EE-371 CONTROL SYSTEMS LABORATORY

Session 1
Introduction to Data Acquisition and Real-Time Control

Purpose

The objectives of this session are

- To gain familiarity with the MultiQ3 board and WinCon software.
- Understand the basic I/O connections.
- Create a WinCon application for the encoder and measure the encoder angle.
- Create a WinCon application to run the motor and measure the tachometer and potentiometer signals.

Introduction

Digital control of a continuous-time system has become very popular as the price and reliability of digital computers has greatly improved. Analog controllers are replaced by a digital computer that performs calculations, which emulate the physical controllers. Very complicated control structures can be implemented easily using a digital controller, whereas an analog controller would require very complex hardware. Digital control offers important advantages in flexibility of modifying controller characteristics by changing the program if the design requirement changes or plant dynamic changes with the operating conditions. Furthermore, analog emulation and real-time control provides advanced features such as adaptive self-tuning, multivariable control, expert systems, and the ability to communicate over a local area network.

Analog-to-digital Converter (ADC) and digital-to-analog Converter (DAC)

Typically, the computer replaces the cascade controller. The measured data is converted from analog form to digital by means of the analog-to-digital converter (ADC). This is accomplished by the data acquisition system which is the “eyes and ears” of the digital computer. The digital computer receives and operates on the signal in digital form (numerical). The computer output is then converted to analog form by the digital-to-analog converter (DAC). This process is the “hands and arm” of the digital computer. The full-scale output is usually determined by an external reference voltage. The DAC resolution is defined as the smallest possible change in output. For an N-bit converter the resolution is

$$\text{Resolution } = \frac{100}{2^N}\% \tag{2.1}$$

For example the resolution of a 10-bit DAC is $\frac{100}{2^{10}} = \frac{100}{1024} = 0.09766\%$. The resolution of an ADC is defined as the smallest detectible change in input, which is also given by (2.1).

When the computer reacts to external events as they occur it is referred to as a real-time control system. An important consideration in real-time control is the update rate. During one controller cycle three things must happen before the next cycle can begin (1) sensor are read (A/D inputs), (2) the microprocessor computes an updated commands, (3) the commands are converted to analog output (D/A outputs). Between cycles, the command outputs are held constant. So, if the controller is updated every 0.01 second, the output of the controller would look like a staircase, changing every 0.01 second. As you might expect, the faster the dynamics are of the system you are trying to control, the faster you must update the controller. Again these are important considerations in specifying a control system since high-speed systems can become very expensive.

Modern real-time control systems use computer-aided software engineering (CASE) such as MATLAB Real-Time Workshop to automatically generate C code from the graphical control model such as Simulink together with vender-specific block such as WinCon for handling data transfer between MATLAB and the servo system.

In this lab, we will focus on gaining familiarities the MultiQ3 DACB and WinCon software. In addition, you will explore the effect of A/D and D/A that are built into the MultiQ3 I/O board, through which all the control signals pass.

Procedure

1. Wiring diagram

Using the set of leads, universal power module (UPM), SRV-02 DC-motor, and the connecting board of the MultiQ3 data acquisition board, complete the wiring diagram as follow as shown in Figure 1.1.

Connect the motor to the UPM		**Cable**
From SRV02 Motor connecter	To UPM/To Load	6 pin Din to 4 pin Din, **Gain 1 Cable**
Connect the tachometer to the UPM		
From Tacho on SRV02	To UPM /S3	6 pin mini Din to 6 pin mini Din
Connect the potentiometer to the to the UPM		
From Potentiometer on SRV02	To UPM/S1	6 pin mini Din to 6 pin mini Din
Connect the encoder on MultiQ terminal board to encoder signal on SRV02		
From MultiQ encoder # 0	To SRV02 Encoder	5 pin Din to 5 pin Din
Connect the analog output from MultiQ terminal board to the power module input		
From analog output #0 D/A on MultiQ	To UPM – From D/A	RCA to 5 pin Din
Connect the A/D on the MultiQ to UPM		
From A/D # 0, 1, 2, 3, on MultiQ	To UPM- To A/D	5 pin Din to 4xRCA

