ME322 Lab 4 Report - Permanent Magnet DC Motor Model

Author: Sterling Hull - Cal Poly Mechanical Engineering

Date Created: 11/1/2023

Date Modified: 11/14/2023

Background:

This report contains information on how to model a permanent magnet DC motor. This includes comparing a motor curve to the motor model, modeling the motor at no-load conditions, loaded conditions, and stall conditions. The model included finding the angular velocity, current, mechanical power, electrical power, efficiency, and temperature of the windings and housing for the three conditions.

Required Files:

LAB4_LOADING_MODEL_322.slx - This file contains the Simulink block diagram that was used to model the motor at loaded conditions.

LAB4_NOLOAD_MODEL_322.slx - This file contains the Simulink block diagram that was used to model the motor at no load conditions.

LAB4_STALL_MODEL_322.slx - This file contains the Simulink block diagram that was used to model the motor at stall conditions.

References:

- The prelab4a code from Professor Charlie Revfem was used for the first two plots.
- ME 322 Lab manual was referenced for formatting and all lab information.

Table of Contents

Problem Statement

This lab involved modeling a permanent magnet DC motor using block diagrams and state space formulation to solve for the electrical, rotational, and thermal system response to loaded, no load, and stall conditions. The electrical and rotational diagrams are shown in **Figure 1**.

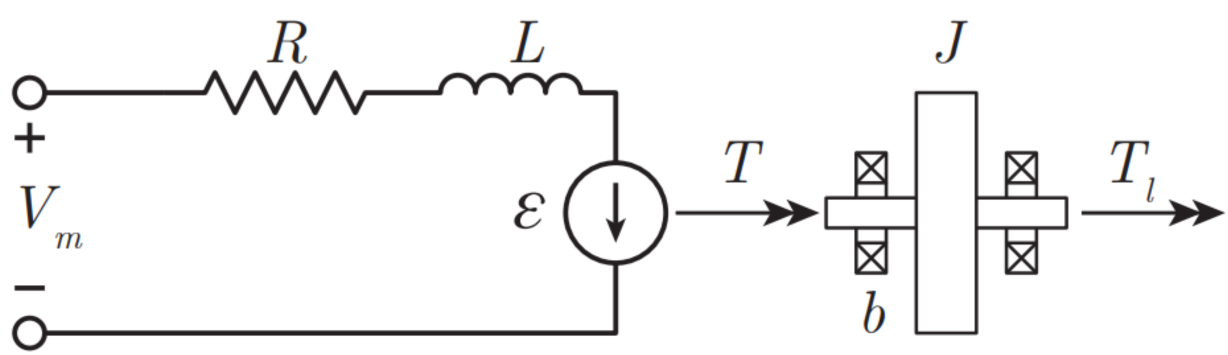

Figure 1. Electrical and rotational diagrams of permanent magnet DC motor (From ME 322 Lab Manual).

The image shown in **Figure 2.** shows the diagram for the thermal system that acts as a simple conduction model.

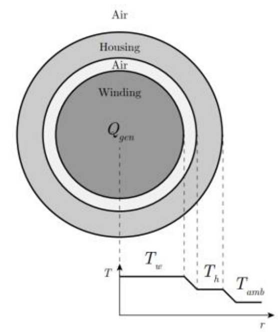

Figure 2. Thermal model of system (From ME 322 Lab Manual).

The solving approach involved using a linear graph and normal tree to develop elemental and constraint equations for the elements of the system which were then used to make state equations to create matrices to get system response values.

Hand Calculations

This section contains all the hand calculations that were performed to model the motor. The hand calculations shown in **Figure 3.** involve developing a normal tree and linear graph for the electrical and rotational system and state variables.

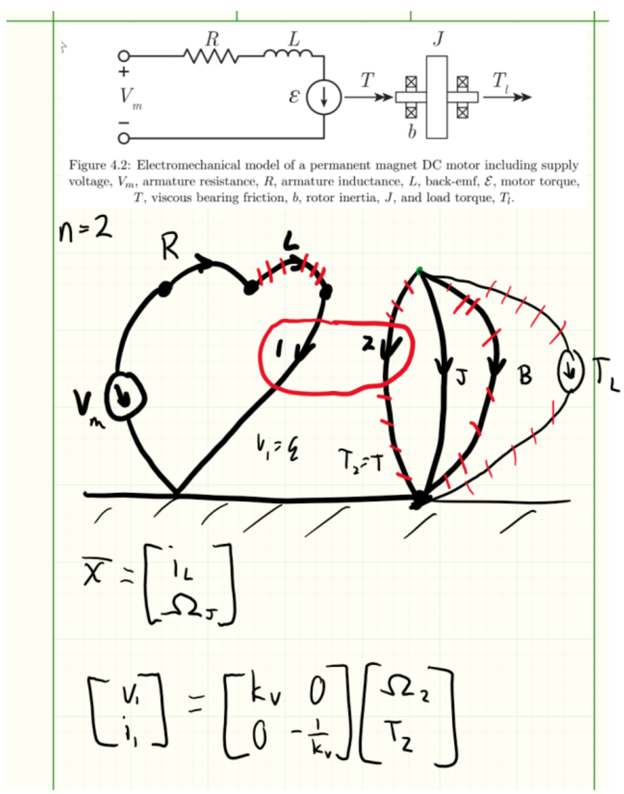

Figure 3. Hand calculations for developing linear graph and normal tree.

The hand calculations shown in **Figure 4.** show the elemental and constraint equations from the normal tree as well as the state equations when there is some load Tl.

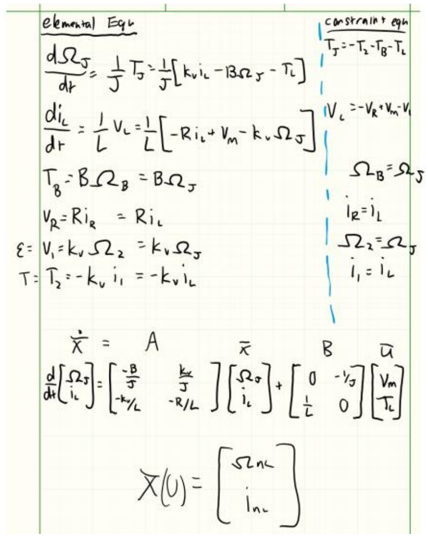

Figure 4. Elemental, constraint, and state equations for motor with load Tl.

The hand calculations shown in **Figure 5.** are the new linear graph and normal tree and the elemental, constraint, and state equations when the motor is stalled, meaning the angular velocity is zero.

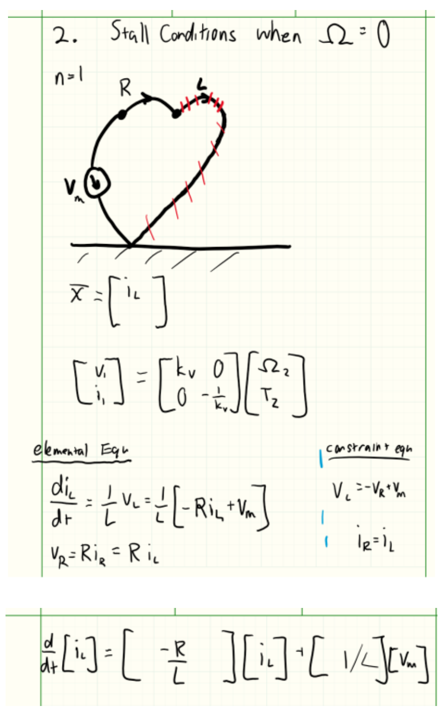

Figure 5. Linear graph, normal tree, and equations when motor is stalled.

