WinCon 3.0.2a

Quanser Consulting Inc.

Realtime Digital Signal Processing and Control
under Windows 95 using SIMULINK and TCP/IP Technology

1.0 Introduction

1.0.1 Principles of Operation

WinCon is a real time Windows 95 application that runs Simulink generated code using Real-Time Workshop on a PC. WinCon consists of a **client** and a **server** referred to as ***WinCon W95Client*** and ***WinCon W95Server***. Each server can communicate with several clients. Each client can communicate with several servers. A PC can have a client as well as a server operating on it at the same time.

1.0.2 WinCon W95Server

The Server component performs the following:
- Converts a Simulink diagram to a PC executable Virtual Device Driver using Real-Time Workshop
- Compiles and links the code using Visual C++
- Downloads the executable code to run on the WinCon W95Client
- Starts and Stops the WinCon client
- Maintains TCP/IP communications with the WinCon W95Client
- Maintains communications with Simulink to perform real-time changes in the WinCon W95Client parameters. i.e. download new gain from Simulink to the WinCon W95Client
- Plots the data streamed from a desired WinCon W95Client in real time
- Saves data to disk

1.0.3 WinCon W95Client

This is the real-time software component that runs the code generated from the Simulink diagram at the sampling rate specified. It performs the following:

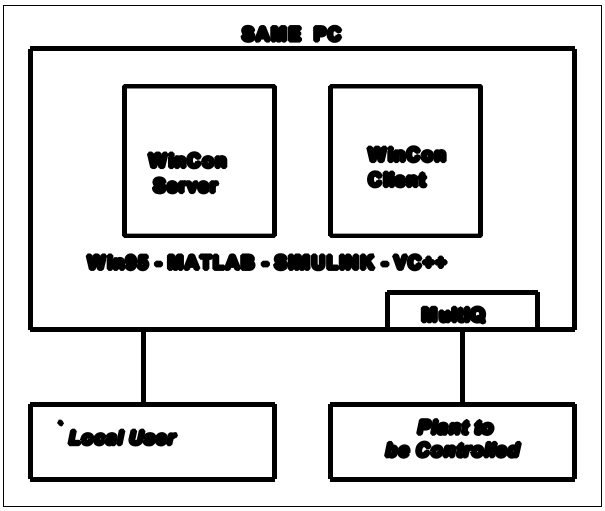
- Receives controller code from the server
- Runs the controller code in real time
- Maintains communications with a W95Server
- Streams realtime data to the W95Server(s) requesting it

The rest of this manual assumes that you are familair with MATLAB and Simulink. A good understanding of networking under Win95 is required when operating in multi-computer Server/Client mode.

1.1 Configurations

1.1.1 One PC - No Network

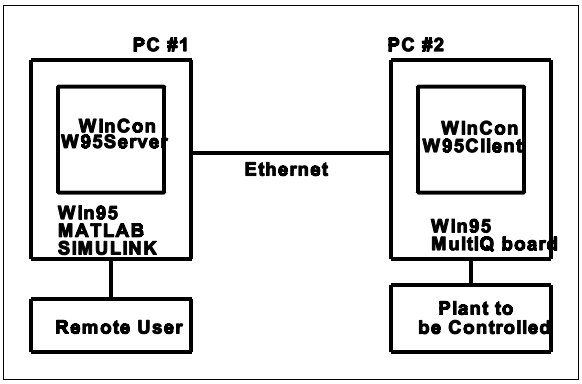
The simplest and most common configuration is a single PC equipped with the required software and hardware. In this configuration, the PC runs both the server and the client and can be used to perform real-time control, tuning and monitoring in the same location. (See Configuration 1)



Configuration 1 - One PC - No Network

1.1.2 Two PCs - Direct Connection

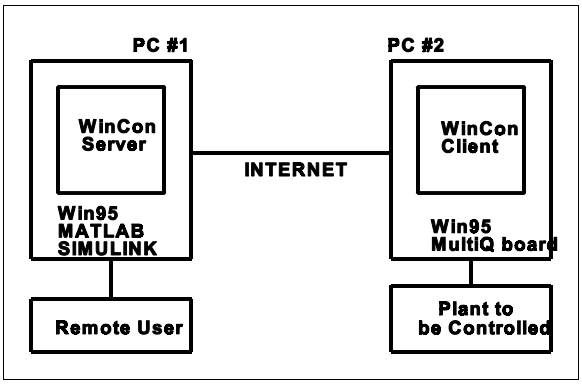
The second configuration consists of two PCs: one running the server and another running the client. In this case the two PCs must be connected to each other directly or via a network. The advantage of this configuration is that the client is running on a PC that is usually not running any software other than Windows 95 and WinCon W95Client. This configuration enables you to obtain the fastest possible sampling rates on the client since the PC is not burdened with other tasks. (See Configuration 2)



Configuration 2 - Two PCs - Direct Connection

1.1.3 Two PCs - Internet connection

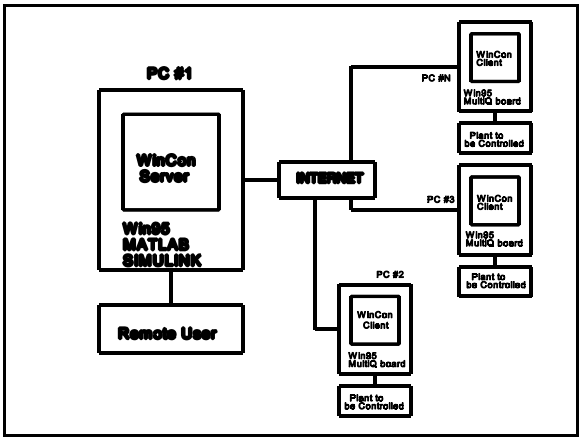
A third configuration is two PCs communicating via The Internet. Each PC is connected to a server and they can each be located anywhere in the world. It is essentially the same as configuration #2 but the connection between the PCs is via The Internet. (See Configuration 3)



Configuration 3 - Two PCs - Internet connection

1.1.4 One W95Server and multiple W95Clients: Internet connection

Another configuration is one W95Server and many W95Clients running as nodes on the Internet. The server can download code to several clients. It can maintain communications with one client at any given time.



Configuration 4 - One W95Server and multiple

1.1.5 Multiple Servers and Multiple Clients - Internet connection

Another possible configuration includes multiple servers and multiple clients running as nodes on a TCP/IP network. For example Sever S1 can download code to Client C1 and other code to client C2. Since it is the first server to connect to these clients, it becomes the ***primary server*** for these clients and

can start, stop, download new code to either client, and change parameters on the fly. Server S2 (from a different PC) can connect to client C1 as a ***secondary server*** and observe data in realtime from it, and start and stop the controller. Server S2 can not download code to client C1, nor can it adjust parameters on the fly. Server S2 can download code to a new client C3 and become the ***primary server*** for client C3.

2.0 System Requirements - Overview

The following table shows the PC requirements for each possible configuration. Note that not all the hardware and software are required on all the PC's being used.

Configuration	**1 PC - No network**		**2 PC's - INTRANET**		**Several PC's - INTERNET**	
PC Running	**WinCon Server**	**WinCon Client**	**WinCon Server**	**WinCon Client**	**WinCon Server**	**WinCon Client**
SOFTWARE REQUIRED						
Windows 95*	✔		✔	✔	✔	✔
WinCon Server	✔		✔		✔	
WinCon Client	✔			✔		✔
Mathworks MATLAB	✔		✔		✔	
Mathworks Simulink	✔		✔		✔	
Mathworks Real-Time Workshop	✔		✔		✔	
Microsoft Visual C++	✔		✔		✔	
HARDWARE REQUIRED						
MultiQ board**	✔			✔		✔
Ethernet card			✔	✔		
Internet connection					✔	✔
NETWORKING REQUIREMENTS						
TCP/IP Protocol Installed	✔		✔		✔	
Microsoft Dial-Up Adapter	✔ **(if no Network card installed)**					
Fully Functional TCP/IP Compliant Network	✔ **(if Network card installed)**		✔			
Fully Functional Internet Connectivity					✔	

The PC requirements are:

- Pentium class processor (the faster the better)
- 16 Meg RAM (the more the better)
- Windows 95 (Without would have been better!)

* The WinCon W95Server will also run with Windows NT version 4.0 however the WinCon W95Client requires Windows 95

** The standard data acquisition board supported by WinCon is the MultiQ board. You may however write your own device drivers as Simulink blocks and link them with WinCon.

2.1 Windows 95 Version

Before you install WinCon 3.0, please ensure the following:

1) Make sure you have the correct version of Windows 95: Click the ***Right mouse button*** on the ***My Computer Icon*** and select ***Properties.*** WinCon requires that certain DirectX components are installed. The required DirectX components are included on Windows 95 versions **4.00.950b or greater.**

*If you have an earlier version you MAY HAVE TO install **Direct X** in order for WinCon to work properly**.***

If you start WinCon and it crashes with a floating point exception (0x07) and you get the famous BLUE SCREEN then you MUST install Direct X 5.0.

Information about DirectX can be found at:
http://www.microsoft.com/directx/

Download the latest version of DirectX (Version 5.0) from:
http://www.microsoft.com/msdownload/directx/dxf/enduser5.0

2.2 MATLAB versions - See section 10.4

To check that you have the correct version of Matlab installed, start up Matlab, and in the command prompt window type in '`ver`'. Press enter. You should see a screen that looks like this:

```
ver
-----------------------------------------------------
MATLAB Version 5.2.0.3084 on PCWIN
MATLAB License Identification Number: ******
-----------------------------------------------------
MATLAB Toolbox       Version 5.2         18-Dec-1997
Real-Time Workshop   Version 2.2.0       01-Jan-1998
Simulink             Version 2.2         21-Nov-1997
```

If you are using Matlab/Simulink, WinCon requires that you have the preceding versions of software from *The Mathworks* installed.

2.3 Visual C++ Professional Version: 5.0 or 5.1 (See section 10.6)

The PC which has Visual C++ and runs the W95 Server must have the environment variables set.

2.3.1 Run ***setenvvar.exe*** found in the **`WinCon3/Bin`** directory and enter the CORRECT paths for all

the components specified.

2.3.2 Also run **mex -setup** in the MATLAB command window to set the parameters of RTW.

2.4 TCP / IP Installation

The WinCon W95 Client and W95 Server communicate with each other via sockets. This requires that you install the TCP/IP protocol on your computer. See the Installation section (section 3) for details.

3.0 Installation

Please ensure that you have the correct software installed on the appropriate PC as covered in section 2.

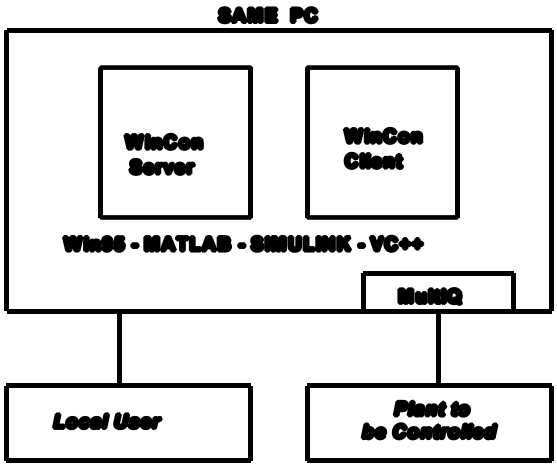
Licencing rules are as follows:

You must acquire and dedicate a license for WinCon for each computer on which the WinCon is used.
Each WinCon package consists of one copy of W95Server and one copy of W95Client. If you want to run configurations that require more than one PC, you will need as many copies of WinCon.

For complete details, please review the End-user licence agreement installed with WinCon.

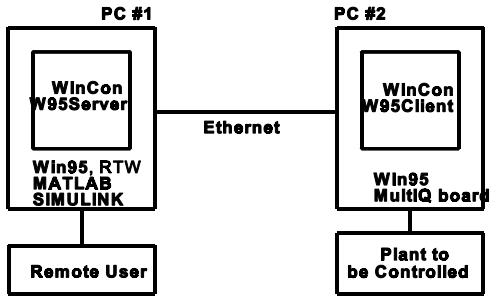
As there are many possible configurations, please follow the installation instructions for your desired configuration. The options are shown in section 1.1

3.1 Configuration 1 - One PC - No Network

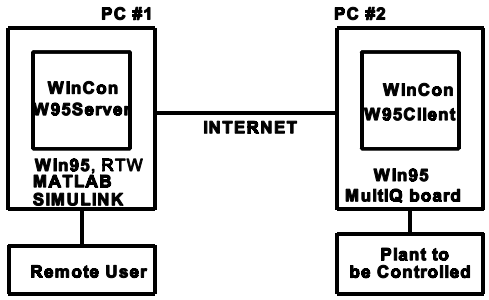
STEP #	ACTION	
1	Ensure that the Operating System is Windows 95. WinCon 3.0 is not compatible with Windows NT.	
2	Ensure that MATLAB 5.2 is installed on the PC before proceeding to the next step. The other components, Real-Time Workshop, Simulink, Microsoft Visual C++ can be installed either before or after W95Server.	
*Check to see that you have two diskettes labeled WinCon W95**Server**. You must install W95Server* **BEFORE** *you install W95Client*		
3	Place WinCon W95**Server** diskette 1 of 2 in the A drive. Click the Start button, and then click Run. In the Open box, type **a:setup**	

Follow the installation instructions, making sure you correctly enter the WinCon W95Server serial number supplied to you. It is on a separate form supplied with your software	
*Once the Server installation is completed, proceed with the installation of W95Client. Check to see that you have two diskettes labelled WinCon W95**Client**.*	
4	Place WinCon W95**Client** diskette 1 of 2 in the A drive. Click the Start button, and then click Run. In the Open box, type **a:setup**
Follow the installation instructions, making sure you correctly enter the WinCon W95Server serial number supplied to you. It is on a separate form supplied with your software as well as on the reverse of the diskette.	
5	Check to see if you have the Microsoft Dial-up Adapter installed. Click the **Start** button, click **Settings** and **Control Panel**. Double-click on the **Network Icon**. With the **Configuration TAB** selected, check to see if **Dial-Up Adapter** is installed. If it is, close the window and proceed to step 7. If not, keep the Configuration window open and proceed to step 6.
6	Click **Add,** select **Adapter**, select **Microsoft** and **Dial-Up Adapter**. Once complete, restart your computer. **DO NOT FORGET TO CONTINUE TO STEP 7 AFTER YOU RESTART.**
7	Check to see if you have the TCP/IP installed. Click the **Start** button, click **Settings** and **Control Panel**. Double-click on the **Network Icon**. With the **Configuration TAB** selected, check to see if **TCP/IP Dial-Up Adapter** is installed. If it is, close the window and proceed to step 9. If not, keep the Configuration window open and proceed to step 8.
8	Click **Add,** select **Protocol**, select **Microsoft** and **TCP/IP**. Setup TCP/IP for your computer. Once complete, restart your computer. **DO NOT FORGET TO CONTINUE TO STEP 9 AFTER YOU RESTART.**
9	If you intend to generate real-time code on this machine, ensure that you have the following components installed before attempting to do so: * Simulink 2.2 * Real-Time Workshop 2.2 * Microsoft Visual C++ 5.0 or 5.1 (Professional or Enterprise Edition) * Ensure that the Data Acquistion card of your choice is installed as specified by the manufacturer * If you are installing a Data Acquisition Card **other than the MultiQ**, you must obtain the appropriate Simulink blocks for them.
10	Set Environment variables by running the SetEnvVars Program. (**Start | Programs | WinCon3 | SetEnvVars**). Fill in the information as it applies to your system then press **OK**.

