0 Preliminary - Hardware and Software Overview

0.1 Introduction

The undergraduate Control Systems Lab is located in ETLC E5-006. In the lab, there are 15 PCs equipped with data acquisition systems running under the Windows XP environment. The hardware equipment and some software tools, e.g., WinCon 5.0, are manufactured by Quanser Consulting (www.quanser.com), a Canadian company developing real-time control systems for education and research.

This chapter introduces some of the hardware equipment and software tools to be used in the Control Systems Lab. Familiarity with this chapter is assumed in the labs. Briefly, this chapter is organized as follows:

Section 0.2 describes the hardware equipment;

Section 0.3 describes the software tools;

Section 0.4 provides an example of using the hardware and software;

Section 0.5 lists the issues related to safety and caution

0.2 Hardware Equipment

The hardware devices to be introduced are analog plant simulators, data acquisition systems, power modules, and servo-motors.

0.2.1 Analog Plant Simulator

An analog plant simulator (APS) is shown in Figure 1, which consists of a set of 11 independent electronic circuits (building blocks) that can be interconnected to study the behavior of dynamical systems. A series of potentiometers allow you to adjust the parameters of each block within a certain range. RCA jacks are used throughout so you can easily connect the output of one block to the input of another. A DC power unit converts AC (from a wall power outlet) to DC for the APS.

The 11 building blocks on the APS are listed below:
### Preliminary

**Figure 1: The analog plant simulator (APS)**

<table>
<thead>
<tr>
<th>Qty</th>
<th>Label</th>
<th>I/O Relation or TF</th>
<th>Parameter Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Sum</td>
<td>( y = u_1 + u_2 )</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Diff</td>
<td>( y = u_1 - u_2 )</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Gain</td>
<td>( y = K \cdot u )</td>
<td>K: 0-10.</td>
</tr>
<tr>
<td>3</td>
<td>Lag (1st-order)</td>
<td>( H_1(s) = \frac{A}{s + A} )</td>
<td>A: 0-10.</td>
</tr>
<tr>
<td>2</td>
<td>Lag (2nd-order)</td>
<td>( H_2(s) = \frac{D_0}{s^2 + D_1 s + D_0} )</td>
<td>D_0: 0-10^4; D_1: 0-2000.</td>
</tr>
<tr>
<td>1</td>
<td>PID (with switch set to 1)</td>
<td>( Y = (K_p + \frac{K_i}{s})(U - U_d) + K_dsU_d )</td>
<td>K_p: 0-10; K_d: 0-2; K_i: 0-10.</td>
</tr>
<tr>
<td>1</td>
<td>PID (with switch set to 2)</td>
<td>( Y = (K_p + K_ds + \frac{K_i}{s})(U - U_d) )</td>
<td>K_p: 0-10; K_d: 0-2; K_i: 0-10.</td>
</tr>
</tbody>
</table>

### 0.2.2 Data Acquisition System

For computer implemented controllers, analog-to-digital (A/D) and digital-to-analog (D/A) conversions are necessary. These are done using the data acquisition and control board (DACB), which inputs the measured signal(s) to the computer and outputs control action to the actuator in the control loop. The DACB in this lab consists of two parts: the MultiQ-PCI data acquisition card and terminal board, which are both made by Quanser Consulting.

Figure 2 shows a photo of the MultiQ-PCI data acquisition card, which has 8 analog inputs, 8 analog outputs (only 4 available with our terminal board), 16 bits of digital inputs and outputs, 3 programmable timers, and up to 6 encoder inputs decoded in quadrature. These data acquisition cards are installed in PCI slots inside the computers.

The MultiQ-PCI terminal board, shown in Figure 3, is a bench-top board for interfacing the MultiQ-PCI data acquisition card with other analog equipment such as the APS, servo-motor, and power module. Five grey ribbon cables are used for connection. Further details on proper connections will be discussed in the later chapters.
Figure 2: The MultiQ-PCI data acquisition card (inside the PC)

Figure 3: The MultiQ-PCI terminal board
0.2.3 Universal Power Module

The universal power module (UPM-2405), part of which is shown in Figure 4, is a linear power operational amplifier. The MultiQ-PCI data acquisition card cannot deliver enough power to the actuators used in this lab; therefore, a signal buffer is needed. The UPM-2405 is used as our signal buffer since it can deliver up to 5A to an actuator in a non-inverting, unity-gain configuration.