The hand calculations shown in **Figure 6.** show the linear graph, normal tree, elemental equations, and constraint equations for the thermal model of the system.

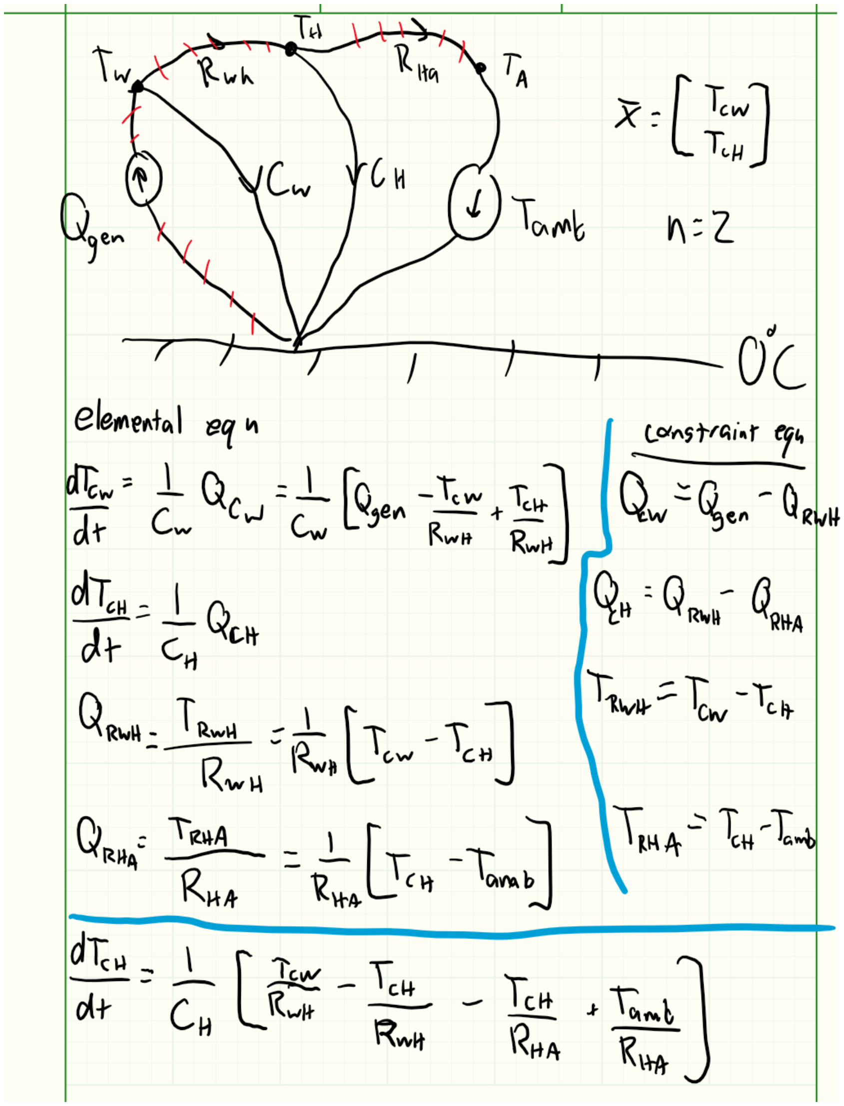

Figure 6. Calculations for thermal model of the system.

The state equations for the thermal model are shown in **Figure 7.** as well as the equation for the generated heat source from motor values.

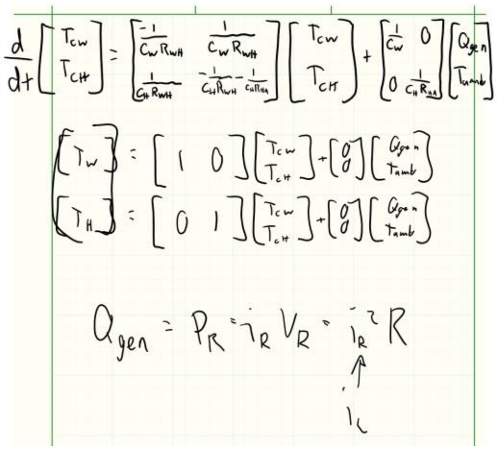

Figure 7. Hand calculations the thermal model state equations.

Figure 8. shows the equations for the outputs of interest for the three motor cases so the values can be plotted and compared.

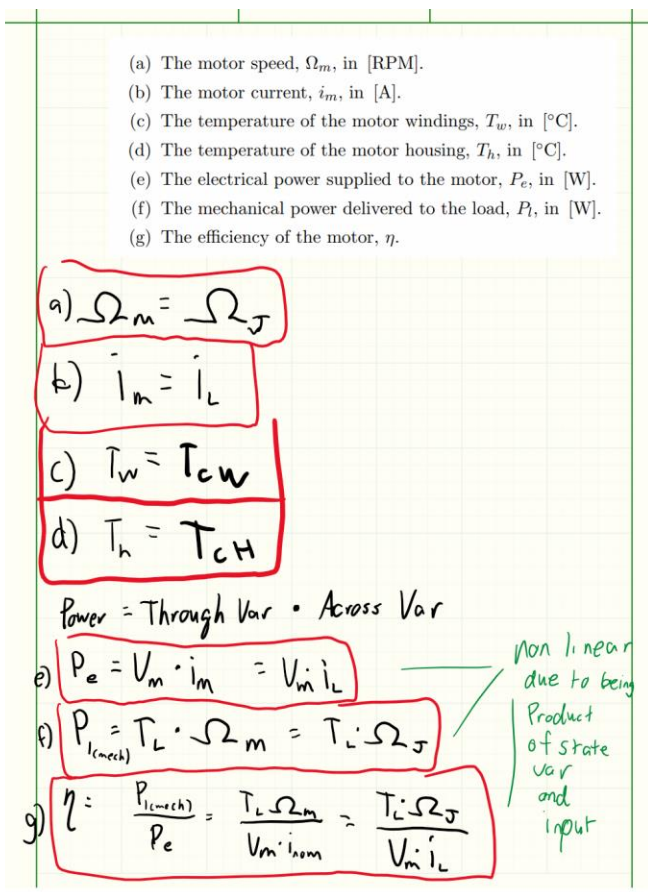

Figure 8. Output equations for the motor model.

