ECE 3413 Lab 08
DC Motor Model for Open-loop Control in MATLAB Simulink

Leomar Durán

12th April 2023

1 Introduction

In this experiment, we build the DC motor using a series negative feedback close-loop system, testing both the angular velocity and position to ensure that the motor works as expected.

We will also experiment with the gains, moment of inertia, and the mechanical system's damping ratio as parameters of the DC motor's simulation.

2 Procedure

2.1 Task 0 – Modeling the DC Motor

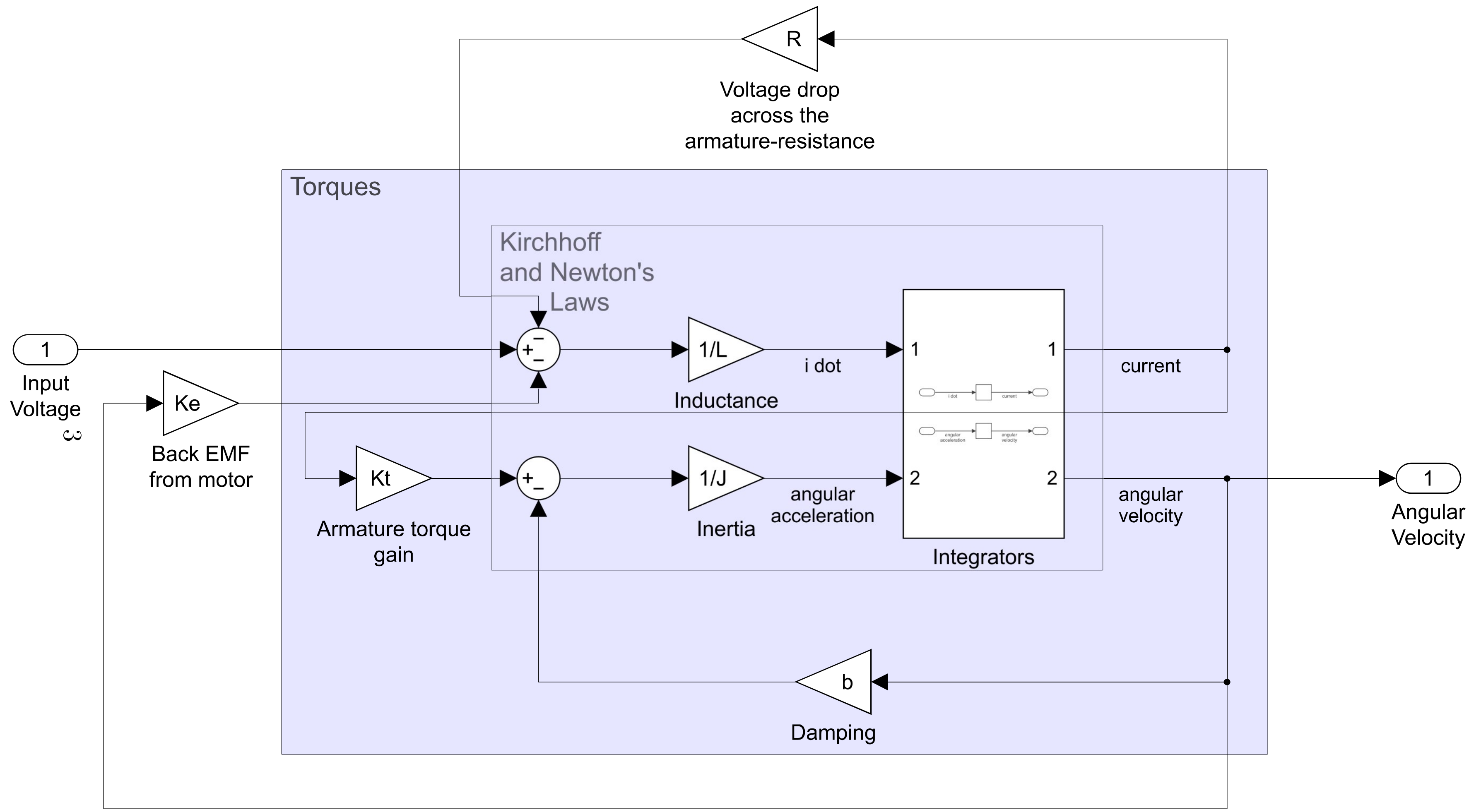
In this lab, we simulate a DC motor according to the system on page 3. The parameters for this are in the script in Appendix subsection A.1.

