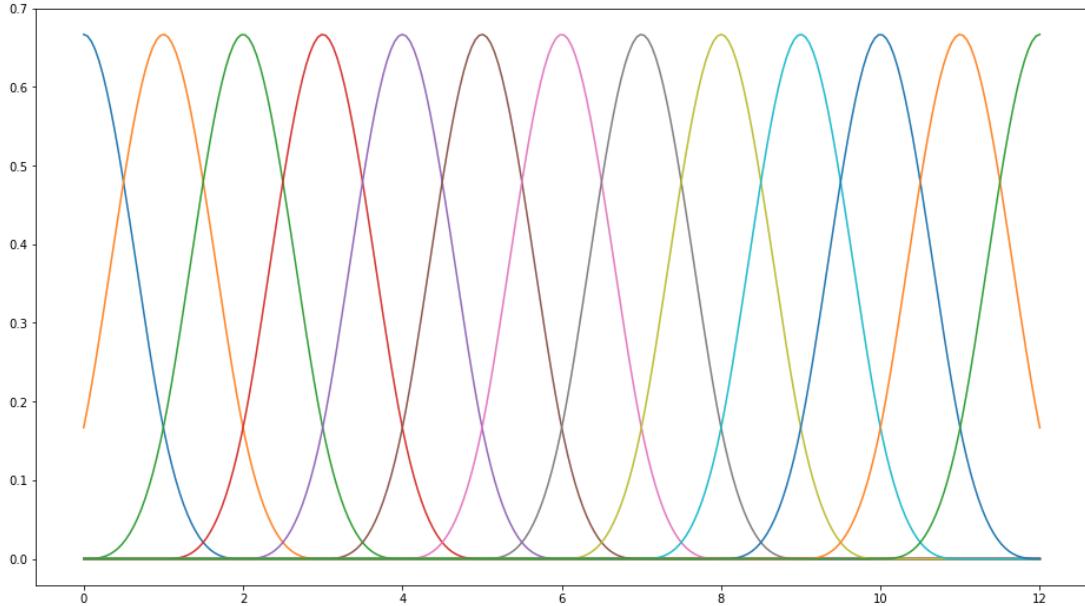


Abbildung 1: Kubische B-Splines für Abtastung an den Werten  $n = 0, \dots, 12$ .

Das heißt, dass man auch einen kompletten Spline  $s$  differenzieren und integrieren kann, indem man nutzt, dass sowohl Differentiation, als auch Integration lineare Operationen sind. Es gilt also mit (1.2.4) und (1.2.8) beispielsweise für die Differentiation, dass

$$s'(t) = \sum_{k \in \mathbb{Z}} c[k] (\beta^\ell)'(t - k) = \sum_{k \in \mathbb{Z}} c[k] (\beta^{\ell-1}(t + 1/2 - k) - \beta^{\ell-1}(t - 1/2 - k)). \quad (1.2.9)$$

Diese Eigenschaft macht man sich auch für kompliziertere Operationen, wie Rotationen und Verzerrungen der Interpolante zunutze.

### 1.2.2 Kubische B-Spline Interpolation

Wir möchten uns eine spezielle Version der B-Splines genauer Ansehen, da sie in der Anwendung den Spagat zwischen Komplexität und Approximationsgüte sehr gut hinbekommen. Wir setzen hierzu  $\ell = 3$  und erhalten somit ein Polynom dritten Grades der Form

$$\beta^3(t) = \begin{cases} \frac{2}{3} - |x|^2 + \frac{|x|^3}{2} & \text{für } |x| < 1 \\ \frac{(2-|x|)^3}{6}, & \text{für } |x| \in [1, 2) \\ 0, & \text{für } |x| > 2, \end{cases} \quad (1.2.10)$$

welche in Listing 1 auch einmal implementiert wurde. In Analogie zum bekannten Nyquist-Sampling wollen wir untersuchen, wie man aus endlich vielen gegebenen Abtastwerten  $x[n]$  mit  $n \leq N$  eine Darstellung wie in (1.2.4) herleiten können, welche die abgetasteten Werte exakt interpoliert. Aufgabe ist es also aus  $x[n]$  die Folge  $c[k]$  zu bestimmen.

Hierzu benötigen wir die sogenannte Interpolationsbedingung, welche für eine zu interpolierende Funktion

```

def cubic_spline(t: np.ndarray) -> np.ndarray:
    s1: np.ndarray = 2.0/3.0 - np.abs(t)**2 + 0.5 * np.abs(t)**3
    s2: np.ndarray = (2 - np.abs(t))**3 / 6
    s3: np.ndarray = np.zeros_like(t)
    return s3 + s1 * (np.abs(t) < 1) + s2 * (np.abs(np.abs(t) - 1.5) <= 0.5)

```

Listing 1: Berechnung des kubischen B-Splines.

