Matlab-Übungen zu
Deterministische Signale und Systeme
5. Übung

1. Aufgabe

In dieser Aufgabe geht es um grundlegende Techniken, die bei der Laplacetransformation Anwendung finden. Um die Aufgabe lösen zu können, sollten Sie sich über den Matlab-Befehl `residue` informieren (wie Sie die Hilfe nutzen, wissen Sie inzwischen vermutlich). *Hinweis: Die Ergebnisausgabe von Matlab und Octave unterscheidet sich hier geringfügig, beachten Sie daher bitte die Funktionsweise des verwendeten Programms.*

1.1. Gegeben sei die Funktion

$$f(p) = \frac{-6p^5 - 10p^4 - p^3 + 12p^2 + 6p - 2}{2p^3 + 4p^2 + 3p}.$$

Berechnen Sie die Polstellen von $f(p)$ und wenden Sie eine Partialbruchzerlegung an. Überprüfen Sie Ihr Ergebnis mit der `residue`-Funktion aus Matlab.

1.2. Plotten Sie die Polstellen der Funktion

$$g(p) = \frac{5p^2 + 10p + 8}{p^3 + 3p^2 + 4p + 2}.$$

Überprüfen Sie das Ergebnis durch Rechnung auf dem Papier.

Hinweis: Sie können den Plot-Befehl anweisen, nur einzelne Punkte zu setzen, indem Sie ihm eine entsprechende Formatierungsanweisung als drittes Argument übergeben, etwa `'o'`.

1.3. Schreiben Sie eine Funktion `poldiag`, der Sie ein Zähler- und ein Nennerpolynom übergeben und die die Polstellen für diese Funktion zeichnet. Achten Sie dabei auf eine korrekte Achsenbeschriftung. Der dargestellte Bereich soll dabei immer dem 1,2-fachen des höchsten vorkommenden Real- oder Imaginärteils entsprechen.

2. Aufgabe

Mit den Befehlen `mesh` und `surf` lassen sich Funktionen dreidimensional darstellen. Dabei erwartet Matlab, dass den Befehlen drei Matrizen übergeben werden. Die erste übergebene

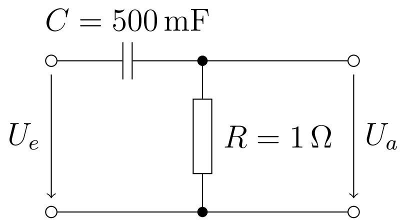

Abbildung 1: Schaltung

Matrix enthält x-Werte und die zweite Matrix y-Werte so, dass sich aus an der gleichen Stelle in den Matrizen liegende Werte Wertepaare ergeben, zu denen ein zugeordneter z-Wert an der entsprechenden Stelle in einer dritten Matrix zu finden ist.

Will man etwa die Funktion $z(x, y) = (x + y)^2$ von -1 bis 1 darstellen, wobei in beide Richtungen jeweils 5 Werte gegeben sein sollen, so benötigt man die Matrizen

$$X = \begin{pmatrix} -1 & -0{,}5 & 0 & 0{,}5 & 1 \\ -1 & -0{,}5 & 0 & 0{,}5 & 1 \\ -1 & -0{,}5 & 0 & 0{,}5 & 1 \\ -1 & -0{,}5 & 0 & 0{,}5 & 1 \\ -1 & -0{,}5 & 0 & 0{,}5 & 1 \end{pmatrix} \quad \text{und} \quad Y = \begin{pmatrix} -1 & -1 & -1 & -1 & -1 \\ -0{,}5 & -0{,}5 & -0{,}5 & -0{,}5 & -0{,}5 \\ 0 & 0 & 0 & 0 & 0 \\ 0{,}5 & 0{,}5 & 0{,}5 & 0{,}5 & 0{,}5 \\ 1 & 1 & 1 & 1 & 1 \end{pmatrix}$$

und kann dann mit

```
>> Z = (X+Y).^2;
```

die Funktion $z(x, y)$ in der Matrix `Z` abbilden. Die Matrizen `X` und `Y` lassen sich sehr einfach mit der Funktion **`meshgrid`** erzeugen, der Sie je einen Vektor mit x- und y-Werten übergeben. Die beiden Matrizen aus dem Beispiel lassen sich etwa mit

```
>> [X, Y] = meshgrid(-1 : .5 : 1, -1 : .5 : 1)
```

erzeugen.