The motor uses integration in the Laplace domain. Thus, the integrator system used is merely as shown on page 4.

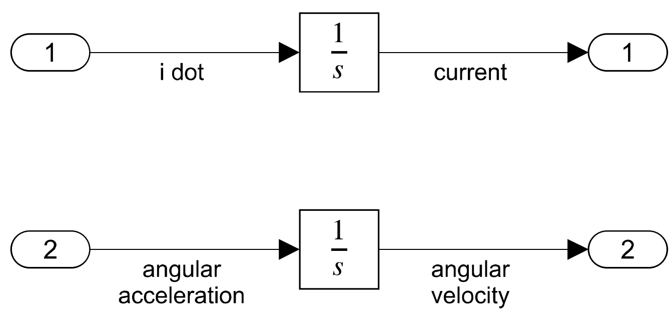
The DC motor is placed in a simple IPO system on page 5 as the professor to analyze the motor. Additionally, we have added a second integrator and a modulus channel to wrap the output around $[0, 2\pi)$. This produces the angular position (or angle in radians).

The motor can be mathematically described using the model

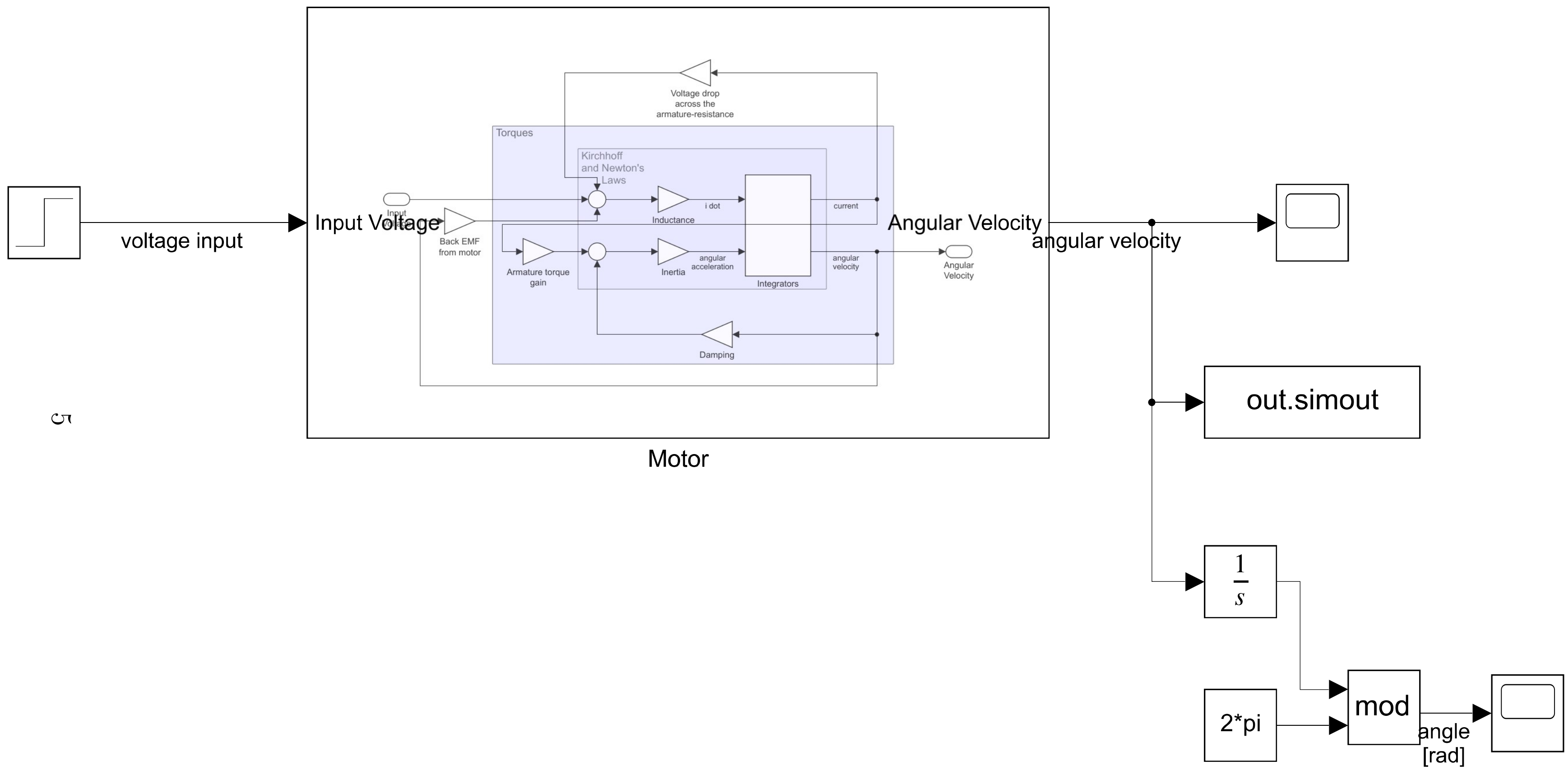
$$\begin{bmatrix} J\ddot{\theta}(t) \\ L\dot{I}(t) \end{bmatrix} = \begin{bmatrix} K_t & 0 & -b \\ -R & 1 & -K_e \end{bmatrix} \begin{bmatrix} I \\ V \\ \dot{\theta}(t) \end{bmatrix}. \tag{1}$$