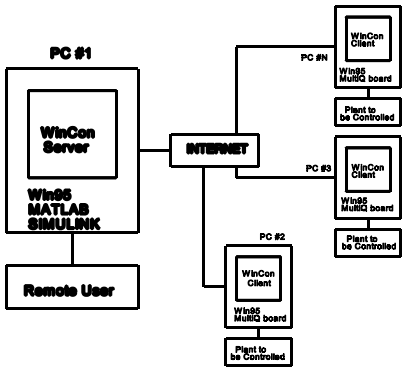
Configuration 2 - Two PCs - Direct Connection

STEP #	ACTION	
1	Prior to installation of **ANY** network configuration of WinCon, you must ensure that the PCs are on a fully functional network with the TCP/IP protocol installed and configured. **If this is not the case, do not proceed to step 2, contact the your Network Administrator or the person(s) responsible for the network and inform them of your requirements.**	
2	Ensure that the Operating System is Windows 95. WinCon 3.0 is not compatible with Windows NT.	
3	**ON PC#1:** Ensure that MATLAB 5.2 is installed on the PC before proceeding to the next step. The other components, Real-Time Workshop, Simulink, Microsoft Visual C++ can be installed on **PC#1**either before or after **W95Server**.	
	*Check to see that you have two diskettes labeled WinCon W95**Server**. You must install W95Server* **BEFORE** *you install W95Client*	
4	**ON PC#1:** Place WinCon W95**Server** diskette 1 of 2 in the A drive. Click the Start button, and then click Run. In the Open box, type **a:setup**	
	Follow the installation instructions, making sure you correctly enter the WinCon W95Server serial number supplied to you. It is on a separate form supplied with your software as well as on the reverse of the diskette.	
5	**ON PC#2:** Place W95**Client** diskette 1 of 2 in the A drive. Click the Start button, and then click Run.In the Open box, type **a:setup**	
	Follow the installation instructions, making sure you correctly enter the WinCon W95Client serial number supplied to you. It is on a separate form supplied with your software.	
6	If you intend to generate real-time code on **PC#1**, ensure that you have the following components installed before attempting to do so: * Simulink 2.2 * Real-Time Workshop 2.2 * Microsoft Visual C++ 5.0 or 5.1 (Professional or Enterprise Edition) Ensure that the Data Acquistion card of your choice is installed as specified by the manufacturer. If you are installing a Data Acquisition Card **other than the MultiQ**, you must obtain the appropriate Simulink blocks for them.	
7	On the PC with the Server Software installed, set Environment variables by running the SetEnvVars Program. (**Start | Programs | WinCon3 | SetEnvVars**). Fill in the information as it applies to your system then press **OK**.	

Configuration 3 - Two PCs - Internet / Intranet / LAN Connection

STEP #	ACTION	
1	Prior to installation of **ANY** network configuration of WinCon, you must ensure that the PCs are on a fully functional network with the TCP/IP protocol installed and configured. **If this is not the case, do not proceed to step 2.** Contact the your Network Administrator or the person(s) responsible for the network and inform them of your requirements.	
2	Identify the Operating System is Windows 95. WinCon 3.0 is not compatible with Windows NT.	
3	**ON PC#1:** Ensure that MATLAB 5.2 is installed on the PC before proceeding to the next step. The other components, Real-Time Workshop, Simulink, Microsoft Visual C++ can be installed on **PC#1**either before or after **W95Server**.	*Configuration 3 - Two PCs - Internet connection*
*Check to see that you have two diskettes labeled WinCon W95**Server**. You must install W95Server* **BEFORE** *you install W95Client*		
4	**ON PC#1:** Place WinCon W95**Server** diskette 1 of 2 in the A drive. Click the Start button, and then click Run. In the Open box, type **a:setup**	
Follow the installation instructions, making sure you correctly enter the WinCon W95Server serial number supplied to you. It is on a separate form supplied with your software as well as on the reverse of the diskette.		
5	**ON PC#2:** Place W95**Client** diskette 1 of 2 in the A drive. Click the Start button, and then click Run.In the Open box, type **a:setup**	
Follow the installation instructions, making sure you correctly enter the WinCon W95Client serial number supplied to you. It is on a separate form supplied with your software.		
6	If you intend to generate real-time code on **PC#1**, ensure that you have the following components installed before attempting to do so: * Simulink 2.2 * Real-Time Workshop 2.2 * Microsoft Visual C++ 5.0 or 5.1 (Professional or Enterprise Edition) Ensure that the Data Acquistion card of your choice is installed as specified by the manufacturer. If you are installing a Data Acquisition Card **other than the MultiQ**, you must obtain the appropriate Simulink blocks for them.	
7	On the PC with the Server Software installed, set Environment variables by running the SetEnvVars Program. (**Start | Programs | WinCon3 | SetEnvVars**). Fill in the information as it applies to your system then press **OK**.	

Configuration 4 - One W95Server and multiple W95Clients: Internet / Intranet or LAN connection

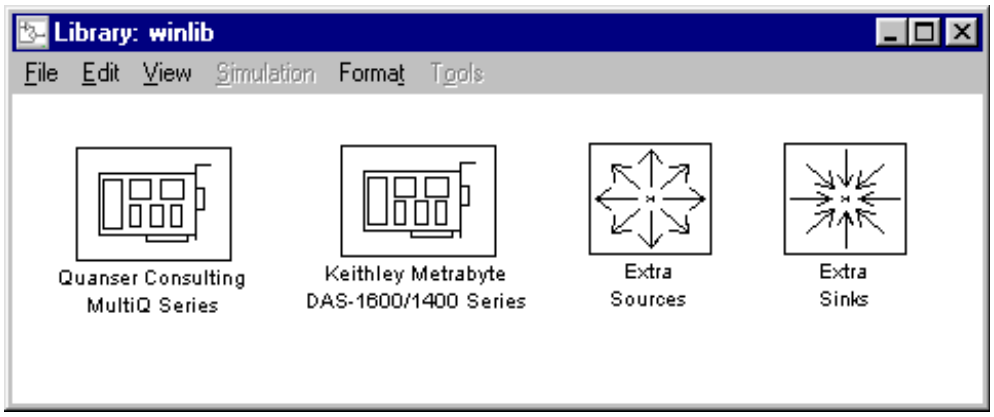
STEP #	ACTION	
1	Prior to installation of **ANY** network configuration of WinCon, you must ensure that the PCs are on a fully functional network with the TCP/IP protocol installed and configured. **If this is not the case, do not proceed to step 2.** Contact the your Network Administrator or the person(s) responsible for the network and inform them of your requirements.	
2	Ensure that the Operating System is Windows 95. WinCon 3.0 is not compatible with Windows NT.	
3	**ON PC#1:** Ensure that MATLAB 5.2 is installed on the PC before proceeding to the next step. The other components, Real-Time Workshop, Simulink, Microsoft Visual C++ can be installed on **PC#1**either before or after **W95Server**.	
*Check to see that you have two diskettes labeled WinCon W95**Server**. You must install W95Server* **BEFORE** *you install W95Client*		
4	**ON PC#1:** Place WinCon W95**Server** diskette 1 of 2 in the A drive. Click the Start button, and then click Run. In the Open box, type **a:setup**	
Follow the installation instructions, making sure you correctly enter the WinCon W95Server serial number supplied to you. It is on a separate form supplied with your software as well as on the reverse of the diskette.		
5	**ON PC#2 through PC#n:** Place W95**Client** diskette 1 of 2 in the A drive. Click the Start button, and then click Run.In the Open box, type **a:setup. Repeat this procedure for each PC, #2 through n. Ensure that you have the correct number of licenced copies for any network installation.**	
Follow the installation instructions, making sure you correctly enter the WinCon W95Client serial number supplied to you. It is on a separate form supplied with your software.		
6	If you intend to generate real-time code on **PC#1**, ensure that you have the following components installed before attempting to do so: * Simulink 2.2 * Real-Time Workshop 2.2 * Microsoft Visual C++ 5.0 or 5.1 (Professional or Enterprise Edition) Ensure that the Data Acquistion card of your choice is installed as specified by the manufacturer. If you are installing a Data Acquisition Card **other than the MultiQ**, you must obtain the appropriate Simulink blocks for them.	

Configuration 5 - Multiple W95Servers and multiple W95Clients: Internet / Intranet or LAN connection

STEP #	ACTION
1	Prior to installation of **ANY** network configuration of WinCon, you must ensure that the PCs are on a fully functional network with the TCP/IP protocol network. **If this is not the case, do not proceed to step 2.** Contact the your Network Administrator or the person(s) responsible for the network and inform them of your requirements.
2	Ensure that the Operating System is Windows 95. WinCon 3.0 is not compatible with Windows NT.
3	**On each PC to act as a Server:** Ensure that MATLAB 5.2 is installed on the PC before proceeding to the next step. The other components, Real-Time Workshop, Simulink, Microsoft Visual C++ can be installed either before or after **W95Server**.
*Check to see that you have two diskettes labeled WinCon W95**Server**. You must install W95Server* **BEFORE** *you install W95Client*	
4	**On the desired W95Server PC(s):** Place WinCon W95**Server** diskette 1 of 2 in the A drive. Click the Start button, and then click Run. In the Open box, type **a:setup**
Follow the installation instructions, making sure you correctly enter the WinCon W95Server serial number supplied to you. It is on a separate form supplied with your software as well as on the reverse of the diskette. Repeat the procedures above for each of the machines that you wish to act a a W95Server. Ensure that you have the correct number of licenced copies of WinCon for any network installation.	
5	**ON each PC that you intend to operate as a W95Client:** Place W95**Client** diskette 1 of 2 in the A drive. Click the Start button, and then click Run. In the Open box, type **a:setup. Repeat this procedure for each PC that you intend to operate as a W95Client. Ensure that you have the correct number of licenced copies of WinCon for any network installation.**
Follow the installation instructions, making sure you correctly enter the WinCon W95Client serial number supplied to you. It is on a separate form supplied with your software. Ensure that you have the correct number of licenced copies of WinCon for any network installation.	
6	If you intend to generate real-time code on any of the PCs with W95Server, ensure that you have the following components installed before attempting to do so: * Simulink 2.2 * Real-Time Workshop 2.2 * Microsoft Visual C++ 5.0 or 5.1 (Professional or Enterprise Edition) Ensure that the Data Acquistion card of your choice is installed as specified by the manufacturer. If you are installing a Data Acquisition Card **other than the MultiQ**, you must obtain the appropriate Simulink blocks for them.
7	On the PC with the Server Software installed, set the environment variables by running the SetEnvVars Program. (**Start | Programs | WinCon3 | SetEnvVars**). Fill in the information as it applies to your system.

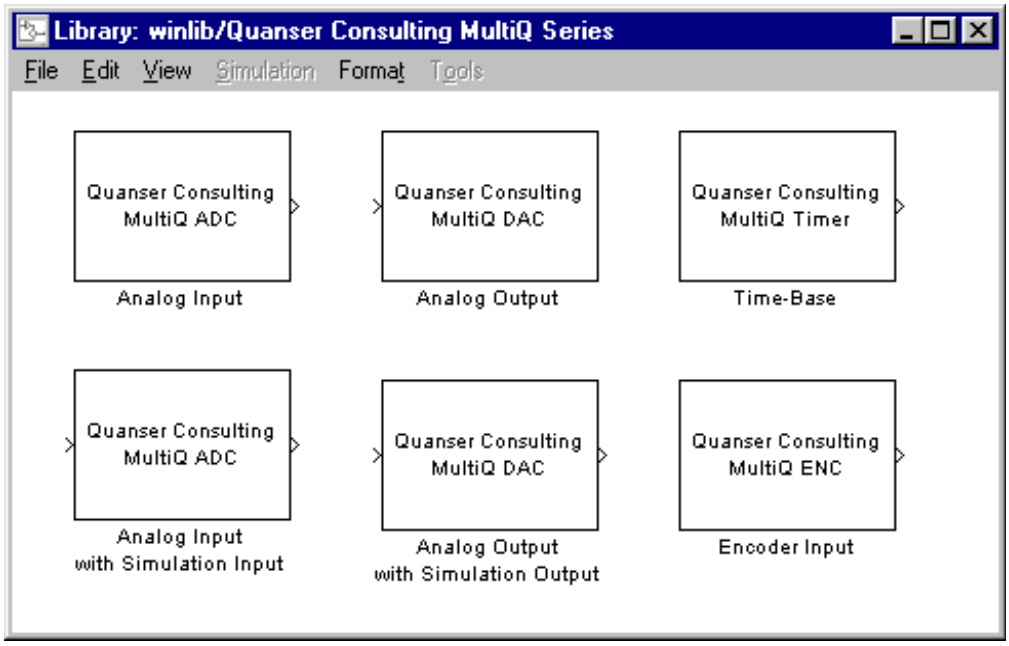
3.1 Integrating the Data Acquisition Board with WinCon

WinCon is fully compatible with the **MultiQ** board. In order to access the **MultiQ** drivers and use them in Simulink diagrams, enter "winlib" in the MATLAB window. This displays the library of board drivers that will function with SIMULINK. The **MultiQ** library offers you the use the following blocks:



3.1.1 Analog input: Select the channel you want to sample from. The output is a value between +5 Volts and -5 Volts. The resolution is 13 bits.

3.1.2 Analog output: Select the channel you want to put the voltage out to. The input to the block is a value between +5 and -5 volts. The resolution is 12 bits.



3.1.3 Encoder input : Select the input channel you want to measure from. The output of the block is an integer value indicating the counts of the encoder. The resolution will depend on the type of encoder you are using.

3.1.4 Digital bit read: The **MultiQ** has an 8 bit digital input port. You can read any of the eight bits (0 to 7) using this block.

3.1.5 Digital bit out: The input to this is a binary value which will be put out to the selected bit on the

digital output port of the **MultiQ**.

If you have another board that you would like to use, you will need to write drivers for it. See the Realtime Workshop User's Guide for details on writing drivers. Quanser Consulting can assist you in writing drivers for a nominal fee. Please contact Quanser Consulting directly.

3.2 Sampling period and IRQ Level

WinCon is a realtime process and as such will require a realtime interrupt source to synchronize it. There are 2 timebase options for WinCon: **System clock** and **Hardware clock**. The system clock uses the PC timer and can provide a maximum sample rate of 1 kHz. A hardware clock uses an independent clock tied to an interrupt line on the PC bus and results in more reliable realtime operation than the system clock.

3.2.1 System Clock

This is not the recommended method of operation.

This method derives an interrupt from the Windows system timer and is limited to a maximum sampling frequency of 1 kHz. Furthermore, the resolution of the timer is 1 msec which results in latencies in the order of 1 msec. This method is not very accurate but may be adequate for slow processes such as chemical plants.

3.2.2 Hardware clock

This is the recommended mode of operation.

In order to use this mode, you need a hardware clock that is connected to an interrupt line on the PC bus. The **MultiQ** board has three such clocks which can be used to generate interrupts. The drivers included with WinCon (in winlib) include a timebase block which controls a **MultiQ** realtime clock. In order to use the hardware clock, each diagram must have a **timebase** block in it. When you include a **MultiQ** timer block in your diagram, ensure that the IRQ level selected in the block is compatible with the jumper configuration on the board. ***The board is configured for IRQ5 timer #1 at the factory. The Simulink timebase block is also set for this . ALSO,*** ensure that no other devices are connected to the IRQ level you are using for the MultiQ. Specifically, **ensure that your modem, network card or sound card is not in conflict with the MultiQ board.** To check this, right click **My Computer**, select **Properties**. Select the device manager tab at the top of the dialog box. Click on the **Computer** icon (at the top of the list) then **Properties**. You should see a list of IRQs. Make sure that the IRQ for the **MultiQ** (IRQ 5 from the factory) is not in use by another device. If it is, you need to resolve this conflict.

The timing diagram in the figure below illustrates the principles of realtime operations. The hardware clock generates interrupts at fixed intervals **Ts** (sampling time). This is the sampling frequency you have selected. Since the processor may be running other operations (Windows, MATLAB etc....) it takes an unknown amount of time to service the interrupt. This duration is termed the latency period $\mathbf{T_L}$(latency time). At the end of the latency period the controller runs for a duration **Tc** (computation time). After the controller finishes running, the processor returns to execute the foreground tasks until another interrupt occurs. The effective sampling period **Te** is the time taken between two services of the interrupt. This time is variable and depends on the latency time. Our tests indicate a minimum latency time of approximately $\boldsymbol{10 < T_L < 20\ \mu secs}$ under Windows95. This value depends on the foreground tasks you are running (the total system load). When running in configurations that have more than 1 PC, with the server on one PC and the client on another, the latency time can be as small as 10 µsecs. The computational delay **Tc** will depend on the complexity of the controller you are running. Note that the scale in the diagram is exaggerated. It is unlikely you will be designing a controller that takes over 50% of the processor time and results in such asymmetry. *One important factor to note is that the latency will generally be relatively constant. This means that if the latency was 20 µsecs on the first sample, it is likely that it will be 20 µsecs at the second sample and thus the sampling period is very close to the desired period. The samples are only delayed by* T_L.

