

Karlsruher Institut für Technologie <b>Institut für Biomedizinische Messtechnik</b>	
Prof. Dr. rer. nat. O. Dössel Kaiserstr. 12 / Geb. 30.33 Tel.: 0721 / 608-42650	Dipl. Ing. J. Schmid Kaiserstr. 12 / Geb. 30.33 Tel.: 0721 / 608-48035

## Lineare Elektrische Netze

# Matlab-Aufgabe

<b>Vorname:</b>	Niklas
<b>Nachname:</b>	Fauth
<b>Matrikelnummer:</b>	1932872
<b>RZ-Account:</b>	utede
<b>Punkte:</b>	

### Angaben zur Bearbeitung der Aufgaben:

Die Aufgaben müssen selbstständig und ohne fremde Hilfe bearbeitet werden.

Der Lösungsweg muss vollständig angegeben und nachvollziehbar sein! Dokumentieren Sie Ihre Überlegungen, geben Sie erläuternde Kommentare!

Die maximale Punktzahl dieser Aufgabe entspricht 3% der Gesamtpunktzahl der Endnote im Fach Lineare Elektrische Netze.

### **Eidesstattliche Erklärung**

Hiermit erkläre ich an Eides statt, dass ich die vorliegende Arbeit selbstständig und ohne unzulässige fremde Hilfsmittel angefertigt habe. Wörtlich oder inhaltlich übernommene Stellen sind als solche kenntlich gemacht und die verwendeten Literaturquellen im Literaturverzeichnis vollständig angegeben. Die „Regeln zur Sicherung guter wissenschaftlicher Praxis im Karlsruher Institut für Technologie (KIT)“ in ihrer gültigen Form wurden beachtet.

Karlsruhe, den \_\_\_\_\_

Datum und Unterschrift



# Inhaltsverzeichnis

<b>1</b>	<b>Einführung</b>	<b>3</b>
<b>2</b>	<b>UDot</b>	<b>3</b>
2.1	Erklärung . . . . .	3
2.2	Quellcode . . . . .	4
2.3	help-Ausgabe . . . . .	5
<b>3</b>	<b>UInt</b>	<b>5</b>
3.1	Erklärung . . . . .	5
3.2	Quellcode . . . . .	6
3.3	help-Ausgabe . . . . .	7
<b>4</b>	<b>Anwendung</b>	<b>8</b>
4.1	Ableiten . . . . .	8
4.2	Ableiten und Integrieren . . . . .	9

## 1 Einführung

Nachfolgend die erarbeiteten Lösungen. Zum Teil wurde von einer minimalistischen Lösung abgesehen, um für schöneren Code zu Sorgen oder die Benutzerfreundlichkeit zu erhöhen.

## 2 UDot

### 2.1 Erklärung

UDot ist eine eigene Implementierung einer Funktion, um (Spannungs)werte gegenüber der Zeit abzuleiten.