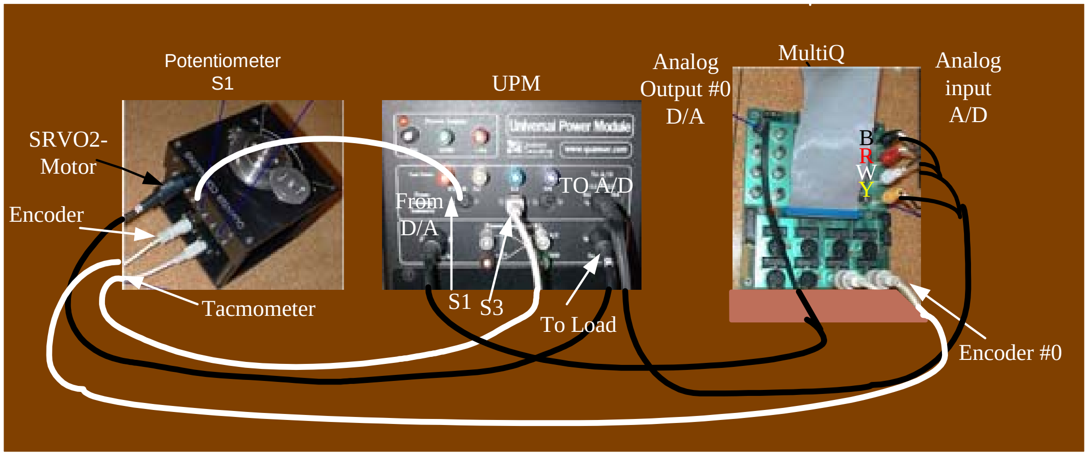

Figure 1.1 Wiring diagram

Do not power up the amplifiers. Make sure the LED on the MultiQ terminal board is illuminated. If it is not then the fuse needs replacement.

2. Sensor Measurements

2.1 Creating the encoder model and measuring Encoder angle

The encoder is used to provide the digital phase information. Start MATLAB, launch Simulink and from the Simulink Library Browser File open a new model. Get the Encoder Input from the Quanser Toolbox/Quanser Consulting MultiQ3 Series as shown in Figure 1.2

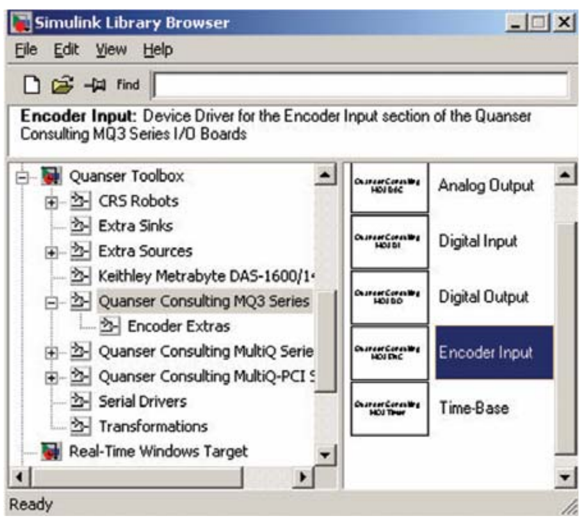

Figure 1.2 Quanser Toolbox and Data Acquisition.

Build a Simulink diagram as shown in the Figure 1.3. The encoder input is connected to Encoder 0 on the MultiQ wiring diagram. Double-click on the Encoder input block to open its dialog box and set the Channel to Use to 0. Save the model (say Lab1A.mdl)

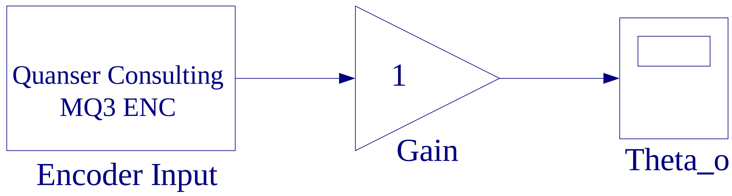

Figure 1.3 Diagram to measure encoder.

