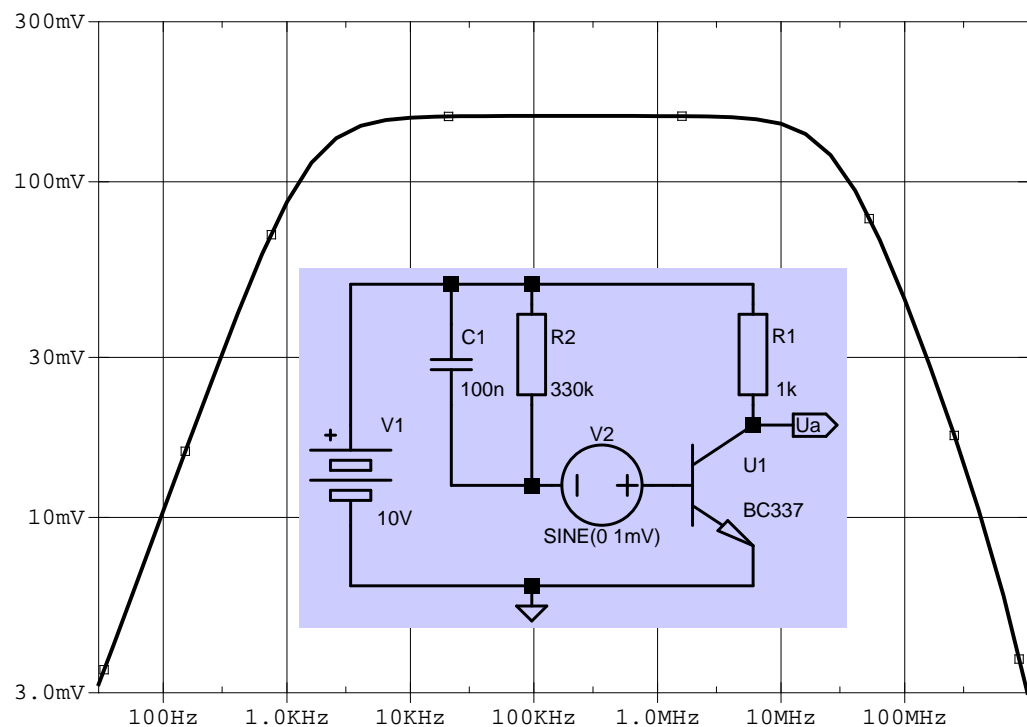


Schaltungs-Simulation

simulation.tex KB 20010930



Bei der Entwicklung von elektronischen Schaltungen – insbesondere aber auch bei integrierten Schaltkreisen – spielen leistungsfähige Design- und Simulations-Verfahren eine ganz entscheidende Rolle. ICs in der heutigen Komplexität wären ohne geeignete Modellierungsmethoden nicht mehr realisierbar. Aber auch bei der Entwicklung von einfachen diskreten Schaltungen sind Elektronik-CAD-Programme eine wertvolle Hilfe. Mit modernen EDA-Programmen (*Electronic Design Automation*) können Schaltungen konstruiert, simuliert und schließlich in ein fertiges Platinenlayout umgesetzt werden. Die elektrischen Eigenschaften der Bauelemente sind in umfangreichen Bauteilebibliotheken zusammengefasst und so im Simulationsprogramm verfügbar, die Funktion einer Schaltung kann somit während des Design-Prozesses ständig überprüft werden.

Das Skriptum soll in die physikalischen und mathematischen Grundlagen der Netzwerkberechnung einführen und sie an einfachen Beispiele erklären. Im Praktikum arbeiten Sie mit dem frei erhältlichen EDA-Programm LTSpice von Linear Technologies.

Inhaltsverzeichnis

1 Grundlagen der Netzwerkberechnung	3
1.1 Netzwerk	3
1.2 Formalisierung	3
1.3 Maschenstromverfahren	4
1.4 Knotenpotenzialverfahren	4
1.5 Beispiel zum Knotenpotenzialverfahren	5
1.6 Nichtlineare Netzwerke	6
1.7 Dynamische Netzwerkberechnung	7
2 Anwendungsbeispiele	8
2.1 Wechselstrombrücke	8
2.2 Emitterschaltung	9
2.3 Schutzschaltungen mit Dioden	10
2.4 Oszilloskop-Tastkopf (komplexer Spannungsteiler)	13
2.5 Schwingkreis	16
2.6 Gleichrichterschaltungen	17
2.7 Zusammenfassung: Netzwerktypen	19
3 EDA-Programme (Electronic Design Automation)	19
Literatur und Links	20

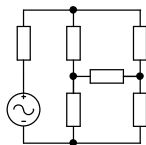
1 Grundlagen der Netzwerkberechnung

Formalisierte Verfahren zur Analyse und Berechnung von Netzwerken sind in der modernen Elektrotechnik zu einem wichtigen, unverzichtbaren Werkzeug in fast allen Teilgebieten geworden. Die oft sehr umfangreichen Algorithmen sind heute in leistungsfähigen Rechnerprogrammen implementiert, deren Bedienung in der Regel sehr benutzerfreundlich gehalten ist. Zum Verständnis und zur Abschätzung der Möglichkeiten und Grenzen ist es jedoch sinnvoll, sich ein wenig mit den Grundlagen zu beschäftigen.

1.1 Netzwerk

Unter einem **Netzwerk** versteht die Elektrotechnik eine zusammenhängende Schaltung aus Zweipolen, gekennzeichnet durch *Knoten*, *Zweige* und *Maschen*. Ein **Knoten** ist der Verbindungspunkt von mindestens drei Zweigen, ein **Zweig** ein Zweipol, der zwei Knoten verbindet, eine **Masche** ein geschlossener Kreis aus Zweigen. **Zweipole** sind Bauelemente mit zwei Anschlüssen, sie können aktiv oder passiv, reell oder komplex, linear oder nichtlinear sein. Müssen Vierpole mit modelliert werden, werden diese in geeignete Zweipole umgeformt. So ist z. B. der Ausgangszweig eines Transistors in Emitterschaltung eine stromgesteuerte Stromquelle mit dazu parallel geschaltetem Ausgangsleitwert (zur Beschreibung dieser Ersatzzweipole sind die h-Parameter nützlich).

Physikalische Basis der Netzwerkberechnung sind die beiden Kirchhoffschen Gesetze, vulgo *Knotenregel* und *Maschenregel*, die aus Grundkurs und Grundpraktikum hinlänglich bekannt sind.

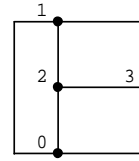


Als einfaches Beispiel für ein Netzwerk betrachten wir die nebenstehende Wheatstonesche Brückenschaltung. Die Zahl der Knoten (4) und Zweige (6) sind einfach abzählbar, bei den möglichen Maschen wird es etwas schwieriger: Voneinander unabhängige sind's drei, aus diesen lassen sich vier weitere, davon abhängige konstruieren. Zur Berechnung könnte man sich also ein Verfahren überlegen, das die Maschenregel auf drei voneinander linear unabhängige Maschen dieses Netzwerks anwendet, außerdem die Knotenregel auf drei der Knoten und als Variablen die sechs Zweigströme verwendet. Das entstehende lineare Gleichungssystem aus sechs Gleichungen würde man mit einem der Standardverfahren lösen.

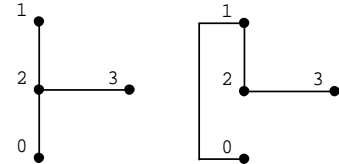
1.2 Formalisierung

Mit dem oben skizzierten Weg, Netzwerke zu berechnen, kommt man bei einfachen Systemen rasch zum Ziel, für komplexere Netzwerke sind jedoch stärker formalisierte Verfahren notwendig. Dies insbesondere, um Fehlermöglichkeiten weitgehend auszuschalten.