$x$  und ihre Abtastwerte  $x[n] = x(n)$  fordert, dass

$$x(n) = x[n] = s(n) = \sum_{k \in \mathbb{Z}} c[k] \beta^3(n - k) \quad (1.2.11)$$

gilt. Wir fordern also *exakte* Interpolation. Nun könnte man für die Bestimmung der Folge  $c[k]$  ein lineares Gleichungssystem aufstellen, welches die Form

$$B \cdot c = y \quad (1.2.12)$$

hat, wobei die Systemmatrix  $B$  durch die Auswertung der B-Splines bestimmt ist und der Vektor  $y$  den Werten  $x[n]$  entspricht. Dem endlichen Träger der Funktionen  $\beta^\ell$  ist es zu verdanken, dass die Matrix  $B$  mit nur weniger von 0 verschiedenen Werten besetzt. (genauer: band-diagonal) und demzufolge effizient invertierbar ist. Doch die Anwendung von  $B^{-1}$  ist numerisch instabil, weshalb wir einen alternativen Weg einschlagen, der auf inverser Filterung beruht.

Sehen wir uns (1.2.11) genauer an. Wir finden, dass sich diese Gleichung nach Definition von  $\beta[k] = \beta^3(k)$  als Faltung via

$$x[n] = (c * b)[n] \quad (1.2.13)$$

schreiben lässt. Nach Transformation in den  $z$ -Bereich erhalten wir

$$X(z) = C(z) \cdot B(z) \Rightarrow C(z) = \frac{X(z)}{B(z)}, \quad (1.2.14)$$

was uns motiviert eine Darstellung von  $1/B(z)$  herzuleiten. Für die kubischen B-Splines folgt, dass

$$B(z) = \frac{z + 4 + z^{-1}}{6} \Rightarrow \frac{1}{B(z)} = 6 \left( \frac{1}{1 - z_1 z^{-1}} \right) \left( \frac{-z_1}{1 - z_1 z} \right) \quad (1.2.15)$$

gilt. Wobei man zeigen kann, dass  $z_1 = \sqrt{3} - 2 < 1$  gilt. Wir betrachten nun  $1/B(z)$  als einen Filter, der auf die Abtastwerte  $x[n]$  angewandt werden soll. Aus (1.2.15) erkennen wir, dass  $1/B(z)$  ein Filter ohne Nullstellen ist und als Hintereinanderausführung von zwei rekursiven Filtern betrachtet werden kann. Wir erhalten mit  $c^-[k] = c[k]/6$ , dass sich  $1/B(z)$  durch

$$c^+[k] = x[n] + z_1 c^+[k-1] \quad \text{für } k = 1, \dots, N-1 \quad (1.2.16)$$

$$-c[k]/6 = c^-[k] = z_1 (c^-[k+1] - c^+[k]) \quad \text{für } k = N-2, \dots, 0 \quad (1.2.17)$$

ausdrücken lässt. Diese Methode der kausalen und anti-kausalen Filterung ist deutlich effizienter und stabiler, als (1.2.12), da beispielsweise keine Divisionen notwendig sind. Nun ist es noch notwendig Anfangswerte für

```

def filter_coeffs(x_n: np.ndarray) -> np.ndarray:
    z_1: float = np.sqrt(3) - 2.0
    c_k_p: np.ndarray = np.zeros_like(x_n)
    c_k_m: np.ndarray = np.zeros_like(x_n)

    c_k_p[0] = np.sum(x_n * np.power(z_1, np.arange(x_n.size)))

    for kk in range(1, x_n.size):
        c_k_p[kk] = x_n[kk] + z_1 * c_k_p[kk-1]

    c_k_m[-1] = z_1 * (c_k_p[-1] + z_1 * c_k_p[-2]) / (1.0 - z_1**2)

    for kk in range(2, x_n.size+1):
        c_k_m[-kk] = z_1 * (c_k_m[-kk+1] + c_k_p[-kk])

    return - 6 * c_k_m

```

Listing 2: Berechnung der B-Spline Koeffizienten  $c[k]$ .

$c^+[k]$  und  $c^-[k]$  zu finden. Dies ist aufgrund der Endlichkeit von  $x$  nicht ohne Weiteres möglich. Man sieht, dass die Impulsantwort von  $c^+$  eine abklingende Exponentialfunktion ist, also gilt

$$c^+[0] = \sum_{k=0}^{\infty} x[n] z_1^k \approx \sum_{k=0}^K x[n] z_1^k, \quad (1.2.18)$$

wobei man  $K \in \mathbb{N}$  so wählen kann, dass  $z_1^K \leq \varepsilon$  erfüllt ist. Nach Ausführung von  $c^+$  kann man  $c^-[N-1]$  durch

$$c^-[N-1] = \frac{z_1}{1-z_1^2} (c^+[N-1] + z_1 c^+[N-2]) \quad (1.2.19)$$

effizient und exakt initialisieren. Beides wurde in Listing 2 beispielhaft implementiert und eine beispielhafte Ausgabe nach Auswertung von (1.2.4) ist in Abb. 2 dargestellt.

### 1.2.3 Verbindung zur Nyquist-Sampling-Theorie

Wir wollen als Abschluss eine Verbindung zum Sampling und der Interpolation [2, Kapitel 6.1] von bandbegrenzten Funktionen mit endlicher Energie ziehen. Nehmen wir als Wiederholung zunächst an, dass das bandbegrenzte Signal  $x_a$  mit endlicher Energie und Fourier-Transformation  $X_a$  mindestens kritisch mit Rate  $F_s$  zu den Werten  $x[n]$  abgetastet wurde. Wir bezeichnen mit  $X$  die **Discrete Time Fourier Transform (DTFT)** von  $x[n]$ . Dann gilt als Zusammenhang zwischen den beiden Spektren, dass

$$X(f) = F_s \sum_{k=-\infty}^{+\infty} X_a((f - k)F_s), \quad (1.2.20)$$

was die Periodifizierung des Frequenzbereiches nach Abtastung ausdrückt. Nach Annahme der kritischen Abtastung findet hier kein Aliasing statt, sodass für  $f \in [-F_s/2, +F_s/2]$  gilt, dass  $F_s \cdot X_a(f) = X(f)$ .

