Shore Western Control System
SWCS

Executive Summary

Features and Benefits of the Shore Western Control System

The Shore Western Control System (SWCS) is an industry-leading controller, with features and benefits that exceed any competitive product. The system is designed, from the ground up, with unprecedented customization and controls that result in accuracy, stability, and usability that cannot be matched by our competition.

Customization

Every lab is different. Every test is different. Our competition tries to meet the needs of everyone with one single user interface. Our approach is unique. We provide the flexibility and the tools to be able to design screens and panels that meet the needs of individual labs, and individual tests. Sliders, meters, labels, lights and buttons can be located and defined on custom panels, and a button can operate a Block Program. This is a program that is graphically built using a series of steps, defined in a flow-based diagram, that can perform simple or very complex multi-threaded tasks. You may define a button to turn on a hydraulic manifold, or you may define it to perform a complete test.

Once the user-interface has been built, operators simply select from a menu of pre-defined screens that conveniently provide them with all the parameters they need to run their test, without presenting them with things that they do not need. The user interface can be built by Shore Western engineers, or you may choose to build them or modify them yourself.

No other controller in the industry has the programming capability of the SWCS, and it provides the basis of a highly configurable, and automated system.
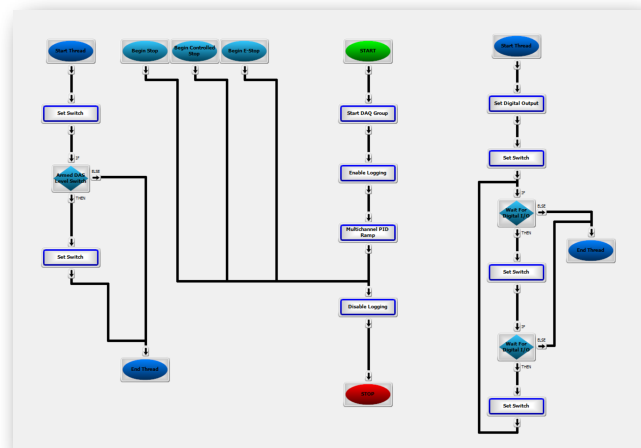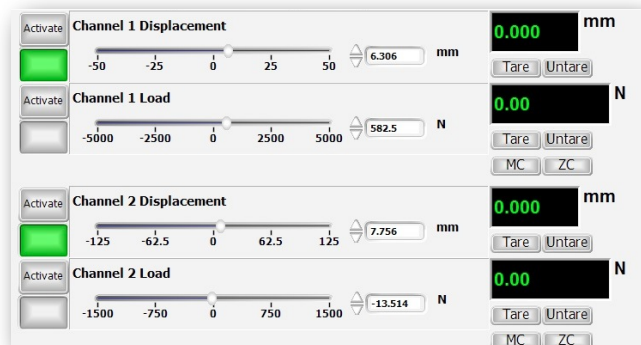
Digitally Optimized Analog Control

The SWCS uses analog control for the servoloop that is set up, operated, compensated, and managed digitally. The digitally optimized analog loop provides control with unprecedented stability, accuracy, resolution, and configurability. With the Shore Western controller, there is no concern for loop-closure rates or bit resolution. These are both infinite. However, all the benefits of direct digital control are still there, because the loop is tuned and optimized digitally. This literally provides the best possible solution.
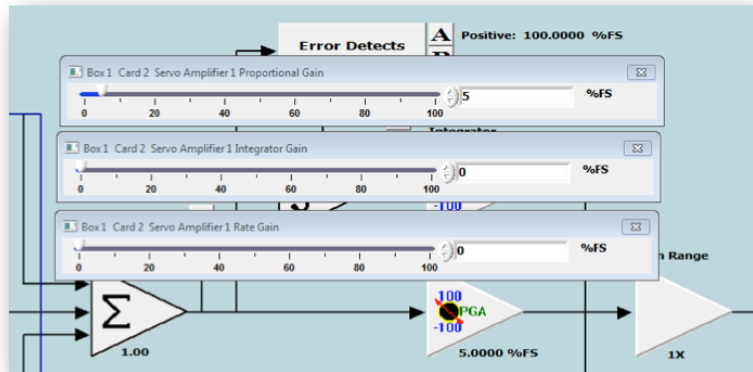
The set up and operation of the controller is performed using a PC, running Windows 7®. The servo loops themselves are set up using a graphical display that provides a clear, easy to understand representation of their configuration. All the servo loop parameters can also be adjusted using the custom panels, so that one single screen can be created for tuning the loop, for example, complete with a function generator, PID controls, and scope. This keeps the operators away from any unnecessary details that can lead to dangerous errors.

Outer-Loop Third Party Controls

Another significant benefit of digitally optimized analog control, is that it is very easy for you to add your own outer loop controller. For example, if you have developed your own hybrid simulation code, and you would like to use a dSPACE controller, it is easily interfaced to the SWCS using an external analog input. We provide the conditioned analog transducer outputs. You do not have to worry about digital protocols, latency, or quantization errors. The signal from your outer-loop controller will be fed directly to the actuator, and you will get the analog response back. A very significant and unique benefit.

