Skript Digitale Signalverarbeitung

Sebastian Semper – FG Elektrische Messtechnik und Signalverarbeitung – EMS

15. Oktober 2024

Inhaltsverzeichnis

Abbildungsverzeichnis

Codeschnipsel-Verzeichnis

Allgemeine Hinweise

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.
Eine stetig aktualisierte Version des Skriptes findet man unter `https://github.com/SebastianSemper/lecturenotes`.
Als Begleitmaterial ist auch eine Auswahl an Codeschnipseln bereitgestellt, die einerseits in Auszügen im Skript direkt eingebunden sind, aber auch unter dem obigen Link aufzufinden sind.
Zum Ausführen der Codeschnipsel empfehlen wir ein Python-Environment[a], in welchem folgende Pakete installiert sein sollten:

- `numpy` – `https://numpy.org`
- `scipy` – `https://scipy.org`
- `matplotlib` – `https://matplotlib.org`
- `cupy` – `https://cupy.dev` (optional für ZOOMG GPU speed)
- `sympy` – `https://sympy.org` (optional für symbolisches Rechnen in Python)

Diese können ganz einfach via

```
conda create -n dsv python=3.10
conda activate dsv
python -m pip install numpy scipy matplotlib
```

installiert werden.
Alternativ steht unter `https://jup.rz.tu-ilmenau.de/hub/login` ein Jupyter-Hub zur Verfügung, der eine Python IDE im Browser bereitstellt.

[a] `https://github.com/conda-forge/miniforge`

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 $z = x + \jmath y$, wobei $x, y \in \mathbb{R}$ und für die imaginäre Einheit $\jmath$ gilt, dass $\jmath^2 = -1$. Wir nutzen hier speziell $\jmath$ in Abgrenzung zu i oder j, da diese oft als Indices oder Laufvariablen auftreten. Bei $z = x + \jmath y$ nennen wir $x = \Re(z)$ den Realteil und respektive $y = \Im(z)$ den Imaginärteil. Komplexe Zahlen lassen sich auch in der Polarform $z = r\exp(\jmath\phi)$ 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 $z = r\exp(\jmath\phi) = x + \jmath y$ komplex konjugierte Zahl ist $z^* = r\exp(-\jmath\phi) = x - \jmath y$.

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

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

die äquivalente Darstellung im komplexen Basisband

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

existiert. Man sagt auch, dass $s_B\exp(\jmath\omega\cdot)$ das analytische Signal zu s darstellt. Dass komplexe Zahlen viele Überaschungen bereithalten sieht man wenn man 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 $s : D \to B$ 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 $D = R^3$, $B = \mathbb{R}$, 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 gegeben 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

$$\dots, 0, 1, 2, \underset{\uparrow}{3}, 2, 1, 0, \dots,$$

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 zunächt 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 hernimmt, 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 in 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 stochastische 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* : $\mathbb{R} \to \mathbb{R}$ mit

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

  Man kann δ als distributionelle Ableitung von u auffassen.

- Die *komplexe Schwingung* $s : \mathbb{R} \to \mathbb{C}$ bei Frequenz $f > 0$ is definiert als $s(t) = \exp(\jmath f t)$ 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

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

- Endliche, diskrete 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 $x : \mathbb{R} \to \mathbb{R}$ 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

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

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

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

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 Mikrofon 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 $s_a : D \to B$ 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 $s : \mathbb{Z} \to Q$ 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 eine *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 einfach 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?
- Wie können wir eine Intuition für den Vorgang der Abtastung entwickeln?

2.1 Frequenz von Signalen

2.1.1 Zeit-Kontinuierliche Sinussignale

Meistens werden wir uns in der Vorlesung mit reell- oder komplexwertigen Zeitsignalen befassen, d.h. wir modellieren unsere Signale als $x_a : \mathbb{R} \to \mathbb{R}$ oder $x_a : \mathbb{R} \to \mathbb{C}$. 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 $1/[s]$ eng mit Zeit $[s]$ verknüpft ist.