Analysis

This section contains variables that were used in the Simulink block diagram. The matrices that were developed through hand calculations and the state space system object was created.

```
clear all;
% No-load speed
omega_noload     = 11800*2*pi/60;    % [rad/s]
% DC bus voltage
V_dc             = 18;               % [V]
% Winding resistance
R                = 0.68;             % [Ω]
% Winding inductance
L                = 0.078e-3;         % [H]
% Rotor inertia
J                = 9.82e-7;          % [kg*m^2]
% Visous damping coefficient
b                = 3.14e-7;          % [N*m*s/rad]
% Thermal resistance (housing to ambient)
R_ha             = 13.6;             % [W/K]
% Thermal resistance (winding to housing)
R_wh             = 4.57;             % [W/K]
% Thermal time-constant (winding)
tau_w            = 22;               % [s]
% Thermal time-constant (housing)
tau_h            = 646;              % [s]
% Max admissible winding temperature
T_all            = 125;              % [°C]
% Ambient temperature
T_amb            = 25;               % [°C]

Km = omega_noload/V_dc;
Kv = max(roots([Km -1 Km*b*R]));
Kt = Kv;
i_stall = V_dc/R;
T_stall = Kt * i_stall;
dNdM = omega_noload/T_stall;
i_noload = (V_dc-Kv*omega_noload)/R;
T_max = T_stall/2;
omega_max = omega_noload/2;
P_max = T_max*omega_max;
eta_max = P_max/(V_dc*T_max/Kt);
omega_eff=(-sqrt(b*R*Kt/Km+b^2*R^2)+Kt/Km+b*R)/(Kt/Km^2+b*R/Km)*V_dc;
T_eff = T_stall-omega_eff/dNdM;
P_eff = omega_eff*T_eff;
eta_eff = P_eff/(V_dc*(i_noload+T_eff/Kt));
```

```
inom = sqrt((T_all-T_amb)/(R*(R_wh+R_ha)));
T_nom = inom*Kt;
P_nom=T_nom*(omega_noload-T_nom*dNdM);
C_w = tau_w/R_wh;
C_h = tau_h/R_ha;

x0 = [0
      0]; %Initial Conditions
A = [-b/J   Kv/J
    -Kv/L   -R/L]; % A matrix for the state space input

B = [0      -1/J
    1/L       0];
AT = [-1/(C_w*R_wh)     1/(C_w*R_wh)
       1/(C_h*R_wh)     (-1/(C_h*R_wh)-1/(C_h*R_ha))]; %A matrix for thermal
model
BT = [1/C_w           0
        0       1/(C_h*R_ha)]; % B matrix for thermal model
CT = [1     0
      0     1]; % Part of output matrix for thermal model
DT = [0     0
      0     0]; %Part of output matrix for thermal model
```

Tabular Data

This section includes the table and the code to obtain the table that compares manufacturer data to data found in the prelab (which I used code from Professor Charlie Revfem).

```
% Define parameters and values
ParameterNames = {'Nominal Voltage'; 'No-Load Speed'; 'Armature
Resistance';'Armature Inductance';'Rotor Inertia';'Viscous Damping
Coefficient';'Thermal Resistance (winding to housing)';'Thermal Resistance
(housing to ambient)';'Thermal Time Constant (winding)';'Thermal Time Constant
(Housing)';'Maximum Acceptable Winding Temperature';'Ambient
Temperature';'Torque Constant';'Back-emf constant';'Motor Constant';'Stall
Current';'Stall Torque';'Speed-Torque Gradient';'No-Load Current';'Maximum
Power';'Maximum Efficiency';'Rated Current';'Rated Torque';'Rated
Power';'Thermal Capacitance (Winding)';'Thermal Capacitance (Housing)'};  % Add
all parameter names

PreLabValues = {'18V';'11800rpm';'0.68Ohm';'0.078mH';'9.82gcm^2';'3.14E-
4mNmsec';'4.57K/W';'13.6K/W';'22sec';'646sec';'125C';'25C';'0.0146Nm/A';'0.0146V
sec/rad';'68.6rpm/V';'26.5Amps';'0.385Nm';'3208rad/Nmsec';'0.027Amps';'119W';'50
%';'2.84Amps';'0.041Nm';'45.7W';'4.81J/K';'47.5J/K'};
ManufacturerValues =
{'18V';'11800rpm';'0.68Ohm';'0.078mH';'9.82gcm^2';'NaN';'4.57K/W';'13.6K/W';'22s
```

```
ec';'646sec';'125C';'NaN';'1.46mNm/A';'NaN';'654rpm/V';'26.5Amps';'385mNm';'30.5
rpm/mNm';'54.6 Amps';'NaN';'NaN';'2.26Amps';'32.2mNm';'NaN';'NaN';'NaN'};

 % Create a table
 data = table(ParameterNames,PreLabValues,ManufacturerValues);

 % Display the table
 disp(data);
```

Table 1. Table containing values from manufacturer and the prelab.

ParameterNames	PreLabValues	ManufacturerValues
{'Nominal Voltage' }	{'18V' }	{'18V' }
{'No-Load Speed' }	{'11800rpm' }	{'11800rpm' }
{'Armature Resistance' }	{'0.68Ohm' }	{'0.68Ohm' }
{'Armature Inductance' }	{'0.078mH' }	{'0.078mH' }
{'Rotor Inertia' }	{'9.82gcm^2' }	{'9.82gcm^2' }
{'Viscous Damping Coefficient' }	{'3.14E-4mNmsec' }	{'NaN' }
{'Thermal Resistance (winding to housing)'}	{'4.57K/W' }	{'4.57K/W' }
{'Thermal Resistance (housing to ambient)'}	{'13.6K/W' }	{'13.6K/W' }
{'Thermal Time Constant (winding)' }	{'22sec' }	{'22sec' }
{'Thermal Time Constant (Housing)' }	{'646sec' }	{'646sec' }
{'Maximum Acceptable Winding Temperature' }	{'125C' }	{'125C' }
{'Ambient Temperature' }	{'25C' }	{'NaN' }
{'Torque Constant' }	{'0.0146Nm/A' }	{'1.46mNm/A' }
{'Back-emf constant' }	{'0.0146Vsec/rad'}	{'NaN' }
{'Motor Constant' }	{'68.6rpm/V' }	{'654rpm/V' }
{'Stall Current' }	{'26.5Amps' }	{'26.5Amps' }
{'Stall Torque' }	{'0.385Nm' }	{'385mNm' }
{'Speed-Torque Gradient' }	{'3208rad/Nmsec' }	{'30.5rpm/mNm'}
{'No-Load Current' }	{'0.027Amps' }	{'54.6 Amps' }
{'Maximum Power' }	{'119W' }	{'NaN' }
{'Maximum Efficiency' }	{'50%' }	{'NaN' }
{'Rated Current' }	{'2.84Amps' }	{'2.26Amps' }
{'Rated Torque' }	{'0.041Nm' }	{'32.2mNm' }
{'Rated Power' }	{'45.7W' }	{'NaN' }
{'Thermal Capacitance (Winding)' }	{'4.81J/K' }	{'NaN' }
{'Thermal Capacitance (Housing)' }	{'47.5J/K' }	{'NaN' }

Block Diagrams

This section shows the block diagrams for the three different motor cases. The function block contains the equations and matrices for the electro-mechanical system and the state space block contains equations for the thermal system. The diagram for the no load case is shown in **Figure 9.**

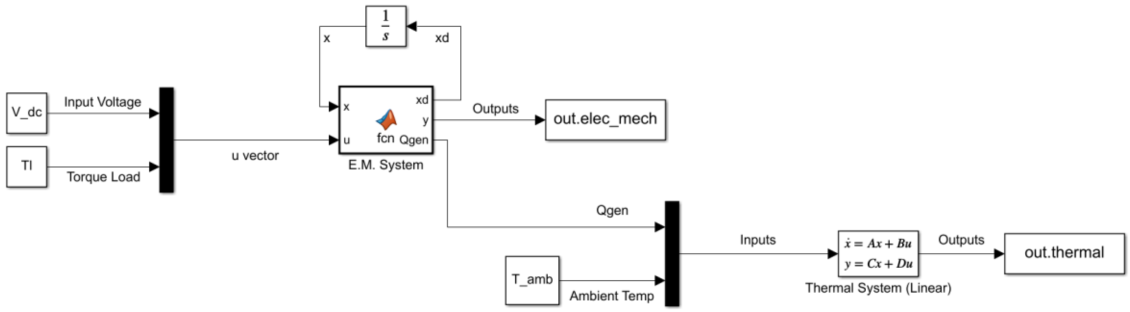

Figure 9. Block diagram to get the system response of the electro-mechanical and thermal components for the no load case.

