

Hardware-in-the-Loop (HIL) Simulator Operator's Manual



*Step-by-step instructions for setting up and running a
Hardware-In-the-Loop simulation*

Intelligent Machine Dynamics Laboratory
Room 225, E.J. Love Manufacturing Building
Georgia Institute of Technology
Atlanta, GA

IMDL
Intelligent Machine
Dynamics Laboratory

prepared by
Joe Frankel, Graduate Research Assistant
September 9, 2003



Contents

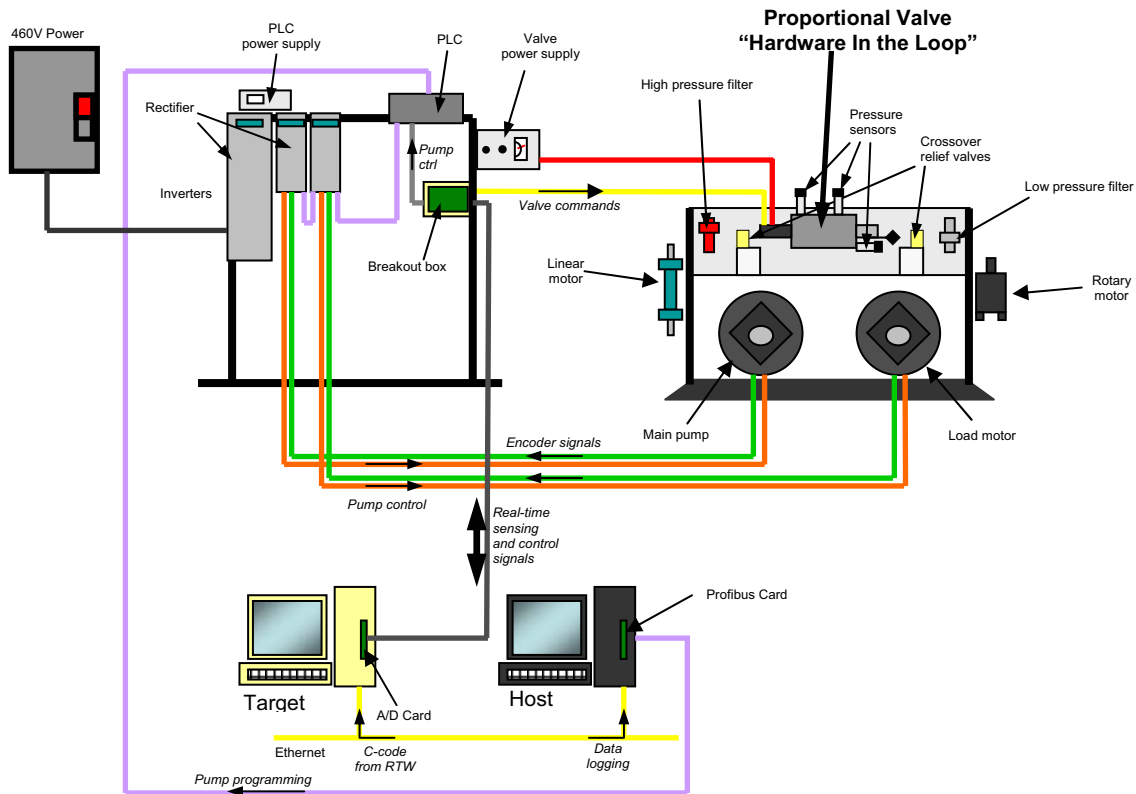
1. Background Information	3
1.1. Simulation Hardware	3
1.2. Simulation Software	4
2. Setting up a simulation	6
2.1. Install hardware	7
2.2. Plumb hardware	8
2.2.1. Supply and return	
2.2.2. Crossover relief	
2.2.3. Workport load	
2.3. Install sensors	9
2.3.1. Pressure transmitters	
2.3.2. Position transducer	
2.4. Connect electrical power and signals	10
2.4.1. Siemens Motors	
2.4.2. Pressure Transmitters	
2.4.3. Valve: Hardware in the Loop	
2.4.4. Temposonics Position Transducer	
2.4.5. SCB-68 Breakout Board	
2.5. Setup host-target communication	15
2.5.1. Configure IP connection	
2.5.2. Create boot disk	
2.6. Program Simulink / xPC Target model	16
2.6.1. xPC Target Blocks	
2.6.2. Simulation Parameters	
2.6.3. Data Acquisition	
2.6.4. Sample Control Program	
2.7. Compile & upload Simulink / xPC Target model	25
3. Running a simulation	26
3.1. Running the Siemens motors	26
3.1.1. Turning on the power	
3.1.2. Running the drive software	
3.2. Running the Target Application	29
3.2.1. Starting and stopping the application	
3.2.2. Adjusting model parameters	
3.2.3. Viewing Host Scopes	
3.2.4. Acquiring Data	
4. Resources	32

1. Background Information

1.1 Simulation Hardware

Figure 1 illustrates the basic components and signal flows of the HIL equipment. Figure 2 is a photo of the overall HIL system.

Figure 1: Hardware-In-the-Loop Components



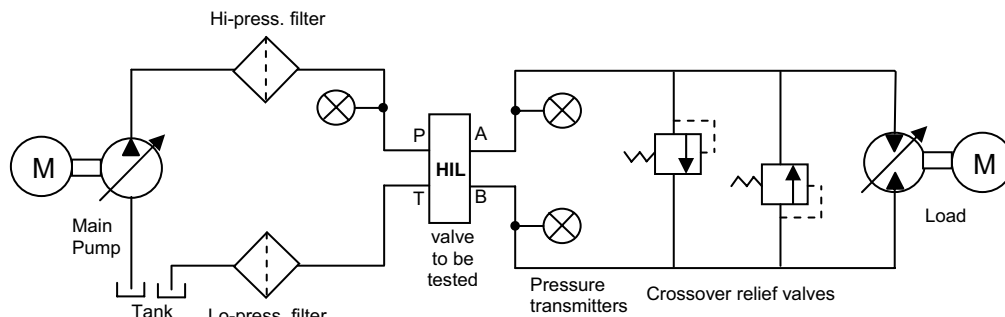
Both the Main Pump and Load Motors shown in Figure 1 are the same unit from the Siemens company, model #1FT6102-8AF71-3AK1. For a typical HIL simulation, the motor on the left is configured as a source for hydraulic power (a pump) and the motor on the right is configured as a sink for hydraulic power (a load). However, the system is flexible in that either motor can be configured as a power source or a power sink.

Figure 3 illustrates the HIL hydraulic circuit in a typical testing configuration. Note that the Siemens load motor cannot be driven in both directions. However, other smaller motors, such as linear hydraulic cylinders or rotational motors may be used, which can be run bidirectionally.

Figure 2: The Hardware-In-the-Loop System



Figure 3: Hydraulic Circuit



Also, although the main pump can be configured to drive fluid in either direction, the high and low pressure filters can only filter in one direction. Therefore, if you want to reverse the pump, you will have to reconfigure the filters so that the flow passes through them in the proper direction.

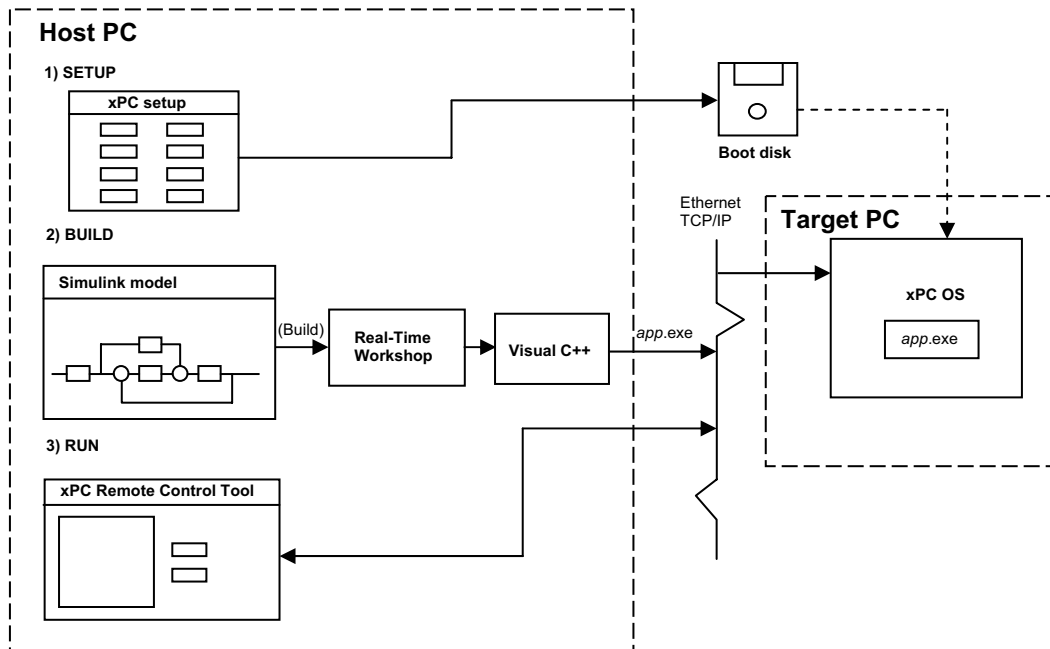
1.2 Simulation Software

The HIL simulator is programmed using Matlab, Simulink, xPC Target, Real-Time Workshop, and MS Visual C++. Figure 4 illustrates the flow of information during the setup and execution of an xPC control system.

To setup a Hardware-in-the-Loop simulation, the user first configures the Host PC-Target PC ethernet connection using the `xpcsetup` tool. This tool is executed from the Matlab command line and used to define the network connection between the two computers. Once all the `xpcsetup` fields are

configured properly, a 3 1/2" floppy Boot Disk is created which contains the xPC operating system and the Host-Target specific information.

Figure 4: Software flow diagram



Next, the user builds a model using standard Simulink blocks, and adds a combination of the A/D, D/A, and DIO blocks from the xPC Target toolbox into the Simulink model. These xPC Target blocks provide the link between the Simulink model and real world devices.

Once the Simulink model has been created, the `xpcrcctool` is executed from the Matlab command line. This tool provides the interface between the host and target PCs during both build and run time. The real-time C-code is generated with the Build button, which automates the compiling and uploading process. Since the entire build-and-upload sequence is automated and executed with a single click of the Build button, rapid prototyping and re-compiling of control algorithms is fast, easy, and does not require any knowledge of C programming.

To run a simulation, the Target PC is first booted from a floppy disk containing the xPC operating system. This is a low-level OS that requires very little memory and can be run very fast. Once the Target PC is running, the C code is built in the Host PC and uploaded to memory in the Target PC. No software runs on the Target PC other than the xPC OS and target application at run time, eliminating interruptions and subroutines typical of Windows operating systems which would hamper real-time processing and control. The target application can be controlled from either the Host PC or the Target PC, and data can be logged while the application is running or after it stops. *Model parameters such as controller gains can even be modified in real time without stopping the program execution.* This allows for tuning system performance on the fly, which can make control design a quick, efficient, and relatively pain-free process.

2. Setting up a simulation

Note: These instructions were written based upon the HIL setup of the Sauer-Danfoss PVG32 electrohydraulic proportional valve. Some of the steps specific to the setup of the PVG32 will vary slightly when setting up tests on other valves. Steps that are specific to the PVG32 are shown in italics.

Follow the steps below to set up a HIL simulation. More detailed descriptions for each step follow below the list.

