RST2 - Epilog zur Übung 1

WS19/20

Vorwort

- Die Epiloge dienen als **freiwillige Ergänzung** zu den Übungen, die lediglich alle zwei Wochen stattfinden, und sollen so eine Möglichkeit zum selbstständigen Rechnen und Umsetzen der Übungsthemen geben. Diese Aufgaben werden zwar in den Übungen nicht explizit besprochen, allerdings können einzelne Punkte durch Fragen zu Beginn oder am Schluss des Übungstermins aufgegriffen werden.

  Eine Bearbeitung in gemeinsamen Lerngruppen erscheint von Vorteil.

- Dabei wird auch eine Einführung in die **numerische Umsetzung** der jeweiligen Verfahren gegeben. Dieses Sammeln von Programmiererfahrung ist besonders als Begleitung zum Praktikum gedacht. Zudem soll aufgezeigt werden, wie all die Berechnungen per Hand für die spätere praktische Arbeit zu implementieren sind. Häufig können die Ergebnisse aus den Übungsaufgaben so direkt überprüft werden.

- Zur Programmierung der Code-Beispiele empfehlen wir die Nutzung numerischer **Software-Werkzeuge** wie MATLAB bzw. Octave. Da MATLAB kostenpflichtig ist, haben Sie einen freien Zugang dazu in den Rechnerlaboren des Rechenzentrums. Die Freeware GNU Octave ist kostenfrei zugänglich und hat eine stark an MATLAB angelehnte Syntax. Installationsanweisungen finden Sie auf den jeweiligen Webseite. Für eine kurze Einarbeitung in die Umgebungen raten wir zum Heranziehen eines MATLAB-Tutorials bzw. eines Octave-Tutorials. Es wird empfohlen für Octave das *control* und ggf. *symbolic* Paket für die nötigen Funktionalitäten mit einzubinden. Auf unserer Webseite gibt es zudem eine MATLAB-Kurzreferenz, in der viele relevante Befehle aufgelistet sind, die ebenso in Octave funktionieren.

  Es gibt natürlich noch weitere mögliche Softwareanbieter wie etwa Python, Scilab, etc., die selbstverständlich benutzt werden können, zu denen wir allerdings keine Code-Beispiele und Verweise vorbereiten.

- Weiterführende Themenbereiche und Literaturempfehlungen aus den Epilogen, die nicht in der Vorlesung vorkommen, dienen zum fortgeschrittenen Selbststudium für Interessierte und sind explizit **NICHT prüfungsrelevant**. Häufig ergeben sich durch aufgeführte Onlinreferenzen und Links auch noch weitere Fassetten der Thematiken.

Ergänzungsaufgaben

Aufgabe 1: Matrix-Exponentialfunktion (Wiederholung theoretischer Grundlagen)

Machen Sie sich vor der Bearbeitung mit dem Zusatzblatt Mathematische Grundlagen / Abschnitt 4 (*Besondere Matrizen*) vertraut. Zeigen Sie dann folgende Eigenschaften der Matrix-Exponentialfunktion:

(a) Schreiben Sie die Matrix-Exponentialfunktion $e^{A\,t}$ als (endliche) Summe von Potenzen von A.

Hinweis: Verwenden Sie den Satz von Cayley-Hamilton.

(b) Die folgende Reihe der Matrix-Exponentialfunktion konvergiert absolut:

$$e^{A\,t} := \sum_{i=0}^{\infty} \frac{(A\,t)^i}{i!}\,.$$

(c) Zeigen Sie mit dem Ergebnis von eben, daß für quadratische Matrizen $A, B \in \mathbb{R}^{n\times n}$ gilt:

$$e^A\, e^B = e^{A+B} \quad \Longleftrightarrow \quad A\,B = B\,A\,.$$

Hinweis: Es gilt die Cauchysche Summationsregel. D.h. sind $\sum_{i=0}^{\infty} a_i$ und $\sum_{j=0}^{\infty} b_j$ beide absolut konvergente Reihen, dann ist auch deren Produkt absolut konvergent und es gilt:

$$\left(\sum_{i=0}^{\infty} a_i\right)\left(\sum_{j=0}^{\infty} b_j\right) = \sum_{k=0}^{\infty} c_k \quad \text{mit} \quad c_k = \sum_{n=0}^{k} a_n\, b_{k-n}\,.$$