## 2.2 Quellcode

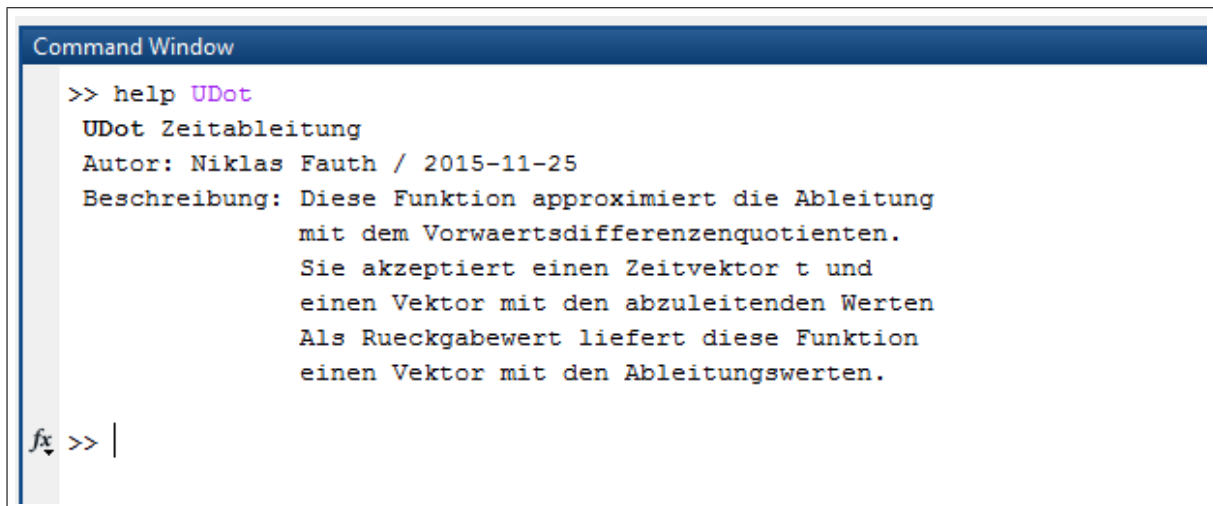
Nachfolgend der Quellcode der Funktion UDot.

```
1 function UDot = UDot(t , U)
2
3 %UDOT Zeitableitung
4 %Autor: Niklas Fauth / 2015-11-25
5 %Beschreibung: Diese Funktion approximiert die Ableitung
6 %               mit dem Vorwaertsdifferenzenquotienten.
7 %               Sie akzeptiert einen Zeitvektor t und
8 %               einen Vektor mit den abzuleitenden Werten
9 %               Als Rueckgabewert liefert diese Funktion
10 %              einen Vektor mit den Ableitungswerten.
11
12 if (length(t) ~= length(U))
13     vectorLength = min([length(t) length(U)]);
14     if (vectorLength == length(t))
15         vectorName = 'time';
16     else
17         vectorName = 'input';
18     end
19     warning('The input vectors of UDot do not have the same length. The %s
20         vector will be used.', vectorName);
21 else
22     vectorLength = length(t);
23 end
24 % Initialization of the returned vector.
25 UDot = zeros(1, vectorLength);
26
27 for i = 1 : vectorLength % Calculate the derivation.
28
29     % Check for last value in vector.
30     if (i == vectorLength)
31         UDot(i-1) = UDot(i);
32     else
33         % Difference between two time values.
34         dt = t(i+1) - t(i);
35
36         % Difference between two input values.
37         dU = U(i+1) - U(i);
38
39         % calculate the actual derivation.
40         UDot(i) = dU/dt;
41     end
42 end
43
44 end
```

Die Funktion erfüllt alle geforderten Bedingungen. Haben die zwei Ausgangsvektoren nicht dieselbe Länge, wird eine entsprechende Warnung ausgegeben. Um diese Ausnahme abzufangen wird im Fehlerfall jedoch nicht einfach abgebrochen, sondern der kürzere der beiden Vektoren zur Berechnung genutzt. Die Angabe, welcher Vektor tatsächlich verwendet wurde, ist in der Warnung enthalten.

## 2.3 help-Ausgabe

Durch Eingabe des Befehls `>help UDot<` erscheint folgende Hilfe:



```
Command Window

>> help UDot
UDot Zeitableitung
Autor: Niklas Fauth / 2015-11-25
Beschreibung: Diese Funktion approximiert die Ableitung
               mit dem Vorwaertsdifferenzenquotienten.
               Sie akzeptiert einen Zeitvektor t und
               einen Vektor mit den abzuleitenden Werten
               Als Rueckgabewert liefert diese Funktion
               einen Vektor mit den Ableitungswerten.

fx >> |
```

Abbildung 1: help-Ausgabe

## 3 UInt

### 3.1 Erklärung

UInt ist eine eigene Implementierung einer Funktion, um (Spannungs)werte gegenüber der Zeit zu integrieren.

