LAB #8: SPEED CONTROL OF A D.C. MOTOR – THE MOTOR DRIVER

INTRODUCTION

This lab continues the main part of the lab project, which is the development of a speed controller for a direct current (dc) motor. The controller is to operate the motor at one of ten speeds, as selected by the previously-implemented keypad (or possibly four switches). The speed is to remain constant at the selected value as the load on the motor is varied.

The project will be implemented over several weeks, and has the following parts:

1. PWM waveform generation (last week).
2. Hardware and software to drive the motor (this week).
3. Hardware and software to measure the speed of the motor (next two weeks).
4. Measurement of the motor characteristics.
5. A control program to control the speed of the motor.
6. Demonstration and presentation of results (final week).

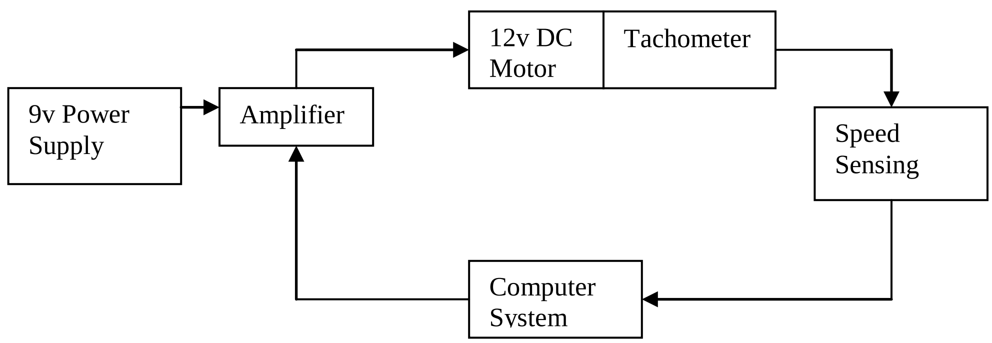

Figure 1. Closed-loop motor control system.

Figure 2. Buehler permanent magnet DC motor with tachometer

MOTOR DRIVER

Under a constant mechanical load, the speed of a permanent-magnet dc motor is determined by the voltage applied to the motor windings. Therefore, a key step in developing a motor controller is to design and implement hardware and software that will allow the microcontroller to control the voltage applied to the motor. The motor to be used in this project, pictured in Figure 2, is a Buehler 12 volt, permanent-magnet dc motor with tachometer (these were used in floppy disk drives). Under constant load, the speed of a dc motor is proportional to the voltage applied to its windings, varying linearly between 0 volts (stopped) and 12 volts (maximum speed for the given load). Thus, the computer must produce a variable control voltage within this range.

Available on the Digilent Electronics Explorer board is a variable dc power supply whose output is adjustable up to 9 Vdc. To produce the required control voltage, the computer will alternately connect and disconnect the power supply to the motor every T seconds, creating a pulse-width modulated (PWM) signal whose value is 9 volts for $T1$ seconds and 0 volts for $T2$ seconds of every period ($T = T1 + T2$). The switching occurs periodically, which means the PWM signal can be analyzed using Fourier series. Fourier series analysis reveals that the PWM signal has a dc term, a fundamental ac component having frequency $1/T$, and higher harmonics that are integer multiples of the frequency $1/T$. The value of the dc term is proportional to the duty cycle

$$D = \frac{T_1}{T_1 + T_2} \ .$$

Due to winding inductance and mechanical inertia, the motor responds best to low frequency signals (the motor dynamics are said to be "low-pass"). The ac fundamental and higher harmonics are dissipated as heat (“filtered out”). In practice, we choose the switching frequency ($1/T$) higher than the natural bandwidth of the motor. **We will experiment with several possible PWM signal frequencies, over a range of values (10Hz to 1000Hz), and select the one that produces the most linear increase in motor speed with change in duty cycle.**

In summary, the motor effectively “sees” the average applied voltage, which is $D \times 9$ volts, where $T1/T$ is the duty cycle of the PWM signal. This average voltage can be changed by varying the duty cycle of the PWM signal, i.e. by varying $T1$ and $T2$ while holding the period T constant. The timing for the PWM signal is to be created with one of the methods (your choice) from the previous lab.

