

ELECTROMECHANICAL SYSTEMS

PREREQUISITES: MODULE 10: MICROCONTROLLERS II; MODULE 14: DISCRETE COMPONENTS. MODULE 13 (SENSORS) WOULD ALSO BE HELPFUL.

OUTLINE OF MODULE 17:

What you will learn about in this Module:

- Solenoids
- Peltier devices (for heating and cooling)
- Step motors and controllers
- DC motors: standard and brushless
- Motor controllers and drivers (use of the LMD18245 DC Motor Controller)
(option: use SGS Thompson p/n L298 H-bridge; Mauser)
- Open and closed loop control of motion, temperature, and force

What you will build in the lab:

You will program a motor driver to control the direction, braking, and torque of a small DC brush motor, and/or then program a different driver to control the direction and speed of a stepper motor. The necessary driver board circuits have already been built and are available for use in the lab (you will need to check them out from the instructor), but if you want to, all of the resources necessary to build your own board are also available.

INTRODUCTION:

Safety is the most important consideration in any design. In this Module we will be using low-power devices (Op-Amps and Microcontrollers) to control much higher power devices, such as motors and solenoids. These motors and solenoids can be used to power or control even more dangerous devices, such as hydraulic or pneumatic valves, saw blades, and complex mechanisms. In the interests of safety, this Module will only include low-power motors and solenoids in the Laboratory Exercises. If you are interested, there are several manufacturers of ICs that specialize in high-power electromechanical interfaces and controls (for example Apex Microtechnology: they make op-amps powerful enough to be used as welders). These are generally for industrial or military applications. The basic principles of all of these will be demonstrated using low-power techniques in the lab.

READINGS FROM HOROWITZ AND HILL (H&H): *ART OF ELECTRONICS*

1.31, 2.15-2.16, 3.14-3.15 (MOSFETs)

Good old reliable H&H are almost entirely mute on the subject of motor control, so we will have to dig out the information from elsewhere.

ADDITIONAL READINGS & INTERNET RESEARCH:

You will need to locate two datasheets on the web:

L298N: Dual full Bridge (Motor) Driver, from ST Microelectronics
(<http://eu.st.com/stonline/products/index.htm>)

LMD18245T: 3 Amp, 55 V DMOS Full Bridge Motor Driver, from National Semiconductor (www.national.com)

The L298N is what is known as a PWM driver (Pulse Width Modulation driver), the LMD18245T is a current amplifier (chopper). These work quite differently.

We will discuss both, but you will use the LMD18245T in the lab because we have a pre-built experiment board for this device with a microcontroller already on-board. If you have a compelling reason to build your own circuit or to use the L298N instead, this is OK, just check with the instructor.

To understand how step motors function, you should also locate and read: "Stepper Motor Basics" (<http://www.ams2000.com/stepping101.html>)

Do an internet search to find information on how to use (1) solenoids, and (2) Peltier devices. You can use the circuits in this module to control both of these devices.

MOTOR AND SOLENOID CONTROL:

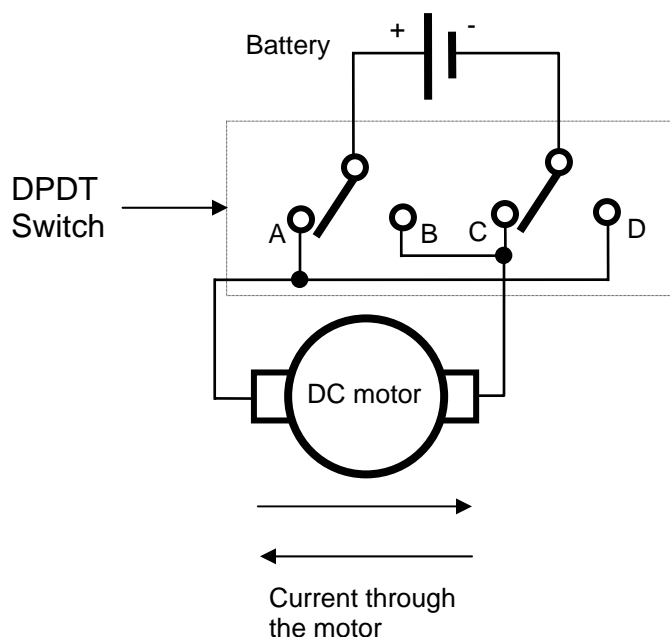
DC motors, step motors, and solenoids are all similar in the sense that they are *inductive* loads that are difficult or impossible to drive directly using the low power analog and digital ICs that are common in most devices (and common in this Laboratory). A *solenoid* is just a coil of wire surrounding an iron core. The core can generally slide in and out of the coil along the axis. When a current is applied, the solenoid generates a force that tends to draw the iron core toward the center of the solenoid. This is the stuff of introductory physics texts, so if you are interested in the underlying physics, that's where you'll have to go for more information. From a practical standpoint, you can cleverly arrange the coil and the iron core such that when current is applied to the electric coil, a mechanical rod is either pushed away from or pulled toward the solenoid. Thus, solenoids are a simple *linear actuator*: they produce force and motion along a straight line, usually for very short distances, typically less than 1 cm. In general, you either turn them ON or OFF to make something happen. They are so useful that you can find them all over the place: electric door locks on cars, water valves for washing machines and sprinkler systems, pinball games, just about anywhere where you need to make a binary mechanical movement. Thus, controlling a solenoid is simply a matter of turning a relatively large electric current ON or OFF.