Folgern Sie daraus, das für beliebige $A \in \mathbb{R}^{n\times n}$ und für beliebige Zeiten $s, t \in \mathbb{R}$ gilt:

$$e^{A\,s}\, e^{A\,t} = e^{A\,(s+t)}\,.$$

(d) Berechnen Sie die Fundamentallösung der Differentialgleichung

$$\dot{x} = Ax, \quad x(0) = x_0, \;\; x(t) \in \mathbb{R}^n, A \in \mathbb{R}^{n\times n}$$

unter Zuhilfenahme der Zustandstransformation $x = Tz$ und der Annahme, daß A nur den $n-$fachen Eigenwert λ mit geometrischer Vielfachheit 1 besitzt.

Hinweis: Transformieren Sie A auf Jordan-Normalform..

(e) Für alle quadratischen Matrizen $A \in \mathbb{R}^{n\times n}$ und für alle endlichen Zeiten $|t| < \infty$ existiert die Inverse von $e^{A\,t}$. Sie lautet:

$$\left(e^{A\,t}\right)^{-1} = e^{-A\,t}\,.$$

Aufgabe 2: Jordan-Zerlegung berechnen

Bestimmen Sie für die Folgenden gegebenen Matrizen die jeweilige Matrixexponentialfunktion e^{At}. Transformieren Sie die Matrix A dazu auf Jordannormalform $J = T^{-1}AT$.

(i) $A = \begin{pmatrix} 2 & 1 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 2 \end{pmatrix}$, *(ii)* $A = \begin{pmatrix} 1 & 2 \\ 3 & 2 \end{pmatrix}$, *(iii)* $A = \begin{pmatrix} -1 & 0 \\ 1 & -1 \end{pmatrix}$,

(iv) $A = \begin{pmatrix} -9 & 1 & 0 \\ -23 & 0 & 1 \\ -15 & 0 & 0 \end{pmatrix}$, *(v)* $A = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}$, *(vi)* $A = \begin{pmatrix} 5 & 9 \\ -1 & -1 \end{pmatrix}$,

(vii) $A = \begin{pmatrix} 1 & -3 & -2 \\ -1 & 1 & -1 \\ 2 & 4 & 5 \end{pmatrix}$, *(viii)* $A = \begin{pmatrix} -3 & 0 & 0 \\ 0 & -\frac{1}{2} & 1 \\ 0 & -1 & -\frac{1}{2} \end{pmatrix}$, *(ix)* $A = \begin{pmatrix} -2 & 2 & 1 \\ -7 & 4 & 2 \\ 5 & 0 & 0 \end{pmatrix}$,

Hinweis: Im Fall komplexer Eigenwerte, nutzen Sie die Euler-Identität $e^{j\varphi} = \cos(\varphi) + j\sin(\varphi)$ und deren Folgerungen.

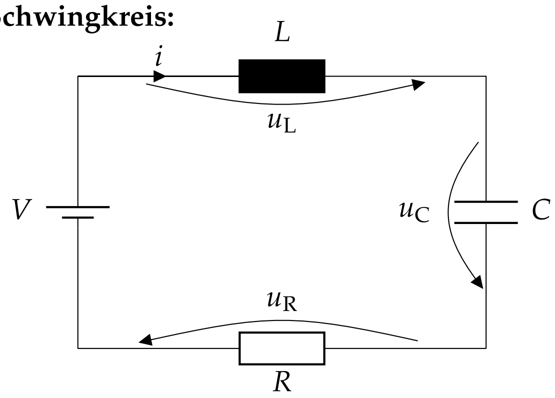
Aufgabe 3: Modellbildung & Zustandsraumdarstellung

Bilden Sie für die folgenden Systemskizzen (a) - (c) lineare mathematische Modelle zur Beschreibung und bringen Sie diese in die Zustandsraumdarstellung

$$\begin{aligned} \dot{x}(t) &= A(t)x(t) + B(t)u(t) \\ y(t) &= Cx(t) \end{aligned}\,,$$

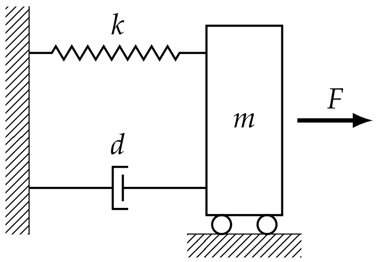
indem Sie den Eingang u und die inneren Zustände x sowie einen möglichen Ausgang $y(t)$ identifizieren.