Determining the fastest sampling frequency possible will depend on how much computing power you are willing to relinquish to the foreground tasks (Tf = Te-Tc in the diagram), especially realtime plotting. You should always start with the slowest possible sampling frequency for the process you are controlling. If

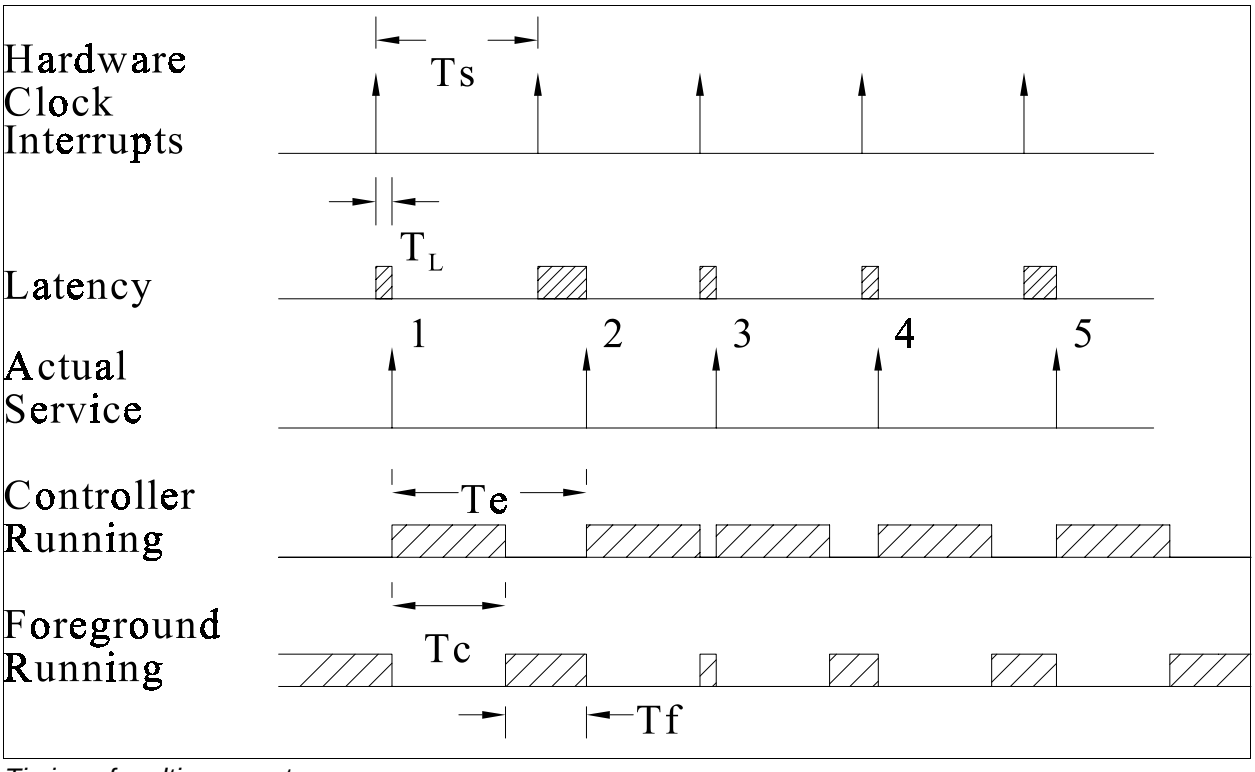
you want faster sampling, increase it slowly and note how much this affects your foreground task. If the sampling frequency is too high, mouse operations and realtime plots will slow down. This should be an indication that the controller is running too fast for the computer you are using.

3.2.3 Threshold Setting

Each W95Client is assigned a threshold (*Threshold*) parameter (**Menu: Control/Threshold** in W95Server Window) that ensures that there is enough foreground time allocated for the operating system to perform its functions. This time is computed at each sample as the time difference between the initiation of the present ISR and the completion of the previous ISR. The client is forced to stop if the foreground time Tf value drops below the specified threshold.

The program causes the client to stop if *Tf < Threshold*

The default value for *Threshold* is 20 μseconds and can be changed from the WinCon W95Server window. This value is rather arbitrary but our tests have shown that a Threshold of 20 μseconds is enough to allow for the foreground to operate normally. *Sometimes, the threshold value will be crossed at startup only and once the program has been put into cache memory it will work fine. In these cases you can start the program and it will run after the first try.* Selecting a *Threshold* value that is too low and a sampling rate that is too fast will crash the system. We recommend the equation Ts = (20e-6+Tc) to be used as your fastest possible sampling period for a given controller.

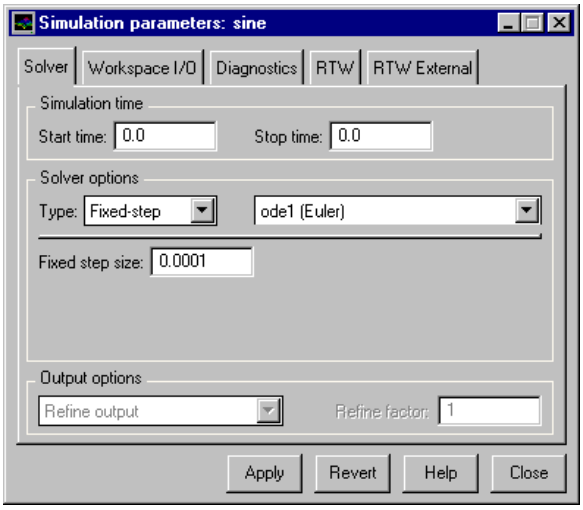


Timing of realtime events

4.0 Generating the code

When you install WinCon on your PC, a blue "WinCon" icon will automatically appear in the bottom right corner of the Windows 95 status bar on the computer with the WinCon Server installed. When you start SIMULINK, a new WinCon Menu item will appear in the SIMULINK window. This menu item will enable you to generate and run the realtime code seamlessly. Simply use SIMULINK to create or load the diagram (SIMULINK model) you want to run in realtime and click on the **WinCon/Build** menu option**.** This initiates the process of generating the realtime code. When the code is generated, a message appears in the MATLAB window indicating that a "wcl" file has been successfully created.

In order to generate the code the following options should be set in the in the SIMULINK diagram. ***Remember to select the sampling frequency you want!***

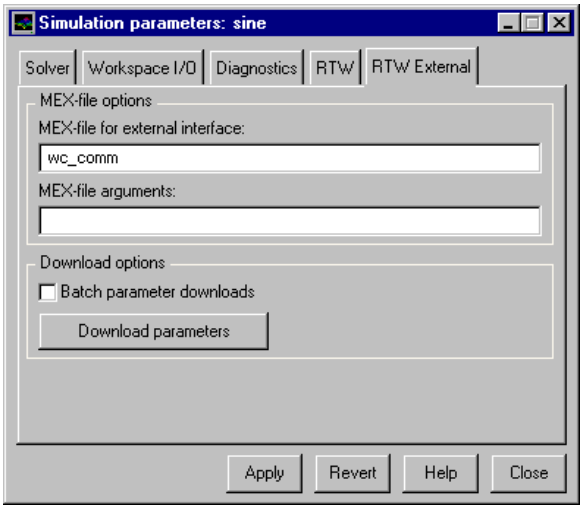


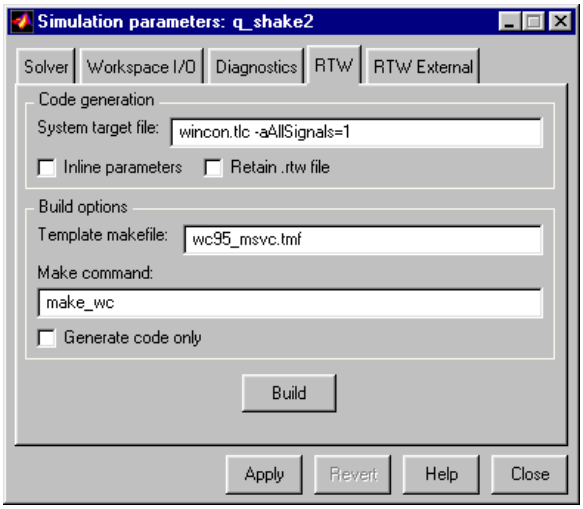
4.1 SIMULATION SOLVER (Step Size) : Select the sampling period **Ts** by entering the value under "Fixed Step Size". The choice of integration method is up to you.

4.2 RTW Options

We have included a Matlab script to set all of the options for generating code for WinCon from Simulink. Type in the following from the Matlab Command Prompt:

`wc_setoptions`





or set the options yourself as follows:

`System target file:`

`wincon.tlc -aAllSignals=1 -aAllParameters=0`

Template Makefile:

`wc95_msvc.tmf`

Make Command :

make_wc

4.3 RTW External Option:

The MEX file used is **wc_comm**

4.4 Potential problems

If you get errors generating the code (e.g. the **.wcl** is not created correctly) the most likely problem is that the paths to either WinCon or the compiler components are not set correctly. Please make sure that you have run the program **(WINCON_DIRECTORY)\bin\setenvar.exe**
(**START | PROGRAMS | WINCON3 | SetEnvVar**) and entered the paths correctly.

If you wish you may confirm that the file **..\matlab\rtw\c\winson\wc95_msvc.tmf** has all the paths and environment variables set correctly, it can be opened using Notepad.

4.5 WinCon Menu in SIMULINK

The "WinCon Link" which is automatically loaded when your computer starts establishes the WinCon Menu item added to the SIMULINK diagram. This gives you access to the following actions:

1) Start/Stop the realtime controller which is associated with this diagram**.**

2) Open Plot: This open a realtime WinCon plot of a desired "Scope" that has been defined in the diagram.

3) New Plot This open the Plot dialogue box which allows you to plot the output of any SIMULINK block in the diagram.

4) Options Set the RTW Realtime options. Same as clicking on [**Tools/RTW**] in SIMULINK**.**

5) Build Compile the realtime controller for this diagram

6) Download the realtime controller to the client to which the server is connected**.**

7) Clean the directory**.** This function deletes all files that were generated from a build and leaves only the original **mdl** file in the directory (the SIMULINK diagram). This function is useful as it forces a build "from scratch". The template makefile (.tmf file) usually does not re-compile functions that have not changed since the last build. Occasionally however these functions may get corrupted and you will need to re-compile everything all over again. In this case, simply click on the "clean" command to delete all the intermediate files and to ensure that you are re-compiling all the necessary files.

5.0 Running the controller

In order to run the controller on a desired client you need to establish a "connection" to it from the W95Server and then download the code that you have generated. You may connect to as many W95Clients as you wish from any one W95Server. Once you are connected to a client, you must download the generated code to that client. Note that you can ***be connected to only one client at any given instant***. In order to download the code, select "**Model/Download**" from the W95Server menu or "**WinCon/Download**" menu option in SIMULINK. This will download the generated code to the client you are presently connected to.

5.1 Connecting to the client

The correct method to connect to the client depends upon which client/server configuration you have installed.

5.1.1 Configuration 1 : NO NETWORK

The simplest configuration is to run the server and client on the same PC. Select the "**Client/Connect**" menu option from the server. Type "**localhost**" for the name of the remote PC. Click OK. The WinCon Client should start up automatically if it is not running already.

5.1.2 Configuration 2 : INTRANET

You need to establish a TCP/IP connection between the two PC's. If a DNS server is installed, you can refer to the client PC by domain name (ie fred.jones.com). Otherwise you will need to know the IP address of the PC running the WinCon Client as well as ensuring that the 2 PC's are physically connected. You can obtain the WinCon Client PC's IP address by running the utility "***winipcfg***" included with Windows on the client PC. Run the W95Client program on the client PC. If you want the client to start automatically as the PC boots, add it to the **(Windows95 Directory)\Start Menu\Programs\StartUp** folder. If the client PC is accessible, connect to it using the WinCon W95Server "**Client/Connect**" Menu option. Type in the IP address (or DNS name) of the remote client PC and click OK.

5.1.3 Configuration 3: INTERNET

In order to download the code to a client who is connected to the Internet you need to have the client's IP address or DNS name. You can obtain this by running the utility "***winipcfg***" included with Windows directory on the client PC. Once you know the IP address of the client, you can then connect to that client using the"**Client/Connect**" menu option from W95Server. Type in the IP address (or DNS name) of the client for the remote PC and click OK.

If it is not possible to connect to the client, the W95Server program on the server PC will appear locked up while it tries to connect. After 30 seconds or so, the Server will respond with a message box indicating that it can not connect to the remote client. The odds are that you do not have the correct IP address for the client PC (or the client PC is down). It may also be possible that the connection is not available or the connection has been dropped (this may happen if the client is connected via modem and there has been no internet activity for a long time).

5.2 Dowloading the code

When you have selected the client you wish to connect to, you download the code by selecting "**Model/Download**" from the server.

5.3 Running the client

You can start and stop the controller on the client to which you are connected using any of the following methods:

- Model/Start or Model/Stop from the Server Menu
- Start / Stop button from the Server Window
- Start / Stop button in SIMULINK WinCon Toolbar
- **Alt-Pause** to start and **Pause** to stop from the keyboard while in any Window.

5.3.1 Potential problems

5.3.1.1 You may encounter the message: **The Model cannot be started. It may be necessary to generate and download the code.** If this happens, make sure that there are no conflicts on the bus between the interrupt and other devices such as a sound card (See section 3).

5.3.1.2 You may encounter the message: **The controller was running too fast**. Try setting the Threshold (**Client/Threshold)** to 1e-6 seconds and start the controller again. If the message appears again then the controller is too complicated and can not run at the sampling rate you have selected. Try a slower sampling rate. Make sure the drivers for the data acquisition boards are not timing out.

5.4 Changing parameters online

Any parameter that you can change in a SIMULINK diagram can be changed in the controller while it is running. Simply change the parameter as you would using SIMULINK. In order to obtain this feature, you must have the "***external***" feature turned on in the SIMULINK diagram (menu option **Simulation/External** should have a check mark) and the **RTW-External MEX-file** should be set to "***wc_comm***"

6.0 Plotting and saving data

Outputs of all SIMULINK blocks are available for plotting and saving. You do not need to define any scope blocks or "sinks" in order to plot or save data. If you have defined Scopes, they are simply faster to access. The can be plotted using the **Plot / Open** Menu of the Server or the Scope Toolbar on the SIMULINK diagram. You may then select to display a desired variable as a realtime [x vs time] plot, a digital meter (looks like a digital voltmeter), a thermometer, or an xy [x vs y] plot.

In addition to the different display types, there are several different methods you can use to collect the data. **Realtime plots** collect decimated data and display it in near-real time (real-time with communication, buffering, and display delays). **Fixed-mode plots** display all data points in a specified interval. These are not updated in realtime but rather are supplied for a given time period.

6.1 Realtime plots

In order to plot the output of any block select **Plot/Variable** from the Server Menu. You may change plot buffers, axes, scales etc. from the appropriate menu. You can select more than 1 variable for a single plot. ***Note that the realtime plots are self-decimating:*** data is automatically displayed at a rate that will minimize the number of pixels being overdrawn. This is done deliberately in order to reduce communications bandwidth and to maximize the number of plots that can be opened at any instant. This self-decimation may result in "**plot aliasing**" where a variable is changing much faster in the realtime process than what is shown in the realtime plot. This can be deceptive and lead you to think that the model is incorrectly specified. You must be careful in selecting the time scale of the data so that the plots are not misinterpreted.