Electric motors, on the other hand, have much more sophisticated and interesting control aspects. *Stepper motors* typically require at least two sets of pulses feeding two coils within the motor. The sequence in which the coils are energized determines the motor direction. The rate at which the pulses are applied determines

the motor speed (steps per second). Stepper motors are somewhat difficult to control unless you have a microprocessor handy, so we will discuss those later. You can find out more about stepper motors by reading the suggested primer on the web (see above). DC Brush motors are the simplest and most common motors on the planet. You will find these motors everywhere from slot cars (and many other toys), to automobile components, to battery-powered appliances (e.g. screw drivers, shavers). The control of DC motors basically boils down to two issues:

- Control of **direction** (make the motor spin one direction or the other)
- Control of **current** (depending on the type of load, this relates to speed, etc.)

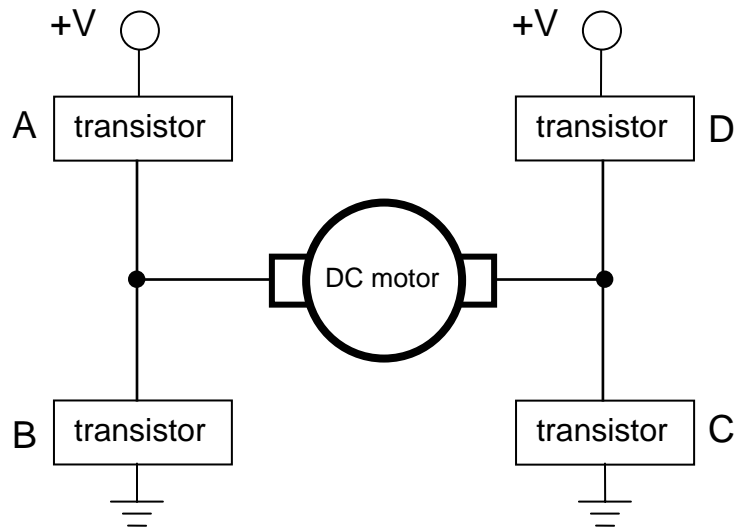
Control of direction is easy: you arrange switches to route the current either of two ways through the motor. This can be accomplished with what is known as a *double pole – double throw* switch. This type of switch has two separate conducting paths (*double pole*), and two possible output positions for each path (*double throw*). A double pole – double throw switch is usually called DPDT for short. It can be wired as shown:



The “+” and “-” terminals of the battery connect to the two conducting paths of the DPDT switch (the “common” terminals on the switch). The two conducting paths in the DPDT switch are mechanically coupled, so that the switch can be in one of two positions: connected to “A” and “C”, or, alternatively, connected to “B” and “D”. The switch is shown in the “A” and “C” position. With the switch in this position, conventional electric current will flow from the “+” terminal of the battery, through the DPDT switch to the “A” terminal, then through the motor from left to right (top arrow), then to the “C” terminal on the switch, then back to the “-” terminal of the battery, completing the electric circuit. Flip the switch to the opposite position, the current

will flow in the opposite direction through the motor, and the motor will turn in the opposite direction. Simple as that.

