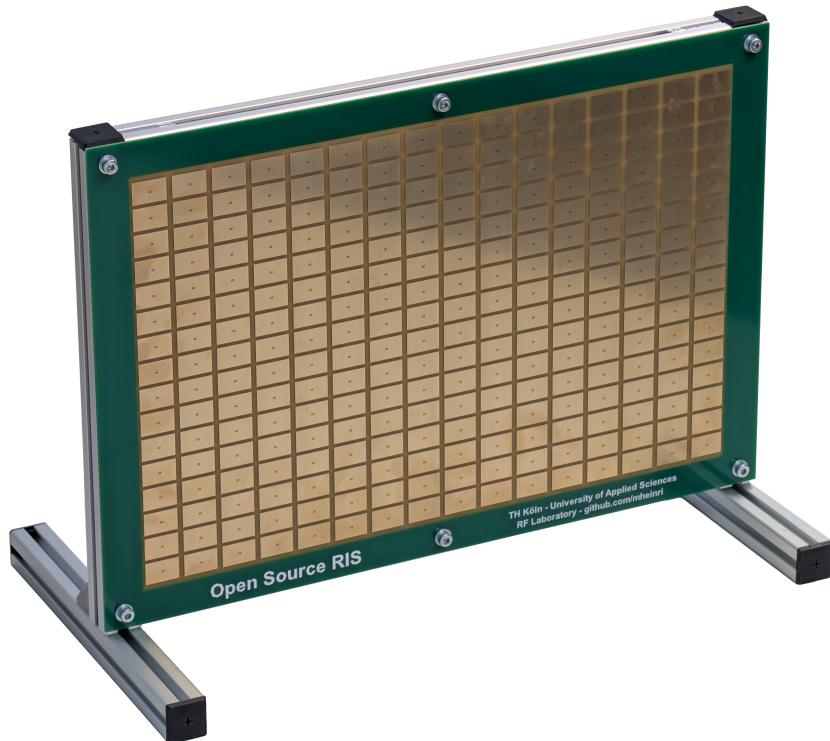


Nachrichtentechnisches Praktikum

Versuch: Rekonfigurierbare Intelligente Oberflächen

Betreuer: Malte Eggers

Stand: 12.11.2025



Inhaltsverzeichnis

1 Einleitung	4
1.1 Was ist eine RIS?	4
1.2 Welche Vorteile hat eine RIS?	4
1.3 Welche Nachteile hat eine RIS?	4
2 Grundlagen	5
2.1 Funksignal	5
2.2 Dezibel	5
2.3 Signal-Rausch-Verhältnis	6
2.4 Mehrwegeausbreitung	6
2.4.1 Amplitude	7
2.4.2 Phase	8
2.4.3 Verzögerung	9
2.4.4 Interferenz	9
3 RIS Phasenmuster	11
3.1 Nahfeld	12
3.2 Fernfeld	13
3.3 Quantisierung	15
4 Software	16
4.1 Python Bibliothek	16
4.2 Grafische Benutzeroberfläche	17
4.2.1 Muster Generator	17
4.2.2 SNR Plot	18
4.2.3 Python Schnittstelle	19
5 Versuchsaufbau	20
5.1 Sender/Empfänger	21
5.1.1 Sender	21
5.1.2 Empfänger	22
5.2 Kurzfragen	23
6 Praktische Aufgaben	23
6.1 Konfiguration anhand bekannter Positionen	23
6.1.1 Versuchsaufbau vermessen	24
6.1.2 Phasenmuster erstellen	24
6.1.3 Einfluss des Musters Messen	24
6.2 Intelligenter vs normaler Reflektor	25
6.3 Mustersuche	26
6.3.1 Simulierte Abkühlung	27

Abbildungsverzeichnis

1 Schwingungen mit unterschiedlicher Amplitude im Zeitbereich	7
2 Schwingungen mit unterschiedlicher Amplitude im Frequenzbereich	7
3 Schwingungen mit unterschiedlicher Phase im Zeitbereich	8

4	Schwingungen mit unterschiedlicher Phase im Frequenzbereich	8
5	Betrag der Summe zweier Phasenterme $e^{j\varphi_0}$ und $e^{j\varphi_1}$	10
6	Schematische Darstellung des RIS im Nahfeld	12
7	Interferenz durch ein Phasenmuster im Nahfeld	13
8	Schematische Darstellung des RIS im Fernfeld	14
9	Interferenz durch ein Phasenmuster im Fernfeld	14
10	Veranschaulichung der Quantisierung der Phase	15
11	Vergleich der Interferenz durch ein Phasenmuster im Nahfeld vor und nach der Quantisierung	15
12	„Control Panel“ der grafischen Benutzeroberfläche	17
13	„Pattern Generator“ der grafischen Benutzeroberfläche	18
14	„SNR Plot“ der grafischen Benutzeroberfläche	18
15	Foto des Versuchsaufbaus	20
16	Foto des für den Versuch verwendeten Empfängers	21
17	Benutzeroberfläche des Senders	21
18	Benutzeroberfläche des Empfängers	22

Tabellenverzeichnis

1	Beispielwerte für die Umrechnung von mW zu dBm	5
2	Beispiel für das Rechnen mit Linearen und Logarithmischen Größen	5

Quellcodeverzeichniss

1	Verbindungsauflbau zu einer RIS	16
2	Erstellung und Übertragung eines einfachen Musters zu einer RIS	16
3	Verwendung der grafischen Benutzeroberfläche in Python	19
4	Starten des Senders	21
5	Starten des Empfängers	22
6	Programmcode zur Mustersuche	26

1 Einleitung

Mobilfunk wird heute in nahezu allen Lebensbereichen genutzt. Die reale Umgebung bietet dafür jedoch selten ideale Ausbreitungsbedingungen. Wände in Gebäuden oder Häuser entlang von Straßen, schatten immer wieder die direkte Sicht von Mobilstation zu Basisstation ab. Infolgedessen kann das Signal-zu-Rausch-Verhältnis zu gering ausfallen, um die erforderlichen Datenraten zuverlässig zu erreichen. Die Verwendung von gut platzierten rekonfigurierbaren intelligenten Oberflächen (RIS) ermöglicht es die Funkwellen in Richtung ansonsten abgeschatteter Teilnehmer zu reflektieren. In diesem Praktikumsversuch soll dir die Verwendung von RIS anhand einiger praktischer Experimente nähergebracht werden.

1.1 Was ist eine RIS?

Eine Rekonfigurierbare Intelligente Oberfläche (RIS) setzt sich aus einem dicht beabstandeten Array an Patch Antennen zusammen. Jede Antenne ist mit einer steuerbaren elektronischen Schaltung verbunden, welche ein empfangenes Signal Phasenverschoben reflektieren kann. Die Phasenverschiebung kann hierbei für jede der Antennen individuell gewählt und während des Betriebs angepasst werden. Durch die gezielte Wahl der Phasenverschiebungen lässt sich das Reflexionsverhalten formen (passives Beamforming). So können unterschiedlicher Oberflächen simuliert und Signale in zuvor festgelegte Richtungen fokussiert werden.

1.2 Welche Vorteile hat eine RIS?

Im Alltag ist die direkte Sichtverbindung (Line-of-Sight, LoS) zwischen Sender und Empfänger häufig verdeckt. Zwar ermöglichen z. B. Reflexionen an Wänden meist dennoch eine Verbindung, allerdings verschlechtert sich der Signal-Rausch-Abstand (SNR) und damit die maximal erreichbare Datenrate. Eine rekonfigurierbare intelligente Oberfläche (RIS) kann im Gegensatz zu einer normalen Wand das Signal kontrolliert reflektieren. Hierdurch kann ein besseres SNR erreicht werden. Gegenüber der Installation zusätzlicher Basisstationen oder Repeater bietet eine RIS einige Vorteile:

- geringere Hardware- und Installationskosten
- nahezu passiver Betrieb mit sehr niedrigem Stromverbrauch

Dies ist besonders vorteilhaft bei sehr hohen Frequenzen, wo Abschattungen vermehrt vorkommen.

1.3 Welche Nachteile hat eine RIS?

Um mithilfe einer RIS das Signal-Rausch-Verhältnis (SNR) zu verbessern, werden die relativen Positionen oder Ausrichtungen von Basisstation und Mobilstation zur RIS benötigt. Das erweist sich in der Praxis als anspruchsvoll, da sich die Position der Mobilstation oft und teilweise schnell ändert. Es ist daher nötig die Position der Mobilstation fortlaufend zu schätzen und die RIS neu zu konfigurieren. Zudem kann bei einer Fehlkonfiguration des RIS das SNR, durch destruktive Interferenz, weiter verschlechtert werden.

