

## Aufgabenblatt 1

Das erste Aufgabenblatt übt das Arbeiten mit Signalen. Im einzelnen wird geklärt, wie man Berechnungen durchführt, wie ein Graph visualisiert wird, wie einfache Skripte programmiert werden. Dabei sollen insbesondere folgende Schlüsselworte und Funktionen verwendet und erlernt werden:

function, sin, cos, round  
fplot, plot, stem, hold, title, legend, xlabel, ylabel,  
audioplayer, playblocking, sound

**Wichtig:** Geben Sie bitte nur eine Gesamtlösung für jede Aufgabe 1.1, 1.2 usw. ab. Lösungen für die Teilaufgaben a, b usw. sind nicht erwünscht. Am besten geben Sie ein einzelnes Matlab-Skript für das Aufgabenblatt 1 ab und verwenden Sektionen für jede einzelne Aufgabe 1.1, 1.2 usw. (dies ist nur in Matlab und nicht in Octave möglich).

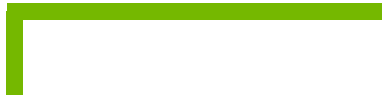
Jede Gruppe (mit maximal zwei Studierenden) gibt die Lösungen bitte gemeinsam ab (also nur einmal). Vergessen Sie nicht, die Namen und Matrikel-Nummern aller an der Lösung beteiligten Studierenden als Kommentar in den ersten Zeilen jedes Matlab-Skripts einzufügen.

### 1.1 Darstellung eines Funktionsgraphen

- Schreiben Sie ein Skript, mit dem sich ein Sinussignal (*sin*) mit einer Frequenz von 2Hz über eine Sekunde darstellen lässt. Verwenden Sie hierzu die Funktion *fplot*.
- Ergänzen Sie das Skript so, dass Sie in demselben Graphen ein zweites Sinussignal ebenfalls mit der Frequenz 2Hz jedoch mit einer Phasenverschiebung von 30° ausgeben. Das zweite Signal soll rot und gestrichelt dargestellt werden. Recherchieren Sie ggf. im Internet, wie der Begriff der Phasenverschiebung definiert ist.
- Wird durch die Phasenverschiebung das Signal nach links oder nach rechts verschoben? Fügen Sie die Antwort als Kommentar in das Skript ein.
- Ergänzen Sie den Graphen um einen Titel, eine Legende und Aussagekräftige Achsenbeschriftungen.

### 1.2 Funktionsdarstellung mit Wertetabelle

Bei der zeitdiskreten Verarbeitung von Signalen stehen nicht mehr alle Funktionswerte zur Verfügung, sondern nur noch die zu bestimmten Zeitpunkten abgetasteten Werte. Um dies in Matlab darstellerisch zu erfassen,



arbeitet man mit Vektoren von Abtastwerten, die man mit Hilfe der Funktionen *plot* oder *stem* anzeigen kann.

- a) Schreiben Sie ein Skript, mit dem Sie als Referenz zunächst ein Sinussignal (*sin*) mit einer Frequenz von 2Hz und einer Amplitude von 2 (Einheitenlos) über eine Sekunde darstellen (in blauer Farbe). Verwenden Sie hierzu die Funktion *fplot* (entspricht weitgehend der Aufgabe 1.1a).
- b) Simulieren Sie nun eine Abtastung des Sinussignals. Erzeugen Sie hierzu einen Vektor *sample* mit 17 Sinuswerten. Der erste und letzte Eintrag des Vektors entsprechen den Funktionswerten des Sinussignals zu den Zeitpunkten  $t=0s$  und  $t=1s$ .

Erweitern Sie das zuvor geschriebene Skript. Stellen Sie den erzeugten Vektor mit Hilfe der Funktion *plot* über eine Sekunde dar (als grün gestrichelte Linie). Die Ausgabe soll zusammen mit dem in Teilaufgabe a erzeugten Graphen angezeigt werden.

- c) Die Darstellung von Abtastwerten mit Hilfe der Funktion *plot* lässt vermuten, dass die zwischen den Abtastwerten interpolierten Funktionswerte der realen Funktion nahe kommen, was jedoch falsch sein kann. Besser ist es deshalb, die Funktion *stem* zur Darstellung von Abtastwerten zu verwenden.

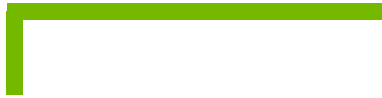
Wiederholen Sie deshalb die vorangehende Aufgabe, indem Sie nun die Funktion *stem* zum Einsatz bringen (Ausgabe als durchgezogene schwarze Linien). Die Ausgaben aus den vorangehenden Teilaufgabe sollen dabei erhalten bleiben.

- d) Ergänzen Sie den Graphen um einen Titel, eine Legende und Aussagekräftige Achsenbeschriftungen.
- e) Mit welcher Abtastfrequenz wird in dieser Aufgabe gearbeitet? Ergänzen Sie die Antwort als Kommentar in Ihrem Skript.