HIL SETUP SEQUENCE

2.1. Install hardware

2.2. Plumb hardware

- 2.2.1. **Supply and return**
- 2.2.2. **Crossover relief**
- 2.2.3. **Workport load**

2.3. Install sensors

- 2.3.1. **Pressure transmitters**
- 2.3.2. **Position transducer**

2.4. Connect electrical power and signals

- 2.4.1. **Siemens Motors**
- 2.4.2. **Pressure Transmitters**
- 2.4.3. **Valve: Hardware in the Loop**
- 2.4.4. **Temposonics Position Transducer**
- 2.4.5. **SCB-68 Breakout Board**

2.5. Setup host-target communication

- 2.5.1. **Configure IP connection**
- 2.5.2. **Create boot disk**

2.6. Program Simulink / xPC Target model

- 2.6.1. **xPC Target Blocks**
- 2.6.2. **Simulation Parameters**
- 2.6.3. **Data Acquisition**
- 2.6.4. **Sample Control Program**

2.7. Compile & upload Simulink / xPC Target model

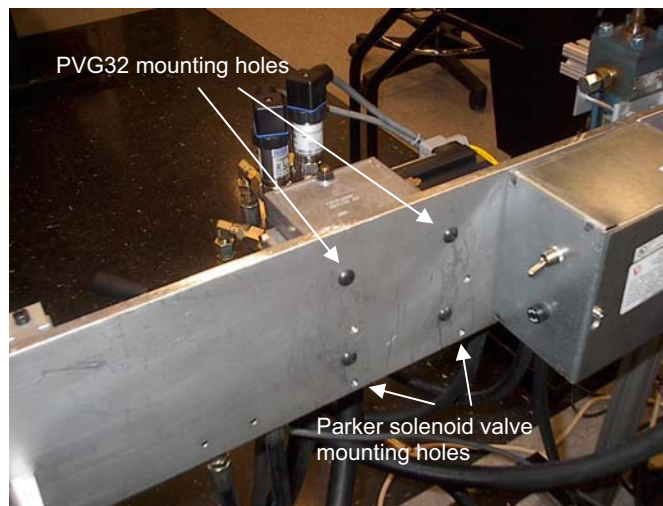
In the instructions that follow, the terms "hardware" and "valve" both refer to the valve that is being tested. General information that may be useful is written in smaller font. ***Basic instructions are written larger in bold blue italics.***

2.1 Install Hardware

Mount the valve to the mounting plate to the motor stand as illustrated in Figure 5.

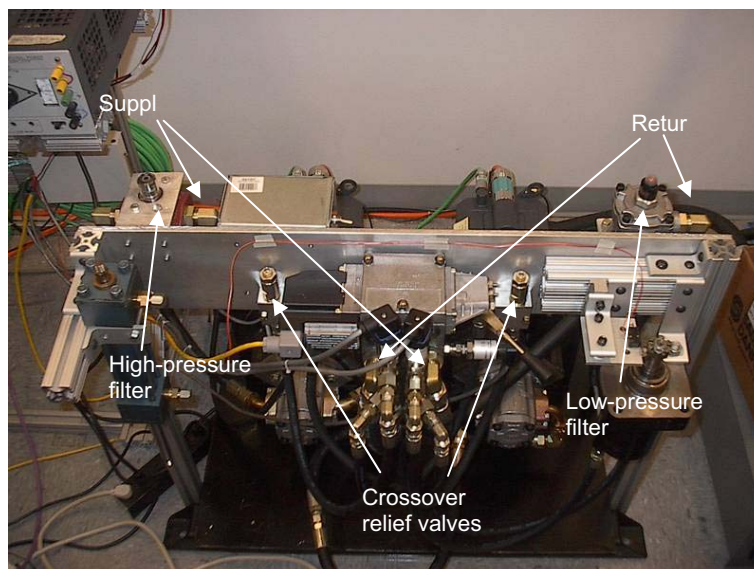
This may require machining new holes to fit the bolt pattern of the mounting holes on the valve body. *The PVG32 required drilling 4 holes in the mounting plate for the 1/4-20 bolts as shown.*

Figure 5: Mount valve to mounting plate



2.2 Plumb Hardware

Figure 6: Supply and return connections



2.2.1 Supply and return

Connect the supply and return hoses to the valve as illustrated in Figure 6.

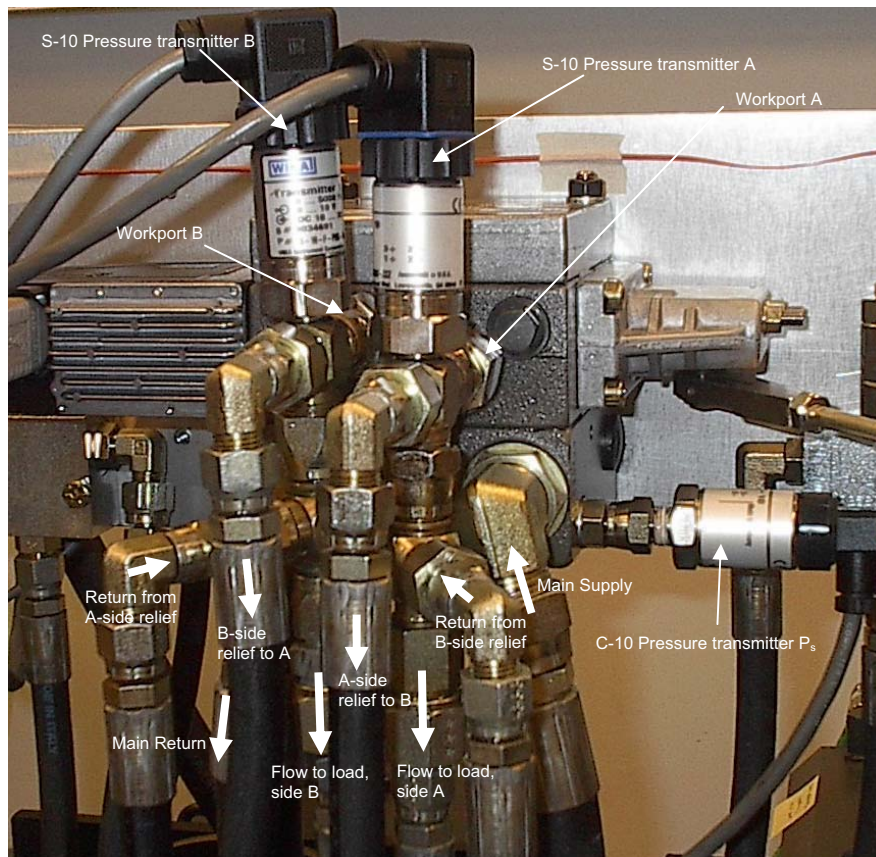
Connect the supply line from the output of the high pressure filter to the supply port in the valve. Connect the return line from the inlet of the low pressure filter to the return port in the valve, as illustrated in Figure 6. *For the PVG32, this required new JIS/SAE threaded fittings.*

2.2.2 Crossover relief

Connect the valve being tested to the crossover relief valves as illustrated in Figure 6 and 7.

Connect the each of the valve's (2) workports to tees to provide for pressure/flow to four places: 1) crossover relief to the other port, 2) crossover relief from the other port, 3) flow to load, and 4) pressure transmitters. This may be accomplished in a variety of ways, so long as each node connected to the workports of the valve is split into these four paths. See Figures 6 and 7. Also see Figure 3 for a diagram of the overall hydraulic circuit.

Figure 7: Valve plumbing

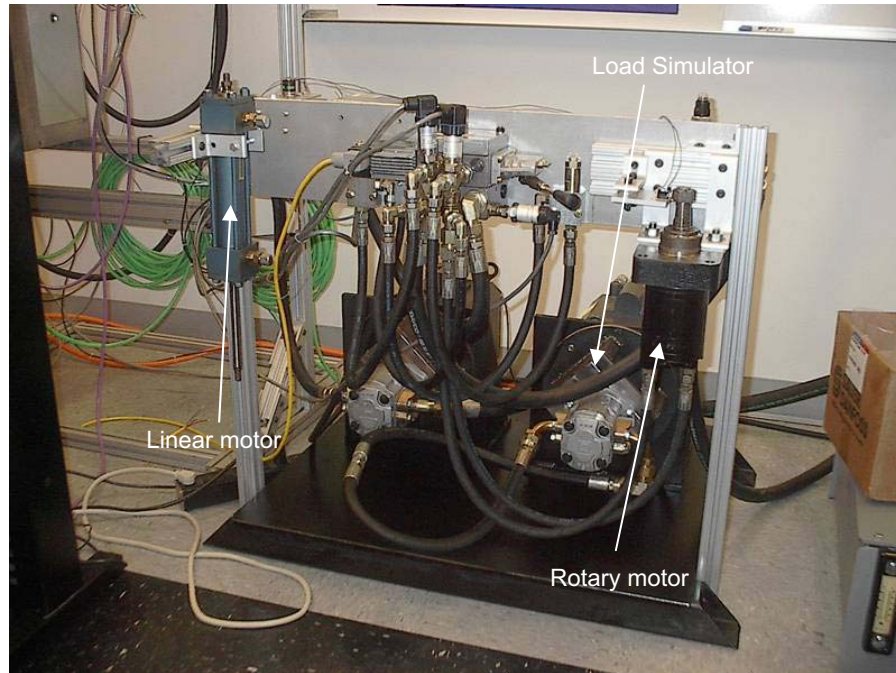


2.2.3 Workport Loads

Connect the workports of the valve to the load of your choice.

Figure 8 illustrates the three choices for loads: the Load Simulator, the Linear Motor (cylinder), and the Rotary Motor.

Figure 8: Workport load options



2.3 Install Sensors

2.3.1 Pressure Transmitters

Connect the pressure transmitters to the workports and supply.

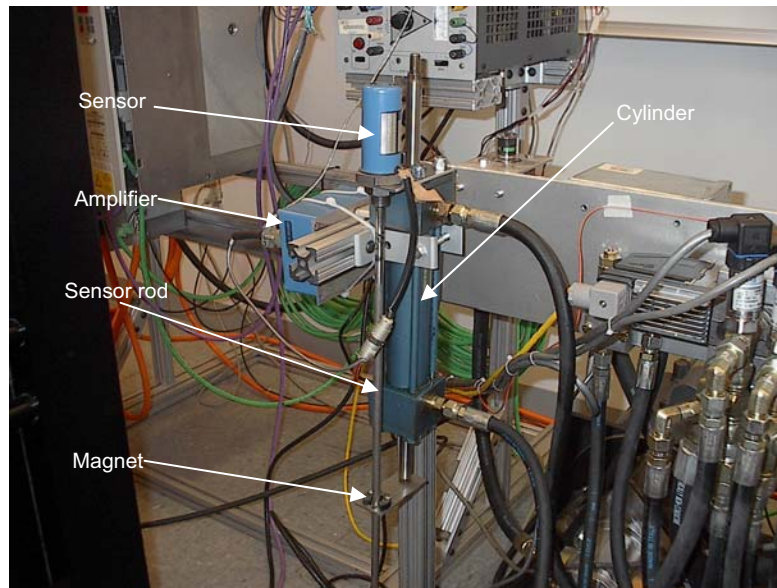
Install the WIKA model C-10 pressure transmitter into the gage port on the valve body. Install the WIKA model S-10 pressure transmitters into a suitable connection in the outlet of workports A and B on the valve. *The PVG32 required purchasing appropriate JIS/SAE/pipe adapter fittings for these connections.* See Figure 7.

2.3.2 Position Transducer

If using the linear motor (cylinder), the Temposonics position transducer should be ready to use.

If not connected already, install the Temposonics linear position transducer to the linear cylinder as shown in Figure 9.