The diagram for the loading case is shown in **Figure 10.**

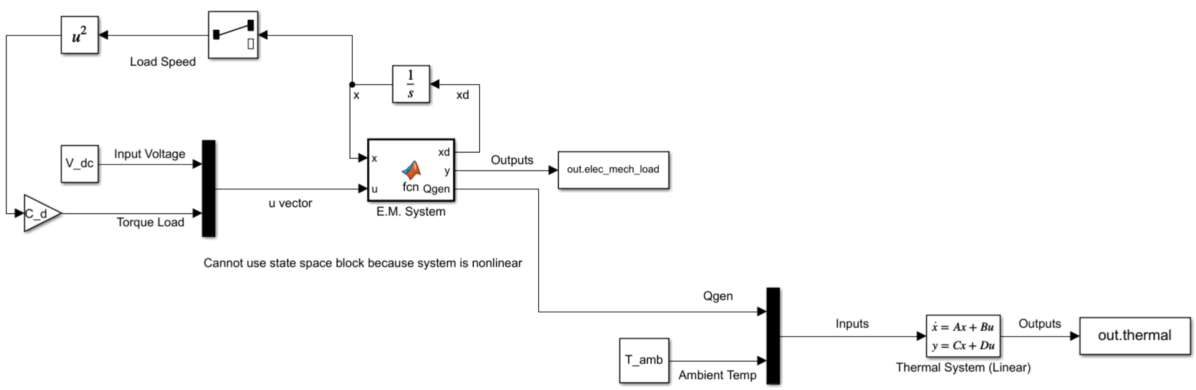

Figure 10. Block diagram to get the system response of the electro-mechanical and thermal components for the loading case.

The diagram for the stall case is shown in **Figure 11.**

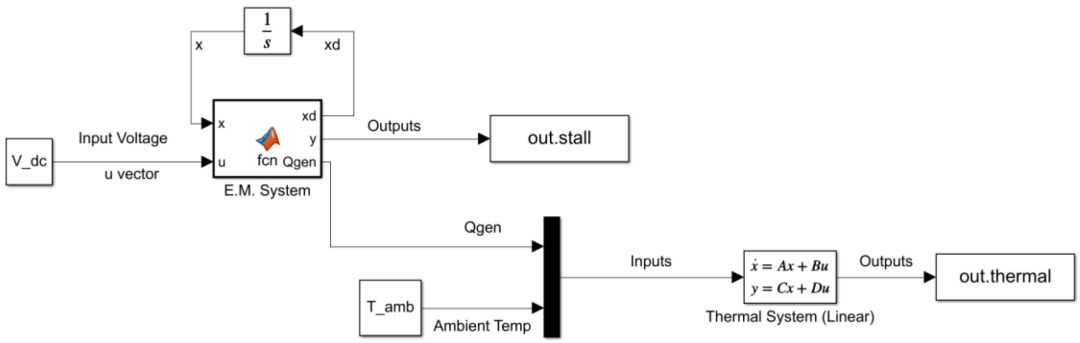

Figure 11. Block diagram to get the system response of the electro-mechanical and thermal components for the stall case.

Plots

This section will plot the outputs from the prelab and the Simulink block diagram outputs.

Plot 1 - Motor Curve

This section shows the motor curve for the permanent magnet DC motor that was found using the prelab values.

```
T_Plot = [0:.001:.4];
Omega_Plot = omega_noload-T_Plot*dNdM;
Power_Plot = T_Plot.*Omega_Plot;

Efficient_Plot = 100*(Kt/V_dc)*((omega_noload*T_Plot-
(dNdM)*T_Plot.^2)./((T_Plot*(1-b*dNdM))+b*omega_noload));
%Efficient_Plot = (omega_noload*T_Plot-
(dNdM)*T_Plot.^2)./((V_dc/Kt)*(T_Plot*(1-(1000*b)*dNdM))+(1000*b)*omega_noload);

% Create the figure
figure(8);
%addaxis(T_Plot, Omega_Plot,[0,1500]);
plot(T_Plot, Omega_Plot,'k');
%plot(T_Plot, Omega_Plot);
%yticks(0:500:1500)
ylim([0 1500])
ylabel('Speed (rad/s)','FontSize',8)
hold on;
addaxis(T_Plot, Power_Plot,[0,150]);
hold on;
addaxis(T_Plot, Efficient_Plot,[0,100],'LineWidth',2);
```

```
%addaxislabel(1,'Speed (rad/s)','FontSize',1);
addaxislabel(2,'Power (W)');
addaxislabel(3,'Efficiency (%)');
xlabel('Load Torque (N-m)');
title('Motor Curve Plot');
legend({'Speed','Power','Efficiency'},'Location','northeast');
```

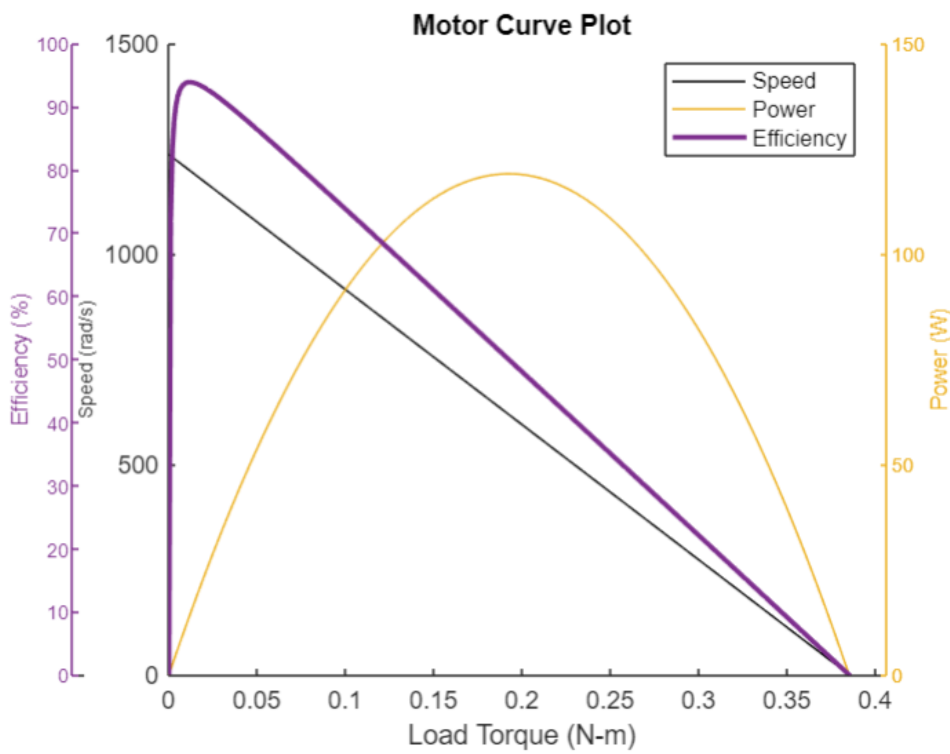

Figure 12. Plot showing the motor curve for the permanent magnet DC motor.