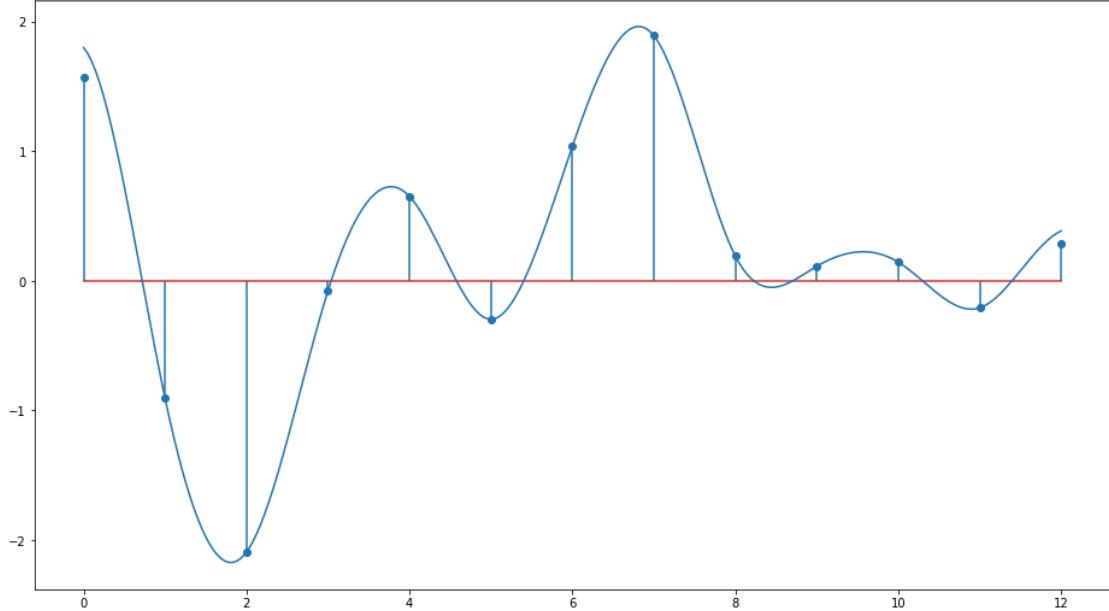


Abbildung 2: Kubische B-Spline-Interpolation für Abtastung an den Werten  $n = 0, \dots, 12$ .

Außerdem können wir das Spektrum der abgetasteten Werte  $X$  durch

$$X(f) = \sum_{n=-\infty}^{+\infty} x[n] \exp(-j2\pi fn/F_s) \quad (1.2.21)$$

ausdrücken können. Nun können wir das analoge Signal  $x_a$  in Abhängigkeit von den Abtastwerten darstellen. Es gilt mit  $T = 1/F_s$ , dass

$$\begin{aligned} x_a(t) &= \int_{-F_s/2}^{+F_s/2} X_a(f) \exp(j2\pi ft) df = \frac{1}{F_s} \int_{-F_s/2}^{+F_s/2} \left( \sum_{n=-\infty}^{+\infty} x[n] \exp(-j2\pi fn/F_s) \right) \exp(j2\pi ft) df \\ &= \frac{1}{F_s} \sum_{n=-\infty}^{+\infty} x[n] \int_{-F_s/2}^{+F_s/2} \exp(-j2\pi fn/F_s) \exp(j2\pi ft) df \\ &= \frac{1}{F_s} \sum_{n=-\infty}^{+\infty} x[n] \int_{-F_s/2}^{+F_s/2} \exp(j2\pi f(t - n/F_s)) df = \sum_{n=-\infty}^{+\infty} x[n] \frac{\sin(\pi(t - nT)/T)}{\pi(t - nT)/T}. \end{aligned}$$

Das heißt, dass die Funktion  $g : \mathbb{R} \rightarrow \mathbb{R}$  mit

$$g(t) = \frac{\sin(\pi t/T)}{(\pi t/T)} \quad (1.2.22)$$

als *Interpolations-Kernel* von bandbegrenzten und abgetasteten Funktionen betrachtet werden kann. Siehe ?? für eine Anwendung dieser Art der Interpolation.

Nun können wir eine analoge Rechnung für die B-Splines durchführen, indem wir einen Filter  $b^{-1}[k]$  als

inverse  $Z$ -Transformation von  $1/B(z)$  aus (1.2.14) definieren. Dann gilt

$$c[k] = (b^{-1} * x)[k], \quad (1.2.23)$$

was wir in (1.2.11) einsetzen und dann

$$s(t) = \sum_{n \in \mathbb{Z}} (b^{-1} * x)[n] \beta^\ell(t - n) = \sum_{n \in \mathbb{Z}} x[n] \sum_{p \in \mathbb{Z}} b^{-1,\ell}[p] \beta^\ell(t - n - p) = \sum_{n \in \mathbb{Z}} x[n] h^\ell(t - n) \quad (1.2.24)$$