C:\simulink\ece3413\lab08\lab08_openloop_control_slx.slx


1
i dot
1
s
current
1
2
angular
acceleration
1
s
angular
velocity
2

lab08_openloop_control_slx



2.2 Task 1 – Varying the gains

We plotted the result of modeling the DC motor with the gains given in (2), representing both the armature torque constant and the back EMF from the motor.

$$K \in \Big\{.01, \quad .02, \quad .3, \quad .4, \quad 1, \quad 2\Big\}. \tag{2}$$

We automated this process by using the script in Appendix subsection A.2.

2.3 Task 2 – Plotting the angular velocity response

After this, we see the effect of the default parameters on the motor, using the script in Appendix subsection A.3.

2.4 Tasks 03(a), 4 – Varying the moment of inertia of the motor

We plotted the result of modeling the DC motor with the moments of inertia given in (3).

$$J \in \Big\{.1, \quad 1, \quad 5\Big\}. \tag{3}$$

We automated this process by using the script in Appendix subsection A.4.

2.5 Tasks 03(b), 4 – Varying the damping ratio of the mechanical system

We plotted the result of modeling the DC motor with the damping ratios of the mechanical systems in (4).

$$b \in \Big\{.01, \quad .1, \quad .2\Big\}. \tag{4}$$

We automated this process by using the script in Appendix subsection A.5.

2.6 Task 05 – Parameters to repeat the angle period

Finally, we use the scope for the angular position mentioned in subsection 2.1.

The parameters for this are in the script in Appendix subsection A.6. Rather than the default time of 10.0 s, we use 100.0 s to show the angle period repeating.