![Image of Universal Power Module](image)

Figure 4: Top front view of the universal power module

The following connections can be made to/from the UPM (see also Figure 4):

- **From analog sensors**: There are four (S1-S4) inputs which can be connected from analog sensors (and then subsequently to the computer); the cable used is a “6-pin mini-din/6-pin mini-din” cable (light tan color), which is now referred to as the analog sensor cable.

- **To A/D**: The four analog sensor signals (S1-S4) can then be connected to the MultiQ-PCI terminal board for A/D conversion into the computer; the cable used is a “5-pin din-stereo/4×RCA” cable (black color), which is now referred to as the A/D cables.

- **From D/A**: This is where you input the D/A signal from the MultiQ-PCI terminal board to the UPM; the cable used is a “5-pin din-mono/RCA” cable (black color), which is now refereed to as the D/A cable.

- **To load**: Here you connect the amplified D/A signal to an actuator (e.g. servo-motor); The cable used is a “6-pin din/4-pin din” cable (black color), which is now refereed to as the load cable.

- **Others**: A few other connections are possible for convenience: e.g., the power supply on the top left is for ±12 volts; the signals from analog sensors S1-S4 can be easily monitored by connecting to a scope.

0.2.4 Servo-motor

The Quanser DC servo-motor (SRV02) is shown in Figure 5. A 3W motor is mounted in a solid aluminum frame and drives a built-in Swiss-made 14.1 : 1 gearbox whose output drives an external gear, which is attached to an independent output shaft that rotates in an aluminum ball-bearing block. The output shaft is equipped with an encoder. The external gear on the output shaft drives an anti-backlash gear connected to a precision potentiometer for measuring the output angle. The external gear ratio can be changed from 1 : 1 to 5 : 1 using different gears. Two inertial loads are supplied with the system in order to examine
the effect of changing inertia on motor performance. The motor is configured using the armature control scheme.

Several connections are available for the servo-motor as seen in Figure 5. The input voltage connects to the UPM using the load cable. The potentiometer and tachometer ports connect to the UPM-2405 using sensor cables and are used to measure angular position and angular velocity, respectively. Additionally, the shaft encoder port connects to the terminal board using an encoder cable and is used to measure angular position.

The following calibration factors are needed in order to use the sensors in units of degrees or radians:

<table>
<thead>
<tr>
<th>Connection</th>
<th>Conversion (Rad)</th>
<th>Conversion (Deg)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Encoder</td>
<td>$\frac{2\pi}{4096}$ rad</td>
<td>$\frac{360}{4096}$ deg</td>
</tr>
<tr>
<td>Tachometer</td>
<td>$\frac{1}{180}$ rad/sec</td>
<td>$\frac{\pi}{180}$ rad/sec</td>
</tr>
<tr>
<td>Potentiometer</td>
<td>$\frac{1}{4096}$ rad</td>
<td>$\frac{\pi}{4096}$ rad</td>
</tr>
</tbody>
</table>

Two external gear ratios are available for the servo-motor: the low gear ratio (Figure 6) and the high gear ratio (Figure 7). Normally, the first one is used in motor position control experiments, and the second is used when the motor acts as an actuator to drive other plants such as the ball and beam, or the inverted pendulum.

Different amounts of inertia can be bolted to middle gear of the servomotor. This achieves a different plant for each servomotors.
0.3 Software Tools

The software tools used in the lab are:

- MATLAB/Simulink with its Control Systems Toolbox
- WinCon
- RTX Runtime
- Visual C++

WinCon is developed by Quanser Consulting to interface MATLAB/Simulink with hardware equipment, e.g., the MultiQ-PCI data acquisition system. WinCon uses a Simulink model to generate, compile using Visual C++, and run the program to control the MultiQ-PCI data acquisition system. Venturcom produces the RTX Runtime application, which gives a real time running environment for the MultiQ-PCI control programs despite using Windows, a non real time OS. In this section, we will focus on Simulink and WinCon.

