



## Kommunikationstechnik I

### 3. Python-Übung

#### Aufgabe 1: Entwurf von Kosinus-roll-off-Filtern

Die Funktion `cosroll` berechnet die Impulsantwort eines Kosinus-roll-off-Filters nach der geschlossenen Formulierung gegeben durch

$$g_{rc0}(t) = \frac{\sin(\pi t/T)}{\pi t/T} \cdot \frac{\cos(\pi rt/T)}{1 - (2rt/T)^2} \quad (1)$$

mit einer Rechteckfensterbegrenzung.<sup>1</sup> Führen Sie Entwürfe für  $r = 0.5$  (Zeitbegrenzung auf  $L = 8$  Symbolintervalle) und  $r = 1.0$  ( $L = 4$ ) durch; sehen Sie 4 Abtastwerte pro Symbolintervall vor. Stellen Sie die Impulsantwort sowie die Übertragungsfunktion (verwenden Sie hierfür die Funktion `fft` mit  $N = 64$  Punkten) graphisch dar.

Was lässt sich anhand der Impulsantworten über Nulldurchgänge ableiten? Zu welchen Werten konvergiert  $g_{rc0}(t)$  für  $t \rightarrow 0$  bzw. für  $t \rightarrow \pm T/(2r)$ ?

#### Aufgabe 2.1 Impulsformung

Simulieren Sie die Datenübertragungsstrecke aus Abbildung 1. Gehen Sie in folgenden Schritten vor:

- Erzeugen Sie zunächst einen Datenvektor  
 $\underline{d} = [1 \ -1 \ 1 \ 1 \ -1 \ -1]^T$ .
- Erzeugen Sie anschließend die Impulsantwort des Kosinus-Roll-Off-Pulsformfilters. Verwenden Sie dafür die Funktion `cosroll` mit den Parameterwerten  $r = 0.5, L = 8, w = 8$ . Hierbei ist  $w$  die Anzahl von Abtastungen pro Symbolintervall ist.
- Überabtasten Sie den Datenvektor  $\underline{d}$  mit Faktor  $w$  mittels Zero-Stuffing, nutzen Sie den Befehl `numpy.zeros`.
- Erzeugen Sie das Ausgangssignal des Pulsformfilters durch Faltung (Funktion `numpy.convolve`) des überabgetasteten Datenvektors mit der Impulsantwort des Filters.

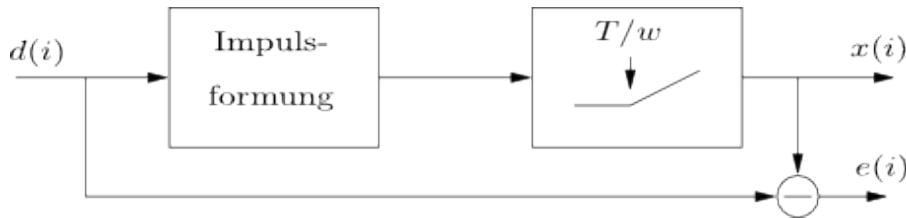
---

<sup>1</sup>Für  $r = 0$  wird in der Funktion `cosroll` automatisch ein  $\sin(x)/x$ -Impuls der Länge  $LT$  eingesetzt.

- Stellen Sie das Ausgangssignal in einem Diagramm dar. Beachten Sie die Ein- und Ausschwingvorgänge.
- Tasten Sie das Ausgangssignal an den Stellen  $t = iT$  ab und vergleichen Sie das Resultat mit dem Datenvektor  $\underline{d}$ .

## Aufgabe 2.2 Intersymbolinterferenz

Intersymbolinterferenz entsteht durch nichtideale Sende- und Empfangsfilter, durch Kanaleinflüsse oder auch durch Nichteinhaltung des idealen Abtastzeitpunktes am Empfänger. Simulieren Sie eine Datenübertragungstrecke wie in Abbildung 1 gegeben für einen Abtastfehler von  $\Delta t = T/16$ .