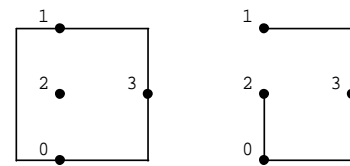
Man formalisiert das Netzwerk durch einen Graphen, in dem die Knoten (Anzahl k) geeignet indiziert werden. Für die obige Brückenschaltung ist das beispielsweise der rechts dargestellte Graph.



Ein **vollständiger Baum** ist ein Teilgraph, bei dem alle Knoten des Netzwerks durch Zweige verbunden sind, ohne dass eine geschlossene Masche entsteht. Zwei mögliche vollständige Bäume für das Beispielnetzwerk sind rechts dargestellt.



Der dazu **komplementäre Baum** ist der Rest des ursprünglichen Graphen, d. h. der Graph, aus dem die Zweige des vollständigen Baums entfernt wurden. Rechts die den darüber stehenden vollständigen Bäumen zugehörigen komplementären Bäume.



1.3 Maschenstromverfahren

Man kann zeigen, dass die Zahl der linear unabhängigen Maschen in einem Netzwerk gleich der Zahl der Zweige des komplementären Baumes ist. Darauf baut das Maschenstromverfahren auf. Als unabhängige Variablen werden die Teilströme in den Zweigen des komplementären Baums gewählt. Jeder dieser Zweige wird durch einen Teil des vollständigen Baums zu einer Masche ergänzt, die Ströme in den Zweigen des vollständigen Baums ergeben sich aus den Strömen im komplementären Baum. Auf jede der so gebildeten Maschen wird die Maschenregel angewandt, das resultierende lineare Gleichungssystem wird gelöst.

1.4 Knotenpotenzialverfahren

Weitgehend ohne 'Wald' kommt man beim Knotenpotenzialverfahren aus. Das Potenzial eines zentralen Knotens wird festgelegt (sinnvollerweise auf 0 V, Massepotenzial), die Potentialdifferenzen der übrigen $(k-1)$ Knoten zu diesem Bezugsknoten werden als unabhängige Variablen gewählt. Auf jeden dieser übrigen Knoten wird die Knotenregel angewandt, das resultierende lineare Gleichungssystem aus $(k-1)$ Gleichungen wird gelöst.

Zur vereinfachten Handhabung wird beim Knotenpotenzialverfahren jeder Zweig des Netzwerks in eine ideale Stromquelle mit einem parallel geschalteten (komplexen) Leitwert umgewandelt. Dabei kann eine der beiden Größen (Strom oder Leitwert) gleich Null sein (sind beide Null, gibt's den Zweig nicht). Eine lineare Spannungsquelle wird mit $I = -U/R$ und $G = 1/R$ in eine Stromquelle und den parallel geschalteten Leitwert umgewandelt. Es ist offensichtlich, dass dies Schwierigkeiten bei idealen Spannungsquellen ergibt ($R = 0$). Beim Knotenpotenzialverfahren ist somit eine spezielle Vorbehandlung von idealen Spannungsquellen

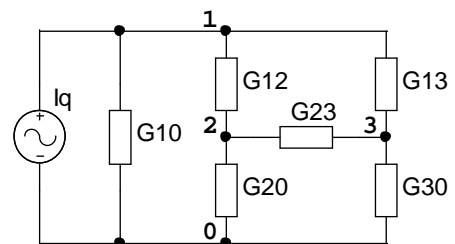
erforderlich (Verlegung der Spannungsquelle, Kontraktion bei der Anwendung der Knotenregel). Die formalisierte Anwendung des Verfahrens folgt mithin dem Ablauf:

- Vorbehandlung von idealen Spannungsquellen,
- Umwandlung aller Zweige in Stromquellen + Leitwert,
- Anwendung der Knotenregel auf die $k-1$ unabhängigen Knoten,
- Ersatz der Ströme durch die Differenzen der Knotenpotenziale und die Leitwerte,
- Lösung des linearen Gleichungssystems.

Wie man sieht, gibt es bei diesem Verfahren praktisch keine Variationsmöglichkeiten, wenn der Bezugsknoten einmal festgelegt ist (Massepotential). Es ist daher optimal für die Implementierung in einem Rechnerprogramm geeignet (das braucht dann nicht über Bäume nachzudenken).

1.5 Beispiel zum Knotenpotenzialverfahren

Exemplarisch soll das Knotenpotenzialverfahren auf das obige Beispiel einer Brückenschaltung angewendet werden. Die einzelnen Zweige sind in Stromquellen und Leitwerte umgewandelt (rechtes Bild). Alle Zweige sind linear, reell oder komplex, und statisch (zeitlich konstante Gleich- oder Wechselstromquellen).



Als Bezugspunkt wird der Knoten 0 gewählt, dessen Potenzial wird auf Null gesetzt. Als Variablen sind damit die Potenziale bzw. Potenzialdifferenzen U_1 , U_2 , U_3 der Knoten 1, 2, 3 bezüglich des Knotens 0 zu verwenden.

Anwendung der Knotenregel auf die Knoten 1, 2, 3:

$$-I_q + I_{10} + I_{12} + I_{13} = 0 \quad (1)$$

$$-I_{12} + I_{23} + I_{20} = 0 \quad (2)$$

$$-I_{13} - I_{23} + I_{30} = 0 \quad (3)$$

Ersetzung der Ströme durch die Knotenspannungen:

$$+U_1 * G_{10} + (U_1 - U_2) * G_{12} + (U_1 - U_3) * G_{13} = I_q \quad (4)$$

$$-(U_1 - U_2) * G_{12} + (U_2 - U_3) * G_{23} + U_2 * G_{20} = 0 \quad (5)$$

$$-(U_1 - U_3) * G_{13} - (U_2 - U_3) * G_{23} + U_3 * G_{30} = 0 \quad (6)$$

Umsortierung:

$$+(G_{10} + G_{12} + G_{13}) * U_1 - G_{12} * U_2 - G_{13} * U_3 = I_q \quad (7)$$

$$-G_{12} * U_1 + (G_{12} + G_{23} + G_{20}) * U_2 - G_{23} * U_3 = 0 \quad (8)$$

$$-G_{13} * U_1 - G_{23} * U_2 + (G_{13} + G_{23} + G_{30}) * U_3 = 0 \quad (9)$$

Das entstandene lineare Gleichungssystem $G * U = I$ hat eine relativ einfache Struktur:

- Die Koeffizientenmatrix ist symmetrisch.
- In der Diagonale stehen die Summen der an dem jeweiligen Knoten hängenden Leitwerte.
- Die Außerdiagonalelemente werden durch die negativen Leitwerte der Zweige zwischen den entsprechenden Knotenindizes gebildet.
- Der Vektor I setzt sich aus den vorhandenen Stromquellen zusammen.

Nach diesen Regeln kann das lineare Gleichungssystem von einem Programm direkt erstellt werden. Etwas komplizierter wird's allerdings dann, wenn Vierpole im Netzwerk vorhanden sind. Gesteuerte Stromquellen z. B. können bei der Erstellung des Gleichungssystems zunächst im Vektor I berücksichtigt werden, müssen anschließend aber in die Koeffizientenmatrix integriert werden.

Was an dem einfachen Beispiel noch nicht deutlich wird: Bei umfangreicheren Netzwerken wird die Koeffizientenmatrix zu einer dünn besetzten Bandmatrix oder kann in eine solche umgewandelt werden (jeder Knoten ist nur mit wenigen anderen Knoten durch Zweige verbunden). Zur effizienten Berechnung werden mithin Algorithmen für dünn besetzte Matrizen (*sparse matrices*) benötigt.

