LAB 3: IDENTIFICATION OF QNET DC MOTOR

1 Objectives

In this lab, we will identify the transfer function and the parameters of QNET Allied Motion CL40 Series Coreless DC Motor (model 16705). We will use MATLAB to intercat with the DC Motor in a similar manner as we did with the DC Motor simulation of *Lab-2*. We will identify the model parameters of the DC motor using the properties of its time and frequency response.

2 Introduction

2.1 Model of QNET DC motor

The QNET DC MOTOR has the same mathematical model of a the DC Motor presented in *Lab-2*.

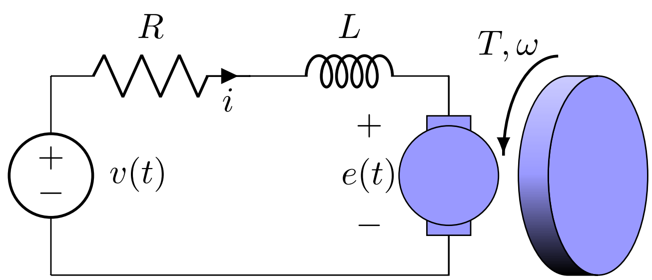

Figure 1 A schematic diagram of a DC motor

According to Newton's second law and Kirchoff's voltage law, we have

$$J\frac{d\omega}{dt} + b\omega = K_t i$$

$$L\frac{di}{dt} + Ri = v - K_e\omega$$

Taking the Laplace transform of the above equations, we get

$$(Js + b)\Omega(s) = K_t I(s)$$

$$(Ls + R)I(s) = V(s) - K_e\Omega(s)$$

Or,

$$H(s) = \frac{\Omega(s)}{V(s)} = \frac{K_t}{(Js + b)(Ls + R) + K_t K_e}$$

From the datasheet of the Qnet DC Motor we know some of those parameters that are shown in Table 1. However, the motor moment of inertia J and motor viscous friction constan b are unkown.

Symbol	Description	Value	Unit
K_t	Torque proportionality constant	0.042	Nm/A
K_e	Back electromotive proportionality constant	0.03	$V/(rad/s)$
L	Electric inductance	1.16	mH
R	Electric resistance	8.4	Ω
J	Motor moment of inertia	Unkown	kgm^2
b	Motor viscous friction constant	Unkown	kgm^2/s

Table 1 Values of the parameters of the Qnet DC Motor

2.2 Accessing the ressource files

We provide a Matlab class which interfaces with the Qnet DC Motor. The following steps explain how to get the necessary Matlab files:

1. Log in to `myCourses` and select the content folder Labs.
2. Download `lab_03_matlab.zip`.
3. Unzip the content in your Matlab workspace.
4. Open Matlab and navigate to the path where you have `Lab_03_SystemIdentification_Hardware.m`.
5. Run `Lab_03_SystemIdentification_Hardware.m`. You shouldn't have any errors.

We also provide a Quick Start Guide that describes how to handle the Qnet DC Motor.

1. Log in to `myCourses` and select the content folder Labs.
2. Download `QNET DC Motor Quick Start Guide.pdf`.

Before proceeding make sure that you have read and followed step 1 to 4 in the "Quick Start Guide: QNET 2.0 DC Motor".

2.3 Interfacing with Qnet DC motor

The interface to the Qnet DC Motor is in `Interface/QnetDCMotor.m`. For the purpose of this laboratory, there is no need to understand the implementation of `QnetDCMotor.m`. To access detailed documentation of `QnetDCMotor`, use:

```
doc QnetDCMotor       % General documentation about QnetDCMotor
```

The script file `Lab_03_SystemIdentification_Hardware.m` contains commented code to guide you through the lab.