Real Time Active Control

Shore Western has developed a proprietary algorithm for performing digital optimization. It is called Real Time Active Control (RTAC). This is an algorithm that takes your target signal, which could be a sine wave or a realtime signal, such as an earthquake or road profile, and reproduce it, point-by-point, with high accuracy. RTAC is set up via a simple tuning screen. Once the gains are established, reproduction of the signal can be performed quickly, and with very little effort. Contrast this with competing algorithms, that use the long process of system identification and iteration to perform the same task. Typically, frequency-domain-based iterative techniques can take days to set up, and require highly experienced controls engineers. RTAC takes minutes, and can be done by anyone.
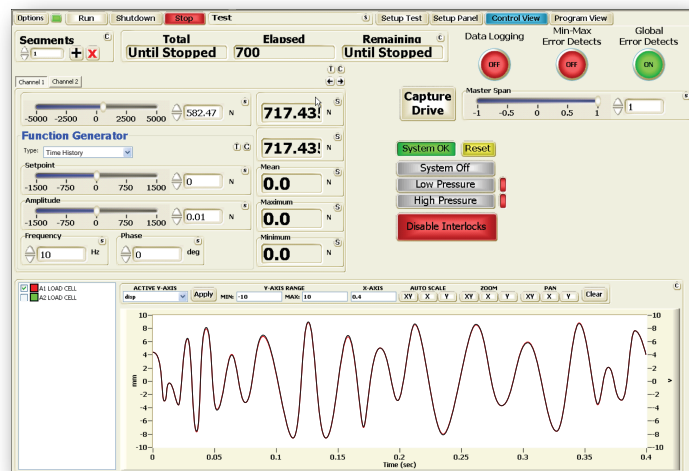
RTAC can be run in a "mixed mode." This means that you can select one transducer for the actuator closed-loop, and another for the RTAC closed-loop. For example, users could run the actuator in displacement control, while reproducing acceleration, load or strain with RTAC.

RTAC is a very powerful industry-leading algorithm, that has proven to be successful across a broad range of different test applications.

Stations

The SWCS is a multi-station controller. Station configuration is easily defined and managed directly in the application. There is no need to launch a separate "builder." All the resources that are in the system and assigned to a station can be easily viewed at any time, in the Resource Tree. There is no theoretical limit to the number of stations that can be defined[1]. Resources are allocated to stations via a simple station setup dialog. There is no need to shut down the software and launch another application. Stations are completely independent from each other, with their own interlock system. A master E-Stop is there to shut off all stations should a catastrophic event occur.

The software is designed to work across physical chassis boundaries too. It is possible to define stations using channels from other networked chassis. This provides for limitless ways of configuring the stations[1]. Another unique benefit of the SWCS.
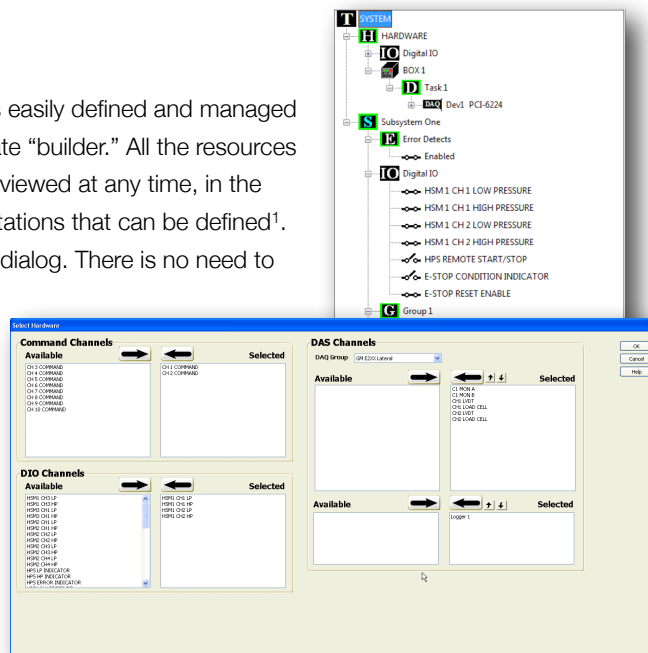
Rugged Serviceable Hardware Design

We take the design of the controls hardware very seriously. We have designed it to be both rugged, and easily serviced. The chassis is divided into two sections: one, the Servo Control section for the closed-loop controls and data acquisition, the other, called the Process Control section, for the high-current needs of solenoids. Each section has its own power supply.

The closed-loop controls use a server-grade redundant power supply, that can be hot swapped. If one power supply has problems, it automatically switches to the other (and vice-versa), and issues a warning. You can continue to run your test while you replace the defective part. For easy replacement, the individual supplies can be removed from the chassis with one simple button. They are kept behind a locked panel for security. Also behind the locked panel, is a removable hard drive. This is the system disk. It can be removed and cloned, so that you have a secure backup. If the disk drive fails, you simply swap it for its clone, and you can continue testing without interruption.
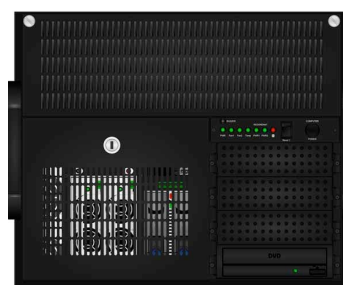
The Process Control chassis has a large robust Medical-Grade power supply that is over-sized to meet the high current needs of all the solenoids and high precision needs of external devices. The IO panel is highly configurable, and can be matched to any of your specific logic needs. The Process Control section also has an excitation module for devices that require conditioned DC power, such as LDTs. This module can provide +24VDC, ±15VDC, or +5VDC.