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

$$x_a(t) = A\cos(\Omega t + \theta), \tag{2.1.1}$$

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

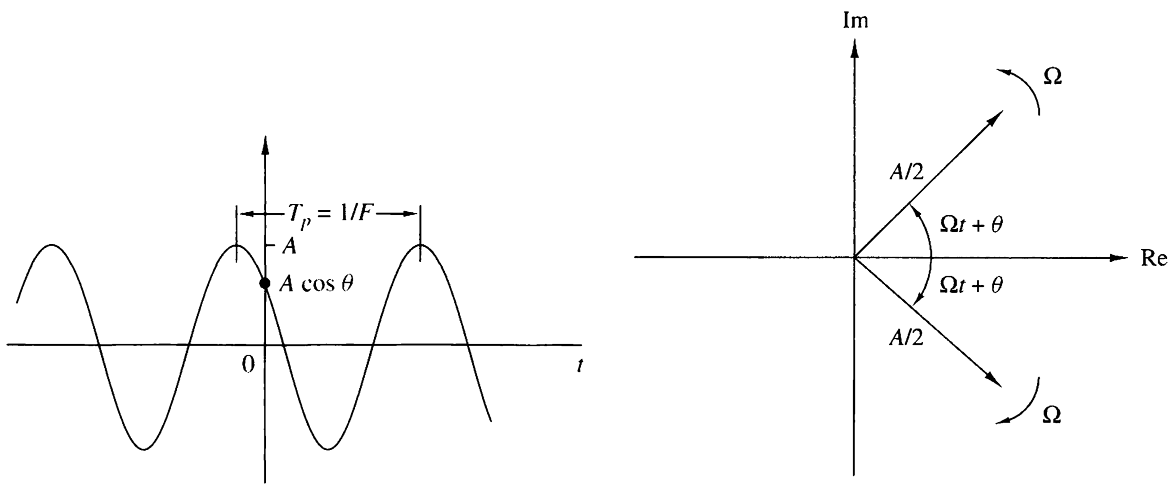


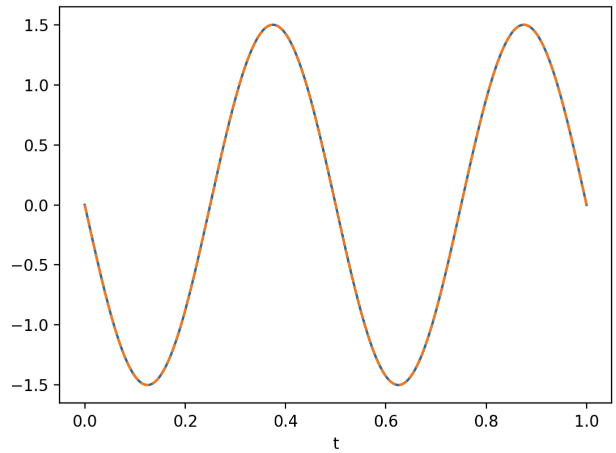
Abbildung 1: $x_a(t) = A\cos(2\pi Ft + \theta)$, Quelle: [1]

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

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

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.

```python
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 Sinussignale

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

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

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

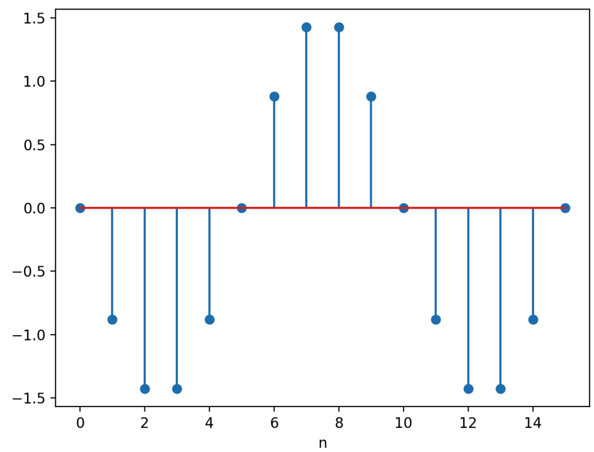
$$\cos(2\pi f n + \theta) = \cos(2\pi f(n+N) + \theta) = \cos(2\pi f n + 2\pi f N + \theta)$$

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).

