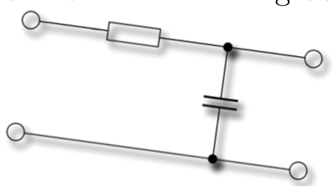

Bernd Huber, Fabian Steiner

Thema: Einführung in lineare Differentialgleichungen

1 Klassifizierung von Differengleichungen

DGL

- gewöhnliche DGL (ges. Funktion abhängig von einer Variablen)
- partielle DGL (ges. Funktion abhängig von mehreren Variablen)

Speziell bei gewöhnlichen DGLs sind ferner folgende Einteilungen wichtig:

Grad der Ordnung : Grad der höchsten vorkommenden Ableitung

linear/nicht linear : die zu integrierende Variable kommt lediglich mit der Potenz $n \leq 1$ vor

explizit/implizit : DGL ist/ist nicht nach der höchsten vorkommenden Ableitung aufgelöst

2 Lineare Differentialgleichungen 1. Ordnung

2.1 Allgemeine Form

Lineare DGLs 1. Ordnung besitzen folgenden allgemeine Form:

$$x'(t) + f(t)x = g(t) \tag{1}$$

Je nach gegebenen $g(x)$ lasssen sich nun weitere Fallunterscheidungen treffen, von denen ausgehend alsdann der passende Lösungsansatz gewählt werden kann.

2.2 Homogener Fall: $g(t) \equiv 0$

Gleichung 1 vereinfacht sich nun zu:

$$x'(t) + f(t)x = 0 \tag{2}$$

Wie sich leicht nachrechnen lässt, lautet die allgemeine Lösung:

$$\boxed{x(t) = C \cdot e^{-F(t)}}, \qquad \text{mit } F(t) = \int f(t)\mathrm{d}t, C \in \mathbb{R} \tag{3}$$

Um ein konkretes Anfangswertproblem $x(t_0) = x_0$ zu lösen, muss die Konstante C jener Bedingung angepasst werden, d.h. es ergibt sich

$$C = \frac{x_0}{e^{-F(t_0)}} \tag{4}$$

Autonomer Fall: $g(t) = A = \text{const.}$

$$x(t) = \frac{A}{f(t)} + C \cdot e^{-F(t)}, \qquad \text{mit } F(t) = \int f(t)\mathrm{d}t, A, C \in \mathbb{R} \tag{5}$$

Allgemeiner Fall

Die Berechnung des allgemeinen Falls für die Erregung $g(t)$ erfolgt durch eine Variation der Konstanten, d.h. es gilt nicht mehr $C = \text{const.}$, sondern $C \equiv C(x)$:

$$x(t) = \underbrace{\left(\int g(t) \cdot e^{F(t)}\mathrm{d}t\right) \cdot e^{-F(t)}}_{\text{zero state response}} + \underbrace{C \cdot e^{-F(t)}}_{\text{zero input response}} \qquad \text{mit } F(t) = \int f(t)\mathrm{d}t, C \in \mathbb{R} \tag{6}$$

Die Ergebnisse für die beiden hervorrigen Fälle können mittels (6) ebenfalls hergeleitet werden. Die Bestimmung der Konstante C erfolgt durch Einsetzen der Anfangsbedingungen und anschließendem Auflösen.
Eine Schaltungsanalyse erfolgt in ST2 in der Regel für den homogenen oder abschnittsweise konstanten Fall, sodass eine direkte Benutzung obige Formel in der Regel nicht notwendig ist. Nichtsdestotrotz sollte sie auf der Formelsammlung nicht fehlen.
Da die Betrachtung des abschnittsweisen konstanten Falles $g(t) = x(t_\infty) = \text{const.}$ eine besondere Bedeutung hat, soll an dieser Stelle auf diesen nochmals besonders eingegangen werden. Ausgehend von Gleichung (1) und (5) sowie der zusätzlichen Annahme $f(t) = \frac{1}{\tau} = \text{const.}$[1], lässt sich bei gegebenen Anfangswert $x(t_0) = x_0$ folgende Lösung für die Differentialgleichung angeben:

$$x(t) = x(t_\infty) + (x(t_0) - x(t_\infty)) \cdot e^{-\frac{t-t_0}{\tau}} \tag{7}$$

Diese Formel werden wir sehr oft bei der Analyse von linearen Schaltungen mit Reaktanzen benötigen und sollte daher – zusammen mit den Einschränkungen für ihren Gültigkeitsbereich – auswendig beherrscht werden.