Figure 9: Linear Motor and Temposonics Position Sensor



The temposonics position sensor can accurately measure the cylinder position, and has a 20in measurement range. Other sensors with wider ranges (longer sensor rods) are also available in the IMDL if desired. For more information on this device, see the documentation at <http://www.mtssensors.com>.

2.4 Connect Electrical Power and Signals

2.4.1 Siemens Motors

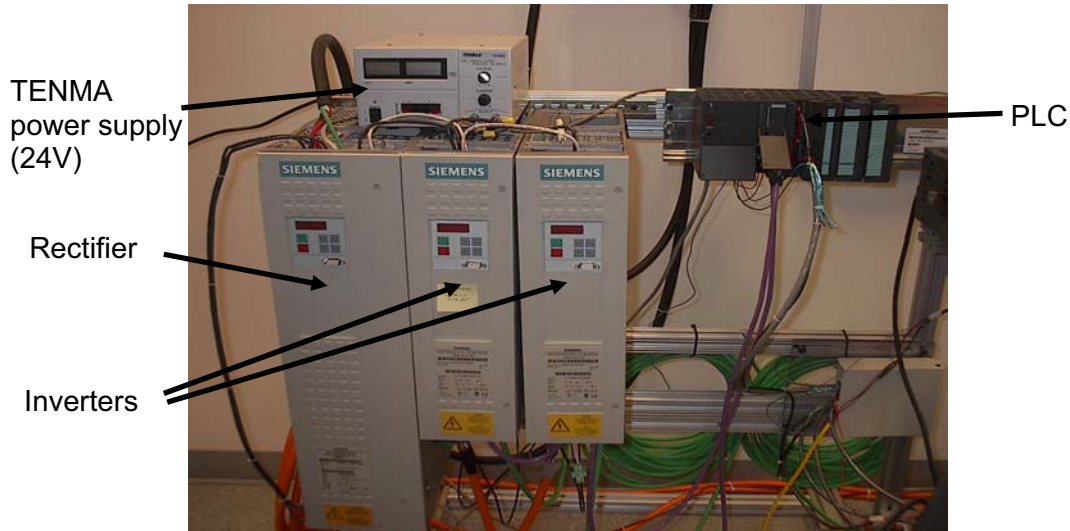
The power supply and controls for the Siemens motors should be ready to use.

The Siemens motors run on 460V, 3 ϕ power, which is first rectified and then inverted again in the Siemens motor drivers. Motor control is handled with the PLC and integrated encoders. The motor drivers and PLC are illustrated in Figure 10.

The TENMA power supply, also shown in Figure 10, powers both the PLC and the WIKA pressure transmitters. This power supply should be set at 24Vdc.

Control of the Siemens motors is accomplished through the Profibus card in the Host PC which is connected to the PLC. The DriveMonitor software is used to set motor speed. See section 3: Running a Simulation.

Figure 10: Siemens drivers, TENMA power supply, and PLC



At the current time, more sophisticated motor control is under development to allow for integrating motor control into Simulink models.

2.4.2 Pressure Transmitters

The power supply and signal connections for the pressure transmitters should be ready to use.

The WIKA pressure transmitters are powered by the TENMA power supply, and send their output signals to channels AICH2, AICH3, and AICH4 on the SCB-68 breakout board. *For the PVG32, the three transmitters were used to measure supply pressure, Workport A pressure, and Workport B pressure.*

2.4.3 Valve: Hardware in the Loop

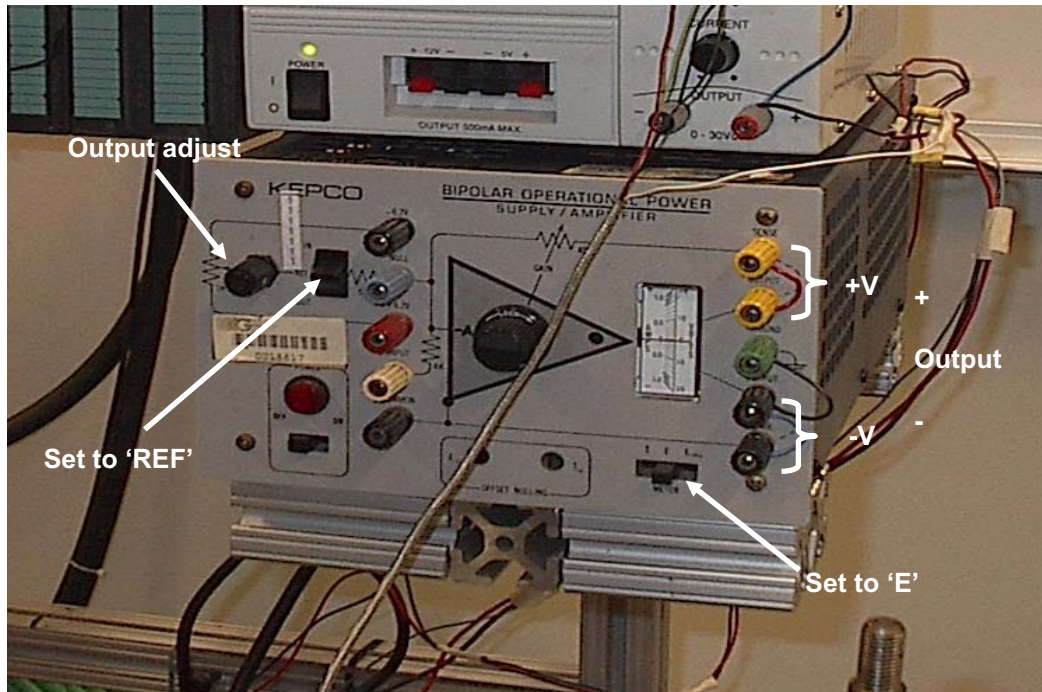
Connect the KEPCO power supply to the valve to be tested and set at the required supply voltage. Connect the control input connection on the valve to be tested to analog output channel 2 (DAC1OUT) on the breakout board.

The KEPCO power supply powers the valve to be tested, i.e. the Hardware in the Loop. The KEPCO power supply is illustrated in Figure 11.

To set the output to a constant, regulated dc voltage, jumper the two yellow connectors (sense & output), for the +V output and jumper the two black connectors (sense & output) for the -V output. This provides for internal feedback regulation of the power sent to the valve. Set the switch on the upper left

to 'REF', the current/voltage meter switch on the bottom right to 'E'. Then adjust the output voltage with the small black dial to the left of the 'REF' switch.

Figure 11: KEPCO Power Supply



The green ground connector is isolated from the case and AC ground cord and may be tied to other grounds as necessary to set the reference between power and control signals. *For the PVG32, the ground connector (green) was tied to both the -V output (black) and the analog ground on the A/D board, and the output was set to 12Vdc.*

For more information on the KEPCO power supply, such as pinout assignments for the terminals on the back of the box, or how to set up an external trigger, consult the KEPCO 36-1.5(M) manual. This particular power supply is capable of up to 36V and 1.5A. Larger KEPCO power supplies are available if necessary from the inventory at the IMDL.

The control signal for the valve to be tested is sent from the analog output channel 2, DAC1OUT, on the SCB-68 breakout board.

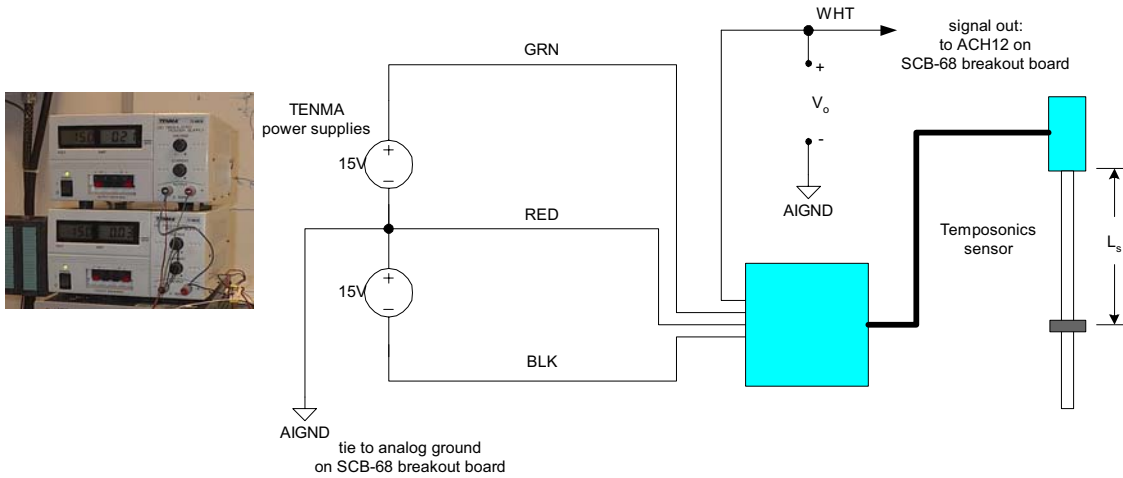
2.4.4 Temposonics Position Transducer

The power supply and signal connections for the Temposonics position transducer should be ready to use.

The Temposonics linear position transducer, illustrated in Figure 9, is powered with +/-15V. Presently, this is accomplished with two TENMA power supplies, each set at 15V and connected in series as illustrated in Figure 12.

Because the sensor has -10V to +10V output over a 20 in range, the voltage to inch conversion is 1.0V/in, and can be quickly and easily calibrated. For testing the PVG32, the length L_s in inches was computed from $L_s = (11.95 - V_o)in$ after calibration.

Figure 12: TENMA Power Supply / Temposonics sensor wiring



2.4.5 SCB-68 Breakout Board

The SCB-68 Breakout board, shown in Figure 13, provides the link between the NI-6052E A/D card in the Target PC to the external hardware. The SCB-68 is used in a single-ended configuration, where all analog voltage inputs and outputs are referenced to a common ground, enabling the use of all 16 analog input channels. The ground connections for all four power supplies (3 TENMA, 1 KEPCO) are connected to the analog ground pins (AIGND) on the SCB-68.