Connecting to the client

Before you run any real-time code, you need to launch the WinCon server and connect to the client PC where the experiment is going to run. Ensure that WinCon Client is running on the PC you want to connect to. Start WinCon server on your Laptop and then use **Client Connect** as shown in Figure 1.4 in the dialog box type the proper Client workstation IP address (The station IP addresses are s312-1 to s312-10).

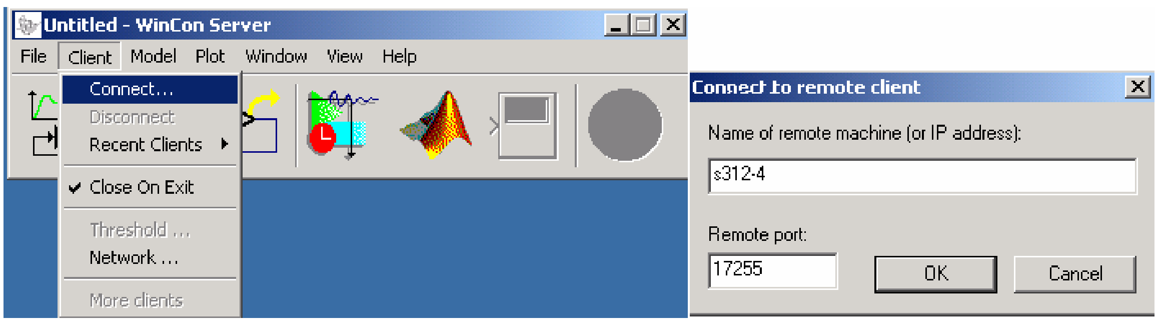

Figure 1.4 Connect to the Client that is running the experiment.

Compiling the model

In order to run the diagram in real-time, you must first build the code for it. This is done using the WinCon/Build option in the diagram shown in Figure 1.5. Select Wincon/Option to open the Simulation parameter dialog box. Select the Solver tab and set the sampling rate on the controller to 0.01 second under "Fixed Step Size". Select ode4 (Rung Kutta) for integration method. In the Simulation drop down menu set the model to External. Then Click WinCon/Build. This will generate the code and compile it. When the compile is done, the code will download to the WinCon Client. You can now use WinCon Server to start and stop the real-time code.

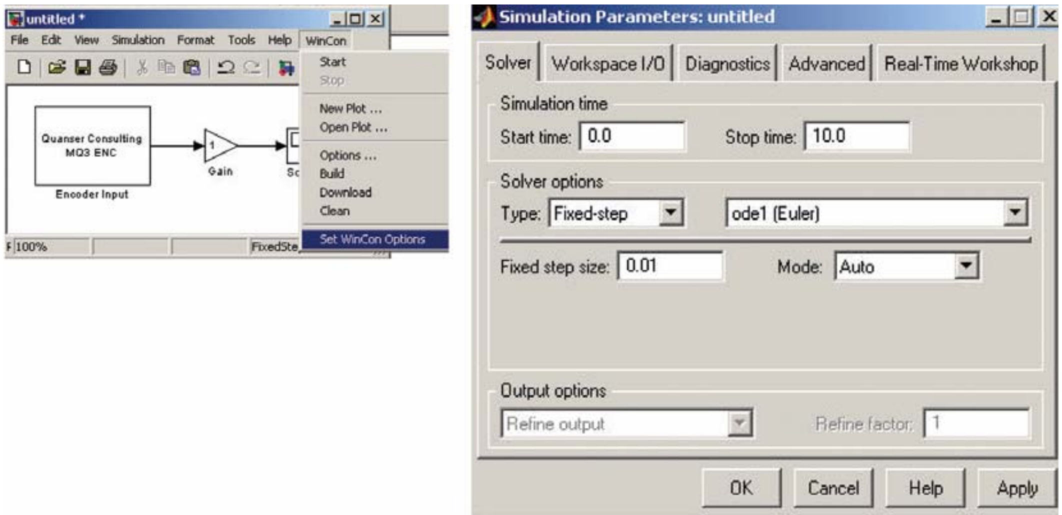

Figure 1.5 Set the WinCon option and Simulation parameters.