1.6 Nichtlineare Netzwerke

Bauelemente in linearen Netzwerken zeichnen sich durch lineare Kennlinien aus ($U = R * I$, R unabhängig von U und I). Ist diese Bedingung nicht mehr erfüllt (sie ist in strenger Weise nie erfüllt), spricht man von nichtlinearen Netzwerken. Die nichtlinearen Kennlinien können funktional oder als Wertetabellen vorgegeben sein. Zur Netzwerkberechnung werden Iterationsverfahren verwendet. Diskutieren Sie Vor- und Nachteile von gebräuchlichen Iterationsverfahren (Intervallschachtelung, Sekantenverfahren, Newton-Verfahren).

In den EDA-Programmen sind die Kennlinien der nichtlinearen Bauelemente (Dioden, Transistoren, ...) geeignet parametrisiert, sinnvolle Iterationsverfahren meist mit implementiert.

Hat man in nichtlinearen Netzwerken sowohl Gleich- wie auch Wechselstromquellen, kann man die Netzwerkberechnung in zwei Schritte aufteilen. Zunächst werden die Gleichspannungsverhältnisse berechnet, dabei werden alle Wechselstromquellen außer acht gelassen. Wie man leicht sieht, kann man bei dieser Berechnung alle Kapazitäten der Schaltung als nicht vorhanden, alle Induktivitäten als ideale Kurzschlüsse annehmen. Ergebnis dieses ersten Schritts sind die *Arbeitspunkte* aller

Komponenten, insbesondere der nichtlinearen. Die nichtlinearen Kennlinien werden dann an diesen Arbeitspunkten linearisiert. Im zweiten Schritt werden die Wechselstromquellen hinzugenommen, für diese Rechnung ist das Netzwerk dann quasi-linear. Diese Art der Berechnung ist natürlich nicht dazu geeignet, z. B. Verzerrungen durch nichtlineare Kennlinien zu charakterisieren. Es wird nur das *Kleinsignalverhalten* berechnet, das heißt das Verhalten der Schaltung in Bereichen, in denen die Kennlinien ohne allzu großen Fehler linearisierbar sind.

1.7 Dynamische Netzwerkberechnung

Sind die Gleich- oder Wechselstromquellen oder einzelne Bauelemente (Schalter o. ä.) zeitlich nicht konstant, müssen die Netzwerke *dynamisch* berechnet werden. Analytische Verfahren dafür sind unter anderem die Lösung der das System beschreibenden Differenzialgleichung(en) mit geeigneten Anfangsbedingungen oder die Transformation des Problems in den Frequenzbereich (Fourier- oder Laplace-Transformation), Lösung dort und Rücktransformation. Solche analytischen Verfahren sind für einfache, lineare Systeme gut geeignet, sie führen zu einer anschaulichen funktionalen Beschreibung. So liefert die Differenzialgleichung für die Entladung eines Kondensators direkt die Exponentialfunktion als Lösung für deren exponentiellen Verlauf, die Antwort einer Schaltung auf ein Rechtecksignal lässt sich durch dessen Fourierkomponenten beschreiben.

Dem Vorteil der Anschaulichkeit steht bei den analytischen Verfahren der dynamischen Netzwerkberechnung der Nachteil der begrenzten Anwendbarkeit (Lösbarkeit der Differenzialgleichungen, Linearität) gegenüber. Darüber hinaus sind die Verfahren nicht immer einfach formalisierbar. Daher werden heute bei der dynamischen Netzwerkberechnung praktisch nur noch moderne numerische Lösungsverfahren eingesetzt. Solche Verfahren können meist auch problemlos mit nichtlinearen Bauelementen arbeiten. Der Zeitablauf wird in kleine Zeitschritte eingeteilt, deren Größe dynamisch an die gewünschte Genauigkeit angepasst wird. Ausgehend von einer Anfangssituation wird die Zeitabhängigkeit aller Größen (Spannungen, Ströme) numerisch berechnet. Für Kapazitäten und Induktivitäten wird deren differenzielles Verhalten eingesetzt, Zeitabhängigkeiten einzelner Komponenten (Stromquellen, Schalter etc.) werden direkt berücksichtigt. Die Anfangssituation ergibt sich aus der statischen Berechnung des Netzwerks mit den Anfangswerten aller Größen.

Dynamische d. h. zeitabhängige Berechnungsverfahren lassen sich natürlich auch anwenden, wenn man rein sinusförmige Spannungen oder Ströme an den Quellen hat. Sie sind dann geeignet, Verzerrungen durch nichtlineare Bauelemente zu beschreiben, die bei den üblichen statischen Berechnungsverfahren (Linearisierung um den Arbeitspunkt, Kleinsignalverhalten) nicht berücksichtigt werden.

2 Anwendungsbeispiele

An verschiedenen konkreten Beispielen – einfachen linearen und nicht-linearen, statischen und dynamischen Netzwerken – sollen Rechenverfahren zur Netzwerksimulation angewendet werden.

2.1 Wechselstrombrücke

Ein typisches Beispiel für ein lineares Netzwerk aus passiven Komponenten, das statisch zu berechnen ist, ist die Brückenschaltung nach Abbildung 1. Sie kann zur Frequenzmessung verwendet werden. Man kann dies zeigen, indem man Amplitude und Phase der Spannung am Messgerät (Widerstand R_5) in Abhängigkeit von der Frequenz der an der Brücke anliegenden Sinusspannung berechnet.

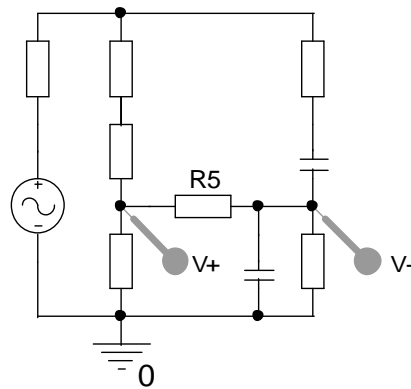


Abbildung 1: Wechselstrombrücke zur Frequenzmessung.

Das Ergebnis der Rechnung zeigt Abbildung 2, die an R_5 anliegende Spannung als Funktion der Frequenz. Mit den für die Simulation eingesetzten Widerstands- und Kapazitätswerten ($1\text{ k}\Omega$ und 1 nF) ist die Brücke bei einer Frequenz von $1/(2\pi)$ MHz abgestimmt. Bei einer Anwendung der Brücke zur Frequenzmessung würde man die Widerstände der beiden rechten Zweige variabel machen, die Kapazitäten umschaltbar.

2.1.1 Berechnung ohne Hilfsmittel

Das lineare Gleichungssystem, das die Schaltung beschreibt, lässt sich durch anschauliche Anwendung der Kirchhoffschen Gesetze aufstellen, besser verwendet man jedoch eins der beschriebenen formalisierten Verfahren (z. B. das Knotenpotentialverfahren). Das Gleichungssystem ist einfach genug, um es mit Papier und Bleistift zu lösen (*Cramersche Regel* o. ä.).

Erinnern Sie sich an numerische Lösungsverfahren, die Sie aus Numerik-Vorlesungen kennen: Gauß-Verfahren (Vorwärtselimination, Pivotisierung, Rückwärtseinsetzen), LR-Zerlegung (LU decomposition), Matrixinversion.

