

# Aktorik Sensorik

## Labor 3

Anton Kress (S872899), Jan Abel (S876662)

Dezember 2020

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung und Ziel</b>	<b>3</b>
<b>2</b>	<b>Grundlagen und Theorie</b>	<b>4</b>
<b>3</b>	<b>Aufgabenstellung und Versuch</b>	<b>5</b>
<b>4</b>	<b>Zusammenfassung</b>	<b>9</b>
<b>5</b>	<b>Anhang</b>	<b>10</b>
5.1	Aufgabenbeschreibung . . . . .	10
5.2	Matlab Code . . . . .	10

# 1 Einleitung und Ziel

Im 3. Labor des Moduls Aktorik und Sensorik soll ein mathematische Modell eines Gleichstrommotors aufgestellt und in Simulink umgesetzt werden.

Die dafür nötigen Konstanten (u.a Ankerwiderstand, Induktivität, etc.) sind in den beiden vergangenen Versuchen bereits bestimmt worden. Um das Trägheits-moment  $J$  – die einzige fehlende Konstante im Modell – zu bestimmen, wird das Modell mit einer Messung am realen System verglichen. In dieser Messung wurde ein Spannungs-Sprung auf den Motor gegeben und der Strom als Sprungantwort aufgenommen.

## 2 Grundlagen und Theorie

Das Blocksschaltbild, welches in Simulink umzusetzen ist, resultiert aus den beiden folgenden DGL.

$$\begin{aligned} \dot{i}(t) &= \frac{1}{L} [u(t) - (R + R_s) \cdot i(t) - k_e \cdot \omega(t)] \\ \dot{\omega}(t) &= \frac{1}{J} [k_m \cdot i(t) - C_r \omega(t)] \end{aligned} \tag{1}$$

Die erste Gleichung ergibt sich aus der Maschengleichung des elektrischen Teils. Der zweite Teil der DGL ist auf die Summation der Drehmomente zurückzuführen.

### 3 Aufgabenstellung und Versuch

Aus den bereits aufgestellten DGL-System kann das Blockschaltbild in Simulink erstellt werden.