SINGLE SWITCH MOTOR DRIVE

The amplitude of the PWM signal produced by the computer alternates between logic high (5 volts) and low (0 volts) and can supply about 1 ma of current (i.e. 5 mW power). The motor requires a control voltage in the range 0-12 volts, and can draw an amp or more of current, so the power required could be well over a watt. Therefore it is necessary to “amplify” the computer-generated PWM signal. The PWM signal from the microcontroller output will be used to control a transistor that functions as a switch. The 9 volt output of the adjustable power supply is to be connected to the motor, which in turn is connected in series to the transistor switch, as illustrated in Fig. 3. When the switch is ON ("input" is high), the circuit is closed, and current can flow from the power supply through the motor (modeled here as a resistor-inductor combination). When the switch is OFF ("input" is low), the circuit is open. The PWM signal generated by the computer alternately enables and disables the transistor switch, in effect creating a PWM signal at the collector terminal of the switch. The duty cycle of this PWM signal should produce the desired average voltage, in the range 0 to 9 volts.

The transistor rating must be such that it can sustain the maximum load current in the ON state (study the maximum current rating), while withstanding the full power supply voltage in the OFF state (study the maximum collector-emitter voltage rating). Past experience shows that the 2N3904 NPN transistor is marginally suited (the current rating is a bit too low, especially if the motor load is high, causing the load current to increase). The slightly more expensive 2N2222 is slightly better suited.

The diode *D* in Fig. 3 is called an antiparallel or "freewheeling" diode, and it protects the transistor during the "turn OFF" transient. The current loop in the figure occurs during the "turn OFF" transient, allowing the current to dissipate in the motor. When the transistor switch is in the ON state, the diode *D* is not conducting and the current loop flows through the load and transistor. Past experience shows that the 1N4001 rectifier diode is a marginally acceptable diode to use with the given motor and transistor. The 1N4148 switching diode reacts more quickly than the 1N4001, but may be underrated for the current.

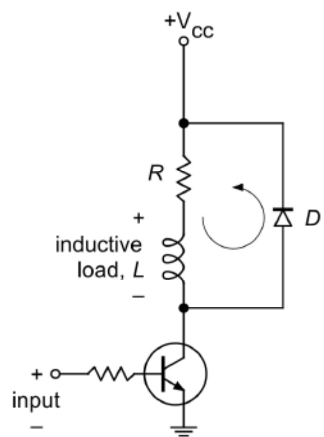

Fig 3. Motor drive circuit using a single transistor. Input is driven by the microcontroller.

The unlabeled resistor that is in series with the base of the transistor must be designed to limit the current drawn from the microcontroller output. The resistance must not be so large, however, that the transistor is not operated in a saturated state when turned ON. To design the resistance value, the following must be known:

- maximum collector current (load current, I_{load})
- bipolar transistor gain (β or h_{FE}) and base-to-emitter turn on voltage ($V_{BE(on)}$, approximately 0.8V under saturation)
- microcontroller output voltage (V_{high})

For the transistor to operate in a saturated state, the base current must be significantly larger than the ratio of the load current divided by the transistor gain

$$I_B >> \frac{I_{load}}{h_{FE}} .$$

The base current is limited by the series resistor between the microcontroller output and the transistor base. The resistance value can be calculated as

$$\text{series resistance} = \frac{V_{high} - V_{BE(on)}}{I_B} .$$

(In modern power electronics, the MOSFET is also widely used for PWM applications. Design equations using a MOSFET for the saturating switch can be found in textbooks.)

HARDWARE AND SOFTWARE DESIGN

The hardware for this lab comprises the motor, the positive power supply on the Digilent EEboard, a transistor, a diode, and the previously-designed timer and I/O ports. The outputs of the EEboard positive and negative power supplies are labeled VP+ and VP-, respectively, on the power block at the bottom of the board. These values are adjustable from 0V to ±9 V, with the current limit adjustable from 0 to 1.5 A. The VP+ output should be used to drive the motor, configured for 9 V with current limit 1.5 A. These values are selected in the *Waveforms Power Supplies and Voltmeters* instrument, shown in Figure 4. Note that "VP+ ON" must also be checked, in addition to "Vcc ON".