### 1.3 Quantisierung

Mit der vorangehenden Aufgabe wurde eine zeitdiskrete Erfassung eines zeitkontinuierlichen Signals in Matlab umgesetzt. Dabei wurden die Funktionswerte des Originalsignals exakt berechnet. In der Realität wird das analoge Eingangssignal jedoch einer AD-Wandlung unterzogen und dabei quantisiert. Im Prinzip werden die Abtastwerte auf vorgegebene diskrete Werte gerundet, was sich in Matlab mit Hilfe der Funktion *round* umsetzen lässt.

- a) Schreiben Sie ein Skript, mit dem Sie als Referenz zunächst ein Sinussignal (*sin*) mit einer Frequenz von 2Hz und einer Amplitude von 2 (Einheitenlos) über eine Sekunde darstellen (in blauer Farbe). Verwen-



den Sie hierzu erneut die Funktion `fplot` (entspricht weitgehend der Aufgabe 1.1a).

- b) Erzeugen Sie einen Vektor mit 17 Abtastwerten entsprechend der Aufgabe 1.2b. Diesmal runden Sie die Amplituden jedoch auf die Werte 0,  $\pm 1$ ,  $\pm 2$ .

Erweitern Sie das zuvor geschriebene Skript, indem Sie die gerundeten Werte mit Hilfe der Funktion `stem` in dem selben Graphen darstellen, in dem auch die Sinusfunktion als Referenz angezeigt wird (Teilaufgabe a). Die gerundeten Werte sollen schwarz angezeigt werden.

- c) Stellen Sie zusätzlich mit Hilfe der Funktion `stem` den Fehler dar, der durch die Quantisierung entsteht (in rot).
- d) Ergänzen Sie den Graphen um einen Titel, eine Legende und Aussagekräftige Achsenbeschriftungen.

#### 1.4 N-Bit Quantisierung

- a) Schreiben Sie eine Funktion mit der sich ein Eingangswert  $x$  quantisieren lässt. Der Einfachheit halber dürfen Sie davon ausgehen, dass das abgetastete, nicht quantisierte Eingangssignal eine Amplitude von 1 aufweist (der Maximalwert ist also  $+1$ , der Minimalwert  $-1$ ). Die Anzahl der Quantisierungsstufen soll durch den Parameter `level` einstellbar sein und sowohl im positiven, als auch im negativen Bereich gleich sein (`level = 3` bedeutet also 3 Stufen im neg. Bereich und 3 Stufen im pos. Bereich).

Der Funktionskopf soll folgendermaßen formuliert werden:

**function** `y = quantize (x, level)`

- b) Ergänzen Sie ein Skript, mit dem Sie als Referenz zunächst ein Sinussignal (`sin`) mit einer Frequenz von 2Hz und einer Amplitude von 1 (Einheitenlos) über eine Sekunde ausgeben (blauer). Verwenden Sie hierzu erneut die Funktion `fplot` (entspricht weitgehend der Aufgabe 1.1a).
- c) Das Sinussignal soll als nächstes mit 16 Hz abgetastet und mit einem 3 Bit Wandler quantisiert werden. Nutzen Sie hierzu die von Ihnen geschriebene Funktion `quantize`. Geben Sie das Ergebnis der Quantisierung (schwarz) und den Fehler (rot) zusammen mit dem in Teilaufgabe b erzeugten Referenzsignal aus. Verwenden Sie hierzu die Funktion `stem`.
- d) Ergänzen Sie den Graphen um einen Titel, eine Legende und Aussagekräftige Achsenbeschriftungen.



- e) Interpretieren Sie das Ergebnis im Vergleich zu dem Ergebnis aus Aufgabe 1.3c. Ergänzen Sie Ihre Interpretation als Kommentar in Ihrem Skript.

## 1.5 Audioausgabe

- a) Schreiben Sie ein Skript, in dem Sie ein 500Hz Sinussignal mit der Amplitude 1 für eine Dauer von 1 Sekunden erzeugen. Das Signal soll mit 44100Hz abgetastet werden.

*Hinweis: Um ein Audiosignal  $x$  mit der Abtastfrequenz  $f_s$  auszugeben, können Sie folgenden Aufruf verwenden:*

*`playblocking (audioplayer (x, fs));`*

*Die ebenfalls verfügbare Funktion `sound` ist veraltet und sollte nicht mehr verwendet werden.*

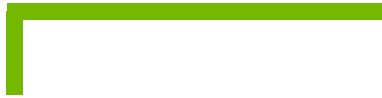
- b) Erweitern Sie das Skript, indem Sie die zuvor geschriebene Funktion *quantize* verwenden und damit eine AD-Wandlung mit 3 Bit simulieren. Geben Sie zusätzlich zu dem Original das quantisierte Signal über die Soundkarte aus (d.h. zuerst das Originalsignal und dann das quantisierte Signal). Wie ändert sich der Klang?