Running the code

Now that the code has been compiled, you are ready to run it in real-time. Click on the Start button in WinCon server. It will turn red and display STOP. Clicking on STOP will stop the real-time code and return to the Green button.

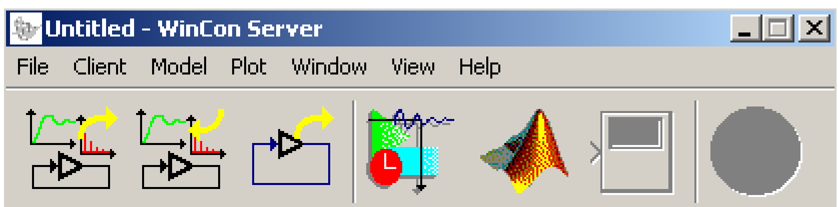

Figure 1.6 WinCon Server

If you have access to the host PC you can see the code in real-time, maximizing the WinCon Client on the host PC will show the following screen.

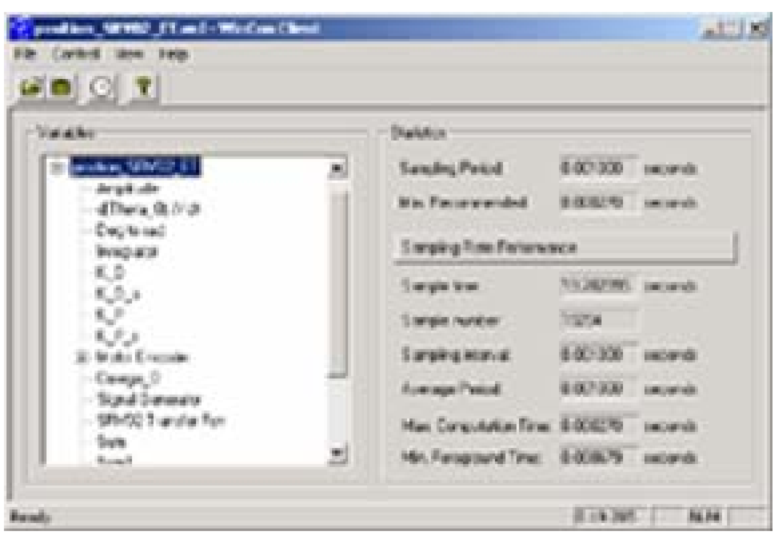

Figure 1.7 Client Applications

The WinCon client is the real-time component of the software and it runs at the period specified under Simulation Parameters / Solver / Fixed Step.

Plotting data

Click on Plot/New/Scope in WinCon Server and in the Select variable to display menu click on Theta_o (i.e., the name you assigned to the Scope) to select the variable you want to plot and press OK. This opens a real-time plot. Click on the Start button on the WinCon Server. As you move the gear by hand the trace will follow your movement. Note the values are not in degrees- they are in counts. The encoder measures 1024 counts per revolution - in quadrature. That means 360 degrees result in 1024*4 = 4096 counts. The resolution of the encoder counter is then 360/4096 = 0.0878906250 degrees. In order to convert the measured value to degrees change the gain block in the Simulink diagram (Figure 1.3) to 360/4096 and the value coming out of the gain block will be in degrees. You also need to ensure that the sign of the measurements is correct. Select the sign of the gain block such that a counterclockwise rotation of the output shaft (middle gear) results in a positive change in the measurement.