Figure 13: SCB-68 Breakout Board

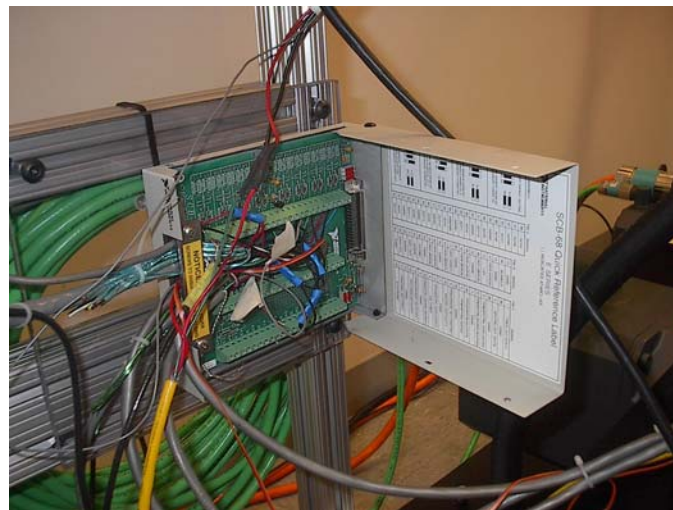
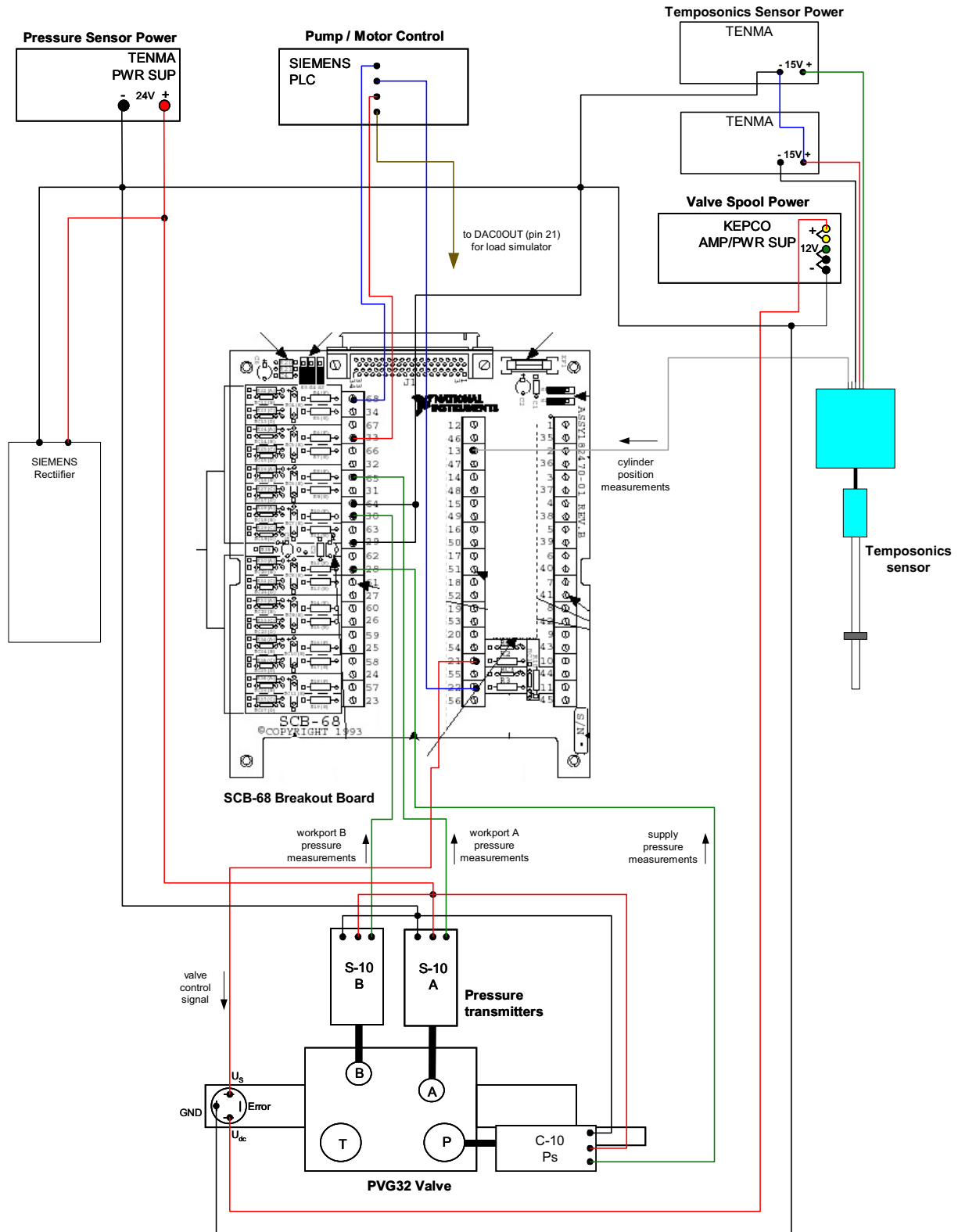


Figure 14 is a partial circuit diagram of the HIL simulator showing the relevant pin/channel assignments on the SCB-68 breakout board.

Figure 14: SCB-68 Breakout Board pin connections



Although the NI-6052E A/D card and SCB-68 breakout board combination have counter/timer circuits suitable for encoder inputs, xPC Target does not support these functions. This means that encoder signals cannot be read from any of the NI-6052E blocks in the xPC Target toolbox.

Table 1: SCB-68 Breakout board channel assignments for PVG32 valve tests

Channel	Pin Number	Type	Signal
ACH0	68	AI	Motor 1 (pump) speed
ACH2	65	AI	Workport A pressure
ACH3	30	AI	Workport B pressure
ACH4	28	AI	Supply pressure
ACH12	61	AI	Temposonics position transducer
DAC1OUT	21	AO	PVG32 Valve control

The channel assignments used to test the PVG32 are listed in Table 1. Since the card has only two analog outputs, the second Siemens (load) motor was disconnected so that DAC1OUT could be used for controlling the valve.

2.5 Setup host-target communication

2.5.1 Configure IP connection

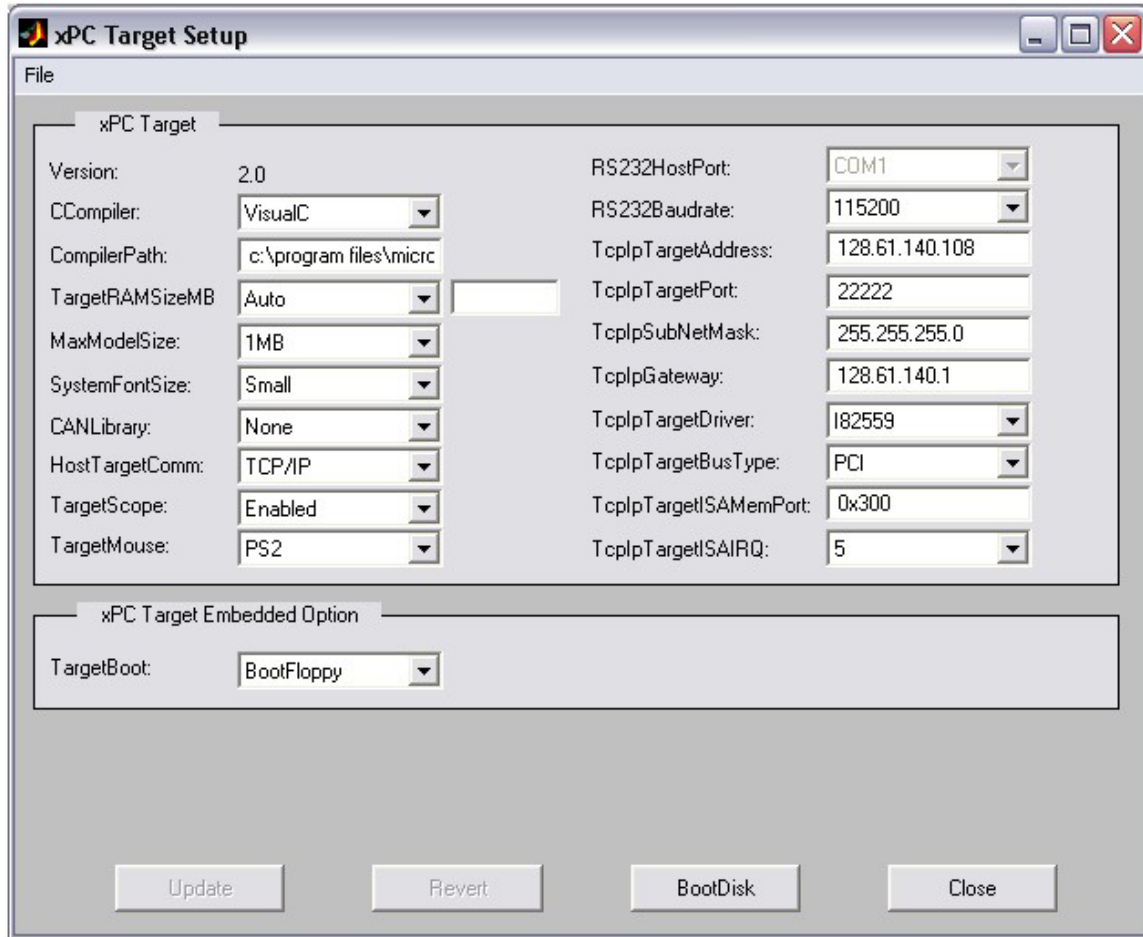
At the Matlab command window, run the command `xpcsetup`. Enter the settings shown in Figure 13.

Once the hardware has been plumbed and wired, the next step is to configure the software. Go to the Matlab command window and run the command `xpcsetup`. This will bring up the window used to set up the host-target PC communication, illustrated in Figure 13.

Figure 13 shows all the settings necessary to communicate with the Target PC. These settings should already be set in the dedicated Host PC in Love 225. However, Target applications can also be run from any computer with an internet connection and the suite of Matlab software (including Simulink, xPC Target and Realtime Workshop), and a C compiler. If MS Visual C++ is to be used, the default CompilerPath is `c:\program files\microsoft visual studio`, where the remainder of the filepath is ignored. Make sure all the settings match those in Figure 13.

After `xpcsetup` has been configured, the connection between computers can be checked using the command `xpctest`. This command will reboot the Target PC remotely, and run a series of tests to verify whether or not the communication link has been successfully configured.

Figure 13: Host-Target communication setup window opened with `xpcsetup`



2.5.2 Create boot disk

Put a blank, formatted 3 1/2" floppy disk into drive A: | of the Host PC and press the BootDisk button.

Once `xpcsetup` has been configured, create a Boot Disk on a 3 1/2" floppy from the button at the bottom of the setup window. This will install the xPC operating system onto the disk, along with the necessary host-target communication data.

2.6 Program Simulink / xPC Target model

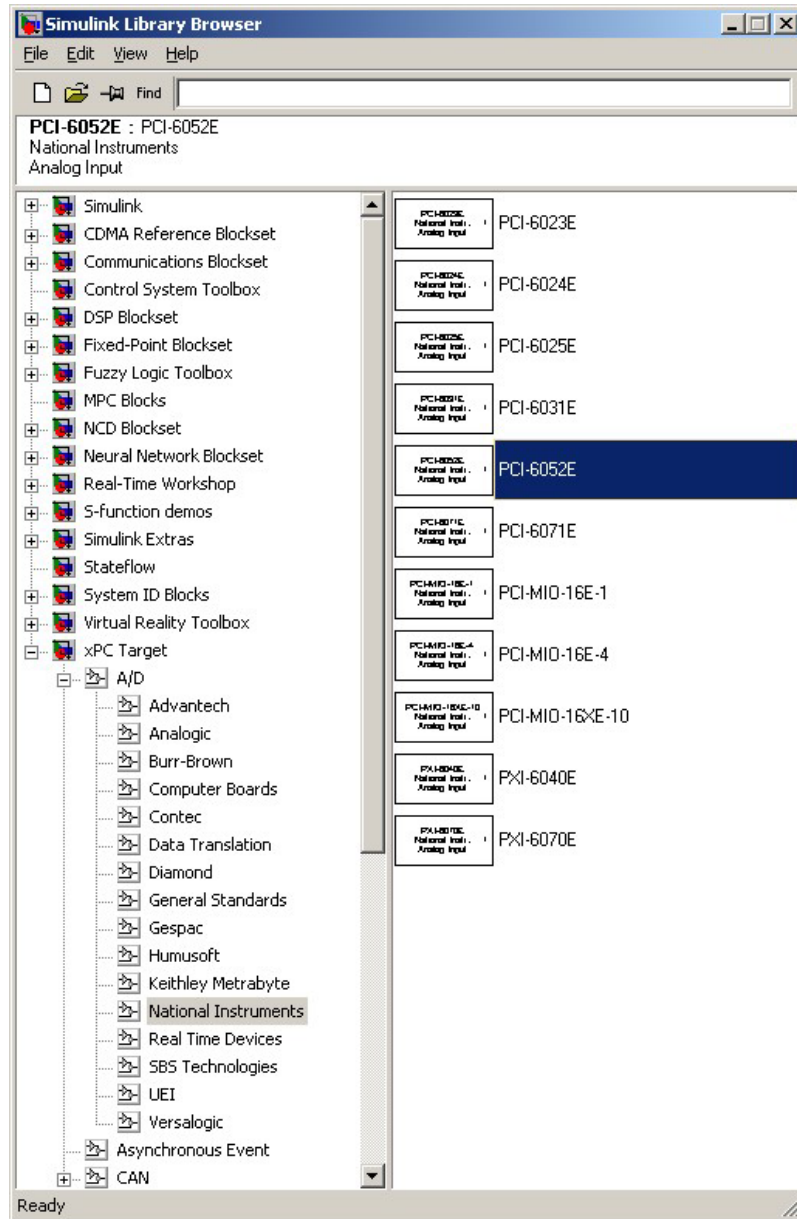
2.6.1 xPC Target Blocks

Add xPC Target blocks to the Simulink model and configure them to interface with the valve, motor, and sensors.