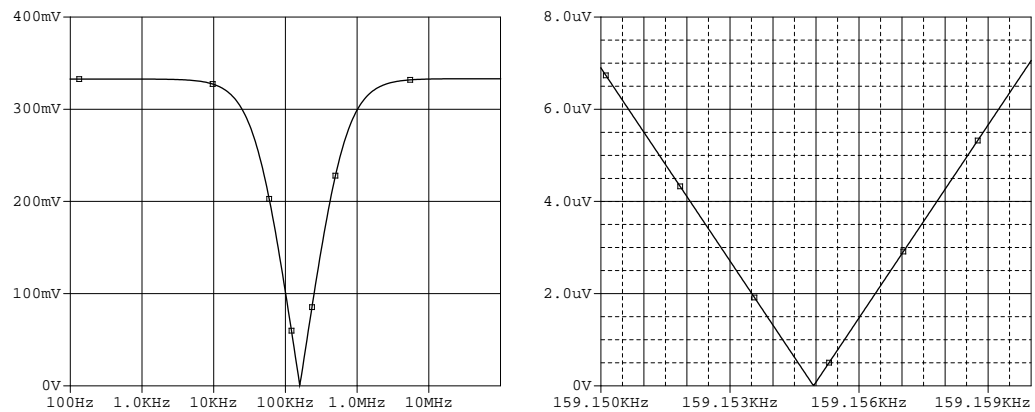


Abbildung 2: Spannung am Messgerät (Widerstand R_5) der Messbrücke aus Abbildung 1 als Funktion der Frequenz der angelegten Wechselspannung. Die Vergrößerung im rechten Teilbild zeigt die Empfindlichkeit des Brückenabgleichs und damit die erreichbare Genauigkeit.

2.1.2 MATLAB

In MATLAB ist die Lösung linearer Gleichungssysteme nach dem Gauß-Verfahren als formale Division implementiert (*Matrix Left Division*). Das lineare Gleichungssystem $G * U = I$ hat die Lösung $U = G \setminus I$. Hilfe in MATLAB dazu unter *help slash* oder *help mldivide*. Bei umfangreicheren Problemen kann man ausnutzen, dass die Koeffizientenmatrix bei Netzwerkproblemen dünn besetzt ist (ein Knoten ist meist nur mit wenigen Knoten seiner nähere Umgebung verbunden), Hilfe unter *help sparse*.

2.1.3 Maple, Mathematica

Während MATLAB vom Konzept her auf Numerik ausgelegt ist, die Frequenzabhängigkeit der Lösung mithin nur punktweise berechnet werden kann, können in Computeralgebraprogrammen lineare Gleichungssysteme mit parametrisierten Koeffizienten auch analytisch gelöst werden. Mathematica stellt hierzu die Funktion *Solve* bereit, sie wird aufgerufen mit der Gleichungsliste und der Variablenliste als Parameter.

2.2 Emitterschaltung

Ebenfalls linear und statisch (d. h. durch die Lösung eines linearen Gleichungssystems), aber mit einer aktiven Komponente, ansonsten nur passiven, kann die Wechselstromverstärkung von einfachen Transistorverstärkern berechnet werden. Abbildung 3 zeigt links die Schaltung eines Verstärkers in Emitterschaltung, rechts deren Frequenzgang (frequenzabhängige Verstärkung) für verschiedene Werte von C_m .

Das Niederfrequenzverhalten der Schaltung wird durch die Größe des Koppelkondensators C bestimmt, das Hochfrequenzverhalten durch die Kollektor-Emitter-Kapazität des verwendeten Transistors. Diese Kapazität wirkt als Wechselstromgegenkopplung und vermindert so die Ver-

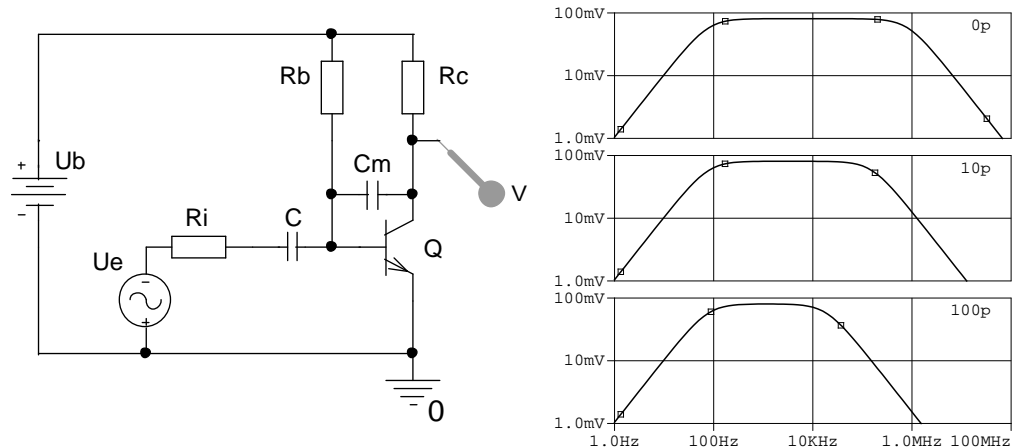


Abbildung 3: Transistorverstärker in Emitterschaltung, links die Schaltung – die Wirkung der Kollektor-Basis-Kapazität des Transistors wird durch einen zusätzlichen Kondensator C_m verdeutlicht, rechts der Frequenzgang der Verstärkung für verschiedene Werte von C_m (0, 10, 100 pF).

stärkung bei hohen Frequenzen (Miller-Effekt). Dieser Effekt wird durch eine zusätzliche, dazu parallel geschaltete Kapazität C_m verdeutlicht.

Zur Berechnung des Wechselstromverhaltens (Kleinsignalverstärkung) kann die Schaltung wie in Abbildung 4 dargestellt formalisiert werden. Der Transistor wird durch eine gesteuerte Stromquelle ersetzt, deren Stromverstärkung B ist durch die Gleichspannungsverhältnisse der Schaltung definiert (Arbeitspunkt im Kennlinienfeld des Transistors).

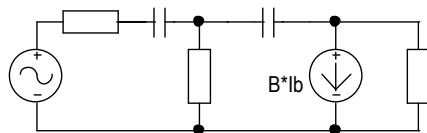


Abbildung 4: Formalisierung der Schaltung aus Abbildung 3 zur Berechnung des Wechselstromverhaltens. Der Eingangswiderstand und die Stromverstärkung B des Transistors werden als konstant angenommen (Kleinsignalberechnung).

Das Kleinsignalverhalten (Wechselstromverstärkung) der Schaltung wird durch lineare Gleichungen beschrieben, deren Lösung ist wie im vorstehenden Beispiel mit 'Papier und Bleistift', Numerikprogrammen (MATLAB) oder Computeralgebraprogrammen (Mathematica, Maple) möglich.

2.3 Schutzschaltungen mit Dioden

Eingänge von Verstärkern, Zählern oder ähnlichem werden vor Überspannungen (Störspannungen oder hohe Signalspannungen) durch Schutzschaltungen geschützt, in denen Dioden eine wesentliche Rolle spielen. Der Grund für die Verwendung von Dioden ist deren extrem nichtlineare Kennlinie. Sieht man von ohmschen Anteilen ab, kann die Kennlinie mit

guter Genauigkeit durch die Shockley-Formel [1]

$$I = I_S(\exp \frac{eU}{kT} - 1) \quad (10)$$

beschrieben werden. Bei negativen Spannungen (Sperrbereich) und auch bei geringen positiven ist der Widerstand von Dioden sehr hoch, bei größeren positiven Spannungen (Durchlassbereich) wird er dagegen stetig geringer. Man kann daher recht einfach einen spannungsabhängigen Spannungsteiler realisieren, der über einen weiten Eingangsspannungsbereich die Ausgangsspannung auf einen annähernd konstanten Wert begrenzt (Abbildung 5).