Data being saved however is not decimated. There is no risk of losing data when it is being saved.

The following options can be set for each plot

Update/ Real-time: Data will be drawn across the plot window for the duration of the buffer (5 seconds is the default). At the end of a trace the plot returns to T=0 and it draws the new data as it arrives.

Update/Freeze plot: This option stops the plot and allows you to examine the data more closely. If you select **Update/Real-time** after **Update/Freeze Plot**, the latest real time data from the client will be displayed, not the data immediately following the freeze.

Update/Freeze all plots: This option makes all plots freeze at the same time.

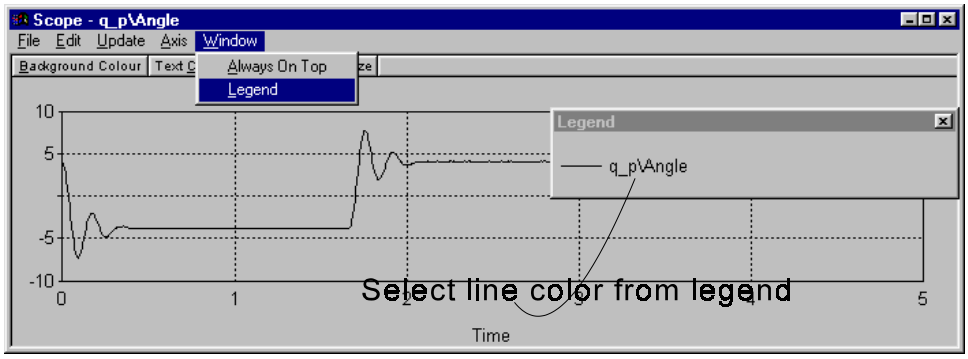
Update/Buffer: Duration of time for which data is collected for each plot between refreshes. The default is 5 seconds.

Update/Frequency: The frequency at which data is sampled for the plot. For example, if the controller is sampling at 10 kHz, there is no need to display the data at that frequency. It would also be impossible to view all the data points on screen at that rate. The plots auto select the decimation frequency of the display. Data being saved is not decimated.

Axis/Time
Time: Duration of time which is displayed on the plot. This cannot exceed the Buffer duration. In fixed mode, if you enter a time that is less than the Buffer time, a scroll bar appears. This allows you to examine the data more closely.

Window/Always on top This option will ensure that the plot window is not being covered by the Windows of other applications.

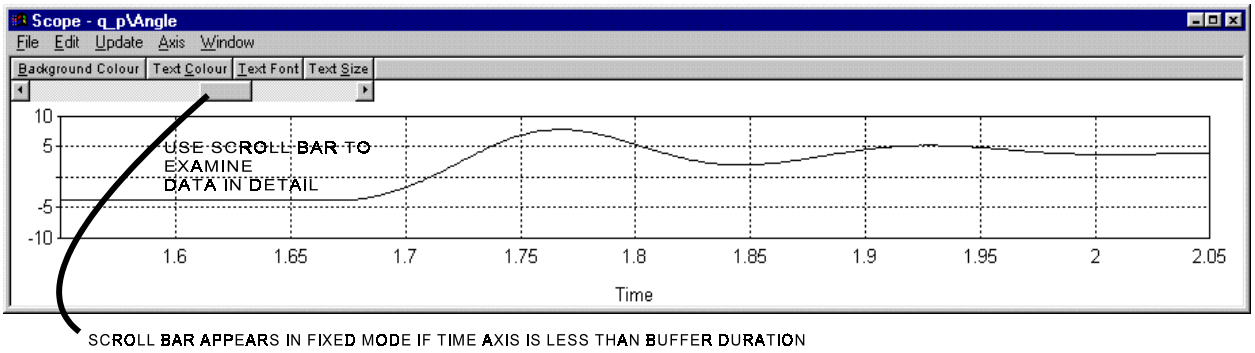
Window/legend
Opens a window with legend info. You can change line colors by clicking on the legend name.



6.2 Saving Data

Clicking on the **File/Save** menu option of any plot will save the data associated with that plot. You can save data in one of two formats:

- m file: will create a **.m** file of the desired name. Entering the name in MATLAB will generate a MATLAB plot of the saved data and bring the variables into the workspace under the array named **plot_data**.
- mat file: will create a **.mat** file. You load the file into the MATLAB workspace using the MATLAB load command. The variable names are then the same as the block outputs you had selected.
- To Workspace : is equivalent to saving to a mat file and then loading it into the workspace.



6.3 Fixed mode

You may switch a plot to fixed mode which means that the plot will not clear itself at the end of trace. This feature allows you to examine the data closely. When you switch to fixed mode all the data for the last N seconds will be collected from the server and displayed on the plot. This data will not be decimated (every data point collected in realtime from the client will be drawn). You can change the timescale on the plot. A timescale smaller than the buffer length will result in a closeup of that time period. You may scroll within that time frame using the scroll bar on the top of the graph.

6.4 Digital Meter

Selecting **Plot/New/Digital Meter** allows you to display a variable as a digital meter independent of time. You may select the accuracy of the display using the **Number** menu item. You can also **displace** (**←→**) the Decimal point by clicking on it and and dragging it left or right. This display is handy for monitoring variables which change at a slow rate. You may also remove the Menu from the display by selecting **Window/Hide Menu.** A **right mouse button** click gives you access to the other options one of which returns the Menu bar to the digital display.

6.5 XY Plots
You may also plot two variables against each other in realtime. You select the X axis variable and the Y axis variable. You may also select whether the plot erases itself at the end of a buffer similar to the realtime plots or leaves a trail as is done with fixed mode. Plot aliasing can be a problem with XY plots.

6.6 Thermometer
A thermometer is a bar-graph like display of a single variable. You can set the minimum and maximum values for each thermometer. It is useful for variables that change at a slow rate.

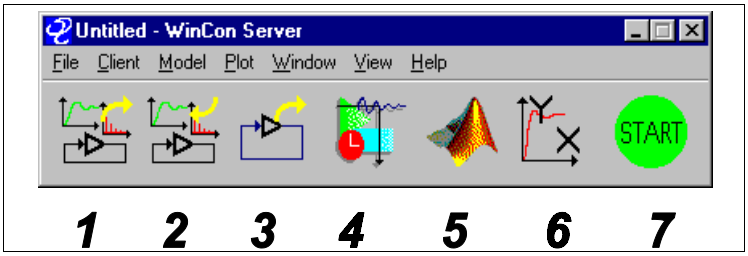
6.5 Performance variables
Each controller can be monitored for performance by plotting the "performance" variables supplied. The most important of these is the "Sampling interval" which shows the actual interval achieved between samples (Te). This should be as close as possible to the desired sampling interval. If this variable diverges significantly from the desired sampling rate, then the controller performance will not be as expected. Using the MultiQ as a time base block results in sampling intervals which diverge by at most 20 microseconds from the desired sampling interval.

Another important parameter is the computation delay (Tc) which can also be monitored. This shows you how long it takes to execute the controller per sample. Knowing Tc lets you compute the fastest possible sampling frequency for that controller. If you determine that you would like to keep Tf seconds for foreground tasks and you know that the computation delay is Tc, then the fastest sampling rate you should select should be Fmax = 1/(Tf+Tc).

7.0 Navigating through the Windows

7.1 WinCon Server

The WinCon Server main window is shown on the right. It has the following menu options



WinCon Server Main Toolbar Window

- File Menu - Operation on *.wcp files.
- Client Menu - Connect and disconnect from a client.
- Model Menu - Open a SIMULINK model (.mdl) or a WinCon Controller (.wcl)
- Plot menu - Plot data - see section 6
- Window Menu -Select whether the WinCon window is "always on top" of all other windows. Navigate to other windows related to WinCon.
- View Menu - View Toolbar or not. Change Toolbar size.
- Help Menu

7.1.1 Toolbar Buttons

The WinCon Toolbar offers shortcuts to the following:
1) Open a project - opens a **.wcp** file
2) Save Project - saves the running controller as a **.wcp** file
3) Open a Model - opens a **.mdl** or **.wcl** file
4) To SIMULINK - takes you to the SIMULINK diagram associated with the selected controller.
5) To MATLAB: Will take you to the MATLAB main window.
6) Plot existing Scopes in the diagram
7) Start / Stop the controller to which the server is connected.

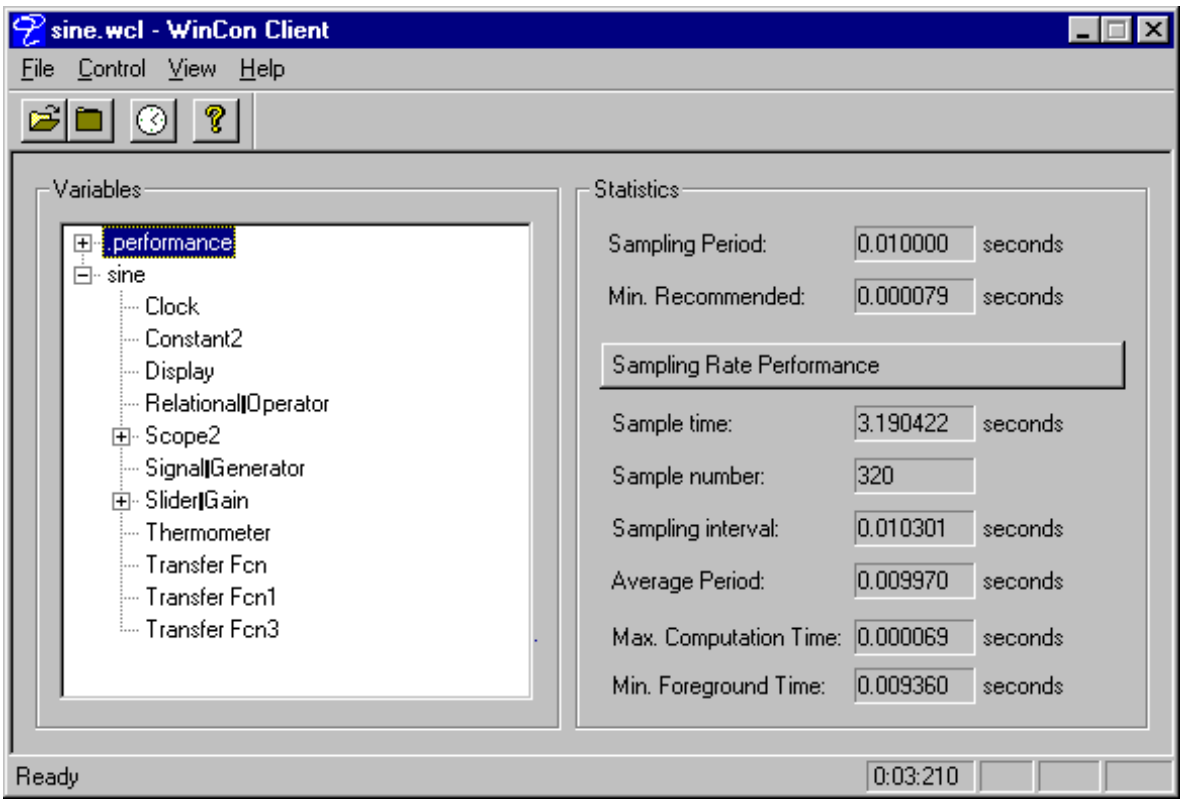
7.1.2 WinCon Projects

A WinCon project file (**wcp** extension) can be created by saving a session of the server. The session can contain several connections, plots and controllers. If you re-load a project, all connections are re-established, controller downloaded and appropriate plots opened. **In order to successfully open a complex project (multi client) make sure that all the required clients are up and running.**

7.2 WinCon Client

The WinCon Client window is shown in the figure below. This window is usually minimized as you do not normally need to interact with it. In case you want to ensure that the client is running, you can maximize this window and observe its content. It displays some of the performance variables discussed above in realtime and allows you to change the threshold parameter that ensures the sufficiency of processor time for foreground tasks. You may select **Client/Close upon Exit from the server** which allows the server to shut down the client upon exit. This is the default value which you may want to change if you are running a remote client which you do not want to terminate when you exit from the server.

For a detailed description of the contents of the Client window see section 11.


sine.wcl - WinCon Client
File Control View Help
Variables
.performance
sine
Clock
Constant2
Display
RelationalOperator
Scope2
SignalGenerator
SliderGain
Thermometer
Transfer Fcn
Transfer Fcn1
Transfer Fcn3
Statistics
Sampling Period: 0.010000 seconds
Min. Recommended: 0.000079 seconds
Sampling Rate Performance
Sample time: 3.190422 seconds
Sample number: 320
Sampling interval: 0.010301 seconds
Average Period: 0.009970 seconds
Max. Computation Time: 0.000069 seconds
Min. Foreground Time: 0.009360 seconds
Ready
0:03:210

8.0 Examples

8.1 Loopback Example

This example is a tutorial that everyone should try. Not only do you practice the use of WinCon but you also test your data acquisition board and learn how all of the components of WinCon and SIMULINK interact. The example is a loopback system. You need to wire the output of channel # 0 on the D/A subsystem of your board to the input of channel #0 on the A/D subsystem on your board. This is shown in Figure 8.1.1

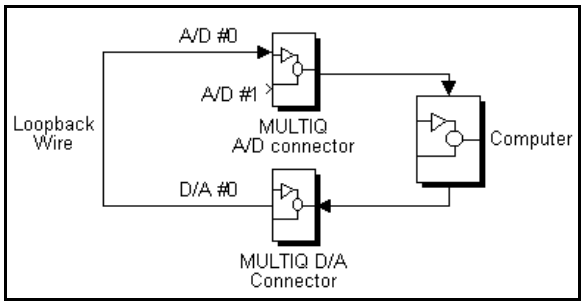
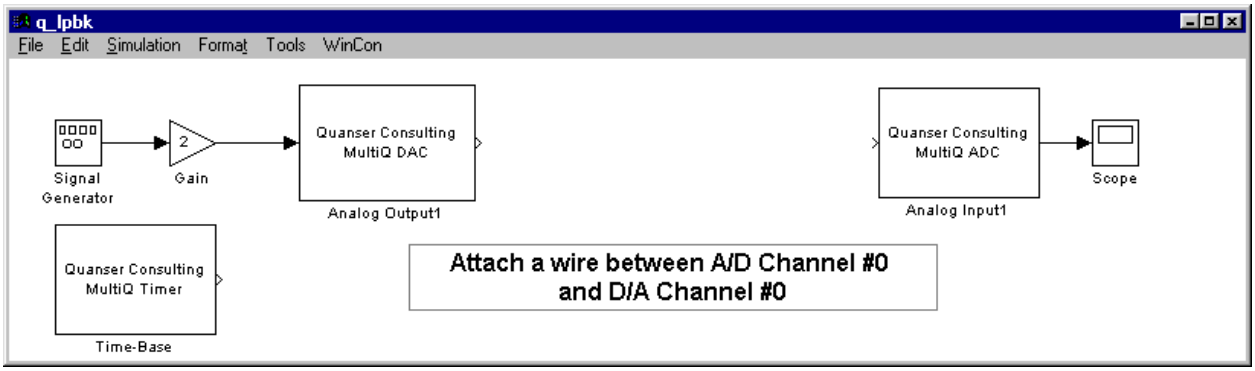


Figure 8.1.1 Wiring for loopback experiment

The "controller" will simply apply a sine wave to the D/A output and will measure the voltage from the A/D. No other hardware is required for this example.



1) Start MATLAB for windows on the PC with the W95Server installed and type `cd c:\wincon3\examples` Type `q_lpbk` This loads the diagram shown in Figure 8.1.2 into SIMULINK. You could have generated this diagram yourself using SIMULINK.

2) If you have a MultiQ board you can skip to step 4

3) If you do not have a MultiQ board installed in your system you will need to replace the data acquisition blocks and the timer block with the appropriate blocks for your board. See the Realtime workshop manual regarding writing drivers for your boards.

