Karlsruher Institut für Technologie
Institut für Biomedizinische Technik

Prof. Dr. rer. nat. O. Dössel	M.Sc. Michael Kircher	M.Sc. Laura Anna Unger
Geb. 30.33, Raum 515	Geb. 30.33, Raum 410	Geb. 30.33, Raum 413.1
Tel.: 0721 / 608-42650	Tel.: 0721 / 608-42751	Tel.: 0721 / 608-48232

Lineare Elektrische Netze
WS 2017/2018

MATLAB-Aufgabe

Die maximale Punktzahl dieser Aufgabe entspricht 3% der Gesamtpunktzahl des Leistungsteils im Fach Lineare Elektrische Netze. Die in dieser MATLAB-Aufgabe erzielten Punkte sind ausschließlich im aktuellen Wintersemester (2017/18) sowie im darauffolgenden Sommersemester (2018) gültig.

Sie erhalten die MATLAB-Aufgabe am Montag, den 13. November 2017 als Download in ILIAS. Der Bearbeitungszeitraum endet am Freitag, den 24. November 2017 um 09:30 Uhr. ***Später eintreffende Aufgaben werden nicht gewertet! Die Eidesstattliche Erklärung muss unbedingt unterschrieben werden!*** *Die Aufgaben können am 24. November 2017 im Anschluss an die LEN-Vorlesung im Hörsaal abgegeben werden. Die Dokumente müssen* ***geheftet*** *(tackern genügt) und mit Deckblatt (Seite 1 enthält* ***Namen, Matrikelnummer, KIT-Account und Eidesstattliche Erklärung****) versehen sein. Wir werden am genannten Termin keinen Tacker bereitstellen.*

Zur Bearbeitung der Aufgaben können die Rechner im Steinbuch Center for Computing (SCC) verwendet werden. Für Studenten der Vorlesung LEN sind der A- und C-Pool des SCC montags von 13:00 Uhr bis 15:30 Uhr und der A-, H- und I-Pool mittwochs von 08:45 Uhr bis 11:15 Uhr reserviert. Zu allen anderen Zeiten können nicht besetzte Rechner im SCC ebenfalls verwendet werden. Die MATLAB-Software kann, muss aber nicht auf den Rechnern in anderen Poolräumen installiert sein.

Am Mittwoch, den 15. November 2017 findet um 14:00 Uhr eine MATLAB-Übung im Daimler-Hörsaal statt. Dort wird eine grundlegende Einführung in MATLAB gegeben. Am Montag, den 20. November 2017 stehen zwischen 13:00 Uhr und 15:30 Uhr in den Poolräumen A und C Tutoren für offene Fragen und Hilfe zur Verfügung. Am Mittwoch, den 22. November 2017 stehen zwischen 08:45 Uhr und 11:15 Uhr in den Poolräumen A, H und I Tutoren für offene Fragen und Hilfe zur Verfügung.

Einführung in die Aufgabenstellung

Gegeben seien die in Abbildung 1 dargestellten Schaltungen: eine Spannungsquelle Q1, eine Beschaltung B1 sowie eine Beschaltung B2. In Aufgabe 1 wird die Spannungsquelle Q1 mit einem Widerstandsnetzwerk und einer weiteren Spannungsquelle (Beschaltung B1) beschaltet. Aufgabe 2 befasst sich mit der Beschaltung durch einen temperaturabhängigen Widerstand R_T (Beschaltung B2).

