

Skript Digitale Signalverarbeitung

Sebastian Semper – FG Elektrische Messtechnik und Signalverarbeitung – EMS

Wintersemester 2024/2025

Inhaltsverzeichnis

Abbildungsverzeichnis	iii
Code-Schnipsel-Verzeichnis	iii
1 Theoretische Grundlagen	2
1.1 Komplexe Zahlen	2
1.2 Signale	2
1.2.1 Definition und Typen	2
1.2.2 Signale als Vektoren	3
1.2.3 Transformation von Signalen	3
1.2.4 Zufällige Signale	4
1.2.5 Spezielle Signale	5
1.2.6 Beispiel: LTI-Systeme	6
2 Abtastung von Signalen	7
2.1 Frequenz von Signalen	7
2.1.1 Zeit-Kontinuierliche Harmonische	7
2.1.2 Zeit-Diskrete Harmonische	9
2.2 Zeit-Kontinuierliche Komplexe Harmonische	10
2.3 Zeit-Diskrete Komplexe Harmonische	11
2.4 Finally: Abtastung von Signalen	12
3 Diskrete Signale und Systeme	16
3.1 Diskrete Signale	16
3.2 Diskrete Systeme	16
3.3 Diskrete LTI Systeme	17
3.4 Cross- und Autokorrelation	17
4 z-Transformation	18

5 Fourier-Transformation	19
5.1 Anwendung der Transformationen	19
5.2 DFT-Interpolation	19
5.3 DFT-Interpolation von Antennenantworten	21
5.3.1 Motivation	21
5.3.2 Messvorgang	22
5.3.3 Ableitung der EADF	24
6 Multiraten Systeme	26
7 STFT	27
8 Wavelets	28
9 B-Splines	29
9.1 B-Splines als Polynome	29
9.2 Kubische B-Spline Interpolation	31
9.3 Verbindung zur Nyquist-Sampling-Theorie	34
10 Zufällige Signale	36
Akronyme	37
Literatur	38

Abbildungsverzeichnis

1	$x_a(t) = A \cos(2\pi F t + \theta)$, Quelle: [1]	8
2	Uniforme Abtastung einer Signals. Quelle: [1]	13
3	Schematische Darstellung einer MIMO Kanalmessung. Grafik aus [5].	21
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 [8, 4].	23
5	Messaufbau zur Kalibrierung eines Antennenarrays. Links: Drehsteller für die Positionierung der AUT . Rechts: Weiterer Blickwinkel mit Referenzantenne.	24
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 [8]	25
7	Kubische B-Splines für Abtastung an den Werten $n = 0, \dots, 12$	30
8	Kubische B-Spline-Interpolation für Abtastung an den Werten $n = 0, \dots, 12$	33

Codeschnipsel-Verzeichnis

1	Berechnung und Darstellung von (2.1.1)	9
2	Berechnung und Darstellung von (2.1.2)	10
3	Visualisierung von $F_k = F + k \cdot F_s$	14
4	Berechnung und Darstellung von Theorem 2.1	15
5	Berechnung des kubischen B-Splines.	31
6	Berechnung der B-Spline Koeffizienten $c[k]$	33

Die Idee hinter dem Skript zur Vorlesung ist, dass es die Zuhörer der Bürde des Mitschreibens entledigt und Zeit und Platz zum Folgen der Vorlesung frei macht. Das Skript sollte deshalb immer zur Vorlesung und Übung mitgebracht und im Idealfall mit Notizen versehen werden, bzw. zum Nachschlagen verwendet werden.

1 Theoretische Grundlagen

Digitale Signalverarbeitung ist ein Feld, das sich vieler verschiedener mathematischer Grundlagen bedient, um die gefundenen Zusammenhänge rigoros, knapp und gleichzeitig elegant zu formulieren. Deshalb kommen wir nicht umhin, uns einiger dieser Grundlagen zu erinnern. Alles hier knapp aufgelistete sollte schon bekannt sein und dient nur als bequemes Nachschlagewerk für das kommende Semester.

1.1 Komplexe Zahlen

Die *komplexen Zahlen* \mathbb{C} sind die Menge aller $z = x + jy$, wobei $x, y \in \mathbb{R}$ und für die imaginäre Einheit j gilt, dass $j^2 = -1$. Wir nutzen hier speziell j in Abgrenzung zu i oder j , da diese oft als Indices oder Laufvariablen auftreten. Bei $z = x + jy$ nennen wir $x = \Re(z)$ den Realteil und respektive $y = \Im(z)$ den Imaginärteil. Komplexe Zahlen lassen sich auch in der Polarform $z = r \exp(j\phi)$ darstellen, wobei $r = |z| = \sqrt{\Re(z)^2 + \Im(z)^2} \geq 0$ den Betrag und $\phi = \angle(z) = \arctan(y, x)$ das Argument von z darstellen. Die zu $z = r \exp(j\phi) = x + jy$ komplex konjugierte Zahl ist $z^* = r \exp(-j\phi) = x - jy$.

Komplexe Zahlen haben viele interessante Eigenschaften und Anwendungen, vor allem in der digitalen Signalverarbeitung. Beispielsweise für die Darstellung von einem modulierten reellen Passband Signal $s : \mathbb{R} \rightarrow \mathbb{R}$ dargestellt durch

$$s(t) = x(t) \cos(\omega t) + y(t) \sin(\omega t) \in \mathbb{R},$$

die äquivalente Darstellung im komplexen Basisband

$$s_B(t) = x(t) + jy(t) \quad \text{mit} \quad \Re(s_B(t) \exp(j\omega t)) = s(t). \quad (1.1.1)$$

existiert. Man sagt auch, dass $s_B \exp(j\omega \cdot)$ das analytische Signal zu s darstellt. Dass komplexe Zahlen viele Überraschungen bereithalten sieht man wenn man sich simuliert für welche $c \in \mathbb{C}$ die Folge

$$z_{n+1} = z_n^2 + c$$

konvergiert oder divergiert, wenn man $z_0 = 0$ setzt (Übung).

1.2 Signale

1.2.1 Definition und Typen

Wir haben gerade schon von Signalen gesprochen, ohne sie etwas genauer einzuführen. Ganz allgemein kann man sich Signale als Objekte vorstellen, die abhängig von Raum, Zeit, oder beidem, physikalische Messgrößen, wie Spannungen, Feldstärken, oder Temperaturen modellieren/abbilden.

Die theoretische Darstellung von Signalen erfolgt durch *Funktionen*. Eine Funktion $s : D \rightarrow B$ besitzt einen Namen (s), einen Definitionsbereich D und einen Bildbereich B . Hierbei sind D und B zunächst irgendwelche Mengen. Die Funktion s bildet nun Paare (d, b) zwischen Mengenelementen von D und B , indem man schreibt $(d, s(d))$, oder $d \mapsto s(d) = b$. Der Witz ist nun, dass man ein Signal mit physikalischer Bedeutung erhält, indem man lediglich D und B geschickt wählt.

Ist $D = B = \mathbb{R}$ so sprechen wir von einem reellen Signal s und meist denken wir dabei bei D an die Zeitachse, weshalb wir auch $s \mapsto s(t)$ schreiben. Ist $D = \mathbb{R}^3$, $B = \mathbb{R}$, so denken wir meist an den dreidimensionalen Raum für den Definitionsbereich und haben als ein Signal im Raum gegeben. Ist nun

jedoch $D = \mathbb{Z}$, $B = \mathbb{R}$, so ist das Signal nur für die ganzen Zahlen \mathbb{Z} definiert, weshalb wir dann von einem Zeitdiskreten Signal sprechen. Meist schreiben wir hierfür kurz $s[k] \in \mathbb{R}, k \in \mathbb{Z}$. Man soll sich hier nicht vorstellen, dass die Werte „zwischen“ den ganzen Zahlen nur fehlen würden. So ist dies *nicht* zu verstehen. Zwischen den gegebenen Werten ist keine Information vorhanden! In manchen Situationen werden wir die diskreten Signale explizit aufschreiben wollen. In diesen Fällen markieren wir die Stelle $k = 0$ via

$$\dots, 0, 1, 2, 3, 2, 1, 0, \dots,$$

um eine bequeme Schreibweise für solche Folgen zu erhalten.

Versuchen sie für möglichst viele verschiedene Kombinationen von D und B Beispiele zu finden (Übung).

1.2.2 Signale als Vektoren

Um mit Signalen gut umgehen zu können, ist es wichtig ihre Eigenschaften als mathematische Objekte zu kennen. Intuitiv stellt man sich vor, dass man Signale in ihrer Intensität verändern können sollte, und für beliebige Änderung der Intensität wieder ein Signal erhält. Wir gehen hier zunächst der Einfachheit halber von $D = B = \mathbb{R}$ aus.

Definiert man für $a \in \mathbb{R}$ das Objekt $a \cdot s$ als $t \mapsto a \cdot s(t)$ so erhält man wieder ein Signal. Die Werte von s werden also einfach skaliert. Betrachtet man nun zwei Signale s_1, s_2 und definiert $s_1 + s_2$ als $t \mapsto s_1(t) + s_2(t)$, so erhalten wir die Summe oder die Superposition von s_1 und s_2 . Da Signale oft physikalische Messgrößen darstellen, macht dies auch oft Sinn, da in der Physik das Prinzip der Superposition oft eine Rolle spielt. Wenn wir die beiden Fakten nun kombinieren erhalten wir für $a_1, a_2 \in \mathbb{R}$ und zwei Signale s_1, s_2 , dass

$$(a_1 s_1 + a_2 s_2)(t) = a_1 s_1(t) + a_2 s_2(t)$$

wieder ein Signal repräsentiert. Objekte, die diese Eigenschaft haben, nennt man *Vektoren* und diese leben in einem *Vektorraum*.

Das mag erstmal nicht so schockieren, aber wir gewinnen dadurch *alle* Werkzeuge aus der linearen Algebra für unsere Zwecke. Beispielsweise können wir nun geschickt Bausteine für eine gewisse Untergruppe von Signalen finden, mit denen sich diese Signale gut und informativ beschreiben lassen. Beispielsweise könnten wir uns fragen, ob es für den Vektorraum der Bild-Signale eine Basis gibt, sodass für jedes Bild b eine Darstellung existiert, dass

$$b(x, y) = c_1 b_1(x, y) + c_2 b_2(x, y) + \dots,$$

gilt. Die Zahlen c_1, c_2, \dots können also das Signal b darstellen, indem man einfach die Elemente aus der Basis hennimmt, entsprechend skaliert und summiert. In gewisser Weise *sind* die Koeffizienten c_i das Signal b . Vielleicht gelingt es uns, die Menge $\{b_1, b_2, \dots\}$ so zu konstruieren, dass wir immer nur *wenige* von diesen b_i brauchen, sodass wir *jedes beliebige* Bild aus einer Fotokamera durch geschickte Kombination von diesen darstellen können (*sadMP3noises*).

1.2.3 Transformation von Signalen

Noch interessanter ist aber die Manipulation von Signalen durch *Transformationen*. Der Sinn von Transformationen ist es, neue oder einfach bestimmte Einsichten in ein Signal zu gewinnen. Es kann aber auch sein, dass man Operationen, die auf Signalen ausgeführt werden sollen, „einfach“ mittransformieren kann. Vielleicht

ist die gewünschte Operation nach Transformation deutlich einfacher anzuwenden? Jede Transformation liefert hierbei andere Informationen oder ist für andere Signale definiert.

