

Ausgewählte Kapitel der Elektronik 2

Aufgaben und Hinweise

SoSe 2022

Inhalt

1	Allgemeines	3
2	Aufgabenstellungen	4
2.1	Echo-Cancellation in Audio-Signal (Gruppe 1)	4
2.2	Frequenzgangmessung mittels Audiosignalen (Gruppe 2)	5
2.3	FM-Modulation/Demodulation (Gruppe 3)	7
2.4	QPSK-Modulation/Demodulation (Gruppe 4)	8

1 Allgemeines

Für die folgenden Aufgabenstellungen sind in den Kleingruppen nach einer kurzen Entwurfs- bzw. Konzeptionierungsphase Simulationen mit MATLAB durchzuführen und anschließend zu analysieren. Die Ergebnisse müssen direkt im MATLAB-Code nachvollziehbar dokumentiert werden und sind gesammelt in einer ZIP-Datei (inkl. Testfiles) über die Moodle-Plattform abzugeben.

Sämtliche Berechnungen/Auswertungen/Vorbereitungen sind vollständig in einem oder mehreren selbsterklärend kommentierten MATLAB-Skripts durchzuführen.

Verwenden Sie dazu sinnvolle Variablenbezeichnungen mit Bezug zur Analyse. Damit sind automatisch Ihre Berechnungen nachvollziehbar und gleich mit Kommentar versehen dokumentiert. Der Vorteil für Sie ist auch, dass Berechnungen schnell und einfach mit geänderten Werten (z.B. nach Korrekturen) wiederholt werden können.

Beispiel:

```
R1 = 10e3;           % Widerstandswert 10k angenommen
UR2 = 2;             % Spannung, die an R2 abfallen soll
UB = 5;              % Versorgungsspannung
R2 = R1 / (UB/UR2-1) % erforderlicher Widerstandswert R2
```

Bei der Abgabe muss auch eine Readme-Textdatei mit Beschreibung der Anwendung der erstellten MATLAB-Skripte enthalten sein. Aus dieser muss klar hervorgehen, in welcher Reihenfolge die einzelnen Skripte ausgeführt werden müssen, um zu einem Ergebnis zu kommen.

Achten Sie auf die Amplitude der generierten Audio-Signale. Diese darf nie größer als 1 sein, sollte sicherheitshalber sogar etwas weniger als 1 sein!

2 Aufgabenstellungen

2.1 Echo-Cancellation in Audio-Signal (Gruppe 1)

- > Erzeugen eines Audiosignals mit mindestens zwei Echos (vorzugsweise Audio-Sprachsignal, weil bei Musiksignalen Echos möglicherweise als „vorteilhafte“ Soundeffekte wahrgenommen werden und weniger als Echos erkennbar sind)
- > Identifizierung aller Echos im Hinblick auf jeweilige Verzögerung und Größe mittels Korrelation (MATLAB: z.B. Befehl `xcorr`)
- > Realisierung eines Equalizers (Entzerrer) zur vollständigen Entfernung aller Echos Hinweis: Die Übertragungsfunktion des Equalizers muss die Verarbeitung zur Erzeugung der Echos kompensieren, d.h. invers dazu sein.
- > Speichern des Ausgangssignals des Equalizers als Audio-Datei zwecks Hörprobe
- > Test mit verschiedenen Audiosignalen

Gehen Sie dabei wie folgt vor:

1. Wählen Sie eine Musikaudio-Datei mit überwiegendem Gesang bzw. eine Sprachaudio-Datei.
2. Erstellen Sie ein MATLAB-Skript für die folgenden Unterpunkte:
 - a. Lesen Sie aus der gewählten Audiodatei das Audiosignal und fügen Sie mindestens zwei unterschiedlich abgeschwächte Echos (d.h. verzögerte Versionen des ursprünglichen Audiosignals) additiv hinzu.
 - b. Wählen Sie eine geeignete Skalierung des Ergebnisses und speichern Sie es in einer Audiodatei. Das Abspielen der generierten Audiodatei muss die Echos klar erkennen lassen!
3. Geben Sie einen allgemeinen mathematischen Ausdruck (Stichwort: Differenzengleichung) für das Ausgangssignal des „Echogenerators“ an. Bestimmen Sie ebenso die zugehörige Z -Transformierte.
4. Bestimmen Sie durch Rechnung die Übertragungsfunktion eines digitalen Systems (Equalizer oder Entzerrer), welches die Wirkung des „Echogenerators“ wieder vollständig kompensiert. Ermitteln Sie daraus die zugehörige Differenzengleichung des Equalizers.
5. Erstellen Sie ein weiteres MATLAB-Skript für die folgenden Unterpunkte:
 - a. Lesen Sie aus der Audiodatei mit Echos das Audiosignal und ermitteln Sie mit Hilfe der MATLAB-Funktion `xcorr` die Verzögerungen und Abschwächungen der einzelnen Echos. Verwenden Sie dazu einen Signalausschnitt unmittelbar zu Beginn des Signales bevor noch das erste Echo auftritt. Mit Hilfe von `xcorr` kann ein wiederholtes Auftreten dieses Ausschnittes (=Echo) gefunden werden. Ein längerer Ausschnitt liefert in der Regel bessere Erkennungsergebnisse bzgl. Echos. Für Details zur Funktion siehe die MATLAB-Hilfefunktion.
 - b. Realisieren Sie den Equalizer im MATLAB-Skript und berücksichtigen Sie dabei die im vorhergehenden Schritt ermittelten Werte.
 - c. Entfernen Sie die Echos aus dem Audiosignal.
 - d. Wählen Sie eine geeignete Skalierung des Ergebnisses vom Equalizer und speichern Sie es in einer Audiodatei. Das Abspielen der generierten Audiodatei muss das ursprüngliche Audiosignal ohne Echos wiedergeben!
6. Testen Sie Ihre Lösung mit Audiodateien verschiedener Musikrichtungen/-stile.

2.2 Frequenzgangmessung mittels Audiosignalen (Gruppe 2)

- > Generierung eines geeigneten Audiosignals zur Ansteuerung einer frequenzabhängigen Schaltung
- > Aufzeichnung des Eingangs- und Ausgangssignals über den LineIn-Eingang einer Soundkarte (synchrone Aufnahme beider Signale als ein Stereosignal, z.B. mittels freier Software Audacity und anschließende Nachbearbeitung)
- > Analyse in MATLAB:
 - Betrag des Übertragungsfaktors $|H(j\omega)| = \frac{\hat{U}_{\text{aus}}}{\hat{U}_{\text{ein}}}$ über Amplitudenmessungen bestimmen.
 - Winkel $\Delta\varphi$ (=Phasenverschiebung) unter Ausnutzung der Beziehung $\sin(\alpha) \cdot \sin(\beta) = \frac{1}{2} [\cos(\alpha - \beta) - \cos(\alpha + \beta)]$ bestimmen. Mit $\alpha = \omega t + \Delta\varphi$ und $\beta = \omega t$ folgt $\sin(\omega t + \Delta\varphi) \cdot \sin(\omega t) = \frac{1}{2} [\cos(\Delta\varphi) - \cos(2\omega t + \Delta\varphi)]$, wobei der zweite, höherfrequente Term durch Filterung eliminiert werden kann.
- > Graphische Ausgabe des Bodediagramms (Betrag + Winkel) mit logarithmischer Frequenzachse und geeigneter Skalierung der Messgrößen

Gehen Sie dabei wie folgt vor:

1. Erstellen Sie ein MATLAB-Skript für die folgenden Unterpunkte:
 - a. Generieren Sie ein geeignetes Mono-Testsignal für die Ausgabe über den Audioausgang zur Ansteuerung einer frequenzabhängigen Schaltung. Berücksichtigen Sie dabei eine gewisse Vorlaufzeit mit Stille, damit bei der praktischen Messung genügend Zeit zum Starten der Aufnahme bleibt.
 - b. Wählen Sie eine geeignete Skalierung des Testsignals und speichern Sie es in einer Audiodatei.
2. Geben Sie das Audio-Testsignal über die Soundkarte aus und steuern Sie damit eine frequenzabhängige Schaltung (z.B. RC-Tiefpass) an. Achten Sie darauf, dass die Eingangsimpedanz der Messschaltung den Audioausgang nicht zu sehr belastet. Es wird empfohlen die Eingangsimpedanz nicht kleiner als 10kΩ zu wählen.
3. Nehmen Sie das Ausgangssignal der frequenzabhängigen Schaltung gleichzeitig mit dem Eingangssignal (Audio-Testsignal von oben) über den LineIn-Eingang einer Soundkarte als Stereosignal (z.B. linker Kanal ist Eingangssignal, rechter Kanal ist Ausgangssignal) auf und speichern Sie es als Audiodatei ab.¹

Die Aufnahme über einen Mikrofoneingang ist nicht ausreichend. Zur Messung der Phasenverschiebung müssen nämlich beide Signale synchron aufgenommen werden. Außerdem weisen Mikrofoneingänge üblicherweise einen limitierten Audio-Frequenzbereich auf. Und zudem werden durch die Messung des Eingangssignals auch eventuelle Belastungseffekte mitberücksichtigt.

