Aktorik Sensorik
Labor 3

Anton Kress (S872899), Jan Abel (S876662)

Dezember 2020

Inhaltsverzeichnis

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 umzusetzten ist, resultiert aus den beiden folgenden DGL.

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

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.

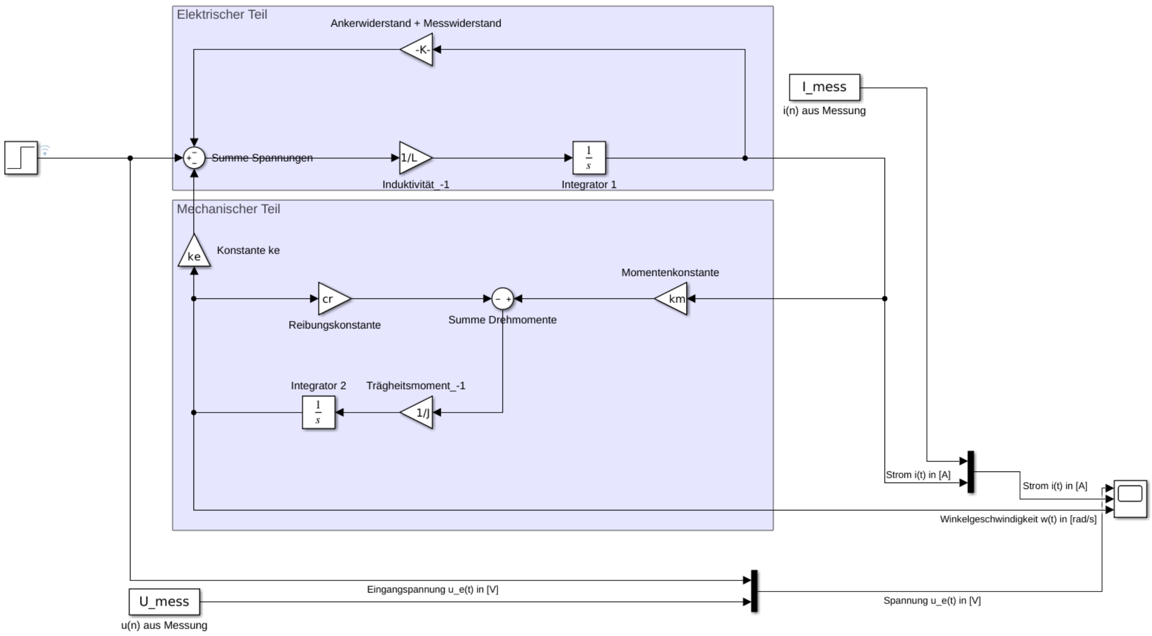

Abbildung 1: Blockschaltbild in Simulink

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 zufinden.

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.

Steptime	0.048-s
Final Value	8.2V
Sampletime	0.001s

Abbildung 2: Festlegung der Simulationsdaten

4 Zusammenfassung

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

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!

5.2 Matlab Code

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

```
%            Jan Abel,        S876662

clear
home
close all

FolderName = "./src/";
E_Name = "E.mat";
E = fullfile(FolderName, E_Name);
load(E);

R = 3.263586106324851;                  % Ankerwiderstand     in [V/A]
Rs = 1;                                 % Messwiderstand      in [V/A]
delta_t = E(:,2);                       % Phasenverschiebung  in [s]
f = E(:,1);                             % Frequenzen          in [1/s]
freq=linspace(500,1500);                % Frequenz            in [1/s]

% Linearisierung
y = tan(2*pi*f.*delta_t);               % Y-Achse Faktor - einheitenlos

% Fitting
f1=polyfit(f, y, 1);
m=f1(1,1);

y1=polyval(f1,freq);

L = (m*(R+Rs))/(2*pi)

y2=atan((2*pi*L*f)/(R+Rs))./(2*pi*f);

figure(1)
subplot(1,2,1)
    plot(f,y, 'o', freq,y1,'r','linewidth',2);
    grid on;
    title('Induktivität 1')
    subtitle(['L=' num2str(L)])
    xlabel('Frequenz f in Hz')
    ylabel('Faktor tan(2 pi f d_t)')
subplot(1,2,2)
    plot(E(:,1), E(:,2), 'o', f, y2,'r','linewidth',2);
    grid;
    title('Induktivität 2')
    xlabel('Frequenz f in Hz')
    ylabel('Zeitverschiebung t in s')

% save current plot to img/-folder
imagePath = fullfile('../img/', mfilename);
print(imagePath,'-dpng');
```

```
% Aktorik & Sensorik - WS 2020
%
% 2.4 Berechnung der Reibungskonstanten
%
% Datum:     12.11.2020
% Autoren:   Anton Kress,    S872899
%            Jan Abel,       S876662

```

```matlab
clear
home
close all

FolderName = "./src/";
Leerlauf_Name = "Leerlauf.mat";
Leerlauf = fullfile(FolderName, Leerlauf_Name);
load(Leerlauf);

Pz = 2000/(2*pi);        % Pulse Inkrementalgeber   in [inc/rad]
lambda = 1000/Pz;        % Umrechnungsfaktor        in [(ms rad)/(s inc)]
I = Leerlauf(:,2);       % Strom I_a                in [A]
INC = Leerlauf(:,3);     % INC per T                in [INC/ms]
w = lambda*INC;          % Winkelgeschwindigkeit    in [rad/s]
km = 0.022031575949394; % Momentenkonstante        in [Nm/A]

% lineares Fitting im Arbeitsbereich
f2 = polyfit(I(2:end), w(2:end), 1);
% Strom Vektor
x2 = linspace(0, 0.05);
% Winkelges. Vektor
y2 = polyval(f2, x2);
% Steigung m hat die Einheit [rad/(A s)]
m = f2(1);
% Reibungskonstante cr in [(Nm s)/rad]
cr = km*1/m

figure(1);
    plot(I,w,'x', x2, y2, 'r', 'linewidth', 2);
    title('Reibungskonstante');
    subtitle(['c_r=' num2str(cr)])
    grid;
    axis([0.038 0.048 0 550]);
    xlabel('Strom I in A');
    ylabel('Winkelgeschwindigkeit w in rad/s');

% save current plot to img/-folder
imagePath = fullfile('../img/', mfilename);
print(imagePath,'-dpng');
```