(a) **Schwingkreis:**



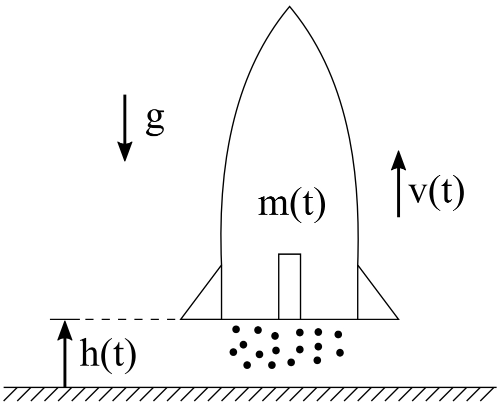
Nutzen Sie die *Kirchhoffschen* Sätze sowie gängige lineare Beziehungen für die Energiespeicher in Spule und Kondensator (Vergleich mit Übung 1 / Aufgabe 2).

(b) **Feder-Masse-Dämpfer System:**



Nutzen Sie das zweite *Newtonsche* Gesetz und bringen Sie entstehende Differentialgleichung auf ein System 1. Ordnung. Die Position des Wagens wird durch die Variable $x(t)$ beschrieben.

(c) **Raketenstart:** (zeitvariant)



Wegen des signifikanten Massenverlust durch die Treibstoffverbrennung sollte die ursprüngliche Form des 2. *Newtonschen* Hauptsatzes

$$\frac{d}{dt}(m(t)v(t)) = \sum F_i$$

angewendet werden. Die Verbrennungsrate $\dot{m}(t) = u_0 < 0$ ist dabei konstant mit dem Starttreibstoff und -gewicht m_0. Die Antriebskraft kann mit $F_a = v_a(t)\dot{m}(t)$ angesetzt werden, wobei die relative Abgasgeschwindigkeit $v_a(t)$ eingestellt werden kann.

Numerische Betrachtungen

Lösen Sie die folgenden Aufgaben mithilfe numerischer Simulationssoftware (z.B. MATLAB oder Octave). Nutzen Sie dazu auch die jeweilige umfangreiche Dokumentation dieser Werkzeuge. Einige grundlegende Funktionalitäten werden durch Beispiele in Code-Kästen erklärt.

1. Berechnen Sie die Jordannormalformen zu den A Matrizen aus der Ergänzungsaufgabe 2 algorithmisch und überprüfen Sie so Ihre Rechnungen. Dazu werden folgende Funktionen empfohlen

```
% Algebraic functions:
[V,D] = eig(A); % gives eigenvectors as columns in V and eigenvalues ...
    in diagonal form in D
Z = null(lambda*eye(n)-A); % calculates null space basis (kern) of a ...
    matrix
X = linsolve(A,B); % solve linear systems of equations A*X=B

```

```
% Symbolic calculations:
syms v1 v2 % define symbolic variables
sol = solve(v1-v2==0); % solve defined equation
```

Hinweis: In der MATLAB-Kurzreferenz können Sie die Grundregeln zum Umgang und für die Erstellung von Matrizen einsehen.

Zum Prüfen der Ergebnisse kann die in der *Symbolic Math Toolbox* verfügbare `jordan` Funktion genutzt werden:

```
As = sym(A); % convert to symbolic
[V,J] = jordan(As);
```

Damit wird die Jordanzerlegung direkt über symbolische Algorithmen berechnet.

2. Bestimmen Sie die Matrixexponentialfunktion e^{At} für Ergänzungsaufgabe 2 *(iii)*, *(v)* und *(ix)*. Plotten Sie die entsprechende Lösung $x(t) = e^{At}x_0$ als Zeitreihe über den Zeitvektor $t \in \{0, 0.01, 0.02, \ldots, 10\}$. Machen Sie sich zunächst den unterschied zwischen den folgenden Versionen der Exponentialfunktionen deutlich:

   ```
   e1 = exp(A);
   e2 = expm(A);
   ```

   Mit den Befehlen `help` und `doc` können Sie genaue Informationen zur Arbeitsweise und Syntax von Funktionen mit zugehörigen Beispielen einsehen. Es wird empfohlen, sich mit den Funktionalitäten und den Optionen der `plot` Funktion vertraut zu machen:

   ```
   help plot
   doc plot
   ```

   Nutzen Sie für die endgültige Berechnung der entsprechenden Lösung selbstgewählte Anfangswerte $x_0 \in \mathbb{R}^n$.

