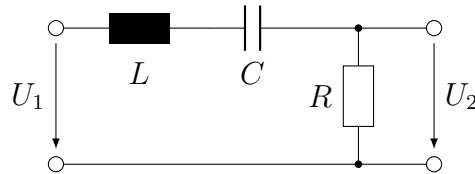


2. Aufgabe: Schwingkreis



1 Einleitung

Als angehende Ingenieure werden Ihnen immer wieder Schwingkreise begegnen. Aufgabe dieses Praktikums ist es, einen seriellen Schwingkreis mit einem Programm zu analysieren. Ihr Programm soll den Frequenzgang für Real- und Imaginärteil, Amplitude und Phase berechnen und als Tabelle in eine Datei im CSV-Format ausgeben.

1.1 Die Übertragungsfunktion

Die Übertragungsfunktion des oben gezeigten Schwingkreises für Sinussignale lautet:

$$\frac{U_2}{U_1} = \frac{R}{j\omega L + \frac{1}{j\omega C} + R}$$

Zur Lösung der Praktikumsaufgabe werden folgende Gleichungen benötigt. Da Sie Mathematik und Elektrotechnik in anderen Vorlesungen hören, werden hier nur die Ergebnisse angegeben, wobei ω für die Kreisfrequenz steht, welche sich aus der Frequenz mal 2π ergibt.

$$\begin{aligned} \operatorname{Re}\left(\frac{U_2}{U_1}\right) &= \frac{(\omega RC)^2}{(1 - \omega^2 LC)^2 + (\omega RC)^2} & \operatorname{Im}\left(\frac{U_2}{U_1}\right) &= \frac{\omega RC(1 - \omega^2 LC)}{(1 - \omega^2 LC)^2 + (\omega RC)^2} \\ \left|\frac{U_2}{U_1}\right| &= \sqrt{\operatorname{Re}\left(\frac{U_2}{U_1}\right)^2 + \operatorname{Im}\left(\frac{U_2}{U_1}\right)^2} & \varphi\left(\frac{U_2}{U_1}\right) &= \arctan\left(\frac{\operatorname{Im}\left(\frac{U_2}{U_1}\right)}{\operatorname{Re}\left(\frac{U_2}{U_1}\right)}\right) \end{aligned}$$

Für die Berechnung benötigen Sie einige Funktionen, die Sie mit `math.h` einbinden können:

| | |
|---------------------------|--|
| <code>pow(x, y);</code> | Potenz x^y |
| <code>sqrt(x);</code> | Quadratwurzel |
| <code>atan(x);</code> | arctan von x |
| <code>atan2(y, x);</code> | Winkel zwischen der Geraden $0/0 \rightarrow x/y$ und der positiven x -Achse |
| <code>exp(x);</code> | Exponentialfunktion |
| <code>log(x);</code> | natürlicher Logarithmus (zur Basis $e = 2,71828\dots$) |
| <code>log10(x);</code> | Logarithmus zur Basis 10 |

In C werden alle Winkel in Bogenmaß angegeben. Wenn Sie vor dem Einbinden von `math.h` das Marko `_USE_MATH_DEFINES` definieren, steht Ihnen mit `M_PI` die Konstante π zur Verfügung.

1.2 Dateien im CSV-Format

CSV steht für *comma separated values*, zu deutsch *durch Komma getrennte Werte*. Alle Einträge einer Zeile werden durch Kommata getrennt. Eine neue Zeile wird durch einen Zeilenumbruch

'\n' markiert. Mit der Endung .csv kann die Datei direkt von Programmen wie *MS-Excel* oder *OpenOffice Calc* eingelesen werden.