erhalten, wobei wir analog zu (1.2.22) den Interpolationskernel  $h : \mathbb{R} \rightarrow \mathbb{R}$  durch

$$h^\ell(t) = \sum_{p \in \mathbb{Z}} b^{-1,\ell}[p] \beta^\ell(t - p) \quad (1.2.25)$$

definiert haben. Man kann zeigen, dass  $\lim_{\ell \rightarrow \infty} h^\ell = g$ . Das heißt, dass die sinc-Interpolation als Grenzwert der B-Spline Interpolation aufgefasst werden kann – oder andersherum die B-Spline-Interpolation als Approximation der sinc-Interpolation. Das heißt, dass auch im Frequenzbereich Konvergenz in der Form

$$H^\ell(\omega) = \left( \frac{\sin(\omega/2)}{\omega/2} \right)^{\ell+1} \frac{1}{B^\ell(\exp(j\omega))} \xrightarrow{\ell \rightarrow \infty} \text{rect}(\omega) \quad (1.2.26)$$

gegeben sein muss.

Zusammenfassend kann man sagen, dass B-Splines einen alternativen Zugang zu digitaler Signalverarbeitung bieten, welcher eng mit dem des Nyquist-Samplings verknüpft ist und als Approximation von diesem gesehen werden kann. B-Splines sind wegen ihrer effizienten und stabilen Implementierung sowohl bei der Analyse (1.2.15), als auch der Synthese (1.2.4) vor allem für hochdimensionale Interpolationen sehr interessant.

## 2 Anwendungen

### 2.1 DFT-Interpolation von Antennenantworten: EADF

#### 2.1.1 Motivation

Mit jeder Erschließung von neuen Frequenzbereichen für die Kommunikation ist es von Interesse das Ausbreitungsverhalten der Elektro-Magnetischen Wellen für verschiedene Umgebungen zu charakterisieren, beispielsweise innerstädtisch, auf der Autobahn, etc. Zwar können solche Umgebungen auch computerbasiert simuliert werden, doch für eine empirisch abgeleitete Statistik solcher sogenannter Kanalmodelle [5] sind repräsentative Messungen unerlässlich. Diese Charakteristiken werden genutzt, um in realistischen Szenarien Kanalkapazitäten, Datenraten und dergleichen zu bestimmen. Schlussendlich fließen solche Statistiken dann in neue Mobilfunkstandards ein.

Das FG EMS hat sich deshalb unter anderem auf solche Messungen und deren Auswertung, das sog. Channel Sounding [7], spezialisiert. Hierbei kommen meist breitbandige **Multiple Input Multiple Output (MIMO)** Messsysteme zum Einsatz, die den Funkkanal in Frequenz, Raum und Zeit kohärent vermessen können, wie in Abbildung 3 dargestellt. Anschließend nutzt man spezielle Signalverarbeitungstechniken [8], die einerseits unter gewissen physikalischen Annahmen das Ausbreitungsverhalten aus den gemessenen Daten ableiten können, und andererseits gleichzeitig den Einfluss des Messsystems so weit wie möglich aus den geschätzten Kanalstatistiken entfernen. Schließlich ist man an der Realität außerhalb des Messaufbaus interessiert.

Natürlich sind hierzu vor der Messung präzise Kalibriermessungen des Systems notwendig. Wir wollen uns im folgenden auf die Wirkung der benutzten Antennenarrays konzentrieren, da diese – wie wir sehen werden – eine gewisse Sonderbehandlung benötigen. Zunächst stellt man bei der Konzipierung und Benutzung des Messsystems sicher, dass es sich um ein **Linear Time-Invariant (LTI)** System handelt. Betrachtet man nun das Verhalten des Systems im Frequenzbereich für ein einzelnes Paar von Sende- und Empfangsantenne, dann gilt demnach zunächst

$$Y(f) = G_{\text{rx}}(f) \cdot H(f) \cdot G_{\text{tx}}(f) \cdot X(f). \quad (2.1.1)$$

Hierbei steht  $X$  für die Anregung des Systems durch ein eingegebenes Signal,  $G_{\text{tx/rx}}$  für die Transferfunktion der Sender- bzw. Empfängerhardware, und  $H$  für die Transferfunktion des Funkkanals, der demnach auch als ein **LTI** System modelliert wird. Es stellt sich aber heraus, dass jede Antenne eine *winkelabhängige*

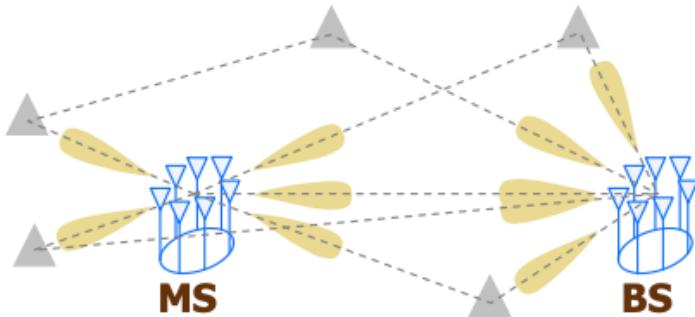


Abbildung 3: Schematische Darstellung einer **MIMO** Kanalmessung. Grafik aus [6].