Mathematisch ist eine Transformation nichts anderes als eine Abbildung zwischen Signalen. D.h. auch eine Transformation T bildet Paare zwischen Objekten aus Mengen – in diesem Fall Signalen – also $s \mapsto Ts = S$. Nach Anwendung der Transformation T auf s erhalten wir also ein anderes Signal $Ts = S$. Gerade haben wir schon festgestellt, dass man Signale beliebig skalieren und addieren kann und es als eine Art grundlegende Eigenschaft von Signalen festgehalten. Nehmen wir nun ein Signal mit Werten

$$s(t) = a_1 s_1(t) + a_2 s_2(t)$$

und wir wenden die Transformation T auf beiden Seiten der Gleichung an

$$\{Ts\}(t) = \{T(a_1 s_1 + a_2 s_2)\}(t).$$

Ist nun die Transformation so, dass wir schreiben können

$$\{Ts\}(t) = \{T(a_1 s_1 + a_2 s_2)\}(t) = a_1 \{Ts_1\}(t) + a_2 \{Ts_2\}(t),$$

so nennen wir T eine *lineare* Transformation. Zusammen mit der Superpositionseigenschaft von Signalen sieht man nun, warum Linearität so wichtig für Transformationen ist, weil es einfach zur Vektorraumstruktur von Signalen passt. Die Linearität erlaubt es uns beispielsweise auch das obige Signal b ganz einfach zu transformieren. Nehmen wir es in seiner Darstellung als

$$b(x, y) = c_1 b_1(x, y) + c_2 b_2(x, y) + \dots,$$

und wir haben eine beliebige lineare Transformation T , deren Effekt wir auf b angewendet sehen wollen. Wir suchen also $\{Tb\}(x, y)$. Aber das ist mit der Linearität ganz einfach. Wir müssen nur Tb_i kennen, also die Wirkung von T auf die Basisvektoren b_i , denn

$$\{Tb\}(x, y) = c_1 \{Tb_1\}(x, y) + c_2 \{Tb_2\}(x, y) + \dots,$$

ist eine valide Darstellung von Tb . Cool!

Beispiele für solche linearen Transformationen sind Differentiation (falls möglich), bilden der Stammfunktion (falls möglich), Verzögerung eines Zeitsignals um Zeit $a \in \mathbb{R}$, Stauchung und Streckung eines Zeitsignals, Rotation eines Bildes, die Fourier-Transformation, die diskrete Fourier-Transformation, zyklische Faltung, oder Korrelation mit einem anderen Signal p . Gegenbeispiele sind $p(t) = \sin(s(t))$, oder $p(t) = (s(t))^{\alpha}$ für $\alpha \neq 1$.

Man sieht, dass viele wichtige Operationen lineare Transformationen darstellen und wir haben mit linearer Algebra ein mächtiges Tool an unserer Seite, um mit ihnen umzugehen.

1.2.4 Zufällige Signale

Man kann auch noch eine weitere Sichtweise auf Signale haben. In manchen Fällen ist es nicht zweckmäßig, dass man ein Signal s als vollständig bekannte und fixe Funktion modelliert. Stattdessen modelliert man die Werte $s(t)$ des Signals s an den Stellen t als *Zufallsgröße*. Das heißt, dass die Werte $s(t)$ einer Verteilung $X(t)$ folgen. An jedem Zeitpunkt t „hängt“ eine solche Verteilung, die bestimmt mit welcher Wahrscheinlichkeit die Werte $s(t)$ in einem gewissen Intervall liegen. Man spricht in diesem Fall auch von *stochastischen* Signalen, im Gegensatz zu den obigen *deterministischen* Signalen.

Es kann verschiedene Gründe haben, dass man ein Signal nicht mehr deterministisch beschreiben kann/will/sollte:

- Sobald die Werte von s durch eine Messung entstanden sind, enthalten diese normalerweise Messrauschen. Dann modelliert man s meistens als Summe

$$s(t) = x(t) + n(t),$$

wobei $n(t) \sim \mathcal{N}(0, \sigma^2(t))$ meist als eine Realisierung einer mittelwertfreien Normalverteilung mit Varianz $\sigma^2(t)$ angenommen wird, und x als ein deterministisches Signal.

- Wenn man generell nicht genug Information über das Signal hat, beispielsweise, kennt man nur dessen Verteilung im Frequenzbereich, also Wahrscheinlichkeiten, dass gewisse Frequenzen vorhanden sind, oder nicht. Dennoch ist man natürlich an dem Verhalten des Signals im Zeitbereich interessiert.
- Wenn es für die Anwendung nicht notwendig ist. Dies kann der Fall sein, wenn man einen Filter entwickelt, der eine gewisse Klasse von Signalen als Eingang bekommt, kann es reichen die Verteilung der Signale zu kennen und dann den Ausgang des Filters nur stochastisch zu beschreiben.

„In 99.99 % der Fälle ist der nachgeschaltete Verstärker nicht übersteuert.“

Für solche Aussagen ist es sogar *notwendig* die Verteilung der Eingangssignale zu kennen, ansonsten ist so eine Aussage gar nicht möglich, da man eben keine Verteilung für ein deterministisches Signal angeben kann.

Um stochastiche Signale korrekt handhaben zu können, ist einige Mathematik notwendig, die wir einfach übergehen und stattdessen versuchen ein *intuitives* Verständnis zu entwickeln.

1.2.5 Spezielle Signale

Uns werden immer wieder einige spezielle Signale begegnen, die wir hier kurz auflisten wollen.

- Die *Delta-Funktion (Dirac- δ)* als Funktional δ , das angewendet auf ein Signal s , liefert, dass $\delta(s) = s(t=0)$. Visualisiert wird dieses nicht-Signal, durch einen Impuls der Höhe 1 bei $t=0$. Es ist nicht ohne Ironie, dass eines der wichtigsten Objekte der Signalverarbeitung selbst kein Signal ist, wie eines behandelt wird, aber immer mit Vorsicht.
- Die *Heavyside-Funktion $u : \mathbb{R} \rightarrow \mathbb{R}$* mit

$$u(t) = \begin{cases} 1 & \text{für } t > 0 \\ \frac{1}{2} & \text{für } t = 0 \\ 0 & \text{für } t < 0. \end{cases}$$

Man kann δ als distributionelle Ableitung von u auffassen.

- Die *komplexe Schwingung $s : \mathbb{R} \rightarrow \mathbb{C}$* bei Frequenz $f > 0$ ist definiert als $s(t) = \exp(jft)$ und wir uns im Verlauf des Semesters noch einige Male begegnen. Beispielsweise gilt $s^*(t) = s(-t)$.
- Der *diskrete δ -Stoß $\delta[k]$* ist definiert als

$$\dots, 0, \underset{\uparrow}{1}, 0, \dots$$

- Endliche Signale können wir entweder durch

$$s = [0, 1, 2, \underset{\uparrow}{3}, 2, 1, 0]$$

darstellen, oder als endliche Summe von einigen diskreten δ -Stößen:

$$s[n] = \sum_{k=-2}^{k=+2} s[k] \delta[n - k]$$

1.2.6 Beispiel: LTI-Systeme

Wir werden uns zwar noch später ausführlich mit **Linear Time-Invariant (LTI)** Systemen beschäftigen, doch sie sollen hier schon als nicht-triviales Beispiel dienen. Wir sind also mit einem System \mathcal{H} konfrontiert, das einerseits die Eigenschaft hat, dass für Anregungen $x : \mathbb{R} \rightarrow \mathbb{R}$ eine Verschiebungsinvarianz mit $y(t - \tau) = (\mathcal{H}x(\cdot - \tau))(t)$ gilt. Außerdem ist \mathcal{H} linear.

Dann kann man die Wirkung von \mathcal{H} auch durch Faltung mit der sog. Impulsantwort h des Systems darstellen, also

$$y(t) = (\mathcal{H}x)(t) = \int_{-\infty}^{+\infty} x(t - \tau) h(\tau) d\tau = (x * h)(t), \quad (1.2.1)$$

wobei $h = \mathcal{H}\delta$, also die Reaktion des Systems auf einen Dirac-Stoß darstellt. An dieser Darstellung sieht man sehr gut, dass \mathcal{H} linear ist, weil die Integration linear in x ist.

Natürlich ist der Zeitbereich für diese Art von System nicht der richtige Anschauungsort. Nach Laplace-Transformation von y zu $Y = \mathcal{L}y$ sehen wir, dass wir stattdessen

$$Y(s) = X(s) \cdot H(s),$$

schreiben können. Hierbei sind $X = \mathcal{L}x$ und $H = \mathcal{L}h$ die Laplace-Transformationen des Eingangs und der Impulsantwort h . Nicht nur hat sich die „Berechnung“ von Y vereinfacht, sondern wir haben auch ein besseres Gefühl für das Verhalten des Systems in Abhängigkeit von h , bzw. H , weil der Einfluss einfach multiplikativ ist.

Wir können die lineare Algebra noch ein wenig weiter treiben. Betrachten wir als Eingang die Funktion $x_s(t) = \exp(st)$ für ein beliebiges $s \in \mathbb{C}$. Dann rechnen wir einfach mit (1.2.1) nach, dass

$$(H \exp(s \cdot))(t) = \int_{-\infty}^{+\infty} \exp(s(t - \tau)) h(\tau) d\tau = \exp(st) \int_{-\infty}^{+\infty} \exp(-s\tau) h(\tau) d\tau = \exp(st) H(s),$$

gilt. Das heißt, dass die Funktionen $\exp(s \cdot)$ die *Eigenvektoren* des Operators \mathcal{H} , weil gilt $(\mathcal{H}x_s)(t) = x_s(t) \cdot H(s)$, wobei H die Laplace-Transformation von h ist. Das heißt auch, dass $H(s)$ die zugehörigen *Eigenwerte* sind. Wir sehen hier also, dass Signale *wirklich* wie Vektoren funktionieren können und es sich im Fall von linearen System förmlich aufzwingt, da die Linearität des Systems zur linearen Vektorraumstruktur „passt“.

2 Abtastung von Signalen

Als ersten Schritt der digitalen Signalverarbeitung wollen wir uns den Übergang von einem analogen Signal zu einem digitalen näher ansehen. Intuitiv können wir diesen Vorgang in vielen Anwendungen beobachten. Wir nehmen im Tonstudio mit einem Mikrophon Ton auf und eine Soundkarte wandelt das analoge Signal in einen WAV-Datenstrom um. In einer Fotokamera, trifft ein Feld von Lichtstrahlen ein und wird von einem **Complementary Metal Oxide Semiconductor (CMOS)**-Sensor „direkt“ abgetastet und in Helligkeitswerte pro Farbkanal umgewandelt. Eine Antenne wandelt ein anliegendes elektro-magnetisches Feld in eine Spannung um, welche nachträglich von einem **Analog-to-Digital Converter (ADC)** abgetastet und quantisiert wird.

Mathematisch modellieren wir analoge Signale $s_a : D \rightarrow B$ meist mit D und B , die auf die reellen Zahlen \mathbb{R} zurückgreifen. Die Wandlung von analog zu digital transformiert dieses Signal in eine Funktion $s : \mathbb{Z} \rightarrow Q$ um, wobei auch $|Q| < \infty$ gilt. Das heißt, dass das Signal nach AD-Wandlung nur noch endliche Werte annehmen kann und, dass es nur noch aus einer *Folge* von Werten aus der Menge Q besteht. Es wurde also zeit- und wertdiskretisiert. Wir werden uns zunächst nur mit der Diskretisierung in Zeit befassen, weil es einfacher ist. Das heißt, dass wir uns vorstellen, dass das diskretisierte Signal nur an einer diskreten Menge an Punkten noch Informationen über das abgetastete Signal beinhaltet. Weiterhin sind wir nicht an der physikalischen Umsetzung von **ADCs** interessiert, sondern höchstens an deren systemtheoretischer Modellierung.

Die zentralen Fragen sind nun:

- Wie muss der Vorgang der Abtastung gestaltet sein, dass keine Information verloren geht?
- Wie können wir die Eigenschaften des analogen Signals in dessen abgetasteter Version wiederfinden?
- Welche Operationen können wir auf digitalen Signalen wie effizient ausführen?

2.1 Frequenz von Signalen

2.1.1 Zeit-Kontinuierliche Harmonische

Meistens werden wir uns in der Vorlesung mit reell- oder komplexwertigen Zeitsignalen befassen, d.h. wir modellieren unsere Signale als $x_a : \mathbb{R} \rightarrow \mathbb{R}$ oder $x_a : \mathbb{R} \rightarrow \mathbb{C}$. Wobei physikalische Signale natürlich nur reellwertig sind, doch manchmal ist die Darstellung als komplexwertige Funktion besser handhabbar, siehe (1.1.1). Das heißt, dass die Abtastung im Zeitbereich vonstatten geht, was sofort den Begriff der *Frequenz* auf den Plan ruft, da Frequenz mit Einheit $1/\text{s}$ eng mit Zeit s verknüpft ist.