3 Systeme linearer Differentialgleichungen 1. Ordnung mit konstanten Koeffizienten

Hierbei handelt es sich um Gleichungssysteme der folgenden Art:

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}v(t) \tag{8}$$

Je nach gegebenen $v(t)$ lassen sich wiederum mehrere Fälle unterscheiden:

- $v(t) \equiv 0$: homogenes System
- $v(t) = \text{const.}$: autonomes System

[1] dies entspricht der Tatsache, dass die Bauteilparameter (Kapazität, Induktivität) als zeitlich konstant angesehen werden können

Zunächst werden homogene Systeme betrachtet, da autonome Systeme unter der Annahme der Invertierbarkeit von $\mathbf{A}$ stets durch eine Koordinatentransformation auf ein homogenes System zurückgeführt werden kann.
Unter Punkt 1 wurde eine allgemeine Lösungsstrategie für lineare DGL 1. Ordnung vorgestellt. Entsprechend wäre es nun äußerst praktisch, diesen Ansatz auch für Systeme von linearen DGL 1. Ordnung mit konstanten Koeffizienten zu benutzen. Eine kleine Schwierigkeit besteht nun jedoch darin, dass man hierzu wohl die Matrix-Exponentielle $\exp(\mathbf{A}t)$ definieren müsste. Dieses Problem kann nun umgangen werden, in dem man sich auf eine der möglichen Definitionen der Exponentialfunktion besinnt. Wie wir im weiteren sehen werden, ist hierbei insbesondere der Weg über die Reihendarstellung besonders hilfreich:

$$e^{\mathbf{A}t} = \sum_{i=0}^{\infty} \frac{(\mathbf{A}t)^i}{i!} \tag{9}$$

Die hierbei auftretenden Operationen sind für Matrizen wohl definiert, sodass das nächste Ziel darin bestehen muss, das Matrizenprodukt $\mathbf{A}^i$ möglichst einfach in einer geschlossenen Form zu berechnen. Unter der Annahme einer möglichen Diagonalisierbarkeit[2] der Matrix $\mathbf{A}$ ist eine solche Berechnung relativ einfach.

3.1 Homogener Fall: $v \equiv 0$

3.1.1 Diagonalisierbarkeit der Systemmatrix

Ist die Systemmatrix $\mathbf{A}$ diagonalisierbar, so ist folgende Faktorisierung möglich:

$$\mathbf{A} = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^{-1} \tag{10}$$

$\mathbf{Q}$ bezeichnet hierbei die sog. Modalmatrix, deren Spalten dabei aus den linear unabhängigen Eigenvektoren $\mathbf{q}_1$, $\mathbf{q}_2$ der Systemmatrix $\mathbf{A}$ bestehen. Für $\mathbf{\Lambda}$ gilt: $\mathbf{\Lambda} = diag(\lambda_1, \lambda_2)$, wobei mit λ_1, λ_2 die Eigenwerte der Matrix $\mathbf{A}$ bezeichnet werden.
Mit den Überlegungen aus (9) und (10) ergibt sich nun:

$$\begin{aligned} e^{\mathbf{A}t} &= \sum_{i=0}^{\infty} \frac{(\mathbf{A}t)^i}{i!} = \mathbf{Q} \cdot \sum_{i=0}^{\infty} \frac{(\mathbf{\Lambda}t)^i}{i!} \cdot \mathbf{Q}^{-1} = \\ &= \mathbf{Q} \cdot \begin{pmatrix} \sum_{i=0}^{\infty} \frac{(\lambda_1 t)^i}{i!} & 0 \\ 0 & \sum_{i=0}^{\infty} \frac{(\lambda_n t)^i}{i!} \end{pmatrix} \cdot \mathbf{Q}^{-1} = \\ &= \mathbf{Q} \cdot \begin{pmatrix} e^{\lambda_1 t} & 0 \\ 0 & e^{\lambda_2 t} \end{pmatrix} \cdot \mathbf{Q}^{-1} \end{aligned} \tag{11}$$