Richtcharakteristik besitzt. Das heißt, dass die Systemantworten  $G_{\text{tx}/\text{rx}}$  davon abhängig sind, in welche Richtungen sich die Wellen vom Sender tx ausbreiten und aus welchen Richtungen, sie am Empfänger rx eintreffen.

Um dies korrekt zu modellieren, muss man sich also zunächst auf einzelne sog. *Ausbreitungspfade* konzentrieren. Das heißt wir nehmen an, dass eine ebene Welle sich in die normierte Richtung  $\Omega_{\text{tx}}$  ausbreitet und nach ihrem Weg durch den Funkkanal am Empfänger aus normierter Richtung  $\Omega_{\text{rx}}$  eintrifft. Folglich ergibt sich für dieses Verhalten

$$Y(f, \Omega_{\text{tx}}, \Omega_{\text{rx}}) = G_{\text{rx}}(f) \cdot a_{\text{rx}}(f, \Omega_{\text{rx}}) \cdot H(f, \Omega_{\text{tx}}, \Omega_{\text{rx}}) \cdot a_{\text{tx}}(f, \Omega_{\text{tx}}) \cdot G_{\text{tx}}(f) \cdot X(f), \quad (2.1.2)$$

wobei  $a_{\text{tx}/\text{rx}}$  für die richtungs- und frequenzabhängige Antwort der Sende- und Empfangsanntenne stehen. In diesem Fall bezeichnet also  $H(f, \Omega_{\text{tx}}, \Omega_{\text{rx}})$  die Transferfunktion eines einzelnen Pfades, der den Sender in Richtung  $\Omega_{\text{tx}}$  verlässt und am Empfänger aus Richtung  $\Omega_{\text{rx}}$  eintrifft. Das heißt, wir haben in diesem Fall das Verhalten der Antennen vom Rest des Systems isoliert.

Die Transferfunktion für die gesamte Messung wird dann als Summe der Transferfunktionen solcher ebenen Wellen modelliert, also via

$$Y(f) = \sum_{s=1}^S Y(f, \Omega_{\text{tx},s}, \Omega_{\text{rx},s}), \quad (2.1.3)$$

was sich dadurch rechtfertigt, dass die Transferfunktion des Kanals sich aus der Lösung einer partiellen Differentialgleichung ergibt, deren Lösungsraum lineare Struktur hat. Es zeigt sich aus (2.1.2), dass wir eine möglichst präzise Formulierung für  $a_{\text{tx}/\text{rx}}$  benötigen, um die Transferfunktion des Kanals  $H$  korrekt bestimmen zu können.

## 2.1.2 Messvorgang

Es ist also unsere Aufgabe für eine gegebene Antenne ein parametrisches Modell  $a : [0, \pi] \times [0, 2\pi] \rightarrow \mathbb{C}$  der Form  $a(\varphi, \vartheta) \in \mathbb{C}$ , also in Betrag und Phase, herzuleiten. Aus Gründen der Einfachheit und der Physik vernachlässigen wir die Frequenzabhängigkeit der Antenne und konzentrieren uns auf ihr Verhalten für die Anregung mit einer einzelnen Frequenz. Auch die Polarisation von ebenen Wellen und das davon abhängige Verhalten einer Antenne vernachlässigen wir hier. Wir konzentrieren uns also auf die *Winkelabhängigkeit* der Antennenantwort.

Wie oben motiviert benötigen wir eine kontinuierliche Beschreibung der Antennenantwort. Doch diese ist uns wegen endlichem Speicherplatz auf Festplatten und angepeilter endlicher Messzeit nicht direkt zugänglich. Man weiß jedoch, dass es einen Zusammenhang zwischen der elektrischen Größe einer Antenne und deren winkelabhängigen Verhalten gibt [5, Kapitel 4]. Das heißt, man kann zeigen, dass die Funktion  $a$  *bandbegrenzt* ist, beziehungsweise sich sehr gut durch eine bandbegrenzte Funktion *approximieren* lässt. Weiterhin ist durch die Stetigkeit der Physik jede Antennencharakteristik periodisch. *\*Fourier-Reihen-Sound intensifies\**

Das heißt weiterhin, dass es uns möglich ist, die Antenne an diskreten Stellen abzutasten, sodass wir mit der Annahme der Bandbegrenzung und der Aussage des Nyquist-Theorems ein Modell ableiten können, welches die Antenne vollständig charakterisiert. In diesem Sinne geht es darum die kontinuierliche Antennenantwort  $a$  zu “digitalisieren”. Die “Abtastung” erfolgt demnach im Winkelbereich. Der zugehörige “Frequenzbereich” ist entsprechend der *räumliche Frequenzbereich*. Abbildung 4 zeigt einerseits schematisch den Messaufbau, das genutzte Koordinatensystem und beispielhaft die 3D-Darstellung einer Antennenantwort eines einzelnen Patch-Elements.

