Lab 6: Speed and Angle Control of the Qnet DC Motor

1 Objectives

In this lab, we want to design controllers in order to control the position and angular speed of the Qnet DC Motor. So far we have controlled the Qnet DC Motor with a P controller. In this lab, we observe the output and performance of the system when PI and PD controllers are in use. Precisely, we want to design controllers to control the position and speed of the Qnet DC Motor to track a square wave with the following specifications.

1.1 Specifications

- Zero steady-state error.
- 5%-Settling time of 0.25 s.
- Overshoot less than 10%.

2 Resource files

Similar to previous labs, we provide a Matlab class which interfaces with the Qnet DC Motor.

1. Log in to `myCourses` and select the content folder Labs.
2. Download `lab_06_matlab.zip`.
3. Unzip the content in your Matlab workspace.
4. Open Matlab and navigate to the path where you have `Lab_06_Control.m`.
5. Run `Lab_06_Control.m`.

3 Review From Lab 3

3.1 Qnet DC Motor model

The transfer function between the shaft angular speed $\omega(s)$ as well as its position $\theta(s)$ and the input voltage $V(s)$ in a DC Motor (Figure 1) is correspondingly given by $\bar{H}(s)$ and $\bar{G}(s)$ as follows.

$$\bar{H}(s) = \frac{\omega(s)}{V(s)} = \frac{K_t}{(Js+b)(Ls+R)+K_tK_e}$$

$$\bar{G}_(s) = \frac{\theta(s)}{V(s)} = \frac{1}{s}\frac{K_t}{(Js+b)(Ls+R)+K_tK_e}$$

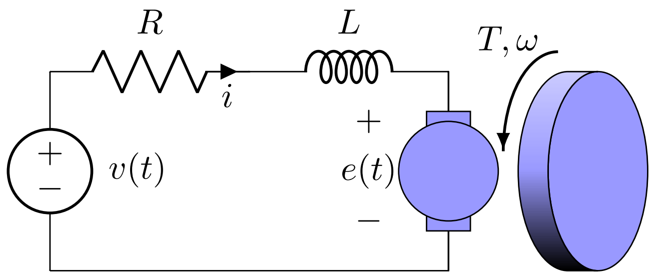

Figure 1 A schematic diagram of a DC motor

In lab-03, we saw that this transfer function can be approximated by a first order transfer function with a DC gain G and time constant τ. Therefore the first order transfer functions are

$$H(s) = \frac{\omega(s)}{V(s)} = \frac{G}{\tau s + 1}$$

$$G(s) = \frac{\theta(s)}{V(s)} = \frac{1}{s}\frac{G}{\tau s + 1}$$

where $G = 28.5$ and $\tau = 0.16$.

4 Position and Speed Controller Design in Matlab

Suppose that we want to design two different controllers in order to command the Qnet DC Motor system to follow reference signals ω_{ref} and θ_{ref} given Figure 2 and in Figure 3 below.

Note: We can not control speed and position of the rotor of a DC motor at the same time. Therefore, we design and implement controllers separately.

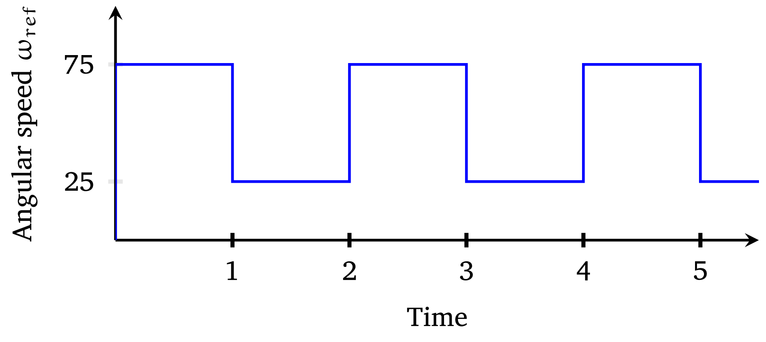

Figure 2 Reference angular speed of the rotor.