Die Wiedergabe und Aufnahme können gleichzeitig auf einem Rechner (z.B. mit Open Source-Programm Audacity unter <https://www.audacityteam.org/>) oder getrennt mit zwei Rechnern erfolgen.
4. Trimmen Sie die aufgenommenen Signale derart, dass nur mehr relevante Signalanteile, d.h. z.B. keine Stille-Phasen, enthalten sind. Dies kann mit einem Audioprogramm (z.B. Audacity) oder automatisiert direkt in MATLAB (siehe nächster Punkt) erfolgen.
5. Erstellen Sie ein weiteres MATLAB-Skript für die folgenden Unterpunkte:
 - a. Ermitteln Sie den Betrag des Frequenzganges durch Bestimmung der beiden Signalstärken \hat{U}_{ein} und \hat{U}_{aus} (bzw. entsprechende Effektivwerte). Dies darf **nicht** durch eine direkte Suche nach dem (lokalen) Maximum der Samples erfolgen! Berücksichtigen

¹ Wenn keine Soundkarte mit LineIn-Eingang zur Verfügung steht, kann eine entsprechende USB-Soundkarte leihweise zur Verfügung gestellt werden.

Sie, dass es sich (quasi) um sinusförmige Signale handelt und die Signalstärke durch geeignete Mittelung über mehrere Samples folgt.

- b. Ermitteln Sie die Phasenverschiebung unter Ausnutzung der Beziehung $\sin(\alpha) \cdot \sin(\beta) = \frac{1}{2} [\cos(\alpha - \beta) - \cos(\alpha + \beta)]$. Mit dem Eingangssignal $u_{\text{ein}}(t) = \hat{U}_{\text{ein}} \sin(\omega t)$ und dem Ausgangssignal $u_{\text{aus}}(t) = \hat{U}_{\text{aus}} \sin(\omega t + \Delta\varphi)$ ergibt sich daraus folgend $u_{\text{aus}}(t) \cdot u_{\text{ein}}(t) = \frac{\hat{U}_{\text{aus}} \hat{U}_{\text{ein}}}{2} [\cos(\Delta\varphi) - \cos(2\omega t + \Delta\varphi)]$. Der zweite Term muss durch eine geeignete Filterung entfernt werden, wodurch sich $\overline{u_{\text{aus}}(t) \cdot u_{\text{ein}}(t)} = \frac{\hat{U}_{\text{aus}} \hat{U}_{\text{ein}}}{2} \cos(\Delta\varphi)$ ergibt. Daraus lässt sich die Phasenverschiebung $\Delta\varphi$ bestimmen.
6. Geben Sie den Frequenzgang des Systems als Bode-Diagramm aus, d.h. den Betrag in Dezibel [dB] und die Phasenverschiebung in Grad [°], jeweils über logarithmisch skalierte Frequenzachse (20Hz ... 20kHz).
7. Überprüfen Sie Ihr Ergebnis für mindestens drei (weitestgehend einfache) unterschiedliche Schaltungen, vornehmlich unterschiedlicher Frequenzcharakteristik wie z.B. Tiefpass, Hochpass und Bandpass.²

² Achten Sie bei Verwendung eines Schwingkreises als Bandpass auf eventuelle Spannungsüberhöhungen zur Vermeidung der Beschädigung des Audioeinganges!

2.3 FM-Modulation/Demodulation (Gruppe 3)

- > Erzeugen eines bandbegrenzten Audiosignals $s(t)$ in MATLAB und generieren eines damit frequenzmodulierten Trägersignals $u_{FM}(t)$

$$\omega(t) = \frac{d\varphi(t)}{dt} = K s(t) + \omega_T \dots \text{Momentankreisfrequenz; es muss gelten } K < \omega_T = 2\pi f_T$$

$$\Rightarrow \varphi(t) = \int_{-\infty}^t \omega(\tau) d\tau = K \int_{-\infty}^t s(\tau) d\tau + \omega_T t$$
$$u_{FM}(t) = \sin(\varphi(t))$$

in einem Skript. Das generierte FM-Signal ist in einer Audio-Datei zu speichern.