```python
A = 1.5
f = 0.1
theta = 0.25


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 Sinussignale

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

$$F_k = k \cdot F_0 \quad \text{für} \quad k \in \mathbb{Z} \quad \text{, was} \quad x_k(t) = \exp(\jmath 2\pi F_k t) = \exp(\jmath 2\pi k F_0 t)$$

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

$$x_a = \sum_{k\in\mathbb{Z}} c_k x_k : \mathbb{R} \to \mathbb{C} \quad \text{mit} \quad t \mapsto x_a(t) = \sum_{k\in\mathbb{Z}} c_k x_k(t) = \sum_{k\in\mathbb{Z}} c_k \exp(\jmath 2\pi k F_0 t). \tag{2.2.1}$$

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 Sinussignale

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 $x_k[\cdot] : \mathbb{N} \to \mathbb{C}$ als

$$x_k[n] = \exp(\jmath 2\pi k f_0 n) \quad \text{mit} \quad k \in \mathbb{Z}. \tag{2.3.1}$$

Man sieht nun leicht, dass $f_k = k/N$ immer eine rationale Zahl ist und $x_k[\cdot]$ Periodenlänge k/N hat, falls $k/N \in \mathbb{Z}$. Dies ist in Codeschnipsel 3 dargestellt.

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

$$x_{k+N}[n] = \exp(\jmath 2\pi (k+N) f_0 n) = \exp\left(\jmath 2\pi \frac{k+N}{N} n\right) = \exp\left(\jmath 2\pi \frac{k}{N} n\right) \exp\left(\jmath 2\pi \frac{N}{N} n\right) = x_k[n].$$

Das heißt, dass nur N verschiedene $x_k[\cdot]$ existieren. Normalerweise nimmt man jene $x_k[\cdot]$ für $k = 0, 1, \ldots, N-1$.

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

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

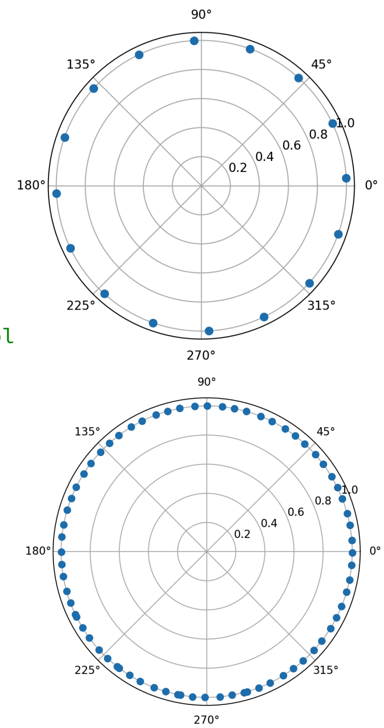
und erhalten so ein Signal mit Werten

$$x[n] = \sum_{k=0}^{N-1} c_k x_k[n] = \sum_{k=0}^{N-1} c_k \exp\left(\jmath 2\pi \frac{k}{N} n\right), \tag{2.3.2}$$

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

Das Signal $x[\cdot]$ selbst ist periodisch mit Periodenlänge N, d.h. das Signal ist durch die Werte $\boldsymbol{x} = [x[n]]_{n=0}^{N-1} \in \mathbb{C}^N$ *vollständig* definiert. Das heißt wir können das Signal $x[\cdot]$ mit dem *endlich-dimensionalen*

```python
A = 1
# f = 1/16
f = 1 / (5 * np.exp(1))
theta = np.pi / 2
# N = 32
N = 72


def harm(n: np.ndarray[int]) -> np.ndarray[compl
    return A * np.exp(
        1j * 2 * np.pi * (f * n + theta))


n = np.arange(N)
x_n = harm(n)
fig, ax = plt.subplots(
    subplot_kw={"projection": "polar"})
ax.plot(np.angle(x_n), np.abs(x_n),
        marker="o", linewidth=0)
print(np.abs(x_n))
plt.show()
```



Codeschnipsel 3: Berechnung und Darstellung von (2.3.1) für $f_0 = 1/16$ und $f_0 = 1/(5e)$, siehe `code/disc_harms_comp.py`