It is assumed that the user has had some experience with Simulink programming. The control program for the HIL simulator is written graphically like any other Simulink model¹, with xPC Target blocks added as interfaces between the model and the external hardware.

The Target PC in the HIL simulator is equipped with two National Instruments NI-6052-E cards. To access these cards in Simulink, browse to the xPC Target blockset at the bottom of the list as illustrated in Figure 14.

Figure 14: Accessing the xPC Target toolbox in Simulink

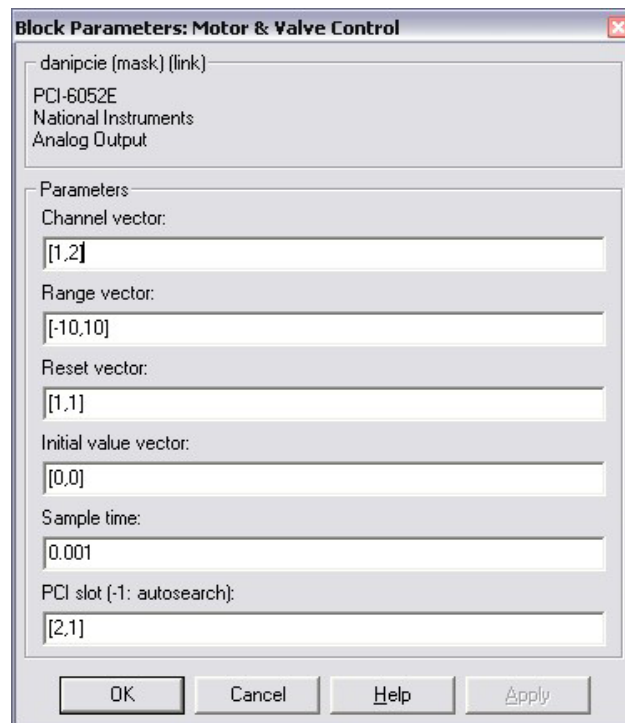


¹ NOTE: S-Functions written in Matlab code (m-file format) are not supported by Real-Time Workshop and therefore cannot be used for real time control applications. See the S-function builder in the Simulink browser located at *Simulink > User-Defined Functions > S-Function Builder* to build real-time S functions.

The NI-6052E is accessed through a combination of one, two, three, or four xPC Target blocks, depending on the signal types being sent and received. Separate blocks are available for Analog Input, Analog Output, Digital Input, or Digital Output. The location of the Analog Input block is illustrated in Figure 14. The other three options are found similarly.

The Analog Output block is found under *xPC Target > D/A > National Instruments > PCI-6052E*. To program the analog output functions, add this block to the Simulink model and double click on it. This will bring up the window illustrated in Figure 17.

Figure 17: Analog Output Block Parameters



The channel vector contains the list of channels to be used. Since the 6052E has only two analog outputs, only a [1], [2] or [1,2] can be entered in this field.

NOTE: When entering Channel vectors, xPC Target always numbers channels beginning with a 1, regardless of how the manufacturer numbers them. Therefore, DAC0OUT on the 6052E corresponds to D/A channel 1 in xPC Target, DAC1OUT corresponds to D/A channel 2, ACH0 corresponds to A/D channel 1, ACH1 corresponds to A/D channel 2, etc.

The Range vector defines the range of voltages to be sent from each channel, where the first element of the range vector corresponds to the first element of the channel vector, etc. Table 2 is a list of ranges and corresponding range codes.

Table 2: D/A Range Codes

Output Range (V)	Range Code	Output Range (V)	Range Code
-10 to +10 V	-10	0 - 10 V	10

For example, if the first channel is -10 to +10 volts, and the second channel is 0 to 5 volts, enter [-10, 5].

The Reset vector and Initial value vector should contain 1's and 0's respectively, equal to the number of output channels being used, as illustrated in Figure 17.

The PCI slot field defines the location of the NI-6052E card in the xPC Target computer. The first number defines the Bus number and the second number defines the slot on that Bus. Figure 17 illustrates accessing the card in Bus 2, slot 1. The other card is located in Bus 2, slot 12. To see the available cards and their Bus numbers and slot numbers, go to the Matlab command window and run the command `getxpcpci`.

Figure 18: Analog Input Block Parameters

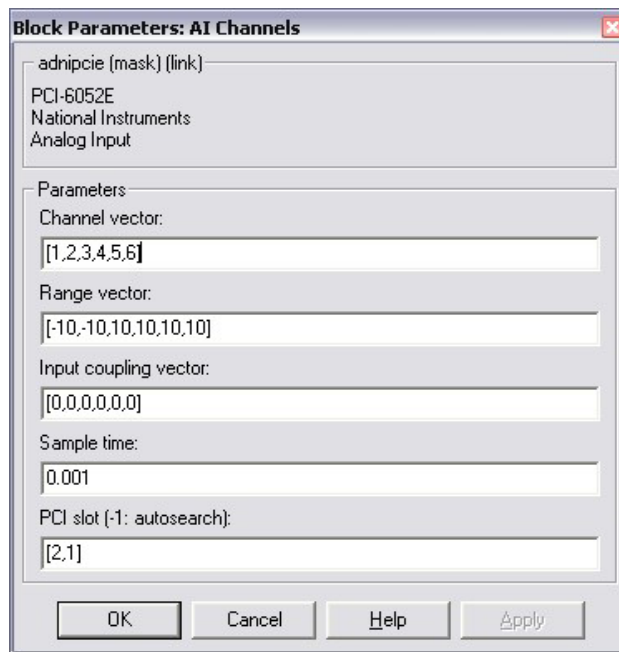


Figure 18 illustrates the block parameters of the Analog Input (A/D) xPC Target block for the NI-6052E card. The fields are programmed similar to the Analog Output block illustrated above, except that up to sixteen channels are available, and each channel has a larger choice of analog input voltage ranges. The range codes for the D/A block are illustrated in Table 3.

The Input coupling vector should contain a number of zeros corresponding to the number of input channels to be read.

Table 3: A/D Range codes

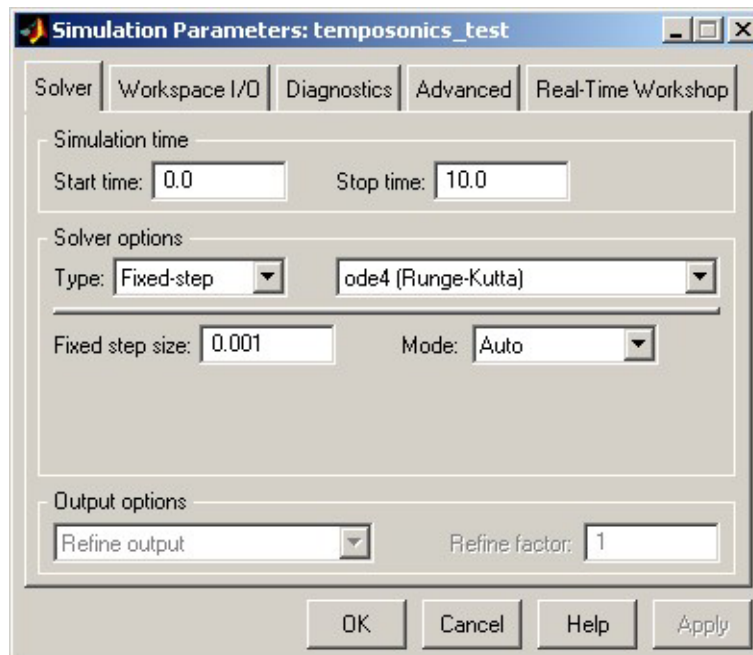
Input Range (V)	Range Code	Input Range (V)	Range Code
-10 to +10	-10	0 - 10	10
-5 to +5	-5	0 - 5	5
-2 to +2	-2	0 - 2	2
-1 to + 1	-1	0 - 1	1
-0.5 to +0.5	-0.5	0 - 0.5	0.5
-0.2 to +0.2	-0.2	0 - 0.2	0.2
-0.1 to +0.1	-0.1	0 - 0.1	0.1
-0.05 to +0.05	-0.05		

For more detailed information about setting the xPC Target block parameters for the NI-6052E A/D card, including the digital I/O features, go to the Mathworks online support located at <http://www.mathworks.com/access/helpdesk/help/toolbox/xpc/xpc.shtml> and scroll down the contents to *National Instruments > PCI-6052E*.

2.6.2 Simulation Parameters

Set the simulation to fixed step, set the sampling rate and execution time, and set the Target application to xPC Target.

Figure 19: Simulation Parameters/Solver



The simulation parameters must be configured properly or else Real Time Workshop will not compile the program. To set the simulation parameters, select *Simulation > Parameters...* from the top of the Simulink model window. Choose the *Solver* tab, which should bring up the window illustrated in Figure 19.

Set the *Solver options > Type* to "Fixed-step". Set the *Fixed step size* in seconds, which defines the sampling period for the program—ie setting the step size to 0.001 would sample at 1kHz, etc. The *Stop time* in seconds can be set as desired, or if a continuously running program is desired, enter "inf". The Remote Control Tool will allow for starting and stopping the program as desired later.

Now select the Real-Time Workshop tab, which brings up the window shown in Figure 20.

Figure 20: Simulation Parameters/Real-Time Workshop

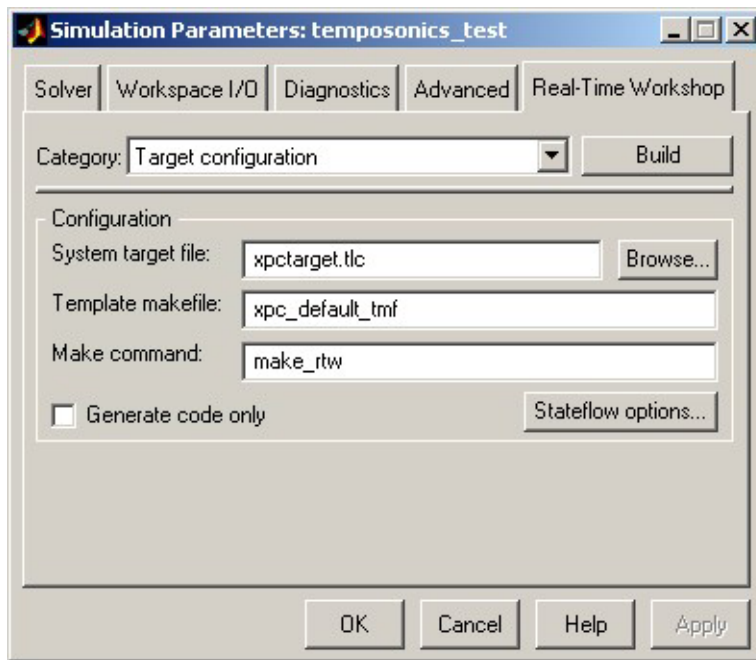
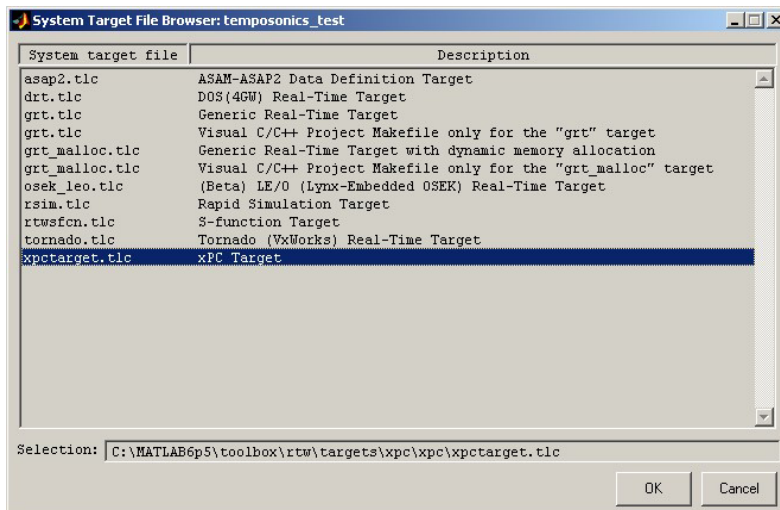


Figure 21: System target file



In the *Category* field, select "Target Configuration". Then click the *Browse* button next to the *System target file* field. This should bring up the window shown in Figure 21. Choose *xPC Target* from the bottom of the list and click OK. The System parameters window should now look like Figure 21. Click OK to exit the Simulation Parameters window.