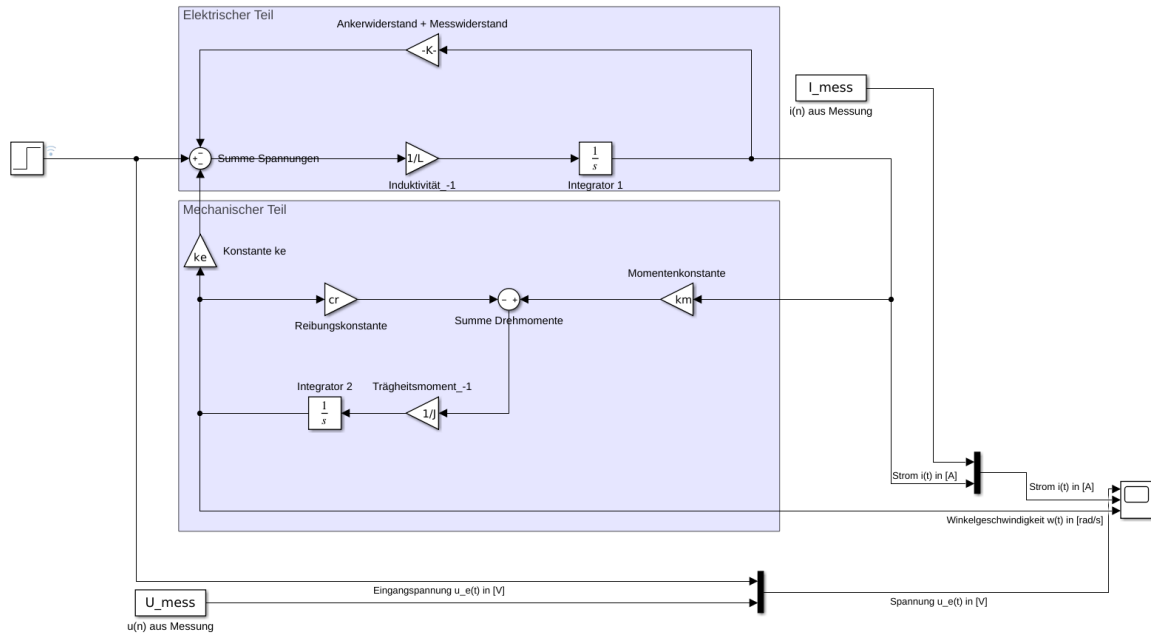


Abbildung 1: Blockschaltbild in Simulink

Als Eingang wurde ein Step-Block benutzt und als Ausgang Scope-Block. Die erste DGL wurden im oberen Teil des Modell realisiert und beschreibt den elektrischen Teil des Systems. Im unteren wird der mechanische Teil des Motors modelliert der aus der zweiten DGL hervorgeht.

Die Konstanten wurden in einem Matlab-File abgespeichert und werden über dieses auch aufgerufen. Auch die Daten der Messung des Spungs und der Sprungantwort sind hier zu finden.

Um den gemessenen Spung und die Sprungantwort in Simulink anzuzeigen wurde der Block "From Workspace" benutzt der jeweils den Zeit-Spannungs-Vektor und den Zeit-Spannungs-Vektor lädt.

Der Spung auf das Modell wurden dem Sprung aus der Messung angepasst.

iiiiiii HEAD

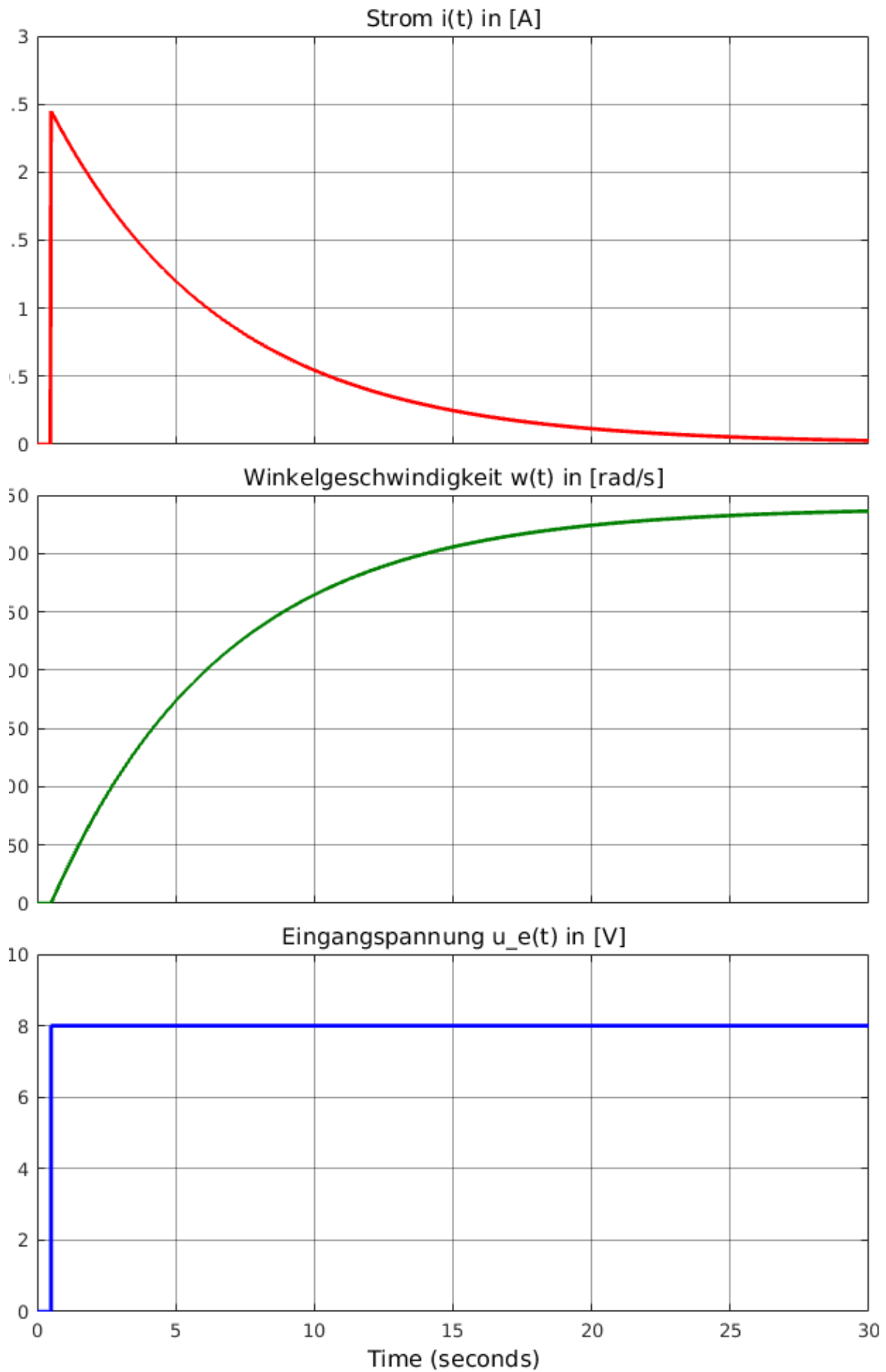
Steptime	0.048-s
Final Value	8.2V
Sampletime	0.001s