Vektor $\boldsymbol{x} \in \mathbb{C}^N$ identifizieren. Da genauso jedes der $x_k[\cdot]$ auch Periodenlänge N hat, erhalten wir auf die gleiche Weise Vektoren $\boldsymbol{x}_k \in \mathbb{C}^N$. Mit diesen endlich-dimensionalen Vektoren sind wir nun in der Lage die Fourierreihe in (2.3.2) mit „normalen“ Vektoren durch

$$\boldsymbol{x} = c_1 \boldsymbol{x}_1 + c_2 \boldsymbol{x}_2 + \ldots + c_{N-1} \boldsymbol{x}_{N-1} \tag{2.3.3}$$

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

Gleichung (2.3.2) kann auch als Abbildung $M : \mathbb{C}^N \to \mathbb{C}^N$ interpretiert werden, da wir in die rechte Seite von (2.3.2) einfach ein $\boldsymbol{c} \in \mathbb{C}^N$ stecken können und wir erhalten das entsprechende $\boldsymbol{x} = M(\boldsymbol{c})$. Noch weiter ist die Abbildung M sogar linear, da

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

Das heißt, dass es auch eine *Matrix* $\boldsymbol{M} \in \mathbb{C}^{N \times N}$ geben muss, welche uns einfach $\boldsymbol{c}$ in $\boldsymbol{x}$ umtransformiert, indem wir $\boldsymbol{x} = \boldsymbol{M} \cdot \boldsymbol{c}$ als Matrix-Vektor-Produkt berechnen. Wenn wir (2.3.3) genau betrachten sehen wir, dass wir die Matrix $\boldsymbol{M}$ bilden können, indem wir deren k-te Spalte $\boldsymbol{M}_{\cdot,k}$ gleich $\boldsymbol{x}_k$ setzen. Wenn wir noch sicher sein könnten, dass $\boldsymbol{M}$ invertierbar ist, könnten wir sogar aus $\boldsymbol{x}$ via $\boldsymbol{c} = \boldsymbol{M}^{-1} \cdot \boldsymbol{x}$ die Fourierkoeffizienten $\boldsymbol{c}$ direkt aus einem gegebenen N-periodischen Signal $x[\cdot]$ bestimmen.

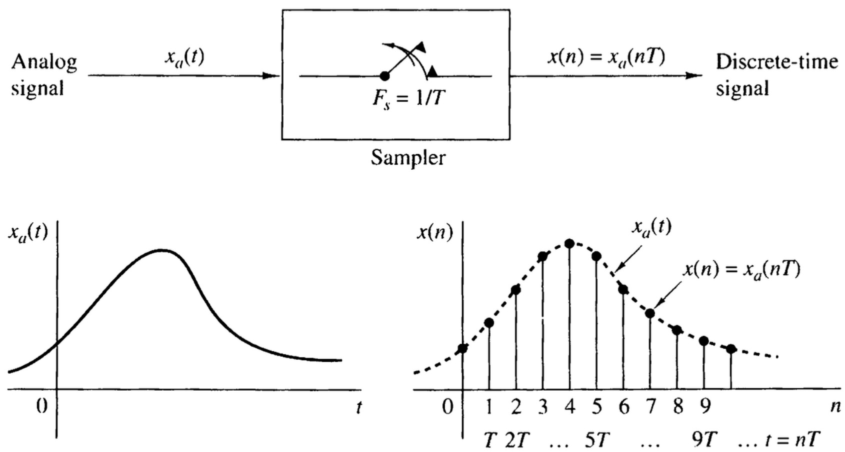


Abbildung 2: Uniforme Abtastung einer Signals. Quelle: [1]

2.4 Abtastung von Sinussignalen

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 $x_a : \mathbb{R} \to \mathbb{R}$ 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

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

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.