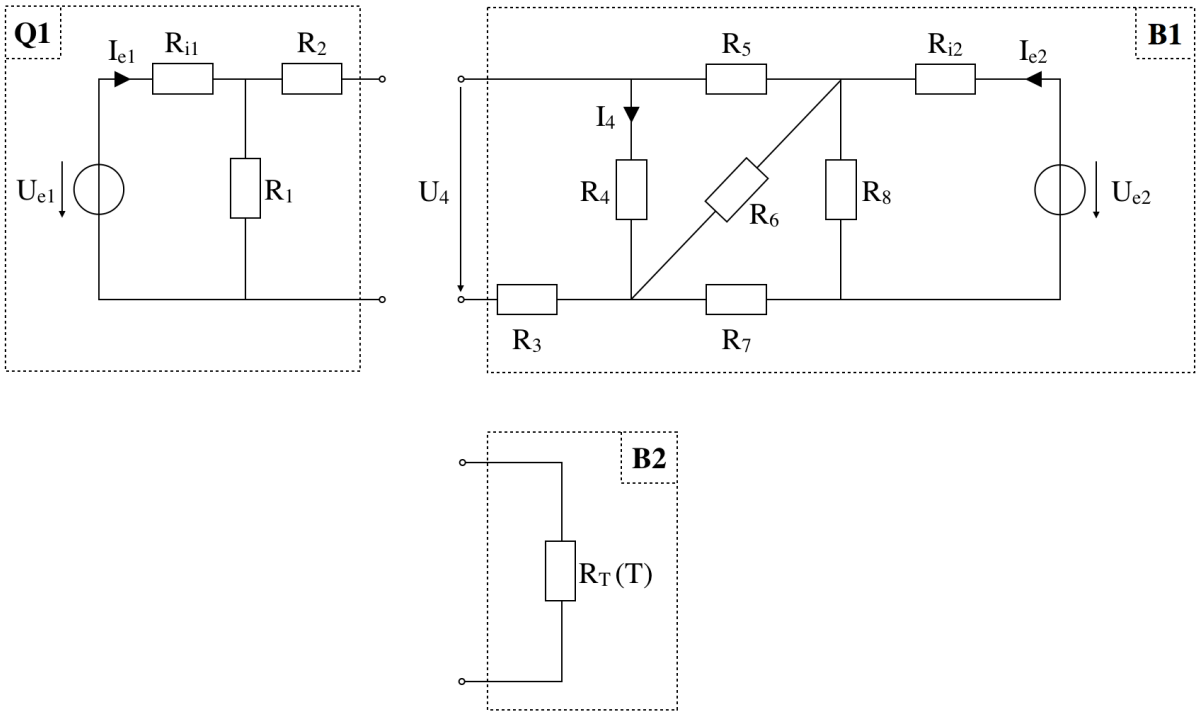

Abbildung 1: Die Spannungsquelle Q1 wird mit Schaltung B1 (Aufgabe 1) bzw. mit Schaltung B2 (Aufgabe 2) beschaltet.

Die Kenngrößen der einzelnen Bauteile wurden den Datenblättern entnommen und sind in Tabelle 1 und Tabelle 2 zusammengefasst.

U_{e1}	U_{e2}	R_{i1}	R_{i2}	R_1	R_2	R_3	R_4	R_5	R_6	R_7	R_8
12 V	3 V	2 Ω	1 Ω	7 Ω	9 Ω	1 Ω	8 Ω	8 Ω	13 Ω	6 Ω	2 Ω

Tabelle 1: Kenngrößen der Bauteile aus den Schaltungsteilen Q1 und B1.

T / °C	0	5	10	15	19	22	28	31	35	40	45	50	60
R_T / Ω	5	5	5	4	3	2	2	3	5	8	10	12	14

Tabelle 2: Auszug aus dem Datenblatt des temperaturabhängigen Widerstandes R_T.

Aufgabe 1

Die Spannungsquelle Q1 wird mit Schaltung B1 beschaltet (siehe Abb. 2).

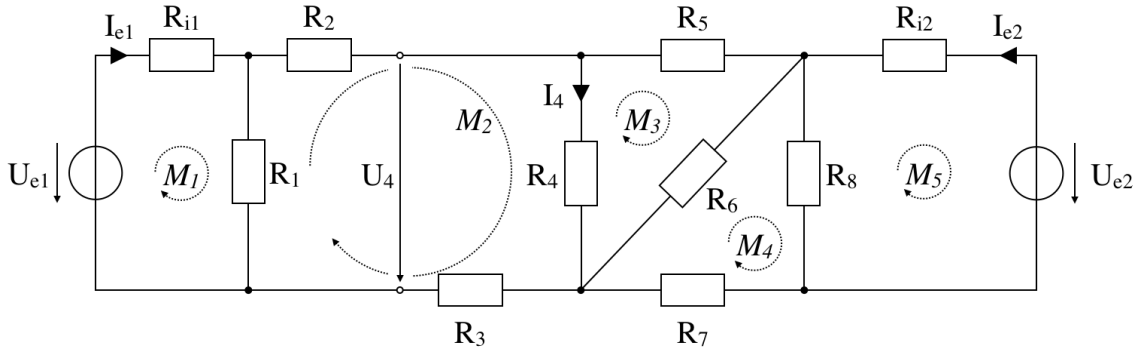

Abbildung 2: Die Spannungsquelle Q1 wird mit Schaltung B1 beschaltet. Die Maschen M_1 bis M_5 sind im mathematisch negativen Drehsinn definiert.