Plot 2 - Motor and System Curve

This sections shows the steady state speed-torque curve and system load curve as well as the operating point.

```
figure (2);
plot(T_Plot, Omega_Plot);
ylabel('Angular Velocity (rad/s)');
title('Motor Curve Plot');
xlabel('Load Torque (N-m)');
ylim([0 1500])
hold on
Omegal_Loaded = sqrt(T_Plot/C_d);
plot(T_Plot, Omegal_Loaded)
hold on;
```

```
intersectionx = .033;
intersectiony = 1128;
plot(intersectionx,intersectiony,'*','LineWidth', 2)
legend({'Steady State Speed','System Curve','Operating
Point'},'Location','northeast');
text(.04,1130,'Operating Point (.033,1128)'); %Creates text on graph and
specifies location
```

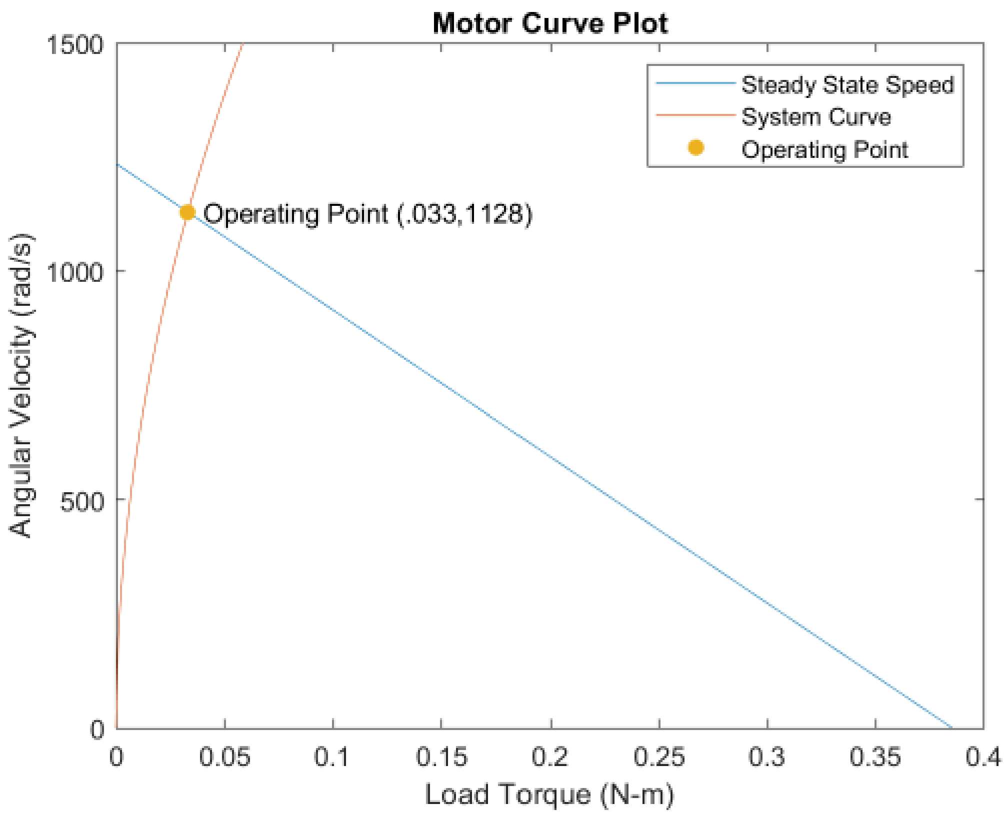

Figure 13. Plot showing the intersection between the speed curve and the system curve, which is the operating point of the motor.

Plot 3 - No Load System Response

This section shows the code and the outputs from Simulink and the state space method for the system with no load.

```
Tl = 0; %Sets load to 0
x0 = [0
      0]; %Sets initial conditions
A = [-b/J   Kv/J
    -Kv/L   -R/L]; %Creates A matrix for electromechanical system input
```

```
B = [0      -1/J
    1/L       0];


AT = [-1/(C_w*R_wh)     1/(C_w*R_wh)
       1/(C_h*R_wh)     (-1/(C_h*R_wh)-1/(C_h*R_ha))]; %A matrix for thermal
model

BT = [1/C_w           0
        0       1/(C_h*R_ha)]; % B matrix for thermal model
CT = [1     0
      0     1]; % Part of output matrix for thermal model

DT = [0     0
      0     0]; %Part of output matrix for thermal model

out = sim("LAB4_NOLOAD_MODEL_322", 5000); %This line is equivalent to pressing
"run" in Simulink
Elec_Mech_Data = squeeze(out.elec_mech); %Reduces 3D matrix to 2D


figure(1);
t = tiledlayout(4,2); %Creates a 4 row and 2 column layout for plots
title(t,'No Load Results'); %Creates a title for the tiled layout

nexttile(1) %Places plot in tile 1
plot(out.tout, Elec_Mech_Data(1,:))
xlim([0 .025])
ylim([0 1500])
xlabel({'Time, t[s]'},'FontSize',8) %Sets x axis label
ylabel({'Ang. Velocity, [rad/s]'},'FontSize',8) %Sets y axis label
xticks([0:.005:.025])
grid on %Turns on grid for plot

nexttile(3)
plot(out.tout, Elec_Mech_Data(2,:))

ylim([0 30])
xlabel({'Time, t[s]'},'FontSize',8) %Sets x axis label
ylabel({'Current, [A]'},'FontSize',8) %Sets y axis label
xticks([0:.005:.025])
xlim([0 .025])
grid on

nexttile(5)
plot(out.tout, Elec_Mech_Data(3,:))
```

```
hold on
plot(out.tout, Elec_Mech_Data(4,:))
xlim([0 .025])
ylim([0 450])
xlabel({'Time, t[s]'},'FontSize',8) %Sets x axis label
ylabel({'Power, [W]'},'FontSize',8) %Sets y axis label
legend({'Input Power','Output Power'},'Location','northeast','FontSize',6);
xticks([0:.005:.025])
grid on

nexttile(7)
plot(out.tout, Elec_Mech_Data(5,:))
xlim([0 .025])
ylim([-1 1])
xlabel({'Time, t[s]'},'FontSize',8) %Sets x axis label
ylabel({'Efficiency'},'FontSize',8) %Sets y axis label
xticks([0:.005:.025])
grid on

nexttile(2, [2 1]) %Places plot in tile 2 and makes it a 2 row 1 column sized
plot
plot(out.tout, out.thermal(:,1))
xlim([0 5000])
ylim([25 25.16])
xlabel({'Time, t[s]'},'FontSize',8) %Sets x axis label
ylabel({'Winding Temperature, [C]'},'FontSize',8) %Sets y axis label
yticks([25:.02:25.16])
grid on

nexttile(6, [2 1])
plot(out.tout, out.thermal(:,2))
xlim([0 5000])
ylim([25 25.014])
xlabel({'Time, t[s]'},'FontSize',8) %Sets x axis label
ylabel({'Housing Temperature, [C]'},'FontSize',8) %Sets y axis label
grid on
yticks([25:.002:25.014])
```