2.6.3 Data Acquisition

Add Host and Target Scopes to the Simulink model to acquire data from the simulation.

Data is acquired from the application using Host Scopes and Target Scopes. xPC Target Scope blocks must be added to the Simulink model to acquire data. From the xPC Target toolbox at the bottom of the list in the Simulink browser window, select *xPC Target > Misc. > Scope (xPC)*, and drag it into the Simulink model. Double click on this block to bring up the parameters window shown in Figure 22.

Figure 22: xPC Host/Target Scope block parameters

The image shows a dialog box titled "Block Parameters: Ouput data" with a close button in the top right corner. The dialog contains the following fields and controls:

- Scope number: 1
- Scope type: Target (dropdown menu)
- Scope mode: Graphical redraw (dropdown menu)
- Grid
- Y-axis limits: [0,10]
- Start scope after download
- Number of samples: 1000
- Number of pre/post samples: 0
- Interleave: 1
- Trigger mode: FreeRun (dropdown menu)
- Trigger signal: 1
- Trigger level: 0.0
- Trigger slope: either (dropdown menu)
- Trigger scope number: 1
- Sample to trigger on [-1 for end of acquisition]: 0

At the bottom of the dialog are four buttons: OK, Cancel, Help, and Apply.

xPC Scopes can be either Host or Target type. A Host scope will provide access to data acquired from the real time application running on the Target PC. Data is sent back to the Host in packets defined in the *Number of samples* field. This data is not sent in real time, and will be delayed depending on the speed of the network connection. For the dedicated Host and Target PC's in Love 225, data acquisition on the Host machine can be considered nearly real time. The remaining fields can be configured to view data as desired.

A Target scope will create a plot on the screen of the Target PC, which is updated in real time. Parameters are also set in the window illustrated in Figure 22.

2.6.4 Sample Control Program

Build a complete program in Simulink to test the valve.

The following section illustrates the Simulink program written to control and test the PVG32 valve. Figure 23 illustrates the main model with three subsystems: the Plant, the Regulator, and the Command Generator.

Figure 23: PVG32 Control System main block diagram

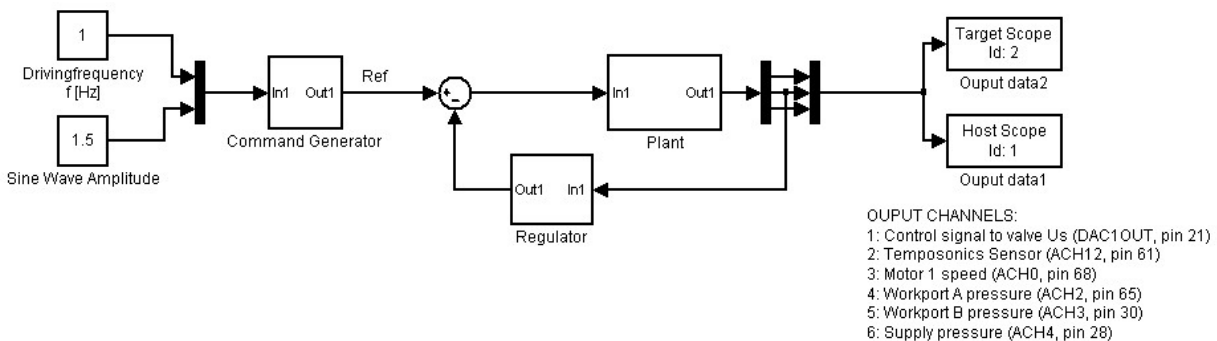


Figure 24: PVG32 Plant subsystem

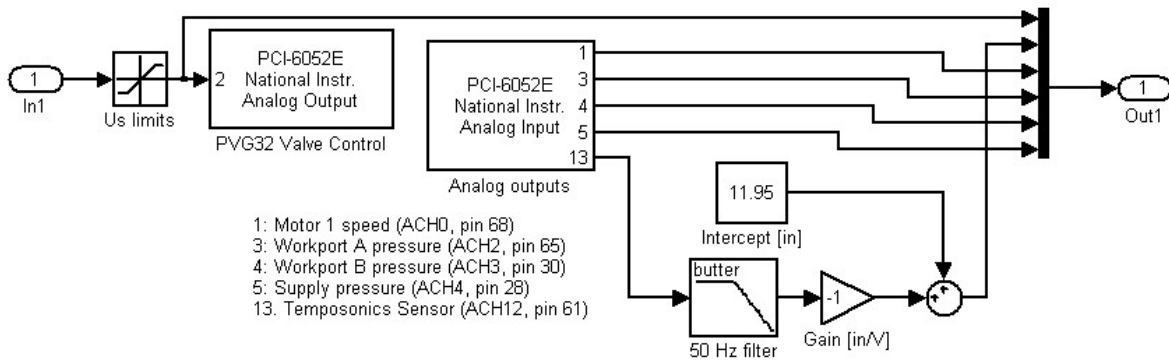


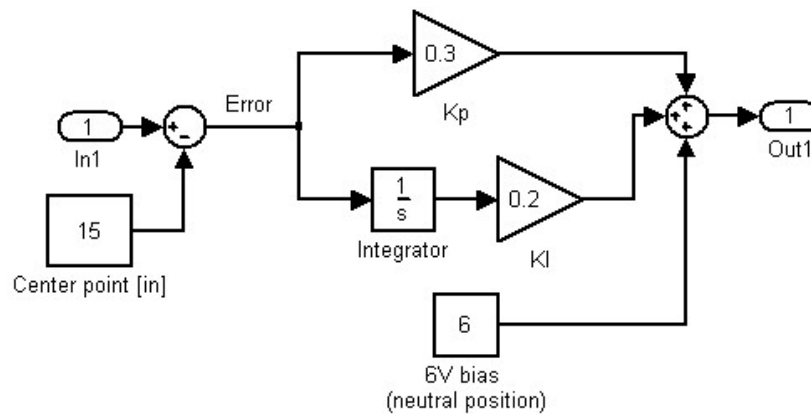
Figure 24 illustrates the Plant subsystem, which contains the Analog Input and Analog Output xPC Target blocks for the NI-6052E card. The port numbers in the Analog Input block correspond to the

analog input channels on the card, where port 1=ACH0, port 3=ACH2,... ,port 13=ACH12. See the note on channel vectors in section 2.6.1 above. Port 2 on the Analog Output block corresponds to DAC1OUT, the channel to which the control signal to the valve is sent.

Note that a low pass, 8th order Butterworth filter at 50Hz cutoff was added to reduce noise from the Temposonics position sensor. Conversion to inches was performed here as well. Also, the control signal sent to the valve was limited with a saturation block because the PVG32 required the signal to remain within a certain voltage range or an error would occur.

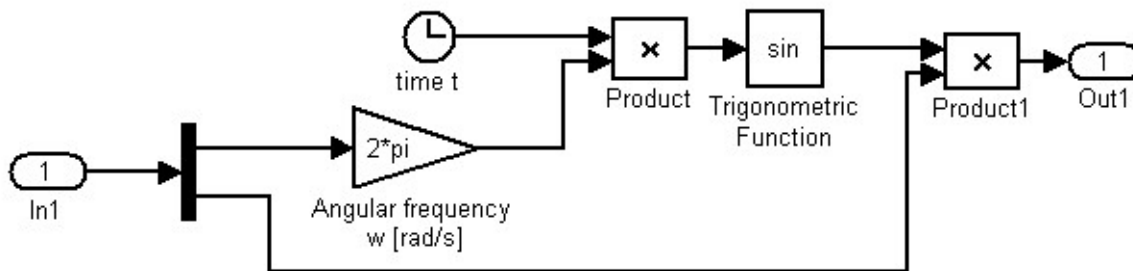
Figures 25 and 26 illustrate the Regulator and Command Generator subsystems, respectively.

Figure 25: PVG32 Regulator subsystem



The Regulator block is a PI controller used to keep the cylinder centered about an operating point. Without this block, the cylinder tended to exhibit hysteresis and quickly drifted to the end of its travel when the valve was given a sinusoidal flow command.

Figure 26: PVG32 Command Generator subsystem



The Command Generator block was used to create the sinusoidal control signal of the form

$$U_s(t) = A \sin(\omega t)$$

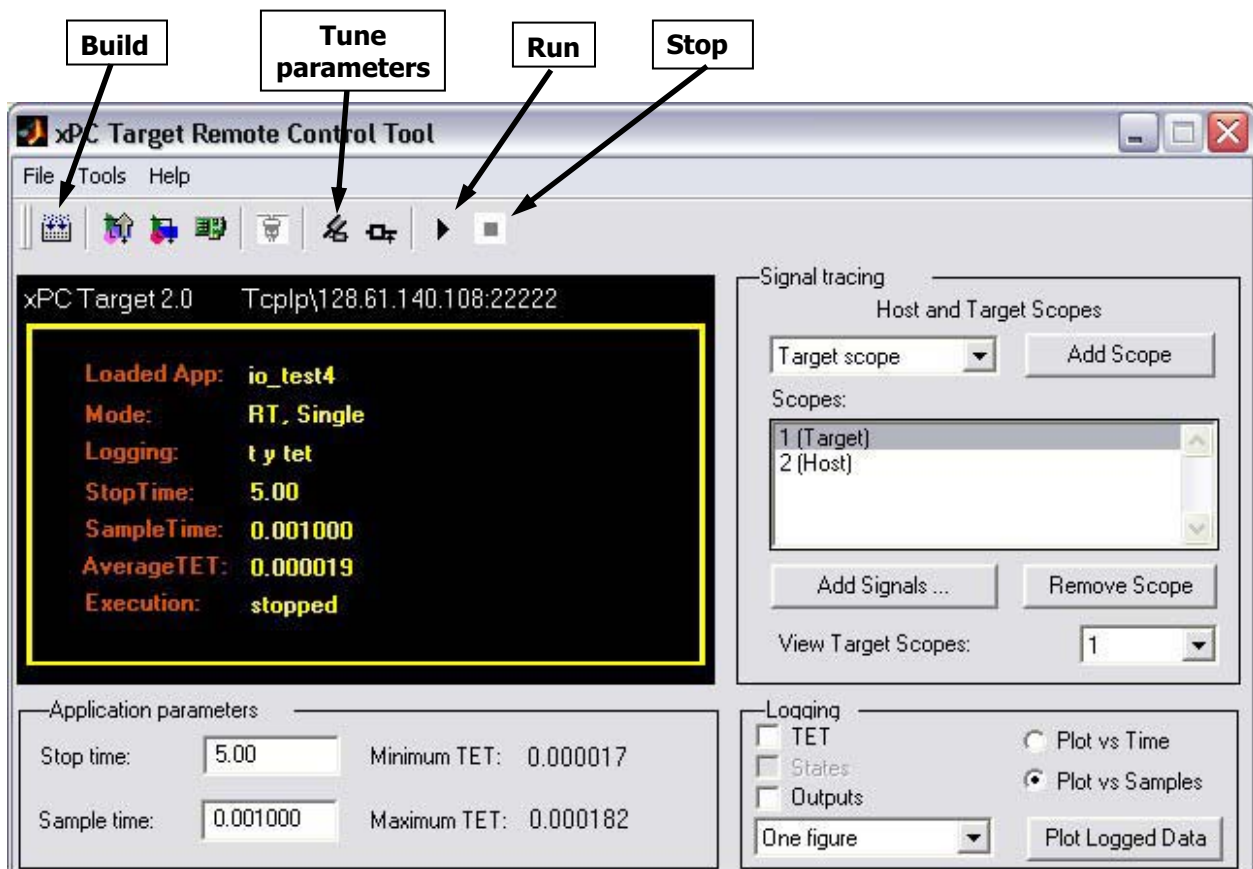
based on the inputs A and f illustrated in Figure 23.