1. `QnetDCMotor` opens a serial connection with the National Instrument Data Acquisition board. The code snippet below shows how to initialize `QnetDCMotor`.

```
% Create an object handle for QnetDCMotor with specified parameters
Motor = QnetDCMotor();
```

After a successful connection the QNET DC MOTORstatus led should turn from red to green. This means that motor is on. You can cut the power, aliment the power, or reset the power to the motor by doing:

```
% Set the power to off
Motor.off();
% Set the power to on
Motor.on();
% Reset the power to off than on.
Motor.reset();
```

2. The outputs of a `QnetDCMotor` object are 4 values in a form of arrays:
   - Use `Motor.time()` to retrieve the time history.
   - Use `Motor.current()` to retrieve the history of the armature current `i` in amper.
   - Use `Motor.velocity()` to retrieve the history of the DC motor angular velocity in radian per second.
   - Use `Motor.angle()` to retrieve the history of the DC motor angle in radian.

   When the motor is powered on, you can visualize the outputs in a real-time plot using:

```
% Opens a scope into the motor inputs and output.
Motor.scope();
```

3. The input interface to a `QnetDCMotor` object is the function `drive()`. The `drive()` function takes 3 arguments, the voltage applied to the motor's armature in volts, the time when this voltage is applied in seconds, and the duration of the drive function in seconds.

   In order to send a fixed voltage for a some duration after some delay, you can do:

```
input = 2;     %volts
delay = 1;     % second
duration = 5;  % seconds
Motor.reset(); % reset the motor power
% Drive Motor with the specified voltage and duration starting from
% the specified delay.
Motor.drive(input, delay, duration)
% Stop the motor
Motor.off();
```

In order to send a time-varying signal we can use the `drive` function inside a `for` loop:

```
dt = 0.01;         % Sampling time  for input signal (in seconds)
T  = 5;            % Total duration of simulation (in seconds)
time  = 0:dt:T;    % A vector containing all time samples
Omega = 2*pi*0.1; % input signal frequency (rad/s)
Motor.reset();     % reset the motor power
for t = time
  % Generate a cosine wave input at current time
  u = 2 + cos(Omega*t);

  % Drive motor for a duration of dt at time t
  Motor.drive(u, t, dt);
end
```

4. The `QnetDCMotor` uses a sampling time to interface with the NI-board. At each sampling time, the last availaible voltage data will be sent and new measurements will be received, the sampling time can be seen by typing:

```
Motor
```

A new sampling time can be set by typing:

```
Motor.setSamplingTime(0.01) % Set a new sampling time for QnetDCMotor
```

3 Identification by Frequency Analysis

3.1 Frequency analysis of a linear system

If a cosine wave $u(t) = A_u \cos(\omega t)$ is injected into a linear system at a given frequency, the system will respond at that same frequency ω with a certain magnitude A_u and a certain phase angle relative to the input ϕ. The steady-state system output can be written as $y(t) = A_y \cos(\omega t + \phi)$.

For a given system, it is possible to draw a "point-by-point" Bode plot by injecting a cosine wave with a fixed frequency and measuring the magnitude and phase shift of the output after it reaches its steady-state.

3.1.1 Lissajous method for phase shift

The Lissajous (ellipse) method dates from the tiem of analog oscilloscopes. Most oscilloscopes have a "XY mode" where it is possible to plot one signal in function of another. Using this view with sinusoidal signals, it is possible to accurately measure the phase shift between the two signals.

Let's assume an input signal $u(t) = A_u \cos(\omega t)$ and its output signal $y(t) = A_y \cos(\omega t + \phi)$. From the cosine addition formula, we can write

$$y(t) = A_y \cos(\omega t) \cos(\varphi) - A_y \sin(\omega t) \sin(\varphi)$$

By eliminating ωt between those signals, we get:

$$\frac{y(t)}{A_y} = \frac{u(t)}{A_u} \cos(\phi) \pm \sqrt{1 - \left(\frac{u(t)}{A_u}\right)^2} \sin(\phi)$$