Betrachten wir also erst einmal, welchen Einfluss Abtastung von Signalen mit einzelnen Frequenzen hat, am Beispiel von

$$x_a(t) = A \cos(\Omega t + \theta), \quad (2.1.1)$$

wobei wir hier $A \in \mathbb{R}$ als Amplitude, $\Omega \in \mathbb{R}_0^+$ als Kreisfrequenz 1 rad s^{-1} , $t \in \mathbb{R}$ als Zeit 1 s und die Phase $\theta \in \mathbb{R}$ mit Einheit 1 rad nutzen. Alternativ können wir auch zur Frequenz $F \in \mathbb{R} 1 \text{ s}^{-1} = 1 \text{ Hz}$ übergehen. Dann erhalten wir

$$x_a(t) = A \cos(2\pi F t + \theta).$$

Diese Funktion ist in Abb. 1 dargestellt. Wir sehen, dass die Funktion periodisch ist mit Periode $T_p = 1/F$.

Das heißt, dass $x_a(t + k \cdot T_p)$ für $k \in \mathbb{Z}$ nicht vom Signal $x_a(t)$ zu unterscheiden ist!

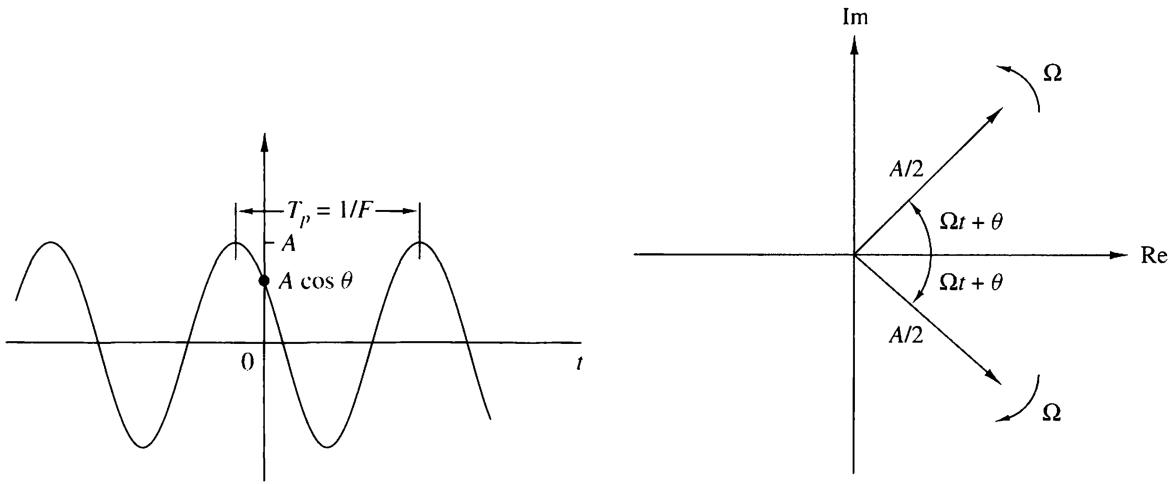


Abbildung 1: $x_a(t) = A \cos(2\pi F t + \theta)$, Quelle: [1]

Es gibt noch eine alternative Darstellung von der obigen Funktion durch die Addition von zwei *Phasoren* als

$$x_a(t) = A \cos(2\pi F t + \theta) = \frac{A}{2} \exp(j(\Omega t + \theta)) + \frac{A}{2} \exp(-j(\Omega t + \theta)).$$

Da die beiden überlagerten Phasoren so interpretiert werden können als rotierten diese in gegensätzliche Richtungen, ist es gerechtfertigt der physikalischen Intuition entgegen auch von „negativen“ Frequenzen zu sprechen. Wir erlauben also $F \in \mathbb{R}$, womit auch der Spezialfall $T_p = \infty$, also $x_a(t) = A$ abgedeckt ist. Ein kleines Beispiel findet man in Codeschnipsel 1.

```

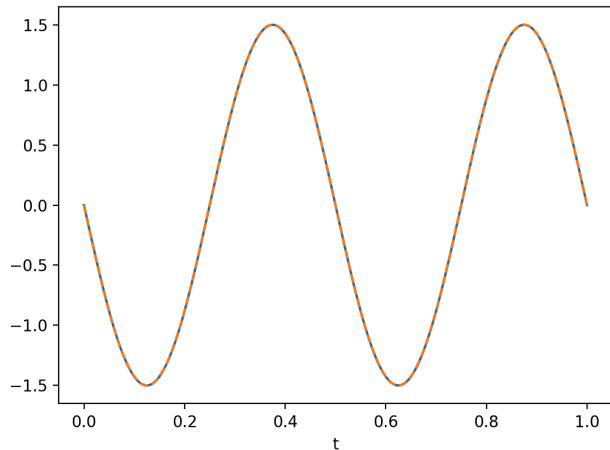
A = 1.5
F = 2
theta = np.pi / 2

def harm(t: float) -> float:
    return A * np.cos(
        2 * np.pi * F * t + theta)

def phasor(t: float) -> complex:
    return 0.5 * A * np.exp(
        -1j * (2 * np.pi * F * t + theta))

T = np.linspace(0, 1, 255)
plt.plot(T, harm(T), label="harm(T)")
plt.plot(T, phasor(T) + phasor(T).conj(),
         linestyle="--", label="harm(T)")
plt.xlabel("t")
plt.show()

```



Codeschnipsel 1: Berechnung und Darstellung von (2.1.1), siehe [code/cont_harms.py](#)

2.1.2 Zeit-Diskrete Harmonische

Als nächstes gehen wir zu dem eigentlich interessanten Fall über, bei welchem wir von zeitdiskreten harmonischen Signalen sprechen. Dabei gehen wir vorerst *nicht* davon aus, dass das Signal durch Abtastung eines analogen Signals entstanden ist, sondern betrachten es ganz losgelöst für sich. In Analogie zu (2.1.1) definieren wir

$$x[n] = A \cos(\omega n + \theta) = A \cos(2\pi f n + \theta). \quad (2.1.2)$$

Wichtig bei diskreten Signalen ist, dass ihre physikalische Interpretierbarkeit nicht direkt gegeben ist, da $n \in \mathbb{Z}$ nur die diskreten Werte „nummeriert“, also *einheitenlos* ist. Deshalb hat $f \in \mathbb{R}$ lediglich als Einheit „Zyklen pro Sample“, was man auch daran sieht, dass für $f = 1$ gilt $x[n] = A \cos(2\pi n + \theta) = A \cos(\theta)$. Es existiert nun ein wichtiger Unterschied zwischen $x[\cdot]$ von (2.1.2) und $x_a(\cdot)$ von (2.1.1). Das Signal $x[\cdot]$ ist nur periodisch, falls f eine rationale Zahl ist, also $f = p/q$ für $p, q \in \mathbb{Z}$ und $q \neq 0$.

Wieso?

Ein zeitdiskrtes Signal ist periodisch, falls $x[n+N] = x[n]$ für alle $n \in \mathbb{Z}$. Für unser Signal in (2.1.2) heißt das also, dass

$$\cos(2\pi f n + \theta) = \cos(2\pi f(n+N) + \theta) + \cos(2\pi f n + 2\pi f N + \theta)$$

Da \cos Periode $2\pi k$ für $k \in \mathbb{Z}$ besitzt, muss $2\pi f N = 2\pi k$ gelten, also

$$f = \frac{k}{N}.$$

Andersherum kann man die kleinste Periode N ermitteln, indem man $f = k/N$ vollständig kürzt, sodass also Zähler und Nenner keine gemeinsamen Teiler mehr haben, und man dann den Nenner des resultierenden Bruches als N setzt. Ein Beispiel wird in Codeschnipsel 2 gezeigt. Man kann interessante Ergebnisse erzielen, wenn man diesen Plot für $\omega = 0, \pi/8, \pi/4, \pi/2$ und $\omega = \pi$ erzeugt (Übung).

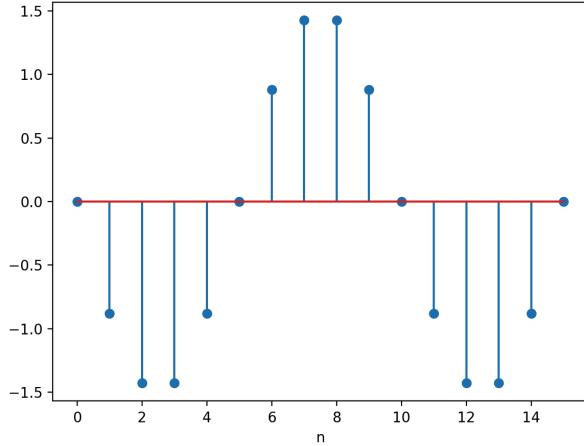
```

A = 1.5
f = 0.1
theta = np.pi / 2

def harm(n: int) -> float:
    return A * np.cos(
        2 * np.pi * f * n + theta)

n = np.arange(16)
plt.stem(n, harm(n), label="harm[n]")
plt.xlabel("n")
plt.show()

```



Codeschnipsel 2: Berechnung und Darstellung von (2.1.2), siehe [code/disc_harms.py](#)

Die Signale der Form (2.1.2) haben noch eine andere interessante Eigenschaft, die sich wieder aus der 2π -Periodizität von \cos ergibt. Betrachten wir noch einmal (2.1.2) und wir finden, dass

$$\cos(\omega n + \theta) = \cos(\omega n + 2\pi n + \theta) = \cos((\omega + 2\pi)n + \theta).$$

Das heißt, dass

$$x[n] = \cos(\omega n + \theta) = \cos((\omega + 2\pi k)n + \theta) = x_k[n]$$

für alle $k \in \mathbb{Z}$ gilt. Das heißt, dass sich $x_k[\cdot]$ nicht von $x[\cdot]$ unterscheiden lässt. Man nennt dann jedes der $x_k[\cdot]$ einen *Alias* von $x[\cdot]$. Man kann deshalb auch sagen, dass für jedes ω mit $|\omega| > \pi$ ein zugehöriges ω_a mit $|\omega_a| < \pi$ existiert, sodass

$$\cos(\omega n + \theta) = \cos(\omega_a n + \theta)$$

gilt. Vergewissern sie sich von dieser Tatsache, indem sie verschiedene Aliase basierend auf Codeschnipsel 2 visualisieren (Übung).

Stellen wir uns für einen kurzen Moment vor, dass wir wissen, dass wir $x[n]$ durch Abtastung einer Funktion wie in (2.1.1) erhalten haben. Selbst wenn wir wissen, dass nur *eine* Frequenz in diesem Signal vor Abtastung vorhanden war, können wir *nicht* entscheiden, welche das war.

2.2 Zeit-Kontinuierliche Komplexe Harmonische

Wir wollen eine bestimmte Menge an Funktionen betrachten. Wir wollen kontinuierliche komplexe Schwingungen betrachten, welche mit einer Frequenz F_k schwingen, welche ein ganzzahliges Vielfaches einer Frequenz F_0 ist. Das heißt, wir betrachten dann

$$F_k = k \cdot F_0 \quad \text{für } k \in \mathbb{Z} \quad , \text{was} \quad x_k(t) = \exp(j2\pi F_k t) = \exp(j2\pi k F_0 t)$$

ergibt. Jedes der x_k hat Periode $1/F_k = T_k = T_0/k$. Das heißt für wachsendes $|k|$ werden die Perioden immer um ein Vielfaches kürzer. Umgekehrt haben dann alle x_k gemeinsame Periode, T_0 , da für jedes k gilt, dass $T_k \cdot k = T_0$. Wir haben auch kein Problem mit Aliasing zwischen den x_k , da bei kontinuierlichen Signalen gilt, dass $x_{k_1} \neq x_{k_2}$, falls $k_1 \neq k_2$.

