

Experiment 1: Introduction to PC-Based Data Acquisition and Real-Time Control

Tools/concepts emphasized: Matlab, Simulink, Real-Time-Workshop (RTW), WinCon, MultiQ-3, data acquisition, and real-time control.

1. Introduction

All real-world applications of feedback control involve

- i)* mathematical modeling of physical plants;
- ii)* system/parameter identification;
- iii)* feedback control design;
- iv)* off-line computer simulation to evaluate closed-loop system performance;
- v)* real-time feedback control implementation using analog/digital hardware; and
- vi)* on-line controller adjustment to optimize closed-loop system performance.

You have been familiarized with steps *i)*, *iii)*, and *iv)* in Automated Control–ME 322. In the Control Laboratory–ME 325, we will reiterate some aspects of steps *i)*, *iii)*, and *iv)*, as required; however, our primary focus will be on steps *ii)*, *v)*, and *vi)*.

Traditionally, control systems have been designed and analyzed using analog methods such as the Laplace transform. In addition, until 1960's, a vast majority of industrial control systems were implemented using analog technology based on mechanics (e.g., moving bars, linkages, etc.), pneumatics, and electronics (e.g., resistors, capacitors, op-amps, etc.). However, with the advent of digital computer technology, control engineering has witnessed a significant shift towards digital implementation of feedback controllers [1]. In contrast to analog implementation of feedback control, digital implementation offers small size and low cost. Furthermore, digital controllers are inherently flexible since they can be changed by reprogramming, whereas analog controllers are changed by extensive rewiring [1].

In many current industrial and commercial applications of feedback control such as machine tools, robotics, automotive system, etc., micro-controllers are extensively used. Micro-controllers

are typically programmed either in low-level machine language or in high-level languages such as C via PC interfaces. The programming of micro-controllers for implementing advanced control algorithms is a specialized task and requires trained personnel. However, in the last decade, with the advent of the fourth generation computer programming tools such as the computer-aided software engineering (CASE), it has become feasible to automatically generate C code from graphical control-system simulation tools such as Simulink. In particular, using the Simulink block library and RTW along with vendor-specific block libraries, one can generate C code from Simulink-based feedback control diagrams for real-time controller implementation on PC and DSP-based data acquisition and control boards (DACB).

In the first laboratory exercise, we will focus on gaining familiarity with the MultiQ-3 DACB [2] and Matlab, Simulink, RTW, and WinCon [3] software. The MultiQ-3 DACB provides the following functionalities: analog to digital conversion (ADC), digital to analog conversion (DAC), digital I/O, and encoder readout. A Simulink compatible block library of MultiQ-3 functions is provided on each laboratory PC. The WinCon software provides a user friendly graphical user interface (GUI) for implementing Simulink-based real-time control on MultiQ-3 DACB. In addition, WinCon can be used to display real-time experimental data on PC. In this experiment, students will learn the basic functionalities of MultiQ-3 DACB, WinCon, and Simulink automated code generation features by implementing a simple loop-back example.

2. Background

In this section, we provide a brief overview of the hardware and software environment to be used throughout this laboratory course.

MultiQ-3 DACB: The MultiQ-3 is a general purpose DACB. It provides 8 single-ended ADCs, 8 DACs, 16 bits of digital inputs, 16 bits of digital outputs, 3 programmable timers, and upto 8 encoder inputs. The MultiQ-3 DACB is accessed through the PC bus and is installed on an ISA bus internal to the laboratory PC. The aforementioned functions of the MultiQ-3 DACB can be accessed via an external terminal board.

Matlab-Simulink-RTW: This is the preferred software environment for the control laboratory. Students enrolled in this laboratory course were familiarized with the Matlab software in ME 322. Simulink is a graphical control-system simulation program. The RTW tool-box enables

automated C code generation from user-designed Simulink control-system diagrams.

WinLib: This is a library of Quanser-supplied DACB drivers (e.g., MultiQ-3) compatible with Simulink (See Figure 1). Some commonly used blocks of MultiQ-3 (MQ3) library are analog input (ADC), analog output (DAC), encoder input, and time-base (See Figure 2).