3 Results

3.1 Task 1 – Varying the gains

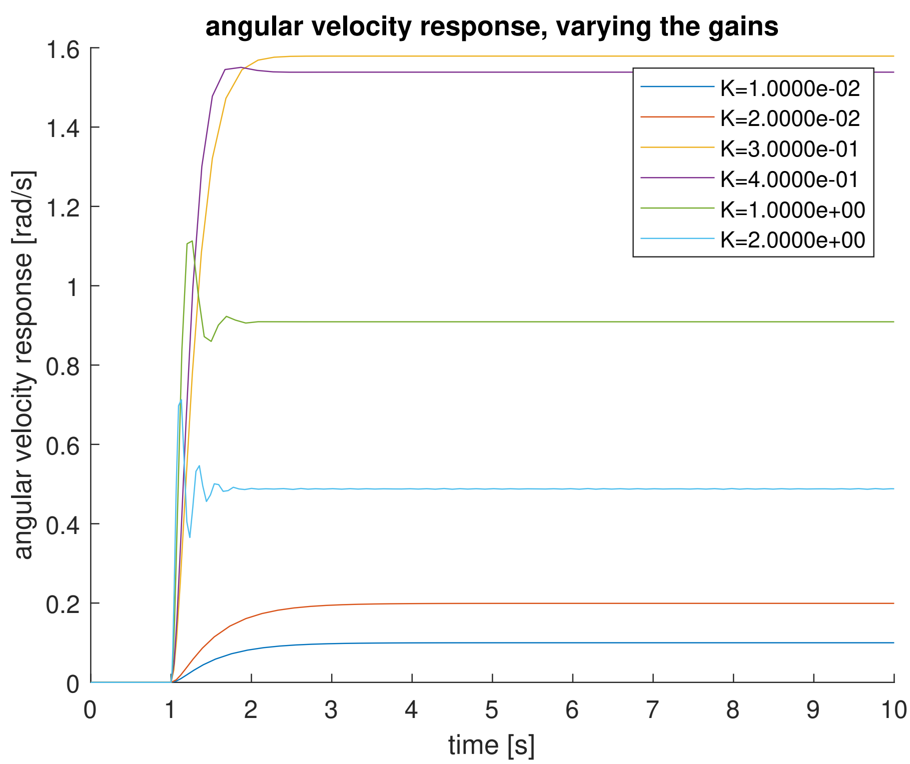


Figure 1: Effect of gains on the angular velocity response.

Here we can see that while $K < .3$, the angular velocity's response seems to to be approaching critically damped from overdamped. The gains make the final value increase. In the interval $K \in].3, .4[$, we see the final value being to decrease and now the angular velocity becomes underdamped. We see the overshoot and the settling time begin to increase with K.

3.2 Task 2 – Plotting the angular velocity response

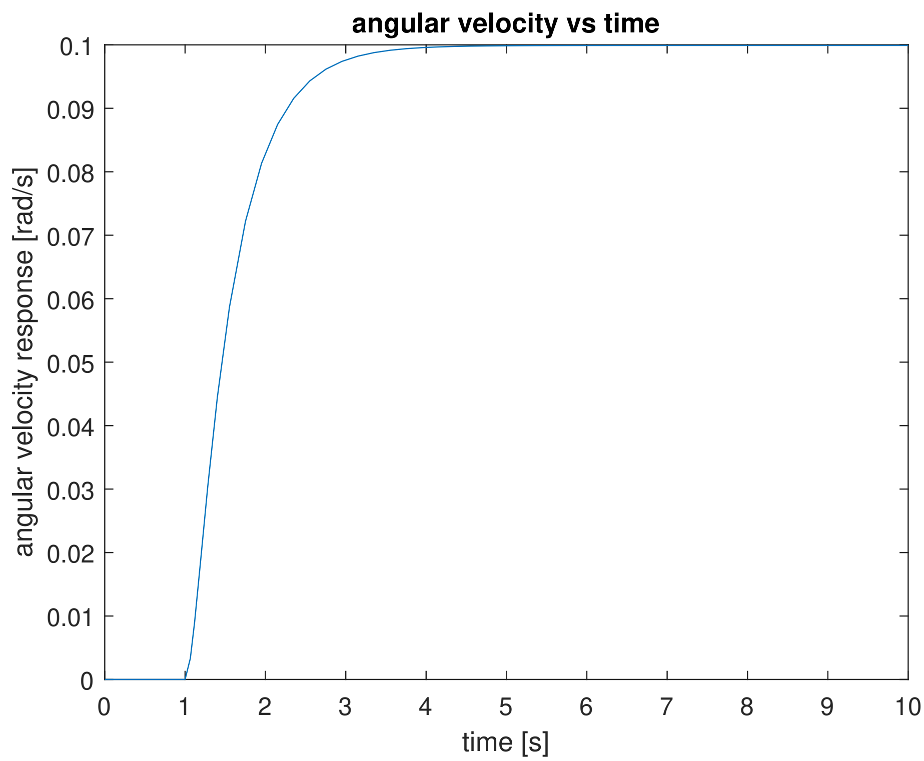


Figure 2: Plot of the angular velocity response.

This is the angular velocity using the default parameters for the DC motor model.