---

[1] Some limitations may apply, please consult SWM to discuss your application

Hardware

Chassis Layout

The SWCS chassis is divided into two sections. The top section, the Servo Control Chassis, houses the digital backplanes, and is the home of the embedded PC, servo control cards, filtering, and data acquisition. The bottom section, called the Process Control Chassis, is where all the high current IO is performed, such as switching solenoids, controlling the pump, and providing conditioned power for external devices.

The SWCS can be packaged in three different ways. For low channel-count systems, we use a small, desktop box. For medium channel counts, we use a single 19in rack-mountable chassis, and for high channel count systems, several rack mount boxes are networked together, and linked to a separate Process Control chassis.
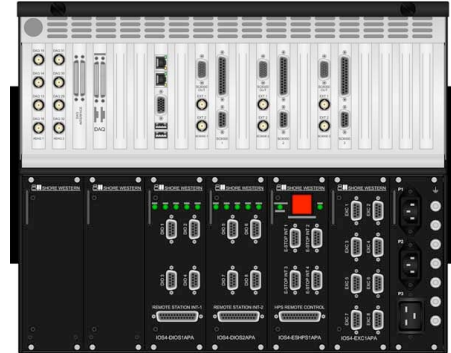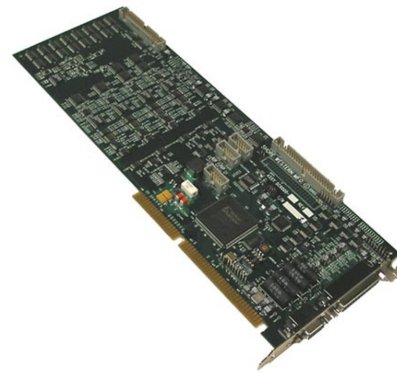
Servo Control Chassis

SC6000 Card

The heart of the SWCS is the SC6000 card. Each card has the following:

- Two Servo Amplifiers
- Four Transducer Amplifiers
- Two Valve Drivers
- Two External Command Ports
- Six Analog Outputs

Cards can be configured for the following:

- One actuator with dynamic mode switch with 2 stage servo valves (one per card)
- Two actuators without dynamic mode switch with 2 stage servo valves (two per card)
- One actuator without dynamic mode switch with 3 stage servo valves (one per card)
- One actuator with dynamic mode switch with 3 stage servo valves, and optional switching low flow valve (utilizing two cards)

Dynamic mode switch allows you to switch from displacement control to load control (or vice versa) with hydraulics on, and without a bump. This is accomplished by dedicating one servo amplifier for displacement control, and the other for load control. Each with it's own PID and set point settings. At the point when control is switched, the software is able to take the actuator's current position (or force) and adjust the other servo amp to match it, so that the switch is seamless.

If dynamic mode switch is not used, the switch from displacement control to load control must be performed with the hydraulics off.

Three stage servo valves require a dedicated servo amplifier to control the spool of the servo valve. All of the resources of a single card are required to perform that control. However, if the three stage servo valve is used in combination with dynamic mode-switch, we can add a second card. The first card's two servo amplifiers are used to provide the dynamic mode switch, and the second card's first servo amplifier is used to close the loop for the spool. This leaves an extra servo amplifier and valve driver on card 2, which can be used to perform a bump free transfer to a low-flow servo valve on the same actuator (if required).

Transducer Amplifier Specifications
- Software configurable to condition AC transducers, Wheatstone Bridge transducers (e.g. load cells or pressure transducers), high-level voltages, or 4-20 milliamp transducers
- Software selectable gain ranges: .5x, 1x, 2x, & 4x, hardware selectable gain range of 1x or 100x (this may be configured differently on request)
- Software selectable polarity switch
- Software adjustable offset up to ± 2.5v
- Software adjustable demodulator for AC transducers (phase and offset)
- Software adjustable excitation 0-10V for DC, and 2.5KHz or 10KHz at 0-20V peak-to-peak for AC

Servo Amplifier Specifications
- Software configurable for P, PD, PID, or PIDF with selectable inputs for the command
- Infinite resolution and bandwidth via stable digitally optimized analog control
- Differential pressure stabilization
- Software spans on two software selectable external inputs for operation of each servo amplifier from an external source

Valve Driver Specifications
- 100mA maximum output or Voltage Driver (hardware switchable)
- Optional valve booster for high current requirements
- Software adjustable valve balance. Range: up to 20% of maximum servo valve current
- Dither amplitude and frequency is software selectable for multiple frequencies between 100Hz and 1000Hz
- Bumpless mode switch support (when applicable)

Each card also has 32 channels of discrete IO for devices such as solenoids valves, pumps, chambers, etc. However, these IO channels are internally wired at the Shore Western factory to the Process Control Chassis. Only the channels assigned to the Process Control Chassis are available to the user.

Embedded PC

The SWCS is operated through an embedded PC running Microsoft® Windows 7. This is a standard PC, with the following specifications:

- Intel® Core™2 Duo processor
- 2GB DDR2 RAM
- Dual Gigabit Ethernet LAN ports
- VGA port
- 4X USB ports
- Slim DVD±RW
- Removable 500GB Hard Drive Module

Data Acquisition

The SC6000 provides six analog output signals. Four are high-level conditioned outputs from the transducer amplifiers, and the other two are from selectable monitor points on the card. Using a graphic depiction of the SC6000 card, users can select any two points on the circuit to monitor, and the signal will be presented at the selected monitor output.

