2.14/2.140 Analysis and Design of Feedback Control Systems

Laboratory Assignment 6: Servomotor Velocity Control

Assigned: Week of April 7, 2014
Due: Week of April 14, in your lab session

1 Lab overview

In this laboratory, we study velocity and position control of a D.C. motor. Each station is equipped with a D.C. motor, with an integral tachometer to measure angular velocity, a single turn potentiometer to measure angular position, and a power amplifier to drive the motor. First, we develop a model for the motor. This model is then used to design velocity controllers. For the purpose of this lab, controller design is done using continuous time methods, and the Labview-based controller emulates these with discrete time algorithms.

Initial experiments focus on velocity control. A proportional velocity controller is implemented first. Resulting steady-state errors are observed and the effect of increasing gain on steady-state error is seen. Next we implement a PI velocity controller to remove steady-state errors. Problems with integral windup are observed and an anti-windup scheme is implemented.

2 Power Amplifier Connections

Each station is equipped with a D. C. motor and a power amplifier. **It is important that you read these instructions carefully so that you do not destroy the power amplifier!** As shown in Figure 2 the power amplifier provides a voltage gain of 2 for the signal from the DAC. (Verify this.) We will study this power amplifier in more detail in the next lab.

Note: The power amplifiers have an absolute maximum of ± 25 volts. Irreversable damage will occur above this level. The supplies in the lab can provide more than ± 30 volts and thus are capable of destroying the amplifier. Therefore, before shutting off or turning on the supply *always* set it to 0 volts on both sides.

Supply connections: In the following, we refer to the two 0–30 volt supplies available on the Tektronix power supply. The unit also has a 5 volt supply which we do not use in this lab. The supplies in the lab have been furnished with jumpers and connected for series operation by depressing the appropriate push button on the front panel of the supply. Please keep the supply in this configuration throughout the laboratory. In this mode, the right-hand voltage control sets the voltage output for both supplies. **Never set the power supply to more than ± 17 V.** As connected, the right-hand supply provides $+V_{cc}$ at its positive terminal, and the left-hand supply provides $-V_{cc}$ at its negative terminal. Common is taken from the jumpered center connections (the negative terminal of the right-hand supply and/or the positive terminal of the left-hand supply).

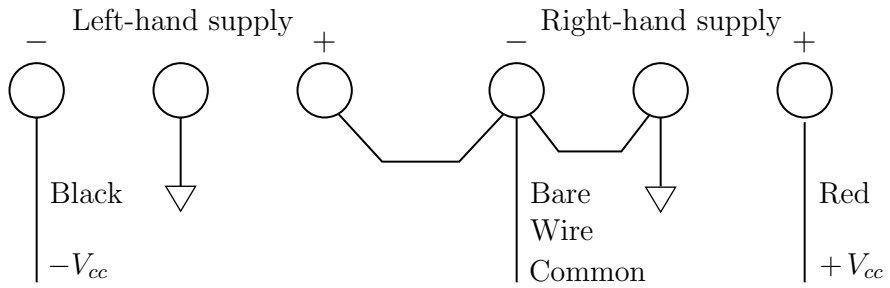

Figure 1: Amplifier/supply connections

Set the current limits on both supplies to the maximum. Whenever you are about to turn on the supply, first adjust both voltage controls to zero. Set the right-hand meter to monitor voltage. Set the left-hand meter to measure current. Then power up the supply. At this point you may increase the supply voltage up to the desired ±17 volt level. (The reason for choosing this voltage is that it is comfortably below the amplifier's absolute maximum rating of ±25 volts.) The amplifier should be wired to the supply as shown in Fig. 1.