*Anmerkung: Die Funktion `Quantize` wurde von uns so realisiert, dass die Anzahl der Stufen für positive und negative Werte vorgegeben wird. Die 0 zählt dabei allerdings sowohl für positive als auch für negative Werte. Diesen Effekt ignorieren wir hier der Einfachheit halber. Falls wir z.B. in `quantize` den Parameter `level` gleich 16 vorgeben, gehen wir davon aus, dass wir das Signal in  $2 \times 16 = 32$  Stufen quantisieren, was einem 5 Bit Wandler entsprechen würde.*

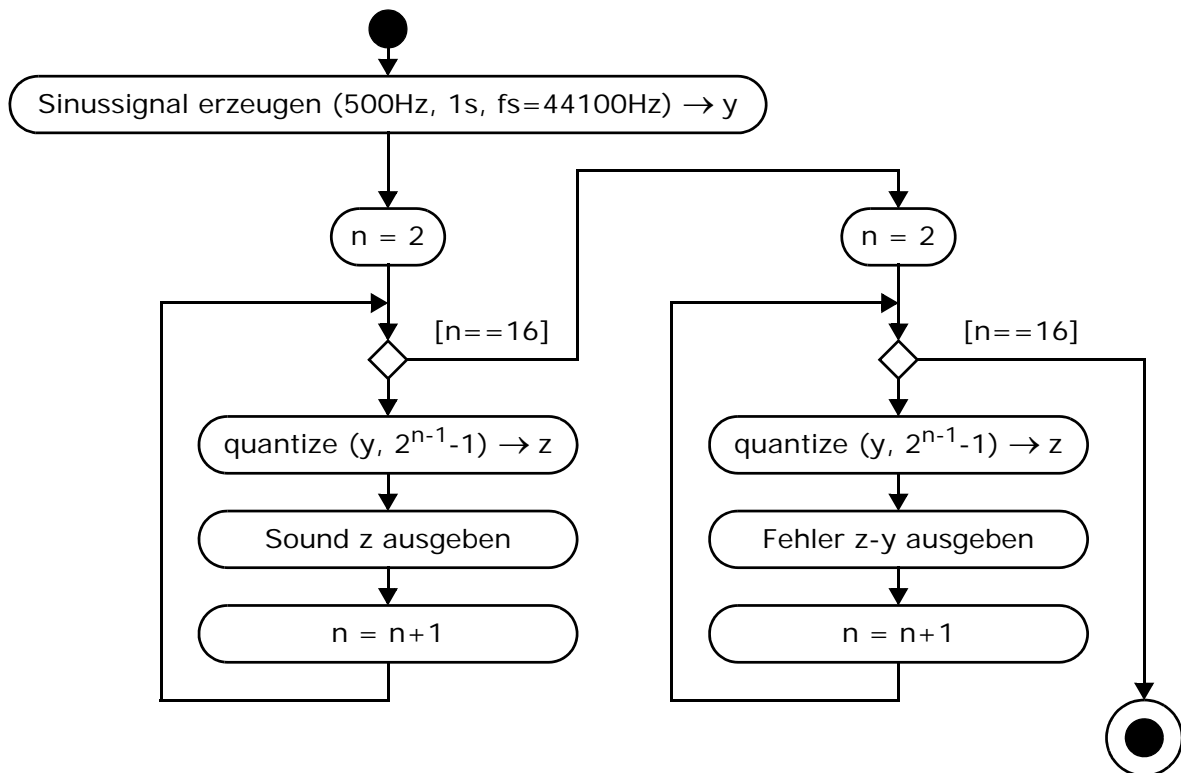
- c) Erweitern Sie das Skript erneut und geben Sie zusätzlich den durch die Quantisierung entstehenden Fehler über die Soundkarte aus. Warum lässt sich die Frequenz des Originalsignals in dem Fehler zu hören?

## 1.6 Vergleichsausgabe

Schreiben Sie ein Skript um die Wirkung unterschiedlicher Bitbreiten von AD-Wandlern zu verdeutlichen. Erweitern Sie hierzu die vorangehende Aufgabe, indem Sie das 500Hz Sinussignal in einer Schleife mit 1 Bit, 12 Bit, 3 Bit ... 10 Bit quantisieren (unter Nutzung der Funktion *quantize*) und jeweils für eine Sekunde ausgegeben. In einer zweiten Schleife wiederholen Sie den Vorgang, wobei Sie jeweils den Quantisierungsfehler ausgeben.



Das für die Umsetzung hilfreiche Aktivitätsdiagramm ist nachfolgend dargestellt:



*Hinweis: Bewertet werden neben der Funktion auch, inwieweit das Aktivitätsdiagramm umgesetzt und wie sauber programmiert wurde (Kommentare, Einrückung, usw.).*