Wie wir in Abschnitt 1.2.2 gesehen haben, können wir beliebige Linearkombination aus Signalen bilden und erhalten wieder ein Signal. Wir können also für eine Folge von $c_k \in \mathbb{C}$ die Linearkombination der x_k bilden und erhalten

$$x_a = \sum_{k \in \mathbb{Z}} c_k x_k : \mathbb{R} \rightarrow \mathbb{C} \quad \text{mit} \quad t \mapsto x_a(t) = \sum_{k \in \mathbb{Z}} c_k x_k(t) = \sum_{k \in \mathbb{Z}} c_k \exp(j2\pi k F_0 t). \quad (2.2.1)$$

Die erste Schreibweise ist absichtlich ohne das Argument t , um zu verdeutlichen, dass Signale *wirklich* als eigenständige Signale behandelt werden können und dass $x_a(t) \in \mathbb{C}$ „nur“ die Auswertung von x_a an der Stelle t ist, welche *strikt* von dem Vektor x_a zu unterscheiden ist.

Natürlich ist (2.2.1) als Fourierreihe von x_a bekannt, und die c_k sind die Fourierkoeffizienten von x_a . Wie in Abschnitt 1.2.2 können wir also die c_k durch (2.2.1) mit x_a *identifizieren*, da uns die c_k eine alternative Darstellung von x_a liefern.

2.3 Zeit-Diskrete Komplexe Harmonische

Analog zu Abschnitt 2.2, wollen wir uns zeit-diskrete komplexe Schwingungen herleiten, die von einer bestimmten Grundfrequenz definiert werden. Da wir, im Unterschied zum kontinuierlichen Fall, nicht für alle $f \in \mathbb{R}$ eine periodische Funktion erhalten, wählen wir $f_0 = 1/N$ für ein $N \in \mathbb{N}$. Die Intension ist, dass wir so $1/N$ Perioden pro Abtastwert erhalten werden. Demzufolge wird die Schwingung mit Frequenz f_0 genau Periodenlänge N haben. Dann definieren wir die Signale $x_k[\cdot] : \mathbb{N} \rightarrow \mathbb{C}$ als

$$x_k[n] = \exp(j2\pi k f_0 n) \quad \text{mit} \quad k \in \mathbb{Z}.$$

Man sieht nun leicht, dass $f_k = k/N$ immer eine rationale Zahl ist und $x_k[\cdot]$ Periodenlänge k/N hat, falls $k/N \in \mathbb{Z}$. Außerdem findet man wieder, dass $x_k[\cdot]$ ein Alias von $x_{k+N}[\cdot]$ sein muss, denn mit $f_0 = 1/N$ ergibt sich

$$x_{k+N}[n] = \exp(j2\pi(k+N)f_0 n) = \exp\left(j2\pi \frac{k+N}{N} n\right) = \exp\left(j2\pi \frac{k}{N} n\right) \exp\left(j2\pi \frac{N}{N} n\right) = x_k[n].$$

Das heißt, dass nur N verschiedene $x_k[\cdot]$ existieren. Normalerweise nimmt man jene $x_k[\cdot]$ für $k = 0, 1, \dots, N-1$. Nun können wir auch wieder, wie in (2.2.1) eine Linearkombination der $x_k[\cdot]$ bilden. Man beachte, dass in diesem Fall die Summation natürlich *endlich* sein wird, da wir nur N verschiedene $x_k[\cdot]$ zur Verfügung haben. Wir bilden also

$$x[\cdot] = \sum_{k=0}^{N-1} c_k x_k[\cdot]$$

und erhalten so ein Signal mit Werten

$$x[n] = \sum_{k=0}^{N-1} c_k x_k[n] = \sum_{k=0}^{N-1} c_k \exp\left(j2\pi \frac{k}{N} n\right), \quad (2.3.1)$$

was die Fourierreihe eines diskreten und periodischen Signals darstellt, wobei in diesem Fall der Vektor $\mathbf{c} = [c_k]_{k=0}^{N-1} \in \mathbb{C}^N$ eine alternative Repräsentation des Signals ist.

Das Signal $x[\cdot]$ selbst ist periodisch mit Periodenlänge N , d.h. das Signal ist durch die Werte $\mathbf{x} = [x[n]]_{n=0}^{N-1} \in \mathbb{C}^N$ *vollständig* definiert. Das heißt wir können das Signal $x[\cdot]$ mit dem *endlich-dimensionalen* Vektor $\mathbf{x} \in \mathbb{C}^N$ identifizieren. Da genauso jedes der $x_k[\cdot]$ auch Periodenlänge N hat, erhalten wir auf die

gleiche Weise Vektoren $\mathbf{x}_k \in \mathbb{C}^N$. Mit diesen endlich-dimensionalen Vektoren sind wir nun in der Lage die Fourierreihe in (2.3.1) mit „normalen“ Vektoren durch

$$\mathbf{x} = c_1 \mathbf{x}_1 + c_2 \mathbf{x}_2 + \dots + c_{N-1} \mathbf{x}_{N-1} \quad (2.3.2)$$

auszudrücken. Wir sehen also *direkt*, dass Signale wirklich wie Vektoren behandelt werden können.

Gleichung (2.3.1) kann auch als Abbildung $M : \mathbb{C}^N \rightarrow \mathbb{C}^N$ interpretiert werden, da wir in die rechte Seite von (2.3.1) einfach ein $c \in \mathbb{C}^N$ stecken können und wir erhalten das entsprechende $\mathbf{x} = M(c)$. Noch weiter ist die Abbildung M sogar linear, da

$$M(\mathbf{c}^1 + \mathbf{c}^2) = \sum_{k=0}^{N-1} (c_k^1 + c_k^2) \exp\left(j2\pi \frac{k}{N} n\right) = \sum_{k=0}^{N-1} c_k^1 \mathbf{x}_k + \sum_{k=0}^{N-1} c_k^2 \mathbf{x}_k = M(\mathbf{c}^1) + M(\mathbf{c}^2).$$

Das heißt, dass es auch eine *Matrix* $M \in \mathbb{C}^{N \times N}$ geben muss, welche uns einfach c in \mathbf{x} umtransformiert, indem wir $\mathbf{x} = M \cdot c$ als Matrix-Vektor-Produkt berechnen. Wenn wir (2.3.2) genau betrachten sehen wir, dass wir die Matrix M bilden können, indem wir deren k -te Spalte $M_{\cdot,k}$ gleich \mathbf{x}_k setzen. Wenn wir noch sicher sein könnten, dass M invertierbar ist, könnten wir sogar aus \mathbf{x} via $c = M^{-1} \cdot \mathbf{x}$ die Fourierkoeffizienten c direkt aus einem gegebenen N -periodischen Signal $x[\cdot]$ bestimmen.

2.4 Finally: Abtastung von Signalen

Es gibt viele Möglichkeiten ein analoges Signal zu digitalisieren. Wir beschränken uns auf Abtastung, welche ein analoges Signal auf eine regelmäßige Art und Weise *direkt* auswertet. Diese Art wird manchmal auch „Nyquist-Sampling“ genannt, weil die theoretische Grundlage für „erfolgreiches“ Sampling durch das Nyquist-Theorem gelegt ist. Wir stellen uns Abtastung so vor, dass wir das Signal $x_a : \mathbb{R} \rightarrow \mathbb{R}$ direkt an gewissen Stellen beobachten können. Wir können uns eine Art Sampling-Operator \mathcal{S} vorstellen, der ein analoges Signal x_a in eine abgetastete Version $x_a \mapsto \mathcal{S}(x_a)[\cdot] = x[\cdot]$ transformiert. Durch die regelmäßige/uniforme Abtastung in Zeitabständen $T > 0$ von x_a erhalten wir also

$$x[n] = \mathcal{S}(x_a)[n] = x_a(nT) \quad \text{mit } n \in \mathbb{Z}.$$

Wir nennen $F_s = T^{-1}$ die Sampling-Frequenz, oder die Abtastrate. Der Vorgang ist schematisch in Abbildung 2 dargestellt.

Da wir uns $x[\cdot]$ so vorstellen, dass es „einfach“ eine Folge von reellen Zahlen ist, müssen wir bei der Interpretation von $x[\cdot]$ auch immer gleichzeitig im Hinterkopf behalten, dass der Wert $x[n]$ dem Wert $x_a(nT) = x_a(n/F_s)$ entspricht. Das bedeutet, dass t (als Argument von x_a) und n (als Argument von $x[\cdot]$) durch

$$t = nT = n/F_s$$

miteinander in Verbindung stehen. Man sieht auch nun eindrucksvoll, dass dadurch $n = t \cdot F_s$ einheitenlos geworden ist, bzw. sein muss.

Weiterhin folgt aus $t = n/F_s$, dass es auch einen Zusammenhang zwischen der Frequenz F einer kontinuierlichen Signals und der Frequenz f im Zeit-diskreten geben muss. Um diesen herzuleiten, betrachten wir eine einfache analoge Schwingung, wie in (2.1.1), also

$$x_a(t) = A \cos(2\pi F t + \theta)$$

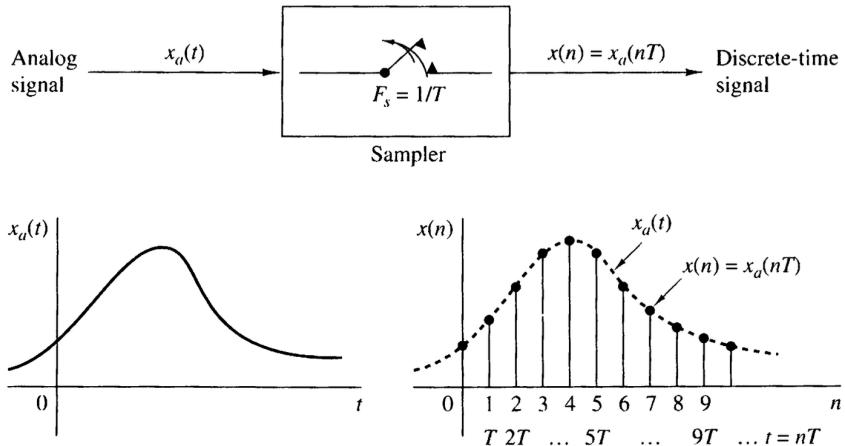


Abbildung 2: Uniforme Abtastung einer Signals. Quelle: [1]

und wir stellen uns vor, dass wir dieses Signal mit Samplerate $F_s = 1/T$ abtasten. Dann erhalten wir zunächst

$$x[n] = x_a(nT) = A \cos(2\pi F n T + \theta) = A \cos\left(\frac{2\pi F n}{F_s} + \theta\right).$$

Wenn wir nun $x[n]$ in die Form von (2.1.2) bringen, sehen wir, dass $f = F/F_s$ gelten muss.

Wie ist dies zu interpretieren? Wir sind mit dem analogen Signal x_a gestartet, welches die Frequenz F enthält. Durch das Sampling mit Rate F_s entsteht eine abgetastete Schwingung mit Frequenz $f = F/F_s$. Doch wir haben bereits gesehen, dass sich die Frequenz f von keiner Frequenzen $f + k$ unterscheiden lässt. Das heißt im Umkehrschluss, dass sich *nach* der Abtastung die ursprüngliche Frequenz nicht eindeutig bestimmen lässt, da alle

$$F_k = F + k \cdot F_s$$

bei Abtastung von $A \cos(2\pi F_k t + \theta)$ dieselben Abtastwerte $x[n]$ ergeben würden. Wenn wir nun also behaupten wollen, dass wir im digitalen irgendetwas sinnvolles zu tun gedenken, dann können wir dies gerade nicht so ohne weiteres. Denn bis jetzt haben wir keine Möglichkeit das wahre analoge Signal zu rekonstruieren. Diese missliche Lage wird in Codeschnipsel 3 dargestellt, wo ein mögliches x_a , dessen Abtastwerte $x[n]$ und zwei mögliche Aliase dargestellt sind. Man sieht, wie die Aliase so geschaffen sind, dass auch sie Ursprung für die Abtastwerte $x[n]$ sein könnten.