0.3.1 Simulink

Simulink allows a MATLAB user to simulate complex control systems through block diagram connections. In this section, we briefly introduce the steps to use Simulink.

- **Starting**: Typing in Simulink in the MATLAB command window will open the Simulink window, see Figure 8. By selecting New from the File menu, you get a new system window, which you use to build a Simulink model. This model can be saved as a mdl-file for further editing and simulation.

- **Creating blocks**: Simulink has a standard block library shown in Figure 8 under the sub-group Simulink; additionally, there are extended block libraries such as Communication Blockset, Control System Toolbox, and so on. To create a Simulink model, drag and drop the required blocks into the system window, as shown in Figure 9 (left side).
• Connecting inputs and outputs: To connect the output of block A to the input of block B, left click the output port of block A, drag it to the input port of block B, and release the mouse button. To connect a point in a data line with an input or output port, press and hold the Ctrl key, and then click on the point in the line and drag it over to the port. When the ports are connected, the angle brackets disappear and directed lines indicate information flow as shown in figure 9 (right side).

• Simulation parameters: View and edit the simulation parameters by selecting Parameters from the Simulation menu as shown in figure 10. The parameters worth noting are:

  (a) Solver method: fixed-step is the default. You should change this to variable step using ODE5 when you are using blocks from the “continuous” toolbox.

  (b) Start and stop time: 0 and 10 seconds is the default.
(c) Step size: 0.001 seconds is the default; this parameter should be increased (e.g. 0.01 seconds) for longer simulations.

After setting the simulation parameters as required, close the dialog box by clicking OK.

Figure 10: Simulink simulation parameters

- **Simulating**: To start the simulation, select **Start** from the **Simulation** menu. The simulation begins and the scope stores the output time response; double click on the scope to see the plot.

- **Plotting**: The best way to capture the generated data is to use the **To File** block under the **Simulink** sub-group. This block will save the data in a *.mat file. The data in the *.mat file can be loaded into the MATLAB workspace using the **load** command. The data can subsequently be viewed, analyzed and plotted.

### 0.3.2 WinCon

WinCon, a software package from Quanser Consulting, is a real-time application that runs Simulink generated code using RTX Runtime on a PC under running Windows. WinCon allows controllers constructed in Simulink diagrams to be implemented in real time via the computer. For such a typical application, you need to start a WinCon Server and a WinCon Client. (One WinCon Server can communicate with several WinCon Clients, and visa versa.) In the simple case, one PC and no network, the configuration and required software tools are illustrated in Figure 11.

**WinCon Server:**

The WinCon Server performs the following tasks:

- Converts a Simulink diagram into a WinCon controller library wcl-file using the MATLAB Real-Time Workshop;
- Compile and link the code using Visual C++;
- Download the WinCon controller library file to run on the WinCon Client;
- Start and stop the WinCon Client;
Figure 11: Stand-alone PC configuration

- Maintain TCP/IP communications with the WinCon Client;
- Maintain communications with Simulink for updating real-time changes in the controller parameters;
- Make changes to WinCon Client parameters using Control Panels;
- Plot the data streamed from a desired WinCon Client in real time;
- Save data to disk;
- Offer an External Interface Window to allow outside applications to receive streamed data from a WinCon Client and to perform real-time changes to the WinCon Client parameters.

**WinCon Client:**

The WinCon Client is a real-time application that runs the code generated from the Simulink diagram at the sampling rate specified; its functions are summarized below:

- Receive the controller code from the WinCon Server as a WinCon controller library wcl-file;
- Run the controller code in real time;
- Maintain communications with the WinCon Server;
- Stream real-time data to any WinCon Server(s) requesting it.

**MultiQ-PCI Terminal Board and WinCon in Simulink:**

WinCon is fully compatible with the MultiQ-PCI terminal board. Open the Simulink Library Browser and find the library of board drivers – see Figure 12. The blocks can be placed in a system window. Double clicking on the block opens the block properties dialog.

