ELEX 4336: Feedback Systems

LAB 6 – DC Motor, Gearbox and Encoder

Student Name: Taeyoon Rim	Student Number: A01160214	Set: B

1 Calculation (Open-Loop System)

1.1 System Parameters

Description	Datasheet Symbol	Symbol	Value	Unit
Rated Voltage	V_r	V_{rated}	24	V
Rated Torque	T_r	T_{rated}	0.5	N·m
Rated Speed	ω	ω_{rated}	366	rpm
Motor Constant	K_T	k_m	0.0458	N·m/A
Resistance	R_{mt}	R_f	2.49	Ω
Inductance	L	L_f	0.00263	H
Rotor Inertia	J_r	J	0.0000071	kg·m^2
Friction Constant	-	b	0.0130*	N·m·s
Gear Ratio	-	N	11.5	-

$* \, T = (Js + b)\omega, \; T_{rated} = b\omega_{rated}, \; b = \frac{T_{rated}}{\omega_{rated}} = \frac{(0.5 \, N\cdot m)}{(366 \, rpm \times \frac{2\pi}{1 \, rev} \times \frac{1 \, min}{60 \, s})} \approx 0.0130 \, N \cdot m \cdot s$

1.2 Transfer Functions

From these equations:

$$v_f(s) = \left(R_f + L_f s\right) \cdot I_f(s)$$
$$T(s) = k_m \cdot I_f(s)$$
$$T(s) = \omega(s)[Js + b]$$

Transfer functions of this system can be calculated as:

$$\frac{\omega(s)}{v_f(s)} = \frac{k_m}{n\left(R_f + L_f s\right)(Js + b)} = \frac{k_m}{nJL_f s^2 + n\left(JR_f + bL_f\right)s + nbR_f}$$
$$= \frac{0.0458}{\left(2.15 \cdot 10^{-7} \, \frac{kg \cdot m^2}{A}\right)s^2 + \left(5.98 \cdot 10^{-4} \, \frac{kg \cdot m^2}{A \cdot sec}\right)s + (3.74 \cdot 10^{-1} \, \frac{kg \cdot m^2}{A \cdot sec^2})}$$

$$\frac{\theta(s)}{v_f(s)} = \frac{\omega(s)}{v_f(s)} \cdot \frac{1}{s} = \frac{k_m}{ns\left(R_f + L_f s\right)(Js + b)} = \frac{k_m}{nJLs^3 + n\left(JR_f + bL_f\right)s^2 + nbR_f s}$$
$$= \frac{0.0458}{\left(2.15 \cdot 10^{-7} \, \frac{kg \cdot m^2}{A}\right)s^3 + \left(5.98 \cdot 10^{-4} \, \frac{kg \cdot m^2}{A \cdot sec}\right)s^2 + (3.74 \cdot 10^{-1} \, \frac{kg \cdot m^2}{A \cdot sec^2})s}$$

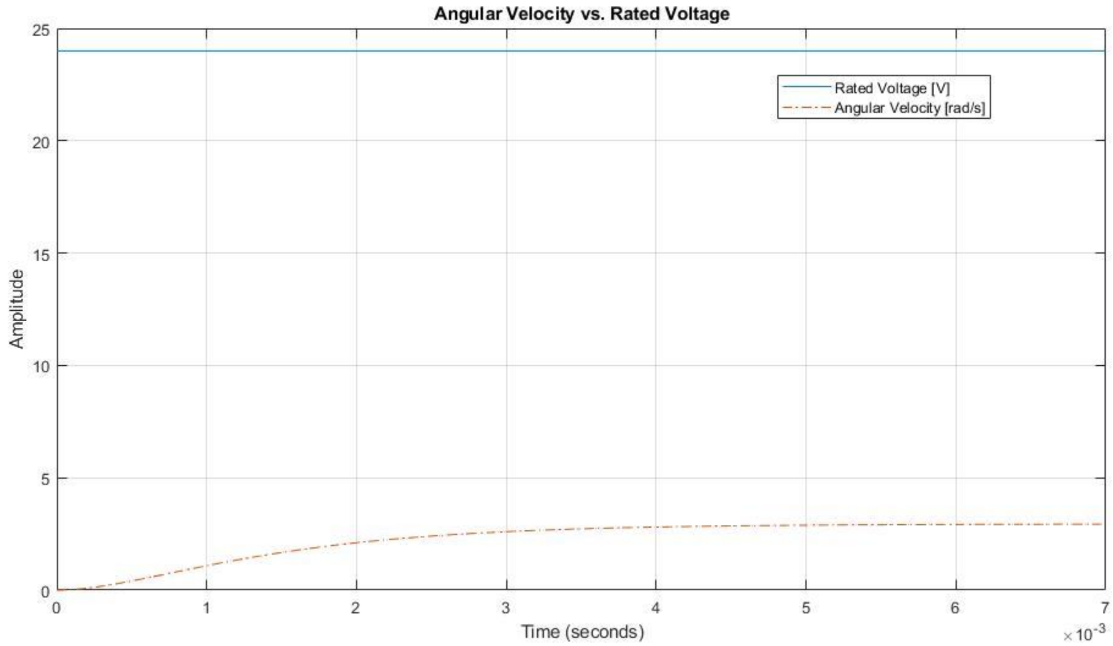
2 Open-Loop System Step Response (MATLAB)