Adjust the output shaft (middle gear) so that the angle on Scope plot is near zero. Turn the center gear through any angle say about 45° ccw and note the trace value. Turn the middle gear cw and note the corresponding reading (a negative value).
Turn the servo gear through a small angle slowly and smoothly back and forth; trace will follow your movement. The trace value will be constant when you stop the movement, In order to record, when you have a trace on the graph Press STOP on the WinCon Server.

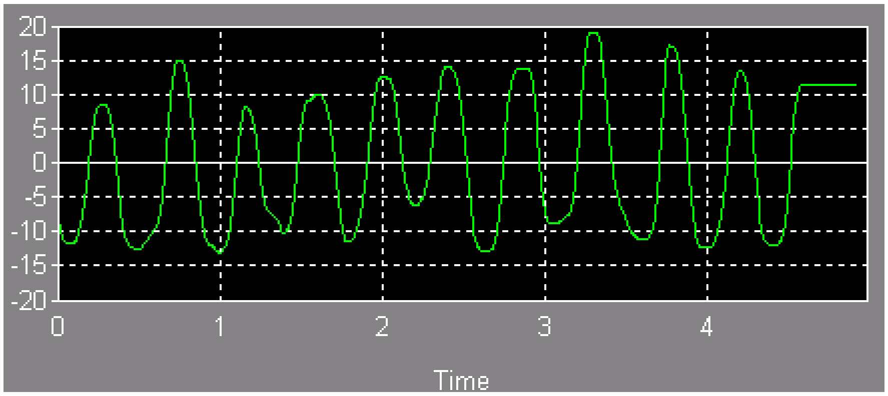

Figure 1.8 Gear movement trace by encode (in degrees)

From the Window pull-down menu, you can use Legend to place a legend on the Figure. Click on any plot in the legend box to select the desired graph background, trace color and the text font.

2.2 Applying a voltage to the motor, and measuring speed with a tachometer and position with a potentiometer

Creating the model

The motor is driven by an amplifier that can deliver the desired power to it. In order to drive a voltage to the motor, you need to output the desired voltage to the desired D/A channel that is connected to the appropriate amplifier. From the wiring you performed, you know that D/A#0 drives the UMP1503, which in turn drives the motor.

Click File/New in WinCon - do not save anything. Start a fresh Simulink diagram. Connect a constant block via a gain block to the Quanser analog output. This will output a constant voltage to the analog block that you specify.

Next add an analog input block to measure the tachometer signal, set Channel to Use to 2. The tachometer generates an analog signal 1.5 V/1000 RPM. Get a gain block and set its gain to –1000/1.5. Get a digital display from the Simulink Sink library and attach it to the gain block. Finally add an analog input block to measure potentiometer signal, set Channel to Use to 0. The potentiometer generates an analog signal 1V/35.2 Deg. Get a gain block and set its gain to –35.2. Get a Scope from the Simulink Sink library and attach it to the gain block. Your completed model should look like the one in Figure 1.9.