Es sollen die Ströme I_{e1}, I_{e2} und I_4 sowie die Spannung U_4 bestimmt werden. Bestimmen Sie zunächst alle Maschenströme mit dem formalisierten Maschenstromverfahren. Verwenden Sie hierzu MATLAB. Gehen Sie wie folgt vor:

(a) Schreiben Sie in MATLAB eine Funktion *cramerscheRegel*, die ein lineares Gleichungssystem $\boldsymbol{A} \cdot \boldsymbol{x} = \boldsymbol{b}$ mit der Cramerschen Regel löst. Hinweise:

- Die Funktion soll eine Matrix $\boldsymbol{A}$ und einen Vektor $\boldsymbol{b}$ als Argumente annehmen.
- Die Funktion soll den Vektor $\boldsymbol{x}$ zurückgeben.
- Die Funktion soll überprüfen, ob die Matrix $\boldsymbol{A}$ quadratisch ist und ob der Vektor $\boldsymbol{b}$ die richtige Länge hat. Anderenfalls soll eine entsprechende Fehlermeldung zurückgegeben werden.
- Die Implementierung soll mit nur einer for-Schleife erfolgen.
- Die Funktion soll beim Aufruf der zugehörigen Hilfe den folgenden Anzeigetext bereitstellen:

```
Funktion cramerscheRegel
Autor            :  - Ihr Name -
Erstellungsdatum :  - Erstellungsdatum -
Beschreibung     :  Die Funktion cramerscheRegel löst ein lineares
                    Gleichungssystem der Form Ax = b mit der
                    Cramerschen Regel.
Input            :  Matrix A (Dimension m x m)
                    Vektor b (Dimension m x 1)
Output           :  Vektor x (Dimension m x 1)
```

- Versehen Sie die wesentlichen Codeabschnitte mit erläuternden Kommentaren.
- Nehmen Sie den entstandenen MATLAB-Code in Ihre Dokumentation auf.
- Nehmen Sie einen Screenshot des Command-Windows nach Aufruf der Hilfe-Funktion in Ihre Dokumentation auf.

- Rufen Sie die Funktion mit unzulässigen Übergabeparametern (z.B. eine nicht quadratische Matrix $\boldsymbol{A}$ oder ein Vektor $\boldsymbol{b}$ von unpassender Länge) auf und dokumentieren Sie die Ausgabe der entsprechenden Fehlermeldung mit einem Screenshot des Command-Windows.

(b) Schreiben Sie in MATLAB eine Funktion *berechneMaschenstroeme*, die das formalisierte Maschenstromverfahren für das Netzwerk aus Abbildung 2 mit Hilfe der Cramerschen Regel durchführt und die Maschenströme zurückgibt. Hinweise:

- Die Funktion soll einen Vektor von Widerstandswerten $\boldsymbol{R} = [R_1, R_2, ..., R_8]$, den Widerstandswert des Innenwiderstands R_{i1}, den Widerstandswert des Innenwiderstands R_{i2} sowie die Spannungswerte U_{e1} und U_{e2} als Argumente annehmen.
- Die Funktion soll die Maschenströme I_1^M bis I_5^M als Vektor $\boldsymbol{IM} = [I_1^M, I_2^M, ..., I_5^M]$ zurückgeben.
- Nutzen Sie die in Abbildung 2 eingezeichneten Maschen M_1 bis M_5.
- Nutzen Sie die in Aufgabenteil (a) erstellte Funktion *cramerscheRegel* zur Berechnung der Maschenströme.
- Erstellen Sie angelehnt an Aufgabenteil (a) eine Ausgabe für den Aufruf der Hilfe dieser Funktion.
- Versehen Sie die wesentlichen Codeabschnitte mit erläuternden Kommentaren.
- Nehmen Sie den entstandenen MATLAB-Code in Ihre Dokumentation auf.
- Nehmen Sie einen Screenshot des Command-Windows nach Aufruf der Hilfe-Funktion in Ihre Dokumentation auf.

