Lecture Module - Electrical Systems

ME3050 - Dynamic Modeling and Controls

Mechanical Engineering

Tennessee Technological University

Topic 4 - Example: Brushed DC Motor

Electrical Systems

- Brushed DC Motor
- Model Derivation
- State Space Form
- Transfer Functions
- Simulated Response

Model Derivation

Armature Controlled Brushed DC Motor

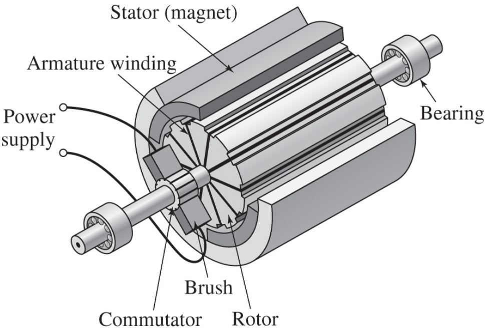
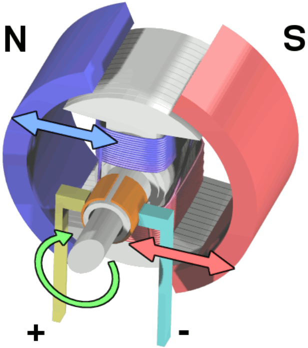


Image: System Dynamics, Palm, 4^{th}, Pg. 376-378

Brushed DC Motor

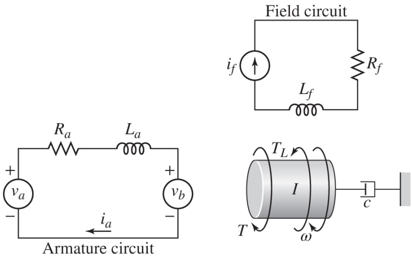


Animation on Web

source: wikipedia

Model Derivation

Armature Controlled Brushed DC Motor



v_a : armature voltage (input)

R_a : armature resistance

Torque on armature

$$T = (nBLi_a)\,r = (nBLr)\,i_a = K_T\,i_a$$

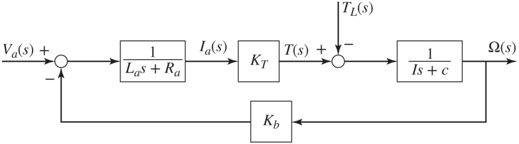
Back EMF (electromotive force) voltage

$$v_b = nBLv = (nBLr)\,\omega = K_b\omega$$

Image: System Dynamics, Palm, 4^{th}, Pg. 376-378

Model Derivation

Armature Controlled Brushed DC Motor



Kirchoff's Voltage Law

$$v_a - R_a i_a - L_a \frac{di_a}{dt} - K_b \omega = 0$$

Newtons's Second Law

$$I \frac{d\omega}{dt} = T - c\omega - T_L = K_T i_a - c\omega - T_L$$

Image: System Dynamics, Palm, 4^{th}, Pg. 376-378

State Space Form

State-Variable (State-Space) form

$$\frac{di_a}{dt} = \dot{x}_1 = \frac{1}{L_a}(v_a - R_a i_a - K_b\omega) = \begin{bmatrix} -\frac{R_a}{L_a} & -\frac{K_b}{L_a} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} \frac{1}{L_a} & 0 \end{bmatrix} \begin{bmatrix} v_a \\ T_L \end{bmatrix}$$

$$\frac{d\omega}{dt} = \dot{x}_2 = \frac{1}{I}(K_T i_a - c\omega - T_L) = \begin{bmatrix} -\frac{K_T}{I} & -\frac{c}{I} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 & -\frac{1}{I} \end{bmatrix} \begin{bmatrix} v_a \\ T_L \end{bmatrix}$$

Write the state equation in matrix form with states $x_1 = i_a$, and $x_2 = \omega$

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} -\frac{R_a}{L_a} & -\frac{K_b}{L_a} \\ \frac{K_T}{I} & -\frac{c}{I} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} \frac{1}{L_a} & 0 \\ 0 & -\frac{1}{I} \end{bmatrix} \begin{bmatrix} v_a \\ T_L \end{bmatrix}$$

Transfer Functions

The input-output relationships can be represented by the following transfer functions.