Abbildung 2: Festlegung der Simulationsdaten

===== STEPTIME 0.048s FINAL VALUE 8.2V SAMPLETIME 0.001s

test

Um das Trägheitsmoment  $J$  zu bestimmen, haben wir es auf einen Initialwert von  $10^{-3}\text{Kgm}^2$  gesetzt. Für diesen Wert haben wir uns an dem Datenblatt aus der Vorlesung orientiert. Die Simulationszeit setzten wir auf 30 Sekunden und bekamen folgenden Output im Scope.



~~~~~ 478b1e179bc43ccca16d0157792f73889abcf5ea



## 4 Zusammenfassung

$$\begin{aligned} L &= 175.462 \mu\text{H} \\ c_r &= 3,24 \cdot 10^{-9} \frac{\text{Nms}}{\text{rad}} \end{aligned} \tag{2}$$

## 5 Anhang

### 5.1 Aufgabenbeschreibung

#### 3 Termin

##### 3.1 Simulation des Motors

Erzeugen Sie unter MATLAB / Simulink eine Simulation des permanent erregten Gleichstrommotors. Benutzen Sie dafür Signalquellen und -senken, Übertragungsfunktionen und Summationspunkte. Stellen Sie die Winkelgeschwindigkeit und den Ankerstrom als Funktion der Zeit in jeweils einem Scope (Signalsenke) dar. Verwenden Sie dabei die von Ihrem Team gemessen bzw. geschätzten Parameter!

Stellen Sie sicher, dass Sie mindestens zwei Sekunden simulieren!

##### 3.2 Messen des Einschaltverhaltens & Vergleich mit Simulation

Messen Sie mit dem PicoScope das Einschaltverhalten des Stromes des Motors und speichern sie die Messwerte ab. Erklären Sie dabei, warum der Stromverlauf so ist, wie Sie ihn gemessen haben. Nun lesen Sie die Messwerte in MATLAB ein und stellen Sie den gemessenen Stromverlauf mit der Anzeige des simulierten Motorstromes in einem Bild dar.

Beantworten Sie folgende Fragen: Sind die Verläufe identisch? Warum nicht? Woran liegt es?

##### 3.3 Diskussion

Diskutieren Sie, ob Ihre vorwiegend aus statischen Messungen stammenden Werte eine vernünftige Simulation des Motorverhaltens ermöglichen.

Passen Sie die Parameter der von Ihnen erzeugten MATLAB-Simulation so an, dass der Unterschied zwischen der Messung und der Simulation eine maximale Abweichung von 5 % beträgt!

|                                        |                                      |                                               |
|----------------------------------------|--------------------------------------|-----------------------------------------------|
| Beuth-Hochschule für<br>Technik Berlin | Labor für<br>Automatisierungstechnik | Übungsveranstaltung für<br>Aktorik & Sensorik |
| Prof. Dr.-Ing. FJ Morales              |                                      | 7 von 10                                      |

### 5.2 Matlab Code