3.3 Tasks 03(a), 4 – Varying the moment of inertia of the motor

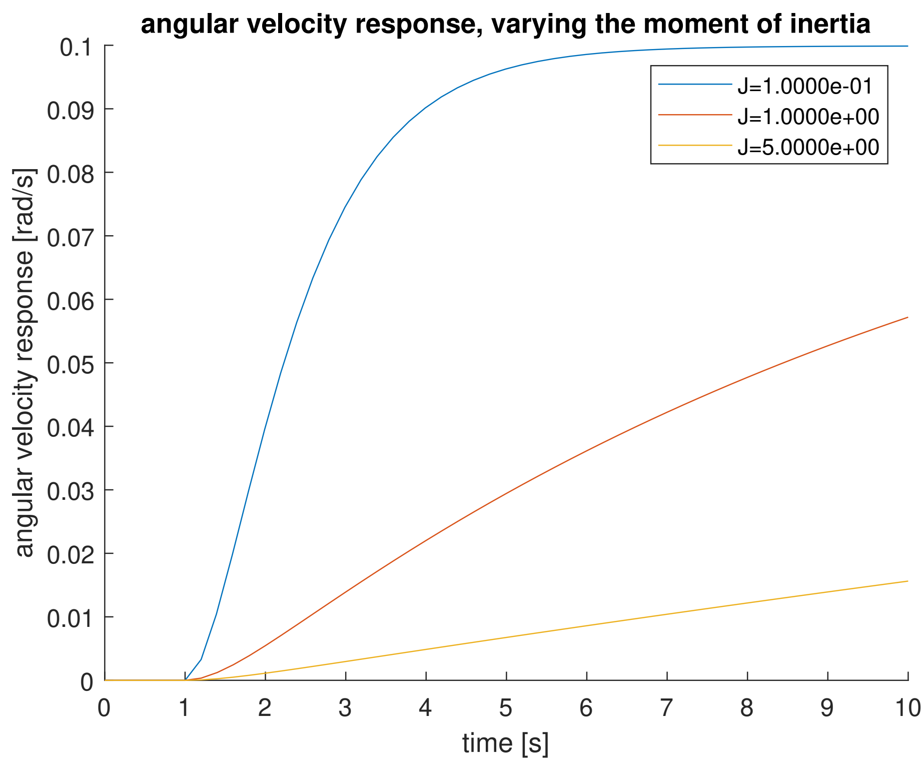


Figure 3: Effect of the moment of inertia on the angular velocity response.

3.4 Tasks 03(b), 4 – Varying the damping ratio of the mechanical system

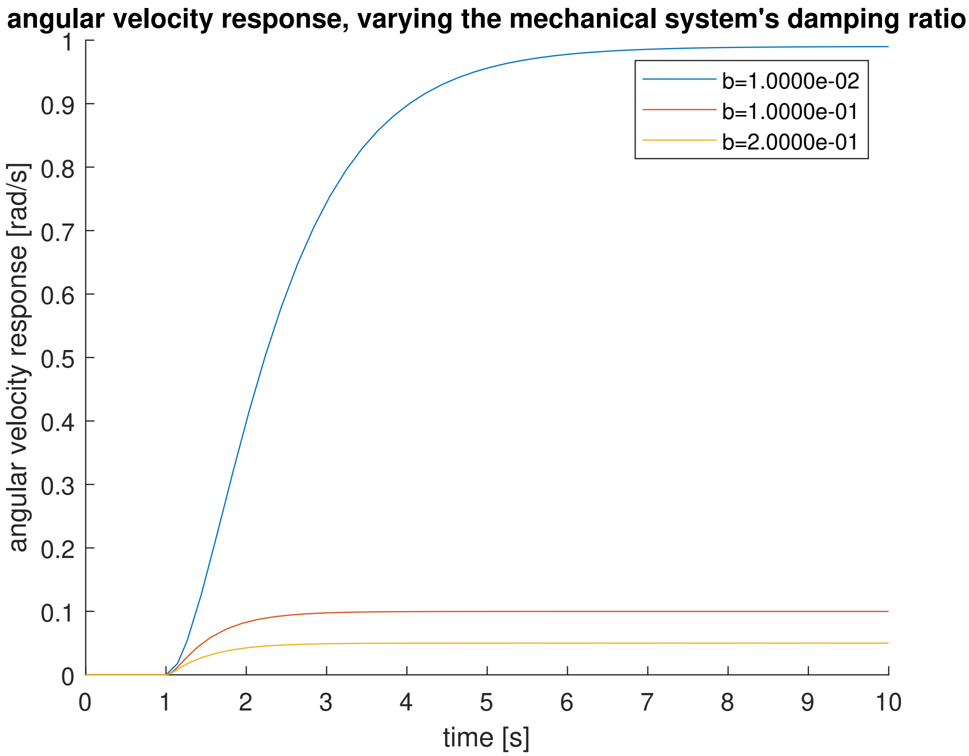


Figure 4: Effect of the damping ratio of the mechanical system on the angular velocity response.