- > Einlesen des generierten FM-Signals und demodulieren in einem zweiten Skript um das ursprüngliche Audiosignal $s(t)$ zurückzugewinnen:

$$\frac{du_{FM}(t)}{dt} = \cos(\varphi(t)) \cdot \frac{d\varphi(t)}{dt} = \cos(\varphi(t)) \cdot (K s(t) + \omega_T) \dots \text{Information steckt in der Amplitude!}$$

\Rightarrow Amplitudenmessung mittels Hüllkurvenbestimmung über Hilber-Transformation

- > Hilbert-Transformator (LTI-System): $H_H(j\omega) = -j \cdot \text{sign}(\omega)$ ($\text{sign}(\omega)$... Vorzeichenfunktion)

Gehen Sie dabei wie folgt vor:

1. Wählen Sie eine Musik-Audiodatei.
2. Erstellen Sie ein MATLAB-Skript für die folgenden Unterpunkte bzgl. des Modulationsvorganges:
 - a. Entwerfen Sie in MATLAB ein digitales Tiefpassfilter mit einer Sperrgrenze von nicht mehr als 8kHz und Sperrdämpfung von zumindest 50dB. Wählen Sie als Filtertyp *Butterworth* oder *Tschebyscheff* mit geeigneter Durchlassgrenze und Durchlassdämpfung. Übernehmen Sie die Samplingfrequenz aus der Musik-Audiodatei. Unterschiede zum Entwurf von analogen Filtern (vgl. Entwurf aktiver Filter in Schaltungstechnik-Laboratorium) entnehmen Sie der MATLAB-Hilfefunktion zu den entsprechenden Befehlen.
 - b. Filtern Sie damit den Audioinhalt der zuvor gewählten Musikaudio-Datei. Für das Filter kann eine Realisierung wie beim behandelten digitalen RC-Tiefpass-Äquivalent erfolgen oder mit Hilfe der MATLAB-Funktion `filter`. Das gefilterte Musiksignal muss immer noch als solches erkennbar sein.
 - c. Modulieren Sie nach obiger Beschreibung ein sinusförmiges Trägersignal mit Trägerfrequenz $f_T = 8\text{kHz}$ in seiner Frequenz und speichern das Ergebnis als Audiodatei ab. Sie können mit Hilfe der FFT verifizieren, ob keine wesentlichen Spektralanteile nahe bei 20kHz auftreten. Bei der Wiedergabe des frequenzmodulierten Signals ist das ursprüngliche Musiksignal nicht wieder erkennbar!
3. Erstellen Sie ein weiteres MATLAB-Skript für die folgenden Unterpunkte bzgl. des Demodulationsvorganges:
 - a. Lesen Sie aus der zuvor generierten Audiodatei das frequenzmodulierte Trägersignal.
 - b. Bestimmen Sie den Differenzenquotienten des frequenzmodulierten Signals als Näherung für die oben beschriebene Differentiation, um die Momentanfrequenz auf eine Amplitudenmodulation umzuformen.
 - c. Filtern Sie das nunmehr amplitudenmodulierten Signal mittels Hilbert-Transformator (Rechnung mittels MATLAB-Funktion `filter` oder über das Spektrum (FFT)) und berechnen Sie damit die Einhüllende (Hüllkurve). Für weitere Details siehe unterstützende Unterlagen auf Moodle.
 - d. Wählen Sie eine geeignete Skalierung des Ergebnisses vom Hilbert-Transformator und speichern Sie es in einer Audiodatei. Das Abspielen der generierten Audiodatei muss das ursprüngliche Musiksignal wieder erkennen lassen!

2.4 QPSK-Modulation/Demodulation (Gruppe 4)

- > Ein Datenbitstrom wird in Zweiergruppen geteilt, von denen jeweils das erste Bit einen cos-Träger und das zweite Bit einen sin-Träger amplitudenmoduliert, nachdem diese durch ein RRC-Filter gefiltert wurden:

$$s(t) = b_{1,\text{RRC}}(t) \cdot \cos(\omega_T t) + b_{2,\text{RRC}}(t) \cdot \sin(\omega_T t)$$

- > Das Ergebnis wird in einer Audio-Datei als Stereo-Signal abgespeichert, wobei der zweite Kanal der verwendete (unmodulierte) cos-Träger ist. Dieser zweite Kanal erlaubt in der Empfängerverarbeitung eine synchrone Demodulation.
- > In einem zweiten Skript erfolgt die Empfängerverarbeitung, wobei zuallererst die cos- und sin-Träger synchron demoduliert werden. Anschließend erfolgt wieder jeweils eine Filterung mit einem RRC-Filter wie oben. Zusammen ergeben die sequenziellen RRC-Filterungen eine einzelne RC-Filterung im Sinne eines *signalangepassten Filters mit optimaler Filterung des (weißen) Rauschens*. Die Peaks in den Ausgangssignalen repräsentieren die logischen 1.