Folglich besitzt das homogene, lineare DGL-System mit dem Anfangswert $\mathbf{x}(t_0) = \mathbf{x}_0$ die Lösung:

$$\mathbf{x}(t) = e^{\mathbf{A}t}\mathbf{x}_0 = \mathbf{Q} \cdot \begin{pmatrix} e^{\lambda_1 t} & 0 \\ 0 & e^{\lambda_2 t} \end{pmatrix} \cdot \mathbf{Q}^{-1} \cdot \mathbf{x}_0 = \tag{12}$$

$$= \mathbf{Q} \cdot \begin{pmatrix} e^{\lambda_1 t} & 0 \\ 0 & e^{\lambda_2 t} \end{pmatrix} \cdot \mathbf{c} = \tag{13}$$

$$= c_1 \mathbf{q}_1 e^{\lambda_1 t} + c_2 \mathbf{q}_2 e^{\lambda_2 t} \tag{14}$$

[2]hierzu müssen die Eigenvektoren linear unabhängig sein und Basis bilden

3.1.2 Jordan Normalform

Eine Diagonalisierung wie im vorangegangenen Abschnitt ist nicht möglich, wenn die algebraische Vielfachtheit der Eigenwerte mit der geometrischen Vielfachheit der Eigenvektoren nicht übereinstimmt. In diesem Fall lässt sich $\mathbf{A}$ also nicht auf die Art und Weise wie in (10) gezeigt faktorisieren. Stattdessen führt man die Faktorisierung $\mathbf{A} = \mathbf{Q}'\mathbf{\Lambda}'\mathbf{Q}'^{-1}$ durch, wobei $\mathbf{\Lambda}'$ nun keine reine Diagonalmatrix mehr ist, sondern sich als als $\mathbf{\Lambda}' = \lambda\mathbf{E} + \mathbf{N}$ mit einer nilpotenten[3] Matrix $\mathbf{N}$ darstellen lässt.
Ausgehend von dieser Zerlegung lässt sich die Matrix-Exponentielle $e^{t\mathbf{A}}$ wiederum relativ einfach berechnen:

$$\begin{aligned} e^{t\mathbf{A}} &= \sum_{i=0}^{\infty} \frac{(\mathbf{A}t)^i}{i!} = \mathbf{Q} \cdot \sum_{i=0}^{\infty} \frac{(\mathbf{\Lambda}t)^i}{i!} \cdot \mathbf{Q}'^{-1} = \\ &= \mathbf{Q} \cdot \sum_{i=0}^{\infty} \frac{t^i}{i!} \cdot \begin{pmatrix} \lambda^i & i\lambda^{i-1} \\ 0 & \lambda^i \end{pmatrix} \cdot \mathbf{Q} = \\ &= \mathbf{Q} \cdot \begin{pmatrix} \sum_{i=0}^{\infty} \frac{(\lambda t)^i}{i!} & \sum_{i=0}^{\infty} \frac{\lambda^{i-1} t^i}{(i-1)!} \\ 0 & \sum_{i=0}^{\infty} \frac{(\lambda t)^i}{i!} \end{pmatrix} \cdot \mathbf{Q}'^{-1} = \\ &= \mathbf{Q} \cdot \begin{pmatrix} e^{\lambda t} & te^{\lambda t} \\ 0 & e^{\lambda t} \end{pmatrix} \cdot \mathbf{Q}'^{-1} \end{aligned} \tag{15}$$

$$\mathbf{\Lambda} = \begin{pmatrix} \lambda_1 & 1 \\ 0 & \lambda_2 \end{pmatrix} \tag{16}$$

ist. Für die Modalmatrix $\mathbf{Q}'$ gilt in diesem Falle $\mathbf{Q}' = (\mathbf{q}_1'\mathbf{q}_2')$, wobei $\mathbf{q}_1'$ einer der Eigenvektoren der Systemmatrix $\mathbf{A}$ ist und sich $\mathbf{q}_2'$ berechnet durch:

$$\mathbf{q}_2' = \begin{pmatrix} -a_{12} \\ \frac{a_{12}-a_{22}}{2} - 1 \end{pmatrix} \tag{17}$$

Allgemeiner gesprochen verbirgt sich hinter diesem Vorgehen der Versuch, eine Basis des $\mathbb{C}^n$ aus Hauptvektoren zu finden.