2.7 Compile & upload Simulink / xPC Target model

*From the Matlab command window, run the command `xpcrc tool`. Press the **Build** button to compile and upload the Simulink model to the Target PC.*

The `xpcrc tool` is the primary user interface for the Host-Target PC system once the Simulink model is complete. Go to the Matlab command window and run this command, which should bring up the window illustrated in Figure 27.

Figure 27: Remote Control Tool `xpcrc tool`



To compile the model and upload it to the Target PC, click on the Build button. The Simulink model window must be open in order to build it. After choosing the model and clicking OK, the Matlab command window will display messages indicating progress as the C code is created, and indicate whether or not the Target application has been successfully compiled and loaded onto the Target PC.

3. Running a simulation

3.1 Running the Siemens motors

3.1.1 Turning on the power

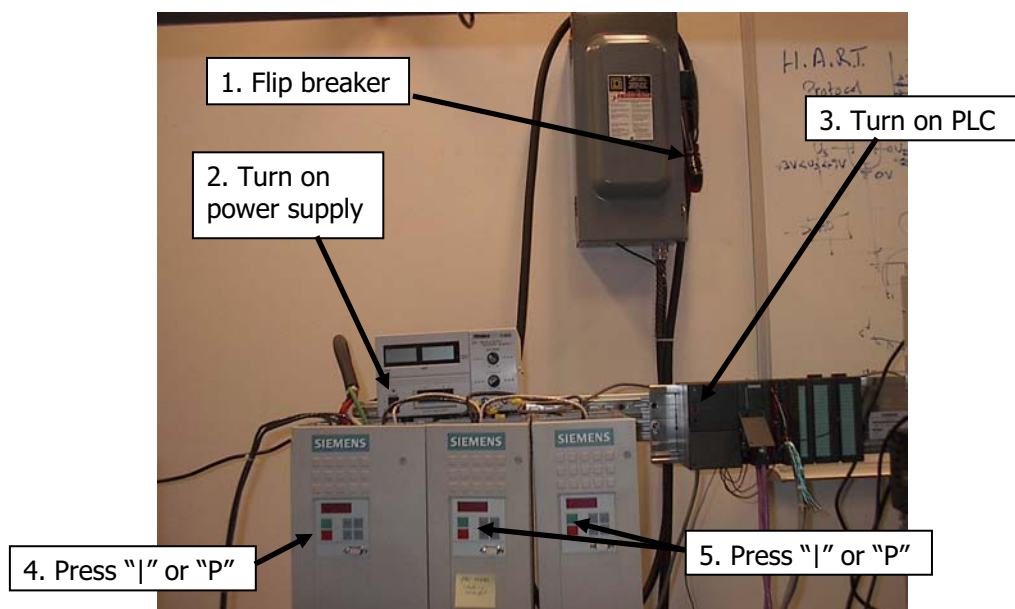
Turn on the 460V breaker switch, TENMA power supply, and PLC. Set the motor drives to run.

To turn on the power to the Siemens motors, perform the following steps:

1. Flip the main lever on the 460V breaker box to ON
2. Turn on the 24V TENMA power supply
3. Turn on the PLC
4. Press the "I" or "P" buttons on the front of the rectifier until the LED readout flickers rapidly
5. Press the "I" or "P" buttons on the front of the inverters until the LED readouts read "000"

When the motors are ready, the three small red, yellow, and green lights on the cable connectors inside the front covers of the inverters should blink steadily, and only a green indicator on the PLC should be illuminated, indicating that there are no errors and the system is ready.

Figure 28: Siemens motors power turn-on sequence



3.1.2 Running the drive software

Open DriveMon, set the motor speed, and turn on the motor.

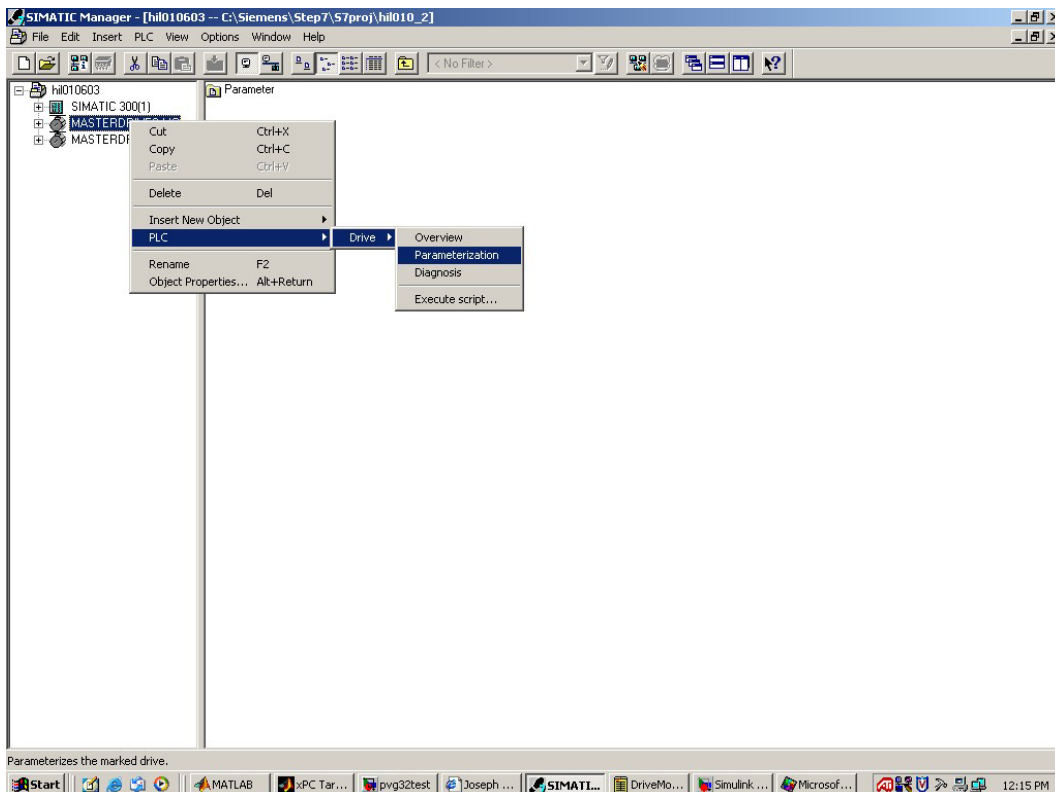
The Siemens motors are controlled with the Simatics and DriveMon software through the Profibus card in the Host PC connected to the PLC. These instructions only describe how to set the motor speed. The PLC controls the speed of the Siemens motors in a closed-loop fashion by means of integrated encoders, so that all the user needs to do is provide a reference command in the form of a percentage of the full scale speed, which is 3000rpm. The Siemens motors have many more capabilities, which are currently being explored at the IMDL, but which are outside the scope of this manual.

The percentage of full scale needed to produce the desired flow rate out of the motors can be computed from

$$R = \frac{Q_{des}}{V_m \omega_{max}},$$

where Q_{des} is the desired flow rate in gpm , l/min , etc., $V_m = 19 \text{ cc/rev}$ is the displacement of the motor, $\omega_{max} = 3000 \text{ rpm}$ is the maximum motor speed, and R is the reference command in percent. *For the PVG32 valve, 8.6gpm was needed to simulate the flow rate from the John Deere tractor. This corresponded to a motor reference command of $R=57.11\%$.*