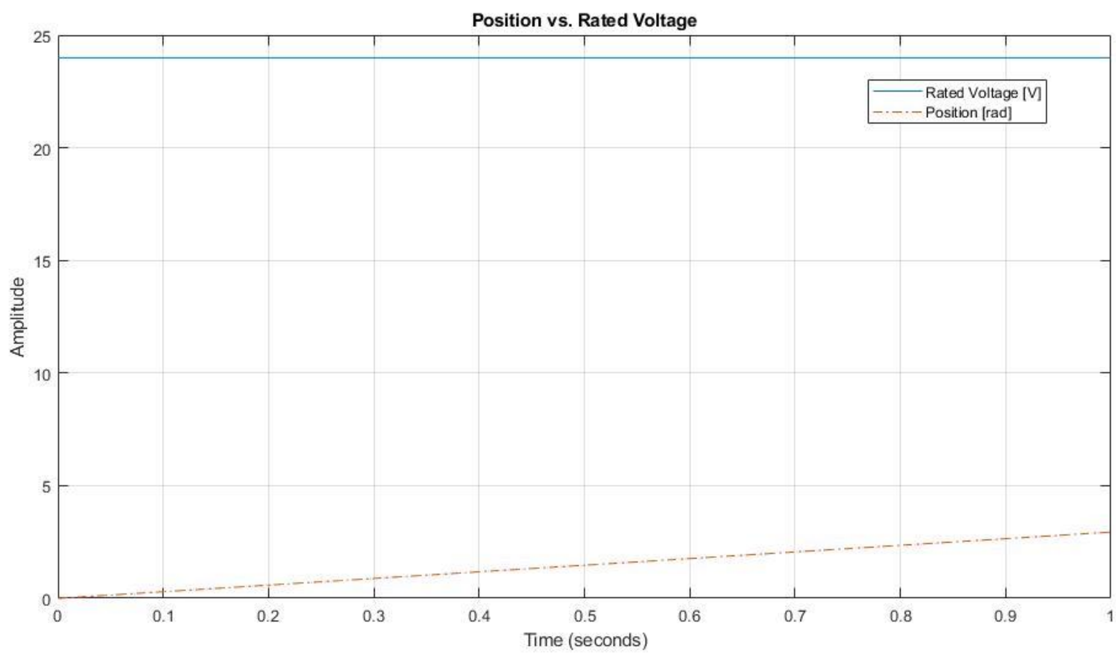
2.1 MATLAB Code

```
% Plot the step response of the open-loop system in MATLAB

% system parameters
Vrated = 24;                          % V
Trated = 0.5;                         % N*m
Wrated = 366;                         % rpm
km = 0.0458;                          % N*m/A
Rf = 2.49;                            % Ohms
Lf = 2.63E-3;                         % H
J = 0.0000071;                        % kg*m^2
b = Trated / (Wrated * 2*pi / 60);    % N*m*s
n = 11.5;                             % dimensionless

% transfer functions
sys1 = tf([km], [n*J*Lf n*(J*Rf+b*Lf) n*b*Rf]);     % angular velocity output
sys2 = tf([km], [n*J*Lf n*(J*Rf+b*Lf) n*b*Rf 0]);   % position output

% calculate and save step response for angular velocity output
t = 0:7E-6:7E-3;
x = step(tf(Vrated, 1), t); % step input
[out, t] = lsim(sys1, x, t);
save('open-loop_ang-vel.mat', 't', 'x', 'out');

% calculate and save step response for angular position output
t = 0:0.001:1;
x = step(tf(Vrated, 1), t); % step input
[out, t] = lsim(sys2, x, t);
save('open-loop_pos.mat', 't', 'x', 'out');

% plot the step response for angular velocity output
figure(1);
load open-loop_ang-vel;
plot(t, x, '-', t, out, '-.');
ylabel('Amplitude');
xlabel('Time (seconds)');
title('Angular Velocity vs. Rated Voltage');
legend('Rated Voltage [V]', 'Angular Velocity [rad/s]');
grid on;

% plot the step response for angular position output
figure(2);
load open-loop_pos;
plot(t, x, '-', t, out, '-.');
ylabel('Amplitude');
xlabel('Time (seconds)');
title('Position vs. Rated Voltage');
legend('Rated Voltage [V]', 'Position [rad]');
grid on;
```

2.2 MATLAB Plot

Input/output signals and time of the system step response saved as .mat files:



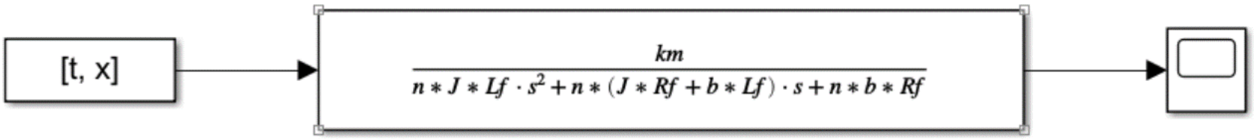


Plots were generated using the transfer functions with 24 V (rated voltage) step input

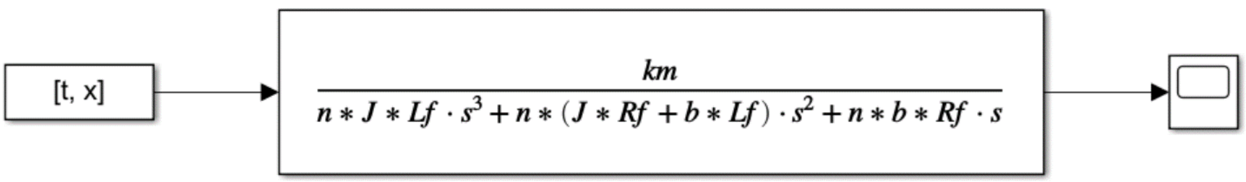
3 Open-Loop System Step Response (Simulink)

3.1 Simulink Model

Model for angular velocity output:

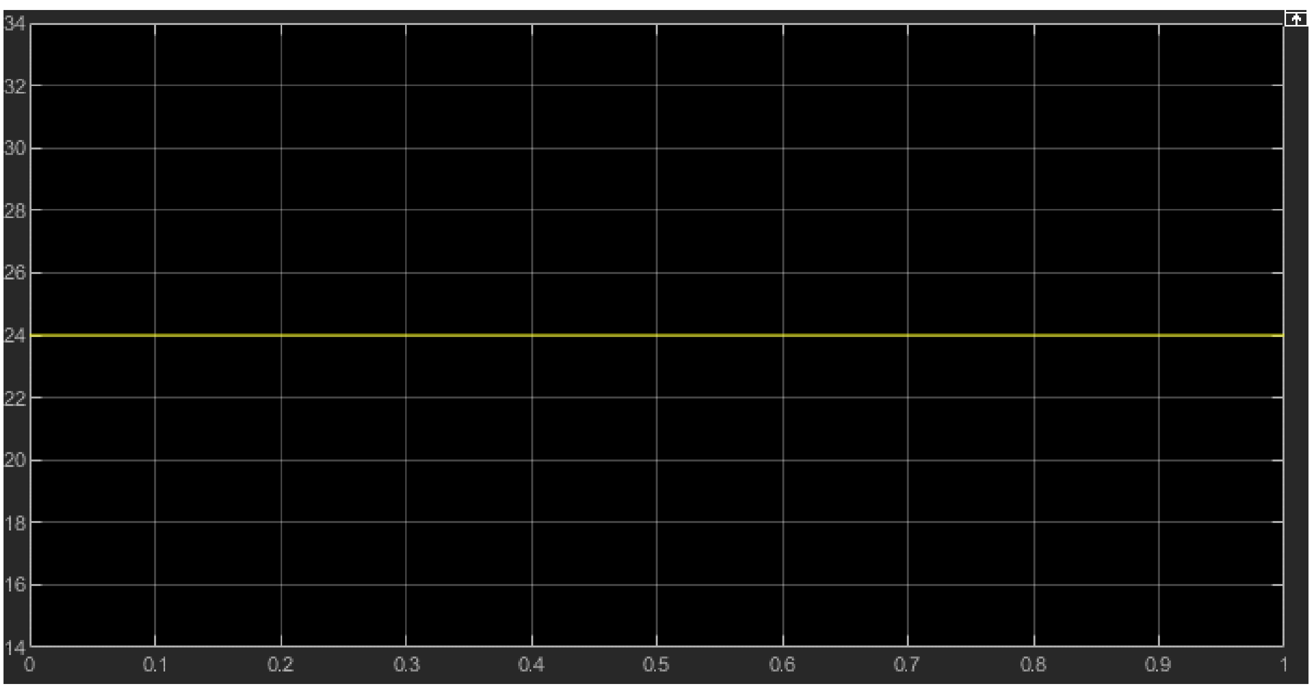


Model for angular position output:

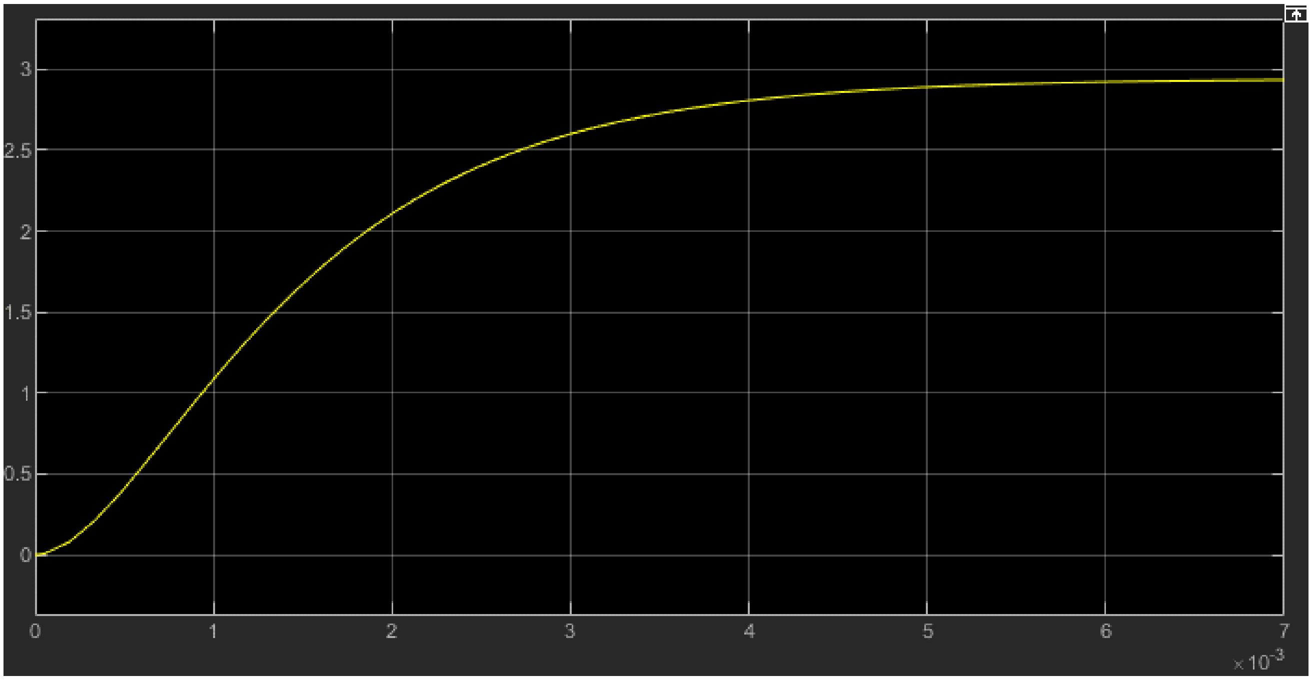


3.2 Simulink Scope

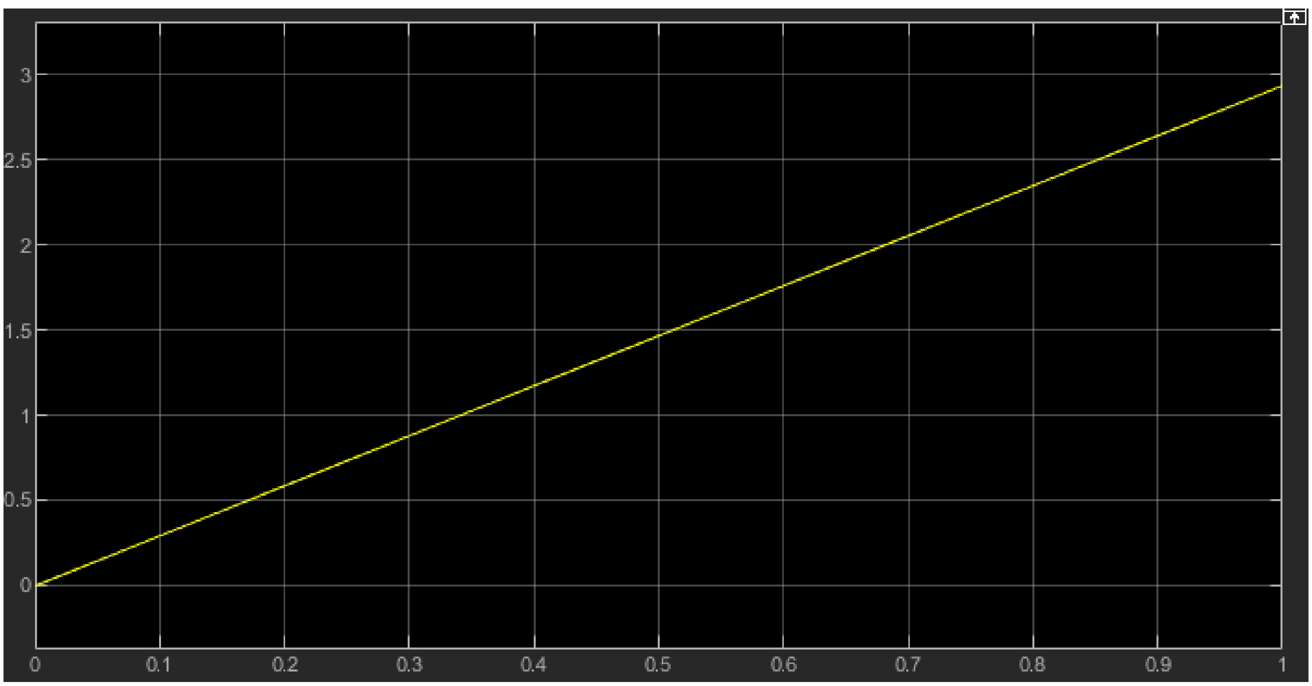
Plot of voltage step input:



Plot of angular velocity output:



Plot of angular position output:

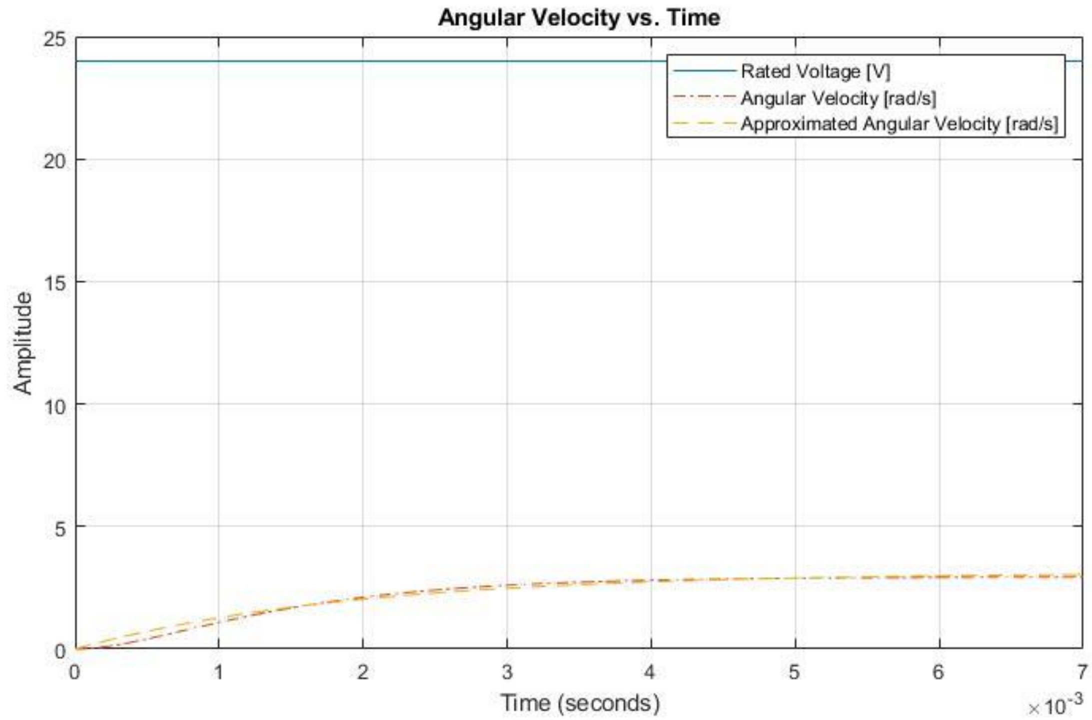


4 Open-Loop Transfer Function Approximation (1st Order)

4.1 MATLAB Code

```
% Model Equation: T(s) = K / (τ*s + 1)
% Approximate the angular velocity as first order system
% using a sample input-output data

load open-loop_ang-vel;

% Array indices
idxStart = 1;
idxEnd = 1001;

% Calculate zero means
Input = x(idxStart:idxEnd);
Output = out(idxStart:idxEnd);

% Find best fit K and tau
K = 0;
tau = 0;
minSSE = -1;

for kTest = 0.12:0.001:0.14
    for tauTest = 0.0018:0.00001:0.0020
        sys = tf([kTest], [tauTest 1]);
        OutputSim = lsim(sys, Input, t(idxStart:idxEnd));
        SSE = sum((Output - OutputSim) .^ 2);

        if minSSE == -1
            minSSE = SSE;
            K = kTest;
            tau = tauTest;
        elseif SSE < minSSE
            minSSE = SSE;
            K = kTest;
            tau = tauTest;
        end
    end
end

% Calculated internal temperature
sys = tf([K], [tau 1]);
OutputSim = lsim(sys,Input, t(idxStart:idxEnd));

% Plot the data
figure(3);
t = t(idxStart:idxEnd);
plot(t, Input, '-', t, Output, '-.', t, OutputSim, '--');
ylabel('Amplitude');
xlabel('Time (seconds)');
title('Angular Velocity vs. Time');
legend('Rated Voltage [V]', 'Angular Velocity [rad/s]', 'Approximated Angular
Velocity [rad/s]');
grid on;
```

4.2 MATLAB Plot



4.3 Description

The transfer function for the system was approximated using a first order model:

$$T(s) = \frac{k}{\tau s + 1}$$

I approximated the transfer function by manually narrowing down the range of the test parameters:

k			τ			k	τ	SSE	Time taken (s)
Min	Step	Max	Min	Step	Max				
0	0.1	3	3	1000	2000000	0.1	0	5922.5	17.034
0	0.1	3	3	10	1000	0.1	0	5921.1	9.902
0	0.1	3	3	0.1	10	2.5	0.1	5786.8	8.971
0	0.1	3	3	0.01	0.2	0.3	0.01	1242	2.753
0	0.1	2	2	0.0001	0.01	0.2	0.0049	136.86	6.637
0	0.01	0.3	0.3	0.0001	0.005	0.13	0.0019	15.3	5.727
0.12	0.001	0.14	0.14	0.00001	0.002	0.13	0.0019	15.3	1.97

The transfer function of the system was approximated as first order model using similar method that was used in Lab 5. Previously, I have been iterating through wide range of test parameters in a high resolution for double nested loop; it took over 40 minutes to approximate the function.

This time, I tried to reduce the iteration time by manually narrowing down the range of the test parameters. I started with reasonable ranges of k and τ in low resolutions. Then I increased the accuracy and the precision of the approximation by increasing the resolution.

The errors for the approximation was calculated using below expressions:

$$E(t) = \omega(t) - \omega_{approx}(t)$$

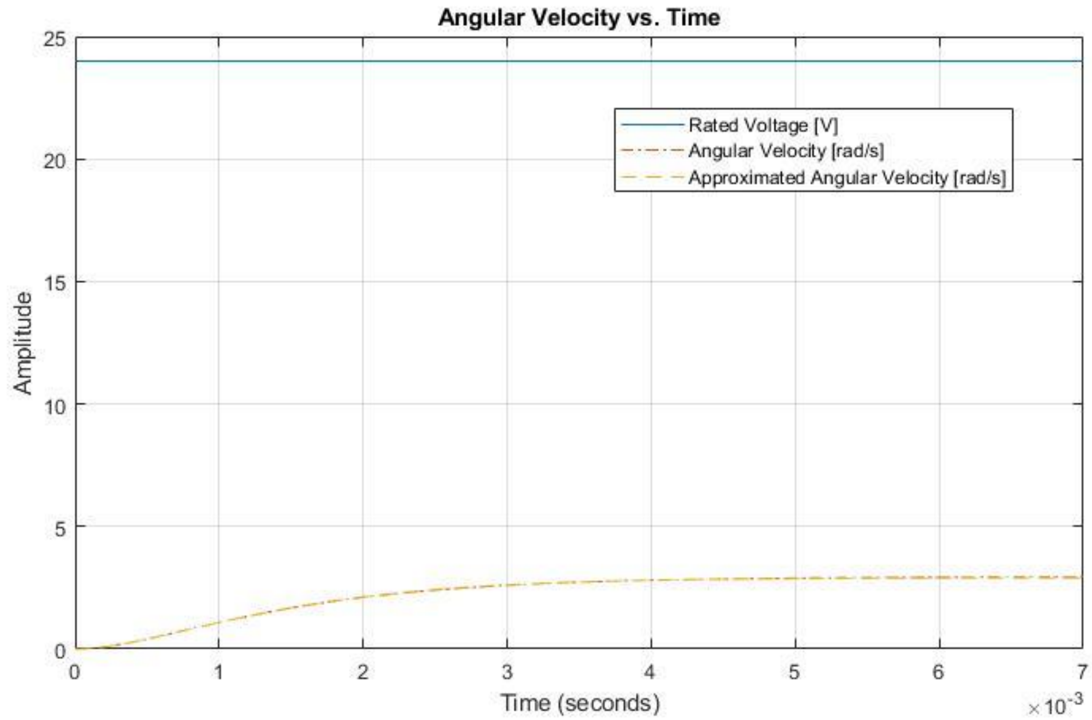
$$SSE = \sum_{k=0}^{T} [(E(k))^2]$$

5 Open-Loop Transfer Function Approximation (2nd Order)

5.1 MATLAB Code

```
% Model Equation: T(s) = K*omega^2 / (s^2 + 2*zeta*omega*s + omega^2)
% Approximate the angular velocity as second order system
% using a sample input-output data

load open-loop_ang-vel;

% Array indices
idxStart = 1;
idxEnd = 1001;

% Calculate zero means
Input = x(idxStart:idxEnd);
Output = out(idxStart:idxEnd);

% Find best fit K and tau
K = 0; %0.125;
zeta = 0; %2.1;
omega = 0; %1318;
minSSE = -1;

for kTest = 0.12:0.001:0.14
    for zetaTest = 1:0.1:3
        for omegaTest = 1000:100:2000
            sys = tf([kTest*omegaTest^2], [1 2*zetaTest*omegaTest omegaTest^2]);
            OutputSim = lsim(sys, Input, t(idxStart:idxEnd));
            SSE = sum((Output - OutputSim) .^ 2);

            if minSSE == -1
                minSSE = SSE;
                K = kTest;
                zeta = zetaTest;
                omega = omegaTest;
            elseif SSE < minSSE
                minSSE = SSE;
                K = kTest;
                zeta = zetaTest;
                omega = omegaTest;
            end
        end
    end
end

% Calculated internal temperature
sys = tf([K*omega^2], [1 2*zeta*omega omega^2]);
OutputSim = lsim(sys,Input, t(idxStart:idxEnd));

% Plot the data
figure(4);
t = t(idxStart:idxEnd);
plot(t, Input, '-', t, Output, '-.', t, OutputSim, '--');
ylabel('Amplitude');
xlabel('Time (seconds)');
title('Angular Velocity vs. Time');
legend('Rated Voltage [V]', 'Angular Velocity [rad/s]', 'Approximated Angular
Velocity [rad/s]');
grid on;
```

5.2 MATLAB Plot



5.3 Description

The transfer function for the system was approximated using a first order model:

$$T(s) = \frac{k \cdot {\omega_n}^2}{s^2 + 2\zeta\omega_n s + {\omega_n}^2}$$

I approximated the transfer function by manually narrowing down the range of the test parameters:

k			ζ			ω_n			k	ζ	ω_n	SSE	Time taken
Min	Step	Max	Min	Step	Max	Min	Step	Max					
0	0.1	1	0	10	100	0	20000	200000	0.2	100	40000	137.49	4.843
0.1	0.01	0.3	0	10	200	30000	1000	50000	0.13	40	42000	15.28	24.63
0.12	0.001	0.14	30	1	50	30000	1000	50000	0.130	38	40000	15.27	25.705
0.13	0.01	0.13	0	2	40	0	1000	42000	0.13	2	2000	9.97	3.808
0.12	0.001	0.14	1	0.1	3	1000	100	2000	0.120	1.0	1300	0.33	15.103

The residual errors and SSE was calculated as shown in Step 4.

6 Calculation (Closed-Loop System)

6.1 Transfer Functions

From these equation:

$$T_{closed}(s) = \frac{T(s)}{1 + T(s)[3.11\,V/rad]}$$

New transfer functions for the closed-loop system can be calculated as:

$$\frac{\omega(s)}{v_f(s)} = \frac{k_m}{nJL_f s^2 + n(JR_f + bL_f)s + nbR_f + (3.11\,V/rad)k_m}$$

$$= \frac{0.0458}{\left(2.15 \cdot 10^{-7}\ \frac{kg \cdot m^2}{A}\right)s^2 + \left(5.98 \cdot 10^{-4}\ \frac{kg \cdot m^2}{A \cdot sec}\right)s + (5.16 \cdot 10^{-1}\ \frac{kg \cdot m^2}{A \cdot sec^2})}$$

$$\frac{\theta(s)}{v_f(s)} = \frac{k_m}{nJLs^3 + n(JR_f + bL_f)s^2 + nbR_f s + (3.11\,V/rad)k_m}$$

$$= \frac{0.0458}{\left(2.15 \cdot 10^{-7}\ \frac{kg \cdot m^2}{A}\right)s^3 + \left(5.98 \cdot 10^{-4}\ \frac{kg \cdot m^2}{A \cdot sec}\right)s^2 + \left(3.74 \cdot 10^{-1}\ \frac{kg \cdot m^2}{A \cdot sec^2}\right)s + (14.2 \cdot 10^{-1}\ \frac{kg \cdot m^2}{A \cdot sec^3})}$$

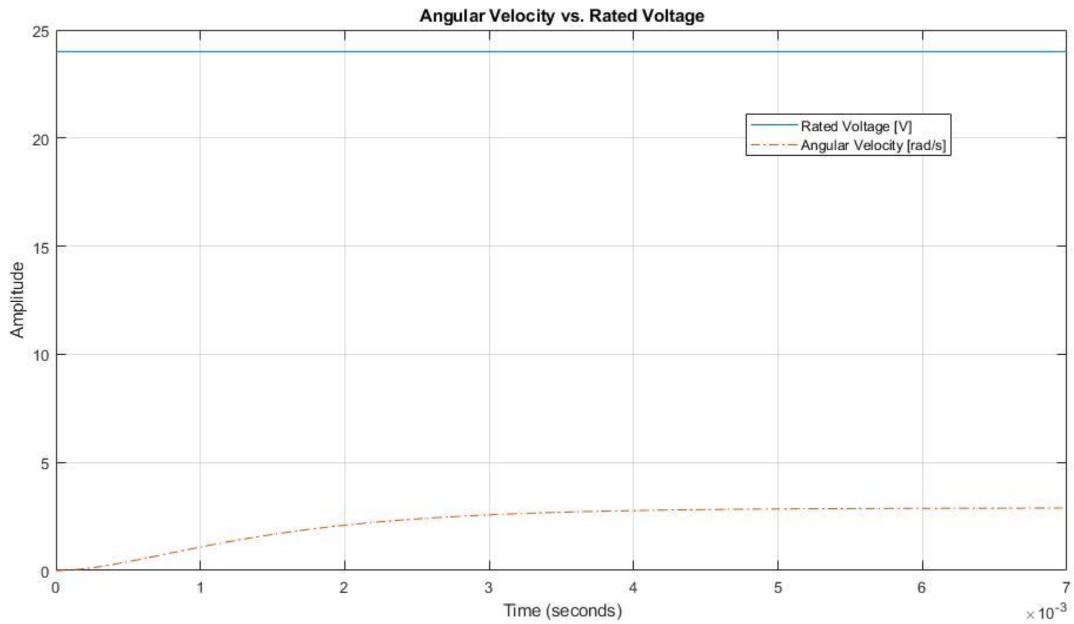
7 Closed-Loop System Step Response (MATLAB)