Die Frage ist nun, wie wir das Sampling gestalten müssen, dass wir aus $x[n]$ eindeutig das Signal x_a rekonstruieren können. In diesem Fall ist mit „rekonstruieren“ gemeint, dass wir aus den Werten $x[n]$ den Wert $x_a(t)$ für beliebige $t \in \mathbb{R}$ korrekt bestimmen können. Wie wir oben gesehen haben, ist im Digitalen nur sinnvoll von Frequenzen $f \in [-1/2, +1/2]$ zu sprechen, da wir für alle anderen $f' \notin [-1/2, +1/2]$ ein $f \in [-1/2, +1/2]$ finden, das dieselben Werte in (2.1.2) produziert. Wegen des Zusammenhangs $f = F/F_s$ macht es also Sinn sich auch im *Analogen* auf den entsprechenden Bereich zu beschränken. Das heißt, wenn wir nur analoge Signale betrachten, bei welchen

$$-\frac{F_s}{2} \leq F \leq +\frac{F_s}{2}$$

```

theta = -np.pi/2 # rad
F = 1.5 # Hz
F_s = 1.2 # Hz
T_s = 1.0 / F_s # s

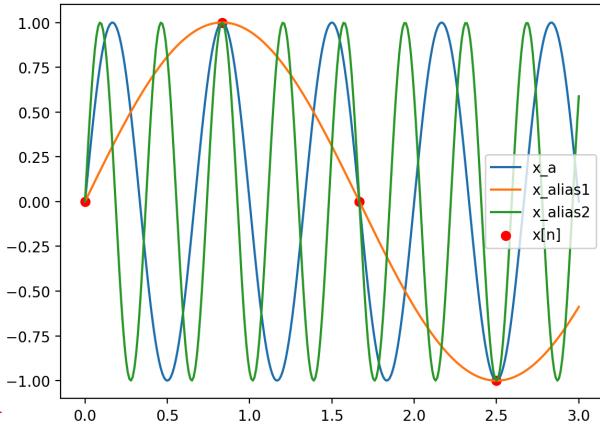
F_alias1 = F - 1 * F_s # Hz
F_alias2 = F + 1 * F_s # Hz

def x_a(t, F, theta):
    return np.cos(2 * np.pi * F * t + theta)

T = np.linspace(0, 3, 301, endpoint=True) # s
n = np.arange(4) * T_s # none

plt.plot(T, x_a(T, F, theta), label="x_a")
plt.plot(T, x_a(T, F_alias1, theta), label="x_alias1")
plt.plot(T, x_a(T, F_alias2, theta), label="x_alias2",
plt.scatter(n, x_a(n, F, theta), color="r", label="x[n]")
plt.legend()
plt.show()

```



Codeschnipsel 3: Visualisierung von $F_k = F + k \cdot F_s$, siehe [code/aliasing.py](#)

gilt, dann sind wir in der Lage aus $x[\cdot]$ das originale x_a zu rekonstruieren, weil wir wissen, auf welche der Aliase wir uns beschränken müssen.

Eine Möglichkeit der perfekten Rekonstruktion von analogen Signalen aus uniformen digitalen Abtastwerten ist die Einschränkung auf einen bestimmten Frequenzbereich.

Umgekehrt können wir auch bei Vorwissen über die maximale Frequenz F_{\max} , also $F \in [-F_{\max}, +F_{\max}]$ in einem Signal x_a die Samplingrate ermitteln, sodass im Digitalen die Aliase $f + k$ für $k \neq 0$ nicht im Bereich $[-F_{\max}, +F_{\max}]$ liegen. Damit

$$-\frac{1}{2} \leq f \leq +\frac{1}{2}$$

gilt, muss also auch

$$-\frac{1}{2} \leq \frac{F}{F_s} \leq +\frac{1}{2} \quad \text{für alle } F \in [-F_{\max}, +F_{\max}]$$

gelten. Deshalb muss schlussendlich $F_s > 2F_{\max}$ gelten.

Was nun noch fehlt ist eine analytische Formel für die Rekonstruktion von dem Signal x_a aus $x[\cdot]$.

Theorem 2.1 (Sampling Theorem). *Gegeben sei ein analoges Signal x_a mit Frequenzen in $[-F_{\max}, +F_{\max}]$ und dessen Abtastwerte $x[\cdot]$ mit $x[n] = x_a(nT)$, wobei $T = 1/F_s$.*

Falls $F_s > 2F_{\max}$, dann gilt

$$x_a(t) = \sum_{n \in \mathbb{Z}} x[n] \cdot g(t - nT), \quad (2.4.1)$$

wobei der Interpolationskern $g : \mathbb{R} \rightarrow \mathbb{R}$ gegeben ist durch

$$g_{F_s}(t) = \frac{\sin(\pi F_s t)}{\pi F_s t}.$$

```

F_max = 1.0
F = np.random.choice(
    np.linspace(-F_max, +F_max, 11, endpoint=True), 2, replace=False
) # Hz
theta = np.random.uniform(0, 2 * np.pi, 2) # rad
A = np.random.randn(2) # amplitude

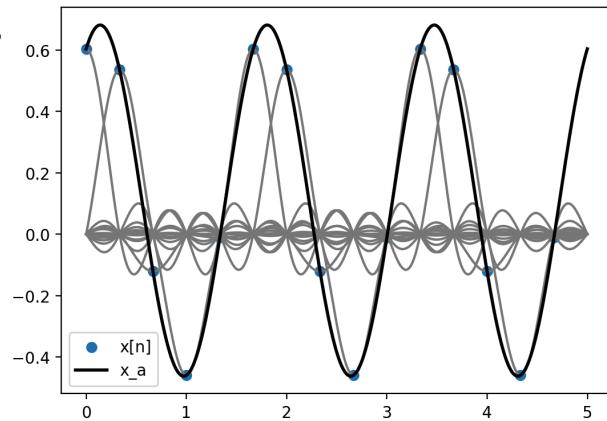
def x_a(t: np.ndarray) -> np.ndarray:
    return np.sum(np.cos(2 * np.pi * np.outer(t,
                                              np.exp(1j * theta) * A)))

t = np.linspace(0, 5, 1024)
F_s = 3

def g(t: np.ndarray) -> np.ndarray:
    nenner = np.pi * F_s * t
    nenner[np.isclose(t, 0)] = 1
    result = np.sin(np.pi * F_s * t) / nenner
    result[np.isclose(t, 0)] = 1
    return result

n = np.arange(int(t[-1] * F_s)).astype(float) / F_s
x_n = x_a(n)
for nn in np.arange(len(n)):
    plt.plot(t, x_n[nn] * g(t - nn / F_s), color="grey")
plt.scatter(n, x_n, label="x[n]")
plt.plot(t, x_a(t), color="black", linewidth=2, label="x_a")
plt.legend()
plt.show()

```



Codeschnipsel 4: Berechnung und Darstellung von Theorem 2.1, siehe [code/sampling_theorem.py](#)

Hinweis: Das Sampling Theorem, wie es in [1] formuliert ist, beinhaltet einen Fehler. Die Definition des Interpolationskernes ist dort mit

$$g(t) = \frac{\sin(2\pi Bt)}{2\pi Bt}.$$

gegeben. Doch in diesem Falle würde der Interpolationskern von der Bandbreite des Signals x_a , aber nicht von der Abtastrate F_s abhängen. Die angegebene Definition ist nur korrekt, falls $F_s = 2B$, wir also mit der *minimalen* Samplerate abgetastet haben. In diesem Fall spricht man auch von *kritischer Abtastung*.

3 Diskrete Signale und Systeme

[1, chp. 2.1]

3.1 Diskrete Signale

- darstellung: funktion, tabelle, folge, grafisch, z-trafo (Abschnitt 4) alle gleichwertig, aber verschiedenen nutzlich, je nach situation
- beispiele: delta, heavyside, ramp, exponential
- energie in signal, durchschnittliche leistung, $E < \infty \Rightarrow P = 0$
- beispiel: heavyside,
- Übung: Problem 2.1, 2.2, 2.3
- periodic, aperiodic
- symmetrisch, anti-symmetrisch
- manipulation: verzoegerung (+-), skalierung, addition, multiplikation

3.2 Diskrete Systeme

- JEDE manipulation eines signals kann als system aufgefasst werden
- system \mathcal{T} ist als abbildung zu verstehen. signal rein, anderes signal raus: $y[n] = (\mathcal{T}x)[n]$ (auf bedeutung von schreibweise eingehen)
- beispiele: [1, ex. 2.2.1] (a-c vorlesung, rest uebung)
- vergangenheit der systeme: akkumulator
- blockschaltbilder: adder, multiplikator, scaler, delay, uebung [1, ex 2.2.3](per hand, per modularem code)
- statisch vs. dynamisch (gedaechtnislos, mit gedaechtnis)
- causal, antikausal, relaxed state
- stabilitaet
- zeitvariant vs zeitinvariant: definition

3.3 Diskrete LTI Systeme

- wiederholung der eigenschaften: shift-invarianz, linearitaet
- reaktion auf signal, das aus bausteinsignalen zusammengesetzt wurde
- darstellung als summe von skalierten, geshifteten einheitsstoessen
- faltungsformel
- algorithmus zur berechnung
- exerzieren von beispiel 2.3.2, uebung programmieren (naiv summieren, scipy convolve)
- exerzieren von beispiel 2.3.3, uebung programmieren der approximation
- assoziativitaet, kommmutativitaet
- cascadierung von mehreren systemen
- stabilitaet: $h[n]$ muss absolut summierbar sein, gegen 0 gehen
- beispiel 2.3.6. (fuer uebung irgendwas ausdenken)

3.4 Cross- und Autokorrelation

- Definition
- eigenschaften
- synthetisches beispiel schwingung + noise, akf zeigt periodizitaet
- beispiel mit woelfer sunspot numbers
- beispiel mit m-sequenzen, niklas einladen, schaltung zeigen
- monster uebung: 2.65

4 z-Transformation

- Wir wollen diskrete LTI systeme analysieren
- Wir wollen gewisse signale vlt kompakter aufschreiben koennen
- Definition, ROC, man kann an koeffizienten von z^n die werte an zeitpunkt n ablesen
- Beispiele mit endlichen signalen, tabelle [1, p155, top]
- beispiel mit unendlichem signal
- analyse von ROC durch betrachtung von $|X(z)|$, tabelle [1, p155, top]
- eigenschaften: linear, timeshift (intuition mit koeffizienten), zeitumkehrung(intuition mit spiegelung am einheitskreis), ableitung in z-Bereich (analogie zu fourier)
- faltungseigenschaft
- algo: beide sequenzen z-trafo, multiplikation, inverse z-trafo
- inverse: tabelle [1, tabelle 3.3], allgemeine invertierung: schwierig
- alle eigenschaften: [1, tabelle 3.2]
- anwendung: korrelation von signalen
- uebung: problem 3.6 in [1], initial value theorem, some plots aus [1]
- rationale z-trafos: definition, pole-nullstellen, umkehrung: von pole-nullstellen zu $X(z)$
- ausfuehrliche diskussion von [1, fig 3.3.5, 3.3.6]
- Anwendung: M-Sequenzen? polynome analysieren, wann maximale laenge?

