

Aktorik Sensorik

Labor 3

Anton Kress (S872899), Jan Abel (S876662)

Dezember 2020

Inhaltsverzeichnis

1	Einleitung und Ziel	3
2	Grundlagen und Theorie	4
3	Aufgabenstellung und Versuch	5
4	Zusammenfassung	6
5	Anhang	7
5.1	Aufgabenbeschreibung	7
5.2	Matlab Code	7

1 Einleitung und Ziel

Im 3. Labor des Moduls Aktorik und Sensorik soll ein mathematisches 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 Trägheitsmoment 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.

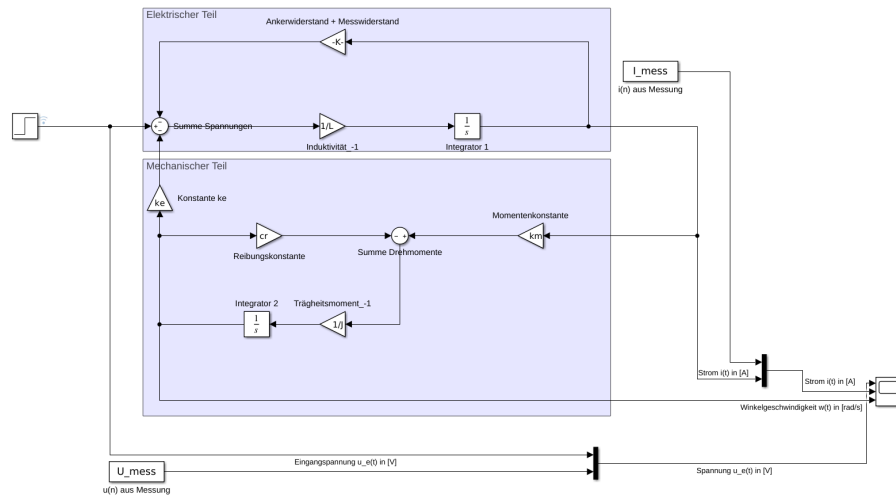


Abbildung 1: Blockschaltbild in Simulink

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.

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 gemessenen 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 Technik Berlin	Labor für Automatisierungstechnik	Übungsveranstaltung für 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');
55
56 % Aktorik & Sensorik - WS 2020
57 %
58 % 2.4 Berechnung der Reibungskonstanten
59 %
60 % Datum:      12.11.2020
61 % Autoren:    Anton Kress,      S872899
62 %             Jan Abel,         S876662
63
64 clear
65 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 I = Leerlauf(:,2);           % Strom I_a         in [A]
21 INC = Leerlauf(:,3);         % INC per T         in [INC/ms]
22 w = lambda*INC;              % Winkelgeschwindigkeit in [rad/s]
23 km = 0.022031575949394;     % Momentenkonstante in [Nm/A]
24
25 % lineares Fitting im Arbeitsbereich
26 f2 = polyfit(I(2:end), w(2:end), 1);
27 % Strom Vektor
28 x2 = linspace(0, 0.05);
29 % Winkelges. Vektor
30 y2 = polyval(f2, x2);
31 % Steigung m hat die Einheit [rad/(A s)]
32 m = f2(1);
33 % Reibungskonstante cr in [(Nm s)/rad]
34 cr = km*1/m
35
36 figure(1);
37 plot(I,w,'x', x2, y2, 'r', 'linewidth', 2);
38 title('Reibungskonstante');
39 subtitle(['c_r=' num2str(cr)])
40 grid;
41 axis([0.038 0.048 0 550]);
42 xlabel('Strom I in A');
43 ylabel('Winkelgeschwindigkeit w in rad/s');
44
45 % save current plot to img/-folder
46 imagePath = fullfile('../img/', mfilename);
47 print(imagePath, '-dpng');

```