3.5 Task 05 – The angular position vs time

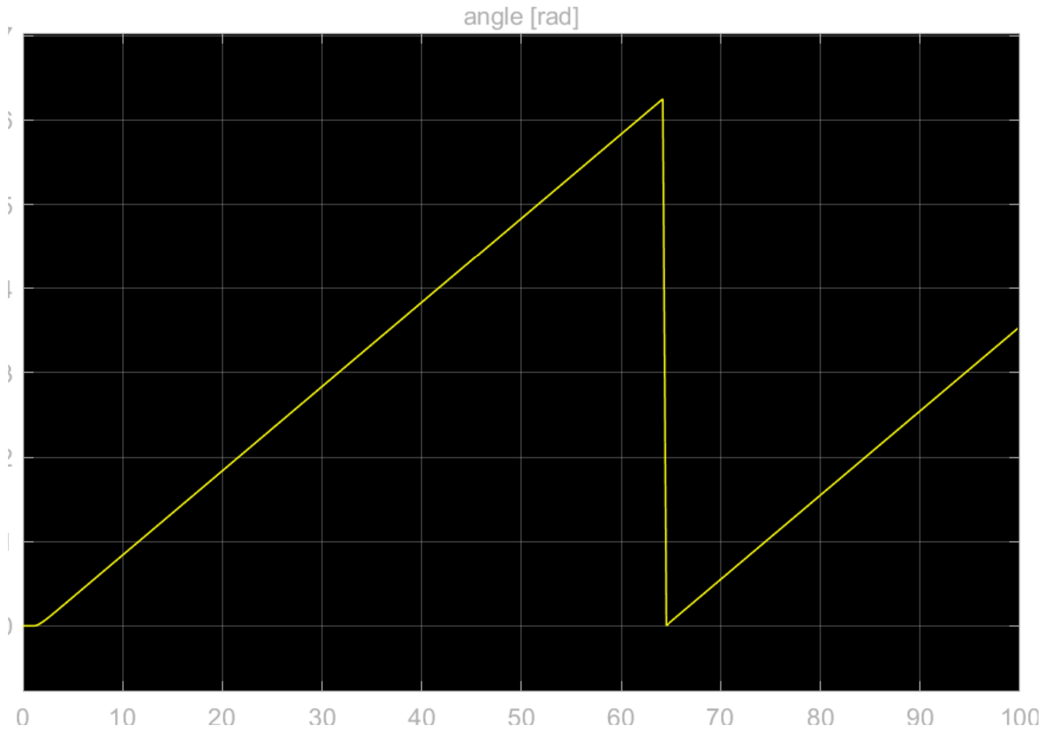


Figure 5: The angular position of the motor.

This is what we expect of the DC motor. We see the angle rising linearly and completing a rotating at $\theta(t) = 2\pi$ while it continues to rise linearly. This tarts as soon as $t = 1\,\mathrm{s}$.

4 Discussion

In this lab, we learned how to model a DC motor with a series closed-loop negative feedback system. We saw that the model behaved as expected rotating linearly and analyzed the effects of gain on the dampness of the function.

Additionally, let's consider the moment of inertia of the motor and the damping ratio of the mechanical system.

In Fig. 3, we see that increasing the moment of inertia decreases the value of the final value of the response while making the system become less stable (forming a ramp). Opposite this, decreasing the moment of inertia makes the final value much larger while increasing the dampness of the system. The value at $J = .1$ appears to be overdamped.

In Fig. 4, the damping ratio of the mechanical system seems to have a similar affect, being inversely proportional to the final value, but without affecting the system's stability.

Additionally, the increase of the final value is more dramatic as the damping ratio decreases, compared to the decrease in final value being more dramatic as the moment of inertia increases.