5 Fourier-Transformation

- Fourier Transform (theoretisches tool, weil digital nicht umsetzbar): kernel, eigenschaften (linear, shift, skalierung), beispiele uebung
- Discrete Time Fourier Transform (theoretisches tool, weil digital nicht umsetzbar)
- Discrete Fourier Transform
 - eigenschaften der kernel-funktion: perioden, wann reel, ..., was passiert bei abtastung?
 - implizite periodifizierung des signals?
 - zusammenhang zur physik
 - dc, nyquist-frequenzen, wann vorhanden?
 - reelles signal: dc und nyquist reell
- zusammenhang zwischen ft, dtft und dft (intuition mit periodifizierung vs. abtastung; non-intuition durch kernel angucken)
- dft als approximation der ft
 - $s(t) = \exp(-t/\tau) \cdot \sin(\omega_0 \cdot t)$,
 - $s(t) = \exp(-(t - \mu)^2/\sigma^2)$ (Uebung)
- spectral leakage;

5.1 Anwendung der Transformationen

- fft (Uebung 1 jup, uebung 2 8.3.2)
- bild-kompression (DCT, JPEG)
- fensterung ¹
- schnelle faltung (cyclic/non-cyclic)
- beliebig lange signale overlap add, overlap save (Übung: irgendwas mit audio prosessing)
- Unschaerfe Relation (Fensterbreite vs. Frequenzauflösung)[2, chpt. 4.2]

5.2 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

$$y(nT) = x[n] = x(nT) \quad (5.2.1)$$

¹<https://docs.scipy.org/doc/scipy/reference/signal.windows.html>

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

$$x(t) = \sum_{k=-\infty}^{+\infty} c[k] \exp(j2\pi ktF_0). \quad (5.2.2)$$

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

$$x[n] = x(nT) = \sum_{k=-\infty}^{+\infty} c[k] \exp(j2\pi knTF_0) = \sum_{k=-\infty}^{+\infty} c[k] \exp\left(j2\pi k \frac{n}{N}\right) \quad \text{für } n \in \mathbb{N} \quad (5.2.3)$$

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

$$x[n] = \sum_{k=0}^{N-1} \left[\sum_{\ell=-\infty}^{+\infty} c[k - \ell N] \right] \exp\left(j2\pi k \frac{n}{N}\right) = \sum_{k=0}^{N-1} \tilde{c}[k] \exp\left(j2\pi k \frac{n}{N}\right), \quad (5.2.4)$$

wobei wir

$$\tilde{c}[k] = \sum_{\ell=-\infty}^{+\infty} c[k - \ell N] \quad (5.2.5)$$

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

$$X(F) = \int_{-\infty}^{+\infty} x(t) \exp(-j2\pi Ft) dt = 0 \quad \text{für } |F| > B. \quad (5.2.6)$$

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

$$c[k] = \frac{1}{T_p} X(kF_0), \quad (5.2.7)$$

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

$$c[k] = 0 \quad \text{für } |k| > \frac{B}{F_0}. \quad (5.2.8)$$

Das heißt, dass wir nun $F_s = N/T_p$ so groß wählen müssen, dass sich in (5.2.5) kein Aliasing für $\tilde{c}[k]$ ergeben darf. Es muss also gelten

$$N > \lceil B/F_0 \rceil \quad \text{bzw. } F_s > \lceil B/(F_0 T_p) \rceil. \quad (5.2.9)$$

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 (5.2.2) können wir also die Folge $x[n]$ interpolieren, indem wir

$$y(t) = \frac{1}{T_p} \sum_{k=-\frac{B}{F_0}}^{+\frac{B}{F_0}} X[k] \exp(j2\pi ktF_0) \quad (5.2.10)$$

schreiben. Dieses y erfüllt die Interpolationsbedingung (5.2.1), weil wegen der Bandbegrenzung von x und der Periodizität sogar $y(t) = x(t)$ für alle t gilt.

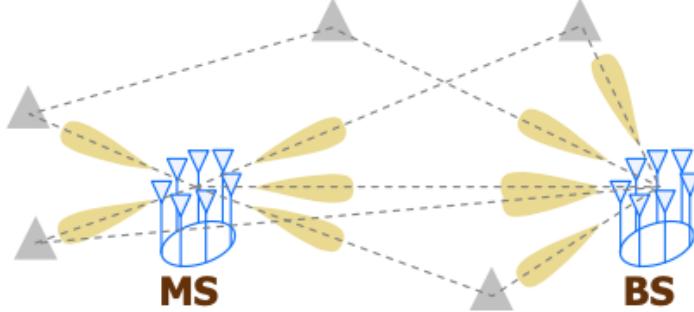


Abbildung 3: Schematische Darstellung einer **MIMO** Kanalmessung. Grafik aus [5].

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)** [3] der Folge $x[n]$. Eine Anwendung der hier vorgestellten Methode zur Interpolation wird in Abschnitt 5.3 aufgezeigt.

5.3 DFT-Interpolation von Antennenantworten

5.3.1 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 [4] 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 [6], 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 [7], 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 **LTI** System handelt. Betrachtet man nun das Verhalten des Systems im Frequenzbereich für ein einzelnes Paar von Sende- und Empfangsantenne, dann gilt demnach zunächst

$$Y(f) = G_{\text{rx}}(f) \cdot H(f) \cdot G_{\text{tx}}(f) \cdot X(f). \quad (5.3.1)$$

Hierbei steht X für die Anregung des Systems durch ein eingegebenes Signal, $G_{tx/rx}$ 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* Richtcharakteristik besitzt. Das heißt, dass die Systemantworten $G_{tx/rx}$ 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 Ω_{tx} ausbreitet und nach ihrem Weg durch den Funkkanal am Empfänger aus normierter Richtung Ω_{rx} eintrifft. Folglich ergibt sich für dieses Verhalten

$$Y(f, \Omega_{tx}, \Omega_{rx}) = G_{rx}(f) \cdot a_{rx}(f, \Omega_{rx}) \cdot H(f, \Omega_{tx}, \Omega_{rx}) \cdot a_{tx}(f, \Omega_{tx}) \cdot G_{tx}(f) \cdot X(f), \quad (5.3.2)$$

wobei $a_{tx/rx}$ für die richtungs- und frequenzabhängige Antwort der Sende- und Empfangsantenne stehen. In diesem Fall bezeichnet also $H(f, \Omega_{tx}, \Omega_{rx})$ die Transferfunktion eines einzelnen Pfades, der den Sender in Richtung Ω_{tx} verlässt und am Empfänger aus Richtung Ω_{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

$$Y(f) = \sum_{s=1}^S Y(f, \Omega_{tx,s}, \Omega_{rx,s}), \quad (5.3.3)$$

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 (5.3.2), dass wir eine möglichst präzise Formulierung für $a_{tx/rx}$ benötigen, um die Transferfunktion des Kanals H korrekt bestimmen zu können.

5.3.2 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 $a(\varphi, \theta) \in \mathbb{C}$, 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 [4, 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

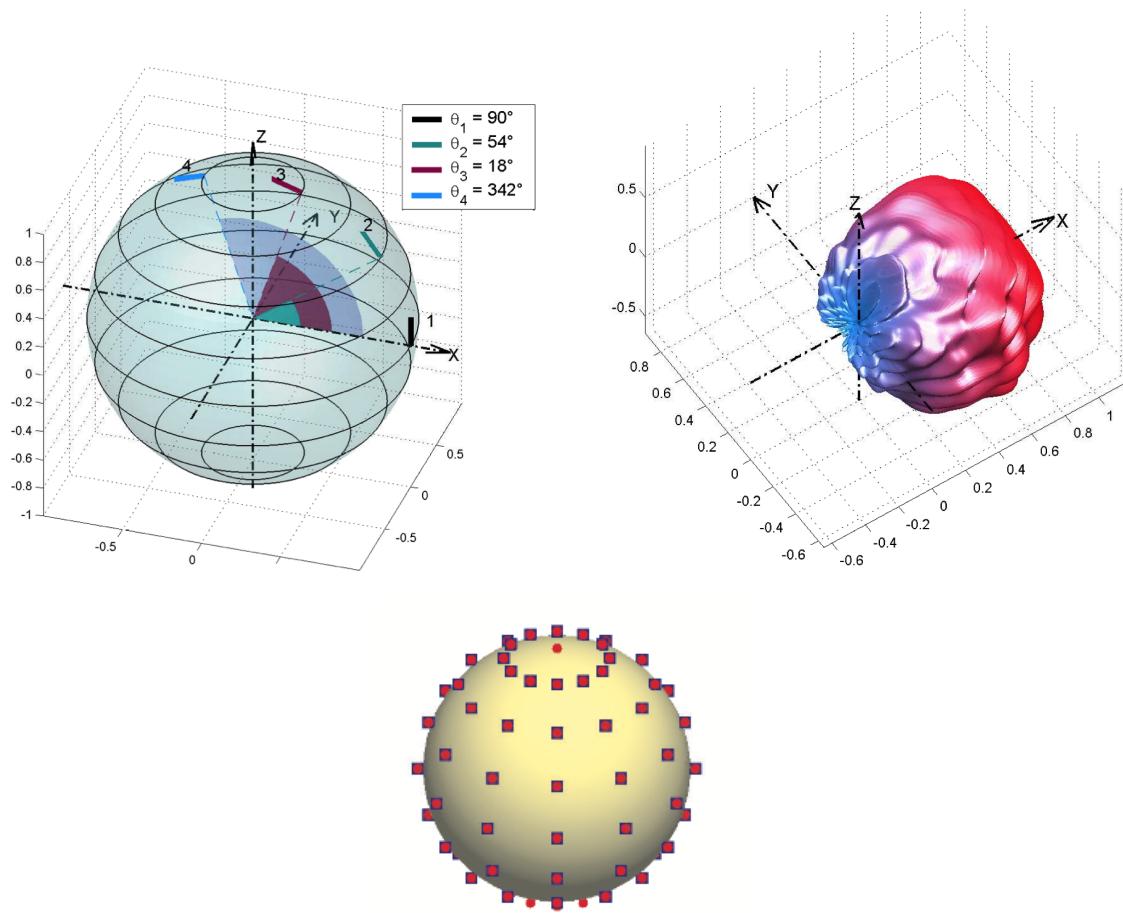


Abbildung 4: Links Oben: Darstellung der Messpositionen für eine Antenne, oder eine Antennenstruktur, welche sich im Ursprung des abgebildeten Koordinatensystems befindet. Rechts Oben: Darstellung der winkelabhängigen Amplitude eines einzelnen Patch-Elements. Unten: Messpunkte für die Abtastung der Funktion a . Grafiken aus [8, 4].

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.

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 drehen, 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

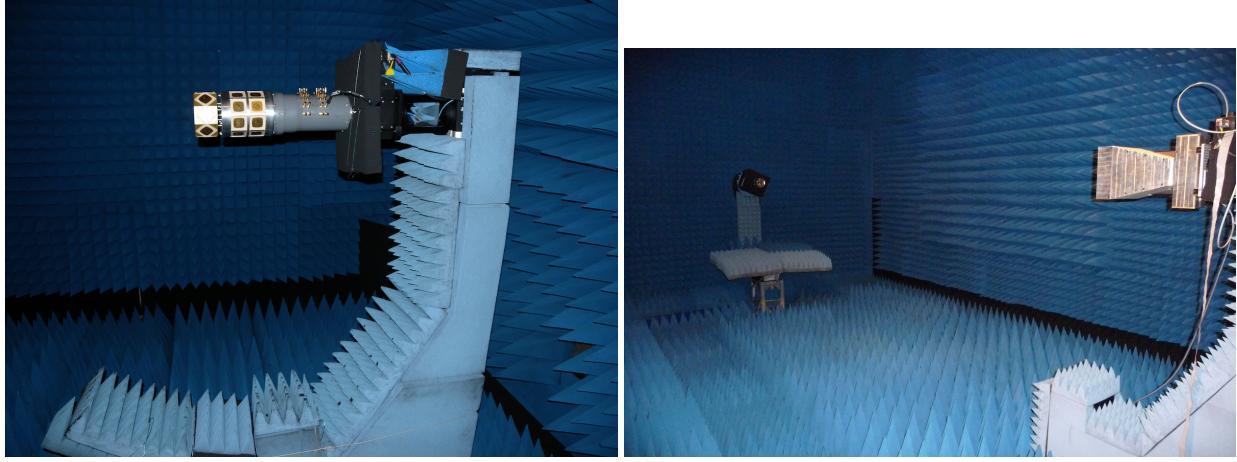


Abbildung 5: Messaufbau zur Kalibrierung eines Antennenarrays. Links: Drehteller für die Positionierung der **AUT**. Rechts: Weiterer Blickwinkel mit Referenzantenne.

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 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.

5.3.3 Ableitung der EADF

Wir wollen nun (5.2.10) aus Abschnitt 5.2 benutzen und auf zwei Dimensionen erweitern. Wir folgen damit effektiv [8]. 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 2π -periodisch ist.