2 Grundlagen

2.1 Funksignal

Bei den in diesem Versuch betrachteten Signalen handelt es sich um Funksignale. Unter einem Funksignal verstehen wir eine elektromagnetische Welle, mit deren Hilfe Informationen drahtlos von einem Sender zu einem Empfänger übertragen werden. Im Sender wird die zu übertragende Information in ein elektrisches Signal gewandelt, auf einen hochfrequenten Träger aufmoduliert und über eine Antenne abgestrahlt. Die in diesem Versuch betrachteten Frequenzen liegen im Frequenzbereich um 5,8 GHz, der für WLAN verwendet wird. Der Empfänger übernimmt die Demodulation des ankommenden Signals und Rückgewinnung der ursprünglichen Informationen. Wenn im Folgenden von der Position von Sender oder Empfänger die Rede ist, ist jeweils die Position der zugehörigen Antenne gemeint.

2.2 Dezibel

Das Dezibel (dB) ist eine Maßeinheit, die verwendet wird, um das Verhältnis zweier logarithmischer Größen darzustellen. Es handelt sich bei Dezibel nicht um eine eigenständige physikalische Einheit, sondern immer um die Beschreibung eines Verhältnisses. Die Umrechnung eines Verhältnisses in Dezibel wird in Gleichung 1 (für Leistungsgrößen) beschrieben.

$$P_{\text{dB}} = 10 \cdot \log_{10} \left(\frac{P_2}{P_1} \right) \Leftrightarrow \frac{P_2}{P_1} = 10^{\frac{P_{\text{dB}}}{10}} \quad (1)$$

Wenn nicht explizit angegeben ist, worauf sich ein dB-Wert bezieht, ist nur das Verhältnis gemeint. Absolute Pegel entstehen erst durch Angabe eines Bezugswertes. Bei der Angabe von Sende- und Empfangsleistung ist es üblich diese in dBm anzugeben. Dies beschreibt das Verhältnis der Leistung gegenüber 1 mW.

Tabelle 1: Beispielwerte für die Umrechnung von mW zu dBm

W	mW	dBm
0,001	1	0
0,01	10	10
0,1	100	20
1	1000	30

Tabelle 2: Beispiel für das Rechnen mit Linearen und Logarithmischen Größen

Linear	Logarithmisch
$\div 10$	-10 dB
$\div 2$	$\approx -3 \text{ dB}$
$\cdot 1$	$+0 \text{ dB}$
$\cdot 2$	$\approx +3 \text{ dB}$
$\cdot 10$	$+10 \text{ dB}$

Durch die Nutzung von Dezibel kann die Multiplikation als Addition und die Division als Subtraktion ausgedrückt werden (Gleichung 2). Es ergeben sich die in Tabelle 2 aufgeführten Zusammenhänge.

$$1 \text{ mW} \div 10 = 0,1 \text{ mW} \Leftrightarrow 0 \text{ dBm} - 10 \text{ dB} = -10 \text{ dBm} \quad (2)$$

Eine alte Glühbirne verbraucht 60 Watt, ein neues äquivalentes LED Leuchtmittel verbraucht hingegen nur 7 Watt. Wie lässt sich dieser Unterschied in dB ausdrücken? Wie lassen sich die absoluten Watt Werte in dBm ausdrücken?

2.3 Signal-Rausch-Verhältnis

Das Signal-Rausch-Verhältnis (Signal-to-noise ratio, SNR) beschreibt das Verhältnis der Leistung des gewünschten Nutzsignals (P_{Signal}) zur Leistung des unerwünschten Rauschens (P_{Rauschen}). In diesem Skript verwenden wir „Rauschen“ als Sammelbegriff für unerwünschte Signalanteile. In Gleichung 3 ist die Berechnung des SNR dargestellt, üblicherweise wird es in Dezibel angegeben.

$$\text{SNR} = \frac{P_{\text{Signal}}}{P_{\text{Rauschen}}} \Leftrightarrow \text{SNR}_{\text{dB}} = 10 \cdot \log_{10} \left(\frac{P_{\text{Signal}}}{P_{\text{Rauschen}}} \right) \quad (3)$$

Ein möglichst großes SNR ist entscheidend für eine zuverlässige, fehlerarme Kommunikation. Anhand der Shannon-Grenze lässt sich die theoretisch maximal erreichbare Datenrate (C) in Bit für eine gegebene Bandbreite (B) in Hertz und ein gegebenes SNR berechnen (Gleichung 4).

$$C = B \cdot \log_2 \left(1 + \frac{P_{\text{Signal}}}{P_{\text{Rauschen}}} \right) \quad (4)$$

2.4 Mehrwegeausbreitung

Unter Mehrwegeausbreitung versteht man das Eintreffen eines gesendeten Funksignals über mehrere Ausbreitungswägen am Empfänger. Dabei gelangt ein gesendetes Signal nicht nur auf direktem Weg zum Empfänger, sondern auch über Reflexionen, Beugungen und Streuungen. Der direkte Pfad wird hierbei auch Sichtverbindung oder line-of-sight (LOS) genannt. Das empfangene Signal setzt sich aus einer Überlagerung vieler Teilkomponenten zusammen, die jeweils unterschiedliche Amplituden und Laufzeiten besitzen.

$$y(t) = \sum_{i=1}^N A_i x(t - \tau_i) \quad (5)$$

In Gleichung 5 wird das durch Mehrwegeausbreitung entstehende Empfangssignal $y(t)$, für einen statischen Kanal, mathematisch beschrieben.

- N : Anzahl der Pfade
- A_i : Amplitude des i -ten Pfads
- $x(t)$: Signal am Sender
- t : Zeit in Sekunden
- τ_i : Verzögerung des i -ten Pfads

2.4.1 Amplitude

Die Amplitude A beschreibt in diesem Zusammenhang die Intensität eines Signals. Eine große Amplitude ist daher gleichbedeutend mit einer hohen Intensität.

$$\begin{aligned}
 y(t) &= A \cdot \cos(\omega t) \\
 \omega &= 2\pi f \\
 f &= 2 \text{ Hz}
 \end{aligned} \tag{6}$$

Wir betrachten im Folgenden die in Gleichung 6 beschriebene Kosinus Schwingung.

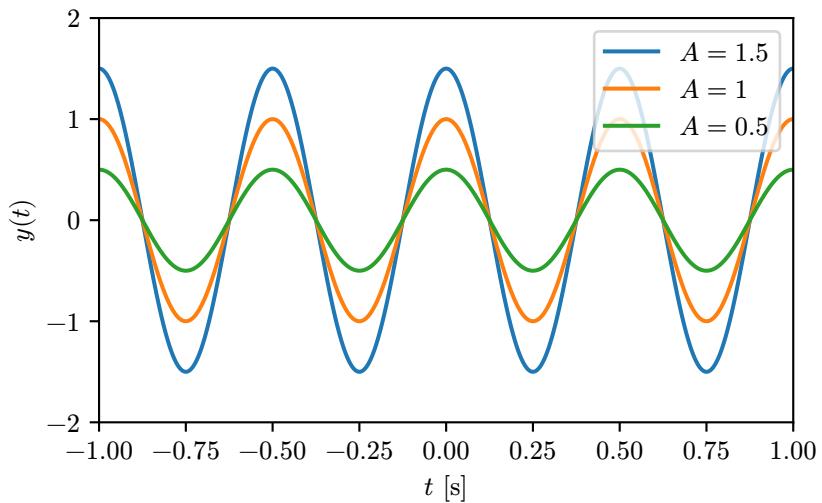


Abbildung 1: Schwingungen mit unterschiedlicher Amplitude im Zeitbereich

In Abbildung 1 kann der Einfluss unterschiedlicher Amplitudenwerte im Zeitbereich beobachtet werden.