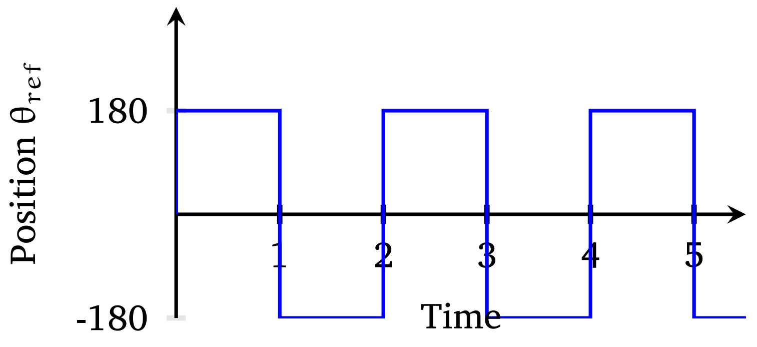

Figure 3 Reference position of the rotor.

Suppose $K_\omega(s)$ controls the speed and $K_\theta(s)$ controls the position. Then, the closed loop controlled system with controllers $K_\omega(s)$ for $H(s)$ and $K_\theta(s)$ for $G(s)$ are given in Figure 4 and Figure 5 respectively.

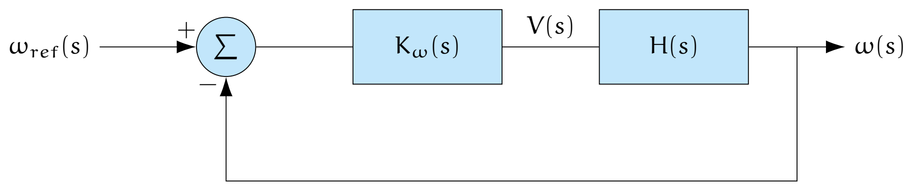

Figure 4 Block diagram of the closed-loop system for speed control.

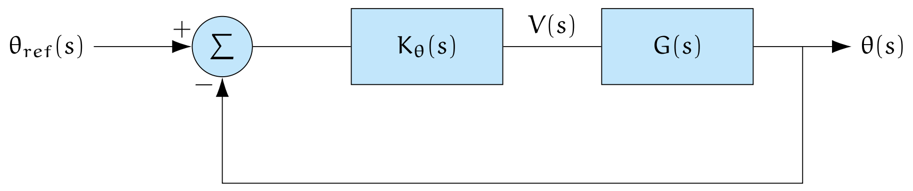

Figure 5 Block diagram of the closed-loop system for position control.

4.1 Question 1

In order to design controllers to control the speed and position of the QNET DC MOTOR follow these steps:

a. Define the transfer functions $H(s)$ and $G(s)$ in MATLAB.

b. Define time interval from 1 to 5 with increments 0.01 and produce the reference signals in Figure 2 and Figure 3.

c. Use MATLAB to find a PI controller for $H(s)$ and a PD controller for $G(s)$ and find the closed loop controlled system using these controllers for each case.

d. Simulate the output of the system and plot it. Plot the output and the input in one figure. Are you getting the performances given in Section 1.1? If not, tune the controller parameters based on lab 4 to meet the specifications.

5 Speed Control in QNET DC MOTOR

In this section, we use the controller in the previous section for the real system. In lab 5, we designed a P controller to control the speed of the shaft of the motor. It was shown that, it is impossible to remove the steady state error even for large values of k_P. Also, increasing the value of k_P leads saturation in the control signal, i.e., the control signal need to go over ± 5 volts but it is impossible because of the hardware. To solve this problem, we add an integrator to the system. This ensures zero steady-state error for step inputs. The PI controller transfer function is as follows.

$$K(s) = k_p + \frac{k_I}{s}$$

Note: We are working with a digital computer which means that every output to the hardware and every input from the hardware is discrete in time. Hence, in order to produce discrete control signals from the PI controller for QNET DC Motor in discrete time T_s, we should implement the integrator in a different form. This implementation sums the previous error values over time.

$$\frac{1}{s}(\omega_{ref}(s) - \omega(s)) \Leftrightarrow \int_0^{NT_s} (\omega_{ref}(t) - \Omega(t))dt = \sum_{k=0}^{N-1} \int_{kT_s}^{(k+1)T_s} (\omega_{ref}(t) - \omega(t))dt$$
$$= \sum_{k=0}^{N-1} (\omega_{ref}(k) - \Omega(k))T_s$$

5.1 Question 2

1. Using the controller gains you found in the previous part for speed control, drive the closed loop controlled output as well as the command signal (output of the controller) for QNET DC Motor . Plot your results superposed with your previously simulated results.