So, to control motor direction, you simply control the direction of current flowing through the motor. Doing this with a mechanical switch is easy, but it is slow, and it is difficult or impossible to interface with a sophisticated controller. Modern motor controllers do this much better by using *power transistors* instead of switches. The transistors can be discrete parts, or they can all be built onto one large IC. When they are built onto one large IC for this purpose, it is often called a “Full Bridge” or “H-Bridge” motor controller, because the conductors and transistors are often arranged around the DC motor on the schematic in the form of the letter “H”:



Think of each of the 4 separate transistors as a switch: When you close “A” and “C”, current will flow through the motor from left-to-right. When you close “D” and “B”, electrical current will flow through the motor in the opposite direction (right-to-left), so the motor will go in the reverse direction. BUT, you need to be sure to use the transistors in a sensible manner: never activate “A” and “B” together (or “C” and “D”), or you will short out the power supply! This will usually result in the generation of prodigious amounts of heat and smoke for a brief period of time.

Advantages of using transistors over mechanical switches:

Transistors act very quickly (microseconds)

Transistors can be easily controlled by a microcontroller (or other interface)

Power transistors can be used (you can control huge currents)

In addition to controlling current (motor) direction, you can:

Control total current (i.e. motor power: torque and RPM)

Modes for current control: PWM (Pulse Width Modulation), Chopper amp.

BRAKE: Activate “B” and “C” to short both motor terminals to ground:

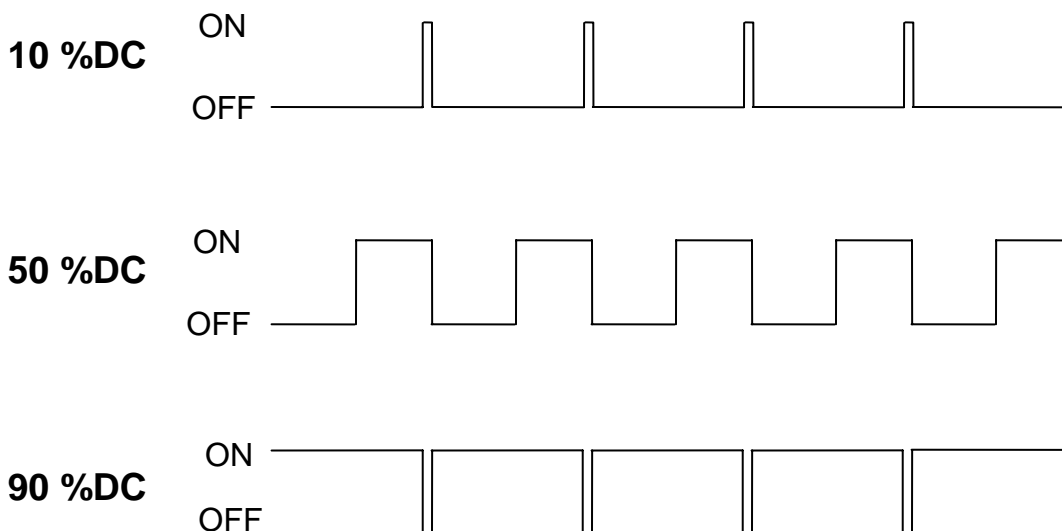
Motor acts as an electric generator with a huge load, so it acts as a BRAKE

CONTROL OF MOTOR POWER:

Basically, we control motor power (Torque and RPM) by controlling the amount of electric *current* that passes through the motor. There are two common ways to control the current flowing through a motor: PWM and Current Control using a current amplifier (such as a “chopper amplifier”)

PWM (Pulse Width Modulation)

PWM essentially amounts to turning a motor (or solenoid, or any other load) ON and OFF very fast. You turn the load ON and OFF so fast that the system does not have time to respond dynamically with the power pulses. The trick here is to control *how long* the relative ON and OFF times are. The ratio of ON time to Total time (Total time = ON + OFF time) is called the **Duty Cycle**, and is often expressed as a % (called %Duty Cycle, or %DC). Examples of 10, 50, and 90% duty cycle are shown below:

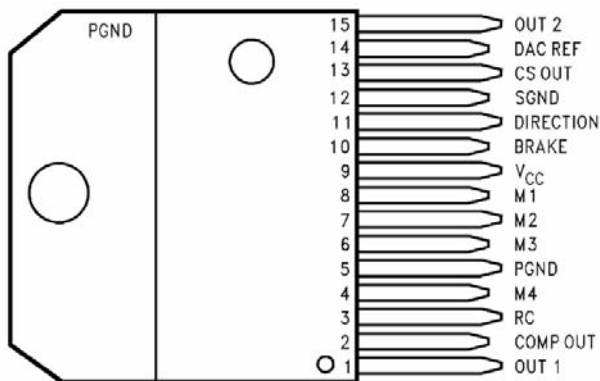


To achieve PWM control of motor current, you simply control the amount of time each pulse is ON during each pulse cycle. Note, generally the pulse period remains constant, you only adjust (modulate) the *width* of each ON pulse. So, you get the same number of pulses in any given time interval, but with increasing %DC, the motor is ON more of the time, so it gets more power. You can implement PWM with a microcontroller and discrete transistors (just use the microcontroller to control both which transistors are active, and to control their %DC). Alternatively, you can get pre-packaged integrated circuits (ICs) that have a built-in PWM function, such as the L298 (made by ST-Microelectronics).

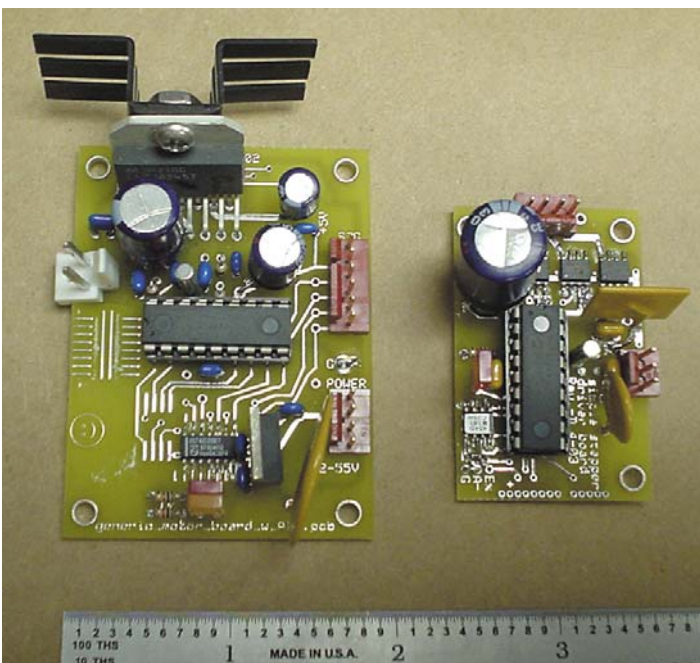
Current Amplification & Control (using a “chopper” amplifier)

Chopper current amplifiers can be packaged together with the transistors in an “H” bridge configuration on a single IC to produce a motor driver chip, often called a *Full Bridge Motor Driver*. One excellent example is the LMD18245, made by National Semiconductor. Unlike PWM, chopper amplifiers use internal current monitoring and feedback to control the electrical current flowing through a motor coil. This is a bit more sophisticated than PWM, but the level of motor control that you can achieve is superb. We will use the LMD18245 for your laboratory exercises. You should read more about how this IC functions. Download the datasheet by going to the National Semiconductor web page (www.national.com) and searching for the chip using the part number and their search engine.