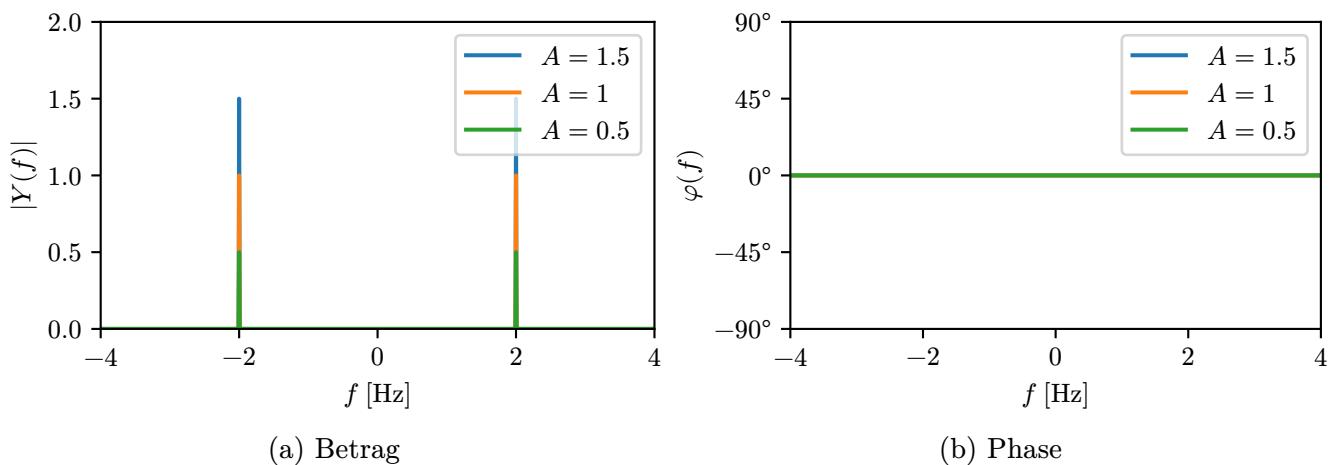


Abbildung 2: Schwingungen mit unterschiedlicher Amplitude im Frequenzbereich

In Abbildung 2 kann der Einfluss unterschiedlicher Amplitudenwerte im Frequenzbereich beobachtet werden. Es ist zu erkennen, dass eine Änderung der Amplitude nur den Betrag des Frequenzbereichs beeinflusst. Die Phase wird durch eine Änderung der Amplitude nicht beeinflusst.

2.4.2 Phase

Die Phase (φ) beschreibt die zeitliche oder räumliche Verschiebung einer Schwingung. Die Verschiebung bezieht sich hierbei auf einen zuvor definierten Referenzpunkt. In der Mehrwegeausbreitung ist die Phase von großer Bedeutung, weil eintreffende Signale derselben Frequenz unterschiedliche Phasen aufweisen können, wenn sie verschiedene Ausbreitungswägen durchlaufen haben. Durch Überlagerung (Superposition) dieser multiplen Signalpfade kann es zu konstruktiven oder destruktiven Interferenzen kommen. Hierauf wird in Unterabschnitt 2.4.4 weiter eingegangen.

$$\begin{aligned}
 y(t) &= \cos(\omega t + \varphi) \\
 \omega &= 2\pi f \\
 f &= 2 \text{ Hz}
 \end{aligned} \tag{7}$$

Wir betrachten im Folgenden die in Gleichung 7 beschriebene Kosinus Schwingung.

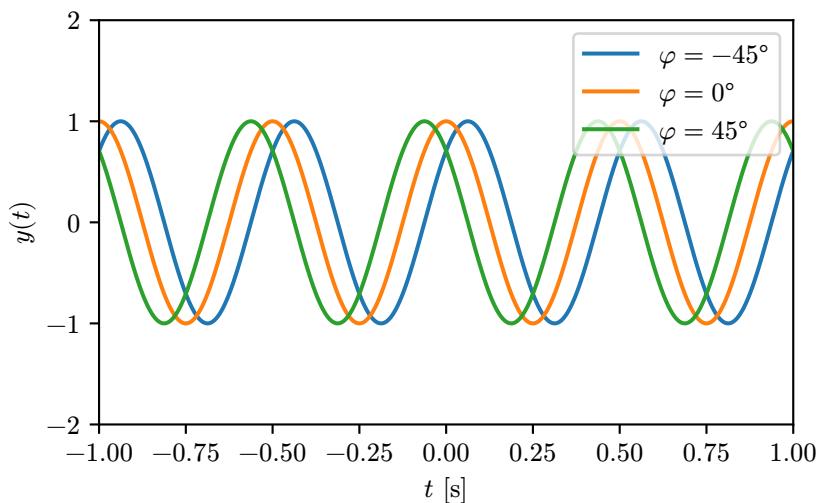


Abbildung 3: Schwingungen mit unterschiedlicher Phase im Zeitbereich

In Abbildung 3 kann der Einfluss unterschiedlicher Phasenwerte im Zeitbereich beobachtet werden.

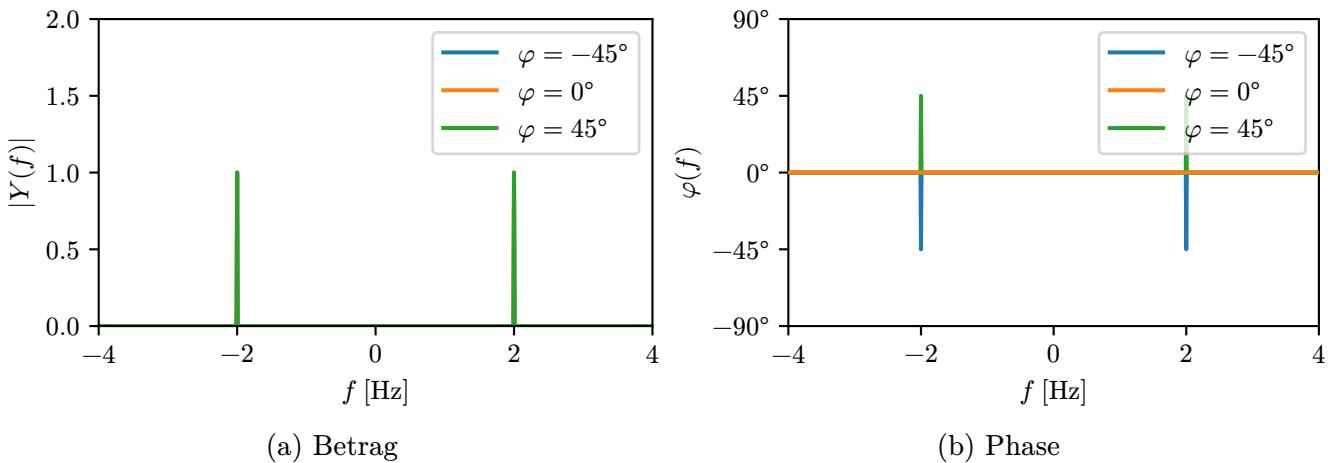


Abbildung 4: Schwingungen mit unterschiedlicher Phase im Frequenzbereich

In Abbildung 4 kann der Einfluss unterschiedlicher Phasenwerte im Frequenzbereich beobachtet

werden. Es ist zu erkennen, dass eine Änderung der Phase den Betrag des Frequenzbereichs nicht beeinflusst.

2.4.3 Verzögerung

Die Verzögerung τ beschreibt, wie lange ein Signal benötigt, um vom Sender zum Empfänger zu gelangen. Bei Mehrwegeausbreitung erreicht das Signal den Empfänger über unterschiedliche Pfade zu unterschiedlichen Zeitpunkten. Hierbei spricht man auch von Laufzeitverzögerung. Wir können diese Verzögerung jedoch auch als Phasenverschiebung darstellen. Hierdurch kommt es bei der Überlagerung mehrerer Signalpfade wiederum zu Interferenz.

$$\begin{aligned}\varphi(\tau) &= \omega \cdot \tau \\ \omega &= 2\pi f\end{aligned}\tag{8}$$

Gleichung 8 beschreibt den Zusammenhang zwischen Phase und Verzögerung. Die Phase wird relativ zur Wellenlänge angegeben. Es ist daher zu beachten, dass dieselbe Verzögerung bei unterschiedlichen Frequenzen eine unterschiedliche Phasenänderung mit sich zieht.