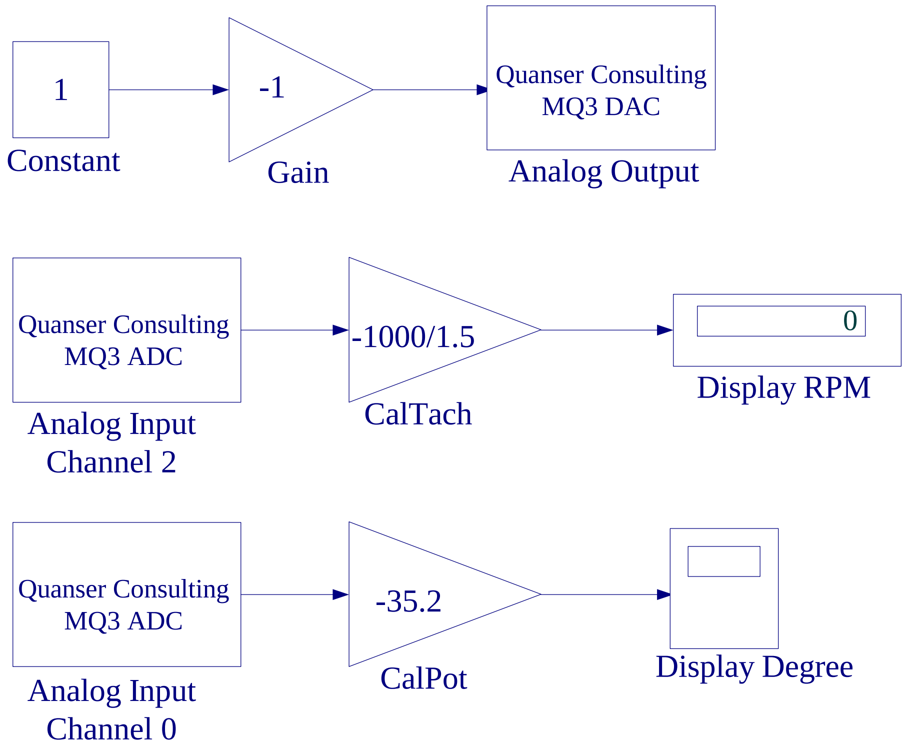

Figure 1.9 WinCon application, applying a voltage to servomotor and measuring with a tachometer and potentiometer

Save the model (say Lab1B.mdl). See that you are connected to the proper client PC. Set the WinCon options (select the Solver tab and set the sampling rate on the controller to 0.001 seconds under "Fixed Step Size". Select ode4 Rung Kutta for integration method). In the Simulation drop down menu set the model to External.

Click WinCon/Build, this will generate the code and download it to the client. Run the model; this will output 1 volt to the analog output and drive the amplifier, which in turn drives the motor. The output shaft (middle gear) will turn in the clockwise direction. Changing the value of the Constant block will change the speed of the motor. Changing the sign of the constant block will change the direction. If we make the gain block = -1, then the output shaft will turn ccw when the constant is positive. If the amplifier is being used with a gain cable, a block should be inserted before the D/A of 1/(cable gain) in order to obtain overall unity gain.

Measuring speed with the tachometer

In order for the motor to turn ccw when a positive voltage is applied to it in the Simulink diagram set the gain block to –1. A positive voltage should result in a ccw rotation of the output shaft (middle gear), and a ccw rotation should result in a positive tachometer measurement. Adjust the signs of the gains to achieve this. You can display the speed on a digital meter. To display a digital meter, click on Plot/New/Digital meter in WinCon Server and in the Select variable to display click on Display RPM (i.e., the name you assigned to the Digital meter) to select the variable you want to display and press OK. This opens a real-time Digital meter.

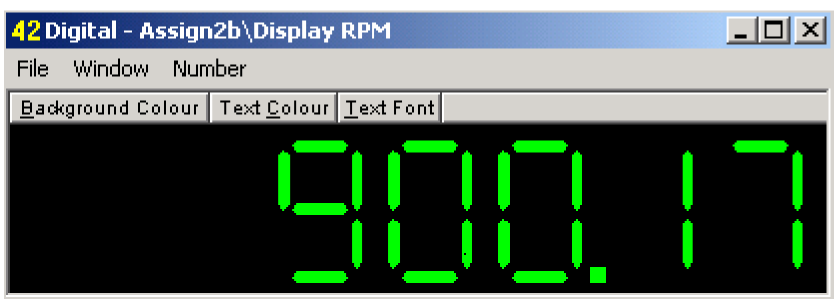

Figure 1.10 Tachometer reading

The reading on the digital meter fluctuates due to a high frequency ripple in the tachometer - The ripple in this tachometer is about 7% peak to peak of the average value. So for 1000 RPM at the motor you will see a ripple of 70 RPM p-p. Furthermore the speed of this open loop system is sensitive to noise and small change in the load. Since the external gears impose nonuniform loading as the motor turns the motor speed is not constant over the rotational range. By passing the signal through a low pass filter the tachometer noise can be reduced. This is done in the speed control design project.