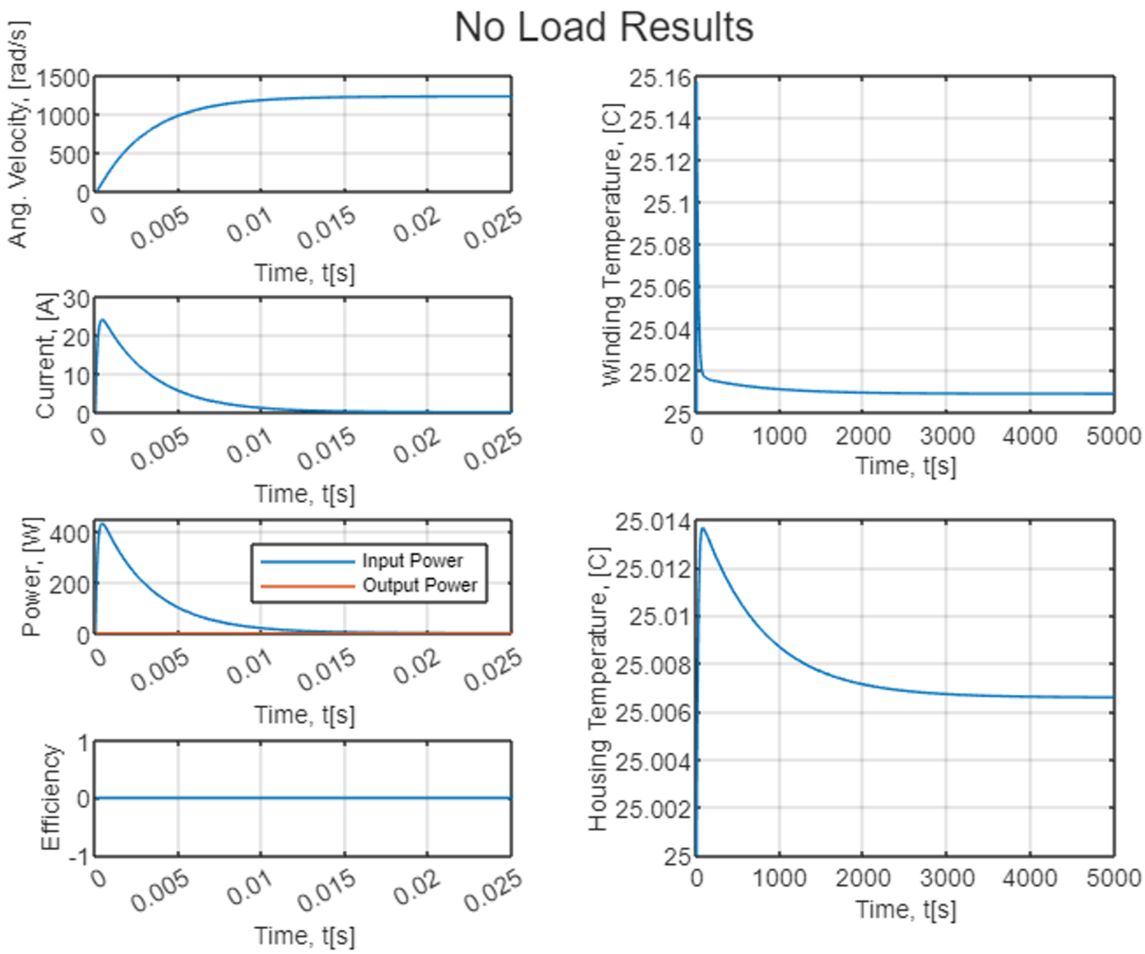

Figure 14. Plot showing the system response values for the no load condition.

Plot 4 - Loaded System Response

This section shows the code and the outputs from Simulink and the state space method for the system with a load.

```
C_d = 2.6E-8; %Drag Coefficent for fan

x0 = [0
      0];

A = [-b/J   Kv/J
    -Kv/L   -R/L];

B = [0      -1/J
    1/L       0];
```

```
AT = [-1/(C_w*R_wh)     1/(C_w*R_wh)
       1/(C_h*R_wh)     (-1/(C_h*R_wh)-1/(C_h*R_ha))];

BT = [1/C_w           0
        0       1/(C_h*R_ha)];

CT = [1     0
      0     1];

DT = [0     0
      0     0];

out = sim("LAB4_LOADING_MODEL_322", 5000); %This line is equivalent to pressing
"run" in Simulink
Elec_Mech_Data_Load = squeeze(out.elec_mech_load);
Thermal_Load = out.thermal;


figure(1);
t = tiledlayout(4,2); %Creates a 3 row and 1 column for plots
title(t,'Loaded Results'); %Creates a title for the tiled layout


nexttile(1)
plot(out.tout, Elec_Mech_Data_Load(1,:))
xlim([0 .025])
ylim([0 1200])
xlabel({'Time, t[s]'},'FontSize',8) %Sets x axis label
ylabel({'Ang. Velocity, [rad/s]'},'FontSize',8) %Sets y axis label
xticks([0:.005:.025])
grid on

nexttile(3)
plot(out.tout, Elec_Mech_Data_Load(2,:))
xlim([0 .025])
ylim([0 30])
xlabel({'Time, t[s]'},'FontSize',8) %Sets x axis label
ylabel({'Current, [A]'},'FontSize',8) %Sets y axis label
xticks([0:.005:.025])
grid on

nexttile(5)
plot(out.tout, Elec_Mech_Data_Load(3,:))
hold on
```

```
plot(out.tout, Elec_Mech_Data_Load(4,:))
xlim([0 .025])
ylim([0 450])
xlabel({'Time, t[s]'},'FontSize',8) %Sets x axis label
ylabel({'Power, [W]'},'FontSize',8) %Sets y axis label
legend({'Input Power','Output Power'},'Location','northeast','FontSize',6);
xticks([0:.005:.025])
grid on

nexttile(7)
plot(out.tout, Elec_Mech_Data_Load(5,:))
xlim([0 .025])
ylim([0 1])
xlabel({'Time, t[s]'},'FontSize',8) %Sets x axis label
ylabel({'Efficiency'},'FontSize',8) %Sets y axis label
xticks([0:.005:.025])
xticks([0:.2:1])
grid on

nexttile(2, [2 1])
plot(out.tout,out.thermal(:,1))
xlim([0 5000])
ylim([20 100])
xlabel({'Time, t[s]'},'FontSize',8) %Sets x axis label
ylabel({'Winding Temperature, [C]'},'FontSize',8) %Sets y axis label

grid on

nexttile(6, [2 1])
plot(out.tout,out.thermal(:,2))
xlim([0 5000])
ylim([20 80])
xlabel({'Time, t[s]'},'FontSize',8) %Sets x axis label
ylabel({'Housing Temperature, [C]'},'FontSize',8) %Sets y axis label
grid on
```