These six signals are routed through hardware adjustable anti-aliasing filters, which may be bypassed. The filters are typically set to a breakpoint of ~450Hz, and are 6-pole Butterworth. The signals are then routed to a BNC breakout cable, and a National Instruments (NI) PCI-based data acquisition card. Depending upon system configuration, there may be additional high-level inputs available for user inputs to the NI card. These are usually made available through a BNC access panel on the back of the SWCS. Note: by default, the factory ships with the monitor point anti-aliasing filters disabled. This is to allow set up of LVDT demodulators via an external scope.

Redundant Power Supplies

The Servo Control Chassis is supplied via a server-grade redundant power supply system. Two power supplies are used, each rated at 550W. If one supply fails, a warning is issued, and the other supply takes over. Either supply can fail without interruption. The supplies are located behind a locked panel for security. Once the panel is unlocked with a key, the failed supply can be easily removed and hot swapped.

Server Grade Removable Disk

The system Hard Disk Drive (HDD) is also located behind the same locked panel as the power supplies. It too can be easily removed, although in this case the computer must be shut down first. The HDD can be removed and cloned onto a backup so that, if you have a failure of the HDD, you will be able to swap it quickly, minimizing test downtime.
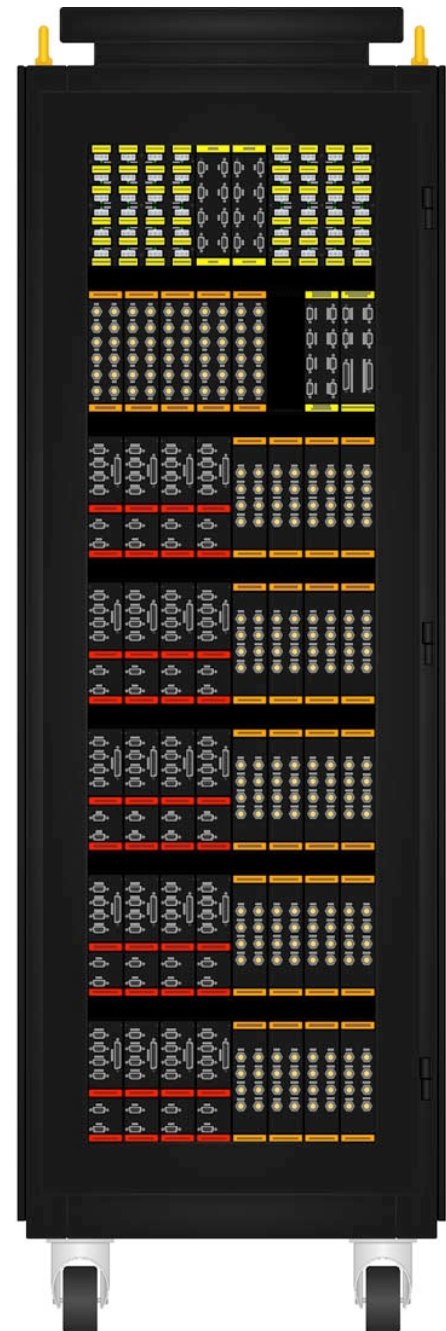
Process Control Chassis

Medical Grade Power Supply

The Process Control chassis incorporates a best-in-class highly regulated and stable power supply that provides +24VDC, ±15VDC and +5VDC to a proprietary bus for hot-swappable IO modules. The power supply can be connected to 100-240VAC at 50-60Hz, and the output wattage varies with chassis configuration. The smallest chassis uses a 400W supply, and the largest supplies can be as high as 1500W.

E-Stop and Pump Control Module

The E-Stop and Pump Control Module contains the relays and logic circuit for a failsafe hardware interlock on emergency stop. The E-Stop pulls power from every relay in the system (unless overridden via a hardware configuration), effectively opening all of the digital outputs. The net result is an immediate shutdown of all the solenoids attached to the controller. If a pump is connected, it will shutdown, along with any HSMs etc. The only way to reactivate the relays, is to clear the E-Stop condition, and press a reset button.

The module also contains the IO logic required to control a pump system. It includes the flexibility to be configured for different vendor's protocols, allowing the SWCS to control a wide range of pumps.

Excitation Power Module

The Excitation Power Module routes the supply voltage to 8 output connectors for use with external devices that require power, such as LDTs etc. The output power is jumper selectable between +24VDC, ±15VDC and +5VDC.