3.1.3 Reelle Normalform

Treten komplexe Eigenwerte auf, so sind diese stets komplex konjugiert zu einander[4], sodass dementsprechend auch die zugehörigen Eigenvektoren stets linear unabhängig sind und daher eine Diagonalisierung möglich ist.
Hierdurch folgt demnach stets:

$$\begin{aligned} \lambda_1 &= \lambda_2^* = \lambda = \alpha + j\beta \\ \mathbf{q}_1 &= \mathbf{q}_2^* = \mathbf{q} \\ \mathbf{c} &= \mathbf{Q}^{-1}\mathbf{x}_0 = \begin{pmatrix} c_1 \\ c_2 \end{pmatrix} \Rightarrow c_1 = c_2^* = c \end{aligned} \tag{18}$$

Mit Hilfe von (14) lässt sich nun der Ansatz

[3] eine Matrix ist nilpotent, falls $\exists k \in \mathbb{N} : \mathbf{N}^k \equiv 0$
[4] folgt aus dem Fundamentalsatz über algebraische Gleichungen

$$
\begin{aligned}
\mathbf{x}(t) &= c_1\mathbf{q}_1 e^{\lambda_1 t} + c_2\mathbf{q}_2 e^{\lambda_2 t} = \\
&= c_1\mathbf{q}_1 e^{\lambda_2 t} + c_1^*\mathbf{q}_1^* e^{\lambda_1^* t} = \\
&= 2\mathrm{Re}\left(c_1\mathbf{q}_1 e^{\lambda_1 t}\right)
\end{aligned} \tag{19}
$$

aufstellen, der – wie leicht einzusehen ist – einen reellen Ausdruck darstellt.
Mit Hilfe der Eulerformel $e^{t(\alpha+j\beta)} = e^{\alpha t}(\cos(\alpha) + j\sin(\beta))$ und der Zerlegung $c_1 = \gamma + j\delta$ ergibt sich nun:

$$
\begin{aligned}
\mathbf{x}(t) &= 2e^{\alpha t}\mathrm{Re}\left((\gamma + j\delta)(\mathbf{q}_r + j\mathbf{q}_i)e^{j\beta t}\right) = \\
&= 2e^{\alpha t}\mathrm{Re}\left((\gamma + j\delta)(\mathbf{q}_r + j\mathbf{q}_i)(\cos(\beta t) + j\sin(\beta t))\right) = \\
&= 2e^{\alpha t}\left[\cos(\beta t)(\gamma\mathbf{q}_r - \delta\mathbf{q}_i) - \sin(\beta t)(\gamma\mathbf{q}_i + \delta\mathbf{q}_r)\right]
\end{aligned} \tag{20}
$$

Hieran lässt sich schön erkennen, dass im Fall von rein imaginären Eigenvektoren ($\alpha \equiv 0$) man als Lösung der DGL eine Superposition von harmonischen Schwingungen erhält – in allen anderen Fällen ist ein zusätzlicher Dämpfungsfaktor ($e^{\alpha t}$) vorhanden, der zu einem Aufschwingen ($\alpha > 0$) oder zu einem Abklingen ($\alpha < 0$) führt.

3.2 Autonomer Fall: $\mathbf{v} = \mathrm{const.}$

Wie bereits zur Einführung erwähnt, lässt sich der autonome Fall einfach durch eine Koordinatentransformation auf den homogenen Fall überführen, sofern $\mathbf{A}$ invertierbar ist:

$$
\begin{aligned}
\mathbf{x}' &= \mathbf{x} - \mathbf{x}_\infty, \qquad \text{mit } \mathbf{x}_\infty = -\mathbf{A}^{-1}\mathbf{B}\mathbf{v} \\
\dot{\mathbf{x}}' &= \mathbf{A}(\mathbf{x}' + \mathbf{x}_\infty) + \mathbf{B}\mathbf{v} = \\
&= \mathbf{A}(\mathbf{x}' - \mathbf{A}^{-1}\mathbf{B}\mathbf{v}) + \mathbf{B}\mathbf{v} = \\
&= \mathbf{A}\mathbf{x}'
\end{aligned} \tag{21}
$$

3.3 Allgemeiner Fall: $v = v(t)$

Wiederum analog zur Gleichung (6) lässt sich für ein System von Differentialgleichungen folgende Lösungsformel für den Fall einer allgemeinen Erregung aufstellen:

$$
\mathbf{x}(t) = \underbrace{\int e^{-\mathbf{A}t}\mathbf{B}\mathbf{v}(t)\mathrm{d}t \cdot e^{\mathbf{A}t}}_{\text{zero state response}} + \underbrace{e^{\mathbf{A}t}\mathbf{x}_0}_{\text{zero input response}} \tag{22}
$$