3. Ziehen Sie im Folgenden das Modell des Tiefsetzstellers aus Übung 1 / Aufgabe 2 heran. Nutzen Sie dafür zunächst die folgenden Parameter:

   $$L = 633\,H,\ C = 40\,\mu F,\ R_L = 700\,\Omega\,.$$

   Simulieren Sie zuerst das System, indem Sie die Differentialgleichung $\dot{x} = Ax + Bu$ durch numerische Lösungsverfahren integrieren. Dazu werden Funktionen der folgenden Form empfohlen:

   ```
   [t,xsol] = ode45(dynamic_function,timerange,initial_state)
   ```

   Plotten Sie die Ergebnisse $x(t)$ für die Simulationsparameter $x_0 = [0,0]^T, t_f = 8s, T_s = 0.01s$ (finale Simulationszeit t_f, Abtastschritte T_s) sowie ein Eingangssignal in zwei verschiedenen Szenarien:

   $$u(t) = \begin{cases} 10V & ,\ t \geq 0 \\ 6V\sin(12\pi t) & ,\ t \geq 0 \end{cases}.$$

   Hinweis: Hinterlegen Sie das Eingangssignal als separate Funktion.

   Um den Einfluss des Lastwiderstands R_L auf die Dynamik zu beurteilen, soll die Verschiebung der Eigenwerte veranschaulicht werden. Erstellen Sie dazu eine Struktur, die die Eigenwerte $\lambda_{1/2} = \text{eig}(A)$ für ansteigende Widerstandswerte $R_L \in \{100\,\Omega, 200\,\Omega, \ldots, 10\,k\Omega\}$ enthält und plotten Sie diese unter folgender Hilfestellung:

```
        hold on % use same figure without overwriting
        plot(real(lambda),imag(lambda),'*') % plot eigenvalues in Gauss plane ...
            as star dot
```

Validieren Sie so die Beobachtungen aus der Übungsdiskussion.

Verweis auf Klausuraufgaben

- Jordannormalform berechnen:
  - Übungsklausur 3 (Sommer 2011) Aufgabe 3 (b,c)
  - Übungsklausur 5 (Sommer 2012) Aufgabe 3
  - Übungsklausur 9 (Sommer 2014) Aufgabe 2 (a,b)
  - Übungsklausur 10 (Winter 2015/16) Aufgabe 2 (a,b)
  - Übungsklausur 11 (Sommer 2016) Aufgabe 1 (a,b)
  - Übungsklausur 12 (Winter 2016/17) Aufgabe 2 (diskretes System)
  - Übungsklausur 15 (Sommer 2018) Aufgabe 2 (a)

Weiterführende Literatur & Links

- Modellierungsbeispiele: [1] (Kapitel 2), [3] (Kapitel 2)
- Übersicht zur Jordannormalform: [1] (Kapitel 3.2), [2] (Kapitel 3.2)
- Extrablatt: Wiederholung Mathematischer Grundlagen
- MATLAB Tutorial: Zustandsraumrealisierung
- **Youtube Links** (kurze Erklärungen)
  - Jordanzerlegung - Überblick
    `https://www.youtube.com/watch?v=83SgQJekeuk`
  - Jordanzerlegung - Verfahren (Video-Reihe)
    `https://www.youtube.com/watch?v=hPAQdmEPU_k`
  - Zustandsraum - Intuition (englisch)
    `https://www.youtube.com/watch?v=hpeKrMG-WP0`
  - Zustandsraum - Beispiel (bayrisch)
    `https://www.youtube.com/watch?v=u-brazrdKBg`

$\rightarrow$ ermöglicht andere Perspektive und einfache Erläuterung der Thematik

[1] Ludyk, G.: *Theoretische Regelungstechnik 1 & 2*. Springer, 1995

[2] Olsder, G. ; Woude, J. van d.: *Mathematical Systems Theory*. 3. Auflage. VSSD, 2004

[3] Rugh, W.: *Linear System Theory*. 2. Auflage. Prentice Hall, 1996