Übergeben Sie Ihrer Funktion die in Tabelle 1 gegebenen Widerstands- und Spannungswerte. Dokumentieren Sie die berechneten Maschenströme.

(c) Berechnen Sie nun die Ströme I_{e1}, I_{e2} und I_4 sowie die Spannung U_4 mit Hilfe der in Aufgabenteil (b) bestimmten Maschenströme. Schreiben Sie hierfür ein MATLAB-Skript *Aufgabe1c*, welches die Funktion *berechneMaschenstroeme* aufruft, die gesuchten Ströme daraus berechnet und diese anschließend im Command-Window anzeigt.

- Die Ausgabe der Ergebnisse im Command-Window soll in der Form "*Variablenname = Zahlenwert Einheit*" erfolgen.
- Versehen Sie die wesentlichen Codeabschnitte mit erläuternden Kommentaren.
- Nehmen Sie den entstandenen MATLAB-Code in Ihre Dokumentation auf.
- Nehmen Sie einen Screenshot des Command-Windows mit der Ausgabe der Ergebnisse in Ihre Dokumentation auf.

Aufgabe 2

Nun wird der temperaturabhängige Widerstand R_T an die Spannungsquelle Q1 angeschlossen wie in Abbildung 3 gezeigt. Tabelle 2 zeigt die aus dem Datenblatt bekannten Widerstandswerte.

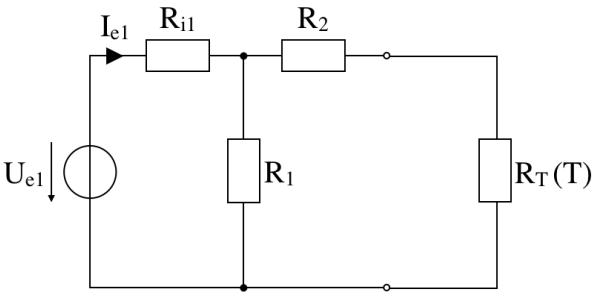

Abbildung 3: Der temperaturabhängige Widerstand R_T wird an die Spannungsquelle angeschlossen.

Die Schaltung soll bei Temperaturen zwischen 0 °C und 60 °C betrieben werden. Aus Sicherheitsgründen muss der maximal auftretende Strom I_{e1} bestimmt werden. Gehen Sie dazu wie folgt vor:

(a) Um den kontinuierlichen Verlauf des Widerstandswertes von R_T in Abhängigkeit von der Temperatur zu approximieren, sollen zunächst die Werte aus Tabelle 2 interpoliert werden, da das Datenblatt nur für wenige Temperaturen den Widerstandswert angibt. Schreiben Sie hierfür in MATLAB eine Funktion *linearеInterpolation*, die anhand von gegebenen Paaren aus Stützstelle und Funktionswert die Funktionswerte an bislang unbekannten Stützstellen durch lineare Interpolation bestimmt. Die lineare Interpolation nimmt zwischen gegebenen Paaren aus Stützstellen und zugehörigen Funktionswerten einen linearen Funktionsverlauf an. So können auch bislang unbekannte Funktionspunkte mithilfe einer Geradengleichung berechnet werden. Abbildung 4 veranschaulicht das Prinzip der linearen Interpolation.

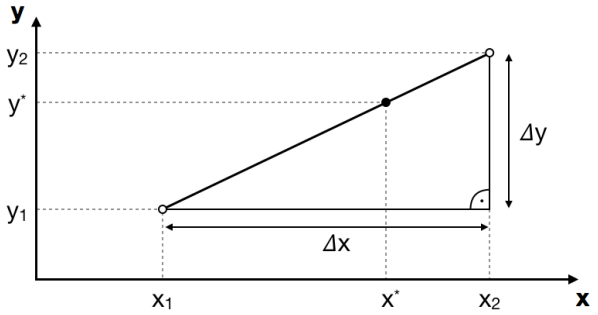

Abbildung 4: Mithilfe der linearen Interpolation lassen sich beliebige Wertepaare aus Stützstelle x^* und Funktionswert y^* berechnen, sofern zwei Funktionspunkte (x_1, y_1) und (x_2, y_2) gegeben sind mit $x_1 \leq x^* \leq x_2$.