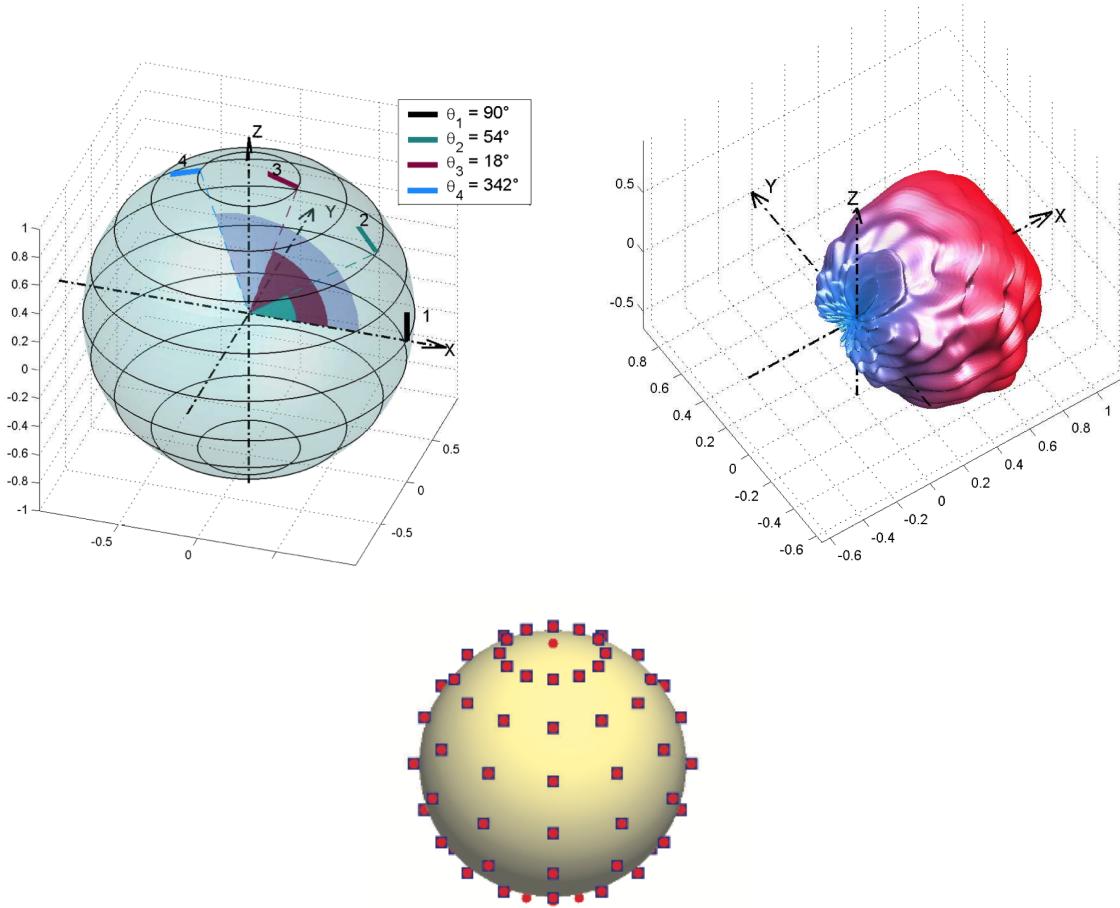


Abbildung 4: Links Oben: Darstellung der Messpositionen für eine Antenne, oder eine Antennenstruktur, welche sich im Ursprung des abgebildeten Koordinatensystems befindet. Rechts Oben: Darstellung der winkelabhängigen Amplitude eines einzelnen Patch-Elements. Unten: Messpunkte für die Abtastung der Funktion  $a$ . Grafiken aus [9, 5].

Die Messung selbst erfolgt in einer echofreien Messkammer, in welcher es möglich ist, die **Antenna Under Test (AUT)** beliebig relativ zu einer bereits kalibrierten Referenzantenne zu verdrehen, sodass ein Abtastraster, wie in Abbildung 4 unten gezeigt, entsteht. Pro Ausrichtung wird die **AUT** für gewöhnlich für mehrere Frequenzen, zwei orthogonale Polarisationen und alle ihre Elemente vermessen, bevor die nächste Ausrichtung angefahren wird. Wie erwähnt konzentrieren wir uns auf eine einzelne Frequenz, ein einzelnes Element und eine seiner Polarisationen. In Abbildung 5 sieht man den Messaufbau, der von unserem FG benutzt wurde, um ein Antennenarray mit 32 Elementen zu vermessen.

Für ein einzelnes Antennenelement beobachten wir also die Funktion  $a$  auf einem Gitter, das aus allen Kombinationen der Punkte

$$\varphi_0, \dots, \varphi_{N_\varphi}, \varphi_i = \frac{i\pi}{N_\varphi} \quad \text{und} \quad \vartheta_0, \dots, \vartheta_{N_\vartheta-1}, \vartheta_j = \frac{j2\pi}{N_\vartheta}$$

besteht. Wir erhalten also ein zweidimensionales Array  $a[i, j] \in \mathbb{C}^{N_\varphi \times N_\vartheta - 1}$ , welches wir noch durch einen