## 3.2 Quellcode

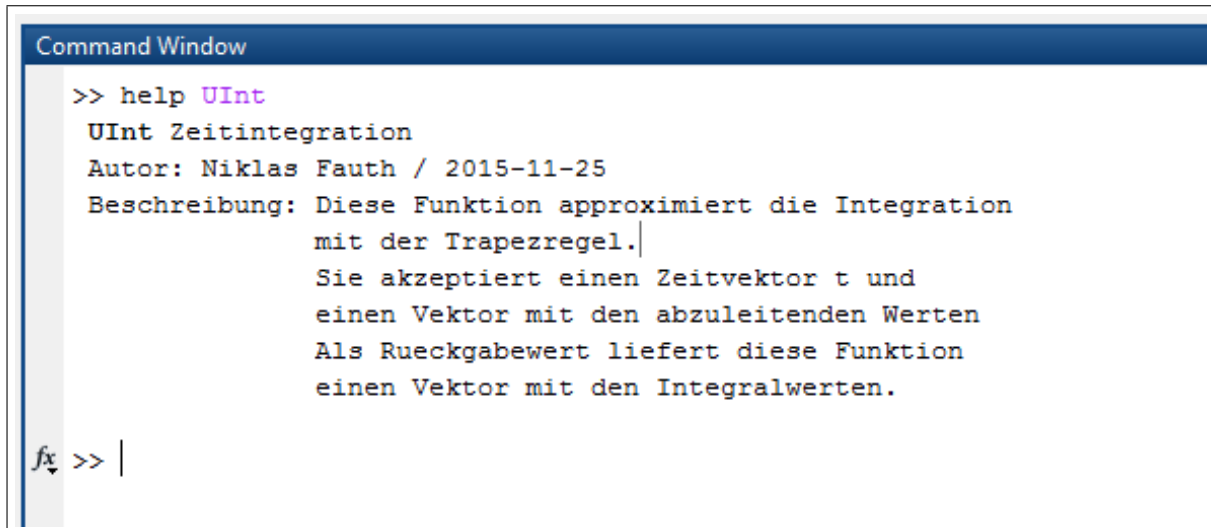
Nachfolgend der Quellcode der Funktion UInt.

```
1 function UInt = UInt(t, U)
2 %UINT Zeitintegration
3 %Autor: Niklas Fauth / 2015-11-25
4 %Beschreibung: Diese Funktion approximiert die Integration
5 %               mit der Trapezregel.
6 %               Sie akzeptiert einen Zeitvektor t und
7 %               einen Vektor mit den abzuleitenden Werten
8 %               Als Rueckgabewert liefert diese Funktion
9 %               einen Vektor mit den Integralwerten.
10
11 if (length(t) ~= length(U))
12     vectorLength = min([length(t) length(U)]);
13     if (vectorLength == length(t))
14         vectorName = 'time';
15     else
16         vectorName = 'input';
17     end
18     warning('The input vectors of UInt do not have the same length. The %s
19         vector will be used.', vectorName);
20 else
21     vectorLength = length(t);
22 end
23 % Initialization of the returned vector.
24 UInt = zeros(1, vectorLength);
25
26 for i = 1 : vectorLength % Calculate the integration.
27
28     % Check for last value in vector.
29     if(i == vectorLength)
30         break;
31
32     elseif(i == 1)
33         Usum = 0;
34
35     else
36         % Difference between two time values.
37         dt = t(i + 1) - t(i);
38
39         % Sum of two input values.
40         Usum = ((U(i) + U(i + 1)) / 2 * dt) + Usum;
41         UInt(i) = Usum;
42     end
43 end
44
45 end
```

Die Funktion erfüllt alle geforderten Bedingungen. Haben die zwei Ausgangsvektoren nicht dieselbe Länge, wird eine entsprechende Warnung ausgegeben. Um diese Ausnahme abzufangen wird im Fehlerfall jedoch nicht einfach abgebrochen, sondern der kürzere der beiden Vektoren zur Berechnung genutzt. Die Angabe, welcher Vektor tatsächlich verwendet wurde, ist in der Warnung enthalten.