Distanz: Die elektromagnetische Welle eines Funksignals breitet sich im Vakuum mit Lichtgeschwindigkeit ($c = 299\,792\,458 \frac{\text{m}}{\text{s}}$) aus. Die Ausbreitungsgeschwindigkeit weicht je nach Temperatur und Zusammensetzung der Raumluft von der Geschwindigkeit im Vakuum ab. Da die Abweichungen jedoch sehr gering sind, ist es für uns ausreichend sie zu vernachlässigen. Hierdurch ergibt sich der in Gleichung 9 beschriebene Zusammenhang zwischen Distanz (d) und Verzögerung (τ).

$$\tau = \frac{d}{c}\tag{9}$$

2.4.4 Interferenz

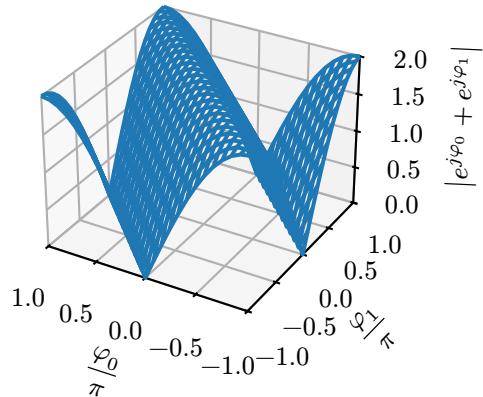
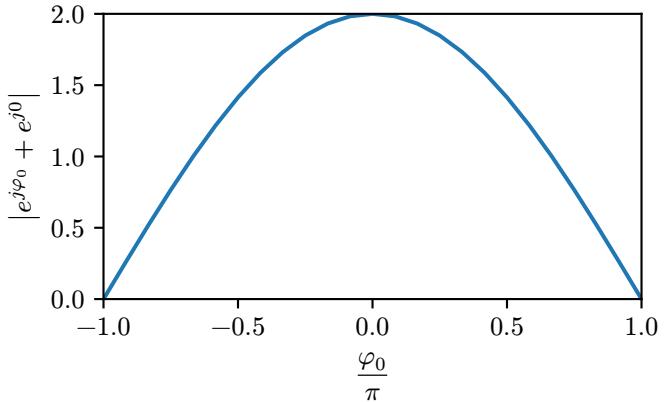
Wie in Gleichung 5 beschrieben, summieren sich die Anteile der Mehrwegeausbreitung am Empfänger. Betrachten wir zwei der sich überlagernden Pfade, erhalten wir den in Gleichung 10 beschriebenen Zusammenhang.

$$\begin{aligned}y_i(t) &= A_i \cdot x(t - \tau_i) \quad \circ \text{---} \bullet \quad Y_i(\omega) = A_i \cdot X(\omega) \cdot e^{j \cdot \omega \cdot \tau_i} \\ \Rightarrow Y(\omega) &= Y_0(\omega) + Y_1(\omega) \\ &= X(\omega) \cdot (A_0 \cdot e^{j \cdot \omega \cdot \tau_0} + A_1 \cdot e^{j \cdot \omega \cdot \tau_1}) \\ &= X(\omega) \cdot (A_0 \cdot e^{j \cdot \varphi_0} + A_1 \cdot e^{j \cdot \varphi_1})\end{aligned}\tag{10}$$

Für dieses Beispiel nehmen wir an, dass für die Amplitude der überlagernden Pfade $A_0 = A_1 = 1$ gilt (Gleichung 11).

$$A_0 = A_1 = 1 \Rightarrow Y(\omega) = X(\omega) \cdot (e^{j \cdot \varphi_0} + e^{j \cdot \varphi_1})\tag{11}$$

Betrachten wir nur die beiden Phasenterme erhalten wir die in Abbildung 5a gezeigte phasenabhängige Interferenz.

(a) $\varphi_0, \varphi_1 \in [-\pi; \pi]$ (b) $\varphi_0 \in [-\pi; \pi]$ und $\varphi_1 = 0$ Abbildung 5: Betrag der Summe zweier Phasenterme $e^{j\varphi_0}$ und $e^{j\varphi_1}$.

In Abbildung 5 können wir erkennen, dass die Überlagerung zweier Schwingungen mit unterschiedlicher Phase zu konstruktiver oder destruktiver Interferenz führt. Im Bestfall (gleiche Phase) verdoppelt sich die Amplitude, während im schlechtesten Fall ($\Delta\varphi = \pi$) eine vollständige Aufhebung des Signals stattfindet.

Zum Zeitpunkt $t_0 = 0$ s kommt am Empfänger eine 1 GHz ($a(t) = \cos(2\pi 10^9 t)$) Schwingung an. Mit einer Zeitdifferenz von 0,25 ns, kommt eine ein ansonsten Identisches Signal am Empfänger an ($b(t) = a(t - 0,25 \text{ ns})$). Es kommt zu einer Überlagerung der beiden Signale. Welche Phasenverschiebung haben die beiden Signale zueinander? Welche Amplitude hat die Summe beider Signale?

3 RIS Phasenmuster

Die in Unterabschnitt 2.4.4 beschriebene Interferenz können wir uns durch Verwendung eines RIS zu Nutzen machen. Durch die korrekte Konfiguration der vom RIS verursachten Phasenverschiebungen, können wir an beliebig wählbaren Orten im Raum sowohl konstruktive als auch destruktive Interferenz erzeugen.

$$\begin{aligned} M &= 16 \\ N &= 16 \\ w &= 0,320 \text{ m} \\ h &= 0,212 \text{ m} \end{aligned} \tag{12}$$

Im Folgenden wird angenommen, dass unser RIS den in Gleichung 12 angegebenen Dimensionen entspricht. Wir verwenden also ein w breites und h hohes RIS, mit M Spalten und N Reihen an Patch Antennen.

$$\begin{aligned} p_{\text{width}} &= \frac{w}{M} \\ p_{\text{height}} &= \frac{h}{N} \\ \vec{p}_{m,n} &= \begin{pmatrix} n \cdot p_{\text{width}} + \frac{p_{\text{width}}}{2} - \frac{w}{2} \\ m \cdot p_{\text{height}} + \frac{p_{\text{height}}}{2} - \frac{h}{2} \\ 0 \end{pmatrix} \end{aligned} \tag{13}$$

Hieraus lässt sich, wie in Gleichung 13 beschrieben, die relative Position jeder Patch Antenne zum Mittelpunkt des RIS bestimmen. Die Indizes m und n geben hierbei die Spalte und Reihe der Patch Antenne an.

Für jede Patch Antenne muss nun die Phasenverschiebung vom Sender zum RIS $\varphi_{\text{tx},m,n}$ und vom RIS zum Empfänger $\varphi_{\text{rx},m,n}$ berechnet werden. In Unterabschnitt 3.1 und Unterabschnitt 3.2 wird auf zwei Möglichkeiten der Berechnung eingegangen.

$$\varphi_{m,n} = -(\varphi_{\text{tx},m,n} + \varphi_{\text{rx},m,n}) \tag{14}$$

Zur Erzeugung konstruktiver Interferenz am Empfänger müssen die Phasenverschiebungen der einzelnen Pfade durch das RIS ausgeglichen werden. Im idealen Fall würde das in Gleichung 14 berechnete Muster verwendet werden. Dies würde jedoch voraussetzen, dass eine beliebige Phasenverschiebung an jeder Patch Antenne konfiguriert werden kann. Im Fall des für diesen Versuch verwendeten RIS wird nur eine Phasenverschiebung von 0° oder 180° unterstützt. Daher müssen die berechneten Phasenverschiebungen vor der Verwendung quantisiert werden. Hierauf wird in Unterabschnitt 3.3 eingegangen.

3.1 Nahfeld

Kennen wir die Position \vec{x} , des Senders oder Empfängers so können wir die Phasenverschiebungen der einzelnen Antennen des RIS für genau diese Position berechnen. Dies ist besonders nützlich, wenn sich die Position sehr nahe am RIS befindet.

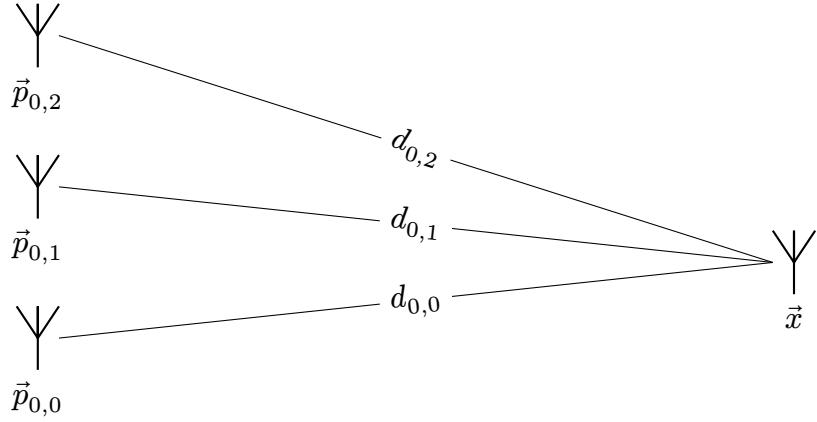


Abbildung 6: Schematische Darstellung des RIS im Nahfeld

In Abbildung 6 werden die Antennen des RIS durch ihre Positionsvektoren $\vec{p}_{m,n}$ dargestellt. Die Distanz zwischen $\vec{p}_{m,n}$ und \vec{x} lässt sich, wie in Gleichung 15 beschrieben, durch die Euklidische Norm berechnen.

$$d_{m,n}(\vec{x}) = \left| \vec{p}_{m,n} - \vec{x} \right|_2 \quad (15)$$

Wie in Unterunterabschnitt 2.4.3 beschrieben, erhalten wir durch Division durch die Lichtgeschwindigkeit, die Laufzeitverzögerung (Gleichung 16).

$$\tau_{m,n}(\vec{x}) = \frac{d_{m,n}(\vec{x})}{c} \quad (16)$$

Weiter können wir die Phase aus der Verzögerung berechnen (Gleichung 17). Alternativ können wir die Rechnung auch als Multiplikation von der Wellenzahl (k) und der Distanz ausdrücken.

$$\begin{aligned} \varphi_{m,n}(\vec{x}) &= 2\pi f \cdot \tau_{m,n}(\vec{x}) \\ &= k \cdot d_{m,n}(\vec{x}) \\ k &= \frac{2 \cdot \pi \cdot f}{c} \end{aligned} \quad (17)$$

Um die Phasenverschiebung mithilfe des RIS auszugleichen, kann $-\varphi_{m,n}(\vec{x})$ als Muster verwendet werden. Hierdurch kommt es an der Position \vec{x} zu konstruktiver Interferenz zwischen allen Pfaden über die Antennen der RIS.