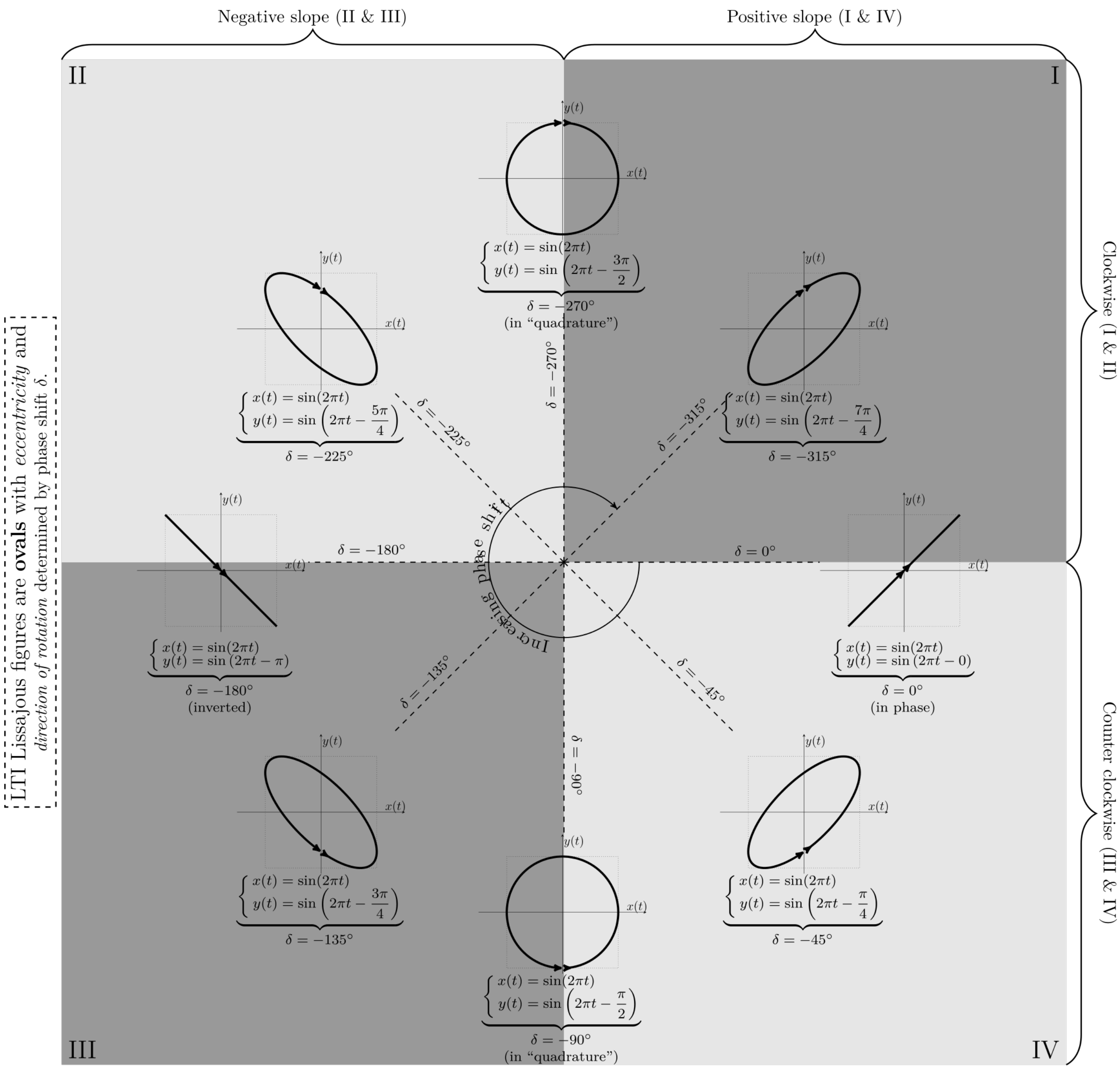

Figure 2 Identifying the phase shift using Lissajous method

Let y^* denote the value of $y(t)$ at the time when $u(t)$ attains it maxima (i.e., $u(t) = A_u$. Then, the shift phase ϕ satisfies:

$$\cos(\phi) = \frac{y^*}{A_y}$$

The fraction $\frac{y^*}{A_y}$ can be measured empirically in the XY plot of $y(t)$ versus $u(t)$.