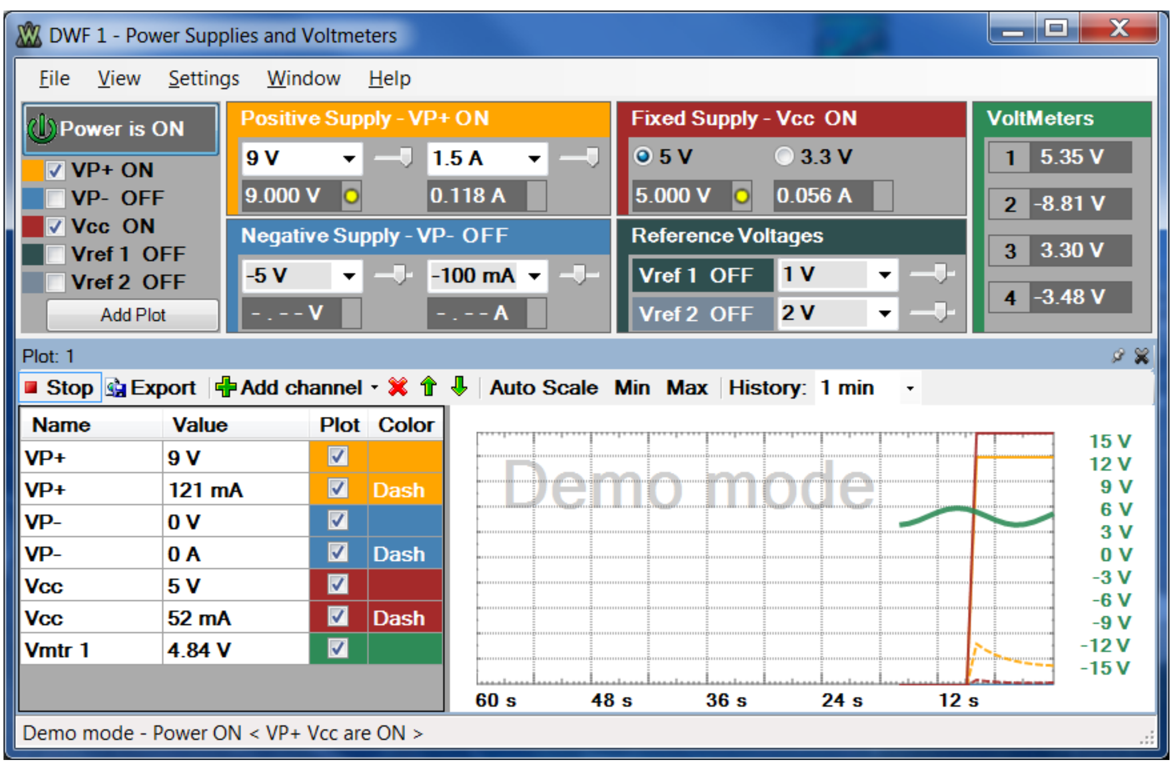

Figure 4. *Waveforms* window, showing Positive Supply VP+ settings and VP+ ON.

Prior to lab, each team should obtain a transistor, series resistor, and diode from the Scientific Supply Store and have a hardware design ready to wire and a test program in a disk file ready to compile at the start of lab, so that lab time can be used for construction and testing.

The C program from the previous Lab should be modified as necessary to produce the PWM signal to control the motor. The period *T* of the PWM signal should be constant, with a variable duty cycle. This program should start the motor at a low speed.

LABORATORY EXPERIMENTS

1. Beginning this week, you will need to design your own experiments to test your hardware and software. Be sure to document each experiment in your lab notebook, and summarize the most significant ones in your report.

2. Construct and test circuits in steps, ensuring that each part works properly before adding the next part. ***Double and triple-check all power supply connections, to prevent damage to the EEboard and components.*** Verify that you can generate the desired PWM signal before connecting it to the transistor switch. Verify that the switch output is the correct signal with a dummy load (resistor) before connecting the actual motor.

3. The final experiment should be to measure the speed of the motor for the different keypad-selectable PWM signal duty cycles. Then plot speed vs. duty cycle. The ideal relationship is linear, but your results may vary. **Motor speed may measured by using the oscilloscope to measure either the amplitude or the frequency of the tachometer ac signal.**

4. Repeat step 3 several times, each time using a different PWM signal frequency over a range of values between 10HZ and 1000Hz. Determine which frequency gives the best performance, i.e. produces the most linear increase in motor speed with change in duty cycle.

Deliverables:
Lab notebooks are to be submitted to the lab instructor this week.

INFORMATION FOR FUTURE LABORATORY REPORTS

1. Briefly describe the circuit (but not “wire by wire”) and the test program (attach a circuit diagram and C program source listing).

2. Discuss your results, including a table and a plot of motor speed vs. PWM signal duty cycle. Compare experimental and theoretical results.