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

$$A_\varphi(F_\varphi, \vartheta) = \int_{-\infty}^{+\infty} a(\varphi, \vartheta) \exp(-j2\pi F_\varphi \varphi) d\varphi = 0 \quad \text{für } F_\varphi > B_\varphi \quad \text{und alle } \vartheta \in [0, 2\pi] \quad \text{und} \quad (5.3.4)$$

$$A_\vartheta(\varphi, F_\vartheta) = \int_{-\infty}^{+\infty} a(\varphi, \vartheta) \exp(-j2\pi F_\vartheta \vartheta) d\vartheta = 0 \quad \text{für } F_\vartheta > B_\vartheta \quad \text{und alle } \varphi \in [0, 2\pi] \quad (5.3.5)$$

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

$$a(\varphi, \vartheta) = \sum_{k_\varphi=-\frac{B_\varphi}{F_\varphi}}^{+\frac{B_\varphi}{F_\varphi}} \sum_{k_\vartheta=-\frac{B_\vartheta}{F_\vartheta}}^{+\frac{B_\vartheta}{F_\vartheta}} A[k_\varphi, k_\vartheta] \cdot \exp(j2\pi k_\varphi \varphi F_\varphi) \cdot \exp(j2\pi k_\vartheta \vartheta F_\vartheta), \quad (5.3.6)$$

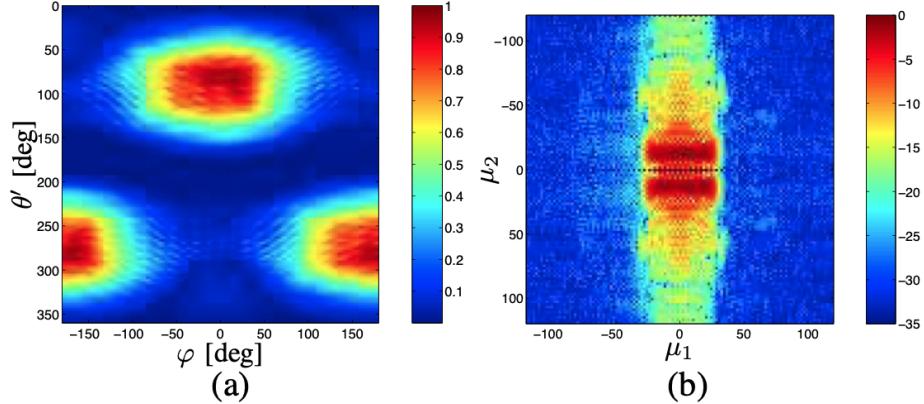


Abbildung 6: Links: Periodifiziertes 2D Array $a[n_\varphi, n_\theta]$ der gemessenen Amplituden eines einzelnen Patch-Elements. Rechts: Der Betrag der zugehörigen EADF, wobei hier $\mu_1 = k_\varphi$ und $\mu_2 = k_\theta$. Grafik aus [8]

welche eine absolut analoge (nicht als Gegenteil zu digitale) 2D-Version zu (5.2.10) darstellt. Auch in diesem Fall, können wir das 2D-Array $A[k_\varphi, k_\theta]$ durch eine 2D-DFT, bzw. der FFT [3], von den uniformen Samples $a[n_\varphi, n_\theta]$ der Funktion a effizient vorberechnen.

Um einen möglichst effizienten Algorithmus für die Auswertung der Interpolante zu erhalten, sollte man (5.3.6) geeignet umschreiben. Moderne Rechenarchitekturen und Scientific-Computing-Libraries sind auf schnelle Matrix-Vektor-Produkte optimiert. Nehmen wir an, wir wollen (5.3.6) für mehrere Winkelpaare $(\varphi_1, \theta_1), \dots, (\varphi_L, \theta_L)$ auswerten. Dann berechnen wir zunächst zwei 2D Arrays

$$D_\varphi = [\exp(j2\pi k_\varphi \varphi F_\varphi)]_{\ell=1, k_\varphi=-\frac{B_\varphi}{F_\varphi}} \in \mathbb{C}^{L \times 2^{\frac{B_\varphi}{F_\varphi}+1}} \quad \text{und} \quad (5.3.7)$$

$$D_\theta = [\exp(j2\pi k_\theta \theta F_\theta)]_{\ell=1, k_\theta=-\frac{B_\theta}{F_\theta}} \in \mathbb{C}^{L \times 2^{\frac{B_\theta}{F_\theta}+1}}, \quad (5.3.8)$$

was uns erlaubt (5.3.6) in das folgende Vektor-Matrix-Vektor-Produkt

$$a(\varphi_\ell, \theta_\ell) = D_\varphi[\ell, :] \cdot A[:, :] \cdot D_\theta[\ell, :]^\top \quad (5.3.9)$$

umzuschreiben. Damit besteht der Interpolations-Algorithmus zunächst aus der Vorberechnung des Arrays A , sowie bei Ausführung dann aus der Berechnung von D_φ und D_θ , sowie der Auswertung von (5.3.9).

Man sieht hier der schön, dass die Laufzeitkomplexität von (5.3.9) 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 9 dargestellt.

6 Multiraten Systeme

- system das signale in mehreren raten gleichzeitig benutzt
- erste option: digital zu analog, filtern, analog zu digital
- vorteil: sampling rate kann beliebig (je nach signal bandbreite) veraendert werden
- nachteil: distortion von da, ad conversion introduce noise, quantization errors
- also: conversion direkt in digitaler domaene
- problem: how? wir haben das signal nur auf diskreten werten gegeben
- wir koennen aber mal die formel fuer perfekte rekonstruktion aufschreiben, die das richtige tut unter der bedingung, dass sampling rate nyquist einhaelt
- dann koennen wir diese interpolation einfach auf den neuen samplingpunkten auswerten
- resampling ist immerhin schonmal ein lineares system
- falls neue samplingrate kleiner, muessen wir auch hier noch nyquist einhalten, oder eben die formel selbst lowpass-filtern
- falls alte samplingrate = neue samplingrate haben wir ein LTI-system (uebung)
- wenn die samplingrate beliebgi sind, muss man genau hinschauen.
- erstmal bild malen
- dann gleichungen rumwursten
- lineares zeitvariantes system, also impulsantwort ist zeitabhaengig
- dann nochmal bild anschauen
- fuer rationales verhaeltnis: intuitiv kommt man je nach evrhaltnis wieder auf die gleichen fractional parts, also ist das resampling ein periodisch zeitvariantes lineares system
- downsampling um faktor D (in uebung programmieren, einmal mit weglassen und einmal als faltung)
- upsampling um faktor I liefert I verschiedene Impulsantworten des systems
- beispiel mit $I = 2$ in der vorlesung (Figure 11.1.4.!), beispiel mit $I = 3$ in uebung. einmal ohne interleaving als periodisch zeitvariantes system, einmal mit interleaving

7 STFT

8 Wavelets

- motivation: analyse in zeit und frequenz gleichzeitig
- definition [2, chpt 4.3, (4.30)]
- convolution, i.e. linear filter
- real heisenberg uncertainty: heisenbergboxes
- mexican hat wavelet: second derivative of gaussian
- gabor wavelets (uebung)
- discrete wavelet transform [2, ch 4.3.3]

9 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 [1, 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 5.2.

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 [9]. Wir führen zunächst die auf Polynomen basierende Signalverarbeitung ein und vergleichen sie anschließend zur bereits bekannten Nyquist-Theorie.

9.1 B-Splines als Polynome

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

$$p(x) = a_\ell t^\ell + a_{\ell-1} t^{\ell-1} + \cdots + a_1 t + a_0. \quad (9.1.1)$$

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

$$s(t) = \begin{cases} p_1(t) & \text{für } x \in [1, 2], \\ p_2(t) & \text{für } x \in [2, 3], \\ \vdots & \end{cases} \quad (9.1.2)$$

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

$$\lim_{t \rightarrow n^-} s^{(m)}(t) = \lim_{t \rightarrow n^+} s^{(m)}(t) \quad (9.1.3)$$

erfüllt ist, wobei $s^{(m)}$ für $m \geq 0$ die m -te Ableitung des Splines s repräsentiert. In einer Arbeit [10], die sogar dem berühmten Paper von Shannon vorausgeht, beschreibt Schoenberg, dass sich diese Splines der Ordnung ℓ via

$$s(t) = \sum_{k \in \mathbb{Z}} c[k] \beta^\ell(t - k) \quad (9.1.4)$$

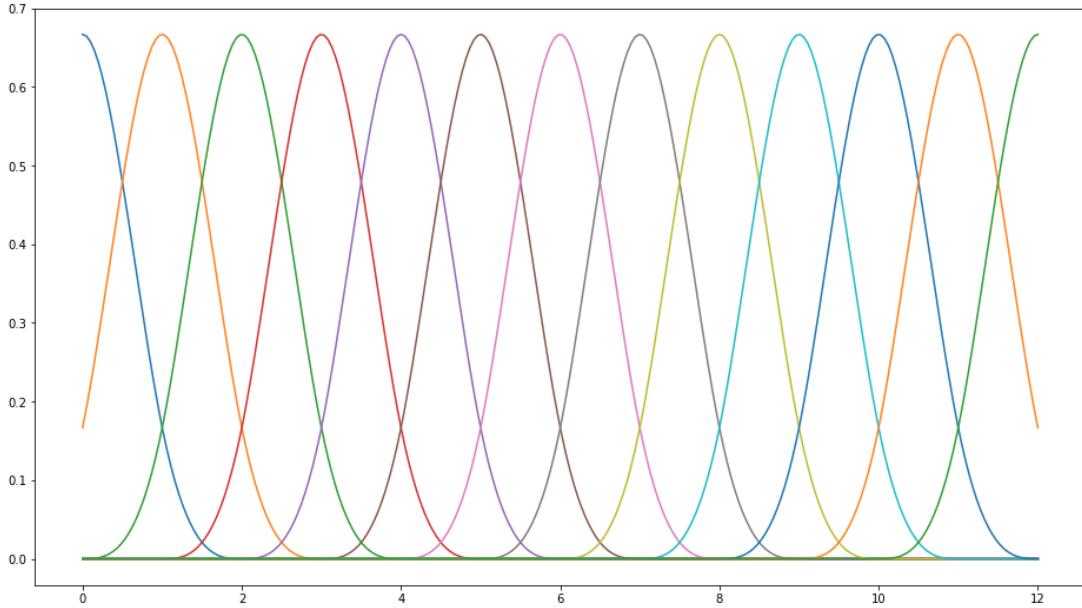


Abbildung 7: Kubische B-Splines für Abtastung an den Werten $n = 0, \dots, 12$.

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

$$\beta^\ell = \underbrace{\beta^0 * \dots * \beta^0}_{(\ell+1) \text{ mal}}, \quad \text{wobei } \beta^0(t) = \begin{cases} 1, & |t| < \frac{1}{2} \\ \frac{1}{2}, & |t| = \frac{1}{2} \\ 0, & \text{sonst.} \end{cases} \quad (9.1.5)$$

In Abb. 7 sind die Funktionen β^ℓ 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 (9.1.4) ist für fixes $t \in \mathbb{R}$ *endlich* und ist auf $\ell + 1$ Summanden beschränkt! Wir wollen nun eine explizite Formel für β^ℓ entwickeln. Hierzu betrachten wir die Fourier-Transformation

$$B^\ell(\omega) = \left(\frac{\sin(\omega/2)}{(\omega/2)} \right)^{\ell+1} = \frac{(\exp(j\omega/2) - \exp(-j\omega/2))^{\ell+1}}{(j\omega)^{\ell+1}} \quad (9.1.6)$$

mit einigen Rechentricks (siehe [9, Box 1.]) kann man dies so lange umformen, bis man

$$\beta^\ell(t) = \frac{1}{\ell!} \sum_{p=0}^{\ell+1} \binom{\ell+1}{p} (-1)^p \left(t - p + \frac{\ell+1}{2} \right)_+^\ell \quad \text{mit } (x)_+ = \begin{cases} x, & \text{für } x \geq 0, \\ 0, & \text{sonst,} \end{cases} \quad (9.1.7)$$

erhält. Damit ist β^ℓ wirklich ein Polynom ℓ -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:

$$(\beta^\ell)'(t) = \beta^{\ell-1}(x + 1/2) - \beta^{\ell-1}(x - 1/2), \quad \int_{-\infty}^t \beta^\ell(s) ds = \sum_{p=0}^{+\infty} \beta^{\ell+1}(t - 1/2 - p). \quad (9.1.8)$$

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 (9.1.4) und (9.1.8) beispielsweise für die Differentiation, dass

$$s'(t) = \sum_{k \in \mathbb{Z}} c[k] (\beta^\ell)'(t - k) = \sum_{k \in \mathbb{Z}} c[k] (\beta^{\ell-1}(t + 1/2 - k) - \beta^{\ell-1}(t - 1/2 - k)). \quad (9.1.9)$$

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

9.2 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

$$\beta^3(t) = \begin{cases} \frac{2}{3} - |x|^2 + \frac{|x|^3}{2} & \text{für } |x| < 1 \\ \frac{(2-|x|)^3}{6}, & \text{für } |x| \in [1, 2) \\ 0, & \text{für } |x| > 2, \end{cases} \quad (9.2.1)$$

welche in Codeschnipsel 5 auch einmal implementiert wurde. In Analogie zum bekannten Nyquist-Sampling wollen wir untersuchen, wie wir aus endlich vielen gegebenen Abtastwerten $x[n]$ mit $n \leq N$ eine Darstellung wie in (9.1.4) 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 x und ihre Abtastwerte $x[n] = x(n)$ fordert, dass

$$x(n) = x[n] = s(n) = \sum_{k \in \mathbb{Z}} c[k] \beta^3(n - k) \quad (9.2.2)$$

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

$$B \cdot c = y \quad (9.2.3)$$

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 β^ℓ ist es zu verdanken, dass die Matrix B mit nur wenigen von 0 verschiedenen Werten besetzt ist. (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.

```
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)
```

Codeschnipsel 5: Berechnung des kubischen B-Splines.