- **Analog input:** The analog input dialog allows you to specify the channel number (i.e. the analog input port used on the terminal board). The board number and sample time should be kept to their respective defaults as shown in figure 13. The input voltage has a resolution of 12 bits and has a programmable voltage range of up to ±10 volts.
Figure 12: Viewing the library of board drivers in Simulink

Figure 13: Dialog for the MultiQ-PCI analog input block
• **Analog output:** The analog output dialog allows you to specify the channel number. (i.e. the analog output port used on the terminal board). The board number, initial value, final value, and sample time should be kept to their respective defaults as shown in figure 14. The output voltage, with a set range of ±10 volts, has a resolution of 13 bits.

![Figure 14: Dialog for the MultiQ-PCI analog output block](image)

• **Encoder input:** The encoder input dialog allows you to specify the channel number (i.e. the encoder input port used on the terminal board). The board number and initial value should be kept to their respective defaults as shown in figure 15. The encoder gives a positive integer count with resolution dependant on the type of encoders selected; the encoder counter has 24 bits.

![Figure 15: Dialog for the MultiQ-PCI encoder input block](image)

• **Digital input and output:** The MultiQ series also have 8-bit digital input and output capability. There are currently no plans to employ these I/O ports.

**Generating Controller Codes for Simulink Diagrams:**

If WinCon is installed properly, a WinCon menu item appears in the Simulink Window, which allows you to generate and run the real-time code seamlessly. Simply construct/load the controller diagram you want to run in real time; and then in the WinCon menu select Build. This will initiates the process of generating the real-time code. When the process is done, a message appears in the MATLAB window indicating that
a wc-file has been successfully created. A pop-up window will appear automatically for executing the controller—see Figure 16.

![Figure 16: Pop-up window for executing controllers](image)

**Running Controllers:**

You can start and stop the controller on the WinCon Client to which the WinCon Server is connected using one of the following two ways:

1. Use the **Start/Stop** button on the WinCon pop-up window to manually start and stop the simulation.
2. Use the MATLAB commands `wc_start` and `wc_stop` to start and stop the simulation, respectively. These commands are particularly useful for automating multiple trial operation.

**Displaying and Saving Data:**

Signals can be displayed from both Simulink by selecting the WinCon→Plot Open menu and the WinCon Server using the Plot→Open menu. You can then choose to display a desired variable as a real-time plot (e.g., $x(t)$ vs. $t$), a digital meter form (like a digital volt-meter), or an $x$-$y$ plot ($y$ vs. $x$).

To save data from a Wincon Server plot, simply click on **File→Save**. The data can be saved in one of three ways:

- **m-file**: This creates a `.m` file. Typing the name of the file in the MATLAB command window brings up the MATLAB plot window; the vector plotted are loaded into the MATLAB workspace.

- **mat-file**: (Recommended) This creates a `.mat` file. Use the command `load` to load the data into the MATLAB workspace. The time vector is named `plot_time` while the data vector has a name `X_Y` where `X` is the Simulink model name and `Y` is the Simulink scope name, which is “Scope” by default. Use MATLAB command `whos` to check the name of the variables from the stored data arrays. Use the commands `plot` and (optionally) `subplot` to plot the data.

- **To workspace**: This is equivalent to saving to a `.mat` file and then loading it into the MATLAB workspace—except that you do not have the data saved in a file.

Alternatively, You can use MATLAB script `wc_saveplot('plot title', 'filename.mat')` to save your data to a mat-file. The plot title is the name appear at the top of the visual scope.

### 0.4 An Illustrative Example

This section illustrates how to use Simulink and WinCon in order to perform basic a I/O task with the DC servo-motor. Follow the outlined procedure below to familiarize yourself with the hardware and software.
• Build a Simulink model as shown in Figure 17. The model applies a constant voltage to the servomotor. The encoder is employed to acquire the angular position over time.

• The input and output channel numbers should match the channels used on the terminal board. Refer to figure 3 for terminal board channel location.