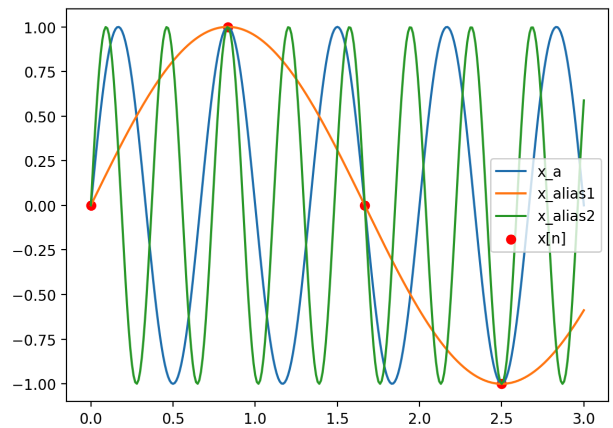
Weiterhin folgt aus $t = n/F_s$, dass es auch einen Zusammenhang zwischen der Frequenz F einer kontinuierlichen Signals und der Frequenz f im Zeit-diskreten geben muss. Um diesen herzuleiten, betrachten wir eine einfache analoge Schwingung, wie in (2.1.1), also

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

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

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

```python
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 4: Visualisierung von $F_k = F + k \cdot F_s$, siehe code/aliasing.py

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 $A\cos(2\pi F_k t + \theta)$ diesselben 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 4 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

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

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 mit einzelnen Frequenzen 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

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

gilt, muss also auch

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

gelten. Deshalb muss schlussendlich $F_s > 2F_{\max}$ gelten. Es ist zu beachten, dass wir diese Überlegungen erst einmal „nur“ für Signale der Form (2.1.1) und (2.1.2) durchgeführt haben. Im nächsten Abschnitt werden wir das Abtasttheorem für beliebige aperiodische Signale herleiten.

2.5 Das Samplingtheorem

Wir beschäftigen uns nun also allgemein mit dem Problem der Abtastung eine beliebigen analogen aperiodischen Signals x_a zur Folge

$$x[n] = x_a(nT).$$

Für die bequeme Analyse von kontinuierlichen Signalen und ihrem Anteil an harmonischen Komponenten, benutzen wir die Fourier Transform (FT) $\mathcal{F}$, welche ein Signal $x : \mathbb{R} \to \mathbb{C}$ transformiert in $x \mapsto \mathcal{F}x = X : \mathbb{R} \to \mathbb{C}$, wobei X definiert durch

$$X(F) = \int_{-\infty}^{+\infty} x(t) \exp(-\jmath 2\pi F t)\, \mathrm{d}t \tag{2.5.1}$$

definiert ist. Wir werden versuchen immer die Kombination von Klein- und Großbuchstaben $x \leftrightarrow X$ als FT-Paar beizubehalten. Die zugehörige inverse FT, also die Abbildung von X auf x ist gegeben durch

$$x(t) = \int_{-\infty}^{+\infty} X(F) \exp(\jmath 2\pi F t)\, \mathrm{d}F. \tag{2.5.2}$$

Wir wollen nun ausarbeiten, wie man diese Art Integral-Transformation rein mathematisch interpretieren kann, um eine alternative Sicht auf solche Art Transformation zu erhalten. Betrachten wir dazu das Signal x wieder als Vektor in einem Vektorraum, in dem ein *Skalarprodukt* definiert ist. Um zu erläutern, was das bringt, starten wir mit etwas Bekanntem. Im endlich-dimensionalen Fall, also wenn $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{C}^N$, dann ist ein mögliches Skalarprodukt $\langle \cdot, \cdot \rangle : \mathbb{C}^N \times \mathbb{C}^N \mathbb{R}$ durch

$$\langle \boldsymbol{x}, \boldsymbol{y} \rangle = \sum_{i=1}^{N} x_i \cdot y_i^* = y^{\mathrm{H}} \cdot x$$