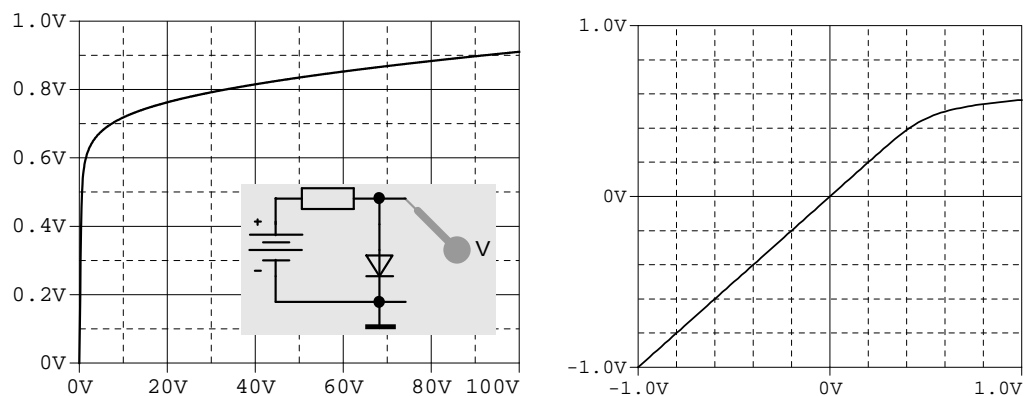


Abbildung 5: Ausgangsspannung eines nichtlinearen Spannungsteilers aus Widerstand und Diode (kleines Teilbild) als Funktion der Eingangsspannung (x-Koordinate: Eingangsspannung, y-Koordinate: Ausgangsspannung). Linkes Bild: Große Eingangsspannungen in Durchlassrichtung der Diode werden progressiv begrenzt. Rechtes Bild: Für negative Eingangsspannungen (Sperrbereich) und kleine positive ist die Kennlinie linear mit Steigung 1, die Ausgangsspannung etwa gleich der Eingangsspannung, da dort der Widerstand der Diode praktisch unendlich ist.

Zur Berechnung kann man die Masche betrachten oder formalisiert das Knotenpotentialverfahren anwenden, beides führt zu einer transzendenten Gleichung des Typs:

$$I_S(\exp \frac{eU}{kT} - 1) = \frac{U_0 - U}{R} \quad (11)$$

U_0 ist darin die Eingangsspannung, U die Ausgangsspannung. Die Struktur der Gleichung und eine Möglichkeit zu deren graphischer Lösung veranschaulicht Abbildung 6, die Lösungsmenge für U ist definiert durch die Schnittpunkte der Exponentialfunktion für den Diodenstrom (linke Seite von Gleichung 11) mit Widerstandsgeraden für verschiedene Werte von U_0 (rechte Seite von Gleichung 11).

Zur numerischen Lösung von transzendenten Gleichungen gibt es verschiedene formalisierte Verfahren: Intervallschachtelung, Newton-Verfahren, Sekanten-Verfahren, Sukzessive Approximation. Man formt die

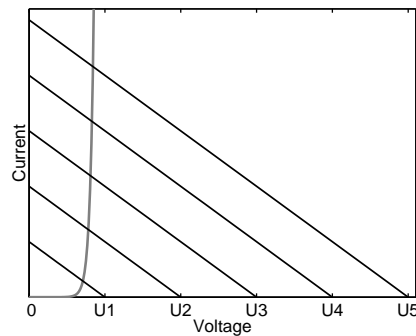


Abbildung 6: Graphische Lösung von Gleichung 11. Die Lösungsmenge besteht aus den Spannungswerten für die Schnittpunkte der Diodenkennlinie mit den Widerstandsgeraden zu den unterschiedlichen Eingangsspannungen.

Gleichung so um, dass man die Nullstellen einer Funktion zu bestimmen hat, im obigen Beispiel sind das die Nullstellen der Funktion

$$f(U) = I_S \left(\exp \frac{eU}{kT} - 1 \right) - \frac{U_0 - U}{R} \quad . \quad (12)$$

Die Berechnung wird am einfachsten mit einem Numerikprogramm wie MATLAB durchgeführt. Was MATLAB dafür anbietet, ist unter dem Indexeintrag 'zero of a function' in der MATLAB-Hilfe zu finden.

Für die Nullstellensuche bei Funktionen mit einer Variablen ist die MATLAB-Funktion `fzero` zuständig, deren Anwendung auf das obige Beispiel veranschaulicht das nachstehende Skript:

```
function u = divider(u0)
if nargin==0, u0=linspace(0,10,100); end;
for i=1:size(u0,2),
    u(i) = fzero(@diode,0,optimset('disp','off'),u0(i));
end;
plot(u0,u,'Linewidth',2,'Color',[0 0 0]);
set(gca,'Linewidth',1.5); .
```

Die Funktion `fzero` wird in einer Schleife für alle Eingangsspannungen aufgerufen und liefert die zugehörigen Ausgangsspannungen. Die im Aufruf von `fzero` als Parameter angegebene Funktion `diode` implementiert die Funktion, deren Nullstelle zu bestimmen ist:

```
function y = diode(u,u0);
r = 100;
i0 = 1e-10;
id = i0*(exp(40*u)-1);
ir = (u0-u)/r;
y = ir-id; .
```

Bei Begrenzerschaltungen verwendet man zwei Dioden und stellt die Spannungsgrenzen durch die Vorspannungen an diesen Dioden ein (Abbildung 7).

Solche Diodenschaltungen schützen die nachfolgende Elektronik sehr zuverlässig vor Überspannungen. Kurzzeitige Überspannungen bleiben folgenlos, bei länger andauernder Überlastung leidet nur der Widerstand, mit wenigen Cents das billigste Bauteil in der Schaltung.

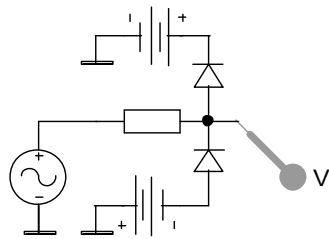


Abbildung 7: Spannungsbegrenzung mit zwei Dioden; die positiven und negativen Spannungsgrenzen werden durch die jeweiligen Vorspannungen an den Dioden eingestellt.

Die Funktionsweise der Schaltung in Abbildung 7 visualisieren die Simulationen in Abbildung 8.

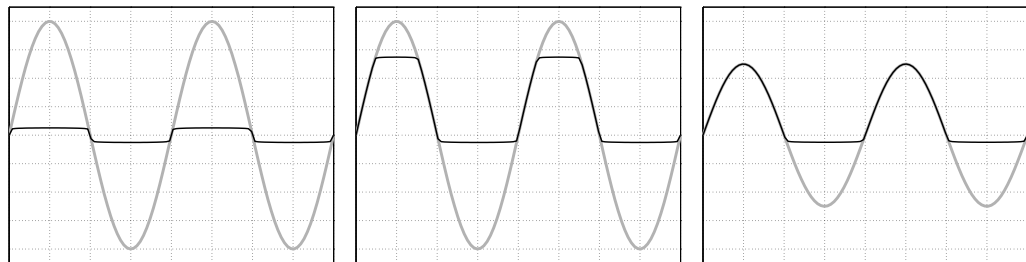


Abbildung 8: Funktionsweise der Begrenzerschaltung nach Abbildung 7 für verschiedene Eingangsspannungen und Spannungsgrenzen, Eingangsspannungen in grau, Ausgangsspannungen in schwarz. Links: Spannungsgrenzen jeweils 0 V – Schutzschaltung für Verstärker mit hoher Eingangsempfindlichkeit, Eingangsamplitude 8 V. Mitte: Spannungsgrenzen +5 V, -0 V – Schutzschaltung für TTL-Eingänge, Eingangsamplitude 8 V. Rechts: wie Mitte, Eingangsamplitude 5 V.