**Abbildung 1:** Übertragungsstrecke

Gehen Sie in folgenden Schritten vor:

- Erzeugen Sie zunächst einen binären Datenvektor  $\underline{d}$  bestehend aus  $N = 10000$  gleichwahrscheinlichen Elementen  $d(i) \in \{-1, 1\}$ ,  $i = 1, \dots, N$ . Sie können dazu zum Beispiel die Funktionen `numpy.random.randn` und `numpy.sign` verwenden.
- Erzeugen Sie anschließend die Impulsantwort des Kosinus-roll-off-förmigen Pulsformfilters. Verwenden Sie dafür die Funktion `cosroll` (Für die Approximation der Kosinus-roll-off-Impulse werden Zeitbegrenzungen  $L = 8$  für  $r = 1$ ,  $L = 16$  für  $r = 0.5$ ,  $L = 256$  für  $r = 0$  vorgeschlagen).
- Erzeugen Sie nun das Datensignal mit Hilfe der Funktion `datensig`. Als Abtastfrequenz wählen Sie  $f_A = w/T$ ,  $w = 16$ . Ein- und Ausschwingvorgänge sind hierbei zu unterdrücken, was durch Wahl der Eingabeparameters `A = True` erreicht wird.
- Entnehmen Sie dem Datensignal die Werte an den Stellen  $t = iT + T/16$ .
- Wie hoch ist das Maximum der Beträge der Intersymbolinterferenz (ISI) an den  $N = 10000$  oben genannten Stellen  $t = iT + T/16$ ?
- Ermitteln Sie für alle  $N = 10000$  oben genannten Stellen  $t = iT + T/16$  das aktuelle Verhältnis  $(S/N)_{ISI,i}$ ,  $i = 1, \dots, N$  von Signal- zu Störleistung durch Intersymbolinterferenz (ISI).
- Berechnen Sie den Mittelwert von  $(S/N)_{ISI,i}$ ,  $i = 1, \dots, N$  und geben diesen in dB an.

### Aufgabe 3: Augendiagramm

Zur Veranschaulichung von Datenübertragungssystemen wird häufig das sogenannte Augendiagramm dargestellt. Man erzeugt es, indem eine große Anzahl von Signalabschnitten der Dauer eines Symbolintervalls übereinandergezeichnet werden - z.B. durch die Wiedergabe über einen Speicheröszillographen, der mit dem Symboltakt getriggert wird. Auf diese Weise erzeugt man augenähnliche Muster, was anhand der nachfolgenden Beispiele illustriert wird. Zur Darstellung von Augendiagrammen steht die Funktion `auge` zur Verfügung.

3.1 Erzeugen Sie Augendiagramme mit Hilfe von Python und den Ihnen zur Verfügung gestellten Funktionen. Gehen Sie dazu wie folgt vor:

- Generieren Sie einen Vektor  $\underline{d}$  mit unabhängigen Daten  $d(i) \in \{-1, 1\}$ .
- Erzeugen Sie anschließend die Impulsantwort des Kosinus-roll-off-förmigen Pulsformfilters, für  $r = 1$ ,  $r = 0.5$  und  $r = 0$ .
- Erzeugen Sie nun mit Hilfe der Funktion `datensig` das Datensignal mit der Abtastfrequenz  $f_A = w/T$ .
- Unterdrücken Sie Ein- und Ausschwingvorgänge mit Hilfe des Parameters `A`.

Für die Festlegung der Datenzahl  $N$  wird vorgeschlagen:  $N = 1000$  für  $r = 1$  und  $r = 0.5$  und  $N = 100000$  für  $r = 0$ . Wenn dies technisch nicht möglich ist, wählen Sie  $N = 5000$  für  $r = 0$ , wobei Sie weniger exakte Ergebnisse erhalten werden.

3.2 Bestimmen Sie aus den Augendiagrammen jeweils die relative vertikale Augenöffnung für einen Abtastfehler  $\Delta t = T/16$ . Die relative vertikale Augenöffnung ist definiert als innerer Augenrand bezogen auf das ungestörte Nutzsignal. Erklären Sie die Ergebnisse anhand der in Aufgabe 2 simulierten ISI-Maximalwerte.

Bestimmen Sie weiterhin aus den Augendiagrammen jeweils die relative horizontale Augenöffnung; sie ergibt sich aus den von der Augenmitte aus gemessenen Zeitpunkten  $t_-$  und  $t_+$ , an denen die vertikale Augenöffnung erstmals Null wird.