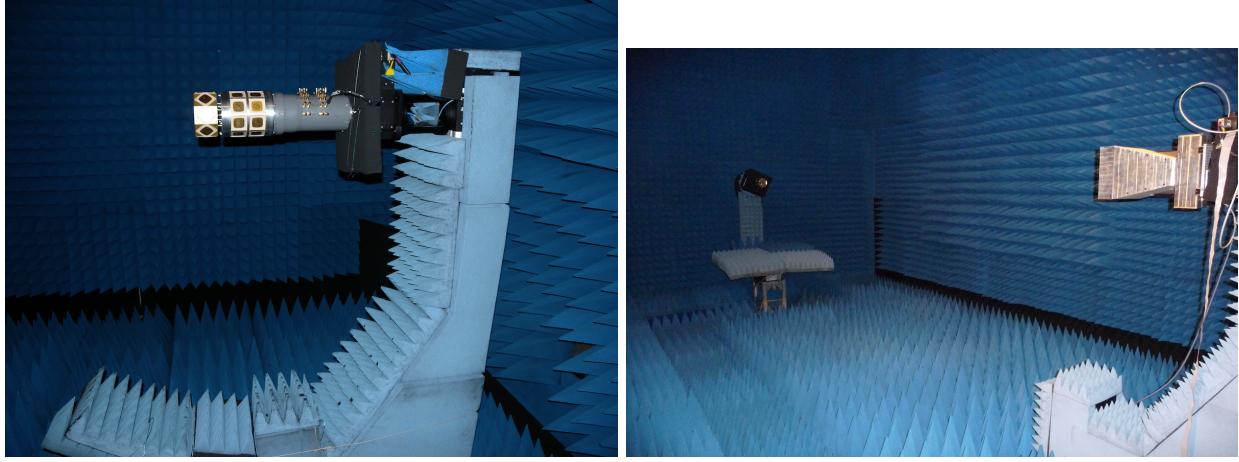


Abbildung 5: Messaufbau zur Kalibrierung eines Antennenarrays. Links: Drehteller für die Positionierung der **AUT**. Rechts: Weiterer Blickwinkel mit Referenzantenne.

Trick geeignet periodifizieren müssen, wie in Abb. 6 links dargestellt. Diese Abtastwerte entspringen also einer zweidimensionalen bandbegrenzten, periodischen Funktion. Aufgabe ist es nun aus diesen Werten eine geeignete Interpolante herzuleiten, die sich diese beiden Eigenschaften zunutze macht.

### 2.1.3 Ableitung der EADF

Wir wollen nun (1.1.10) aus Abschnitt 1.1 benutzen und auf zwei Dimensionen erweitern. Wir folgen damit effektiv [9]. Außerdem ändern wir das Argument der Funktion  $x$  und deren Namen zu der oben eingeführten Schreibweise  $a : [0, 2\pi] \times [0, 2\pi] \rightarrow \mathbb{C}$  mit Werten  $a(\varphi, \vartheta)$ . In unserem Fall der Interpolation von Antennenantworten wissen wir, dass  $a$  in *beiden* Argumenten  $2\pi$ -periodisch ist.

Die Bandlimitierung von  $a$  lässt sich formulieren, indem man fordert, dass die Bedingungen

$$A_\varphi(F_\varphi, \vartheta) = \int_{-\infty}^{+\infty} a(\varphi, \vartheta) \exp(-j2\pi F_\varphi \varphi) d\varphi = 0 \quad \text{für } F_\varphi > B_\varphi \quad \text{und alle } \vartheta \in [0, 2\pi] \quad \text{und} \quad (2.1.4)$$

$$A_\vartheta(\varphi, F_\vartheta) = \int_{-\infty}^{+\infty} a(\varphi, \vartheta) \exp(-j2\pi F_\vartheta \vartheta) d\vartheta = 0 \quad \text{für } F_\vartheta > B_\vartheta \quad \text{und alle } \varphi \in [0, 2\pi] \quad (2.1.5)$$

erfüllt sein müssen. Nun lassen sich alle obigen Argumente ‘‘schnittweise’’ auf eine abgetastete Version von  $a$  in der Form  $a[n_\varphi, n_\vartheta]$  anwenden. Das heißt, wir landen schlussendlich bei einer Interpolations-Formel

$$a(\varphi, \vartheta) = \sum_{k_\varphi=-\frac{B_\varphi}{F_\varphi}}^{+\frac{B_\varphi}{F_\varphi}} \sum_{k_\vartheta=-\frac{B_\vartheta}{F_\vartheta}}^{+\frac{B_\vartheta}{F_\vartheta}} A[k_\varphi, k_\vartheta] \cdot \exp(j2\pi k_\varphi \varphi F_\varphi) \cdot \exp(j2\pi k_\vartheta \vartheta F_\vartheta), \quad (2.1.6)$$

welche eine absolut analoge (nicht als Gegenteil zu digitale) 2D-Version zu (1.1.10) darstellt. Auch in diesem Fall, können wir das 2D-Array  $A[k_\varphi, k_\vartheta]$  durch eine 2D-DFT, bzw. der FFT [1], von den uniformen Samples  $a[n_\varphi, n_\vartheta]$  der Funktion  $a$  effizient vorberechnen.

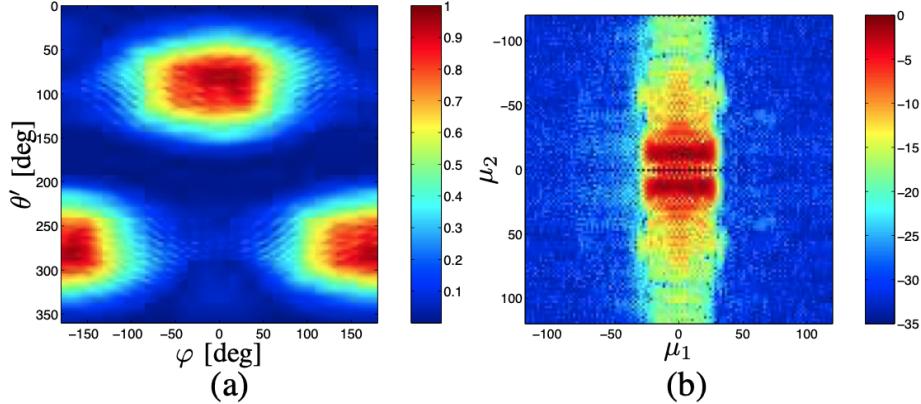


Abbildung 6: Links: Periodifiziertes 2D Array  $a[n_\varphi, n_\theta]$  der gemessenen Amplituden eines einzelnen Patch-Elements. Rechts: Der Betrag der zugehörigen EADF, wobei hier  $\mu_1 = k_\varphi$  und  $\mu_2 = k_\theta$ . Grafik aus [9]