A Appendix

A.1 Task 00 – Default parameters, Matlab script

```
%% lab08_task00_default_dc_motor_motor_params.m
% Sets the default parameters for the DC motor model.
% By      : Leomar Duran
% When    : 2023-04-12t10:42Q
% For     : ECE 3413 Classical Control Systems
%

clear

% simulation runtime [s]
StopTime = 10.0

% moment of inertia of rotor [kg m^2]
J = 0.01
% damping ratio of mechanical system [N m*s]
b = 0.1
% armature torque gain [N m/A]
Kt = 0.01
% back EMF from motor [V s]
Ke = Kt
% voltage drop across the armature-resistance [ohm]
R = 1
% inductance [H]
L = 0.5
```

A.2 Task 01 – Varying the gains, Matlab script

```
%% lab08_task01_vary_gain.m
% Plots the result of varying the gains : armature torque constant and
% back EMF.
% By      : Leomar Duran
% When    : 2023-04-12t21:50Q
```

```
% For      : ECE 3413 Classical Control Systems
%

% get the default values
clear
lab08_task00_default_dc_motor_motor_params;

K_values = [0.01 0.02 0.3 0.4 1 2]
% for each K value
for K=K_values
    % setting the armature torque constant, back EMF accordingly
    Kt = K
    Ke = K
    % run the simulation
    simout = sim('lab08_openloop_control_slx', StopTime)
    % plot the resulting time series
    hold on
    plot(simout.simout)
    hold off
end % next K
legend(arrayfun((@(x) sprintf("K=%.4e", x)), K_values))
title('angular velocity response, varying the gains')
xlabel('time [s]')
ylabel('angular velocity response [rad/s]')
```

A.3 Task 02 – Plotting the angular velocity response, Matlab script

```
%% lab08_task02_plot_angular_velocity.m
% Plots the angular velocity from the default parameters.
% By       : Leomar Duran
% When     : 2023-04-12t21:58Q
% For      : ECE 3413 Classical Control Systems
%
```

```
% get the default values
clear
lab08_task00_default_dc_motor_motor_params;

% run the simulation
simout = sim('lab08_openloop_control_slx', StopTime)
% plot the resulting time series
plot(simout.simout)
title('angular velocity vs time')
xlabel('time [s]')
ylabel('angular velocity response [rad/s]')
```

A.4 Task 03(a) – Varying the moment of inertia of the motor, Matlab script

```
%% lab08_task03a_vary_motor_moment_of_inertia.m
% Plots the result of varying the gains : armature torque constant and
% back EMF.
% By       : Leomar Duran
% When     : 2023-04-12t21:50Q
% For      : ECE 3413 Classical Control Systems
%

% get the default values
clear
lab08_task00_default_dc_motor_motor_params;

moments_of_inertia = [0.1 1 5]
% for each moment of inertia
for J=moments_of_inertia
    % run the simulation
    simout = sim('lab08_openloop_control_slx', StopTime)
    % plot the resulting time series
    hold on
    plot(simout.simout)
```

```
    hold off
end % next J
legend(arrayfun((@(x) sprintf("J=%.4e", x)), moments_of_inertia))
title('angular velocity response, varying the moment of inertia')
xlabel('time [s]')
ylabel('angular velocity response [rad/s]')
```

A.5 Task 03(b) – Varying the damping ratio of the mechanical system, Matlab script

```
%% lab08_task03b_vary_mechanical_sys_damping.m
% Plots the result of varying the gains : armature torque constant and
% back EMF.
% By      : Leomar Duran
% When    : 2023-04-12t22:03Q
% For     : ECE 3413 Classical Control Systems
%

% get the default values
clear
lab08_task00_default_dc_motor_motor_params;

mechanical_sys_damping = [0.01 0.1 0.2]
% for each moment of inertia
for b=mechanical_sys_damping
    % run the simulation
    simout = sim('lab08_openloop_control_slx', StopTime)
    % plot the resulting time series
    hold on
    plot(simout.simout)
    hold off
end % next b
legend(arrayfun((@(x) sprintf("b=%.4e", x)), mechanical_sys_damping))
title("angular velocity response, varying the mechanical system's damping ratio")
xlabel('time [s]')
ylabel('angular velocity response [rad/s]')
```

A.6 Task 05 – Parameters to repeat the angle period, Matlab script

```
%% lab08_task05_angle_repeat_params.m
% Sets the parameters for the DC motor model so that angle shows to be in
% [0, 2pi[
% By      : Leomar Duran
% When    : 2023-04-12t22:14Q
% For     : ECE 3413 Classical Control Systems
%

% set defaults
clear
lab08_task00_default_dc_motor_motor_params

% simulation runtime [s]
% to show the repetition
StopTime = 100.0
```