4 Numerische Behandlung von DGL mit Hilfe von Matlab

Ein wichtiges Hilfsmittel zur Kontrolle und insbesondere zum Verständnis von Schaltungen (z.B. Oszillatorschaltungen) besteht darin, sich entsprechende Phasenportraits und den zeitlichen Verlauf der Systemgrößen anzuschauen. In der Regel unterliegen diese bei Schaltungen mit Reaktanzen Differentialgleichungen, die nicht unmittelbar auf eine einfache Art und Weise visualisierbar sind. Aus diesem Grunde sollen euch nachfolgend entsprechende Möglichkeiten aufgezeigt werden. Hierbei geht es jeweils nicht um eine analytische Lösung (die in ST2 ausschließlich gefragt sind), sondern um eine rein numerische Herangehensweise, mit der zum Einen sogar eine größere Vielfalt an Aufgabenstellungen bearbeitet werden kann und zum Anderen auch vollkommen ausreichen, um einen Eindruck über die Eigenschaften eines Systems zu erhalten.

4.1 Was ist Matlab?

Bei Matlab handelt es sich um *Computer-Algebra-System (CAS)*, dessen Stärken insbesondere in der Linearen Algebra liegen[5].
Neben Matlab existiert auch nach das unter einer freien Lizenz verfügbare Octave, das im Grunde die gleiche Funktionalität wie Matlab bietet. Leider unterscheiden sich sich jedoch gerade in Hinsicht auf die verfügbaren Funktionen im Bereich der gewöhnlichen DGLs, sodass wir uns an dieser Stelle nur auf Matlab beschränken wollen.
Weitere Informationen zu den beiden Paketen (und deren Installation unter den jeweiligen Betriebssystemen) findet sich unter den folgenden Adressen:

- `http://octave.org`
- `http://mathworks.com`

Da eine allgemeine Einführung in die Benutzung von Matlab den Umfang dieses Dokuments sprengen würde, sei an dieser Stelle nur auf einige Tutorials bzw. Einführungen verwiesen – besonderes empfehlenswert sind hierbei die Einführungsfolien der Uni Ulm. Bei konkreten Fragen besteht natürlich auch stets die Möglichkeit sich an die Tutoren zu wenden.

- `http://www.mathematik.uni-ulm.de/numerik/teaching/ss08/WimaPrak`
- `http://www-m3.ma.tum.de/m3old/ftp/matlab.pdf`

4.2 Numerische Lösen von DGL

Beide vorgestellten Systeme bringen eine Reihe von Funktionen mit, die eine numerische Berechnung von Lösungen von Differentialgleichen ermöglichen. Je nach Art der Differentialgleichung und deren besonderen Eigenschaften, wird man unterschiedliche Funktion der `ode??` Klasse benutzen. Die zwei Fragezeichen stehen dabei für die genaue Art des Verfahrens, z.B.

- `ode15`
- `ode23`
- `ode45`
- ...

Ein guter Ansatzpunkt ist dabei (für alle ST2 relevanten Anwendungen) stets die Funktion `ode45`, die ein Runge-Kutta-Verfahren 4. Ordnung[6] implementiert.
Die Signatur der Funktion `ode45` sieht wie folgt aus (siehe auch `help ode45`):

```
[TOUT,YOUT] = ODE45(ODEFUN,TSPAN,Y0,OPTIONS)
```

Die Bedeutung ist dabei die folgende:

- Eingangsparameter:
  - `odefun`: Function Handle, das auf die auszuwertende Funktion zeigt
  - `tspan`: Intervall, für das die DGL gelöst werden soll, hierbei soll gelten `tspan(1)` $= t_0$ mit `y0(1)` $= \mathbf{y}_0$, sodass die Anfangswertbedingung(en) $\mathbf{y}(t_0) = \mathbf{y}_0$ erfüllt ist

[5] im Vergleich zu anderen derartigen Systemen, wie Maple oder Mathematica, die hauptsächlich zum analytischen Lösen Einsatz finden
[6] mehr dazu in HM4