• The calibration factors need to be set so that the servo-motor angular position and velocity is acquired in appropriate units. Refer to subsection 0.2.4 for calibration details of the servo-motor.

• Click on Simulation → Simulation Parameters from your SIMULINK screen. In our example we are not using blocks from the continuous library. Therefore, the solver options do not need to be changed.

• Click on WinCon → Build from Simulink menu. This generate real-time C code for the diagram. Wait until the compilation process is completed. The MATLAB window displays the progress of the code generation task and when it is completed the message appears:

    Successful completion of RTW build procedure for model: xxxxx

Following the code generation, WinCon Server and WinCon Client are automatically started. The generated code is then automatically downloaded to the client and the system is ready to run.

• Click on the Plot button in Wincon Server and select Plot → New → Scope. The names of all blocks in the diagram appear in a Multi Select Variable Tree. You can select the variables which you want to display.

• Once the generated code is downloaded to the client, click on Start/Stop button in the Wincon Client window to start execution of the SIMULINK model in real-time. The open-loop response of the motor position measured from the encoder (Display1) is shown in figure 18

• You can save the data by clicking on File—Save on the plot window, or by using the MATLAB command wc_saveplot as mentioned earlier.

0.5 Safety and Caution

Because mechanical and electrical/electronic devices are involved, always handle them with care. Our goal is to eliminate unsafe operations and damage to equipment. Listed below are some important points:

• Read all instructions (this manual and relevant lab chapter) before coming to the lab. Mark down things you do not understand or are not sure about, and discuss them with the lab instructor/TA.
Whenever possible, the lab instructor will demonstrate wiring the required equipment at the beginning of each lab; make sure you come to the lab on time and observe the demos.

When connecting wires, keep power off the equipment, e.g., the MultiQ-PCI terminal board and the APS. Ask the lab instructor/TA to check you wiring before turn on power.

For experiments involving a servo-motor unit, it is very important to remember not to send high-frequency voltage signals directly to the motor or damage will occur in the motor gearbox and brushes. As an example, if one uses a differentiator to compute the speed signal from the position signal, make sure the high-frequency components are removed—using band-limited differentiator. Normally, if one hears a “buzz” noise in the motor, turn off the motor immediately and check the control system.
**ROTARY CONTROL CHALLENGE**

**SRV02 + FLEXGAGE = Rotary Flexible Link Experiment**

**Description**

The Rotary Flexible Link module (FLEXGAGE) module is designed as an attachment to the SRV02 plant. The module consists of a thin stainless steel link instrumented with a strain gage. The arm deflection is measured via the strain gage output. The model is designed to accentuate the effects of flexible links in robot control systems. Such flexibility is common in lightweight robots designed for space applications.

**Key Features**

- Fully compatible with MATLAB/Simulink
- Modular design (experiments are easily interchangeable)
- High Quality Aluminum chassis with precision crafted parts
- High Resolution Strain Gage to sense link deflection
- Fully documented system models & parameters
- Fast and easy attachment to the SRV02 plant
- Open architecture design

**Curriculum Topics**

- Disturbance Rejection
- Tracking Control & Regulation
- Full State-Feedback
- Observer Design & Implementation
- Frequency Analysis
- Lead / Lag Compensation
- Vibration & Resonance
- System Modeling & Simulation
- Root Locus Design
- Nyquist Stability
- Robotics
- Real-Time Control
- Discrete Time Sampling
- System Identification
- Multivariable Control Design

The SRV02 series serves as the base of Quanser’s Rotary Control Challenges. With easily interchangeable modules, you can transform the SRV02 into any of these experiments:

**SISO Configurations (Single Input, Single Output)**

- SRV02: Position Control
- SRV02-T: Rate Control
- BB01: Ball & Beam
- ROTFLEX: Rotary Flexible Joint
- FLEXGAGE: Rotary Flexible Link
- ROTPEN: Rotary Gantry
- ROTPEN: Rotary Inverted Pendulum
- ROTPEN-SE: Rotary Self-Erecting Inverted Pendulum
- DBPEN: Double Inverted Pendulum

**MIMO Configurations (Multiple Input, Multiple Output)**

- 2D ROBOT: 2 SRV02 Modules Coupled together to control 2 axes.
- 2D GANTRY: Use the 2D ROBOT to Control the position of the gantry in 2 planes.
- 2D PENDULUM: Control the Inverted Pendulum with 2 degrees of freedom.