4) ***If you are running the client on a different PC than the server, please follow the instructions in section 9 before you proceed.***

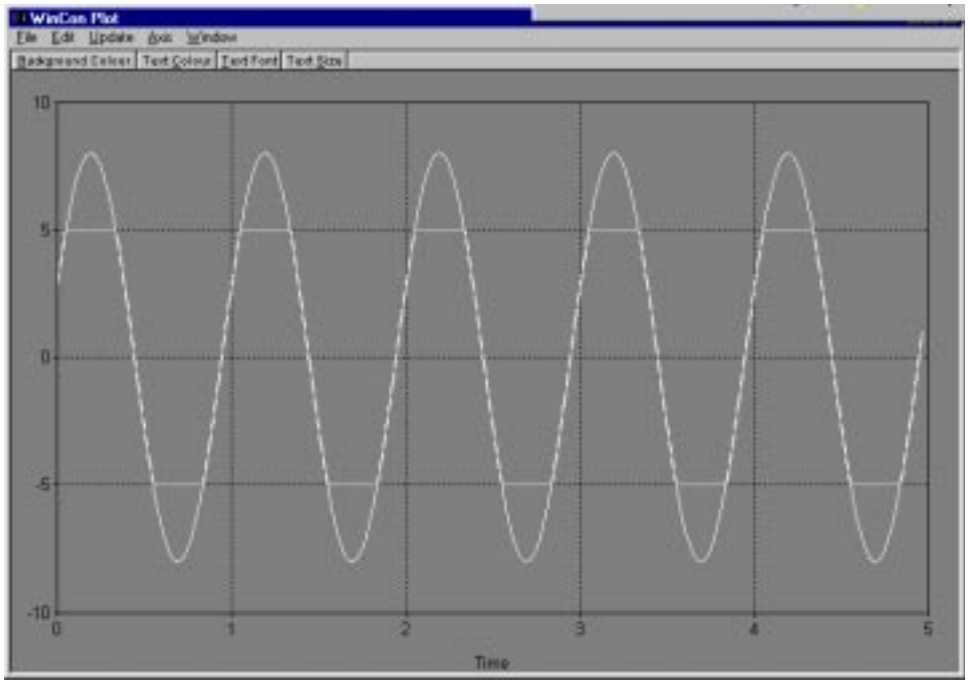
5) Select WinCon from the SIMULINK menu and click on Build. This generates the realtime code for the diagram. Wait until the compilation is complete. The MATLAB window displays the progress of the code generation task and when it is complete the following message appears:

```
### Successful completion of RTW build procedure for model : q_lpbk
```

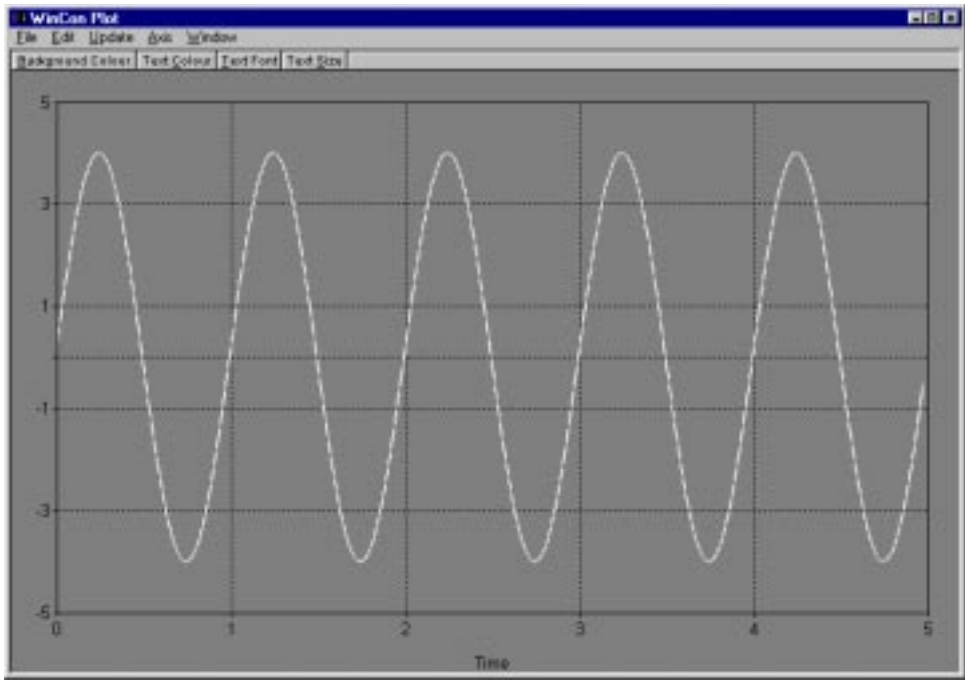
Following the code generation, WinCon Sever and WinCon client are automatically started. The generated code is then automatically downloaded to the client on the local machine and the system is ready to run.

6) Once you have downloaded the controller to the client, click on WinCon/Start from the SIMULINK window. This starts the controller at the sampling rate specified. In this case it is set to 200 Hz as determined by the RTW/Solver options
window.

7) Click on **Plot** or the **Plot button** in WinCon Server and select New/Scope. The names of all block in the diagram appear in a Multi Select Variable Tree. You can then select the variables you want plotted. In this case select Gain and Scope. The output of Gain is the output of the signal generator. The output of Scope is the measured voltage using A/D Channel 0. Select OK. This displays a clipped sine wave and a full sine wave as shown above. The clipped sine wave is the measured voltage from the A/D. The clipped sine wave is the output of the gain block. The Gain block is putting out a sine wave of amplitude 8 volts but the D/A output has a range of +/- 5 volts. For this reason, the measured signal has an amplitude of only 5 volts.



8) Let the plot keep drawing and click **To SIMULINK** from WinConServer. Click on the gain block and change it to 4. Note the changes in the plots. Now the gain output is not clipped and the input and output match in voltage levels. This shows that the output of the D/A is being measured by the A/D block. ***Note also that you were able to change a gain in realtime!*** Stop the controller and change the Signal generator to a sawtooth function. Start the controller



again. This does not work since changing the signal generator function is equivalent to a change in the SIMULINK diagram, which requires changing the realtime code. If you want to change the signal you need to re-compile. Change the signal back to a sine wave and start the controller. The 4 volt sine wave signals should return.

9) Stop the controller. ***Save*** wc_lpbk.mdl from the SIMULINK window with the amplitude set to 4 volts. Close all other applications without saving anything.

10) Ensure that WinCOn W95Server is closed. Then start WinCon Server. Select **Model/Open** and load `c:\wincon3\examples\q_lpbk.wcl`. Note that **File/Open** is used for WinCon Project files (*.wcp) and **not WinCon controller files** (*.wcl). This is the code that was generated in step 4. Click Start. This starts WinCon client, downloads the code to it and starts the controller. Open the plots as in section 6. What are the plots showing?
Clipped or unclipped waves?
It *will be clipped* because the compiled code in step 4 was using an amplitude of 8 volts. Stop the controller using the Pause button. Click To SIMULINK. This starts MATLAB and loads `q_lpbk.mdl` diagram into SIMULINK. Click on WinCon Start. Switch to the plot Window.
What do the plots show now?
An unclipped sine wave at 4 volts because you just loaded the parameters from the SIMULINK diagram into the running controller! *This is where you have to be cautious.* ***WinCon runs generated in realtime code that can only be changed via SIMULINK or re-compiling.***

8.2 PID Controller

This example demonstrates a PID controller that positions the output of the Servo plant shown in figure 8.2.1.

Start WinCon W95Server and load the controller \wincon3\examples\q_pid.wcl **(Model/Load)**. Click **To SIMULINK** , this brings up the diagram shown below.

Figure 8.2.1 SRV02 Rotary position servo plant

The controlled output of the Servo Plant is measured using the A/D channel #0. The command to the system is generated from the Command Generation block (in this example, the Signal Generator Block is the Command Generation Block). The difference between the command angle and the actual angle is the error signal which is multiplied by a proportional gain Kp. The error is also integrated and multiplied by an integral gain Ki. The measured variable (eg motor angle) is also differentiated using a high pass filter and multiplied by a gain Kd. The three gains are the PID gains of the controller. The sum of these terms is applied to the D/A output channel #0. The signal from the Signal Generator could be anything you want. In this example it is a signal generator applying a square wave of +/-10 degrees. You may replace this block with another A/D channel for example which has a potentiometer attached to it. The controlled system will track the signal generated by the command generation block. The PID gains in the Figure could be constants or variables defined in the MATLAB Command Window. If you enter variable names in the PID gains (or in any other block for that matter) they must have a value assigned to them before you can generate the realtime code. Once you generate the code, the controller (.wcl file extension) will have these gain values hard coded into it. Changing the values of the variables in the MATLAB command window will not immediately change the value in a running controller. You will have to either stop and start the simulation or, open the dialog box for the SIMULINK block which uses the variable name in it and then click [OK] or [APPLY]. This can be done while the simulation is running.

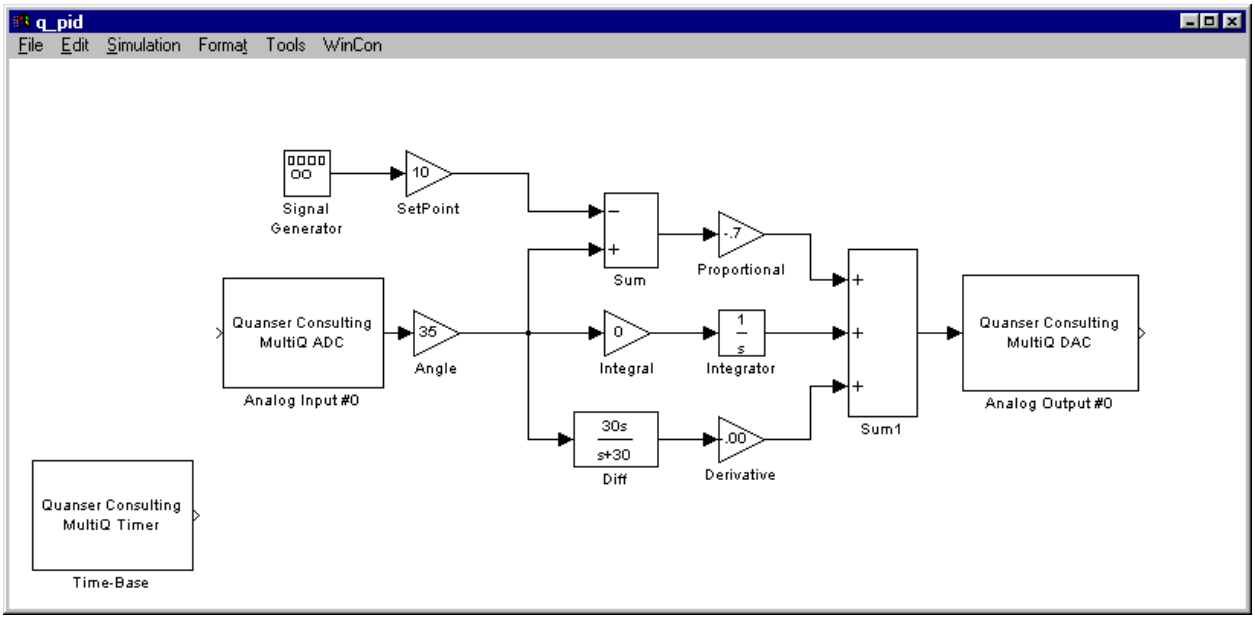


Figure 4.2.2 SIMULINK controller of position servo (PID)

Figure 8.2.3 Change in step response proportional gain is increased.

The plot shown above shows the difference in response when the proportional gain is changed in realtime from 0.7 to 1.0.

8.3 Self Erecting Inverted Pendulum

This example covers the self erecting inverted pendulum system shown in Figure 8.3.1. It consists of a motor driven cart which is equipped with two quadrature encoders. One encoder measures the position of the cart via a pinion which meshes with the track. The other encoder measures the angle of the pendulum which is free to swing in front of the cart. The implemented controller starts with the pendulum in the "down" position, swings it up and maintains it upright. The controller is shown in Figure 8.3.4

The system is implemented in SIMULINK with the controller q_ip2_se.mdl and run in realtime using WinCon.

The subsystems of the controller are the following:

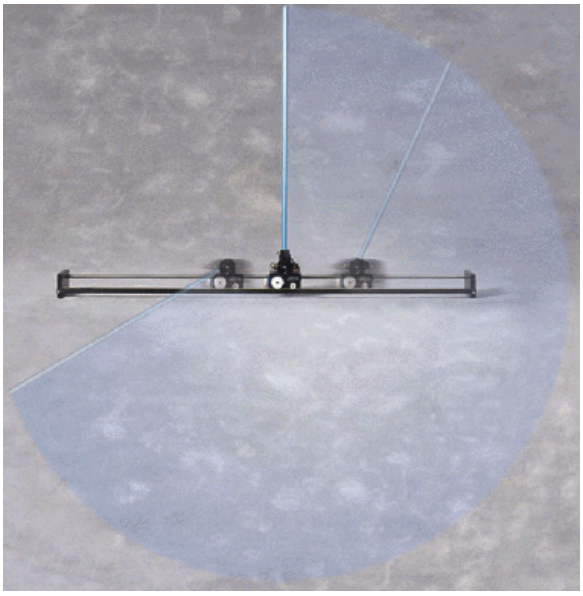


Figure 8.3.1: Self erecting inverted pendulum

8.3.1 Calibration and differentiation

This block simply converts the count value measured from the encoder channels on the MultiQ board to the appropriate units. The values are also fed through high pass filters which essentially differentiate the signals.

8.3.2 Pendulum Full/Up/Down measurements

We define three frames of reference
This block converts the measurement of the pendulum angle from the "Full" position to two other frames of reference shown in figure 8.3.2 and 8.3.3.

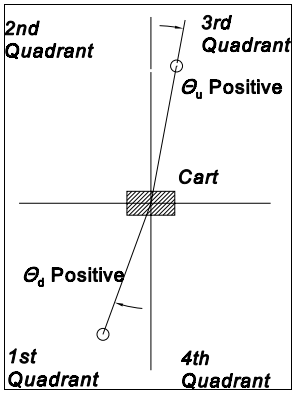


Figure 8.3.2 Up and down positions and their directions

8.3.2.1 θf (FULL) : The FULL position measures the absolute angle of the pendulum from its initial position hanging down. The value start at zero (hanging down) and increases in the positive direction when the pendulum rotates clockwise and decreases when the pendulum rotates counterclockwise. The value of the angle measured can exceed 360 degrees many times over in both directions. This value is obtained directly from the encoder counter.

8.3.2.2 θu (UP) : The UP position is relative to the vertical axis in the up position. It gives a zero value when the pendulum is straight up and a positive value when the pendulum is tilting to the right of the vertical and a negative value when it is to the left of the vertical. It is obtained by performing the following operation:

$$\theta_u = -\sin^{-1}(\sin(\theta_f)) \approx -\sin(\theta_f)$$

8.3.2.3 θd (DOWN) : The DOWN position is essentially the same as the FULL position expect that it is limited to values between ±180 degrees (*NOT SATURATED*!). This is the value used in the "swing-up"controller since the feedback is relative to the down position. If we feed back the FULL value and the pendulum does a full swing then swing up control will be ineffective! The value θd is obtained using the following equation:

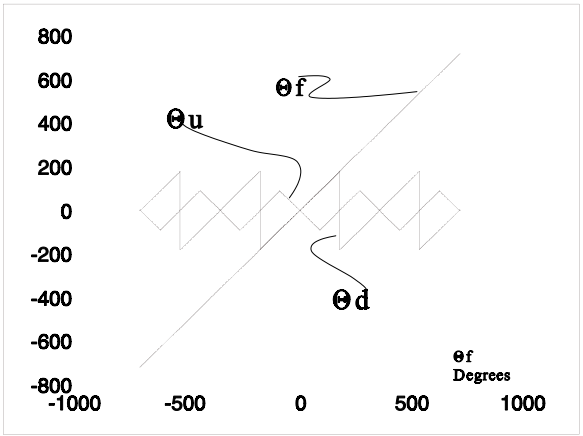


Figure 8.3.3 : Up and down angles

$$\theta_d = tan2^{-1}\ (\ \sin(\theta_f)\ ,\ \cos(\theta_f)\ )$$

(note $tan2^{-1}$ returns values between $\pm\pi$)

Note that all trigonometric operations are performed on the angles converted to radians. The results are then converted back to degrees.