2.4 Oszilloskop-Tastkopf (komplexer Spannungsteiler)

Als einfaches Beispiel für die dynamische Berechnung eines passiven Netzwerks mit linearen Zweigen diskutieren wir den (komplexen) Spannungsteiler nach Abbildung 9, der die Wirkungsweise eines Oszilloskop-Tastkopfes schematisiert. Tastköpfe dieser Art werden verwendet, um die ohmsche und kapazitive Belastung der Messstelle möglichst gering zu halten. Üblich sind Teilverhältnisse von 10:1 oder 100:1, bei speziellen Hochspannungstastköpfen auch 1000:1.

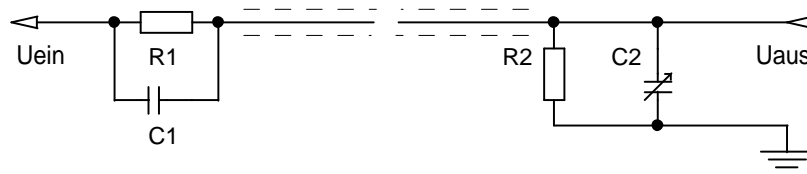


Abbildung 9: Ersatzschaltung eines Oszilloskop-Eingangs mit angeschlossenem Tastkopf.

R_1 und der Eingangswiderstand des Oszilloskops R_2 bestimmen das Teilverhältnis, in C_2 sind Eingangskapazität und Kabelkapazität zusammengefasst, C_1 wird zur Kompensation von C_2 benötigt. Wichtig ist, dass

das Teilverhältnis rein reell, d. h. frequenzunabhängig ist, nur dann können Spannungsverläufe unverzerrt gemessen werden. Das Teilverhältnis ist der Quotient aus (komplexem) Eingangs- und Ausgangswiderstand:

$$T = \frac{(R_1 \parallel C_1) + (R_2 \parallel C_2)}{R_2 \parallel C_2} \quad , \quad (13)$$

die Bedingung für reelles T (*Kompensation*) ist

$$R_1 : R_2 = C_2 : C_1 \quad . \quad (14)$$

Um diese Bedingung einzustellen, kann bei Tastköpfen entweder C_1 oder eine Zusatzkapazität zu C_2 variiert werden. Für diesen Tastkopfabgleich stellen fast alle Oszilloskope eine Rechteckspannung zur Verfügung. Abbildung 10 zeigt die Form dieses Rechtecksignals am Oszilloskop bei fehlerhafter und bei richtiger Einstellung des Kapazitätsverhältnisses.

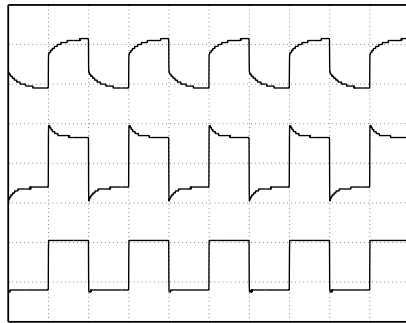


Abbildung 10: Rechtecksignal bei verschiedenen Einstellungen der Tastkopfkapazität (Messwerte).

Oben:

unterkompensiert ($C_1 : C_2 < R_2 : R_1$),

Mitte:

überkompensiert ($C_1 : C_2 > R_2 : R_1$),

unten:

richtig kompensiert ($C_1 : C_2 = R_2 : R_1$).

Zur Simulation der Wirkungsweise einer Schaltung bei Rechteck- oder sonstigen nicht sinusförmigen Signalen gibt es (mindestens) zwei grundsätzliche Möglichkeiten:

Fourierzerlegung : Das Signal wird in seine Fourierkomponenten zerlegt, eine statische Netzwerkberechnung für jede der Komponenten wird durchgeführt, am Ende wird wieder summiert. Ein Anwendungsbeispiel dafür – die Anpassung des simulierten an ein gemessenes Tastkopfsignal – finden Sie im Skriptum zur Vorlesung *Graphik-Workshop* [2], Seite 79 ff.

Dynamische Berechnung : Das zeitabhängige Verhalten des Netzwerks wird direkt berechnet (Transienten-Analyse). Von wenigen Ausnahmefällen abgesehen ist dies nur numerisch möglich: Das Zeitverhalten wird in genügend kleine Zeitschritte unterteilt und punktweise berechnet. Das das Netzwerk beschreibende System gewöhnlicher Differenzialgleichungen wird mithin numerisch gelöst.

Das Verfahren der Fourierzerlegung setzt voraus, dass man verschiedene Signale in einem Netzwerk ungestört überlagern kann, lässt sich mithin nur auf streng lineare Netzwerke anwenden. Die direkte dynamische Berechnung unterliegt dieser Einschränkung nicht, ist daher allgemeiner anwendbar. Auch aus diesem Grund werden wir uns hier nur mit dieser zweiten Möglichkeit beschäftigen.

Mit den Größen und Bezeichnungen der Schaltung in Abbildung 11 wird das Gleichungssystem aufgestellt: Die Knotenpotenziale U_1 und U_2 werden als Variablen gewählt, die Stromsumme in diesen Knoten wird berechnet (der Strom durch eine Kapazität ist $I = C\dot{U}$). Damit ergibt sich:

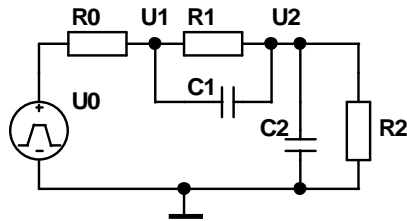


Abbildung 11: Netzwerk zur Beschreibung eines Oszilloskop-Tastkopfs an einer Rechteckspannung. Das Massepotential wird als Bezugsknoten ($U = 0$) festgelegt, die beiden Knotenspannungen U_1 und U_2 werden als Variablen verwendet.

$$(U_1 - U_0)/R_0 + (U_1 - U_2)/R_1 + (\dot{U}_1 - \dot{U}_2)C_1 = 0 \quad (15)$$

$$(U_2 - U_1)/R_1 + (\dot{U}_2 - \dot{U}_1)C_1 + U_2/R_2 + \dot{U}_2C_2 = 0 \quad (16)$$

Zur numerischen Integration von Systemen gewöhnlicher Differenzialgleichungen erster Ordnung (Anfangswertprobleme) verfügt MATLAB über mehrere unterschiedliche Algorithmen; welcher davon am zweckmäßigsten ist, hängt von der Art des Problems ab. Näheres dazu findet man in der MATLAB-Hilfe unter dem Stichwort *ODE* (Ordinary Differential Equations).

Für jede dieser Funktionen benötigt MATLAB geeignete Ausdrücke für die Differenziale. Im Beispiel kann man die beiden Gleichungen addieren und das Ergebnis zusammen mit der ersten Gleichung für MATLAB aufbereiten:

$$\dot{U}_2 = -1/C_2 [(U_1 - U_0)/R_0 + U_2/R_2] \quad (17)$$

$$\dot{U}_1 = \dot{U}_2 - 1/C_1 [(U_1 - U_0)/R_0 + (U_1 - U_2)/R_1] \quad (18)$$

Die konkrete MATLAB-Formulierung des Systems aus zwei gewöhnlichen Differenzialgleichungen erster Ordnung mit geeignet gewählten Widerständen und Kondensatoren (C_2 variabel) könnte dann so aussehen:

```
function dudt = dgl(t,u,c2)
r0 = 10000;
r1 = 9e6;
r2 = 1e6;
c1 = 10e-12;
t = (t+0.0009)*2000;
u0 = 1-2*fix(mod(t,2));
du2dt = -1/c2*((u(1)-u0)/r0+u(2)/r2);
du1dt = du2dt-1/c1*((u(1)-u0)/r0+(u(1)-u(2))/r1);
dudt = [du1dt;du2dt];
```

U_0 ist darin als Rechteckspannung mit einer Frequenz von 1 kHz definiert.