IO Module

The IO Module is a highly configurable to provide an interface for up to 8 logic signals. Options include:
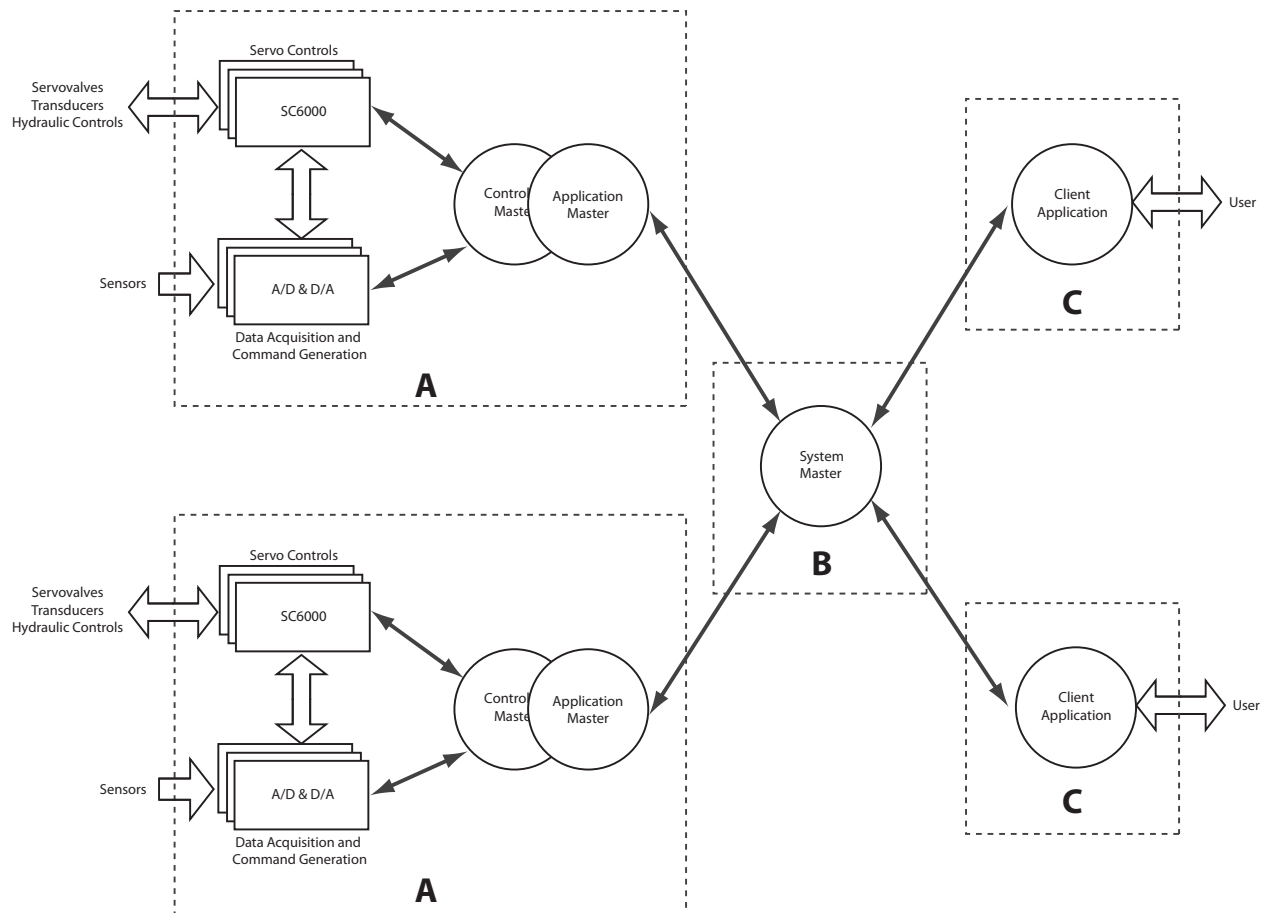
- 24VDC output for solenoids
- Dry Contact output
- Opto Relay Input (10-32VDC or 2.5-28VDC)
- Others available on request
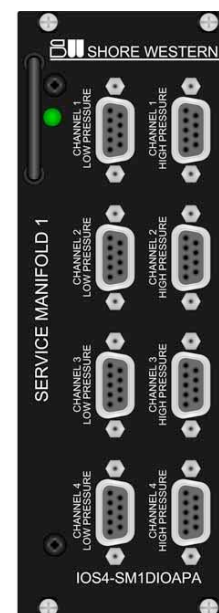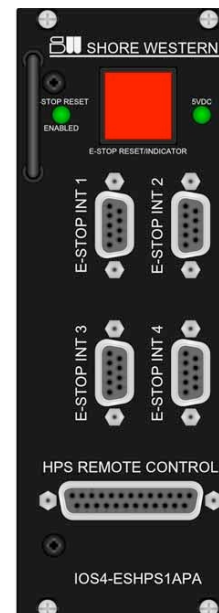
Software

System Set Up Software

Control Master, System Master and User Interface

The software is structured into three main components: the Control Master (A), System Master (B), and the Graphical User Interface (C). The Control Master directly interfaces to a given controller box, with its collection of SC6000 cards, data acquisition, and single PC. The System Master can manage several Controller Masters. This means that a single system can be comprised of multiple boxes (Control Masters). Stations are defined at the System Master level, meaning that all the resources of multiple boxes are available for assignment to a given station. The User Interface interacts with the System Master. One System Master can have several User Interfaces, running on networked PCs. This hierarchical structure provides unprecedented flexibility for system configuration.
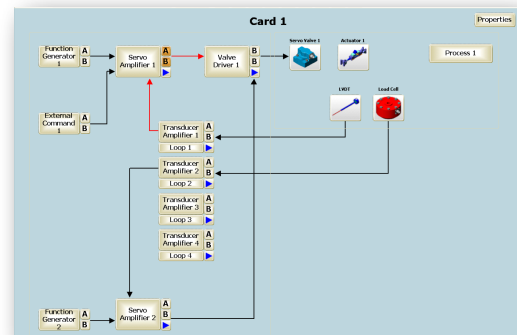
Stations