8.3.3 Swing up control

This is the “positive feedback loop” used to swing the pendulum up. It consists of two loops. The inner loop performs position control of the cart :

$$V_d = K_p\ (x - x_d) + K_d\ \dot{x}$$

The outer loop of the system generates the command x_d via the following equation:

$$x_d = P\ \theta_d + D\ \dot{\theta}_d$$

with P = 0.5 cm/deg and D = .004 cm / (deg/sec). These two values are crucial in bringing up the pendulum smoothly. You can tune the value of D to adjust the “damping” in the system.

Note also that x_d is limited to ±3 cm. For the first 2.75 (T_{high}) seconds of control however, x_d is multiplied by (1+Kh) resulting in a “high gain” system for a short time. This effectively speeds up the initial response to get maximum swing in the shortest possible time. Ideally, you want to move the cart back and forth a minimum number of times and bring the pendulum up quickly. If you do not have a high gain period, the system will still come up but it may take up to 20 oscillations of the cart to bring the pendulum up.

8.3.4 Mode control

This block determines which of the two voltages should be fed to the motor. Initially, at startup, we want to feed the voltage computed by the swing up controller. When all of the conditions listed below are met, we want to switch to the “stabilizing” controller output voltage.

Stabilizing conditions:

$$|\theta_u| < \theta_k \ (ie\ angle\ small\ enough)$$
$$|x| < x_k \ (ie\ cart\ position\ small\ enough$$
$$\cos(\theta_f) < 0 \ (ie\ pendulum\ is\ up)$$
$$|\dot{\theta}_f| < \dot{\theta}_k \ (ie\ pendulum\ moving\ slowly\ enough)$$

When all of the above conditions are true, the output of the AND gate is 1. We limit the rate of change of the output and pass it through a backlash block in order to “de-bounce” it. This output is the signal **MODE**. When MODE is 0, the voltage fed to the motor is obtained from the “swing-up” controller, When MODE is 1, the voltage to the motor is obtained from the stabilizing controller.

Furthermore, MODE is fed to a delay block (1.8s / (s+1.8)) and its output is fed to a comparator. When the output of the comparator is 1, the value of θ_k is reduced by 13 degrees. This now changes the stabilizing conditions to the following “***let go***” conditions. When any of the following conditions is NOT TRUE, the system switches MODE to 0 and goes back to swing up mode.

$|\theta_u| < \theta_k$ (*ie angle disturbed too much*)

$|x| < x_k$ (*ie cart position tool far*

$\cos(\theta_f) < 0$ (*ie pendulum is not up*)

$|\dot{\theta}_f| < \dot{\theta}_k$ (*ie pendulum moving too quickly*)

8.3.5 Balance control

This is the stabilizing state feedback controller. It simply feeds back the voltage:

$$V_s = -(k_1 x + k_2 \theta + k_3 \dot{x} + k_4 \dot{\theta})$$

as obtained in the LQR design.

This voltage maintains the pendulum upright

8.3.4 Results

Typical results are shown in Figure 8.3.5. These are all obtained from the same run. The controller starts by applying a de-stabilizing cart position command limited to ±3*(1+Kh) cm. This results in the pendulum swinging with increasing amplitude. At T = 2.75 seconds the cart command is limited to ±3 cm. The destabilizing controller continues however to increase the amplitude of the pendulum oscillations.

At some point, all the following conditions are met:

- Pendulum UP angle < 15 degrees
- Pendulum speed < 100 deg/sec
- Cart position < 25 cm
- Pendulum Full angle in 2nd or 3rd quadrant

This sets MODE = 1 and the state feedback controller output V_s is fed to the motor rather than the destabilizing voltage V_d. This stabilizes the pendulum and keeps it upright.

The Mode is maintained to the value 1 unless one of the following conditions becomes false:

- Pendulum UP angle less than 2 degrees
- Cart position < 25 cm
- Pendulum velocity < 100 deg/sec
- Pendulum Full angle is in 2nd or 3rd quadrant as defined in Figure 8.3.2

If you apply a tap to the pendulum, it will remain upright unless the tap was so hard that one of the above conditions is not maintained. If MODE switches to zero, then the pendulum falls and the cart will start oscillating again until the pendulum comes up and MODE switches to 1. Note however that there may be situations when the tap was so hard that the pendulum swinging in full rotations and never stabilizes. It is of course possible to design a controller that avoids this situation. One way would be to implement a controller that detects full rotations and turns off the system momentarily until the pendulum is simply swinging in the down mode and then resume to the swing up controller in low gain.

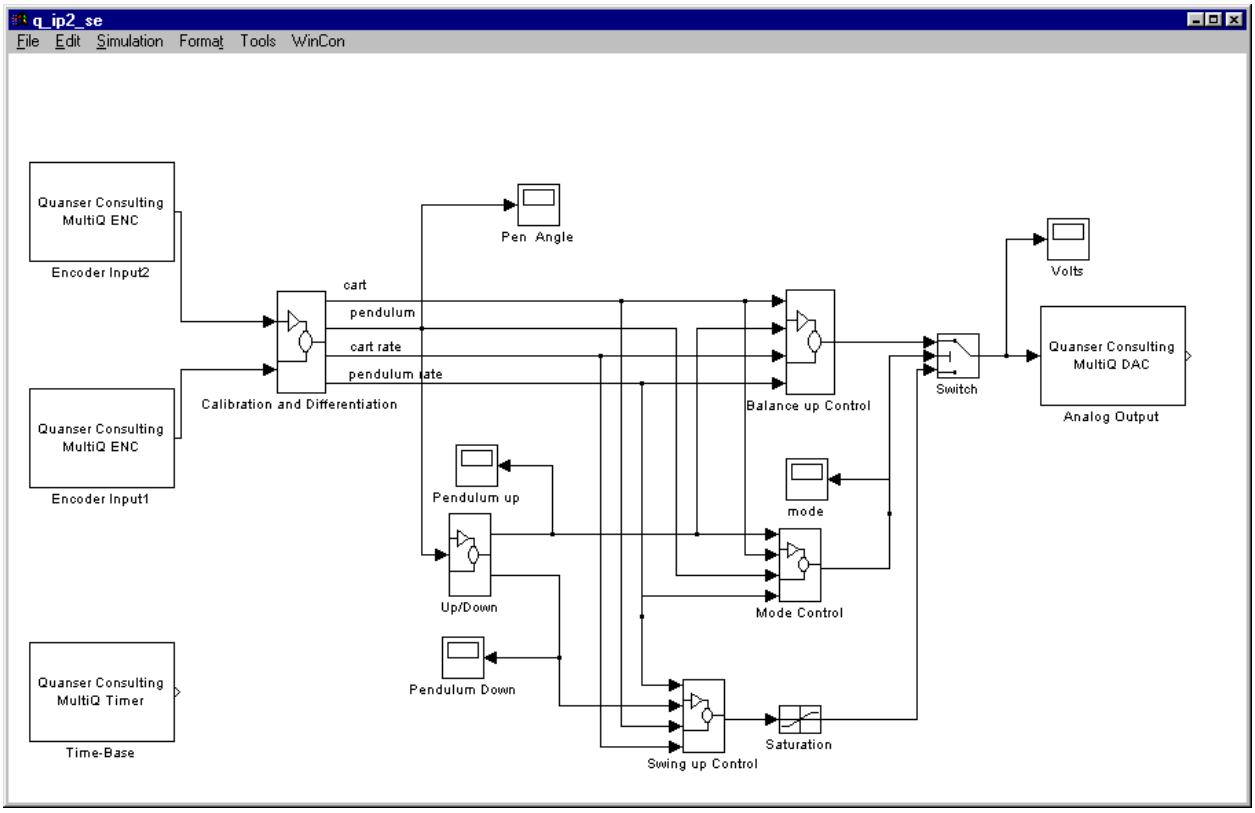


Figure 8.3.4 Simulink Controller for the Self-Erecting Inverted Pendulum

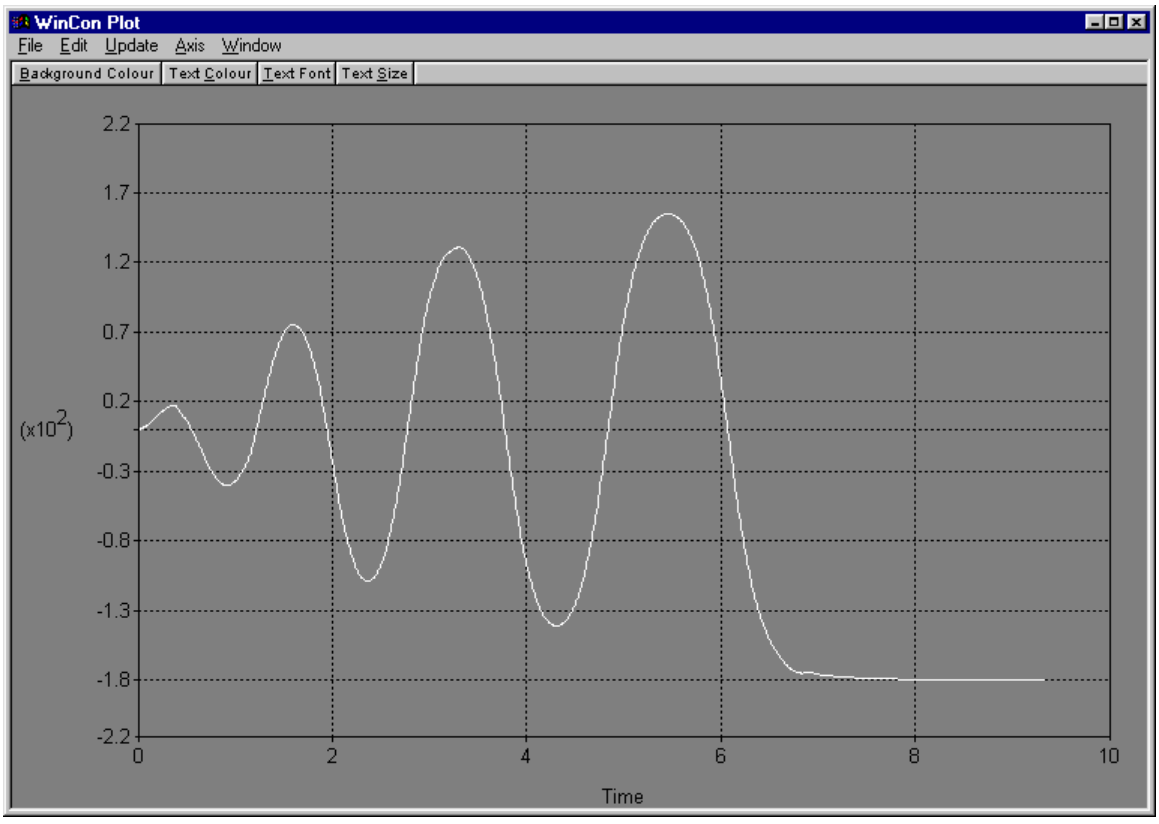


Figure 8.3.5 Plot demonsrating the typical results achieved with implemenation described in Section 8.3.4

8.4 High Wire walker - state estimation

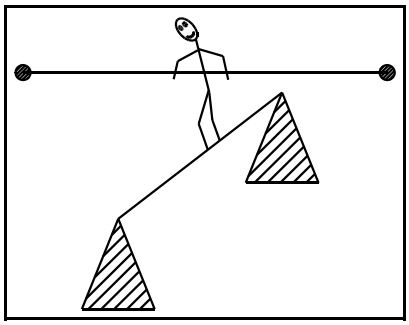


Figure 8.4.1 High wire artist

The high wire walker experiment (HWW) is an electromechanical system which is dynamically equivalent to the tightrope circus walker shown in Figure 8.4.1.

It consists of a body that resembles an "***inverted obelisk***" which has a sharp bottom making it unstable in the vertical position. A motor is mounted to the body which can apply a torque to a long beam mounted in front of the body. The moment of inertia of the stick is much higher than the body and can thus be used as a "fixed" inertial frame against which the torque applied by the motor can be used to keep the body upright. The fall of the body relative to the vertical axis however ***cannot be measured***. An instrumented pendulum mounted to the body swivels as the body falls and is used to estimate the fall of the body relative to the vertical axis. Although the dynamics of the pendulum movements are very different from the falling body, an observer (Kalman filter) is designed which estimates the body angle relative to the vertical axis. The observer state and the measured states are then used to stabilize the entire structure.

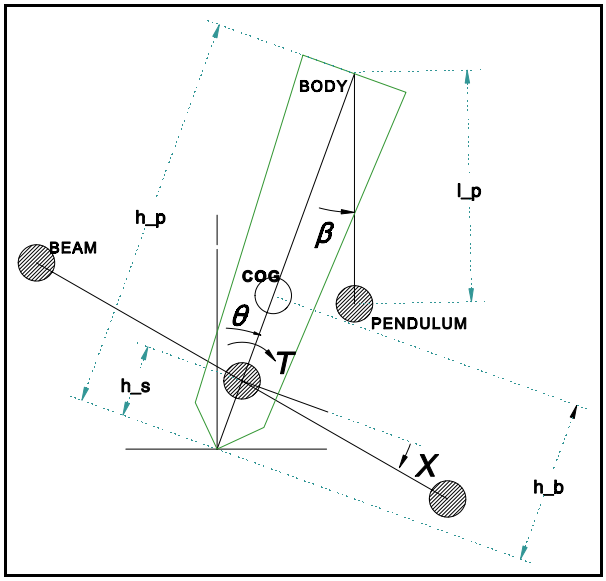
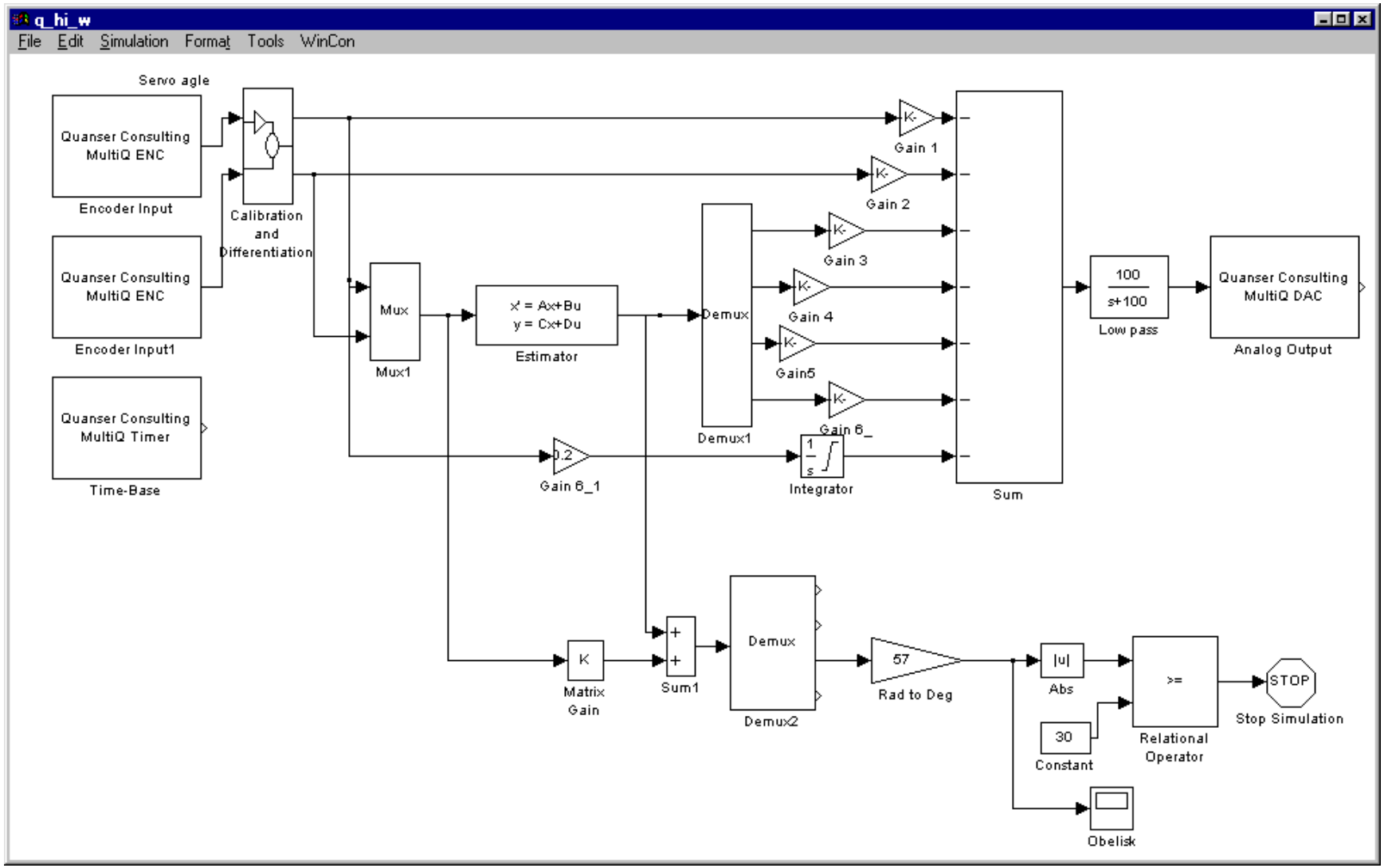


Figure 8.4.2 High wire experiment hardware

The controller is implemented in SIMULINK and run in realtime using WinCon. Figure 8.4.3 shows the main block diagram of the controller.