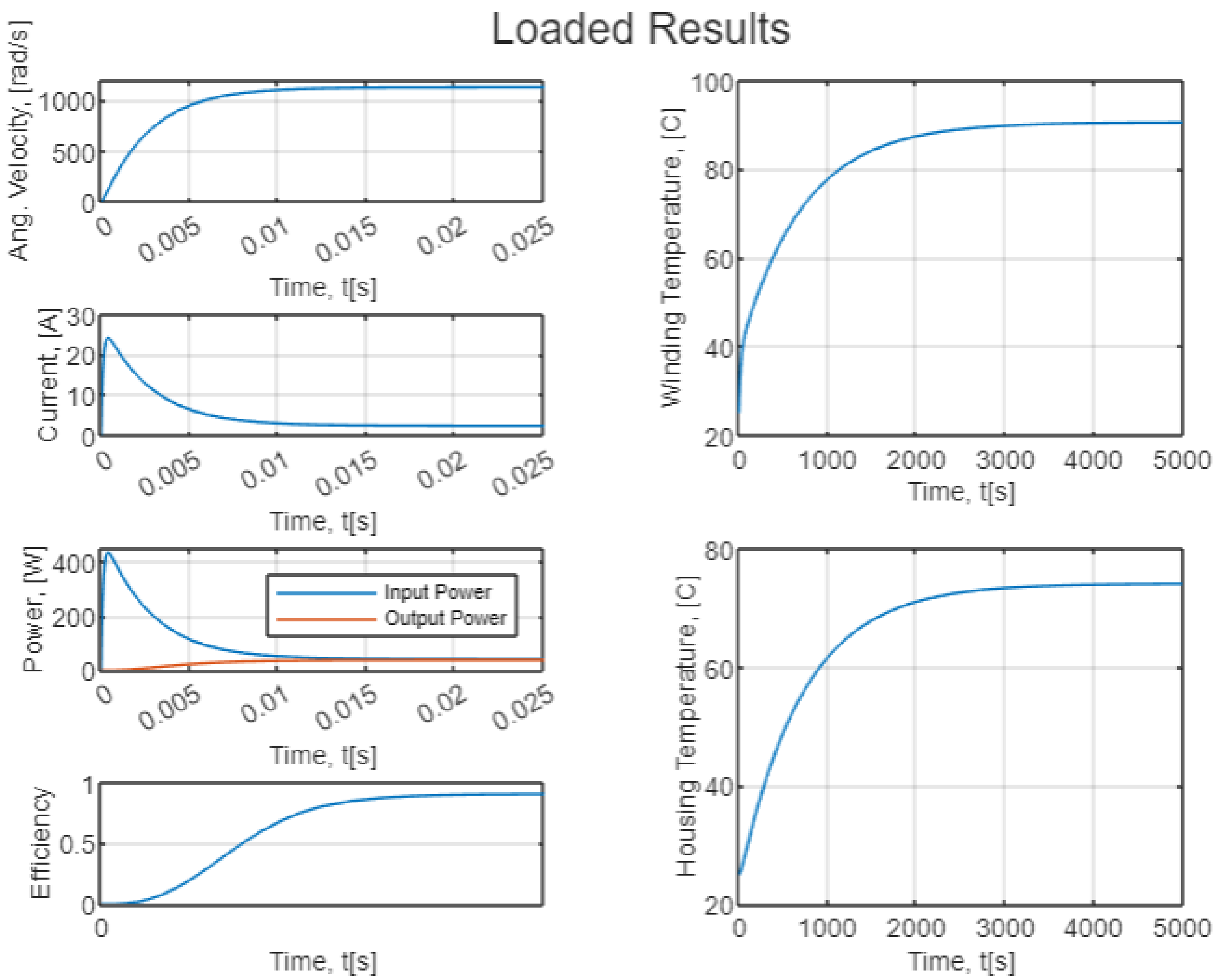

Figure 15. Plot showing the system response values for the loaded condition.

Plot 5 -Stall System Response

This section shows the code and the outputs from Simulink and the state space method for the system at stall conditions.

```
A = [-R/L];
B = [1/L];
x0 = [0];

out = sim("LAB4_STALL_MODEL_322", 5000); %This line is equivalent to pressing
"run" in Simulink
Stall_Data = squeeze(out.stall);


figure(1);
```

```
t = tiledlayout(4,2); %Creates a 3 row and 1 column for plots
title(t,'Stall Torque Results'); %Creates a title for the tiled layout


nexttile(1)
plot(out.tout, Stall_Data(1,:))
xlim([0 .025])
ylim([0 1500])
xlabel({'Time, t[s]'},'FontSize',8) %Sets x axis label
ylabel({'Ang. Velocity, [rad/s]'},'FontSize',8) %Sets y axis label
xticks([0:.005:.025])
ylim([-1 1])
grid on

nexttile(3)
plot(out.tout, Stall_Data(2,:))
ylim([0 30])
xlabel({'Time, t[s]'},'FontSize',8) %Sets x axis label
ylabel({'Current, [A]'},'FontSize',8) %Sets y axis label
xticks([0:.2E-3:1E-3])
xlim([0 1E-3])
grid on

nexttile(5)
plot(out.tout, Stall_Data(3,:))
hold on
plot(out.tout, Stall_Data(4,:))
ylim([0 485])
xlabel({'Time, t[s]'},'FontSize',8) %Sets x axis label
ylabel({'Power, [W]'},'FontSize',8) %Sets y axis label
legend({'Input Power','Output Power'},'Location','northeast','FontSize',6);
xticks([0:.2E-3:1E-3])
yticks([0:100:400])
xlim([0 1E-3])
grid on

nexttile(7)
plot(out.tout, Stall_Data(5,:))
xlim([0 .025])
ylim([-1 1])
xlabel({'Time, t[s]'},'FontSize',8) %Sets x axis label
ylabel({'Efficiency'},'FontSize',8) %Sets y axis label
xticks([0:.005:.025])
grid on

nexttile(2, [2 1])
```

```
plot(out.tout, out.thermal(:,1))
xlim([0 5000])
ylim([0 9000])
xlabel({'Time, t[s]'},'FontSize',8) %Sets x axis label
ylabel({'Winding Temperature, [C]'},'FontSize',8) %Sets y axis label
yticks([0:1000:9000])
grid on

nexttile(6, [2 1])
plot(out.tout, out.thermal(:,2))
xlim([0 5000])
ylim([0 7000])
xlabel({'Time, t[s]'},'FontSize',8) %Sets x axis label
ylabel({'Housing Temperature, [C]'},'FontSize',8) %Sets y axis label
grid on
yticks([0:1000:7000])
```