Connection Diagram



Top View
15-Lead TO-220 Molded Power Package
Order Number LMD18245T



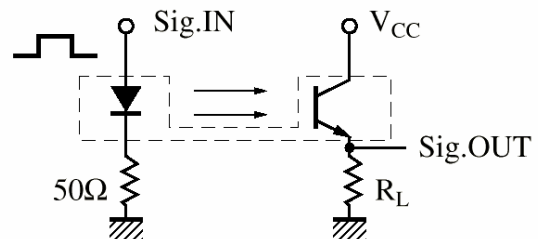
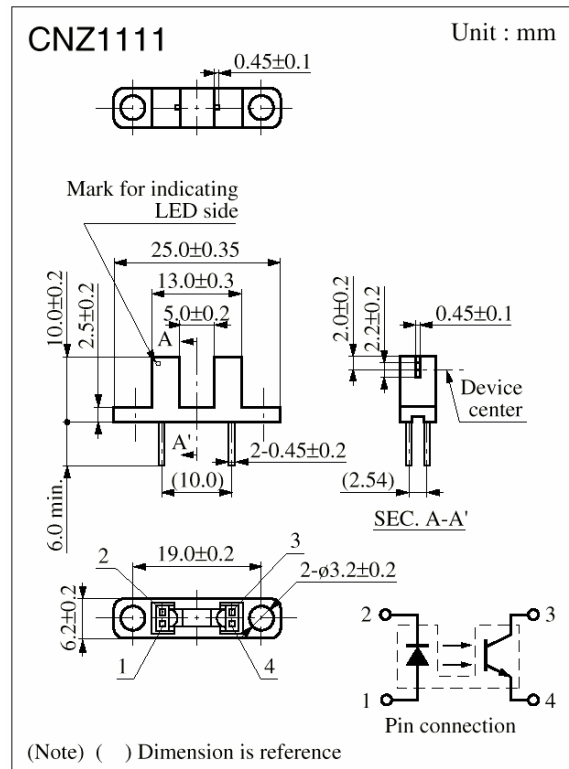
To the left is a photograph of the two driver boards that we have available for use in the laboratory. Both were designed and built by the instructors for this course using ExpressSCH and ExpressPCB. The larger board on the left is the DC motor controller board. You can download the schematic (DC-SCH) and the PC board layout (DC-PCB) for this from the course web page. On the right, the smaller board is the driver for stepper motors. The schematic can be downloaded using the link on the course web page (Step-SCH), and the PC board layout can be

downloaded using the link (Step-PCB). You will need to understand how these boards work in order to write appropriate C code for the microcontrollers, so that they operate properly. Study the schematics for the board you are interested in, and then write some source code to control an appropriate motor using the board. Boards and motors can be checked out from instructors when you are ready. Request assistance from the instructors as required.

PHOTO INTERRUPTERS:

Photo interrupters are extremely common devices that are used in many places, including detectors for people passing through doorways, and to detect the movement of the ball within a computer mouse. A photo interrupter is just a paired optical *emitter* and *detector*. The emitter is usually an LED, which emits infrared light (so you can not see it). The detector is usually just a phototransistor, pointed directly at the output of the LED. The LED is left ON continuously, so it is always detected by the phototransistor. When an object passes between the emitter and detector, the light beam is interrupted and the phototransistor can no longer detect the light from the LED. This results in a loss of current through the phototransistor, which we can easily detect.

For your laboratory exercise, you can also use the photo interrupter shown at the right: the Panasonic CNZ1111. The pin connections are shown at the lower right of the top figure. In the bottom figure you can see the simple electrical circuit that will be used to hook up the photo interrupter. The output signal (Sig. OUT) will be used as a logic level signal (ON or OFF) to tell the microcontroller when your device has reached the end of travel. Thus, we will use photo interrupters as *limit switches*. When the microcontroller detects that the light path has been interrupted, it will reverse the direction of the DC motor that is powering your system. Since you are using a microcontroller, you can do many more sophisticated things than just changing direction: you can gradually slow the system by ramping the power down; you can stop for a fixed delay then reverse or continue; you can stop the motor and activate the BRAKE function on the motor driver IC until you receive another command; or you can do just about anything else you can think of.



SELF QUIZ

- 1: What is a Peltier Device, and what is it used for?
- 2: Sketch a graph showing the relationship between displacement (on the X-axis) and force for a typical solenoid actuator. Just show the basic shape, exact units are not necessary.
- 3: Sketch how you might build a PWM circuit to control the force on a solenoid, or the current flowing through a Peltier device, or the torque output of a DC electric motor, using only a microcontroller (the PIC16F84A) and a HexFET power transistor. Assume everything can operate at +5V. Do your sketch using ExpressSCH. (Hint: you only need the microcontroller and a single transistor to do this).
- 4: What would happen if you simultaneously closed all 4 of the switches (transistors) in an H-Bridge motor controller?