Damit die CSV-Dateien unter einer deutschen Installation von Windows korrekt eingelesen werden können, müssen Sie folgende Einstellungen ändern: Unter '*Systemsteuerung*' wählen Sie die Option '*Zeit, Sprache und Region*', dann '*Datums-, Uhrzeit- oder Zahlenformat ändern*' und schließlich die Taste '*Weitere Einstellungen...*'. Dort nehmen Sie folgende Einstellungen vor:

- *Dezimaltrennzeichen* von Komma auf Punkt ändern
- *Symbol für Zifferngruppierung* von Punkt auf Komma ändern
- *Listentrennzeichen* von Semikolon auf Komma ändern

Von nun an müssen Sie auch in vielen anderen Programmen statt dem Komma einen Punkt als Dezimaltrennzeichen eingeben.

1.3 Aufbau des Programms

Das Programm soll mit zwei Strukturen arbeiten: eine für die Eingabeparameter und eine für die Ergebnisse der Berechnung. Eine erste Funktion fragt alle Parameter vom Benutzer ab, eine zweite Funktion gibt diese auf dem Bildschirm aus, eine dritte berechnet die Übertragungsfunktion des Schwingkreises und eine vierte Funktion schreibt die Ergebnisse in eine CSV-Datei. Das ganze Programm soll über ein Menü gesteuert werden.

Für die Eingabeparameter wird im Hauptprogramm eine Variable angelegt, während für die Ergebnisse zur ein Zeiger angelegt wird. Die Funktion zur Berechnung der Übertragungsfunktion reserviert Speicher und gibt die Adresse des reservierten Speichers gefüllt mit den Ergebnissen der Berechnung zurück. Das Hauptprogramm muss sich um die Freigabe des Speichers kümmern.

2 Obligatorische Aufgaben

2.1 Strukturen und Datentypen

Erstellen Sie zwei neue Strukturen:

1. Eine Struktur für alle Eingabeparameter: Widerstand R , Kapazität C , Induktivität L , Startfrequenz f_1 , Endfrequenz f_2 (alle als *double*) und die Anzahl der Frequenzschritte n (diese als *int*).
2. Eine Struktur für die Ergebnisse: Frequenz, Real-, Imaginärteil, Amplitude, Phase (alle als *double*)

2.2 Abfrage der Eingabeparameter

Erstellen Sie eine Funktion, die alle Eingabeparameter vom Benutzer abfragt. Die Funktion soll *getParameter* heißen, als einzigen Parameter einen Zeiger auf die Eingabestruktur erwarten und keinen Wert zurückgeben. Die Funktion soll alle Fehleingaben des Benutzers mit passenden Fehlermeldungen abfangen. Wird nur die Eingabetaste betätigt, soll der vorige Wert übernommen werden. Legen Sie sinnvolle Maximal- und Minimalwerte für die Parameter fest.

2.3 Ausgabe der Eingabeparameter

Erstellen Sie eine Funktion mit Namen *plotParameter*, welche die eingegebenen Parameter ordentlich auf dem Bildschirm ausgibt.

2.4 Berechnung des Schwingkreises

Erstellen Sie eine Funktion mit Namen *evaluateResonator* zur Berechnung des Schwingkreises. Die Funktion erwartet als Parameter die Struktur der Eingabeparameter und gibt einen Zeiger auf die Ausgabestruktur zurück.

Die Funktion reserviert dynamisch Speicher für die Ergebnisse (Anzahl der Frequenzschritte plus eins), berechnet die Frequenzen in logarithmischer Einteilung und bestimmt für alle Frequenzen Real- und Imaginärteil, sowie Amplitude und Phase der Übertragungsfunktion. Die Funktion gibt die Adresse des dynamisch reservierten Vektors zurück.