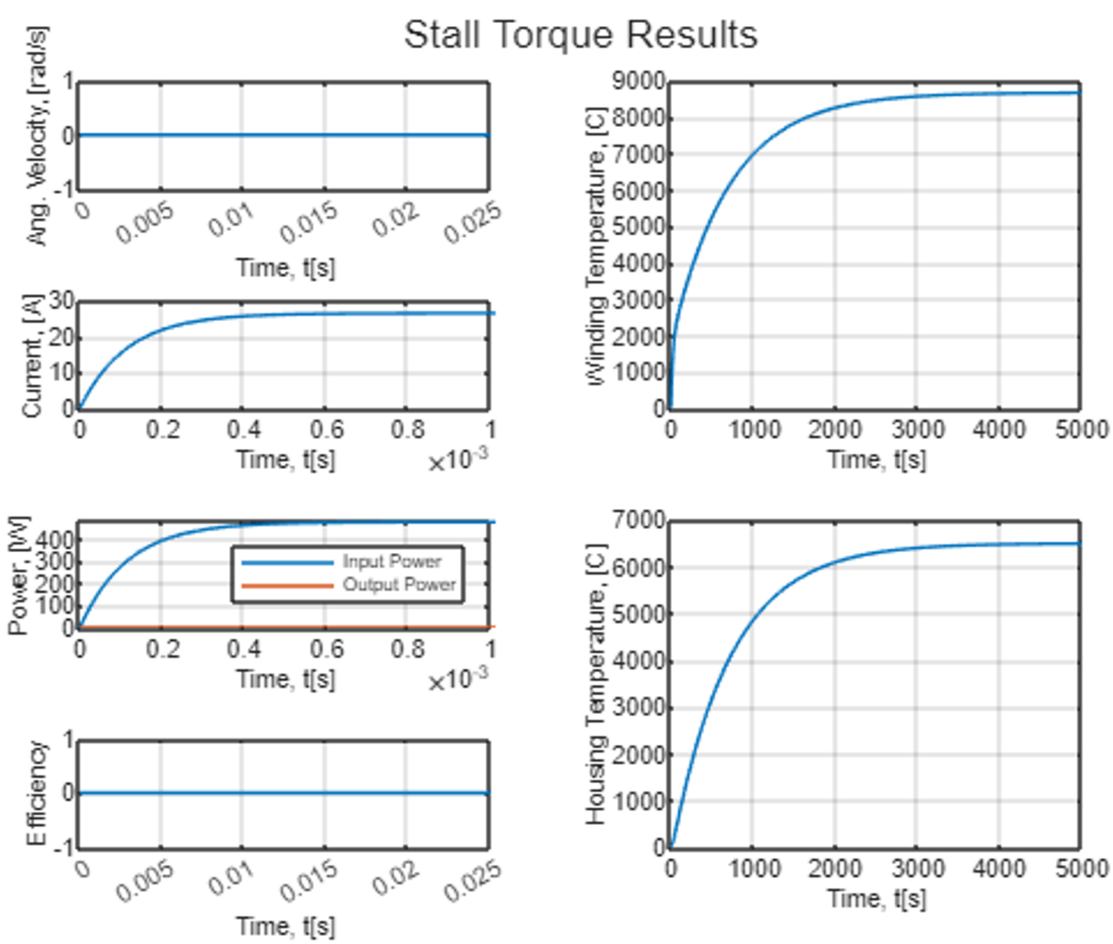

Figure 16. Plot showing the transmissibility ratio and phase data plotted versus frequency.

Discussion Questions

Question #1

Examine your second plot showing the motor speed-torque curve overlaid with the system load curve.

• What does the intersection represent? • Does your simulation output match the conditions shown on this steady-state plot?

The intersection represents the operating point of the motor under the specified conditions. The simulation output matches the conditions of the steady state plot in that angular velocity is increasing with increasing torque.

Question #2

The majority of DC motors are not designed to operate at or near stall conditions due to the buildup of heat in the motor windings. Specifically, what is the main cause of failure for a motor in a stall condition? Hint: almost all motors fail at temperatures between 125 [◦C] and 180 [◦C].

The main cause of motor failure in stall conditions is due to the degradation of the insulation, bearing, magnets/magnetic field, and also degradation to any lubricants because they were not meant to operate at the high temperatures of stall.

Question #3

DC motors are most efficient near their no-load speed. Provide a compelling and intuitive explanation why this is the case. In your explanation, quantitatively compare the two sources of heat generation in the motor: resistive losses in the windings and frictional losses in the bearings; these two losses can be compared by considering their relative magnitudes in worst-case circumstances - either stall or no-load.

Motors are most efficient near their no load speed because there is no mechanical load that the motor needs to overcome, which results in less current being drawn so there is less heat generation, and the frictional losses in the bearings are much less than the resistive losses in the windings at high torque. When comparing the No Load and the Stall Torque graphs, this is shown because the temperatures in the motors are very different with the no load temperatures being very close to ambient temperatures while the stall temperatures keep increasing until the motor fails.

Question #4

How can the safe operating current of a motor be increased without modifying the internal components of the motor itself? That is, what could you add to the motor or change about the motor's environment to allow a larger operating current without exceeding the motors temperature rating?

The safe operating current of a motor can be increased by increasing the heat transfer out of the coils. This can be done by placing the motor in a cold environment to increase the temperature difference, or through introducing a fan that aims at the motor to increase the convection heat transfer coefficient. Both would result in less heat building up in the motor over time.

Question #5

Calculate the steady-state temperature rise predicted by the thermal model during a stall condition. You should find the computed value to be unrealistically high. What unmodeled behavior might limit the temperature rise in a real motor? Reconsider your response to Question 2 above while answering this question.

The steady state temperature rise for the windings would be 8700 degrees Celsius. The unmodeled behavior that limits the temperature would be that components would fail, the heat transfer would not be only conduction (there would be radiation), and the heat distribution would not be one dimensional.

Question #6

Even though DC motors should remain at or below their nominal current ratings during steady operation, they can greatly exceed these ratings for short durations. Explain why this is the case. In your response you may want to talk about the "duty cycle" of the motor operation or the "I^2 t" principle.

They can exceed these ratings for a short duration because there is not enough time for significant heat to build up in the windings which would cause failure if operating above nominal ratings for too long. The duty cycle of a motor has short periods where they exceed nominal current ratings, but then has longer periods of much lower current draw, which allows for the motor to cool down.

Question #7

The square-law load on the motor requires some special consideration to simulate properly in both forward and backward directions. That is, your simulation may fail if your motor velocity ends up negative unless you take certain precautions. What issues may arrive when running the motor with negative velocity? How can you account for this in your block diagram to allow negative velocities without issue?

If the motor velocity ends up negative, it will result in the fan reaching an infinite velocity because the torque will always be in the same direction as the angular velocity, which mean speed will always be increasing. This can be accounted for by multiplying the angular velocity by the absolute angular velocity, which retains the sign, instead of squaring the term.

Question #8

To properly model the stall conditions for the DC motor you were asked to produce a second system model with reduced order. Explain, in your own words, why this approach better represents stall conditions as compared to applying a constant load torque equal to the stall torque. You may want to try running your general model with the load torque input equal to the stall torque to help answer this question effectively. In your explanation describe both: the effects corrected by usage of the reduced order model, and, why the reduced order model provides this correction.

Applying a constant load torque equal to the stall torque is a bad approach because it fails to capture the dynamic behavior of the system where the voltage increases up to its maximum value. The reduced order models corrects the issue of not capturing the dynamic effects of the system by starting all values (current, torque, etc) at zero, like a motor would when first being started.

Question #9

DC motors are reversible due to the lossless transduction within the motor. That is, a motor can also act as a generator. Explain, in your own words, how the model developed to represent our motor may be used to represent a generator.

• *How does our perspective on the model change and what sort of inputs and outputs would be of interest looking at our motor as a generator? You may want to consider the four “quadrants” representing different operating regimes for motors - which quadrants represent “motoring” and which represent “generating”?*

• *How can you determine the quadrant for a motor based on the state variables used in this simulation (im and Ωm).*

The model could represent a generator by having angular velocity and torque be an input and the current and voltage being an output, so the mechanical power would be generating electrical power. The quadrants that represent "motoring" is when electrical power is being used to create mechanical power, and the generating quadrants is when electrical power is being made. The four quadrants are because positive power can create positive power, negative power can create negative power, and negative power can create positive power for both electrical and mechanical.

The quadrant for a motor can be determined based on which state variables are positive or negative and what the input and output are to the system.