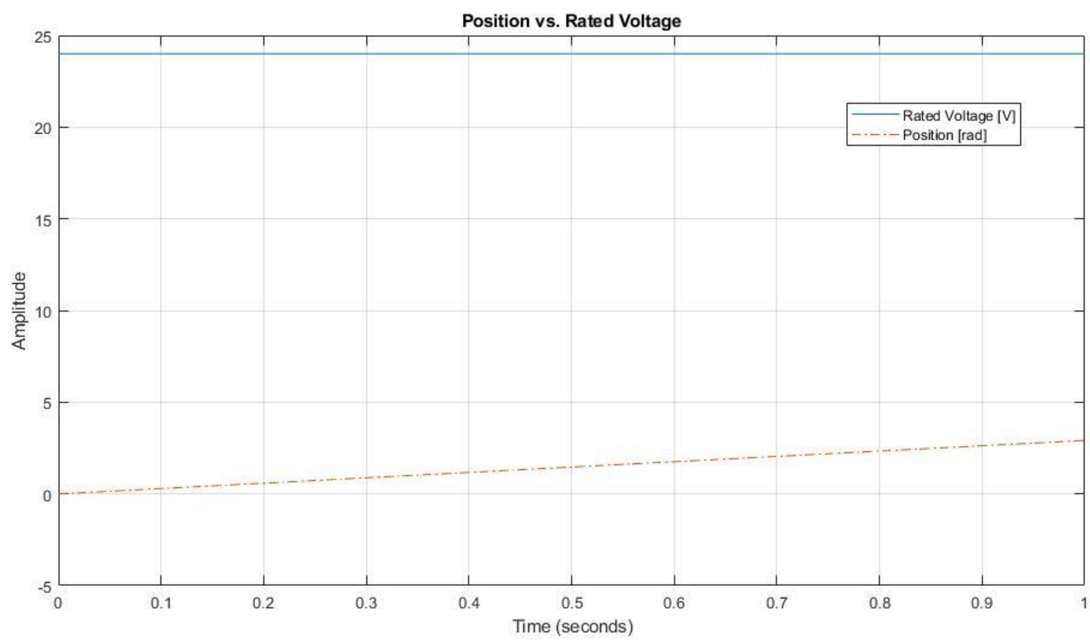
7.1 MATLAB Code

```
% Plot the step response of the closed-loop system in MATLAB

% system parameters
Vrated = 24;                        % V
Trated = 0.5;                       % N*m
Wrated = 366;                       % rpm
km = 0.0458;                        % N*m/A
Rf = 2.49;                          % Ohms
Lf = 2.63E-3;                       % H
J = 0.0000071;                      % kg*m^2
b = Trated / (Wrated * 2*pi / 60);  % N*m*s
n = 11.5;                           % dimensionless

% optical encoder feedback
H = 3.11 * km;                      % kg*m^2/A/s^3; 3.11 V/rad

% transfer functions
sys1 = tf([km], [n*J*Lf n*(J*Rf+b*Lf) n*b*Rf]);
sys1 = feedback(sys1, H, -1);                       % angular velocity output
sys2 = tf([km], [n*J*Lf n*(J*Rf+b*Lf) n*b*Rf 0]);
sys2 = feedback(sys2, H, -1);                       % angular position output

% calculate and save step response for angular velocity output
t = 0:7E-6:7E-3;
x = step(tf(Vrated, 1), t); % step input
[out, t] = lsim(sys1, x, t);
save('closed-loop_ang-vel.mat', 't', 'x', 'out');

% calculate and save step response for angular position output
t = 0:0.001:1;
x = step(tf(Vrated, 1), t); % step input
[out, t] = lsim(sys2, x, t);
save('closed-loop_pos.mat', 't', 'x', 'out');

% plot the step response for angular velocity output
figure(1);
load closed-loop_ang-vel;
plot(t, x, '-', t, out, '-.');
ylabel('Amplitude');
xlabel('Time (seconds)');
title('Angular Velocity vs. Rated Voltage');
legend('Rated Voltage [V]', 'Angular Velocity [rad/s]');
grid on;

% plot the step response for position output
figure(2);
load closed-loop_pos;
plot(t, x, '-', t, out, '-.');
ylabel('Amplitude');
xlabel('Time (seconds)');
title('Position vs. Rated Voltage');
legend('Rated Voltage [V]', 'Position [rad]');
grid on;
```

7.2 MATLAB Plot

Input/output signals and time of the system step response saved as .mat files:



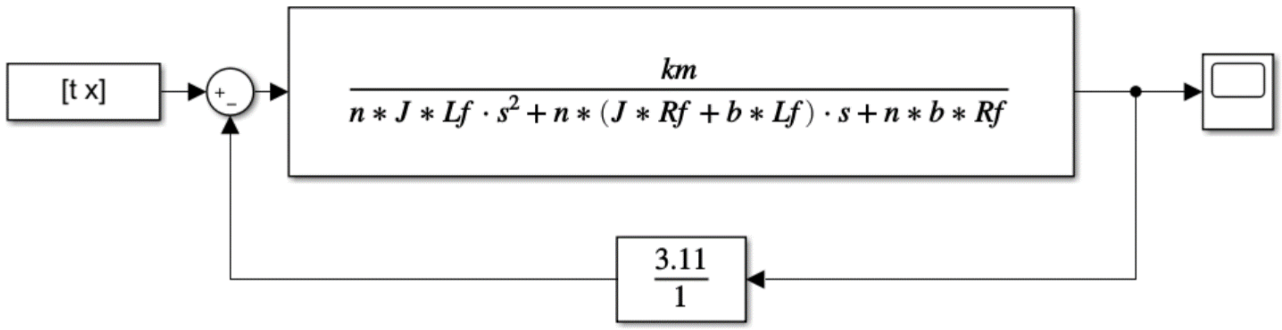


Plots were generated using the transfer functions with 24 V (rated voltage) step input

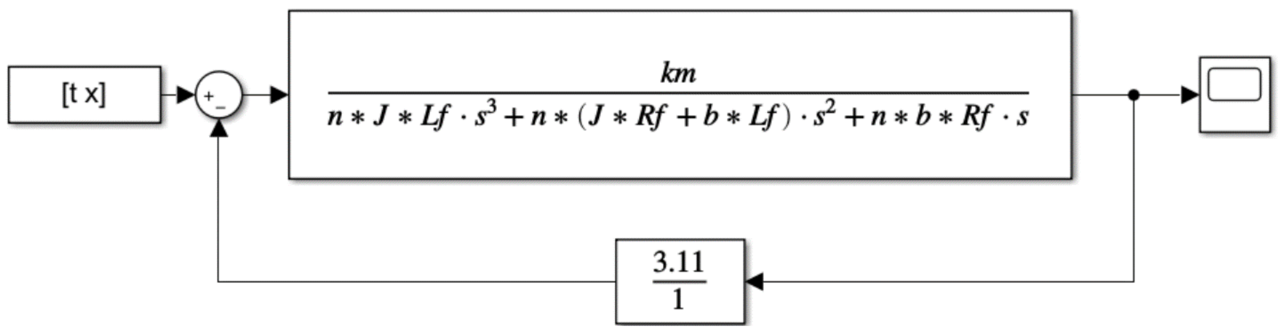
8 Closed-Loop System Step Response (Simulink)

8.1 Simulink Model

Model for angular velocity output:

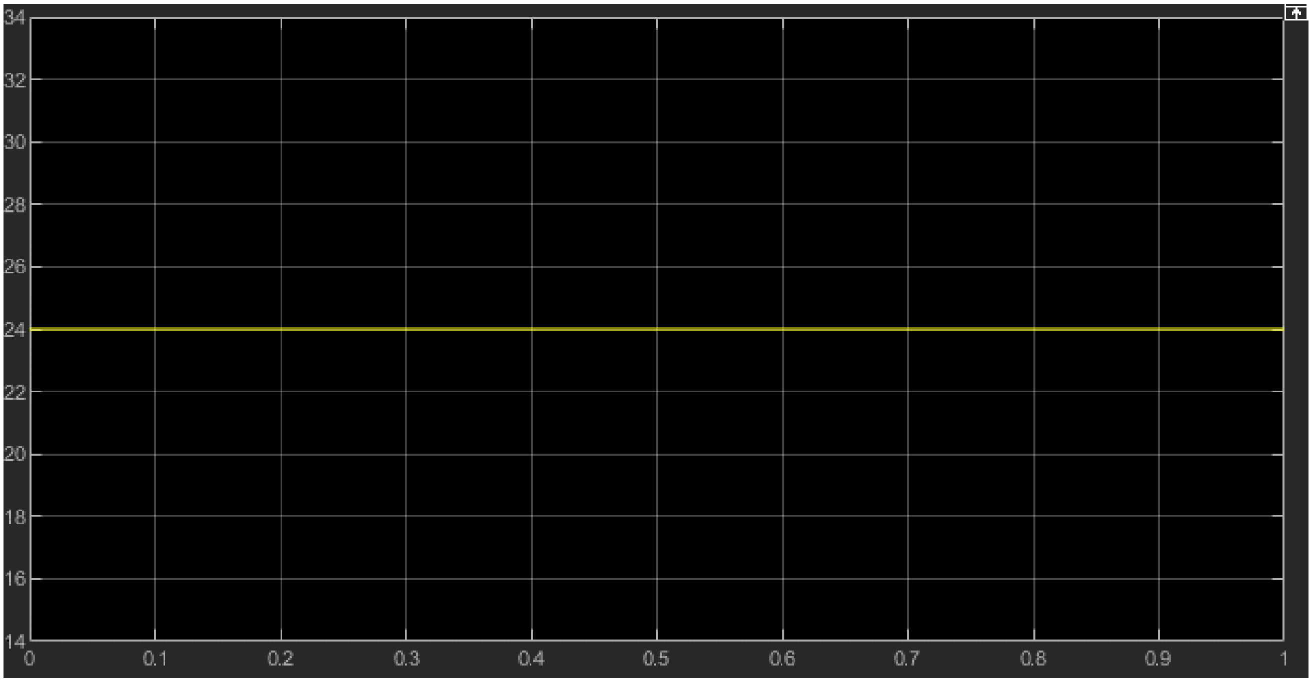


Model for angular position output:

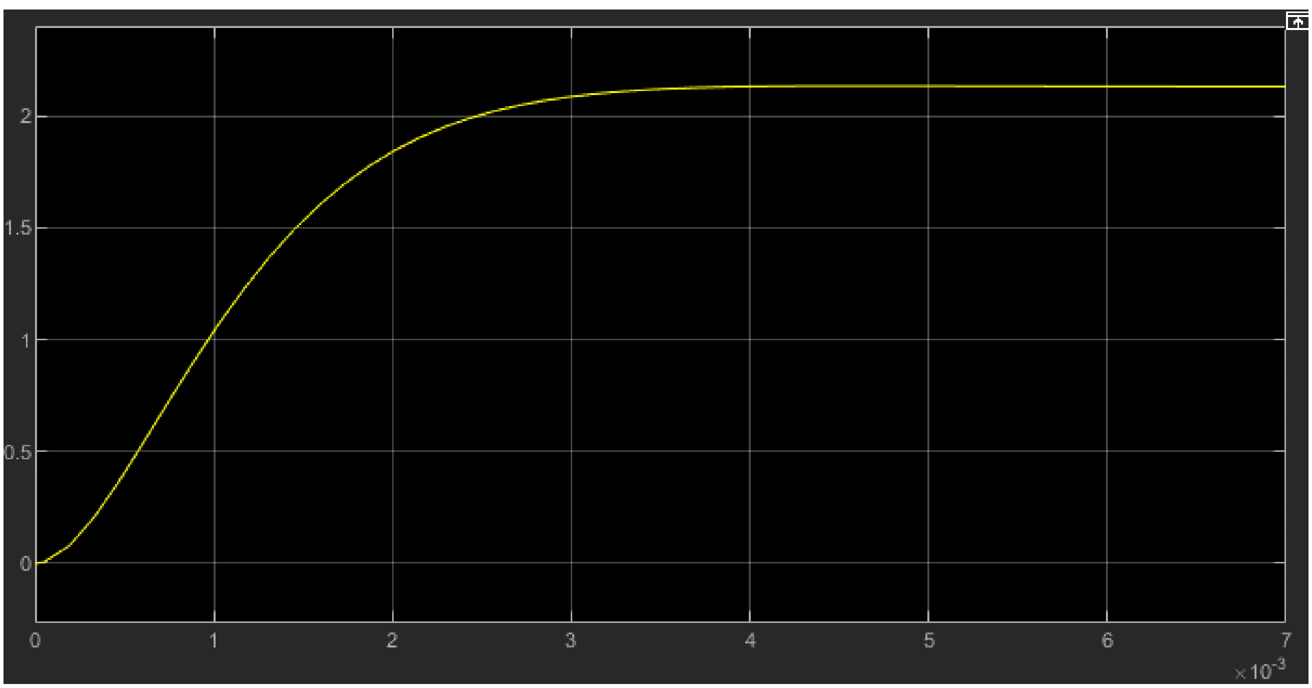


8.2 Simulink Scope

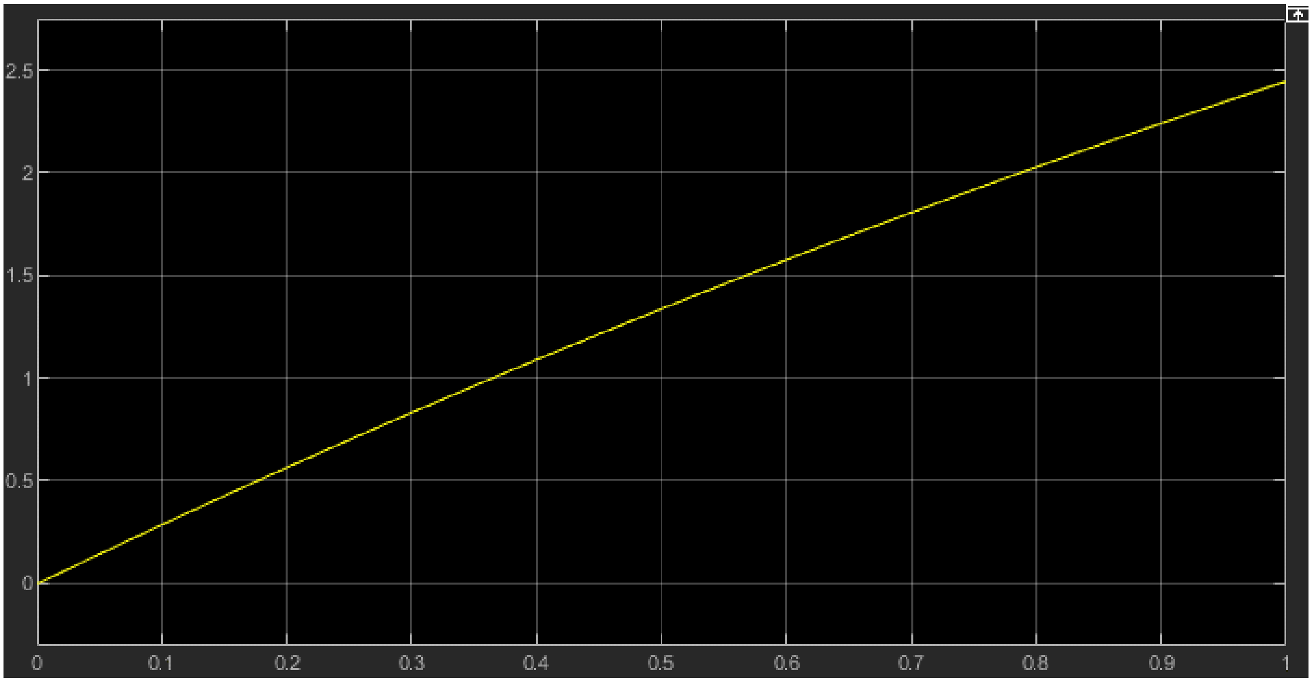
Plot of voltage step input:



Plot of angular velocity output:



Plot of angular position output:

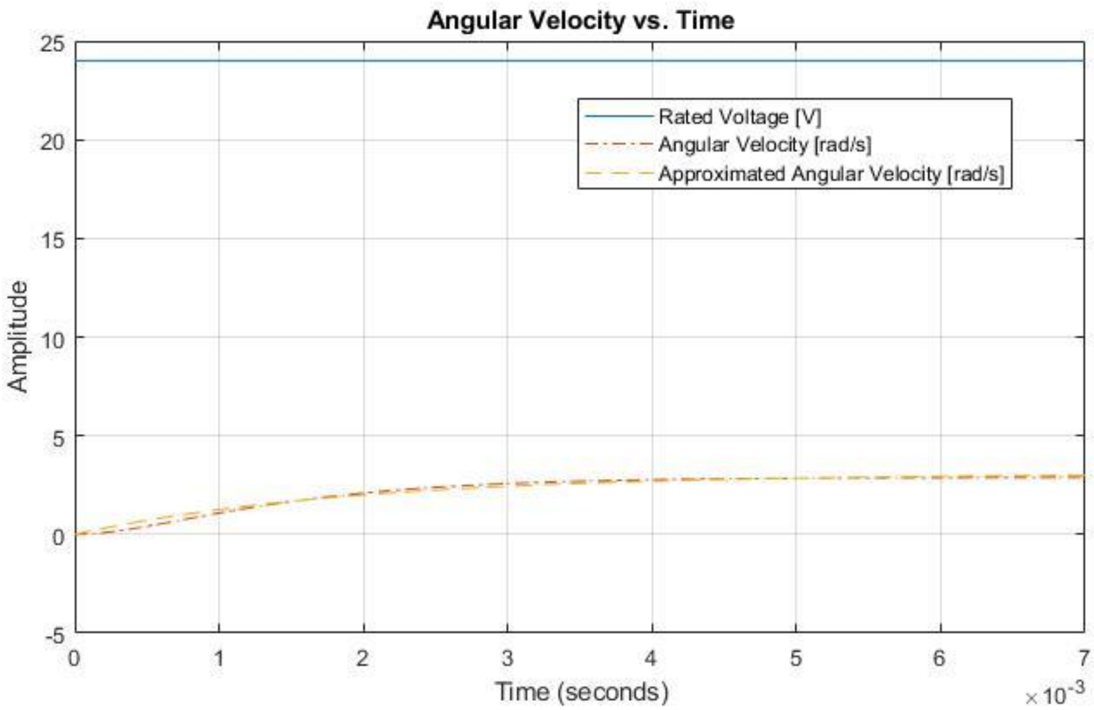


9 Closed-Loop Transfer Function Approximation (1st Order)

9.1 MATLAB Code

MATLAB Code for Step 4 was reused with different .mat file

9.2 MATLAB Plot



9.3 Description

The transfer function for the system was approximated as shown in Step 4.

I approximated the transfer function by manually narrowing down the range of the test parameters:

k			τ			k	τ	SSE	Time taken (s)
Min	Step	Max	Min	Step	Max				
0	0.1	1	0	0.1	1	0.1	0	160.82	1.916
0	0.1	1	0	0.01	0.1	0.3	0.01	258.72	1.622
0	0.1	1	0	0.001	0.02	0.2	0.005	143.55	1.956
0.1	0.01	0.3	0.004	0.0001	0.006	0.17	0.004	98.87	1.754
0.1	0.01	0.3	0.002	0.0001	0.008	0.13	0.002	16.07	5.301
0.12	0.001	0.14	0.001	0.0001	0.003	0.128	0.0019	15.24	2.317

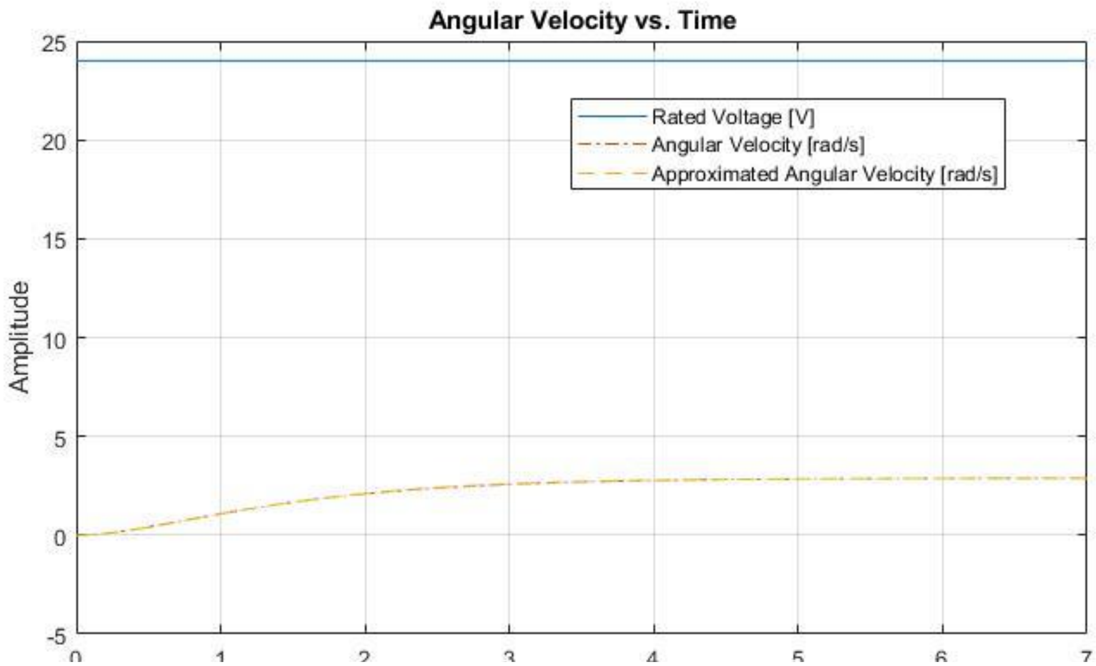
The residual errors and SSE was calculated as shown in Step 4.

10 Closed-Loop Transfer Function Approximation (2nd Order)

10.1 MATLAB Code

MATLAB Code for Step 5 was reused with different .mat file

10.2 MATLAB Plot



10.3 Description

The transfer function for the system was approximated as shown in Step 5.

I approximated the transfer function by manually narrowing down the range of the test parameters:

k			ζ			ω_n			k	ζ	ω_n	SSE	Time taken
Min	Step	Max	Min	Step	Max	Min	Step	Max					
0.1	0.01	0.2	0	1	10	0	100	2000	0.12	1	1300	0.1321	10.049

The residual errors and SSE was calculated as shown in Step 4.