### 3.3 help-Ausgabe

Durch Eingabe des Befehls `>help UInt<` erscheint folgende Hilfe:



```
Command Window

>> help UInt
UInt Zeitintegration
Autor: Niklas Fauth / 2015-11-25
Beschreibung: Diese Funktion approximiert die Integration
               mit der Trapezregel.
               Sie akzeptiert einen Zeitvektor t und
               einen Vektor mit den abzuleitenden Werten
               Als Rueckgabewert liefert diese Funktion
               einen Vektor mit den Integralwerten.

fx >> |
```

Abbildung 2: help-Ausgabe

## 4 Anwendung

### 4.1 Ableiten

bla bla bla

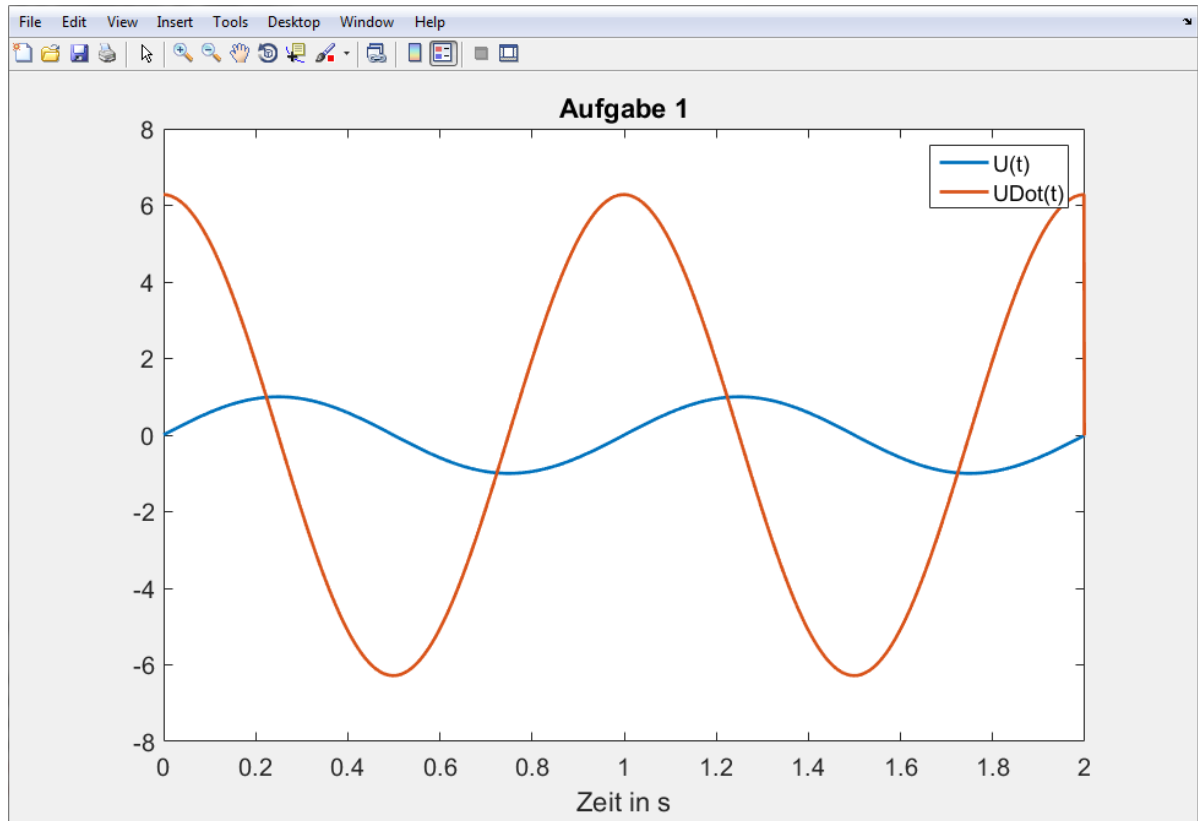


Abbildung 3: Plot der Ausgangsfunktion und ihrer Ableitungsfunktion

Dazu wurden folgende Befehle verwendet:

```
1 samples = 2001;
2 t = linspace(0, 2, samples);
3 U = linspace(0, samples, samples);
4
5 for i = 1 : samples
6     U(i) = sin((i / samples) * 4 * pi);
7 end
8
9 plot(t, U, t, UDot(t, U), 'LineWidth', 2);
10 set(gca, 'FontSize', 15);
11 xlabel('Zeit in s');
12 title('Aufgabe 1');
13 legend('U(t)', 'UDot(t)');
```



