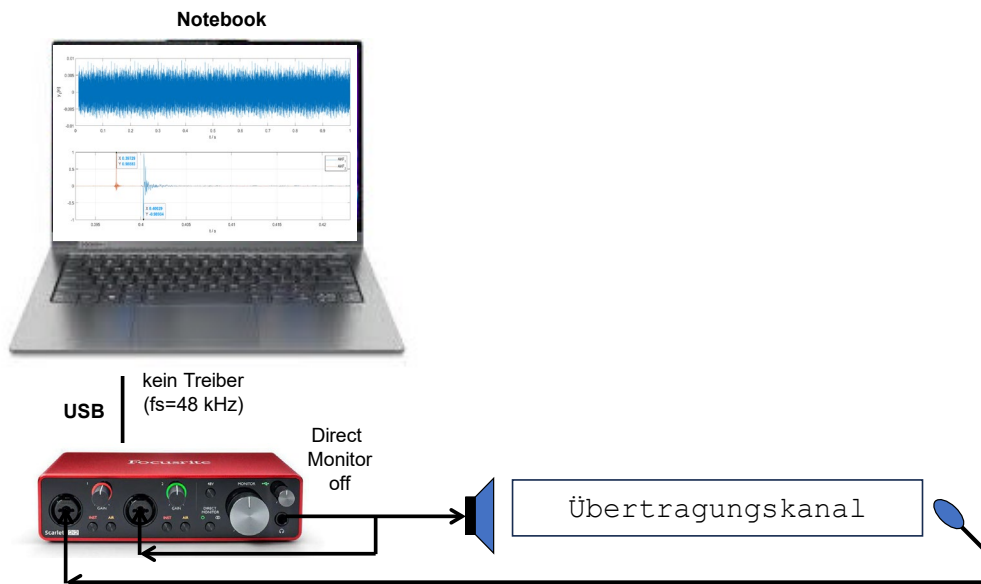


## Lab 3: Messung der Raumimpulsantwort

### 1. Einleitung

In diesem Praktikum soll mit Hilfe eines **Pseudo-Noise (PN)** Signals und der Kreuzkorrelation eine Impulsantwort gemessen werden, um

- ein lineares System (lineares Filter) zu identifizieren, und
- den «Hall» (genauer die Raumimpulsantwort) verschiedener Räume auszumessen.



**Abbildung 1:** Messsystem zur Systemidentifikation.

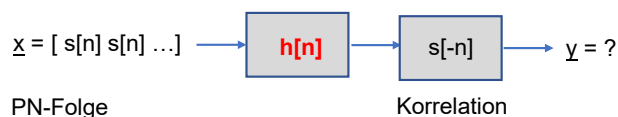
## 2. System-Identifikation

a) Verwenden Sie im Messaufbau von Abbildung 1 statt dem Lautsprecher und Mikrofon das am Arbeitsplatz liegende Tiefpass-Filter und bestimmen Sie dessen Impulsantwort  $h[n]$ . Nehmen Sie hierfür das Skript `impulse_response_sysID_Template.py` als Ausgangspunkt.

b) Vergewissern Sie sich, dass Sie tatsächlich die Impulsantwort des Filters messen und dass Sie dafür dank der impulsförmigen AKF keine «grosse» Peak-Amplitude brauchen.

Vielleicht hilft Ihnen die folgende Zeichnung:

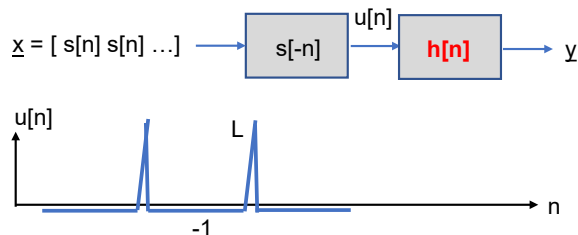
### korrektes Blockschaltbild



Man kann zeigen, dass  $\underline{y} = [L \cdot h[n] \ L \cdot h[n] \dots]$  d.h. am Ausgang erscheint periodisch die L-fach (Länge der PN Sequenz) verstärkte Impulsantwort  $h[n]$

Begründung: Die beiden linearen Filter darf man für die Analyse vertauschen, ohne dass sich das Ausgangssignal  $\underline{y}$  verändert!

### Bezüglich Ausgangssignal äquivalentes Blockschaltbild



$u[n]$  ist also eine periodische Dirac-Impuls-Folge mit Verstärkung  $L$  damit erscheint am Ausgang periodisch die L-fach verstärkte Impulsantwort  $h[n]$

c) Extrahieren Sie die Impulsantwort  $h[n]$  und bestimmen Sie mit der FFT den Frequenzgang  $H(f)$ . Sie können mit der Numpy-Funktion `argmax()` den Index der maximalen Korrelationsspitze bestimmen und dann  $L$  Samples vor oder nach dem Index ausschneiden. Erklären Sie kurz Ihr Messresultat (Pulsdauer, Bandbreite, DC, ...).

### 3. Raum-Impuls-Antwort

- a) Verwenden Sie im Messaufbau von Abbildung 1 den Lautsprecher und das Mikrofon. Erweitern Sie Ihr Programm so, dass Sie die Impuls-Antwort eines Raums messen können. Verwenden Sie hierfür ein PN-Signal mit  $L = 2^{18}-1$  Samples ( $s[m] = (s[m-18] + s[m-11]) \% 2$ ) und verlängern Sie die Aufnahmezeit. Extrahieren Sie ca. 1s der Raum-Impuls-Antwort  $h[n]$ .

Plotten Sie die Impulsantwort einmal mit linearer und einmal mit logarithmischer Amplitude in dB. Wie sieht die Impulsantwort aus und wie lange «dauert» sie?

- b) Messen Sie nun die Raum-Impuls-Antwort eines «halligen» Raums (TS-Treppenhaus) (Achtung: PN-Sequenz nur noch 3 Mal wiederholen und Aufnahmezeit auf 15s setzen).

Plotten Sie Raum-Impuls-Antwort  $h_{RIR}[n]$  mit linearer und logarithmischer Amplitude.

Können Sie einzelne Echos zeitlich «lokalisieren», z.B. auf dem direkten Pfad zwischen Lautsprecher und Mikrofon, oder von starken Reflexionen an den Wänden?

Wann beginnt der diffuse Nachhall und mit wie vielen dB pro Zeiteinheit nimmt der Nachhall am Anfang ab?

- c) Speichern Sie die Impuls-Antwort in einer wav-Datei ab und benutzen Sie das Python-Programm `audio_mit_hall.py`, um ein (trockenes) Audiosignal in Ihrem akustisch ausgemessenen Raum anzuhören.

**Bitte geben Sie eine wav-Datei eines Audio-Beispiels ohne/mit resultierendem Hall ab.**

- d) Analysieren Sie die gemessene Raum-Impuls-Antwort  $h_{RIR}[n]$  mit einem Spektrogramm und zeigen Sie, dass tiefe Frequenzkomponenten länger «nach-hallen» als hohe.

Code:

```
NFFT = 1024
window = np.hamming(NFFT)

plt.specgram(h, window=window, noverlap=0, NFFT=NFFT, Fs=fs)
```

Optional: Wenn Sie noch Zeit haben:

- 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)
```