Warum verwenden wir $-\varphi_{m,n}(\vec{x})$ statt $\varphi_{m,n}(\vec{x})$ als Muster?

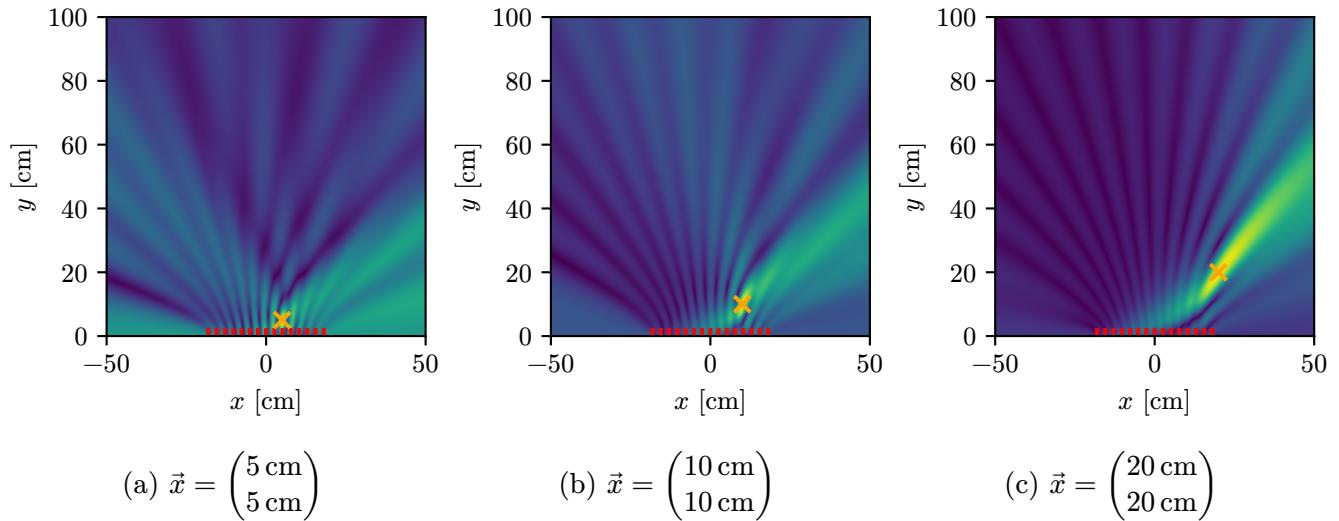


Abbildung 7: Interferenz durch ein Phasenmuster im Nahfeld

In Abbildung 7 ist zu erkennen welche Auswirkung ein solches Muster auf die Interferenz an verschiedenen Punkten im Raum hat. Bereiche, an denen es zu starker konstruktiver Interferenz kommt, sind gelb gefärbt. Zur besseren Darstellung wurde hierbei eine Reihe an Antennen (rote Striche), statt einer zweidimensionalen Anordnung angenommen. Es ist zu erkennen, dass sich bei weiter vom RIS entfernten \vec{x} (gelbes Kreuz) ein Strahl bildet. In Unterabschnitt 3.2 wird beschrieben wie man sich dieses Fernfeldverhalten zu Nutze machen kann.

3.2 Fernfeld

Befindet sich die Position \vec{x} , des Senders oder Empfängers weit vom RIS entfernt, so strebt das in Abbildung 6 gezeigte Interferenzmuster gegen einen Strahl in Richtung \vec{x} . Nehmen wir an \vec{x} befindet sich unendlich weit vom RIS entfernt, dann verlaufen alle Pfade von $\vec{p}_{m,n}$ zu \vec{x} parallel zueinander. Die Differenz der Distanz der einzelnen Pfade ist nun nur noch von der Richtung und nicht der exakten Position von \vec{x} relativ zum RIS abhängig. Diese Richtung kann wie in Gleichung 18 als Normalenvektor dargestellt werden.

$$\vec{n} = \begin{pmatrix} \sin \phi \cdot \cos \theta \\ \sin \phi \cdot \sin \theta \\ \cos \phi \end{pmatrix} \quad (18)$$

- ϕ : Azimut in rad
- θ : Elevation in rad

Der Vektor \vec{n} , ist so definiert, das er von der Mitte des RIS in Richtung \vec{x} zeigt.

$$d_{m,n}(\vec{n}) = \langle \vec{p}_{m,n}, \vec{n} \rangle \quad (19)$$

Es kann nun die Differenz der Distanz zwischen \vec{x} zur Mitte des RIS im Vergleich zu einer Antenne berechnet werden. Hierfür kann, wie in Gleichung 19 beschrieben, das Skalarprodukt verwendet werden. Die Verzögerung und Phase lassen sich wie zuvor in Gleichung 16 und Gleichung 17 beschrieben berechnen.

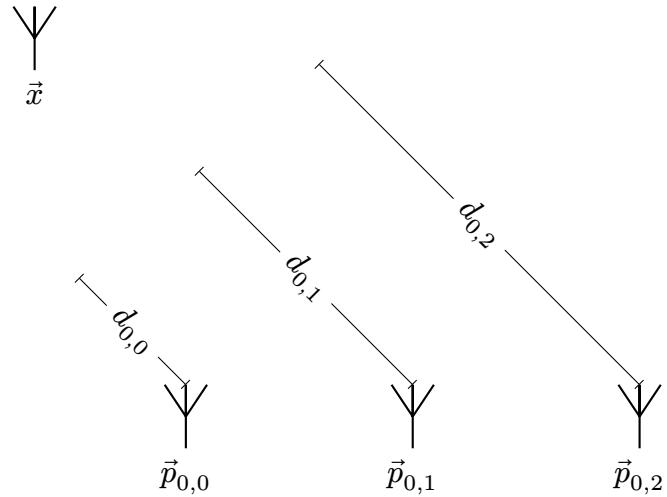


Abbildung 8: Schematische Darstellung des RIS im Fernfeld

Um die Phasenverschiebung mithilfe des RIS auszugleichen, kann $-\varphi_{m,n}(\vec{n})$ als Muster verwendet werden. In Abbildung 9 ist zu erkennen welche Auswirkung ein solches Muster auf die Interferenz an verschiedenen Punkten im Raum hat. Wie zuvor, sind Bereiche, an denen es zu starker konstruktiver Interferenz kommt, gelb gefärbt. Zur besseren Darstellung wurde erneut eine Reihe an Antennen (rote Striche), statt einer zweidimensionalen Anordnung angenommen. Es ist zu erkennen, dass sich ein Strahl in Richtung des eingestellten \vec{n} (gelber Pfeil) bildet.