Sehen wir uns (9.2.2) genauer an. Wir finden, dass sich diese Gleichung nach Definition von $\beta[k] = \beta^3(k)$ als Faltung via

$$x[n] = (c * b)[k] \quad (9.2.4)$$

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

$$X(z) = C(z) \cdot B(z) \Rightarrow C(z) = \frac{X(z)}{B(z)}, \quad (9.2.5)$$

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

$$B(z) = \frac{z + 4 + z^{-1}}{6} \Rightarrow \frac{1}{B(z)} = 6 \left(\frac{1}{1 - z_1 z^{-1}} \right) \left(\frac{-z_1}{1 - z_1 z} \right) \quad (9.2.6)$$

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 (9.2.6) 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

$$c^+[k] = x[n] + z_1 c^+[k-1] \quad \text{für } k = 1, \dots, N-1 \quad (9.2.7)$$

$$-c[k]/6 = c^-[k] = z_1 (c^-[k+1] - c^+[k]) \quad \text{für } k = N-2, \dots, 0 \quad (9.2.8)$$

ausdrücken lässt. Diese Methode der kausalen und anti-kausalen Filterung ist deutlich effizienter und stabiler, als (9.2.3), da beispielsweise keine Divisionen notwendig sind. Nun ist es noch notwendig Anfangswerte für $c^+[k]$ 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

$$c^+[0] = \sum_{k=0}^{\infty} x[n] z_1^k \approx \sum_{k=0}^K x[n] z_1^k, \quad (9.2.9)$$

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

$$c^-[N-1] = \frac{z_1}{1 - z_1^2} (c^+[N-1] + z_1 c^+[N-2]) \quad (9.2.10)$$

effizient und exakt initialisieren. Beides wurde in Codeschnipsel 6 beispielhaft implementiert und eine beispielhafte Ausgabe nach Auswertung von (9.1.4) ist in Abb. 8 dargestellt.

```

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

```

Codeschnipsel 6: Berechnung der B-Spline Koeffizienten $c[k]$.

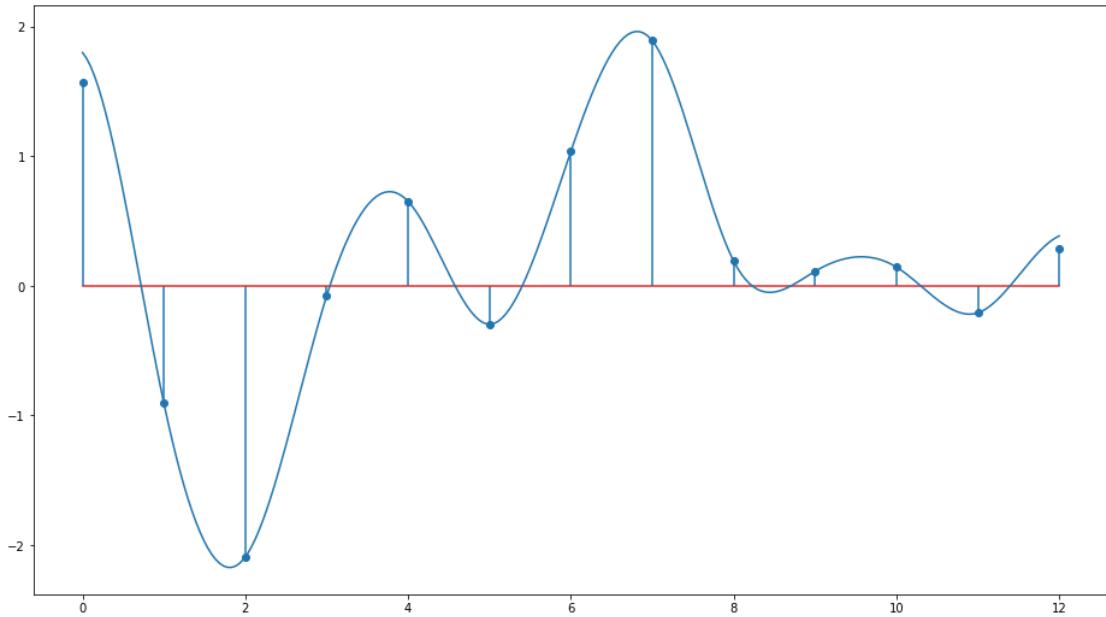


Abbildung 8: Kubische B-Spline-Interpolation für Abtastung an den Werten $n = 0, \dots, 12$.

9.3 Verbindung zur Nyquist-Sampling-Theorie

Wir wollen als Abschluss eine Verbindung zum Sampling und der Interpolation [1, 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

$$X(f) = F_s \sum_{k=-\infty}^{+\infty} X_a((f - k)F_s), \quad (9.3.1)$$

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 $F_s \cdot X_a(f) = X(f)$. Außerdem können wir das Spektrum der abgetasteten Werte X durch

$$X(f) = \sum_{n=-\infty}^{+\infty} x[n] \exp(-j2\pi f n / F_s) \quad (9.3.2)$$

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

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

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

$$g(t) = \frac{\sin(\pi t/T)}{(\pi t/T)} \quad (9.3.3)$$

als *Interpolations-Kernel* von bandbegrenzten und abgetasteten Funktionen betrachtet werden kann. Siehe Abschnitt 5.3 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 (9.2.5) definieren. Dann gilt

$$c[k] = (b^{-1} * x)[k], \quad (9.3.4)$$

was wir in (9.2.2) einsetzen und dann

$$s(t) = \sum_{n \in \mathbb{Z}} (b^{-1} * x)[n] \beta^\ell(t - n) = \sum_{n \in \mathbb{Z}} x[n] \sum_{p \in \mathbb{Z}} b^{-1,\ell}[p] \beta^\ell(t - n - p) = \sum_{n \in \mathbb{Z}} x[n] h^\ell(t - n) \quad (9.3.5)$$

erhalten, wobei wir analog zu (9.3.3) den Interpolationskernel $h : \mathbb{R} \rightarrow \mathbb{R}$ durch

$$h^\ell(t) = \sum_{p \in \mathbb{Z}} b^{-1,\ell}[p] \beta^\ell(t - p) \quad (9.3.6)$$

definiert haben. Man kann zeigen, dass $\lim_{l \rightarrow \infty} h^l = g$. 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

$$H^l(\omega) = \left(\frac{\sin(\omega/2)}{\omega/2} \right)^{\ell+1} \frac{1}{B^\ell(\exp(j\omega))} \xrightarrow{\ell \rightarrow \infty} \text{rect}(\omega) \quad (9.3.7)$$

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 (9.2.6), als auch der Synthese (9.1.4) vor allem für hochdimensionale Interpolationen sehr interessant.

10 Zufällige Signale

- zufallsgroessen: diskret, kontinuierlich (Uebung: laplace, binomial, normal)
- dichte von kontinuierlichen zufallsgroessen (Uebung: normalverteilung)
- fourier-trafo von dichten
- power estimation
- quantization

Akronyme

ADC Analog-to-Digital Converter. 7

AUT Antenna Under Test. 23, 24

CMOS Complementary Metal Oxide Semiconductor. 7

DFT Discrete Fourier Transform. 20, 25

DTFT Discrete Time Fourier Transform. 34

EADF Effective Aperture Distribution Function. 25

FFT Fast Fourier Transform. 21, 25

LTI Linear Time-Invariant. 6, 17, 18, 21, 22

MIMO Multiple Input Multiple Output. 21

Literatur

- [1] J. Proakis und D. Manolakis. *Digital Signal Processing*. Pearson Deutschland, 2013. URL: <https://elibrary.pearson.de/book/99.150005/9781292038162> (siehe S. 8, 13, 15, 16, 18, 29, 34).
- [2] S. Mallat. *A Wavelet Tour of Signal Processing*. 3rd. Elsevier, 2008. DOI: [10.1016/b978-0-12-466606-1.x5000-4](https://doi.org/10.1016/b978-0-12-466606-1.x5000-4). URL: <https://doi.org/10.1016/b978-0-12-466606-1.x5000-4> (siehe S. 19, 28).
- [3] M. Frigo und S. Johnson. „The Design and Implementation of FFTW3“. In: *Proc. IEEE* 2 (2005). Special issue on “Program Generation, Optimization, and Platform Adaptation”. DOI: [10.1109/jproc.2004.840301](https://doi.org/10.1109/jproc.2004.840301). URL: <https://doi.org/10.1109/jproc.2004.840301> (siehe S. 21, 25).
- [4] G. Del Galdo Prof. Dr.-Ing. „Geometry-Based Channel Modeling for Multi-User MIMO Systems and Applications“. en. Diss. 2007. URL: https://www.db-thueringen.de/receive/dbt_mods_00011002 (siehe S. 21–23).
- [5] A. Richter. „Estimation of Radio Channel Parameters“. In: (2005). URL: https://www.db-thueringen.de/receive/dbt_mods_00004815 (siehe S. 21).
- [6] M. Haardt, R. Thomä und A. Richter. „Multidimensional High-Resolution Parameter Estimation with Applications to Channel Sounding“. English. In: *High-Resolution and Robust Signal Processing*. Hrsg. von Y. Hua. 2003 (siehe S. 21).
- [7] S. Semper, M. Döbereiner, C. Steinmetz, M. Landmann und R. Thomä. „High Resolution Parameter Estimation for Wideband Radio Channel Sounding“. In: *IEEE Transactions on Antennas and Propagation* (2023). DOI: [10.1109/TAP.2023.3286024](https://doi.org/10.1109/TAP.2023.3286024) (siehe S. 21).
- [8] M. Landmann und G. D. Galdo. „Efficient antenna description for MIMO channel modelling and estimation“. In: *7th European Conference on Wireless Technology, 2004*. 2004 (siehe S. 23–25).
- [9] M. Unser. „Splines: a perfect fit for signal and image processing“. In: *IEEE Signal Processing Magazine* 6 (1999). DOI: [10.1109/79.799930](https://doi.org/10.1109/79.799930) (siehe S. 29, 30).
- [10] I. J. Schoenberg. „Contributions to the Problem of Approximation of Equidistant Data by Analytic Functions“. In: *I. J. Schoenberg Selected Papers*. Hrsg. von C. de Boor. Birkhäuser Boston, 1988. DOI: [10.1007/978-1-4899-0433-1_1](https://doi.org/10.1007/978-1-4899-0433-1_1). URL: https://doi.org/10.1007/978-1-4899-0433-1_1 (siehe S. 29).