– `y0`: Vektor der Anfangswerte (s.o.)
– `options`: Struktur mit weiteren Optionen für den Lösungsvorgang der DGL (erzeugt mittels `odeset`

- Ausgangsparameter:
  – `tout`: Zeitvektor
  – `yout`: Ergebnisvektor

Nachdem man nun Matlab gestartet hat und den Prompt erhält, kann nun begonnen werden. Im Folgenden soll als einführendes Beispiel die Differentialgleichung des Van-der-Pol Oszillators untersucht werden[7]:

$$\ddot{x} - \varepsilon(1 - x^2)\dot{x} + x = 0, \qquad \varepsilon \geq 0 \tag{23}$$

Da mit den Matlab-Funktionen jedoch nur DGL 1. Ordnung behandelt werden können, müssen wir (23) durch Substitution auf eine System von zwei DGL 1. Ordnung bringen. Diese Vorgehen ist bei höheren DGL stets möglich und kann unter Umständen den Lösungsansatz erheblich vereinfachen:

$$\begin{aligned} a_n y^{(n)} + a_{(n-1)} y^{(n-1)} + \ldots + a_1 y' + a_0 y = 0 \\ x_1 &= y \\ \dot{x}_1 &= x_2 \\ \dot{x}_2 &= x_3 \\ &\vdots \\ \dot{x}_n &= -\frac{a_{n-1}}{a_n} x_n - \frac{a_{n-2}}{a_n} x_{n-1} - \ldots - \frac{a_1}{a_n} x_2 - \frac{a_0}{a_n} x_1 \end{aligned} \tag{24}$$

Diese Methodik sollte man sich gut verinnerlichen, da sie auch in ST2 bereits des Öfteren in Prüfungen zu verwenden war, ohne explizit in der Vorlesung besprochen zu werden.
Bezogen auf unser Beispiel setzen wir also

$$\begin{aligned} x &= u_1 \\ \dot{x} &= u_2 \end{aligned}$$

und erhalten somit:

$$\begin{aligned} \dot{u_1} &= u_2 \\ \dot{u_2} - \varepsilon(1 - u_1^2)u_2 + u_1 &= 0 \end{aligned}$$

Ausgehend hiervon kann nun mit Matlab gearbeitet werden. `> >` symbolisiert den Eingabeprompt, alles nach den Prozentzeichen `%` sind Malab-Kommentare.

```
>> f = @(t, u)[u(2); 10*(1-u(1)^2)*u(2)-u(1)];
>> [T, Y] = ode45(f, [0 50], [2; 0]);
>> plot(T, Y(:, 1))
>> plot(T, Y(:, 2))
```

[7]hierbei handelt es sich um eine nichtlineare, homogene DGL 2. Ordnung

1. Hier definieren wir eine sog. Inline-Funktion (erkennbar an dem einleitenden @-Zeichen), die zwei Parameter t, u übernimmt. Anschließend wird in den eckigen Klammern das System der DGLs eingetragen – jede Zeile wird dabei von einem Semikolon getrennt.

2. Aufruf der DGL-Solvers, mit der Inline Funktion als erstes Argument. Anschließend wir die Zeitspanne [1 50] angegeben. Abschließend werden noch die Anfangswerte übergeben, sodass gilt: $u_1(1) = 0, u_2(1) = 0$.

3. Plot von $u_1(t)$ im Bereich [1 50]

4. Plot von $u_2(t)$ im Bereich [1 50]

Als Ausgabe hiervon erhält man alsdann die Plots:

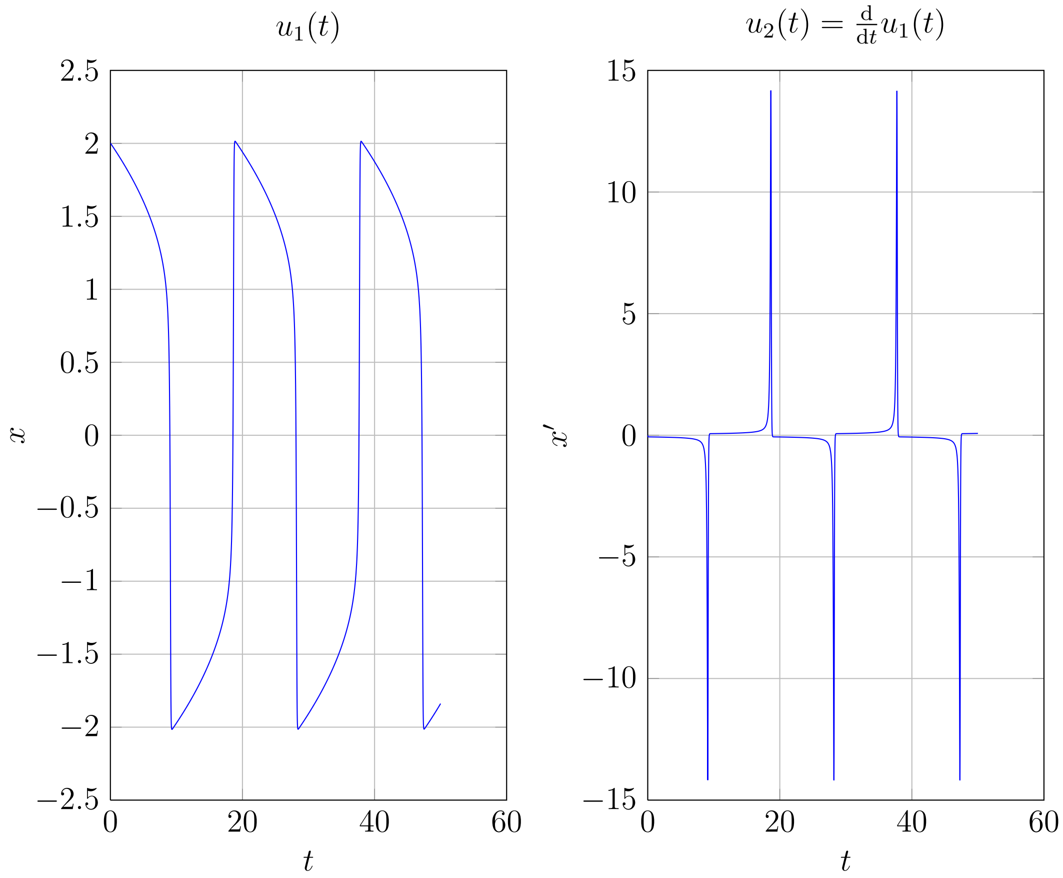

Abbildung 1: Zeitlicher Verlauf der Plots

Zudem soll nun auch nochmal eine lineare Schaltung zweiten Grades untersucht werden:

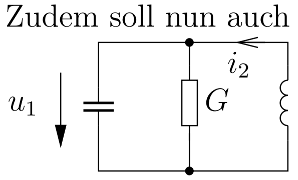

Abbildung 2: Lineare Schaltung 2. Grades

Die Zustandsgrößen der nebenstehenden Schaltung sind u_1 und i_2, entsprechend müssen also zum Aufstellen der Systemmatrix die Gleichungen für $\dot{u}_1(u_1, i_2)$ und $\dot{i}_2(u_1, i_2)$ gefunden werden. Durch KCL/KVL und den beiden Bauteilgleichungen stößt man so auf:

$$\dot{u}_1 = -\frac{G}{C}u_1 + \frac{i_2}{C} \tag{25}$$
$$\dot{i}_2 = -\frac{u_1}{L} \tag{26}$$
$$\Rightarrow \mathbf{A} = \begin{pmatrix} -\frac{G}{C} & \frac{1}{C} \\ -\frac{1}{L} & 0 \end{pmatrix} \tag{27}$$

Für den Fall, dass die Bauteilwerte nun als $G = 2S, C = 1F, L = 1/2H$ gegeben sind, folgt für $\mathbf{A}$:

$$\mathbf{A} = \begin{pmatrix} -2 & 1 \\ 2 & 0 \end{pmatrix}$$

Eine Berechnung der Eigenwerte über die Lösung des charakteristischen Polynoms $\det(\mathbf{A} - \lambda\mathbf{E}) = 0$ ergibt die zwei komplex konjugierten Eigenwerte $\lambda_1 = \lambda_2^* = -1 + j$. Aus der Lösungstheorie von linearen Differentialgleichungssystemen 1. Ordnung mit konstanten Koeffizienten wissen wir, dass das Phasenportrait ein stabiler Strudel ist ($\alpha < 0$) und das die zeitlichen Verläufe der Zustandsvariablen wohl eine harmonische Schwingung darstellen.
Die obige Vorbetrachtung hätten wir ebenfalls wieder mit Matlab durchführen können:

```
>> A = [-2 1; -2 0]
>> [EVec, EVal] = eig(A)

EVec =

   0.4082 - 0.4082i   0.4082 + 0.4082i
   0.8165             0.8165


EVal =

  -1.0000 + 1.0000i        0
        0            -1.0000 - 1.0000i
```