Figure 1: WinLib Block Library

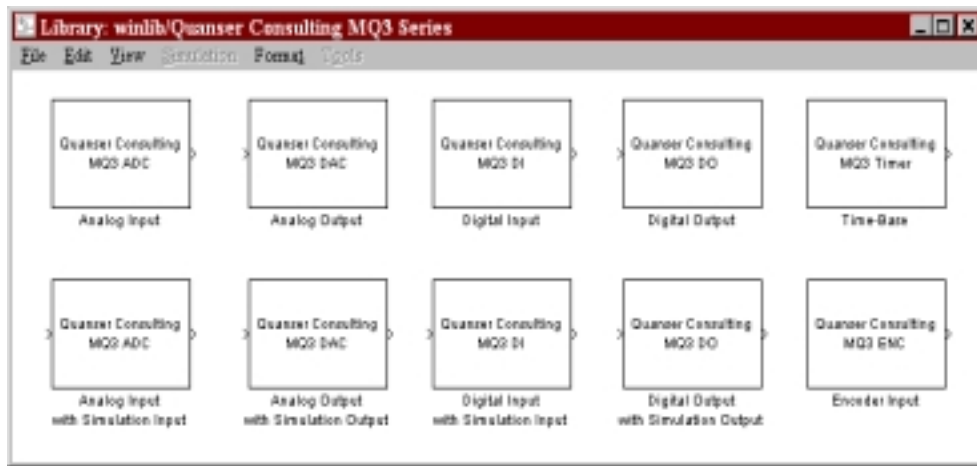


Figure 2: MultiQ-3 Drivers' Block Library

WinCon: The WinCon program interfaces the Simulink generated C code with the MultiQ-3 board in a seamless manner. In addition, it provides useful features for plotting real-time data and for designing GUI-based controls for on-the-fly controller tuning. The WinCon program consists of two principal components, *viz.*, WinCon client and WinCon server. The WinCon client is installed on the host computer with the MultiQ-3 DACB. The WinCon server may be installed on the host or the remote computer. The user designs a Simulink control diagram and generates the C code on the remote computer. The C code from the remote computer is transferred to the host computer

by the WinCon server. The WinCon client and host computer's processor communicate with the MultiQ-3 DACB for real-time data acquisition and control. The WinCon client also relays the real-time data to the WinCon server for plotting purposes.

3. Objective

- i)* Gain familiarity with various functions of the MultiQ-3 board.
- ii)* Learn the laboratory software environment consisting of Matlab, Simulink, RTW, WinLib, and WinCon.
- iii)* Design and implement a simple loop-back control system.

4. Equipment List

- i)* PC with MultiQ-3 DACB and terminal board
- ii)* Software environment: Windows, Matlab, Simulink, RTW, and WinCon
- iii)* Set of leads

5. Experimental Procedure

In this experiment, we will design a controller that outputs a user specified voltage to a selected DAC channel and measures the incoming voltage at a selected ADC channel.

- i)* Using the MultiQ-3 terminal board and a double-ended RCA connector, connect the channel 0 of DAC (analog output) to channel 0 of ADC (analog input), as illustrated in Figure 3.
- ii)* From the **Start** button of the Windows toolbar, select the option sequence **Programs—Matlab—Matlab** to launch the Matlab application.
- iii)* In the Matlab window, at the command prompt, type “Experiment1” and hit the **Enter** key. This Matlab script will change the directory from the default Matlab directory to the working directory for Experiment 1.