The two encoders are used to measure the stick angle relative to the body and the pendulum angle relative to the body. These are then fed to calibration blocks which convert from "counts" to radians as required by the controller. The two signals are then fed to the observer block which implements the observer dynamical system described by $[A_{obs},B_{obs}]$ whose output is the state z. The feedback gain Kf is computed in the MATLAB program "hi_wr.m". The measured angles and the observed state are passed through the gain Kf to calculate the output voltage of the controller which is then fed to the D/A block which is tied to the motor.

Starting the controller requires that you hold the system in the "zero" position: body vertical, beam horizontal and the pendulum vertical and ***not swinging***. As soon as you let go, the body remains balanced and the beam rotates to compensate for the fall of the body. This controller demonstrates that observers can be used successfully in estimating unknown states and controlling an unstable system.


q_hi_w
File Edit Simulation Format Tools WinCon
Servo agle
Quanser Consulting MultiQ ENC
Encoder Input
Quanser Consulting MultiQ ENC
Encoder Input1
Quanser Consulting MultiQ Timer
Time-Base
Calibration and Differentiation
Mux
Mux1
x' = Ax+Bu
y = Cx+Du
Estimator
Demux
Demux1
Gain 1
Gain 2
Gain 3
Gain 4
Gain5
Gain 6_
0.2
Gain 6_1
1/s
Integrator
Sum
100/(s+100)
Low pass
Quanser Consulting MultiQ DAC
Analog Output
K
Matrix Gain
Sum1
Demux
Demux2
57
Rad to Deg
|u|
Abs
30
Constant
>=
Relational Operator
STOP
Stop Simulation
Obelisk

9.0 Scripting From MATLAB

Quanser Consulting provides Matlab scripts which can be used to control WinCon from the Matlab window. These are very useful for applications which require automated execution such as gain scheduling, data collection, adaptive learning control and parameter estimation.

9.1 Available script commands

9.1.1 WC_PATH: returns the path for the given model.

```
    P = WC_PATH
    P = WC_PATH('model')
where:
    'model' = string containing name of model
```

For example **P = WC_PATH('q_p')** where q_p is the name of the Simulink diagram for the model. If the model hasn't been saved then an empty string is returned. If the model is not open, then it must be in the Matlab path (see the **PATH** command in the Matlab documentation). If no argument is specified, then the path for the current system is returned. See also: **GCS**

9.1.2 WC_BUILD: build the code for a model.

```
   WC_BUILD
   WC_BUILD('model')
   WC_BUILD('model', path')
 where:
   'model' = string containing model name, such as 'test'
   'path'  = string containing the path to the model folder.
```

The path is not required if the model is open, or if it is in the Matlab path (see **WC_PATH**). If no model is specified then the current system is built. **WC_BUILD** generates and compiles the model code and then downloads it to WinCon. For example **wc_build('q_p','c:\wincon3\examples');**

9.1.3 WC_RUN: Run WinCon Server. If it is already running, it brings it to the foreground.

```
    R = WC_RUN
where:
    R = 1 if WinCon had not been running
        0 if WinCon was already running or failed to run.
```

For example **R = wc_run;**

9.1.4 WC_DOWNLOAD: download the code for a model to the active client

```
    R = WC_DOWNLOAD
    R = WC_DOWNLOAD('model')
    R = WC_DOWNLOAD('model', 'path')
 where:
    R       = boolean (0 or 1) result where 1 indicates success
    'model' = string containing model name, such as 'test'
    'path'  = string containing path to generated code
```

WC_DOWNLOAD will run **WinCon W95Server** automatically if it is not already running. The model path may be obtained using **WC_PATH**. The path need not be specified if the model is open or if it is in the Matlab path. If no model is specified, the current system is used. For example
wc_download('q_p','c:\wincon3\examples');

9.1.5 WC_START: start the code for a model.

```
    WC_START
    WC_START('model')
    WC_START('model', 'path')
where:
    'model' = string containing model name, such as 'test'
    'path'  = string containing path to generated code
```

The path is not required if the model is open or if it is in the Matlab path (see **WC_PATH**). If no model is specified then the current system is started. **WC_START** will run WcServer automatically if it is not already running. For example **wc_start('q_p','c:\wincon3\examples');**

9.1.6 WC_STOP: stop the code for a model.

```
    WC_STOP
    WC_STOP('model')
    WC_STOP('model', 'path')
where:
    'model' = string containing model name, such as 'test'
    'path'  = string containing path to generated code
```

The path is not required if the model is open, or if it is in the Matlab path (see **WC_PATH**). If no model is specified, then the current system is used. **WC_STOP** assumes **WinCon W95Server** is running.
For example **wc_stop('q_p','c:\wincon3\examples');**

9.1.7 WC_NEWPLOT: open a new plot in WinCon.

```
      WC_NEWPLOT
      WC_NEWPLOT('model')
  where:
      'model' = string containing name of model
```

If no argument is specified, **WC_NEWPLOT** uses the current model. The user will be presented with a list of the different plot types to choose from. **WC_NEWPLOT** will run **WinCon W95Server** automatically if it is not already running. For example **wc_newplot('q_p');**

9.1.8 WC_OPENPLOT: open a plot in WinCon.

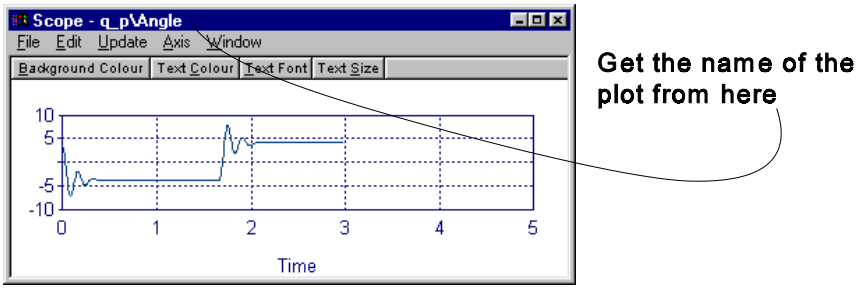
```
      WC_OPENPLOT
      WC_OPENPLOT('model')
  where:
      'model' = string containing name of model
```

If no argument is specified, **WC_OPENPLOT** uses the current model. The user will be presented with a list of the different plots to choose from. **WC_OPENPLOT** will run **WinCon W95Server** automatically if it is not already running. For example **wc_openplot('q_p');**

9.1.9 WC_SAVEPLOT: save plot data to a MAT file

```
      WC_SAVEPLOT('plot title', 'filename')
  where:
      'Plot title' = title of the plot window
      'filename'   = name of file to save data to
```

WC_SAVEPLOT looks for the plot with the given title and tells it to save its data to the given file in MAT format. Note that the .mat extension **MUST** be specified in the filename. It is not added automatically. For example **wc_saveplot('Scope - q_p\Angle','scope.mat')** will save the data displayed into the file scope.mat which can be loaded into the matlab workspace by issuing `load scope`.(See your



Matlab documentation) Note the name of the plot you are copying from is shown in the Task bar of the plot. Also note that when you save data from a plot, it switches it to fixed mode temporarily and collects the data from the server without any decimation (for details see section 6 of this manual). ***It is important to wait until all the data has been collected before you start the controller again.*** See the script example that ensures that the data has been saved before you start a second run.

9.1.10 `WC_ISRUNNING:` test if a model is running

```
R = WC_ISRUNNING
R = WC_ISRUNNING('model')

where:
'model' = string containing model name
```

The return value is 1 if the model is running, 0 otherwise. If no model is specified, the current system is used. For example the statement **`while(wc_isrunning('q_p','c:\wincon3\examples')`** is a while loop that polls for execution termination of the WinCon controller q_p from the Matlab window

9.1.11 `WC_SELECT:` Select the given model in WinCon

```
WC_SELECT
WC_SELECT('model')
WC_SELECT('model', 'path')
where:
'model' = string containing model name
'path'  = string containing path to generated code
```

`WC_SELECT` will run **WinCon W95Server** automatically if it is not already running. The model path may be obtained using **`WC_PATH`**. The path need not be specified if the model is open or in the Matlab path. If no model is specified, the current system is used. For example
`wc_select('q_p','c:\wincon3\examples');` Will select q_p as the active WinCon model.

9.1.12 `WC_OPEN:` open the specified WinCon project.

```
R = WC_OPEN('project')
where:
R = boolean (0 or 1) result where 1 indicates success
'project' = string containing project filename
```

`WC_OPEN` will run WcServer automatically if it is not already running. For example `wc_open('test');` will open a project file named test.wcp.

9.1.13 `WC_SAVE:` Save the current WinCon project.

```
WC_SAVE
WC_SAVE('file')
where:
```

```
'file' = is the name of the file to save to
```

Saves the current WinCon project. If a filename is specified then the project is saved to the specified file. If no extension is specified, then an appropriate extension will be added. For example **`wc_save('test');`** will save the current model and associated plots etc. in a WinCon project file named test.wcp.

9.1.14 `WC_CLOSE:` Close the current WinCon project. You will be prompted to save the project if it has been modified. For example **`wc_close;`**

9.1.15 `WC_SETOPTIONS:` set the build options for a model to generate real-time code for WinCon.

```
WC_SETOPTIONS
WC_SETOPTIONS('MODEL')
WC_SETOPTIONS('MODEL' ,OPT)
where:
MODEL = name of model diagram
OPT   = 'win', 'dos' or 'nt'
```

If an option is not specified, 'win' is assumed. If no model is specified, then the options are set as the defaults. If a model is specified, it must be open. Call **`wc_setoptions()`** in **`startup.m`** to make WinCon the default for real-time code generation. To set the defaults for a client other than a Windows client, call `wc_setoptions` with a model of zero.
For example:

`wc_setoptions('test')` - set the options for the 'test' diagram to build real-time code for a **WinCon W95Client**.

`wc_setoptions('test','dos')` - set the options for the 'test' diagram to build real-time code for a **WinCon DOSClient**.

`wc_setoptions` - set the default options to build real-time code for a **WinCon W95Client**.

`wc_setoptions(0, 'dos')` - set the default options to build real-time code for a **WinCon DOSClient**.

9.1.16 Initializing function

You may want to design controllers that use parameter values computed in the **Matlab workspace** using a Matlab script file. You can automate this process using the

`set_param('modelname','StartFcn','m_file_name')`

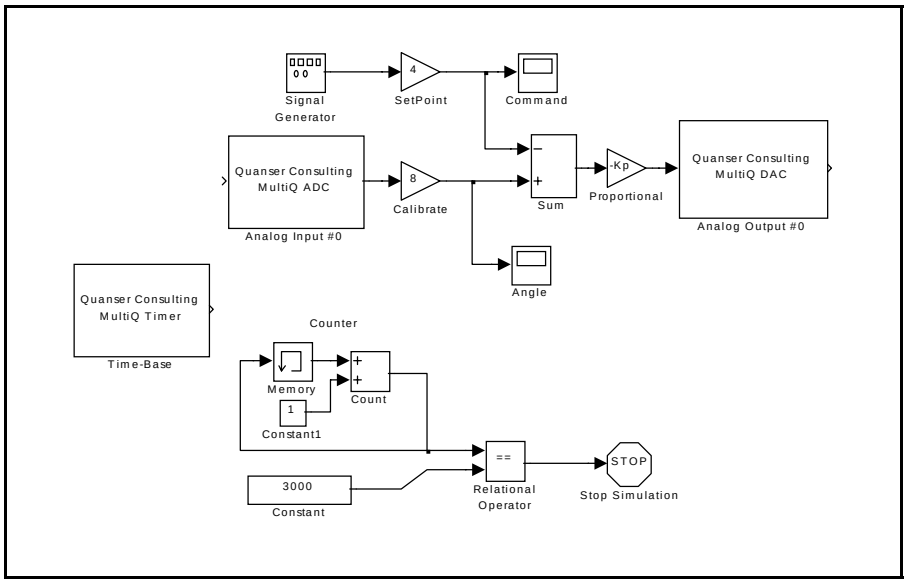
function. If you perform this on the active Simulink model, then each time you start the controller, the m_file_name will be executed in Matlab, then the controller will run with the new parameters values. The function

`set_param('modelname','InitFcn','m_file_name')`

runs the script file as soon as you load the model into Matlab.

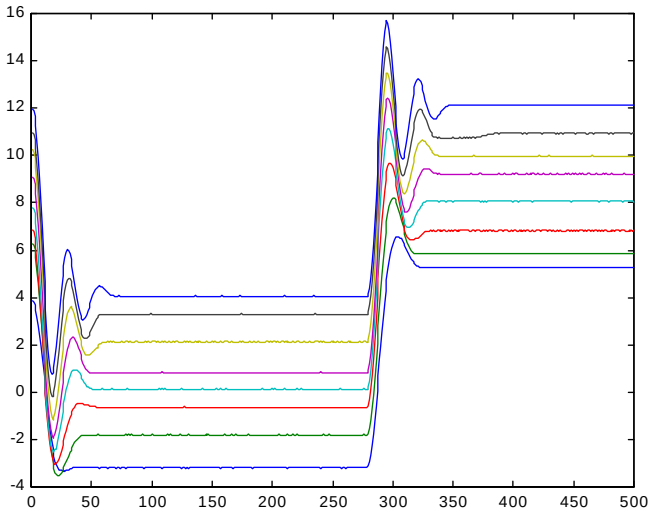
9.2 Scripting Example

Consider the Simulink diagram shown below (**q_p.mdl**) which is set up such that it stops execution sample #3000. The following script in MATLAB can be used to collect the response of the system for proportional gain Kp changing from Kp = 0.35 to Kp = 1.05 in steps of 0.1.




```
Kp = 2.5;
for I = 1:8
 Kp = Kp + 1.0;
 wc_start('q_p','C:\WINCON3\Examples')
 a = wc_isrunning('q_p');
 while(a == 1),
 a = wc_isrunning('q_p');
 end

 fname = ['res_' int2str(I) '.mat'];
 delete(fname) % to make sure that it does not exist before we save it
 wc_savePlot('Scope - q_p\Angle',fname)

 fprintf('Saving File %s',fname) % wait for the file to be saved before
proceeding
 done =0;
 while(done ==0),
 if(exist(fname,'file') == 2),
 done = 1
 end
end

end

for I = 1:2,
 fname = ['res_' int2str(I) '.mat']
```

```
 load(fname)
 nn = size(q_p_Angle)
  for J = 1:nn(1),
  out(J,I) = q_p_Angle(J)+I;
  end
end
plot(out)
```



Using the above script the Matlab program starts by initializing the gain **Kp** and executes the command **wc_start** which starts the controller. The script then waits for the termination of the controller which occurs in the diagram at sample #3000. When the execution is terminated, a filename is created (**res_n.mat, n = 1..8**) and the contents of the plot named "**Scope - q_p_Angle**" are saved to the file. The script then increments Kp and runs the controller again with a new value of Kp. This process is performed 8 times and at the end of the cycle, the files that were created are loaded to the workspace and the plots shown are generated. In this manner, the user can evaluate system response for values of Kp.