Each System Master presents a list of resources to the User Interface in the form of a Resource Tree. Command Channels, Data Acquisition, Loggers and IO channels that have not been assigned to a test station are listed in an available hardware section. A station is created via a simple right-click in the tree. Resources are then assigned to it using a Station Setup dialog. Command channels, data acquisition channels, data loggers and IO channels are selected. Depending on the configuration, a station may be on a single controller, or across several. If one PC is viewing several stations, each one can be quickly accessed via a drop-down list, all the views and settings are preserved as you switch between stations. It is also possible to assign an external PC for each station, to maximize screen real estate. Each station is independent of the others, with its own set of menus, screens, status etc. And each station has its own limits, function generation and block programs that can be started and stopped independently. Data loggers are independently defined, with their own sample rates, destination files and triggering. One station may have multiple loggers.
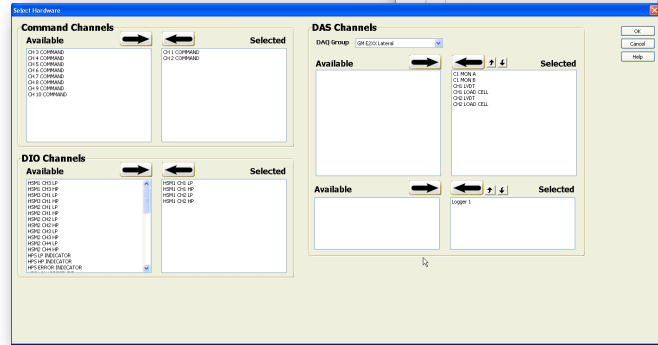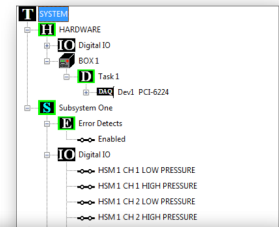
Card Diagrams

Once they have been assigned to a subsystem, SC6000 cards are set up using an easy to navigate card diagram. Connections between the transducer amplifiers, servo amplifiers and valve drivers are used to define the configuration of the control system. Software selectable monitor points on the cards are used to check the loops to ensure they are operating correctly.

Calibration

The SWCS has a highly intuitive, easy to use calibration screen. A step-by-step guide leads the user through the process, from setting the date, choosing the units, adding the points, and applying the changes. Different calibration curves can be applied: linear interpolation, step-wise, or a polynomial curve fit.