Measuring position with the potentiometer

The potentiometer is a single turn 10 KΩ sensor with no physical stops. Its electrical range is 352 degrees. The potentiometer generates an analog signal proportional to the angle of rotation. The difference between a potentiometer and an encoder is that the encoder can measure several turns continuously while the potentiometer can only measure one full turn and then roll over. A ccw turn results in a positive measurement. Note that the potentiometer is an absolute position device while the encoder is a relative position device. You can adjust the potentiometer shaft rotation relative to the gear such that the zero measurement is at a desired angle always. You can plot the display on either a WinCon scope or digital meter. To open a scope, click on Plot/New/Scope in WinCon Server and in the Select variable to display dialog box, click on Display Degree (i.e., the name you assigned to the Scope) to select the variable you want to plot and press OK. This opens a real-time plot. In order to obtain a MATLAB Figure plot, from the File menu choose save the plot as M-file, and specify a file name. Now at the MATLAB prompt if you type the file name you can obtain the MATLAB Figure plot. You can type grid to place a grid on the graph or edit the Figure as you wish.

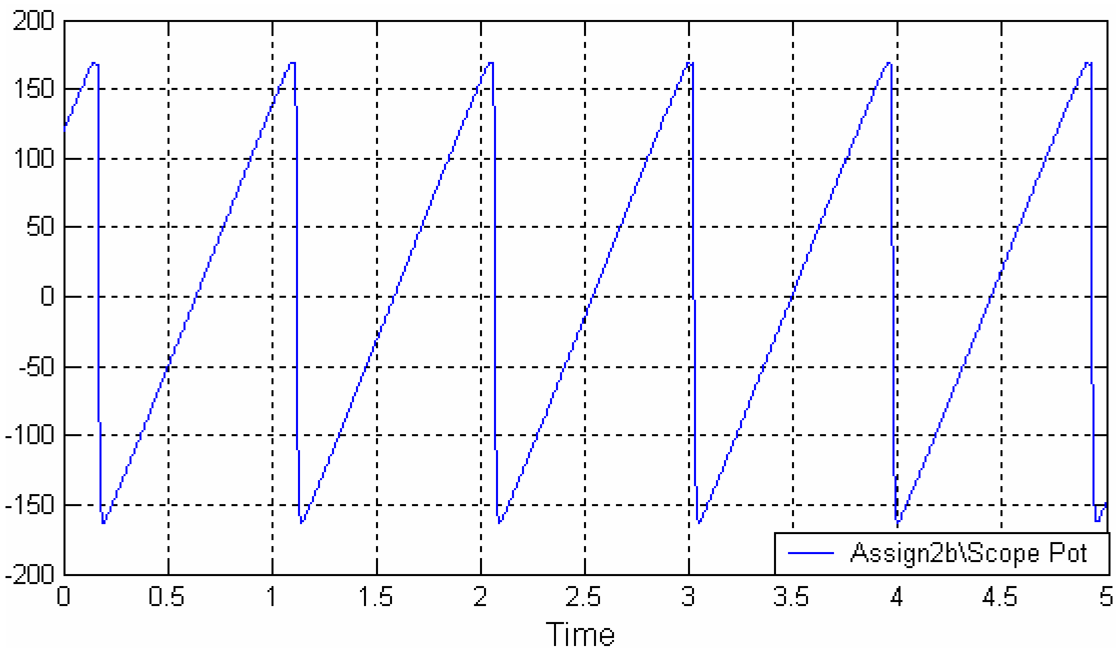

Figure 1.11 potentiometer trace

3. Knowledge test

As this is an introductory session, there is no report to write. You should however ensure that you have understood the principles of operation of WinCon. You should also have a good understanding of how angles are measured and how the voltages are applied to the motor. In order to ensure that you understand the systems ensure that you can reply to the following questions correctly (and confidently!).

- How many actuators and how many sensors does the system have?
- Which D/A channel drive the UMP?
- Which encoder channel measures the encoder?
- Why is the sign of the calibration constant important?
- Why is the sign of the voltage signal important?
- Why does the cable gain have to be compensated for in the motor model?