   Hint 1: The solution to this part is similar to the solution of P controller in lab 5.

   Hint 2: In order to implement the integrator in the controller we use sum function in Matlab-sum returns the sum of the elements of its input. To find more about sum type the following in the command window:

```
doc sum
```

   Hence, the implementation of the PI controller is as follows:

```
u_ = Kp*w_error(n) + Ki*sum(w_error)*dt;
% w_error(n) is the error between the reference and the output at time n.
% w_error is the entire error vector.
```

2. What is the performance of this controller compared to the specifications in Section 1.1 (Steady-state error, 5%-Settling time, Overshoot)? Tune the values of k_I and k_P to get the desired performance. Plot and superpose your test results with the one in the previous part.

3. Do we have saturation again?

6 Position Control for QNET DC Motor

Suppose that an arm is connected to the shaft of this motor and our goal is to control this arm. This can be done with the position control of the QNET DC Motor . In this part, we implement the controller we designed in Question 1 to control the position of the shaft of the real QNET DC Motor . The PD controller transfer function is as follows.

$$K(s) = k_p + k_D s$$

Note: Again, because we are working with a digital computer we have to implement the derivative part in a discrete way. In order to use a PD controller for a real system like QNET DC Motor in discrete time with increments dt we approximate the derivative as follows.

$$\frac{1}{dt}(\Delta(t) - \Delta(t - dt))$$

where $\Delta(t) = \theta_{ref}(t) - \theta(t)$. The implementation of the derivative in Matlab is as follows:

```
u_ = Kp*p_error(n) + Kd/dt*(p_error(n)-p_error(n-1));
% p_error(n) is the error between the reference and the output at time n.
```

6.1 Question 3

1. Using the controller gains you found in Question 1 to control a continuous time transfer function in MATLAB , drive the closed loop controlled output position as well as the command signal (output of the controller) for QNET DC MOTOR .

   Hint: The QNET DC MOTOR unit for input position signal of the QNET DC MOTOR is radian. Hence, we have to change the input reference signal from degree to radian. This can be done as follows:

$$\text{radian} = \text{degree} \times \frac{\pi}{180}.$$

2. What is the performance of this controller (Steady-state error, 5%-Settling time, Overshoot). Tune the values of k_D and k_P to get the desired performance. Plot and superpose your test results and simulated results.

   Note: If you want to plot the results in degree you have to change the unit of the output and input to degree again.

6.2 Question 4

1. Suppose that the disk which is installed on the shaft of the QNET DC MOTOR is the wheel of a car that moves only in one direction: forward and backward. We want to design a controller that controls the car to go forward for 0.4 meter and then move backward 0.2 meter. Design a PD controller and plot the output position of the system as well as the control signal of the car.

   Hint 1: The diameter of the disk is 0.05 meter and the circumference of a circle is πd. The circumference of a circle shows the distance it can pass in one cycle (considering that the road is not slippery). Therefore, for a specified distance in meter we can find the number of cycles the rotor has to turn from the following calculation.

$$\text{Number of cycles} = \frac{\text{distance}(m)}{\pi \times 0.05}.$$

   Hint 2: Each cycle is 360° or 2π rad. Therefore, for a specific number of turns we can find the angle in radian for the motor input.

2. In this question, what is the physical interpretation of rise time, settling time, and overshoot?

3. In 1 second and with the gains you found from Question 3, what is the maximum distance that your car can traverse?

7 Assignment

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 assignment is due 7 days after your lab.