An illustration of using the Lissajous method (taken from Wikipedia) is shown in Fig 2.

3.2 Question 1

1. Using MATLAB, send a cosine wave to QNET DC MOTOR of magnitude 1, phase 0, an offset of 2 V, and a frequency $f = 0.1$ Hz.

$$u(t) = 2 + \cos(2\pi f t)$$

2. Find the magnitude and phase of the transfer function of the QNET DC MOTOR for the following frequencies:

Frequency (Hz)	A_y	y^*	Magnitude (db)	Phase (deg)
0.05				
0.1				
0.2				
0.4				
0.6				
0.8				
1.0				
2.5				
5.0				

   You may need to adjust the duration of simulation and the sampling time depending of the chosen frequency.

3. Draw the resuling Bode plot (Magnitude (dB) vs Frequency (Hz) on a semilog plot and Phase (deg) vs Frequency (Hz) on a semilog plot).
4. From the Bode plot, what is the system order?
5. From the Bode plot, measure the DC gain.
6. From the Bode plot, measure the cut-off frequency (Frequency at which the magnitude drops by a factor of $-20\log_{10}\left(\frac{1}{\sqrt{2}}\right) = -3$(dB) from the DC gain).

4 Reduced DC Motor Model

The block diagram of the transfer function of the DC Motor is presented in (Figure 3). Notice the presence of 2 inner-loops (elctrical and mechanichal) and one outer-loop.

4.1 Question 2

1. Let $E(s) = V(s) - K_e\Omega(s)$. Identify the transfer function $H_{in}(s) = \frac{I(s)}{E(s)}$ from the block diagram.
2. Using MATLAB draw the one unit step response of H_{in} using the values from Table 1.
3. What is the rise time of H_{in}. Provide the order-of-magnitude difference between the rise time of H_{in} and the rise time of the QNET DC MOTOR. (Use the cut-off frequency obtained from Bode plot)

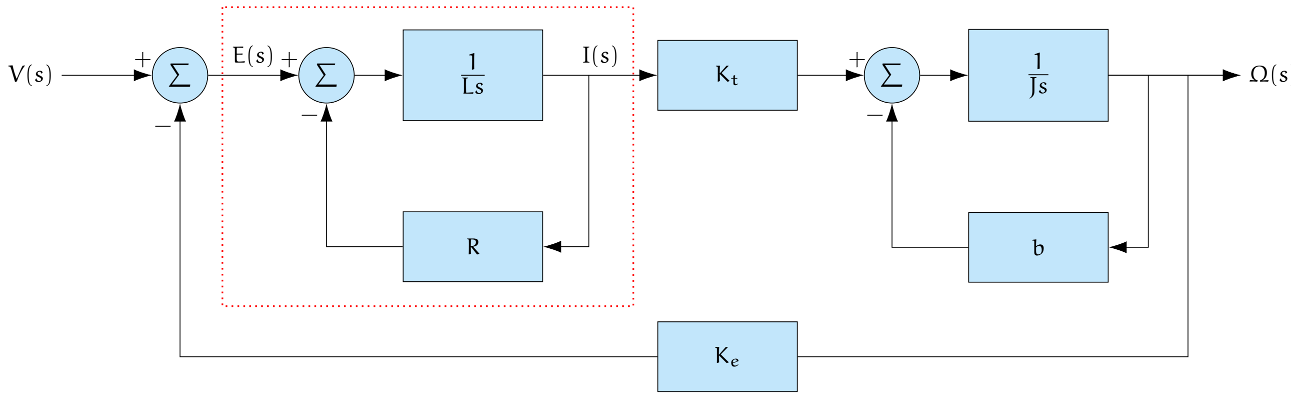

Figure 3 Block diagram of the transfer function of the DC Motor. The red dashed box shows the electrical inner-loop H_{in}

4. Indeed, the inner-loop H_{in} is very fast compared to the observed dynamics of the QNET DC MOTOR. We can replace the inner-loop transfer function by a simple gain that equals its steady-state gain. Draw the reduced block diagram.
5. Find the reduced transfer function.