Some configurations require a specific SRV02 model, please confirm at time of order.

All SRV02 models are supplied with additional gears to configure the required ratio as well as an extra set of external loads to vary the inertia. The following models are available:

<table>
<thead>
<tr>
<th>Model</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SRV02</td>
<td>Standard Servo plant. Instrumented with a continuous turn potentiometer to measure output/load angular position.</td>
</tr>
<tr>
<td>SRV02-E</td>
<td>Same as the SRV02 with an optical encoder measuring the output shaft position.</td>
</tr>
<tr>
<td>SRV02-EHR</td>
<td>Same as the SRV02 model equipped with a high resolution optical encoder to acquire high precision position data.</td>
</tr>
<tr>
<td>SRV02-ET</td>
<td>Same as the SRV02-E with a tachometer attached to measure the speed of the motor.</td>
</tr>
<tr>
<td>SRV02-ETS</td>
<td>Same as the SRV02-ET but with a slip-ring mounted to the load gear allowing a continuous 360° motion.</td>
</tr>
</tbody>
</table>
The following graph demonstrates the difference between PD position control and full-state feedback control. Notice how the full-state feedback controller has dampened the arm oscillations while maintaining a fast response time.

![Graph showing PD position control vs. full-state feedback control with vibration damping.](image)

Figure 1 - Vibration Damping Using Full-State Feedback

The Rotary Flexible Link Module (FLEXGAGE) is designed as an attachment to the SRV02 plant. Along with the SRV02 plant, the following components are required to complete the experimental setup.

<table>
<thead>
<tr>
<th>Component</th>
<th>Quanser Recommended (Common Configuration)</th>
<th>Alternative</th>
</tr>
</thead>
<tbody>
<tr>
<td>Power Module</td>
<td>Quanser UPM 1503/2405</td>
<td>Alternate Power Amplifier (Minimum requirements: +/- 12V, 3A)</td>
</tr>
<tr>
<td>Control Hardware</td>
<td>Quanser Q4, Q8 Series</td>
<td>dSPACE DS1104*</td>
</tr>
<tr>
<td>Control Software</td>
<td>Quanser WinCon</td>
<td>National Instruments E-Series DAQs*</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The Mathworks – RTWT, xPC</td>
</tr>
<tr>
<td></td>
<td></td>
<td>dSPACE – ControlDesk</td>
</tr>
<tr>
<td></td>
<td></td>
<td>National Instruments – Labview RT</td>
</tr>
</tbody>
</table>

* Quanser offers interface boards for NI E-series & dSPACE DS1104 boards.

<table>
<thead>
<tr>
<th>Specification</th>
<th>Value</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>Module Dimensions</td>
<td>48 x 2</td>
<td>cm²</td>
</tr>
<tr>
<td>Main Arm Length</td>
<td>30</td>
<td>cm</td>
</tr>
<tr>
<td>Strain Gage Bias Power</td>
<td>±12</td>
<td>Volts</td>
</tr>
<tr>
<td>Strain Gage Measurement Range</td>
<td>±5</td>
<td>Volts</td>
</tr>
<tr>
<td>Strain Gage Calibration Gain</td>
<td>1</td>
<td>Volts/Inch</td>
</tr>
<tr>
<td>Flexible Link mass</td>
<td>0.065</td>
<td>kg</td>
</tr>
<tr>
<td>Flexible Link rigid body inertia</td>
<td>0.005</td>
<td>kgm²</td>
</tr>
<tr>
<td>Equivalent Link Stiffness</td>
<td>0.5</td>
<td>Nm/rad</td>
</tr>
<tr>
<td>Fundamental Natural Frequency</td>
<td>3.2</td>
<td>Hz</td>
</tr>
</tbody>
</table>

For SRV02-Series specifications please refer to Product Information Sheet R1.