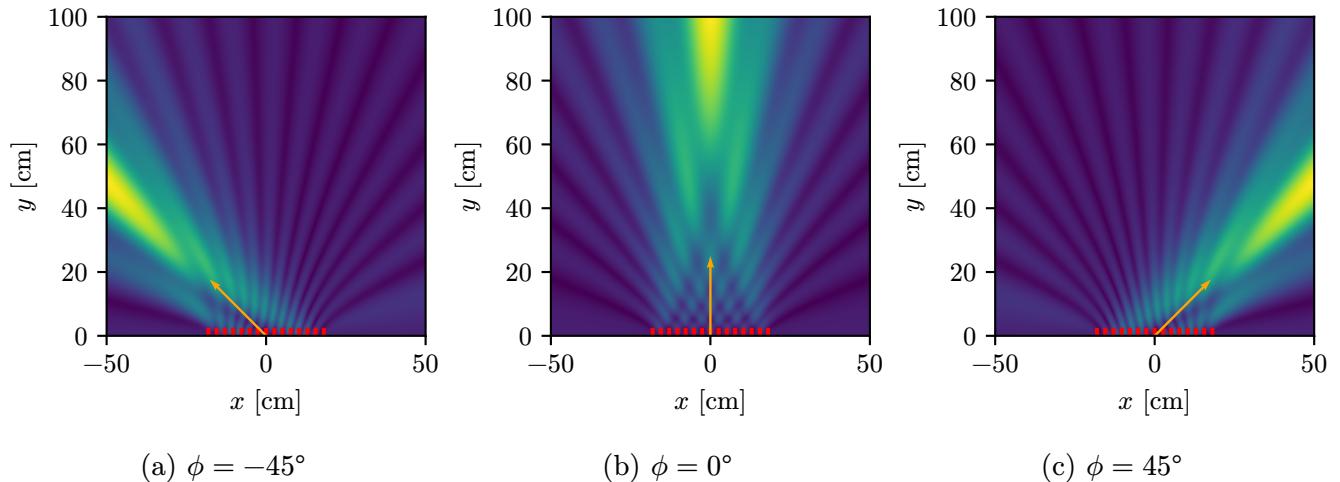


Abbildung 9: Interferenz durch ein Phasenmuster im Fernfeld

3.3 Quantisierung

Das Verwendete RIS ist nicht in der Lage beliebige Phasenverschiebungen zu erzeugen. Es werden nur 0° und 180° Phasenverschiebung unterstützt. Daher ist es nötig das berechnete Phasenmuster auf diese Werte zu Quantisieren (Gleichung 20).

$$\varphi_{m,n}^{\text{quant}} = \pi \cdot \lfloor \frac{\varphi_{m,n} \bmod \pi}{\pi} \rfloor \quad (20)$$

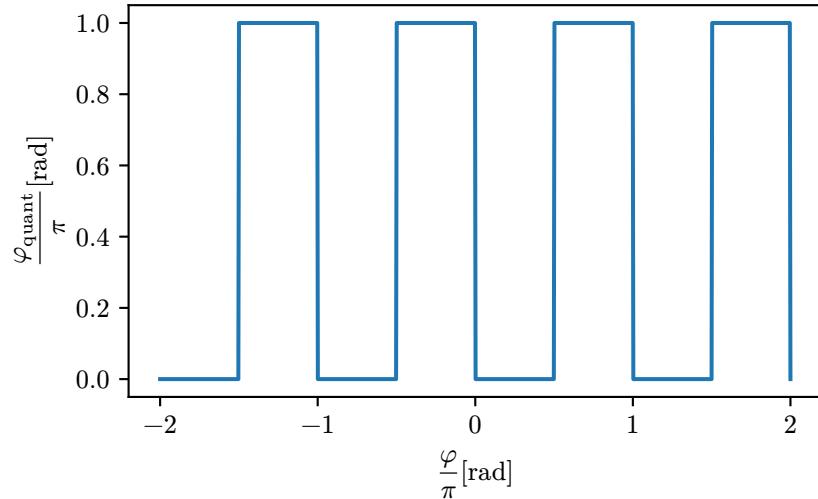


Abbildung 10: Veranschaulichung der Quantisierung der Phase

Wie in Abbildung 10 zu erkennen werden die Phasenwerte so auf 0 oder π gerundet.

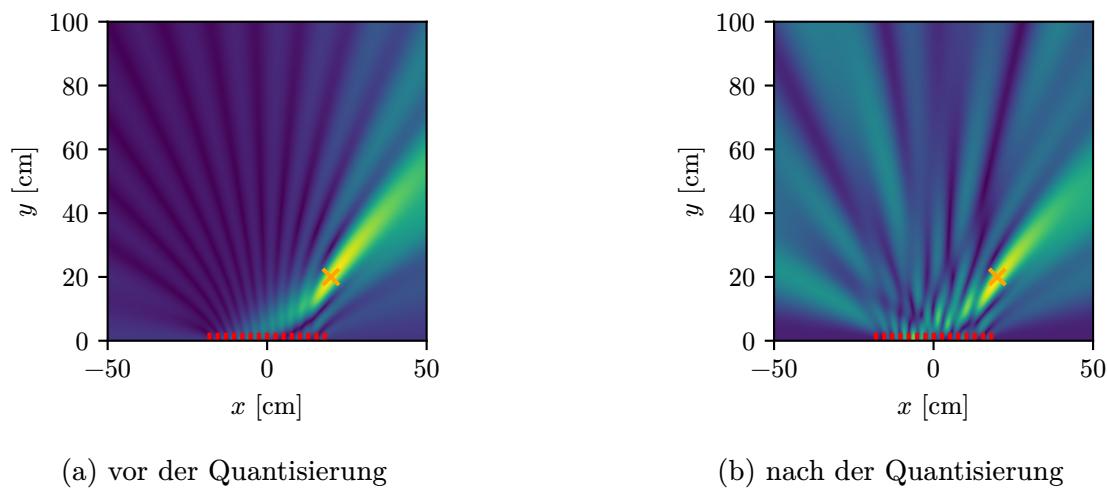


Abbildung 11: Vergleich der Interferenz durch ein Phasenmuster im Nahfeld vor und nach der Quantisierung

4 Software

Die für den Praktikumsversuch zur Verfügung gestellte RIS verwendet eine serielle Schnittstelle zur Kommunikation. Zur einfacheren Bedienung wurden einige Python Bibliotheken und eine grafische Benutzeroberfläche erstellt.

4.1 Python Bibliothek

Im folgenden soll die Verwendung der Python Bibliothek veranschaulicht werden. Zu aller erst müssen wir eine Verbindung zu einer RIS herstellen (1).

```

1  from opensourceris.common import IDConnection
2  from opensourceris.zeroconf.client import ZeroconfConnectionProvider
3
4  # der ZeroconfConnectionProvider findet verfügbare RIS
5  connection_provider = ZeroconfConnectionProvider()
6
7  # mit der IDConnection bauen wir eine Verbindung zu einer bestimmten RIS auf
8  connection = IDConnection("Demo", connection_provider)

```

Quellcode 1: Verbindungsauftbau zu einer RIS

Anschließend können wir ein Muster erzeugen und es an die RIS übertragen. In 2 ist zu sehen, wie ein einfaches Schachbrettmuster erstellt und an eine RIS übertragen werden kann.

```

1  import numpy as np
2  from opensourceris.common import bool_pattern_to_hex_pattern
3
4  # Wir benötigen eine Verbindung zum RIS.
5  connection = ... # hier nur ein Platzhalter
6
7  # Für das Muster erstellen wir ein 16x16 bool Array.
8  # True = 180 Grad Phasenverschiebung
9  # False = 0 Grad Phasenverschiebung
10 simple_pattern = np.zeros((16, 16), dtype=bool)
11
12 # In diesem Beispiel erstellen wir ein Schachbrettmuster.
13 for m in range(16):
14     for n in range(16):
15         simple_pattern[m][n] = (m + n) % 2 == 0
16
17 # Um das Muster zu übertragen,
18 # muss es noch in eine Hexadezimalschreibweise umgeformt werden.
19 # 4 Ziffern (16 Bit) repräsentieren hierbei immer eine Zeile.
20 hex_pattern = bool_pattern_to_hex_pattern(simple_pattern)
21
22 # Durch Setzen des pattern Attributs des Connection Objekts
23 # wird das Muster an die RIS übertragen.
24 connection.pattern = hex_pattern

```

Quellcode 2: Erstellung und Übertragung eines einfachen Musters zu einer RIS

4.2 Grafische Benutzeroberfläche

Die RIS können auch über eine grafische Benutzeroberfläche gesteuert werden.