Die so implementierte Differenzialgleichungsfunktion wird der Lösungsfunktion (hier `ode15s`) für gewöhnliche Differenzialgleichungen erster Ordnung als erstes Argument (`@dgl`) beim Aufruf übergeben:

```
function [t,u] = step(c2)
if nargin<1, c2 = 90e-12; end;
[t,u] = ode15s(@dgl,[0,2e-3],[-1;-0.1],[],c2);
```

Weitere Argumente für die ode-Funktionen sind das Zeitintervall, in dem die Lösung berechnet werden soll, Startwerte für die Integration (die statisch berechnete Anfangssituation: -1 V an einem rein ohmschen Spannungsteiler), Optionen (MATLAB-Hilfe unter *ODESET*), sowie Parameter, die an die Differenzialgleichungsfunktion übergeben werden (hier C_2).

Das Simulationsergebnis, die mit MATLAB berechneten Ausgangsspannungen am Tastkopf für unterschiedliche Werte von C_2 – zu groß, zu klein, passend – ist in *Abbildung 12* zusammengestellt. Die Übereinstimmung mit den Messwerten (*Abbildung 10* ist offensichtlich).

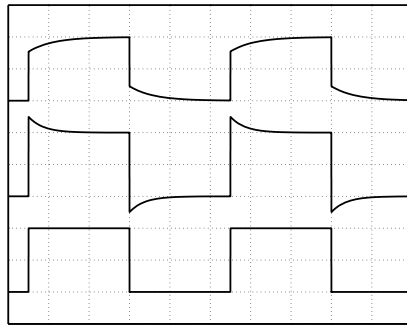


Abbildung 12: Rechtecksignal bei verschiedenen Einstellungen der Tastkopfkapazität (Simulation mit MATLAB).

Oben:

unterkompensiert ($C_1 : C_2 < R_2 : R_1$),

Mitte:

überkompensiert ($C_1 : C_2 > R_2 : R_1$),

unten:

richtig kompensiert ($C_1 : C_2 = R_2 : R_1$).

R_1).

2.5 Schwingkreis

Ein weiteres einfaches Beispiel für die dynamische Berechnung eines Netzwerkes mit linearen, komplexen Zweigen ist der Schwingkreis in *Abbildung 13*, der durch eine Rechteckspannung zu (gedämpften) Schwingungen angeregt wird.

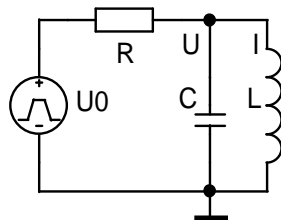


Abbildung 13: LC-Schwingkreis, der durch eine Spannungsquelle U_0 (Rechteckspannung) zu Schwingungen angeregt wird.

Mathematisch beschrieben wird ein Schwingkreis durch eine Differenzialgleichung zweiter Ordnung; diese wird für die numerische Berechnung mit MATLAB in zwei Differenzialgleichungen erster Ordnung umgeformt. Durch geeignete Wahl der Variablen lassen sich zwei Differenzialgleichungen, die das System beschreiben, auch ohne den Zwischenschritt über die Differenzialgleichung zweiter Ordnung formulieren. Mit der Spannung U am Schwingkreis und dem Strom I durch die Indukti-

vität als Variablen ergeben sich:

$$\dot{U} = -1/C [(U - U_0)/R + I] \quad (19)$$

$$\dot{I} = U/L \quad (20)$$

U_0 ist die aktuelle Quellenspannung.

Für MATLAB aufbereitet – mit geeignet gewählten Größen R , L und C und U_0 als Rechteckspannung mit 1 kHz:

```
function dydt = dglres(t,y)
R = 2200;
L = 1e-2;
C = 2.2e-8;
u0 = 1-2*fix(mod(2000*t+1.9,2));
dudt = -1/C*((y(1)-u0)/R+y(2));
didt = y(1)/L;
dydt = [dudt; didt];
```

Die Berechnung erfolgt dann wieder durch Aufruf einer geeigneten ode-Funktion:

```
[t,y] = ode45(@dglres,[0,2e-3],[0;-1/2200]);
```

Die Startwerte ergeben sich aus der Gleichstromberechnung des Netzwerks (Induktivität als idealer Kurzschluss, Kapazität ohne Wirkung) mit der Anfangsbedingung $U_0 = -1$ V.

Das Ergebnis der Integration – gedämpfte Schwingungen, die jeweils durch die Spannungssprünge der Quellenspannung angeregt werden – zeigt Abbildung 14.

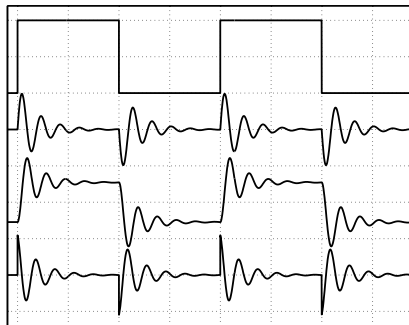


Abbildung 14: Simulation eines durch eine Rechteckspannung zu Schwingungen angeregten LC-Kreises (Schaltung wie in Abbildung 13).

Von oben nach unten:

- anregende Rechteckspannung,
- Spannung am LC-Kreis,
- Strom durch die Induktivität L ,
- Strom durch die Kapazität C .

2.6 Gleichrichterschaltungen

Einfache Beispiele für Netzwerke, die sowohl nichtlinear als auch dynamisch berechnet werden müssen, sind Gleichrichterschaltungen. Abbildung 15 zeigt im linken Teil eine einfache Einweggleichrichterschaltung, rechts eine Brückenschaltung. Die Spannungsquellen mit dem Innenwiderstand R_i liefern sinusförmige Spannungen, die gleichgerichteten Spannungen werden mit einem Kondensator C geglättet und mit einem Lastwiderstand R_l belastet.

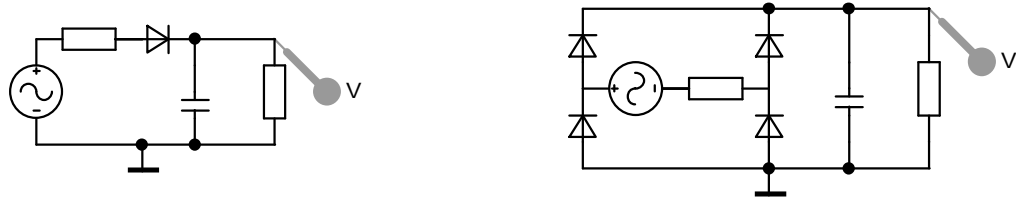


Abbildung 15: Gleichrichterschaltungen: Links Einweggleichrichter mit einer Diode, rechts Brückenschaltung mit vier Dioden.