Hinweise:

- Die Funktion soll einen Vektor von gegebenen Stützstellen ***gegX*** und einen dazugehörigen Vektor von Funktionswerten ***gegY*** sowie einen Vektor von Ziel-Stützstellen ***gesX*** als Argumente entgegennehmen.
- Die Nutzung von Interpolationsfunktionen, welche von MATLAB bereitgestellt werden, ist hier nicht gestattet.
- Die Funktion soll die durch lineare Interpolation bestimmten Funktionswerte an den Ziel-Stützstellen als Vektor *|bmgesY* zurückgeben.
- Erstellen Sie angelehnt an Aufgabe 1 (a) eine Ausgabe für den Aufruf der Hilfe dieser Funktion.
- Versehen Sie die wesentlichen Codeabschnitte mit erläuternden Kommentaren.
- Nehmen Sie den entstandenen MATLAB-Code in Ihre Dokumentation auf.
- Nehmen Sie einen Screenshot des Command-Windows nach Aufruf der Hilfe-Funktion in Ihre Dokumentation auf.

(b) Nutzen Sie nun Ihre Funktion *lineareInterpolation* aus Aufgabenteil (a), um den Widerstandswert von R_T für Temperaturen zwischen 0 °C und 60 °C in Schritten von 1 K zu interpolieren. Erstellen Sie hierfür in MATLAB ein Skript *Aufgabe2*, welches die Funktion *lineareInterpolation* mit entsprechenden Übergabeparametern aufruft. Visualisieren Sie Ihr Ergebnis als Diagramm, das die Temperatur auf der Abszissenachse und die zugehörigen Widerstandswerte auf der Ordinatenachse aufträgt. Hinweise:

- Die Funktionswerte an den gegebenen Stützstellen und an den Ziel-Stützstellen ([0, 1, 2, 3, ..., 60] °C) sollen im Graphen durch rote, quadratische Marker markiert werden.
- Achten Sie auf eine korrekte Achsenbeschriftung mit Angabe der jeweiligen Einheiten.
- Versehen Sie die wesentlichen Codeabschnitte mit erläuternden Kommentaren.
- Nehmen Sie den entstandenen MATLAB-Code in Ihre Dokumentation auf.
- Nehmen Sie das Diagramm in Ihre Dokumentation auf.

(c) Neben der linearen Interpolation, soll nun auch die Spline-Interpolation genutzt werden, um die Widerstandswerte von R_T für Temperaturen zwischen 0 °C und 60 °C in Schritten von 1 K zu interpolieren. Während die lineare Interpolation ausschließlich Geraden nutzt, um den Funktionsverlauf zwischen den Stützstellen zu approximieren, greift die Spline-Interpolation auch auf Polynome höherer Ordnung zurück. Nutzen Sie die MATLAB-Funktion *interp1*, um die Spline-Interpolation durchzuführen, und erweitern Sie Ihr MATLAB-Skript aus Aufgabenteil (b). Erstellen Sie ein Diagramm, welches die Temperatur auf der Abszissenachse und die zugehörigen Widerstandswerte auf der Ordinatenachse aufträgt. Das Diagramm soll sowohl einen Graphen für die lineare Interpolation als auch einen Graphen für die Spline-Interpolation in einem gemeinsamen Koordinatensystem darstellen. Hinweise:

- Die Spline-Interpolation soll mit der MATLAB-Funktion *interp1* durchgeführt werden. Die Spline-Interpolation muss nicht selbst implementiert werden.

- Die Funktionswerte an den gegebenen Stützstellen und an den Ziel-Stützstellen ([0, 1, 2, 3, ..., 60] °C) sollen im Graphen durch Marker markiert werden. Nutzen Sie für die Spline-Interpolation schwarze, kreisförmige Marker und für die lineare Interpolation rote, quadratische Marker.
- Achten Sie auf eine korrekte Achsenbeschriftung mit Angabe der jeweiligen Einheiten. Erstellen Sie eine Legende.
- Versehen Sie die wesentlichen Codeabschnitte mit erläuternden Kommentaren.
- Nehmen Sie den entstandenen MATLAB-Code in Ihre Dokumentation auf.
- Nehmen Sie das Diagramm in Ihre Dokumentation auf.