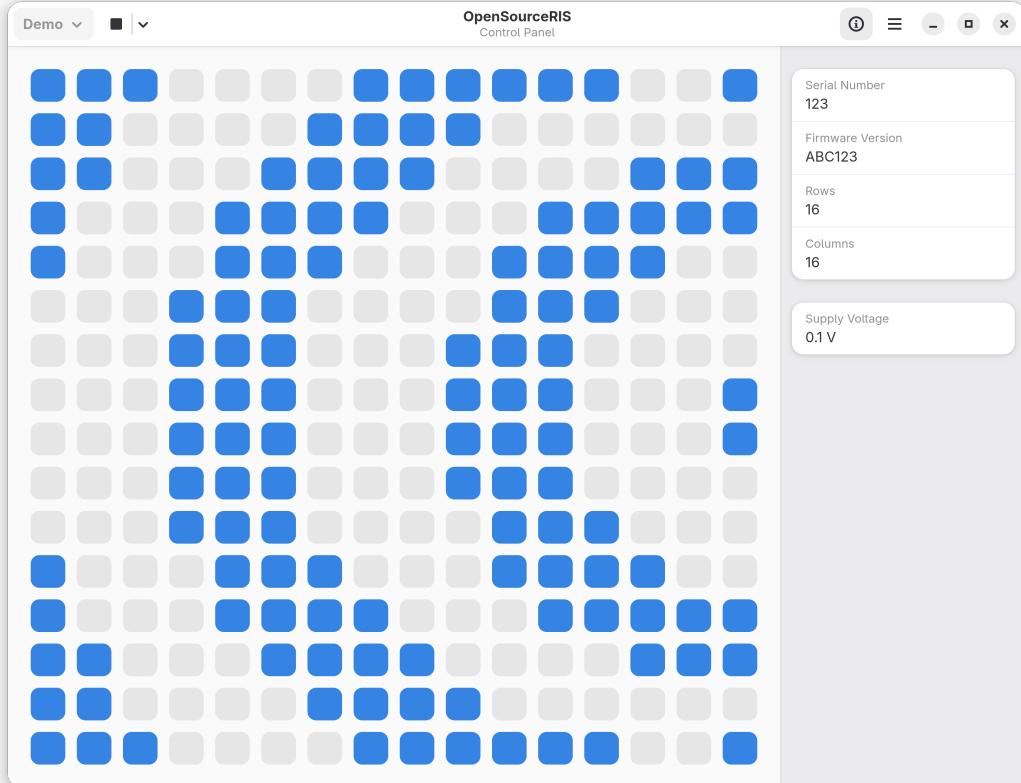


Abbildung 12: „Control Panel“ der grafischen Benutzeroberfläche

Im Hauptfenster (Abbildung 12) werden das aktive Muster, sowie Informationen zum Verbundenen RIS angezeigt. Die gewünschte RIS kann aus dem Auswahlmenü oben links gewählt werden. Um die Verbindung herzustellen, muss anschließend der Verbindungsbutton (►) angeklickt werden. Die Anwendung lädt nun alle Statusinformationen, sowie das aktuell auf dem RIS aktive Muster. Das aktive Muster kann durch Klicken der Schaltflächen manuell verändert werden. Das geänderte Muster wird automatisch auf das verbundene RIS übertragen.

4.2.1 Muster Generator

Neben dem manuellen Erzeugen von Mustern ist es zudem möglich, Muster anhand von Winkel und Distanz der Sende und Empfangsantenne zu generieren. Um Muster generieren zu können ist es zuvor nötig den „Pattern Generator“ (Abbildung 13) zu öffnen. Dieser kann im Menü (≡) in der Kategorie „Plugins“ gefunden werden.

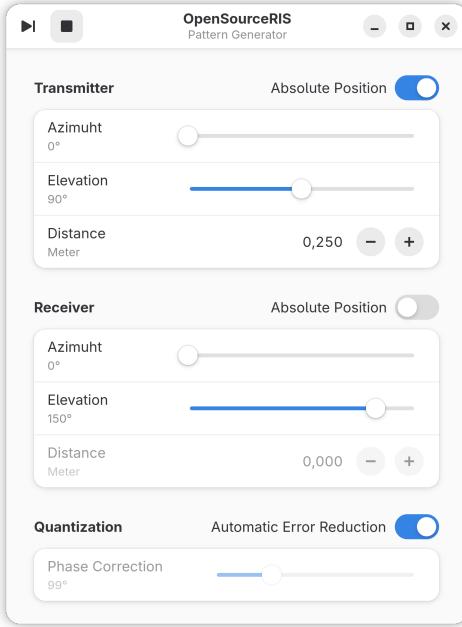


Abbildung 13: „Pattern Generator“ der grafischen Benutzeroberfläche

Nach klicken auf den Startbutton (▶) wird, bei Änderung der Einstellungen, automatisch das Muster der RIS aktualisiert.

4.2.2 SNR Plot

Um das SNR zu messen und den Verlauf über die Zeit darzustellen, kann der „SNR Plot“ (Abbildung 14) der grafischen Benutzeroberfläche verwendet werden. Dieser kann im Menü (≡) in der Kategorie „Plugins“ gefunden werden. Es ist wichtig den „SNR Plot“ zu öffnen bevor der Empfänger (Unterunterabschnitt 5.1.2) gestartet wird.

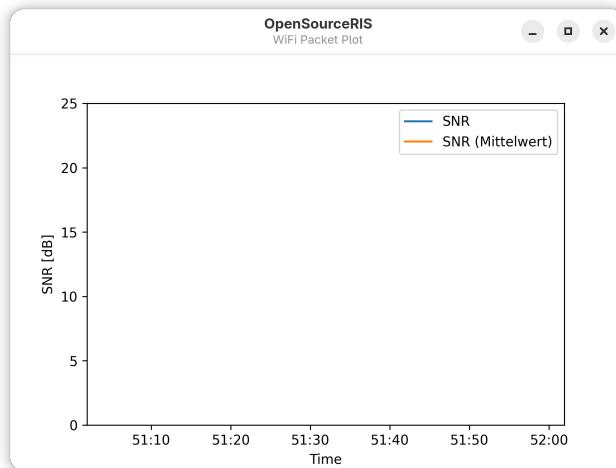


Abbildung 14: „SNR Plot“ der grafischen Benutzeroberfläche

4.2.3 Python Schnittstelle

Die grafische Benutzeroberfläche kann auch über eine Python Schnittstelle gesteuert werden (3).

```
1  from threading import Thread
2
3  from opensourceris.gui import PyOpensourceRISApplication
4
5  # Wir benötigen eine Verbindung zum RIS.
6  connection = ... # hier nur ein Platzhalter
7
8  # Die PyOpensourceRISApplication beinhaltet die Benutzeroberfläche.
9  app = PyOpensourceRISApplication.from_connection(connection)
10
11 # Die Oberfläche muss eine Verbindung zur RIS herstellen.
12 app.connect()
13
14 # Die Benutzeroberfläche muss in einem separaten Thread angezeigt werden.
15 Thread(target=lambda: app.run(None)).start()
16
17 # Nun können wir Muster erstellen und sie an die Benutzeroberfläche übergeben.
18 app.pattern = ... # hier nur ein Platzhalter
```

Quellcode 3: Verwendung der grafischen Benutzeroberfläche in Python

5 Versuchsaufbau

Für den Versuch stehen ein Sender, ein Empfänger und eine RIS zur Verfügung. Sender und Empfänger bestehen jeweils aus einem Software-Defined-Radio (Abbildung 16) und einer Antenne die auf die RIS gerichtet ist (Abbildung 15). Die RIS ist auf Höhe der Antennen platziert.



Abbildung 15: Foto des Versuchsaufbaus

Durch die starke Richtcharakteristik der Antennen besteht keine Sichtverbindung. Daher ist es nötig das Signal mithilfe der RIS von Sender in Richtung Empfänger zu reflektieren.

5.1 Sender/Empfänger



Abbildung 16: Foto des für den Versuch verwendeten Empfängers

Um eine Datenübertragung zwischen Sender und Empfänger zu starten stehen für den Versuch zwei Python Anwendungen zur Verfügung.