3 D.C. Motor model

Each station has a D.C. motor and a power amplifier. Please follow the above precautions before powering up your setup. The D.C. motor has a tachometer to measure velocity and a potentiometer to measure angular position. The motor parameters are :

$$\begin{aligned}
R &= 7.5\ \Omega \\
L &= 5.55\ \text{mH} \\
\Rightarrow \tau_e &= \frac{L}{R} = 0.7\ \text{ms} \\
K_T &= 0.024\ \text{Nm/A} \\
J &= 1.5 \times 10^{-5}\ \text{Kgm}^2\ \text{(all loads referred to motor shaft)} \\
K_{\text{tach}} &\approx 0.023\ \text{V/rad/sec} \\
K_{\text{pot}} &= \frac{V_{pp}}{2\pi \cdot (\text{gear ratio})} \approx 0.5\ \text{V/rad (of motor)} \\
\text{No. of teeth on pinion} &= 36 \\
\text{No. of teeth on driven gear} &= 216 \\
\text{Gearing ratio, N} &= 6 \\
\Rightarrow \tau_m &= \frac{JR}{K_T^2} \approx 200\ \text{ms} \\
\Rightarrow \frac{\omega_m(s)}{V_m(s)} &= \frac{1}{K_T(\tau_m s + 1)} \qquad (1)
\end{aligned}$$

In this derivation, we are assuming that the motor inductance is low enough to be ignored, i.e., $L \simeq 0$. How can this assumption be justified? Please show how the simplified transfer function above

results from this assumption.

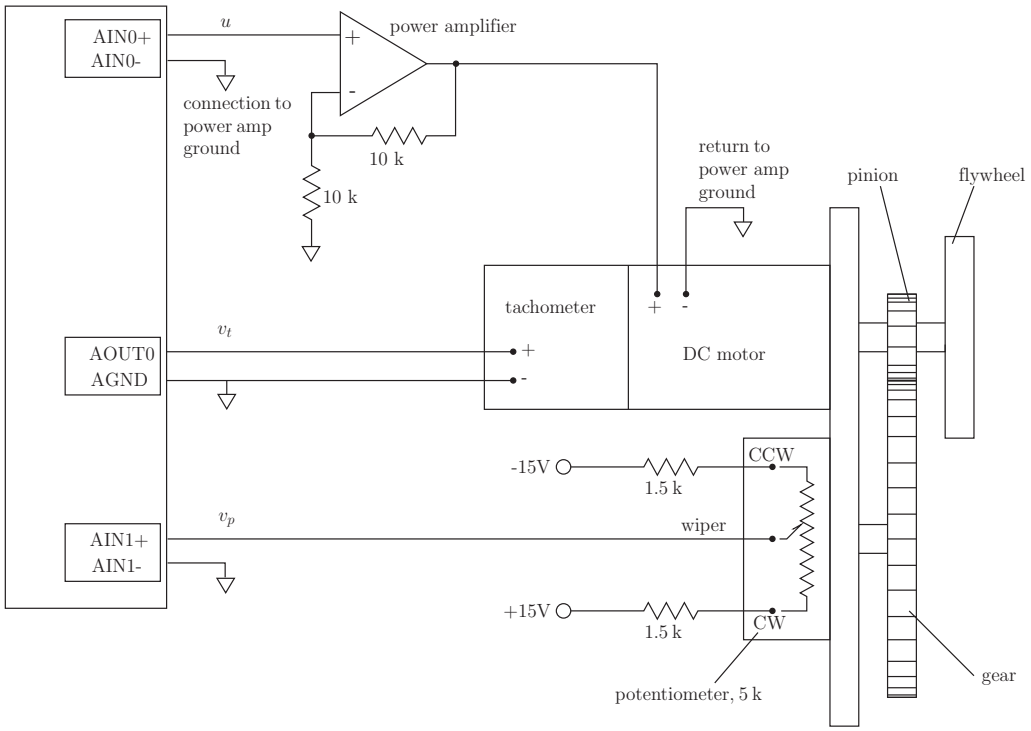

Figure 2: Connections for the DC motor setup.

The connection of the motor setup is shown in Figure 2. For the purposes of this lab, the reference signal r is generated within the Labview-based controller. As shown, we have connected the tachometer signal to AIN0, and the potentiometer signal to AIN1. These are differential inputs, but for this lab the negative input is connected to analog ground. We have also connected the analog output of AOUT0 to the input of the power amplifer as the command voltage.