Die somit erhaltenen Ergebnisse stimmen mit den obigen von uns analytisch Ermittelten überein. Ein Hinweis noch zur Variable `EVec`: diese enthält die normierten Eigenvektoren zu den Eigenwerten, die in der Variable `EVal` abgespeichert wurden. Da uns nun noch zusätzlich die zeitlichen Verläufe und das Phasenportrait für die Anfangswerte $u_1(0) = 1V, i_2(0) = 0A$ interessiert, wollen wir diese auch noch berechnen lassen:

```
>> f = @(t, x)[-2*x(1)+ 1*x(2); -2 * x(1) + 0 * x(2)];
>> [T, Y] = ode45(f, [0 6], [-2; -2]);
>> plot(T, Y(:, 1))
>> plot(T, Y(:, 2))
```

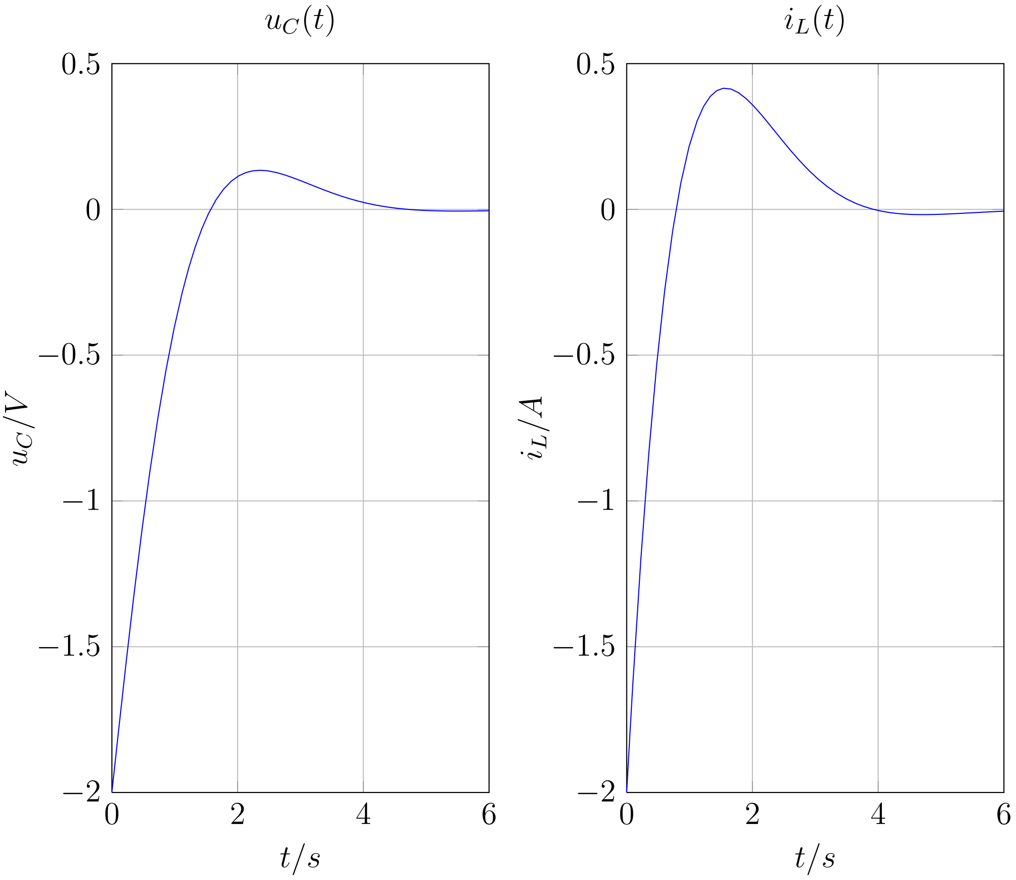

Abbildung 3: Zeitlicher Verlauf der beiden Zustandsgrößen $u_1(t)$ und $i_2(t)$