Armature Current to Armature Voltage

$$\frac{I_a(s)}{V_a(s)} = \frac{Is + c}{L_a I s^2 + (R_a I + c L_a) s + c R_a + K_b K_T}$$

Armature Current to External Load

$$\frac{I_a(s)}{T_L(s)} = \frac{K_b}{L_a I s^2 + (R_a I + c L_a) s + c R_a + K_b K_T}$$

Transfer Functions

Armature Angular Velocity to Armature Voltage

$$\frac{\Omega(s)}{V_a(s)} = \frac{K_T}{L_a I s^2 + (R_a I + c L_a) s + c R_a + K_b K_T}$$

Armature Angular Velocity to External Load

$$\frac{\Omega(s)}{T_L(s)} = \frac{L_a s + R_a}{L_a I s^2 + (R_a I + c L_a) s + c R_a + K_b K_T}$$

Transfer Functions

Final Value Theorem: To find the value of a function $x(t)$ as $t \to \infty$

$$x(\infty) = \lim_{x \to \infty} f(x) = \lim_{s \to 0} sX(s)$$

Use final value theorem to find steady state value to step input on V_a, T_L

$$i_a = \frac{cV_a + K_b T_L}{cR_a + K_b K_T}$$

$$\omega = \frac{K_T V_a - R_a T_L}{cR_a + K_b K_T}$$

Simulated Response

The following MATLAB code defines a state space system object and simulates the system response to various inputs.

```
% DC Motor Example (System Dynamics 4th ed., Palm, Pg
    376-378)
clear; close all; clc

% define system parameters
KT=0.05; % (N*m/A)
Kb=KT;   % (N*m/A)
c=10e-4; % (N*m*s/rad)
Ra=0.5;  % (Ohm)
La=2e-3; % (H)
I=9e-5;  % (kg*m^2)
```

Simulated Response

```
% define components of the state equation
A=[-Ra/La -Kb/La
   KT/I -c/I];
% B matrix is 2x2 because u vector is 2x1
B=[1/La 0
   0 -1/I];

% use first two states as outputs
C=[1 0
   0 1];
% the D matrix shape of B matrix
D=[0 0
   0 0];
```

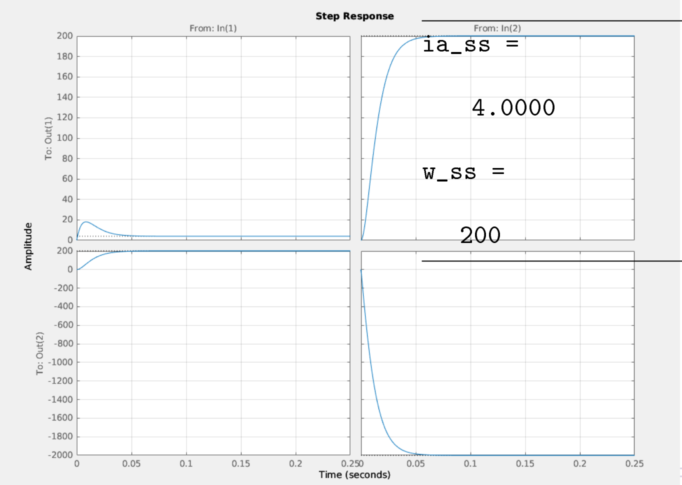
Simulated Response

```
% calculate the steady state step response
Va=12;
TL=0;
ia_ss=(c*Va+Kb*TL)/(c*Ra+Kb*KT)

% create a state space model object
sys1=ss(A,B,C,D);

% simulate a step response
figure(1)
time=0:0.001:1;
opts=stepDataOptions('StepAmplitude',Va);
step(sys1,time,opts); grid on
```

Simulated Response




```
ia_ss =

    4.0000

w_ss =

   200
```