5.1.1 Sender

```

1 #!/bin/sh
2 conda activate ieee80211
3 python wifi_tx.py

```

Quellcode 4: Starten des Senders

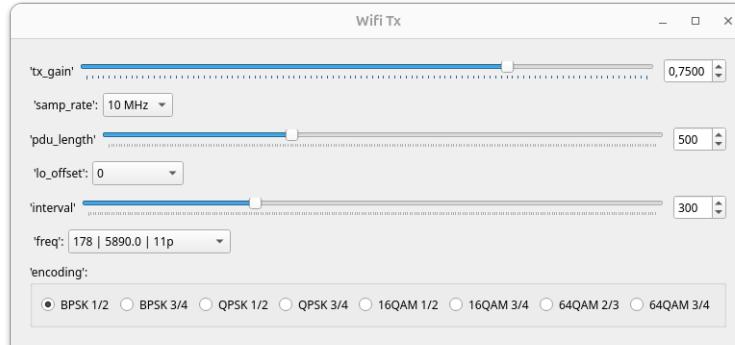


Abbildung 17: Benutzeroberfläche des Senders

5.1.2 Empfänger

```
1 #!/bin/sh
2 conda activate ieee80211
3 python wifi_rx.py
```

Quellcode 5: Starten des Empfängers

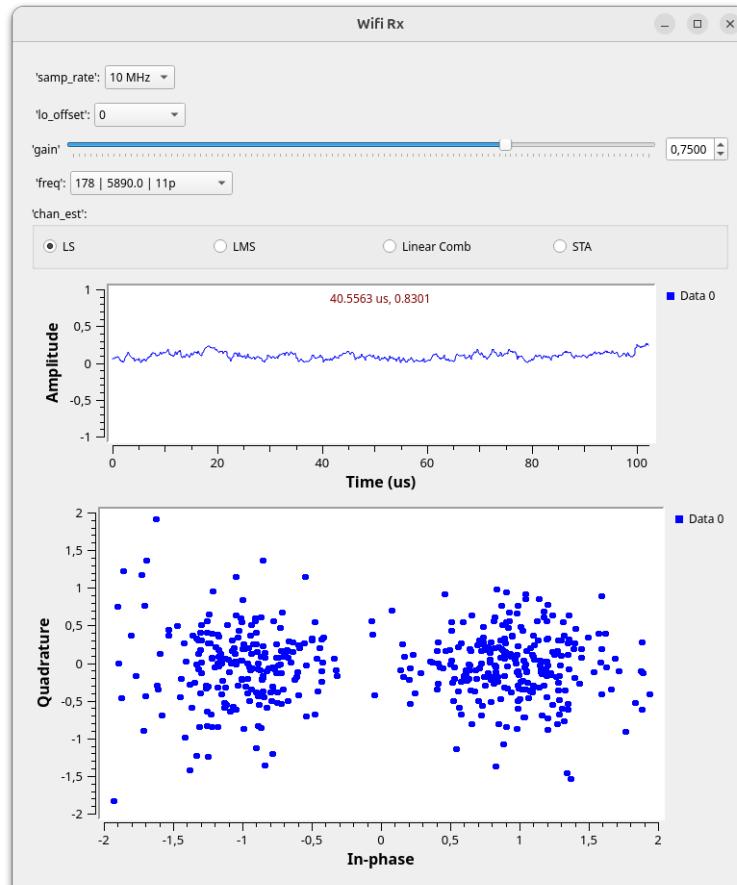


Abbildung 18: Benutzeroberfläche des Empfängers

5.2 Kurzfragen

Die folgenden Fragen (und die auf den vorherigen Seiten) sollst du Zuhause vorbereiten. Zu Beginn deines Praktikumstermins werden die Antworten abgefragt und besprochen.

In was für einem Szenario könnte ein RIS eingesetzt werden? Hier kannst du gerne kreativ sein.

Wie erzeugt die RIS konstruktive Interferenz?

6 Praktische Aufgaben

Nun sollst du selber mit der RIS arbeiten! Die folgenden Aufgaben wirst du während des Praktikums vor Ort bearbeiten. Zur besseren Vorbereitung solltest du dir die Aufgaben schon einmal im Voraus durchlesen.

6.1 Konfiguration anhand bekannter Positionen

Zuallererst soll ein Phasenmuster auf Basis des Versuchsaufbaus erstellt werden, durch welches am Empfänger konstruktive Interferenz erzeugt wird.

6.1.1 Versuchsaufbau vermessen

Hierfür muss der Versuchsaufbau vermessen werden. Es werden die Abstände der Sende- und Empfangsantenne zum RIS und der horizontale Einfall und Ausfallswinkel relativ zum RIS benötigt. Die gemessenen Werte können hier in die Tabelle eintragen werden.

Abstand: TX ↔ RIS	
Winkel: TX ↔ RIS	
Abstand: RIS ↔ RX	
Winkel: RIS ↔ RX	

6.1.2 Phasenmuster erstellen

Um das Muster zu erstellen, verwenden wir die in diesem Fall die grafische Benutzeroberfläche (Unterabschnitt 4.2). Öffne hierfür die py-OpenSourceRIS-GUI Anwendung und verbinde dich mit der bereitgestellten RIS. Anschließend öffne den Pattern Generator (Unterunterabschnitt 4.2.1) und stelle die zuvor gemessenen Werte ein. Übertrage nun das erstellte Muster an die RIS.

6.1.3 Einfluss des Musters Messen

Im nächsten Schritt wollen wir den Einfluss der RIS auf das SNR messen. Öffne hierfür den SNR Plot (Unterunterabschnitt 4.2.2) und starte anschließend Sender und Empfänger.

Welches SNR kannst du erreichen?

Ändere nun die im Pattern Generator eingestellten Winkel und beobachte die SNR Werte.

Warum verändern sich die SNR Werte?

6.2 Intelligenter vs normaler Reflektor

Nun soll herausgefunden werden, wie sich die RIS im Vergleich zu einem normalen Reflektor verhält. Hierfür soll eine Metallplatte das RIS ersetzen.

Welche Änderung der SNR erwartest du?

Welche Änderung der SNR kannst du erkennen?

Wie kommen diener Meinung nach die Unterschiede zustande?

Drehe nun die Metallplatte.

Welche Änderung der SNR erwartest du?

Welche Änderung der SNR kannst du erkennen?

Wie kommen diener Meinung nach die Unterschiede zustande?

6.3 Mustersuche

In der ersten Aufgabe hast du das Muster manuell erstellt. Nehmen wir an, wir könnten den genauen Versuchsaufbau nicht sehen. In diesem Fall können wir ein Muster iterativ suchen. Implementiere hierfür die Funktion „calculate_next_state“ (6).

```

1  from typing import TypedDict
2
3  import numpy as np
4
5  Pattern = np.ndarray[tuple[int, int], np.dtype[bool]]
6
7
8  class State(TypedDict):
9      pattern: Pattern  # 2D Boolean Array
10
11
12 def calculate_next_state(state: State, snr: float) -> State:
13     pattern = state['pattern']
14     ...
15     return State(pattern=pattern)

```

Quellcode 6: Programmcode zur Mustersuche

Gehe davon aus, das die Funktion regelmäßig aufgerufen wird. Beim Aufruf wird der aktuelle Zustand und die mittlere SNR seit dem letzten Aufruf übergeben. Deine Aufgabe ist es hieraus einen möglichst besseren Zustand zu berechnen. Das erzeugte Zustandsobjekt muss immer ein Muster beinhalten. Dieses Muster wird nach jedem Aufruf der Funktion, an die RIS übertragen. Ansonsten kannst du deiner Kreativität freien lauf lassen.

Als Beispiel, könntest du eine „simulierte Abkühlung“ implementieren. Du kannst aber auch sehr gerne versuchen einen anderen Weg zu finden.

Wie funktioniert der von dir eingesetzte Algorithmus? Wie verhält er sich gegenüber den berechneten Mustern?

6.3.1 Simulierte Abkühlung

Bei der „simulierten Abkühlung“ handelt es sich um ein probabilistisches Optimierungsverfahren. In unserem Fall versuchen wir ein globales Maximum des SNR zu finden. Nennen wir unser initiales Phasenmuster $\vec{\varphi}_i$. Auf Basis von $\vec{\varphi}_i$ können wir ein Phasenmuster $\vec{\varphi}_j$ erstellen. Dies kann z. B. durch Umschalten einer zufälligen Spalte des Musters geschehen. Da wir für jedes Phasenmuster auch eine SNR erhalten können wir erkennen, welches Muster besser geeignet ist. Wir können uns hierbei vorstellen, dass wir den in Gleichung 21 beschriebenen Zusammenhang betrachten.

$$\begin{aligned} f(\vec{\varphi}_i) &= \text{SNR}_i \\ f(\vec{\varphi}_j) &= \text{SNR}_j \\ \Delta \text{SNR}_{i,j} &= \text{SNR}_i - \text{SNR}_j \end{aligned} \tag{21}$$

Würden wir nun immer Muster auf Basis der besseren Muster erzeugen, würden wir zwar ein lokales Maximum finden. Dies könnte aber weit von dem globalen Maximum entfernt liegen. Durch die „simulierte Abkühlung“ können wir die Wahrscheinlichkeit erhöhen das globale, oder zumindest ein besseres lokales Maximum zu finden. Hierfür akzeptieren wir mit einer gewissen Wahrscheinlichkeit auch schlechtere Muster. Diese Wahrscheinlichkeit sinkt jedoch mit jedem neuen Muster und fällt geringer aus, je schlechter das neue gegenüber dem vorherigen Muster ist. In Gleichung 22 ist die Berechnung der Wahrscheinlichkeit beschrieben.

$$p_{i,j} = \begin{cases} 1 & , \Delta \text{SNR}_{i,j} \leq 0 \\ \exp\left(-\frac{\Delta \text{SNR}_{i,j}}{T}\right) & , \Delta \text{SNR}_{i,j} > 0 \end{cases} \tag{22}$$

Der Parameter T wird nach jeder Iteration mit einem Faktor $\alpha < 1$ multipliziert.