This screen, like many others, is customizable. Users can add extra sliders, buttons, lights or meters onto the window to make the calibration process easier for them. For example, if the transducer needs a shunt calibration, a button can be added onto the window to provide them with simple one-click on-screen control of the shunt. If the actuator needs to be moved during the calibration, the actuator command slider can be added to the same controls. The user can even add buttons to turn the hydraulics on and off, making the calibration screen a self-contained turn key process, with everything you need in one window. We will be discussing screen customization in more detail later, because it is a very powerful feature of the SWCS.
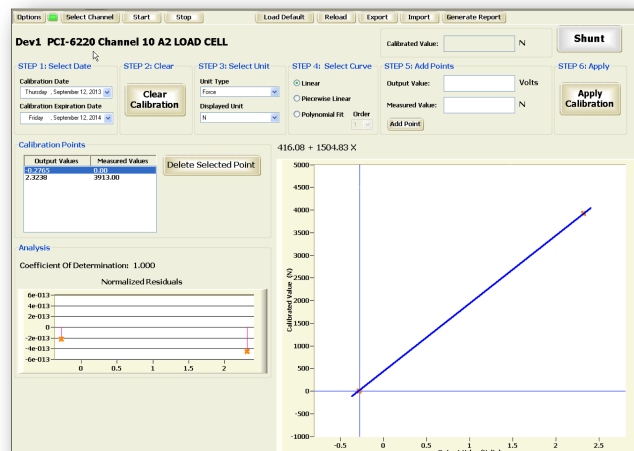
Block Programs
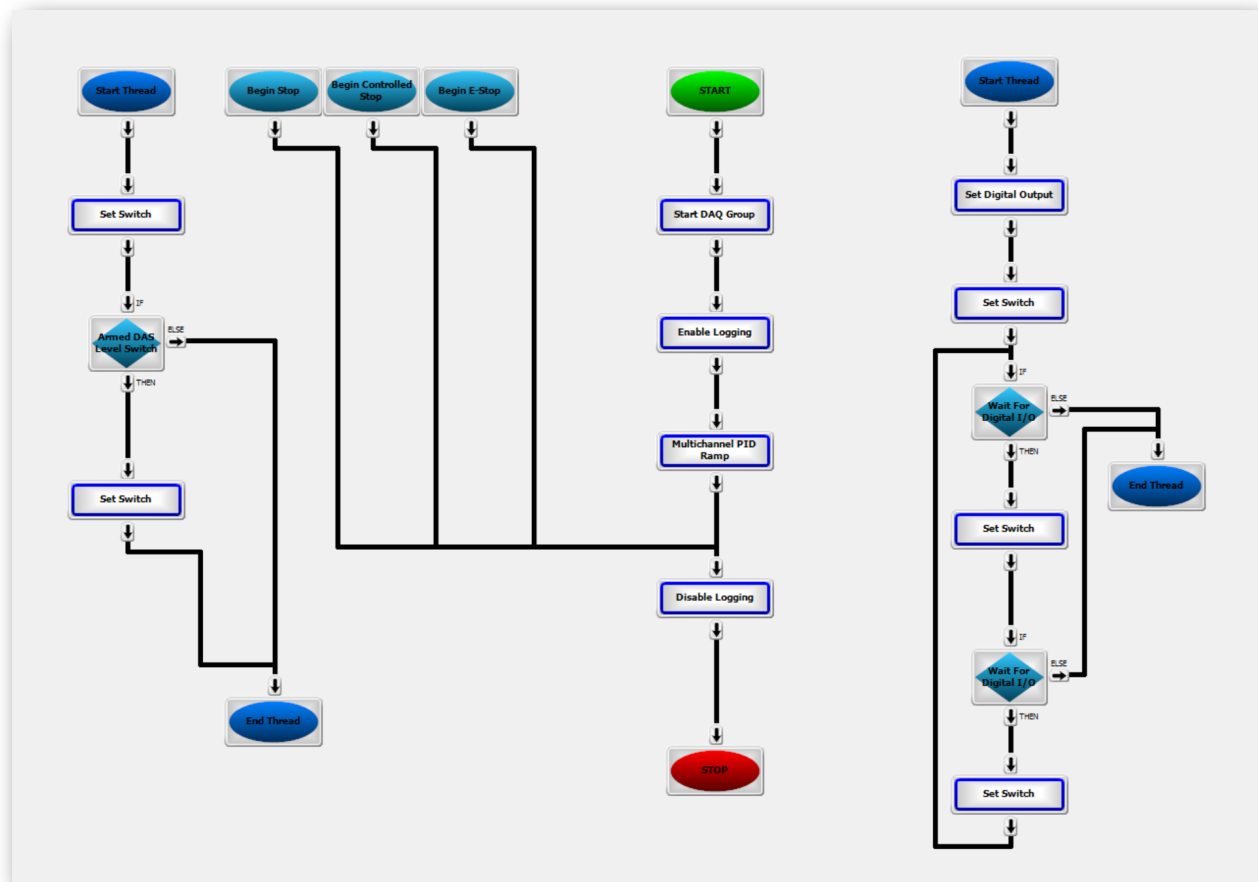
Another very powerful feature of the SWCS is its Block Programming capability. A feature rich menu of blocks are available to string together to perform highly complex process control tasks. Blocks are added from a pallet into the program, and wired together with a simple click on the output tab of one, and on the input tab of another. Blocks can easily be rewired or deleted too. And with a right click, they can be copied and pasted within the program, or across to another one. The user can move the blocks on a grid, and name them to create a visual view of the block program that is very easy to understand at a glance.

The block programmer provides automatic branching when the system encounters a Warning, Soft Stop, Hard Stop, Station Stop or E-Stop event, so that users can define actions that are responses to those conditions. The programmer even supports multiple threads, with inter-thread communication via virtual switches or flags, so that multiple actions can be performed in parallel. For example, you may want to ramp up in displacement control, while monitoring load. When a certain load is reached, you may want to ramp to a different level. In this case, you can define the ramps in the main program, and create a second thread that monitors loads. When the load is reached, the second thread notifies the main thread via a virtual switch, and the main thread moves on to the next step.

Blocks are available in the following categories:

Algorithm

- Mode Switch. For example, from displacement control to load control. This is only used if the hardware supports it (see the Hardware Section above).
- Auto balance. This block is typically used in a program that turns on low pressure to the actuator. Before applying pressure, you need to be sure the servo loop output is zero, to close the servo valve and prevent actuator motion.

Data Acquisition

- Start/Stop Data Acquisition and Logging
- Taring. To remove transducer and command offsets. For example, if you load a specimen in displacement control, and the actuator is not at mid-stroke, you may want to call the new position "zero" so that your test data is easier to read.
- Level Switches. These blocks are used to monitor response channels, and switch when they reach a certain levels. The levels can be absolute, or there is a block that can watch for a level drop, or level increase. This is useful for monotonic tests, where you want to detect specimen failure, or perhaps you need to wait for a pressure to rise, or a temperature to stabilize before continuing. These blocks provide conditional branching with timeouts.
- Load Calibrations. Calibration data can be loaded from a file