Gegeben sei die Schaltung aus Abbildung 1 sowie die Laplace-Übertragungsfunktion

$$G(p) = \frac{1}{p + 2 + \frac{1}{p}}.$$

Der Kondensator ist zum Zeitpunkt $t = 0$ ungeladen.

2.1. Bestimmen Sie die Laplace-Übertragungsfunktion

$$F(p) = \frac{U_a(p)}{U_e(p)}$$

der Schaltung aus Abbildung 1.

2.2. Stelle Sie die beiden Übertragungsfunktionen grafisch in einem Bereich von $\sigma \in [-5, 5]$ und $\omega \in [-5, 5]$ dar.

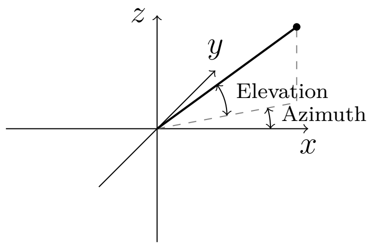

Abbildung 2: Azimuth und Elevation.

2.3. Mit dem Befehl `view` können Sie die »virtuelle Kamera« des 3D-Plots bewegen. Dem Befehl wird dabei ein Vektor mit zwei Elementen übergeben. Der erste Wert bestimmt den Azimuth, also den Horizontalwinkel der Kamera, der zweite Wert die Elevation, also den Vertikalwinkel (siehe auch Abbildung 2).

   Mit den Befehlen `xlim`, `ylim` und `zlim` können sie wie beim 2D-Plot den gewünschten Achsenausschnitt wählen.

   Stellen Sie mit diesen Hilfsmitteln die Fourier-Übertragungsfunktion des Systems aus Abbildung 1 ausgehend von der Laplace-Übertragungsfunktion dar.

2.4. Verwenden Sie die Technik aus der vorherigen Teilaufgabe, um herauszufinden, um was für ein System es sich bei $G(p)$ handelt (Hoch-, Tief-, Bandpass oder Bandsperre).

2.5. Fügen Sie $G(p)$ eine Nullstelle bei $p = 0$ hinzu. Um was für ein System handelt es sich nun? Zeichnen Sie eine Schaltung, die dieses System realisiert.

3. Aufgabe

Diese Aufgabe ist leider nur in Matlab und nicht in Octave lösbar, da Octave die GUI-Funktionalität von Matlab fehlt.

Für diese Aufgabe gibt es ein Matlab-GUI[1] und damit ein vorgegebenes Framework, dessen Funktion Sie vervollständigen sollen. Die Dateien finden Sie zum Herunterladen auf der Web-Seite zur Vorlesung.

Zum GUI gehören die Datei `faltung.fig`, die die grafische Darstellung enthält und die Datei `faltung.m`, die den Quelltext zum GUI enthält. Zum Starten des GUI führen Sie bitte `faltung.m` mit Matlab aus.

Die Datei `faltung.m` versucht dabei, auf eine Funktion `falt.m` im gleichen Verzeichnis zurückzugreifen, die allerdings fehlt (und die Sie schreiben sollen, Sie haben ja in einer vergangenen Übung schon einmal die Faltung implementiert, wissen also, wie das geht).

Die Funktion `falt` soll dabei die Signatur

```
function out = falt(t0, t, siga, tau, sigb)
```

haben. Die übergebenen Parameter sind

`t0` der aktuelle Zeitpunkt,

[1] Graphical User Interface

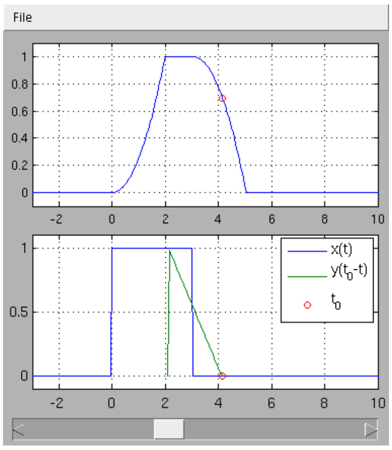

Abbildung 3: GUI zur Faltung.

`t` die »globale« Zeitachse und gleichzeitig die Zeitachse von Signal `siga`,

`siga` das eine Signal,

`tau` die Zeitachse des Signals `sigb` und

`sigb` das zweite Signal.

Schreiben Sie die Funktion so, dass immer für die übergebenen Werte der Wert der Faltung zum Zeitpunkt `t0` berechnet wird. Beachten Sie dabei, dass die Zeitachsen trotz scheinbar »runder« Werte kleine Abweichungen von diesen Werten aufweisen können, ein direkter Vergleich wie `find(t==0.1)` muß also nicht unbedingt funktionieren.