gegeben, wobei der letzte Ausdruck das Matrix-Vektor-Produkt einer $\mathbb{C}^{N\times 1}$ Matrix und einem Vektor in $\mathbb{C}^N$ darstellt. Man kann nun $\langle \boldsymbol{x}, \boldsymbol{y} \rangle$ betrachten, um zu ermitteln, wie „ähnlich" die Vektoren $\boldsymbol{x}$ und $\boldsymbol{y}$ sind. Beispielsweise nennen wir die beiden Vektoren orthogonal/senkrecht, falls $\langle \boldsymbol{x}, \boldsymbol{y} \rangle = 0$. In diesem Fall interpretieren wir dies intuitiv so, dass $\boldsymbol{x}$ und $\boldsymbol{y}$ keine „gemeinsamen" Anteile haben. Beispielsweise, wenn wir einen dritten Vektor $\boldsymbol{z} = \alpha \boldsymbol{x} + \beta \boldsymbol{y}$ betrachten und es gilt, dass $\langle \boldsymbol{x}, \boldsymbol{y} \rangle = 0$, dann schlägt sich dies auch auf das Skalarprodukt von $\langle \boldsymbol{x}, \boldsymbol{z} \rangle$ nieder, da

$$\langle \boldsymbol{x}, \boldsymbol{z} \rangle = \langle \boldsymbol{x}, \alpha \boldsymbol{x} + \beta \boldsymbol{y} \rangle = \alpha \langle \boldsymbol{x}, \boldsymbol{x} \rangle.$$

Beim Bilden von $\langle \boldsymbol{x}, \boldsymbol{z} \rangle$ können wir also einen etwas asymmetrischen Standpunkt einnehmen und uns vorstellen, dass wir ermitteln, wie viele „Anteile" von $\boldsymbol{x}$ in $\boldsymbol{z}$ zu finden sind.

Man sieht, dass dies in gewisser Weise erlaubt mit hochdimensionalen Objekten eine Art Geometrie zu betreiben. Man kann noch weiter gehen und mit einem Skalarprodukt Korrelationen, Winkel und Längen definieren.

Wir wollen nun dieses Konzept auf kontinuierliche Signale erweitern und dann die FT in diesem Licht interpretieren. Man kann zeigen, dass für zwei Funktionen $x, y : \mathbb{R} \to \mathbb{C}$ die Abbildung $(x, y) \mapsto \langle x, y \rangle$ mit

$$\langle x, y \rangle = \int_{-\infty}^{+\infty} x(t) \cdot y^*(t) \, \mathrm{d}t \tag{2.5.3}$$

ein Skalarprodukt definiert. Auch hier halten wir an der Interpretation fest, dass wir damit berechnen, wie ähnlich sich x und y sind. Definieren wir nun die Signale $k_F : \mathbb{R} \to \mathbb{C}$ durch $k_F(t) = \exp(\jmath 2\pi F t)$, dann können wir (2.5.1) umschreiben zu

$$\langle x, k_F \rangle = \int_{-\infty}^{+\infty} x(t) \cdot \exp(\jmath 2\pi F t)^* \, \mathrm{d}t = \int_{-\infty}^{+\infty} x(t) \cdot \exp(-\jmath 2\pi F t) \, \mathrm{d}t = X(f).$$

Dies suggeriert uns also, dass wir die FT als Skalarprodukt von dem zu transformierenden Signal und einem geeignet gewählten Transformationskern interpretieren können. Die Zuordnung von $F \mapsto \langle x, k_F \rangle$ definiert dann im Grunde die Funktion X für alle F.

Genauso erhalten wir für die Folge der Abtastwerte $x[\cdot]$ die Discrete Time Fourier Transform (DTFT) als Skalarprodukt von $x[\cdot]$ und der Folge $k_f[n] = \exp(\jmath 2\pi f n)$, also

$$X(f) = \sum_{n \in \mathbb{Z}} x[n] k_f[n]^* = \sum_{n \in \mathbb{Z}} x[n] exp(-\jmath 2\pi f n).$$

Die inverse DTFT ergibt sich durch

$$x[n] = \int_{-1/2}^{+1/2} X(f) \exp(\jmath 2\pi f n) \, \mathrm{d}f.$$

Wir haben nun alle Werkzeuge zusammen, um genauer zu Analysieren, was bei Abtastung von x_a zu $x[\cdot]$ geschieht. Wir erinnern uns, dass gilt für die Abtastpunkte gilt, dass $t = nT = n/F_s$ gilt. Wir erhalten dann mit $x[n] = x_a(nT)$ und der inversen FT angewandt auf X_a, dass