Die Berechnung erfolgt in MATLAB wie üblich durch Integration der beschreibenden Differenzialgleichung mit einer der `ode`-Funktionen:

```
[t,u] = ode15s(@rectdgl,[0,T],[8]);
```

Die Differenzialgleichung mit typischen Größen für die einzelnen Bauelemente:

```
function dudt = rectdgl(t,u)
global ut ir
u0 = 14*sin(2*pi*50*t); % 50 Hz, 10 V eff
c = 3300e-6;
r1 = 22;
ut = u0-u;
fzero(@diode,[-30 10]);
dudt = -u/r1/c + ir/c;
```

Wegen der nichtlinearen Diodenkennlinie muss für den betreffenden Zweig jeweils eine Nullstellensuche gemacht werden (`fzero`).

Die Reihenschaltung aus Diode und Innenwiderstand der Spannungsquelle wird beschrieben durch:

```
function y = diode(ud);
global ut ir
ri = 2.2;
id = 1e-10*(exp(40*ud)-1);
ir = (ut-ud)/ri;
y = ir-id;
```

Die Parameter U_t (Gesamtspannung an der Reihenschaltung Diode- R_i) und I_r (Strom durch Diode und R_i) werden über globale Variable zwischen den beiden Funktionen ausgetauscht (schneller als über Funktionsparameter).

Typische Simulationsergebnisse sind in Abbildung 16 dargestellt. Deutlich wird insbesondere, dass der Spitzenstrom durch die Dioden wesentlich höher als der mittlere Strom ist. Darauf ist bei der Auswahl von Dioden für solche Schaltungen zu achten.

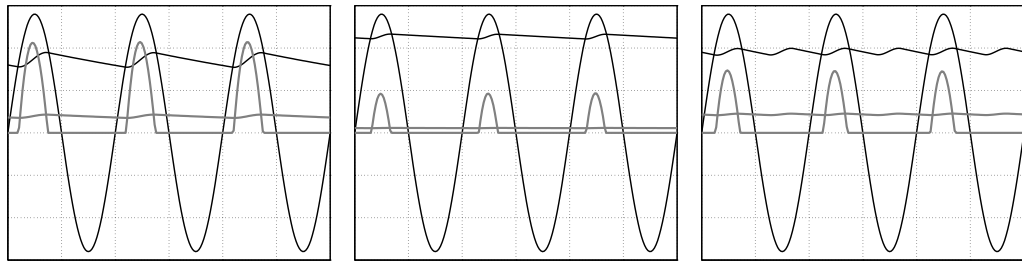


Abbildung 16: Spannungen und Ströme bei Gleichrichterschaltungen. Links: Einweggleichrichtung mit 22Ω Lastwiderstand, Mitte: Einweggleichrichtung mit 100Ω Lastwiderstand, rechts: Brückenschaltung mit 22Ω Lastwiderstand. Schwarz die Eingangs- und Ausgangsspannungen, grau die Diodenströme und die Ströme am Lastwiderstand. Vertikal-Maßstab: 5 V/Div , bzw. 1 A/Div .

2.7 Zusammenfassung: Netzwerktypen

Passiv, linear, statisch: Die einfachste Art von Netzwerken, reine Gleichstromnetzwerke oder Wechselstromnetzwerke bei einer Frequenz, Beschreibung durch ein lineares Gleichungssystem mit konstanten komplexen Koeffizienten, Berechnung in MATLAB durch Matrix-Division (`mldivide`).

Aktive Komponenten: Vierpole, Ersatz durch gesteuerte Strom- oder Spannungsquellen.

Nichtlineare Zweige: Beschreibung dieser Zweige durch geeignete nicht-lineare Funktionen oder durch Tabellen und Interpolationsverfahren, das Ergebnis ist ein nichtlineares Gleichungssystem, Berechnung in MATLAB mit `fsolve`, in sehr einfachen Fällen mit `fzero`.

Linearisierung: Nichtlineare Netzwerke mit Gleich- und Wechselspannungen, Berechnung der Arbeitspunkte mit den Gleichspannungen, Linearisierung der Kennlinien, anschließend (lineare) Berechnung des Wechselspannungsverhaltens.

Dynamische Berechnung: Zeitabhängige Berechnung eines Netzwerks (Transientenanalyse), Beschreibung durch ein System gewöhnlicher Differentialgleichungen erster Ordnung, Startwerte aus der statischen Berechnung der anfänglichen Gleichstromsituation, Integration der Differentialgleichung in MATLAB mit einer der `ode`-Funktionen.

3 EDA-Programme (Electronic Design Automation)

Die Beispiele in MATLAB zeigen, dass die Schaltungs-Simulation grundsätzlich in einem allgemeinen Numerik-Programm machbar ist, bei umfangreicheren Schaltungen aber auch sehr aufwendig sein kann. Das Gleichungssystem muss von Hand zusammengestellt werden, die zu

verwendenden Algorithmen müssen – dem Problem angepasst – aus- gesucht und implementiert werden. Die Herstellerfirma von MATLAB – [The MathWorks Inc.](#) [3] hat dies natürlich auch erkannt und bietet seit längerem ein Zusatzprodukt zu MATLAB an – SIMULINK, das als Front- End für Simulationen aller Art die Bedienung von MATLAB in diesem Bereich deutlich vereinfacht.

Ganz auf die elektronische Schaltungstechnik spezialisiert sind EDA- Programme (*Electronic Design Automation*), die den Gesamtbereich vom Schaltungsentwurf über die Simulation bis zum Platinenlayout und ge- gebenenfalls der Verwaltung des Bauteilelagers umfassen. Die Schal- tung kann meist innerhalb einer graphischen Benutzeroberfläche inter- aktiv konstruiert werden, die Programme erstellen daraus eine Netzwer- kliste, aus der auf formalisierte Weise das Gleichungssystem für die Si- mulation erstellt wird. Zu den Programmpaketen gehören umfangreiche Bibliotheken mit den technischen Daten von Bauteilen aller Art. Solche Parameterdateien, die zu bestimmten Simulationsprogrammen kompa- tibel sind (*Spice*), werden auch von den meisten Bauteileherstellern zur Verfügung gestellt.

Die in den Programmpaketen enthaltenen Simulationsprogramme gehen meist auf das ursprünglich in Berkeley entwickelte *Spice* zurück. Die Si- mulationsmöglichkeiten umfassen in der Regel Gleichspannungsanaly- sen mit festen oder veränderlichen Spannungen oder Strömen, Wechsel- spannungsanalysen mit variabler Frequenz (Frequenzgang einer Schal- tung – Bode-Diagramm) und Transientenanalysen zur Simulation des Zeitverlaufs bei beliebigen Spannungsformen oder zur Analyse von Ver- zerrungen. Viele der Programme können sowohl analoge wie auch digi- tale Simulationen durchführen, meist auch gemischt.

Im Praktikum empfehlen wir, mit LTSpice zu arbeiten, einem frei erhält- lichen EDA-Programm, das primär auf die Produkte von Linear Techno- logies abgestimmt ist, aber auch allgemeine Bauteile in seinen Biblio- theken enthält. Die Bibliotheken lassen sich beliebig erweitern, wenn man sich an die Möglichkeiten von LTSpice hält. Die verschiedenen Vor- gehensweisen sind in der Hilfe unter *Help Topics* ⇒ *FAQs* ⇒ *Third-party Models* beschrieben.

Die Bauelemente des Osnabrücker Elektronik-Praktikums sind in einer eigenen Bibliothek `uos05.lib` zusammengefasst.

Literatur und Links

- [1] F. S. Goucher, G. L. Pearson, M. Spark, G. K. Teal, W. Shockley. *Theory and Experiment for a Germanium p-n Junction*. Phys. Rev. **81**, 637 (1951).
- [2] <http://www.physik.uni-osnabrueck.de/kbetzler/gw/gw.html>.
- [3] <http://www.mathworks.com/support>.