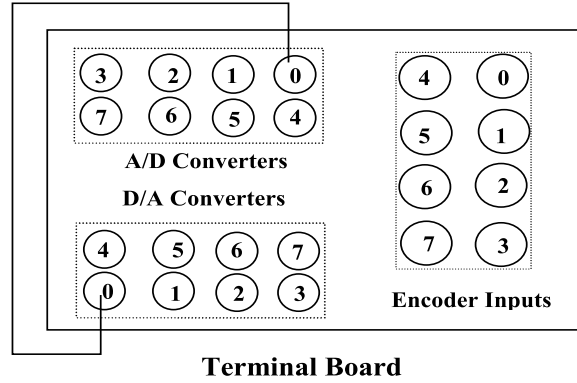


Figure 3: Wiring Diagram for the Loop-Back Experiment

- iv)* In the Matlab window, at the command prompt, type Simulink and hit the **Enter** key. Next, in the Matlab window, type WinLib and hit the **Enter** key. The preceding two commands open the Simulink and the MultiQ-3 DACB drivers libraries, respectively.
- v)* From the Simulink tool bar, select **File–Open** to open “Template.mdl” file. The file “Template.mdl” is a blank Simulink model. This file has been created with a set of RTW options that enable C code generation for Visual C++, RTX (a real-time kernel for Windows NT), and MultiQ-3 environment. You can determine the selected RTW-specific parameters by following the option sequence **Tools–RTW Options**. Please **do not** change any of the parameters while doing this.
- vi)* From the MultiQ-3 series icon in the WinLib library, select and drag the icons labelled ADC analog input, DAC analog output, and Time-Base, into the blank “Template.mdl” model file. In addition, from the Simulink block library, under the icons Sources, Sinks, and Connections, select and drag the icons labelled Constant, Scope, and Terminator, respectively, into the “Template.mdl” model file. Using the copied icons, complete a Simulink block-diagram as shown in Figure 4. Next, set the value of the constant under the icon constant to 1, to output 1 volt at the DAC. In addition, set the channel numbers under the icons ADC and DAC to 0. Finally, save the completed Simulink control-system diagram as “Experiment1.mdl.”
- vii)* From the toolbar of “Experiment1.mdl” file, select the option sequence **Tools–RTW Build** to link, compile, and generate the C++ code for the Simulink diagram. After the

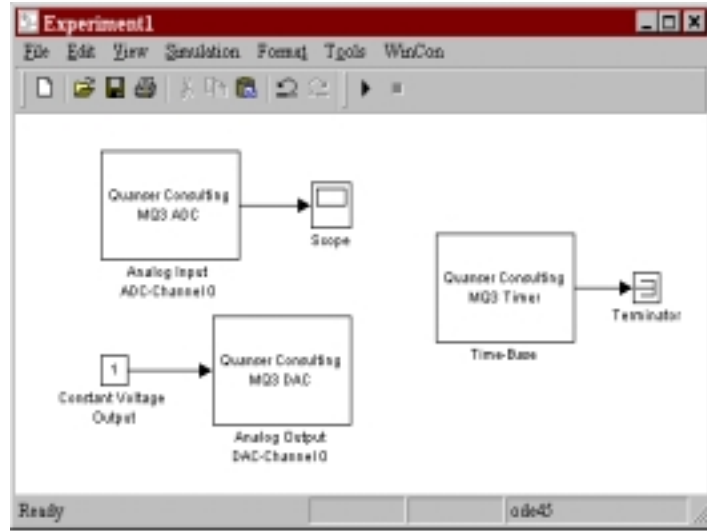


Figure 4: Simulink Block-Diagram for the Loop-Back Experiment

completion of C++ code generation process, WinCon server application is automatically launched.

- viii)* From the toolbar of WinCon Server window, select the option sequence **Plot–New–Digital Meter**. This will launch a digital meter window along with a dialog-box for selecting a variable to display. Select the variable “Scope” from the list of given variables to display the input at the ADC.
- ix)* You can now perform the loop-back experiment. However, before proceeding, you **must** request your laboratory teaching assistant to approve your electrical connections and your Simulink control-system diagram.
- x)* In the WinCon Server window, click the green **Start** button to acquire the real-time data for the loop-back experiment. You can change the output voltage at the DAC by changing the value of constant in the constant icon. Try experimenting, without exceeding the constant value by 5 volts.
- xi)* After sufficient experimentation, press the red **Stop** button in the WinCon Server window to stop execution of your program on the MultiQ-3 DACB.
- xii)* Explore and document various menu options available in the WinCon Server program.

6. Analysis/Assignment

- i)* In step *x)* of Section 5, what is the value of the scope variable, displayed in the digital meter, when you change the constant voltage applied at the DAC from 1 volt to 4 volt? Explain.
- ii)* Based on the loop-back experiment, develop a Simulink control-system diagram to run a diagnostic test on the 8 DAC and 8 ADC channels available on the MultiQ-3 DACB.
- iii)* Briefly explain the principle of operation of ADC and DAC.
- iv)* What is the purpose of the Time-Base driver in the MQ3 block library?

References

1. K. J. Åström and B. Wittenmark *Computer-Controlled Systems: Theory and Design*, Prentice-Hall, Upper Saddle River, NJ, 1997, 3rd Ed.
2. *MultiQ-3 Programming Manual*, Quanser Consulting Inc.
3. *WinCon User's Manual*, Quanser Consulting Inc.