5 Identification by Step Response Analysis

Now, we know that the QNET DC MOTOR can be reduced to a first system order, we will analyse its step response to determine the values of motor moment of inertia J and the motor viscous friction constant b.

5.1 First-order systems

A generic first-order system is described by a first-order linear differential equation

$$a_1 \frac{dy(t)}{dt} + a_0 y(t) = b_0 u(t)$$

The transfer function of this system is given by

$$H(s) = \frac{Y(s)}{U(s)} = \frac{b_0}{a_1 s + a_0}$$

This transfer function be written in the canonical form as

$$H(s) = \frac{K}{\tau s + 1}$$

- $K = \frac{b_0}{a_0}$ is the `DC gain`, also called `steady state gain`. The `DC gain` represent the amplitude ratio between the steady state step response and the step input.
- $\tau = \frac{a_1}{a_0}$ is the `time constant`. The `time constant` might be regarded as the time for the system's step response to reach $1 - e^{-1} \approx 63.2\%$ of its final value.

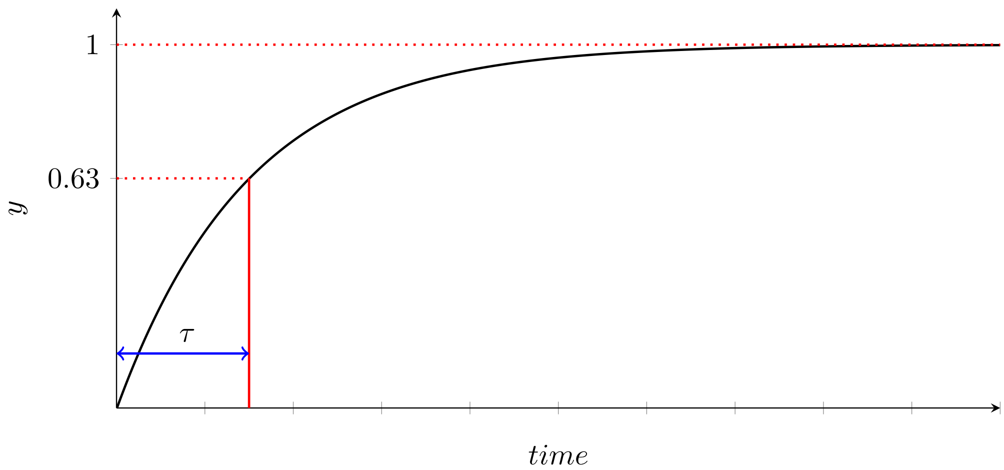

Figure 4 Step response of a first-order system

Figure 3 demonstrates the unit step response of a first-order system with the DC gain $K = 1$ and the time constant $\tau = 1.5$.

In the time domain, the step response $s_1(t)$ of a first-order system with DC gain K and the time constant τ is given by:

$$s_1(t) = K\left(1 - e^{-t/\tau}\right)\mathbb{1}(t)$$

5.2 Question 3

1. Using MATLAB script `Lab_03_SystemIdentification_Hardware.m` simulate a step response of value 2 V for 5 seconds to a `QnetDCMotor`.
2. Using MATLAB, plot in the same graph both the voltage input $u(t)$ and the angular speed of the `Motor` $y(t)$.
3. From the plot, identify the parameter of the transfer function of the QNET DC MOTOR: DC gain and time constant.
4. Using the DC motor mathematical model, provide an approximation of the motor moment of inertia J and the motor viscous friction constant b.

6 Assignement

In a report format, answer the laboratory questions. The report should contain:

- An introduction and a conclusion, outlining the purpose of the laboratory and what you have learned.
- Explanation of the steps to answer the laboratory questions.
- All figures should have a legend and a caption.
- Include your code in the report appendix.

The report due date is: October 23, 2017.