## 4.2 Ableiten und Integrieren

bla bla bla

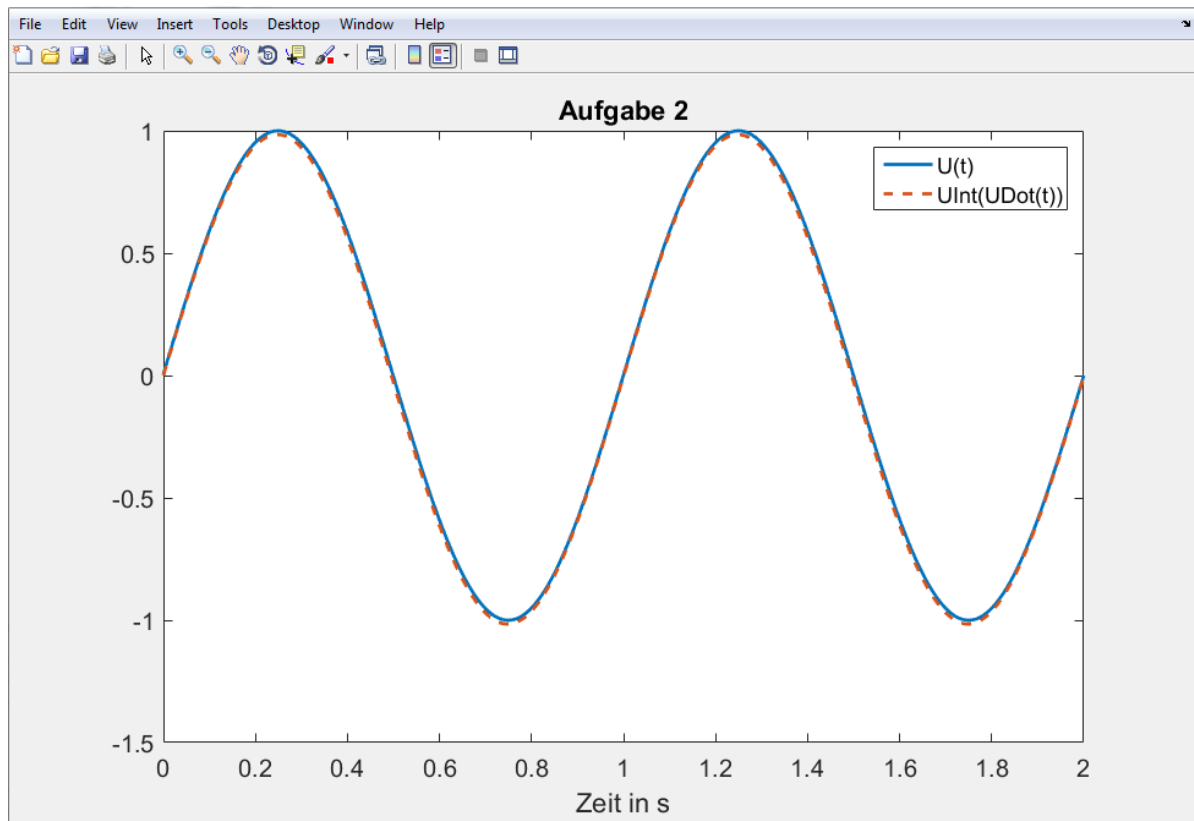


Abbildung 4: Plot

Dazu wurden folgende Befehle verwendet:

```
1 samples = 2001;
2 t = linspace(0, 2, samples);
3 U = linspace(0, samples, samples);
4
5 for i = 1 : samples
6     U(i) = sin((i / samples) * 4 * pi);
7 end
8
9 plot(t, U, t, UInt(t, UDot(t, U)), '—', 'LineWidth', 2);
10 set(gca, 'FontSize', 15);
11 xlabel('Zeit in s');
12 title('Aufgabe 2');
13 legend('U(t)', 'UInt(UDot(t))');
```