Gehen Sie dabei wie folgt vor:

- Erstellen Sie ein MATLAB-Skript für die folgenden Unterpunkte bzgl. der Modulation:
 - Erzeugen Sie eine Binärsequenz (z.B. aus dem ASCII-Code eines Textes) $\{b_i\}$ mit $i \in \mathbb{N}_0$, in der logische Nullen durch den Wert -1 repräsentiert werden (z.B. $[-1,1,-1,-1,1,-1,1,1]$ für die Binärsequenz $[0,1,0,0,1,0,1,1]$).
 - Teilen Sie den Binärstrom $\{b_i\}$ mit $i \in \mathbb{N}_0$ in zwei Teilströme $\{b_{2i}\}$ und $\{b_{2i+1}\}$, welche nur die geraden bzw. ungeraden Bits enthalten.
 - Die Bitwerte $b_{2i/2i+1} = \pm 1$ eines jeden Teilstromes werden jeweils in einem für den jeweiligen Teilstrom zuständigen Root-Raised-Cosine-Filter mit Impulsantwort

$$h_{\text{RRC}}[k] = \frac{4\rho}{4\rho + \pi(1-\rho)} \cdot \frac{\cos\left(\pi(1+\rho)\frac{kT_s}{T_{\text{bit}}}\right) + \frac{\sin\left(\pi(1-\rho)\frac{kT_s}{T_{\text{bit}}}\right)}{\frac{4\rho kT_s}{T_{\text{bit}}}}}{1 - \left(\frac{4\rho kT_s}{T_{\text{bit}}}\right)^2}$$

gefiltert. Verwenden Sie dabei folgende Parameterwerte:

- Roll-off-Faktor $\rho = 0.7$
- Abtastfrequenz $f_s = 48\text{kHz}$
- Bitrate (eines Teilstromes) $\frac{1}{T_{\text{bit}}} = 1\text{ kbit/s}$
- Zeitliche Beschränkung von $h_{\text{RRC}}[k]$ auf $-\frac{N}{2} \leq k \leq \left(\frac{N}{2} - 1\right)$ mit $N = 6 \cdot \frac{T_{\text{bit}}}{T_s}$

Die Ausgangssignale der beiden RRC-Filter seien $b_{1,\text{RRC}}[k]$ (gerade Bits) und $b_{2,\text{RRC}}[k]$ (ungerade Bits).

- Erzeugen Sie das QPSK-modulierte Sendesignal $s[k] = b_{1,\text{RRC}}[k] \cdot \cos(\omega_T kT_s) + b_{2,\text{RRC}}[k] \cdot \sin(\omega_T kT_s)$ mit einer Trägerfrequenz $f_T = 8\text{kHz}$ und speichern Sie dieses in einem Kanal einer Stereo-Audiodatei. Im zweiten Stereokanal wird der unmodulierte cos-Träger gespeichert. Achten Sie auf eine geeignete Skalierung der Audiosignale vor dem Schreiben in die Audiodatei.
- Erstellen Sie ein weiteres MATLAB-Skript für die folgenden Unterpunkte bzgl. der Demodulation:
 - Lesen Sie aus der vorherigen Audiodatei sowohl den cos-Träger als auch das modulierte Datensignal.
 - Erzeugen Sie aus dem cos-Träger einen dazu passenden sin-Träger.
 - Multiplizieren Sie das Datensignal einmal mit dem cos-Träger (um die Bits $\{b_{2i}\}$ zu erhalten) und einmal mit dem sin-Träger (um die Bits $\{b_{2i+1}\}$ zu erhalten). Die beiden

erhaltenen Signale werden anschließend für sich jeweils wieder durch ein eigenes RRC-Filter (siehe oben) gefiltert.

- d. In den Ausgangssignalen der beiden RRC-Filter treten Peaks auf, deren Vorzeichen den jeweiligen ursprünglichen Bits $\{b_{2i}\}$ bzw. $\{b_{2i+1}\}$ entsprechen.
- e. Ordnen Sie die erhaltenen Bits wieder abwechselnd (gerade/ungerade/gerade/...) an, um die ursprüngliche Binärsequenz zu erhalten. Speichern Sie das Ergebnis geeignet ab, damit ein einfacher Vergleich mit der ursprünglichen Binärsequenz möglich ist.