

Lab 2: Distanzmessung mit Pseudo-Noise-Signalen

1. Einleitung

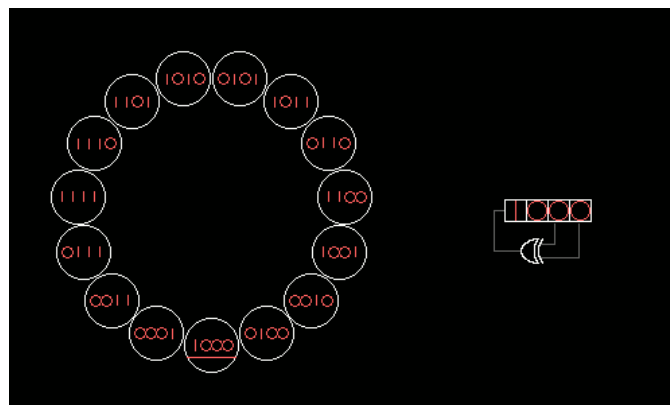
In diesem Praktikum soll mit Hilfe eines **Pseudo-Noise (PN)** Signals und der Autokorrelation die Schalllaufzeit gemessen werden, um damit eine Distanz zu bestimmen. Ein alternatives «Pulskompressionssignal» ist das Chirp-Signal, siehe auch [https://de.wikipedia.org/wiki/Pulskompressionsverfahren_\(Ortung\)](https://de.wikipedia.org/wiki/Pulskompressionsverfahren_(Ortung)).

2. Vorbereitung: PN-Sequenz und LFSR

- a) Generieren Sie mit dem Python-Skript `akf_pn_template.py` ein binäres PN-Signal $s[\cdot]$ mit $L = 15$ Samples mittels eines linear rückgekoppelten Schieberegisters, siehe https://de.wikipedia.org/wiki/Linear_r%C3%BCckgekoppeltes_Schieberegister. Die englische Bezeichnung dafür lautet Linear-feedback Shift Register (LFSR). Vergewissern Sie sich, dass im PN-Signal die Werte 0 und 1 fast gleich häufig vorkommen und alle 4-Bit-Zustände ausser dem Null-Zustand durchlaufen werden, bevor sich das Signal wiederholt. Eine Periode des Signals ist damit maximal lang und deshalb wird das Signal oft auch als «Maximal Length Sequence» (MLS) bezeichnet.

In folgendem Video finden Sie Informationen zu LFSR verschiedener Länge:

https://www.youtube.com/watch?v=iqGKAu5_IIE



- b) Generieren Sie nun mehrere Perioden eines bipolaren (nur +1 und -1 Werte) PN-Signals, z.B. $\underline{x} = [\underline{s} \ \underline{s} \ \underline{s} \ \underline{s}]$, und korrelieren Sie das resultierende Signal mit dem bipolaren PN-Signal $s[n]$.

Die Autokorrelation kann durch Filterung mit einem FIR-Filter mit den Koeffizienten $b = s[:, : -1]$ (s rückwärts ausgelesen) bestimmt werden.

Vergewissern Sie sich, dass die Autokorrelationsfunktion AKF impulsartig ist, genau wie es für eine längere «Münzwurf»-Folge zu erwarten ist.

3. Distanzmessung

- c) Bauen Sie das folgende Messsystem zur schallbasierten Bestimmung der Distanz auf:

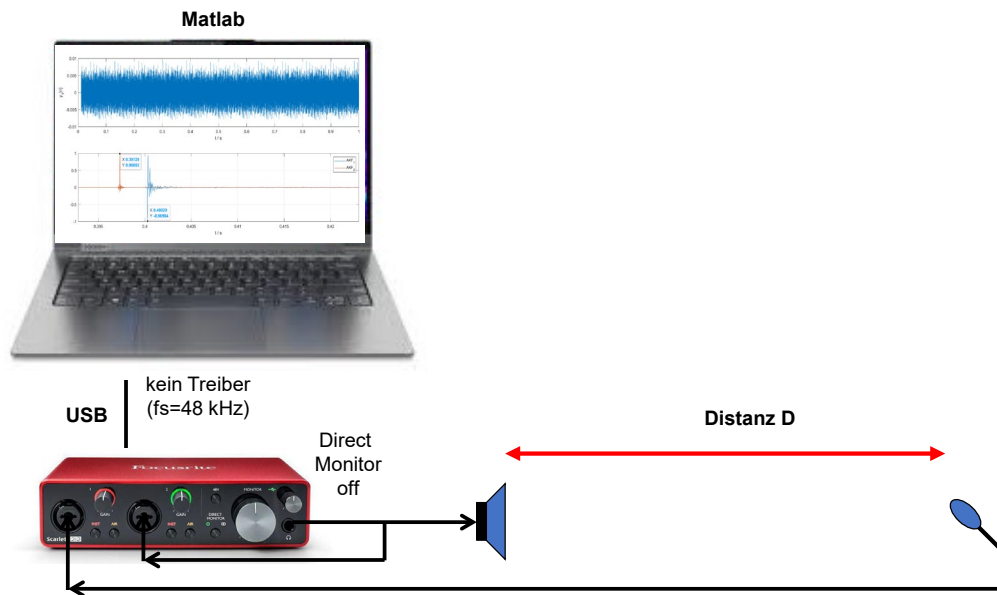


Abbildung 1: Messsystem zur Bestimmung der Distanz D.

- d) Generieren Sie mit dem LFSR $1+X^6+X^8+X^{11}+X^{12}$ ein PN-Signal $s[n]$ der Länge $L = 2^{12}-1$ und senden Sie es mehrmals nacheinander mit der `playrec`-Funktion aus. Sie können dazu die Vorlage `impulse_response_tof.py` verwenden.

Lesen Sie mit dem USB-Audio-Interface das ausgesendete Signal und das Mikrofon-Signal ein, berechnen Sie je die AKF mit dem PN-Signal $s[n]$ und lesen Sie die Zeitdifferenz Δt der beiden Korrelationsspitzen ab. Tipp: Normieren Sie die AKF so, dass das der Betrag das Maximum 1 aufweist.

Die Distanz D kann dann wie folgt berechnet werden: $D = \Delta t \cdot c$, wobei die Schallgeschwindigkeit in der Luft $c \approx 343$ m/s.

Wie stark stören Fremdgeräusche die Distanzbestimmung?

Wie gross ist die Messrate, d.h. wie viele Messungen pro s werden gemacht?

- e) Verwenden Sie statt dem PN-Signal einmal das folgende chirp-Signal und vergewissern Sie sich, dass es auch eine impulsförmige AKF hat.

```
fs = 48e3
f0 = 20
f1 = 20e3
Tsweep = 5
t = np.arange(Tsweep*fs-1)/fs
s = scipy.signal.chirp(t, f0, Tsweep, f1)
```