Im weiteren Verlauf der Aufgabe soll nur noch die Spline-Interpolation genutzt werden.

(d) Bestimmen Sie nun den sicherheitsrelevanten, maximalen Strom I_{e1}. Bei welcher Temperatur fließt der maximale Strom? Erweitern Sie hierzu Ihr MATLAB-Skript aus den vorangegangenen Aufgabenteilen. Hinweise:
- Eine Abtastung der Temperaturskala in Schritten von 1 K und die sich daraus ergebende Genauigkeit des maximalen Stroms ist ausreichend. Nutzen Sie zur Bestimmung des maximalen Stroms I_{e1} die Spline-Interpolation aus Aufgabenteil (d).
- Vereinfachen Sie zur Bestimmung von I_{e1} die Schaltung bestehend aus Q1 und B2 zu einem geeigneten Ersatzschaltbild und berechnen Sie den Ersatzwiderstand R_{ers}, welcher dem Gesamtwiderstand der Schaltung entspricht.
- Dokumentieren Sie den Wert des maximalen Stroms I_{e1} sowie die Temperatur, bei der dieser fließt.
- Nehmen Sie den entstandenen MATLAB-Code in Ihre Dokumentation auf.

(e) Es wird eine einstündige Messung durchgeführt. Die Umgebungstemperatur steigt von 15 °C zu Beginn der Messung linear auf 45 °C am Ende der Messung an. Bestimmen Sie den Gesamtwiderstand $R_{ges}(t)$ der Schaltung, den Strom $I_{e1}(t)$ sowie die von der Quelle generierte Leistung $P(t)$ über die Zeit t. Erweitern Sie hierzu Ihr MATLAB-Skript aus den vorangegangenen Aufgabenteilen. Hinweise:
- Die zeitliche Abtastung soll in Schritten von einer Minute erfolgen. Der erste Messwert wird zum Zeitpunkt t = 0 min erfasst; der letzte Messwert zum Zeitpunkt t = 59 min.
- Nutzen Sie die Spline-Interpolation, um Funktionswerte zwischen den gegebenen Stützstellen zu bestimmen.
- Visualisieren Sie Ihre Ergebnisse in vier vertikal übereinander angeordneten Koordinatensystemen innerhalb einer MATLAB-Figure. Auf der Abszissenachse soll jeweils die Zeit aufgetragen werden. Auf den vier Ordinatenachsen sollen die Temperatur T (in min), der Gesamtwiderstand R_{ges} (in Ω) der Schaltung, der Strom I_{e1} (in Ampere) sowie die Leistung P (in Watt) aufgetragen werden. Nutzen Sie den MATLAB-Befehl *subplot* für die Anordnung der vier Koordinatenachsen.

- Achten Sie auf eine korrekte Achsenbeschriftung mit Angabe der jeweiligen Einheiten.
- Nehmen Sie die Diagramme in Ihre Dokumentation auf.
- Nehmen Sie den entstandenen MATLAB-Code in Ihre Dokumentation auf.

(f) Welche Arbeit wurde zwischen t = 20 min und t = 30 min in der Gesamtschaltung aus Q1 und B2 umgesetzt? Erweitern Sie Ihr MATLAB-Skript aus den vorangegangenen Aufgabenteilen zur Berechnung der Arbeit in der Einheit Joule. Hinweise:

- Achten Sie auf korrekte Indizes. Der Zeitpunkt t = 0 min ist dem ersten Feld im Zeitvektor zugewiesen und erhält daher in MATLAB den Index 1.
- Berechnen Sie die Arbeit als Fläche unter der Leistungskurve. Nutzen Sie hierzu die MATLAB-Funktion *trapz*, welche die Trapezregel zur numerischen Integration anwendet.
- Dokumentieren Sie die resultierende Arbeit in der Einheit Joule.
- Nehmen Sie den entstandenen MATLAB-Code in Ihre Dokumentation auf.