Wenn Sie die Funktion `faltung` ohne Parameter in Matlab aufrufen, so sollte sich – falls Sie die Funktion `falt` implementiert haben – ein Fenster wie in Abbildung 3 öffnen. Das untere Diagramm zeigt dabei die zu faltenden Funktionen (wobei die Funktion $y(t)$ schon an der Ordinate gespiegelt dargestellt wird). Mit dem Schieberegler unter dem Graphen können Sie diese Funktion gegen die Funktion $x(t)$ verschieben. Ein roter Kreis markiert dabei die Stelle t_0. Im oberen Diagramm wird das Ergebnis der Faltung der beiden Funktionen zur Kontrolle Ihres Programms gezeichnet und der rote Punkt fährt, sollten Sie `falt` richtig implementiert haben, diese Kurve entsprechend der Stellung des Schiebereglers nach.

Im Folgenden finden Sie noch vier Zusatzaufgaben, die Sie lösen können, falls Sie sich in die GUI-Programmierung in Matlab weiter einarbeiten wollen (sie werden aber nicht in der Übung vorgerechnet und es wird auch keine Musterlösung erstellt, wir können aber Ihre Lösung über die Web-Seite zur Vorlesung anderen Studenten zur Verfügung stellen, falls Sie Interesse daran haben, schicken Sie einfach eine kurze E-Mail an `d.papsdorf@nt.tu-darmstadt.de`).

3.1. Erweitern Sie das GUI so, dass das Ergebnis der Faltung nur bis zum aktuellen Zeitpunkt t_0 sichtbar ist.

3.2. Ändern Sie das GUI so, dass zwei andere Funktionen gefaltet werden.

3.3. Erweitern Sie das GUI so, dass die Zeitachsen und Funktionen in einem weiteren M-File eingegeben werden können und man so beliebige Funktionen miteinander Falten kann.

3.4. Erweitern Sie das GUI so, dass man den Namen der Datei mit den Funktionen und Zeitachsen in einem Eingabefeld in dem GUI eingeben (oder über einen Auswahldialog auswählen) kann.

4. Aufgabe

Durch das Einbinden zusätzlicher Programmoptionen, sogenannter Toolboxen, ist Matlab in der Lage spezielle mathematische Operationen aus natur- und ingenieurwissenschaftlichen Disziplinen zu berechen. So kann Matlab beispielsweise Probleme wie die Laplace-Transformation auch analytisch lösen, was allerdings voraussetzt, dass auf Ihrem Arbeitsplatz die Symbolic Toolbox installiert ist. Diese Aufgabe richtet sich also nur an Studenten, an deren Arbeitsplatz diese Option installiert ist.

Diese Aufgabe ist leider nur in Matlab und nicht in Octave lösbar, da für Octave keine Erweiterung zum symbolischen Rechnen existiert. Sie können diese Aufgabe aber in einem der zahlreichen Computer-Algebra-Systeme wie Maple, Derive, Mathematica oder als freie Alternative z. B. (WX)Maxima oder Axiom lösen, unser Lösungsvorschlag wird sich allerdings auf Matlab beschränken.

4.1. Führen Sie zunächst eine Partialbruchzerlegung des Ausdrucks

$$X(p) = \frac{9p^2 + 28p + 37}{p^3 + 8p^2 + 37p + 50}$$

aus Aufgabe 2 der 7. Übung zu Deterministische Signale und Systeme durch. Benutzen Sie den Befehl `residue`. Entspricht das Ergebnis der Lösung aus der Übungsaufgabe?

4.2. Führen Sie nun die inverse Laplace-Transformation des Signals durch. Schauen Sie sich hierzu zunächst die Hilfe zum Befehl `syms` an und benutzen Sie diesen um die Lösung analytisch zu berechen. Vergleichen Sie die inverse Transformierte der ursprünglichen mit der der partialbruchzerlegten Funktion.

4.3. Erzeugen Sie einen Vektor t, der die Werte von 0 bis 3 in Schritten von 0,1 enthält. Mit welchem Befehl kann man den »analytischen Variablen« nun Zahlenwerte zuweisen? Berechnen Sie die Funktionswerte der inversen Transformierten in dem genannten Bereich und stellen Sie sie graphisch dar – vergessen Sie die Achsenbeschriftung nicht.