In order to match the A/D input range, we connect the potentiometer to operate at a little less than ± 10 volts (± 9.4 V). To do this, we use the ± 15 volt power source from the protoboard, and use a pair of resistors to reduce the voltage to the desired value. The value of the potentiometer resistance is 5 kΩ, thus the pair of voltage dropping resistors are chosen as 1.5 kΩ. These resistors are soldered in place on the motor setup, within insulating heat-shrink tubing.

a) Draw a block diagram model which represents the power amplifier, motor, tachometer, and potentiometer using the parameters given above. Throughout this lab, we will drive the motor with the power amp configured for a voltage gain of 2 as shown in the figure.

b) Derive the transfer functions of the system from input u to outputs v_t and v_p. That is, calculate the transfer functions $v_t(s)/u(s)$ and $v_p(s)/u(s)$. First calculate these in terms of the variables defined above, then substitute in the numerical values. Use Matlab to plot the frequency response for these transfer functions in Bode form.

4 Velocity control

We now use the motor model developed in the previous section for designing velocity control of the D.C. motor. The tachometer output is used as the feedback signal. In this and subsequent sections of the lab, we use a sampling time of 1 msec for the implementation on the real-time hardware. This is fast enough that the effects of sampling are negligible.

a) Design a proportional velocity controller of the form $G_1(s) = K_p$, which realizes a *closed-loop* system time constant of $\tau = 20$ ms. (Set $K_i = 0$ in this section to disable integral control. Implement this controller on the real-time hardware. Collect step response data and compare with the predicted step response. What mechanical impedance (i.e., what equivalent spring/mass/dashpot) is created at the output shaft? Why? (You can sense this impedance most easily at the potentiometer shaft, since the gearing makes this the higher stiffness point in the drivetrain.)

b) What happens as you increase the gain K_p above its nominal design value? What limits the achievable bandwidth? What physical mechanisms might explain this?

(Note: In the lab you will see that the tachometer voltage has significant ripple when the motor is spinning. This is due to the brushed construction of the tachometer.)

4.1 PI velocity control

Next, we implement PI control which removes the DC errors that were present with a proportional controller. Use a PI controller of the form

$$G_2(s) = K_p \left(1 + \frac{K_i}{s}\right)$$

The proportional gain K_p and the integral gain K_i are chosen to meet specifications on the response.

The presence of an integrator in your control law can lead to *integrator windup*. This can be noticed if you cause the error to accumulate in spite of saturation of the actuators. The integrating action of the controller keeps accumulating the error and leads to large overshoots. You can observe this by setting a constant velocity command, and by imposing a velocity error on the flywheel by clamping it with your hand. Be careful not to get yourself hurt while doing this (*i.e., Keep your fingers out of the gear mesh*). When you release the flywheel, you should be able to see a rather large overshoot. For the purpose of observing windup, use the DC offset of the signal generator to set the velocity to an essentially constant level. Note that the size of the overshoot is dependent on the time for which you hold the flywheel stationary.

It is also interesting to conduct this experiment with the velocity command set to zero. Here you can directly feel the response torque as you move the motor. Keep in mind that the feedback is on velocity, which is the derivative of position.

a) Design a PI velocity controller which nominally creates closed-loop roots with $\zeta = 0.5$ and $\omega_n = 30$. Explain how you calculated the necessary values of K_p and K_i. What is the

resulting crossover frequency and phase margin for this design? Include a Bode plot for the return ratio in your lab report, showing the predicted crossover frequency and phase margin.

Implement this controller on the real-time hardware. Note that a saturation limit of ± 10 is required to limit the DAC output voltage to ± 10 V, which is near the saturation value.

For a non-zero DC velocity reference, and with the integrator saturation limits set to infinity, clamp the flywheel with your hand for some period of time, and notice the large overshoots when you release it under a constant velocity reference. Modify the integrator saturation limits to prevent windup and re-run the program. Again, impose velocity errors by holding the flywheel with your hand. Notice the greatly improved recovery performance when the integral term is bounded. Explain the difference in the response. What is the 'best' value for the saturation limit? Why?

b) What mechanical impedance (i.e., equivalent spring/mass/dashpot) is created at the output shaft under PI velocity control? Why? How does this impedance change when anti-windup is implemented? Why? This impedance is best sensed with the velocity command set to zero.

2.14 / 2.140 Analysis and Design of Feedback Control Systems

Spring 2014