Um einen möglichst effizienten Algorithmus für die Auswertung der Interpolante zu erhalten, sollte man (2.1.6) geeignet umschreiben. Moderne Rechenarchitekturen und Scientific-Computing-Libraries sind auf schnelle Matrix-Vektor-Produkte optimiert. Nehmen wir an, wir wollen (2.1.6) für mehrere Winkelpaare  $(\varphi_1, \theta_1), \dots, (\varphi_L, \theta_L)$  auswerten. Dann berechnen wir zunächst zwei 2D Arrays

$$D_\varphi = [\exp(j2\pi k_\varphi \varphi F_\varphi)]_{\ell=1, k_\varphi=-\frac{B_\varphi}{F_\varphi}} \in \mathbb{C}^{L \times 2^{\frac{B_\varphi}{F_\varphi}+1}} \quad \text{und} \quad (2.1.7)$$

$$D_\theta = [\exp(j2\pi k_\theta \theta F_\theta)]_{\ell=1, k_\theta=-\frac{B_\theta}{F_\theta}} \in \mathbb{C}^{L \times 2^{\frac{B_\theta}{F_\theta}+1}}, \quad (2.1.8)$$

was uns erlaubt (2.1.6) in das folgende Vektor-Matrix-Vektor-Produkt

$$a(\varphi_\ell, \theta_\ell) = D_\varphi[\ell, :] \cdot A[:, :] \cdot D_\theta[\ell, :]^\top \quad (2.1.9)$$

umzuschreiben. Damit besteht der Interpolations-Algorithmus zunächst aus der Vorberechnung des Arrays  $A$ , sowie bei Ausführung dann aus der Berechnung von  $D_\varphi$  und  $D_\theta$ , sowie der Auswertung von (2.1.9).

Man sieht hier der schön, dass die Laufzeitkomplexität von (2.1.9) maßgeblich von der räumlichen Bandbegrenzung der Antennen-Richtcharakteristik beeinflusst wird. Je höher die Bandbreite, desto höher ist nicht nur der Aufwand bei der Messung, sondern auch bei der Interpolation. Eine alternative Form der Interpolation, welche diese eventuell nachteilige Eigenschaft nicht hat, ist in Abschnitt 1.2 dargestellt.

## Akronyme

**AUT** Antenna Under Test. [13](#), [14](#)

**DFT** Discrete Fourier Transform. [4](#), [14](#)

**DTFT** Discrete Time Fourier Transform. [8](#)

**EADF** Effective Aperture Distribution Function. [15](#)

**FFT** Fast Fourier Transform. [4](#), [14](#)

**LTI** Linear Time-Invariant. [11](#)

**MIMO** Multiple Input Multiple Output. [11](#)

## Literatur

- [1] M. Frigo und S.G. Johnson. „The Design and Implementation of FFTW3“. In: *Proc. IEEE* 2 (2005). Special issue on “Program Generation, Optimization, and Platform Adaptation” (siehe S. 4, 14).
- [2] John Proakis und Dimitris Manolakis. *Digital Signal Processing*. Pearson Deutschland, 2013 (siehe S. 4, 8).
- [3] Michael Unser. „Splines: a perfect fit for signal and image processing“. In: *IEEE Signal Processing Magazine* 6 (1999) (siehe S. 4, 5).
- [4] I. J. Schoenberg. „Contributions to the Problem of Approximation of Equidistant Data by Analytic Functions“. In: *I. J. Schoenberg Selected Papers*. Hrsg. von Carl de Boor. Birkhäuser Boston, 1988 (siehe S. 5).
- [5] Giovanni Del Galdo Prof. Dr.-Ing. „Geometry-Based Channel Modeling for Multi-User MIMO Systems and Applications“. en. Diss. 2007 (siehe S. 11–13).
- [6] Andreas Richter. „Estimation of Radio Channel Parameters“. In: (2005) (siehe S. 11).
- [7] Martin Haardt, Reiner S. Thomä und Andreas Richter. „Multidimensional High-Resolution Parameter Estimation with Applications to Channel Sounding“. English. In: *High-Resolution and Robust Signal Processing*. Hrsg. von Yingbo Hua. 2003 (siehe S. 11).
- [8] Sebastian Semper u. a. „High Resolution Parameter Estimation for Wideband Radio Channel Sounding“. In: *IEEE Transactions on Antennas and Propagation* (2023) (siehe S. 11).
- [9] M. Landmann und G. Del Galdo. „Efficient antenna description for MIMO channel modelling and estimation“. In: *7th European Conference on Wireless Technology, 2004*. 2004 (siehe S. 13–15).