Note that the ***buffer length of the realtime plot should be longer than the duration of the run***. This way you ensure that you do not end up collecting only partial data from the run.

.

10.0 Hints and Troubleshooting

10.1 Multiple PC operation

If you are running the client and server on two different PC's, both Client and Server PC's must have an IP address assigned to them. Consult with your network expert in your institution to have this performed. In short, you must set the IP address from **Control Panel/Network/TCP-IP/Properties**. If you are using the **WinCon W95Server** on a Windows NT system, you will need administrative rights to access these properties.

Once you have both PC's booted and configured with TCP-IP address, then you must perform the following operations in sequence:

1) Start WcClient on the Client PC (Computer that controls the plant).
2) Start WcServer on the Server PC (Computer that has SIMULINK etc..).
3) From WcServer Client Connect to the Client PC using the client's IP address.

Now you have connected the Server to the desired client. When you click on download, you will download the code for the model you have selected to the client you have selected. If you connect to another PC on the network, you can also download a different model to it.

10.2 Compile results in error

The most likely cause is that one of the blocks you are using cannot be converted to realtime code. If that is the case, please contact us and we try to update the libraries to compensate for this.

The following blocks are not possible to convert to realtime operation since they require hardware access during the service of the controller:

- To File: In order to achieve this, you must plot the data and then save it to a file using the **WinCon W95Server**
- To Workspace: In order to achieve this function you must plot the data and save it to the workspace

10.3 I do not know my IP address

You can obtain the IP address of the PC you are using by either running "myipcfg" from the Windows directory or by clicking on the **Client/Network** menu in the **WinCon W95Server** Window or **File/Network** in the **WinCon W95Client** Window.

10.4 MATLAB Versions

This version of WinCon (Ver 3.0) has been tested using the following versions of MATLAB and its components. We will upgrade WinCon accordingly and new versions will be available through our web site **www.wincon.quanser.com**

You can obtain this list by typing ver in the MATLAB command window.

```
ver
-----------------------------------------------------
MATLAB Version 5.2.0.3084 on PCWIN
MATLAB License Identification Number: ******
-----------------------------------------------------
MATLAB Toolbox       Version 5.2         18-Dec-1997
```

```
Real-Time Workshop   Version 2.2.0       01-Jan-1998
Simulink             Version 2.2         21-Nov-1997
```

Please ensure that you have these versions installed on your system.

10.5 Compilers

We presently support the **Visual C++ Version 5.0 (or 5.1)Compiler Professional or Enterprise Edition**. We will keep upgrading WinCon with Visual C++ releases. There are no plans to support other compilers at the moment. Previous versions of WinCon were using Borland and Watcom. Presently, Visual C++ is the only compiler which can create Virtual Device Drivers (VxD's) without any additional 3 rd party tools. This is why we converted to Visual C++.

10.6 Crash!

WinCon is designed as a prototyping software and is not intended for use in applications that may result in any kind of injury. Although we have thoroughly tested WinCon, occasionally a Server or one of its windows may stop responding. Windows 95 does not usually crash. If one of the Windows stops responding, please enter [Alt-Ctrl-Del] and terminate the task that is not responding. This may close other Windows so it is best to shut down the PC and restart. The controller itself, ie the client almost never crashes. If you want to stop the controller, you will need to stop it from the client Window. ***We recommend you save your work before you start running a controller***.

10.7 Builder does not find compiler or its components

Check that all the options under RTW/Build are set as described in section 4

Run the program : **`c:\WinCon\bin\setenvvar.exe`** to set the appropriate paths for the environment variables. You may also need to run **`mex -setup`** from the MATLAB command window and enter the path to the Visual C++ compiler as shown below. Note that long names are truncated and appended with "**`~1`**".

10.8 I want to embed my own code into WinCon without SIMULINK and RTW

WinCon-Pro is available for professionals who want to execute their own C programs in realtime independent of SIMULINK.
For more information, please visit ***http://www.wincon.quanser.com***

10.9 I have been using WinCon 2.0 ... anything I should know?
- All Data acquisition blocks should be replaced with new ones from WinLib
- You do not need to make any scopes! You can plot the output of any block. You can access scopes more easily via **Plot/Open**
- You cannot change sampling periods on the fly
- You can save the contents of a plot to the Workspace.
- You can Script from MATLAB

10.10 How do I make a discrete block always run at the specified sample frequency?

Consider the discrete transfer function shown. You can make it run at the sample rate specified in **RAW/Options/Solver** by including the following line under sample time:

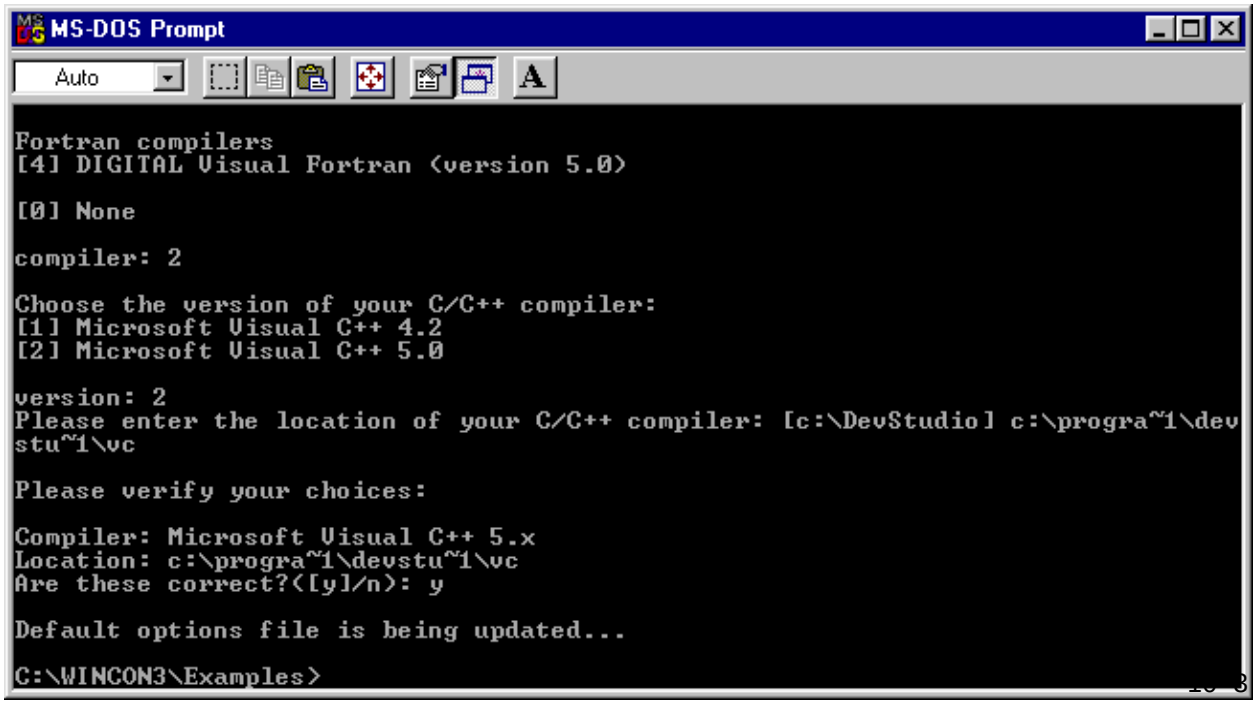
```
str2num(get_param(bdroot, 'FixedStep'))
```

10.11 The Controller cannot be started.

Make sure there are no conflicting devices on the PC that is running the client. For example, a sound card may interfere with the interrupt line for the MultiQ board. Remove the sound card if you do not need it.

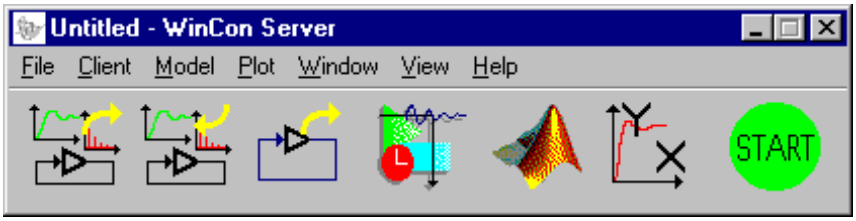
10.12 Cannot establish communications with client

Install a TCP / IP protocol following the install instructions



11. 0 Quick Reference to WinCon Windows

11.1 WinCon Server Main window



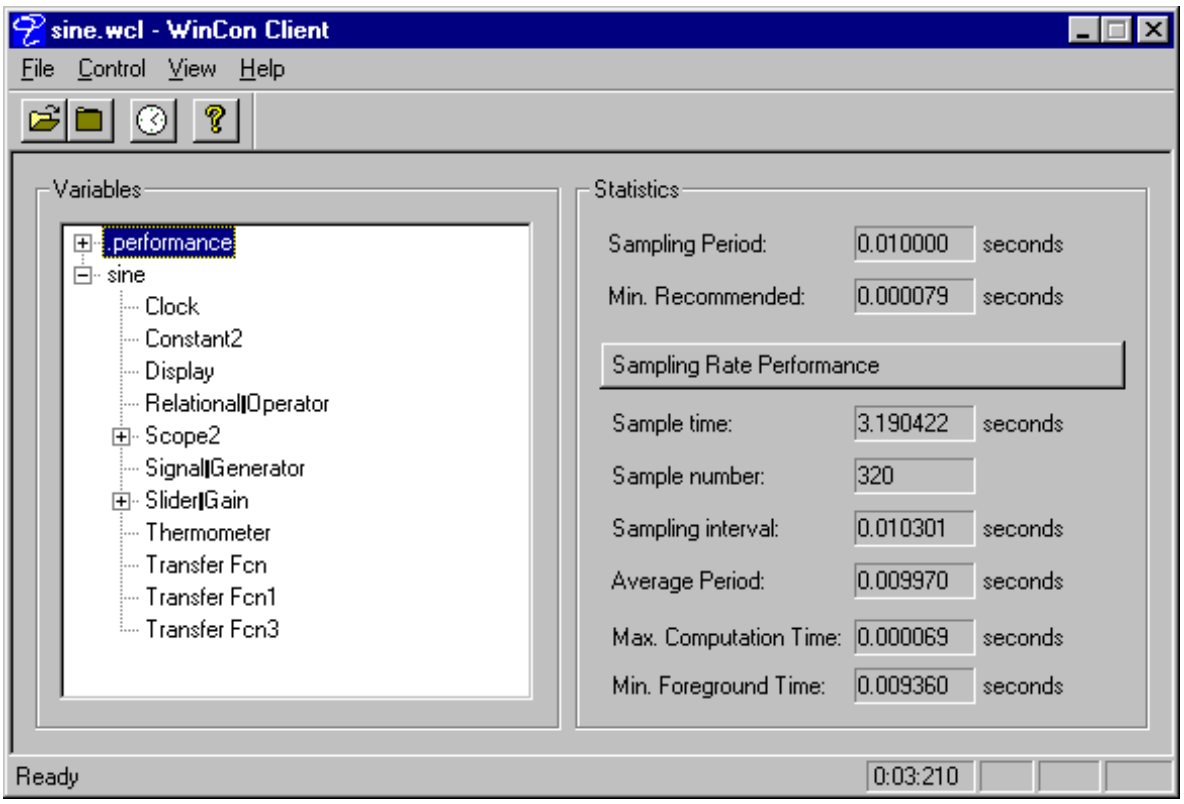
File		**Open and close WinCon Projects**
	New	Start a new WinCon project
	Open	Open an existing WinCon project
	Save	Save a WinCon project
	Save As	Save this project under a new name
	LIST	Quick access list of recent projects.
Client		**Information about and access to the Client to which the server is presently connected**
	Connect	Connect to a desired client. Enter the IP number and PORT
	Disconnect	Disconnect from the client with a check (✔) mark (See LIST below)
	Recent Clients	List of recent clients you were connected to
	Close on Exit	Closes ALL clients to which you are connect upon exit from the server
	Tolerance	Set the Tolerance on Tf (Foreground time) for the selected client
	Network	Runs "`winipcfg`" to identify the IP address of the computer
	More Cients	List of Clients with which you can communicate. A Check mark (✔) indicates that if you attempt a "download" you will download to the client which has the Check Mark. A ✱ at the end of the name indicates that you are a secondary server to that client. A secondary server cannot change controller parameters.
Model		**Open and Close WinCon controllers and SIMULINK models**
	Open	Open a WinCon controller (.wcl) or SIMULINK model (.mdl)
	Close	Close the active model (✔)
	Reload	Reloads the model you were working with
	Build	Build the realtime code of the active model. SIMULINK must be up and running
	Download	Download the model to the client
	Start	Start the model in the client to which you are actively connected
	Stop	Stop the active client
	Start All	Start all clients which you have connected to
	Stop All	Stop all clients
	LIST	List controllers which are loaded into the clients you have connected to. Selected Controller has a Check (✔) if you have selected the client in which it is loaded . This will activate the Plot variables for that client

PLOT		**Display variables in Realtime**
	New	Open a new display. Types are: Scope, XY, Digital, Thermometer, FFT and Analog Gauge
	Open	Opens an existing Scope, Digital Meter or Thermometer defined in the SIMULINK diagram.
	Close All	Closes all open realtime plots.
WINDOW		**Navigate to other Windows and set options.**
	Always on Top	Select whether you want the WinCon Server Toolbar on top of all other windows.
	Matlab	Go to MATLAB command window
	Simulink	Go to SIMULINK diagram of selected controller (if it is on your computer)
	Other Windows	Navigate to plots that you have opened
VIEW		**Control the appearance of the WinCon Server Window**
	Toolbar	Remove or show the toolbar
	Use Small Toolbar	Use Small ✔ toolbar

11.2 Client Window

You ***do not normally interact*** with this window. This window is used to monitor the progress of the Client. If the Server stops communicating with the client, then you can control it from this window. **The most commonly used feature is the Network menu which lets you obtain the IP address of the PC.**

FILE		**Open and close WinCon Controllers**
	Open	Open a .wcl file.
	Close	Close a .wcl file.
THESE ITEMS ARE VERY USEFUL	***Network***	**Obtain the network IP address of the computer which is running the client.**
	Port	**Set the PORT number. The port default number is 17255. If you are running other software that interferes with this port you may change this number.**
	Exit	Exit the client. Stops the controller and exits.
CONTROL		**Control the client: DO NOT NORMALLY USE UNLESS YOU HAVE LOST CONTACT WITH THE CLIENT THROUGH THE SERVER**
	Tolerance	Set the Tolerance Tf
	Start	Start the controller
	Stop	Stop the controller
VIEW		**Control the display of the window**
	Toolbar	View the toollbar
	Status Bar	View the status bar



Sampling Period	The desired sampling period specified in RTW options.
Min Recommended	Minimum recommended sampling period for the running controller based on observed Tc and specified Threshold Tf. Running the controller faster than this rate will result in a foreground time that will drop below the specified threshold and thus stop the controller.
Sample time	T, The instantaneous sample time from T = 0 (Start) of the running controller
Sample number	N, The instantaneous sample number for the running controller
Average Period	The average period attained in the running controller: Ta = T/N
Max Computation Time	The maximum computation time Tc observed for this controller. This gives you a measure of the worst case computation delay.
Min Foreground Time	The minimum foreground time Tf observed when using this controller running at the specified sampling period. This means that the %CPU used by the controller is 100(Tc/(Tc+Tf))