PLEASE ANSWER THE ABOVE QUESTIONS AND E-MAIL TO THE INSTRUCTOR
“I have neither given nor received aid on this examination, nor have I concealed any violation of the Honor Code”

X _____

LABORATORY PROJECTS

For this module you need to understand the circuit for the two motor drivers. You should study the following files, which have links on the course web page:

DC-SCH (if you want to try to control a DC motor, or)
Step-SCH (if you want to control a stepper motor)

Then, you will need to check out the following items from one of the instructors:

DC motor driver board or Step motor driver board
A microcontroller (PIC16F84A)
A DC motor or a step motor
A power supply and any necessary connectors, cables and sensors

1- Control of DC motors:

First have one of the instructors check your source code. Example code can be obtained from the instructors. Use MP-Lab to program one of the microcontrollers with your C code and insert the microcontroller into the IC socket in the DC motor controller board (same programming procedure as you did in modules 09 &10). Be sure that the microcontroller is inserted in the correct direction (align the notch on the IC socket with the notch on the end of the microcontroller). Connect the cable between the motor and the driver board. Set everything on the table so that nothing bad will happen when the motor starts to turn. Connect the driver board to a power supply, +12V to +26V (this will depend on the motor you intend to drive). When you make the power connection the motor may begin to turn almost immediately.

Study the circuit diagram for the DC motor controller and the data sheet for the LMD18245T motor driver chip. You will write C code to make the motor perform the following functions:

- 1- Determine the lowest power setting (in software, for the LMD18245T chip, this is a 4-bit number) that will allow the motor to start without a load.
- 2- Using an adequate current setting to spin the motor quickly, make the motor spin in one direction for several seconds, then coast to a stop for about 4 seconds, then reverse and repeat.
- 3- Enable and disable the "brake" on every other cycle. Does the motor stop more quickly with the brake engaged?
- 4- Use an opaque piece of tape on the acrylic motor flywheel and a CNZ1111 photo interrupter for feedback control. Make the motor spin 10 turns in one direction, then brake, dwell $\frac{1}{2}$ second, reverse, and spin 10 turns in the opposite direction. Make this motion continue in a loop.

2- Control of step motors:

Have one of the instructors check your source code before programming. Be sure not to enable any transistor states that will result in a short circuit. Use MP-Lab to program one of the PIC16F84A microcontrollers with your C code and insert the microcontroller into the IC socket in the step motor controller board (same programming procedure as you did in modules 09 & 10). Connect the step motor to the driver board with the 4- or 5-wire cable (depends on which motor you are using). Turn the power supply OFF while making connections. Connect the ground (green wire) and the power to +5V (yellow wire) and +15V (red wire) on the power supply. Turn the power supply ON. The motor may begin turning automatically. Write source code to make the step motor perform the following tasks:

- 1- Write a short bit of code to drive the motor in one direction for a fixed number of steps, then stop briefly, then reverse for the same number of steps, and repeat this continuously until the power is disconnected. Start with about 50 steps per second. Some step motors can not go much faster than this.
- 2- Determine how many steps there are per revolution of the motor. You can do this by experimenting with the code to change the number of steps before the motor reverses. You may need to label the motor shaft with a piece of tape so you can observe the rotation. Is it labeled on the motor?
- 3- Revert back to your original C code, and then modify it as necessary for this experiment. Change the motor dwell time between steps to make the motor spin as fast as possible. You may need to change the C code a bit to use microsecond delays rather than millisecond delays in the source code. What is the maximum number of steps per second that you can achieve? What happens when you run the step motor too fast (too many steps per second)?
- 4- Notice in the last experiment that you commanded the motor to start up at full speed after every reversal. Now try the following experiment: program the microcontroller to ramp the speed up. The ramp can be pretty quick...less than one second. This way you will not ask the motor to start too abruptly. Using a smooth ramp, can you get the motor to turn at higher step rates than in the previous experiment?
- 5- Program the motor to turn in intervals:
 - 10 turns clockwise (as viewed when looking at the motor shaft)
 - stop and dwell for $\frac{1}{2}$ second
 - 9 turns counter clockwise
 - stop and dwell for $\frac{1}{2}$ second
 - 8 turns clockwise
 - stop and dwell for $\frac{1}{2}$ second... keep going until the motor stops entirely

If you have time and interest, you may devise a system that uses motor control to perform some useful or interesting function. Use simple feedback control to make the project more interesting. Alternatively, you could use the driver boards to control a solenoid or Peltier device. This could readily be used for credit as Module-X1 or -X2, toward completion of the necessary number of modules to get full credit for this course.

FEEDBACK

Was this Module useful and informative?

Is there a topic that should get more or better coverage?

In what way can this Module be improved:

Content: _____

Depth of Coverage: _____

Style: _____

Any additional comments that will help us to improve this course:

If you prefer, you may e-mail comments directly to Bob Dennis: yoda@umich.edu