```
1 % Aktorik & Sensorik - WS 2020
2 %
3 % 2.1-2.3 Berechnung der Induktivität mittels des Phasenwinkels
4 %
5 % Datum: 12.11.2020
6 % Autoren: Anton Kress, S872899
```

```

7 % Jan Abel, S876662
8
9 clear
10 home
11 close all
12
13 FolderName = "./src/";
14 E_Name = "E.mat";
15 E = fullfile(FolderName, E_Name);
16 load(E);
17
18 R = 3.263586106324851; % Ankerwiderstand in [V/A]
19 Rs = 1; % Messwiderstand in [V/A]
20 delta_t = E(:,2); % Phasenverschiebung in [s]
21 f = E(:,1); % Frequenzen in [1/s]
22 freq=linspace(500,1500); % Frequenz in [1/s]
23
24 % Linearisierung
25 y = tan(2*pi*f.*delta_t); % Y-Achse Faktor - einheitenlos
26
27 % Fitting
28 f1=polyfit(f, y, 1);
29 m=f1(1,1);
30
31 y1=polyval(f1,freq);
32
33 L = (m*(R+Rs))/(2*pi)
34
35 y2=atan((2*pi*L*f)/(R+Rs))./(2*pi*f);
36
37 figure(1)
38 subplot(1,2,1)
39 plot(f,y, 'o', freq,y1,'r','linewidth',2);
40 grid on;
41 title('Induktivität 1')
42 subtitle(['L=' num2str(L)])
43 xlabel('Frequenz f in Hz')
44 ylabel('Faktor tan(2 pi f d_t)')
45 subplot(1,2,2)
46 plot(E(:,1), E(:,2), 'o', f, y2,'r','linewidth',2);
47 grid;
48 title('Induktivität 2')
49 xlabel('Frequenz f in Hz')
50 ylabel('Zeitverschiebung t in s')
51
52 % save current plot to img/-folder
53 imagePath = fullfile('../img/', mfilename);
54 print(imagePath, '-dpng');

```

```

1 % Aktorik & Sensorik - WS 2020
2 %
3 % 2.4 Berechnung der Reibungskonstanten
4 %
5 % Datum: 12.11.2020
6 % Autoren: Anton Kress, S872899
7 % Jan Abel, S876662
8

```

```

9 clear
10 home
11 close all
12
13 FolderName = "./src/";
14 Leerlauf_Name = "Leerlauf.mat";
15 Leerlauf = fullfile(FolderName, Leerlauf_Name);
16 load(Leerlauf);
17
18 Pz = 2000/(2*pi);           % Pulse Inkrementalgeber      in [inc/rad]
19 lambda = 1000/Pz;          % Umrechnungsfaktor          in [(ms rad)/(s inc)]
20 ]
21 I = Leerlauf(:,2);          % Strom I_a          in [A]
22 INC = Leerlauf(:,3);        % INC per T          in [INC/ms]
23 w = lambda*INC;             % Winkelgeschwindigkeit in [rad/s]
24 km = 0.022031575949394;    % Momentenkonstante  in [Nm/A]
25
26 % lineares Fitting im Arbeitsbereich
27 f2 = polyfit(I(2:end), w(2:end), 1);
28 % Strom Vektor
29 x2 = linspace(0, 0.05);
30 % Winkelges. Vektor
31 y2 = polyval(f2, x2);
32 % Steigung m hat die Einheit [rad/(A s)]
33 m = f2(1);
34 % Reibungskonstante cr in [(Nm s)/rad]
35 cr = km*1/m
36
37 figure(1);
38 plot(I,w,'x', x2, y2, 'r', 'linewidth', 2);
39 title('Reibungskonstante');
40 subtitle(['c_r=' num2str(cr)])
41 grid;
42 axis([0.038 0.048 0 550]);
43 xlabel('Strom I in A');
44 ylabel('Winkelgeschwindigkeit w in rad/s');
45
46 % save current plot to img/-folder
47 imagePath = fullfile('../img/', mfilename);
48 print(imagePath, '-dpng');

```