$$x[n] = x_a(nT) = \int_{-\infty}^{+\infty} X_a(F) \exp(\jmath 2\pi F / F_s) \, \mathrm{d}F$$

gelten muss. Setzen wir nun auch für $x[\cdot]$ die inverse DTFT ein, erhalten wir mit $f = F/F_s$ (siehe Abschnitt 2.4) und einer Variablentransformation, dass

$$\frac{1}{F_s}\int_{-1/2}^{+1/2} X(f)\exp(\jmath 2\pi n F/F_s)\,\mathrm{d}F = \int_{-\infty}^{+\infty} X_a(F)\exp(\jmath 2\pi F/F_s)\,\mathrm{d}F$$

gilt. Wichtig ist an diesem Ausdruck, dass wir nun einen Zusammenhang von X un X_a gefunden haben. Die Integration auf beiden Seiten ist auch bezüglich desselben Integrationskernes $\exp(\jmath 2\pi n/F_s\cdot)$. Das heißt, dass es möglich sein sollte, einen Ausdruck herzuleiten, der X in Abhängigkeit von X_a ausdrückt. Mit ein wenig algebraischem Grindwork findet man

$$X(f) = F_s \sum_{k\in\mathbb{Z}} X_a((f-k)F_s). \tag{2.5.4}$$

Wie ist dies nun zu interpretieren? Im Grunde bestätigt (2.5.4) für beliebige Signale, was wir bereits für Signale mit einzelnen Frequenzen (siehe (2.1.1)) gefunden hatten. Das Spektrum wird durch die Abtastung periodifiziert und zwar entsteht das periodische Spektrum X durch Wiederholung von um $1/F_s$ gestauchte Kopien von X_a mit Abstand $1/F_s$. Man sieht nun auch wieder schön, dass unsere obige Argumentation zum Verhältnis von Samplingfrequenz und maximaler Frequenz im Signal immernoch gilt, da nun gelten muss, dass *alle* belegten Frequenzen von x_a sich im Intervall $[-F_s/2, +F_s/2]$ befinden müssen, da sich sonst die Kopien von X_a in (2.5.4) überlappen. Dies ist äquivalent zur Forderung, dass

$$|X_a(F)| = 0 \quad \text{für} \quad |F| > F_s/2$$

gelten muss. In diesem Fall spricht man davon, dass X_a *bandbegrenzt* ist. Das kleinste $F \in \mathbb{R}$, was obige Ungleichung erfüllt, nennen wir $F_{\max}$.

Wir sind nun beinahe am Ziel angelangt, dass wir das Samplingtheorem formulieren können. Wir benötigen lediglich noch eine Möglichkeit aus den Abtastwerten $x[\cdot]$ das Signal x_a zu rekonstruieren. Wenn kein Aliasing vorliegt, also $F_{\max} < F_s/2$, dann können wir X_a aus X berechnen, indem wir

$$X_a(F) = \begin{cases} \frac{X(f)}{F_s}, |F| \leqslant F_s/2, \\ 0 \quad \text{sonst} \end{cases}$$

setzen. Wir wissen außerdem, dass sich X durch die DTFT durch

$$X(f) = \sum_{n\in\mathbb{Z}} x[n]\exp(-\jmath 2\pi n F/F_s)$$

berechnen lässt. Weiter können wir nun auch x_a durch Integration über einen endlichen Bereich durch

$$x_a(t) = \int_{-F_s/2}^{+F_s/2} X_a(F)\exp(\jmath 2\pi F t)\,\mathrm{d}F.$$

erhalten. Wir setzen nun einfach alles ein und erhalten

$$x_a(t) = \frac{1}{F_s}\int_{-F_s/2}^{+F_s/2} \sum_{n\in\mathbb{Z}} x[n]\exp(-\jmath 2\pi n F/F_s)\exp(\jmath 2\pi F t)\,\mathrm{d}F.$$

Toll, aber wir müssen dies nun noch ein wenig umformen. Zuerst vertauschen wir Integration und Summation, weil wir es dürfen und nutzen $x[n] = x_a(n/F_s)$. Dies ergibt

$$x_a(t) = \frac{1}{F_s} \sum_{n \in \mathbb{Z}} x_a(n/F_s) \int\limits_{-F_s/2}^{+F_s/2} \exp(-\jmath 2\pi nF/F_s) \exp(\jmath 2\pi Ft) \, \mathrm{d}F.$$