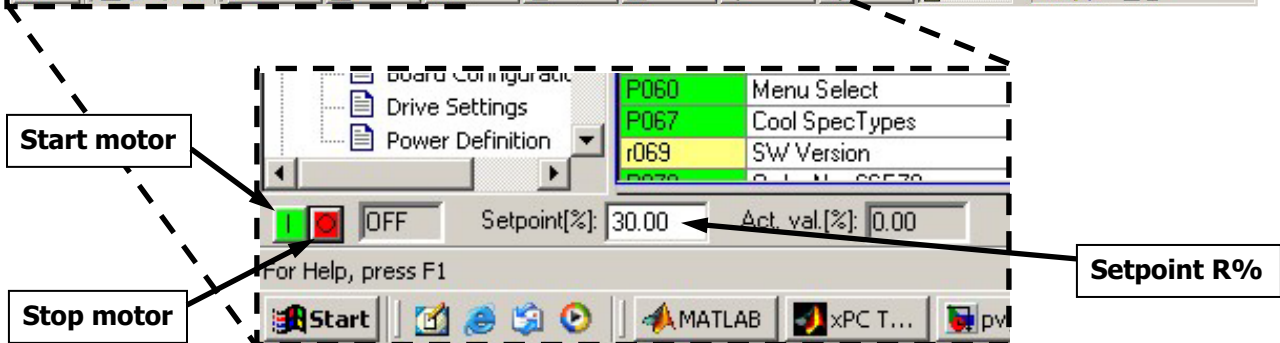
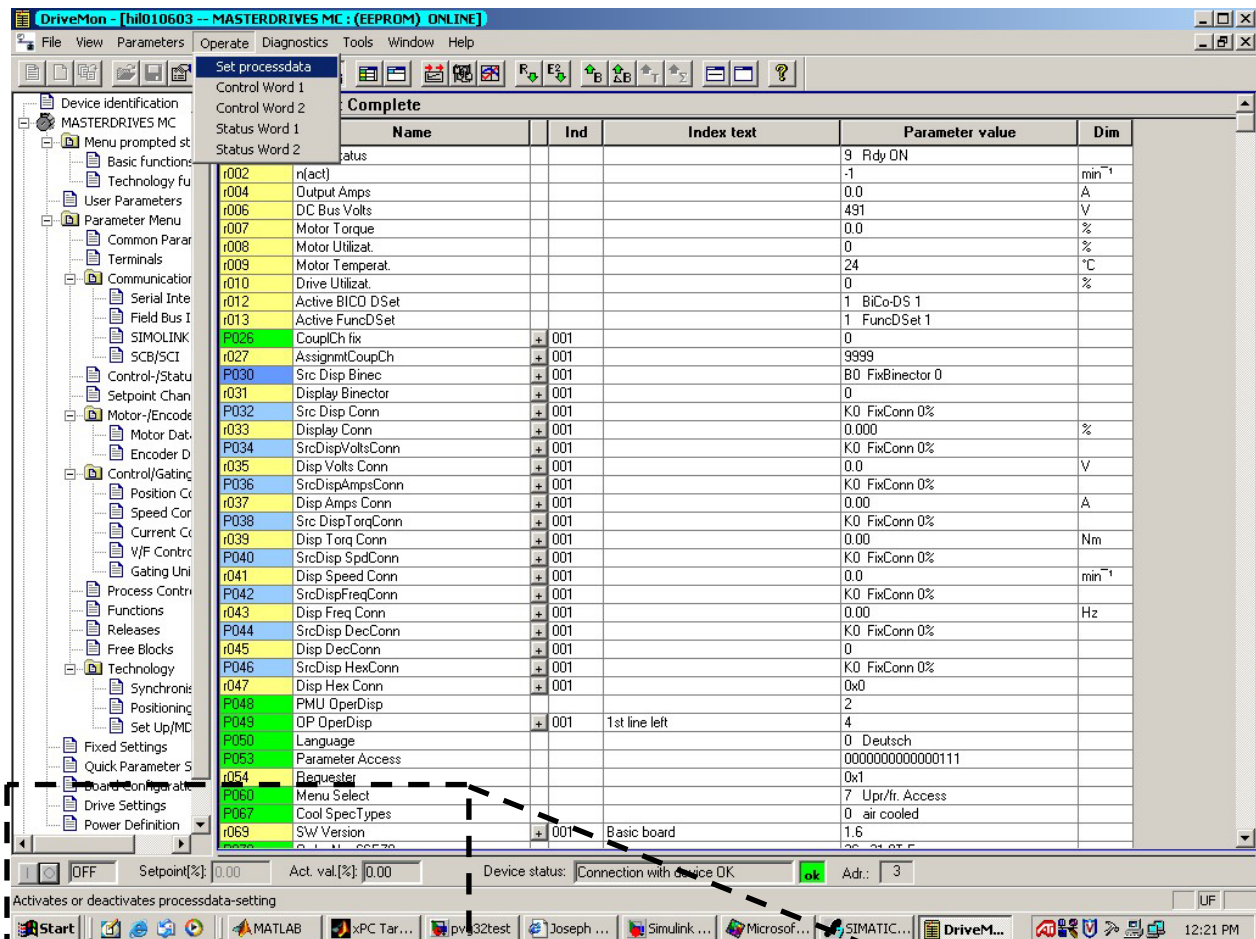
Figure 29: Accessing DriveMon from Simatic Manager



To open the software, go to the desktop on the Host PC and double click on the Simatic Manager icon. When the program opens, go to the column on the left side of the screen, and right click on the line that reads MASTERDRIVE A:. From the popup window choose *PLC > Drive > Parameterization*, as illustrated in Figure 29. This will launch the DriveMon software.

When the DriveMon program opens, click OK twice on the popup windows, until you see the screen that appears in Figure 30. From the dropdown menus at the top of the screen, choose *Operate > Set processdata*. Click OK on the next popup window. The Start motor and Stop motor buttons in the bottom left corner should now be illuminated as illustrated in Figure 30.

Figure 30: DriveMon window



Once the Start motor and Stop motor buttons are illuminated, the motor is ready to run. Set the Setpoint field to the desired value in percentage of full speed to obtain the desired flow from the motor. Press the green button to start the Siemens motor and the red button to stop.

3.2 Running the Target Application

3.2.1 Starting and stopping the application

To start and stop the target application, press the Run and Stop buttons on the `xpcrcctool` window.

The target application can be run from either the `xpcrcctool` on the Host PC, as illustrated in Figure 27, or from the Target PC by first pressing the "C" (command) key and then entering Cmd: *start* (enter), and Cmd: *stop* (enter). If the *Stop time* field in the Simulation Parameters window was set to a number, the target application will stop running when the time has elapsed. See Figure 19. If the *Stop time* was set to "inf", the target application will run indefinitely until otherwise commanded to stop. The value of the *Stop time* field is displayed in the `xpcrcctool` window.

3.2.2 Adjusting model parameters

To adjust model parameters at run time, click on the Edit Model Parameters button in the `xpcrcctool`, and then double click on the block in the diagram to access the parameter.

Model parameters such as controller gains, waveform amplitude and frequency, etc. can be adjusted on the fly using the *Tune Parameters* button in the `xpcrcctool`. The Simulink model will open in a new window, where the parameters of the blocks can be changed simply by double clicking on them and then changing the values. This is a very powerful capability that allows for tuning the system without the need to recompile between each adjustment.

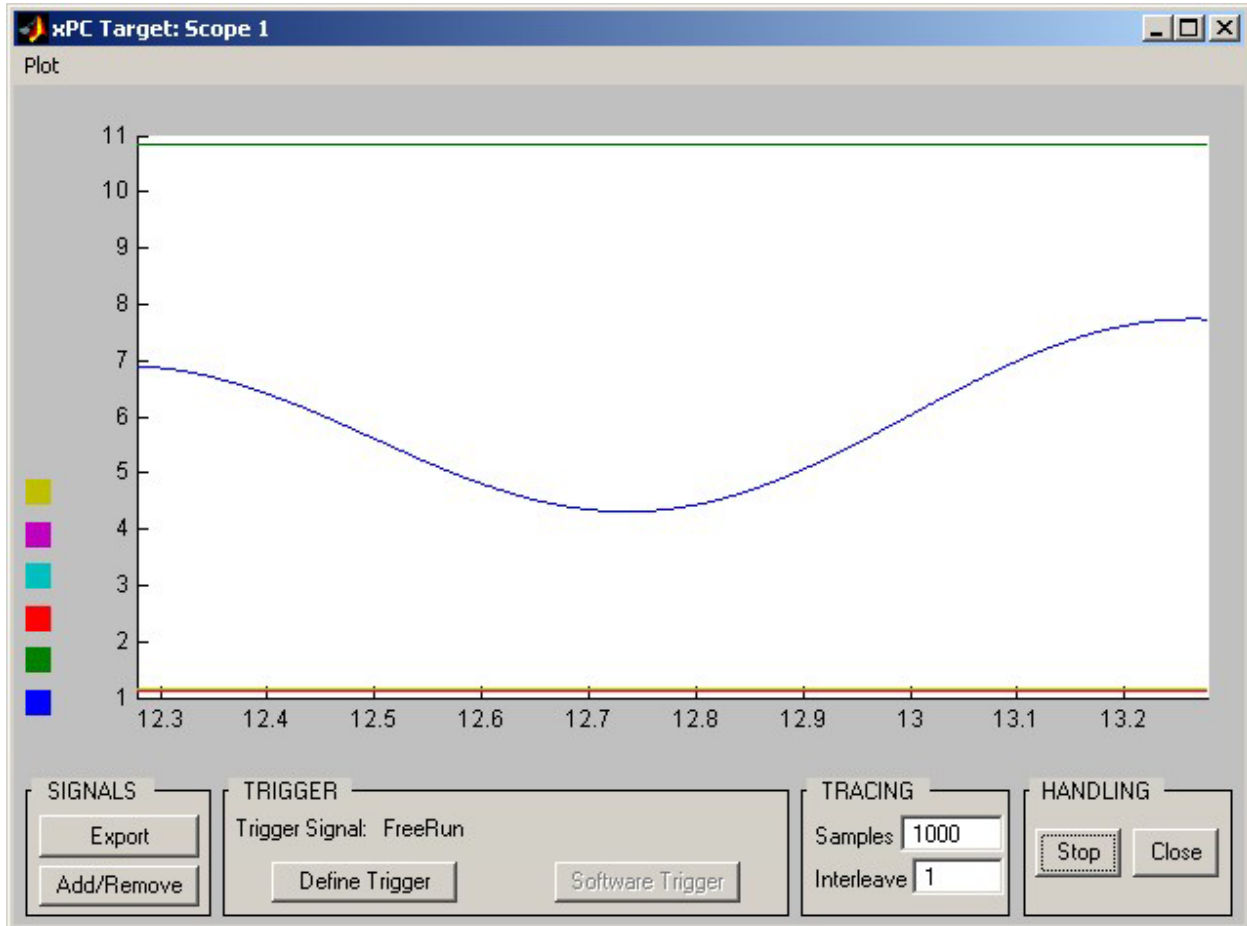
3.2.3 Viewing Host Scopes

To view simulation data on the Host PC, go to the `xpcrcctool` and select Tools > Host Scope Manager.

If a Host Scope has been included in the Simulink model, data sent to it can be viewed on the Host PC at run time by selecting *Tools > Host Scope Manager* from the `xpcrcctool` window. The Host Scope automatically plots the data packets that are sent to it. The size of the data packets the scope displays is defined by the *Number of samples* field illustrated in Figure 22. Note that the time period of

the data displayed on the Host Scope depends on both the *Number of samples* and the sampling rate. The Number of samples may also be changed at run time from the Host Scope window in the block labeled *TRACING > Samples*. Figure 31 illustrates the Host Scope window.

Figure 31: Host Scope window



3.2.4 Acquiring Data

Once the data is captured on the Host Scope, press the Stop button, then the Export button. Data will be saved to the Matlab workspace.

When the Host Scope is displayed, the data plot is refreshed when each packet of data is transferred from the buffer in the Target PC, at which time the previous data packet is overwritten. To save the contents of the current screen's data to the Matlab workspace, click on the *Stop* button, and then on the *Export* button. Two vectors will be created in the workspace with a number of rows equal to the *Number of samples*, one for the time vector and another matrix of the data, with each row corresponding to the data value at each time step, and each column representing each signal on the scope. The default time and data variable names are *scope1_time* and *scope1_data* respectively, which

may be changed by selecting *Plot > Variable Name for Export*. The data in the two vectors are overwritten each time the *Export* button is pressed, so the output variable names should be changed between exports if it is desired to log multiple data sets.

After the *Stop* button is pressed, it changes to *Start*, enabling more data to be captured. Data may not be exported while the Host Scope is running.

Also, if the *Stop time* (Simulation Parameters) is set to a finite number, the last *Number of samples* can also be exported directly from the `xpcrcctool` window by using the *Plot Logged Data* button, as illustrated in Figure 27.

4. Resources

Mathworks Online support: Matlab, Simulink, and xPC Target

<http://www.mathworks.com/access/helpdesk/help/helpdesk.shtml>

National Instruments: DAQ Cards

PCI-6052E A/D card: <http://sine.ni.com/apps/we/nioc.vp?cid=2601&lang=US>

SCB-68 Breakout Board: <http://sine.ni.com/apps/we/nioc.vp?cid=1180&lang=US>

Also see printed documentation and CD-ROM available at the IMDL

Siemens Motors

See the General Motion Control instructions on CD-ROM available at the IMDL



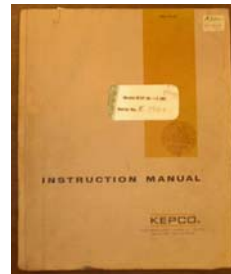
Temposonics Linear Position Transducer

<http://www.products.mtslinearsensors.com/namedPageViewer.php?keyword=dstyle&PHPSESSID=027da180a0cb1bd943020d0a81c1324f>

KEPCO 36-1.5(M) Power Supply

<http://www.kepcopower.com/ate.htm>

Also see operator's manual available at the IMDL



TENMA 72-6628 Power Supplies

<http://www.tenma.com/>

Hydac Hydraulic Filters, Check Valves

<http://www.hydacusa.com/>