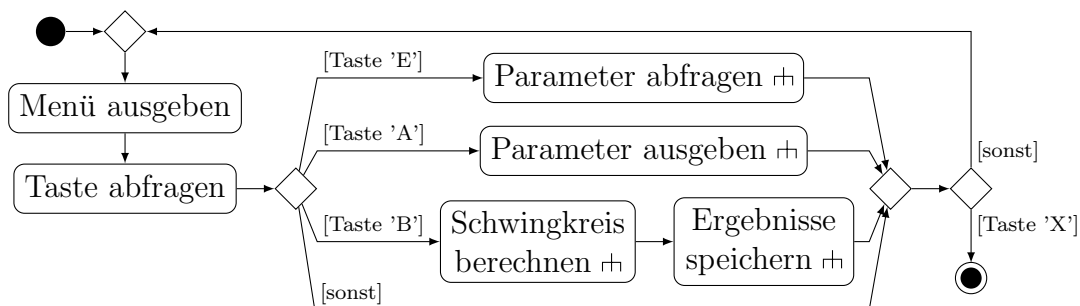
2.5 Ausgabe der Übertragungsfunktion in eine CSV-Datei

Erstellen Sie eine Funktion mit Namen *writeResonator* zum Schreiben der Übertragungsfunktion in eine CSV-Datei. Die Funktion erhält als Parameter den Dateinamen, die Struktur mit den Eingabeparametern und einen Zeiger auf die berechneten Ergebnisse.

In den ersten drei Zeilen stehen die Werte des Schwingkreises, danach folgt eine Zeile mit den Spaltenüberschriften und eine weitere Zeile mit den Einheiten der Spalten. Die Tabelle soll fünf Spalten haben: Frequenz, Realteil, Imaginärteil, Amplitude und Phase. Die Phase soll in Grad ($360 \text{ deg} = 2\pi \text{ rad}$) angegeben werden. Verwenden Sie für die Ausgabe aller *double*-Variablen den Platzhalter `%lg`.

2.6 Steuerung des Programms über ein Menü

Erstellen Sie ein Hauptprogramm zur Steuerung der zuvor erstellten Funktionen. Implementieren Sie dabei das folgende Aktivitätsdiagramm:



Denken Sie daran, den dynamisch reservierten Speicher wieder freizugeben. Auch wenn mehrfach die Taste *B* zur Berechnung der Übertragungsfunktion betätigt wird, muss der zuvor reservierte Speicher zunächst wieder freigegeben werden.

2.7 Test des Programms

Verwenden Sie zum Testen folgende Parameter:

$$L = 15 \text{ mH}, \quad C = 1,8 \mu\text{F}, \quad R = 100 \Omega, \quad f_1 = 200 \text{ Hz}, \quad f_2 = 5 \text{ kHz}, \quad n = 6$$

Es sollte sich von den Werten her folgende CSV-Datei ergeben:

Widerstandswert,100,Ohm
Induktionswert,0.015,H
Kapazitätswert,1.8e-06,F
Frequenz,Realteil,Imaginärteil,Amplitude,Phase
(Hz),(-),(-),(-),(deg)
200,0.0528713,0.223777,0.229938,76.7066
341.995,0.163358,0.369692,0.404176,66.1605
584.804,0.519991,0.4996,0.721104,43.8543
1000,0.996615,-0.0580863,0.998306,-3.33563
1709.98,0.454958,-0.497967,0.674506,-47.5842
2924.02,0.142464,-0.349525,0.377444,-67.8246
5000,0.0463581,-0.210259,0.215309,-77.5663

3 Weitere, optionale Schritte

3.1 Schaltplan anzeigen

Bauen Sie in das Hauptmenü eine Option ein, die den Schaltplan auf dem Bildschirm anzeigt. Verwenden Sie hierfür einfache Textzeichen und die Funktion printf.

3.2 Parameter binär speichern

Erstellen Sie jeweils eine Funktion zum Schreiben und Lesen der Parameter (*writeParam* und *readParam*). Am Ende des Programms werden die Parameter gespeichert und beim erneuten Programmstart wieder eingelesen. Existiert beim ersten Programmstart noch keine entsprechende Datei, so werden die Beispielwerte aus Abschnitt 2.7 als Standard verwendet.

Viel Erfolg beim Programmieren!