Wir sind nicht wirklich schlauer, aber wir wissen aus unserer Erfahrung von Integration von komplexen Funktionen, dass

$$\frac{1}{F_s} \int\limits_{-F_s/2}^{+F_s/2} \exp(-\jmath 2\pi nF/F_s) \exp(\jmath 2\pi Ft) \, \mathrm{d}F = \frac{\sin(\pi F_s(t - n/Fs))}{\pi F_s(t - n/F_s)} = \operatorname{sinc}(F_s(t - n/Fs)).$$

Wir nutzen nun wiederum die Interpretation von Integration eines Produktes von Funktionen als deren Skalarprodukt. Wir bilden also das Skalarprodukt von zwei Funktionen mit Periode F_s, d.h. es genügt das Integral im Intervall $[-F_s/2, +F_s/2]$ zu bilden. Sehen wir genauer hin, bilden wir also das Skalarprodukt $\langle k_t, k_{n/F_s} \rangle$, wobei ähnlich wie oben

$$k_t(F) = \exp(\jmath 2\pi Ft)$$

definiert wurde. Das heißt nun in der Sprache von Skalarprodukten, dass wir die sinc-Funktion erhalten, als

$$\operatorname{sinc}(F_s(t - n/Fs)) = \langle k_t, k_{n/F_s} \rangle.$$

Wenn man es ein wenig allgemeiner Formuliert, kann man sagen, dass die sinc-Funktion nur ein spezieller *Interpolationskern* ist, der sich aus der Definition von k_t ergibt. Wir werden später noch ein weiteres Beispiel für so einen Kern betrachten.

Zusammenfassend für diesen Abschnitt 2 steht nun folgendes Theorem.

Theorem 2.1 (Samplingtheorem)**.** *Gegeben sei ein analoges Signal x_a mit Frequezen 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

$$x_a(t) = \sum_{n \in \mathbb{Z}} x[n] \cdot g_{F_s}(t - nT), \tag{2.5.5}$$

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

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

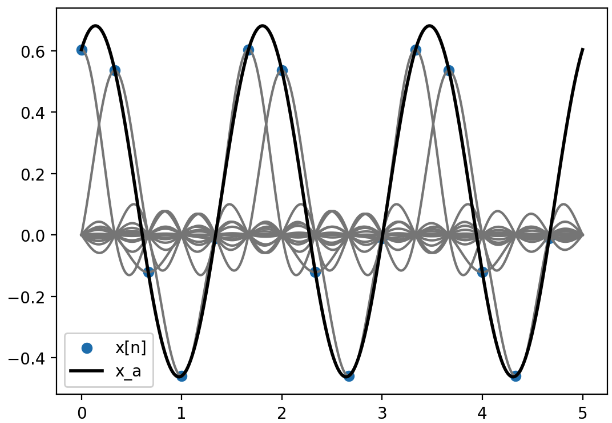
Hinweis: Das Samplingtheorem, wie es in [1] formuliert ist, beinhaltet einen kleinen, aber entscheidenden Fehler. Die Definition des Interpolationskernes g_F ist dort mit

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

gegeben. Doch in diesem Falle würde der Interpolationskern von der Bandbreite des Signals x_a, aber nicht von der Abtastrate F_s abhängen. Die angegebe 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*.

Es ist weiterhin noch zu erwähnen, dass wir genau genommen immer noch nicht wissen, wie wir überprüfen können, welche maximale Frequenz $F_{\max}$ von einem analogen Signal x_a belegt wird. Dies liegt

```
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, F) + 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 5: Berechnung und Darstellung von Theorem 2.1, siehe code/sampling_theorem.py daran, dass wir im Vorhinein – also vor Abtastung – keinen Zugriff auf das abzutastende Signal haben. Das heißt, dass wir auch noch nicht überprüfen können, ob wir das Samplingtheorem eingehalten haben, oder einhalten werden, falls wir Abtasten sollten. Man muss sich also für ein gegebenes System, das einem Signale produziert, die abgetastet werden sollen, auf anderem Weg überlegen, wie man die Bedingung $|X_a(F)| = 0 \quad \text{für} \quad |F| > F_s/2$ überprüfen kann.

Akronyme

ADC Analog-to-Digital Converter. 7

CMOS Complementary Metal Oxide Semiconductor. 7

DTFT Discrete Time Fourier Transform. 16, 17

FT Fourier Transform. 15, 16

LTI Linear Time-Invariant. 6

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, 18).