Digital IO
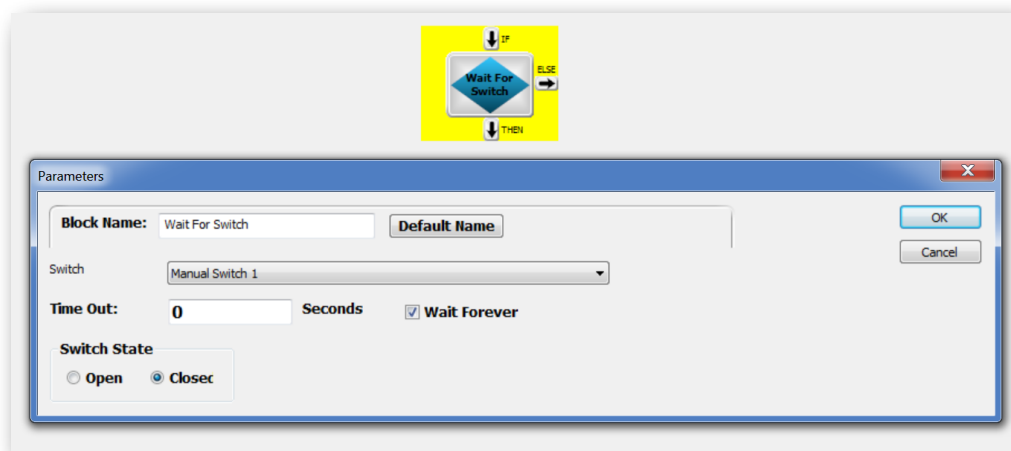
- Set Digital Output. Use this to change the state of a digital output signal (from high to low, or from low to high). For example, setting a digital output can turn on low or high pressure on an HSM
- Check IO State. You can check the state of a digital input, or a digital output, and branch based on whether it is high or low
- Wait for Digital IO. You can use this block to hold the flow of your program until a digital input changes state
- Set or Clear Switch. These are virtual switches that can be used to store state in your program, or to communicate across threads
- Wait for Switch. You can use this block to hold the flow of your program until a virtual switch changes state. For example, another thread in your program might change the virtual switch, when certain conditions are satisfied

Function Generator Blocks

- Actuators can be commanded with Ramps. Ramps can utilize compensation, and can be set based on a rate, or a time. The destination level can be an absolute value, or a value relative to the actuator's current position. Note that the defined target and compensation parameter can be different from the feedback used in the PID control loop. For example, for highly non-linear specimens, it is common for the PID loop to be in displacement control, with the ramp target and compensation in load control. If a ramp block is placed in a "For" loop, transformation values define an increasing, or decreasing target each execution, and so it is possible to put a simple pair of blocks in a loop that will progressively increase cyclic loads to specimen failure. Ramps can be paused or stopped via a virtual switch. This is useful when you are using another thread to monitor a signal during the ramp, and you would like to interrupt the ramp based on certain conditions.

- For Cyclic signals, or time histories, we use the Multi Axis Function Generator, which has the power to build complex sequences of periodic and non periodic signals, along with full Real Time Active Control (RTAC) compensation. This will be discussed in detail below. Block programs are also integrated into the Multi Axis Function Generator

Global Error Detect

- Disable/Enable/Set/Clear/Reset. The Global Error Detect is the system-wide interlock system. If an E-Stop is pressed, a limit is crossed, or a designated IO line changes state, the Global Error Detect trips, and the system shuts down. We will be discussing Limits and Interlocks later in this document. These blocks allow you to have full control of the Global Error Detect system. Note, for safety, the Global Error Detect cannot be cleared while the event that tripped remains active, and the Global Error Detect will only remain inactive for 30 seconds, before automatically re-enabling. This is to provide time for the operator, or block program, to manipulate the system out of an E-Stop condition.

Hardware Parameter Blocks

These blocks allow you to change the configuration of the SC6000 cards from within your Block Program. These blocks provide a lot of power to the user, allowing complete changes of system configuration programmatically.

Logical Blocks

- Subroutine. This allows you to call an entire block program as a subroutine, allowing you to reuse your block programs in multiple locations

- For Loop. This allows looping, and includes a cycle count. You can use this to loop back over a specific test multiple times. For Loops can be nested too.

- Begin/End Thread. You can create multiple parallel threads that execute in parallel to the main thread. These threads can communicate with each other via virtual flags called switches.

- Begin Warning/Soft Stop/Hard Stop/E-Stop. When certain system events occur, system execution jumps to the corresponding block and path definition. This allows you to define programs that execute under the various conditions. For example, if you are collecting data, you want to be sure you stop collecting when there is a Soft Stop, Hard Stop or Emergency Stop event. You can use these tools to have block programs run automatically at limit crossings.

Utility Blocks
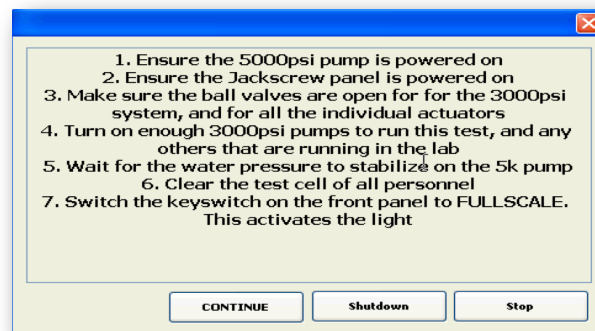
- Ask User/Send Instructions. When your program runs, you may want to pop a window to the user to inform them of the status, or of things they need to do. Alternatively, you may want to ask them a question, and branch based on their answer. These blocks provide you with the ability to do either.

- Delay. Sometimes you may want to have the program wait a few seconds. While pressure builds from low to high for example. In this block, you define the number of milliseconds for the delay, and the program will hold in the block for the predetermined length of time.

Once you see how these blocks can be wired together in any order, executed on a button press, and how multiple paths can be connected together, you will quickly realize that no other controller in the industry has the power and flexibility of

the Shore Western Block Programmer. The wide range of tasks that can be performed is difficult to grasp without seeing it first hand. This is another example of the industry-leading technology that we incorporate in our controller.
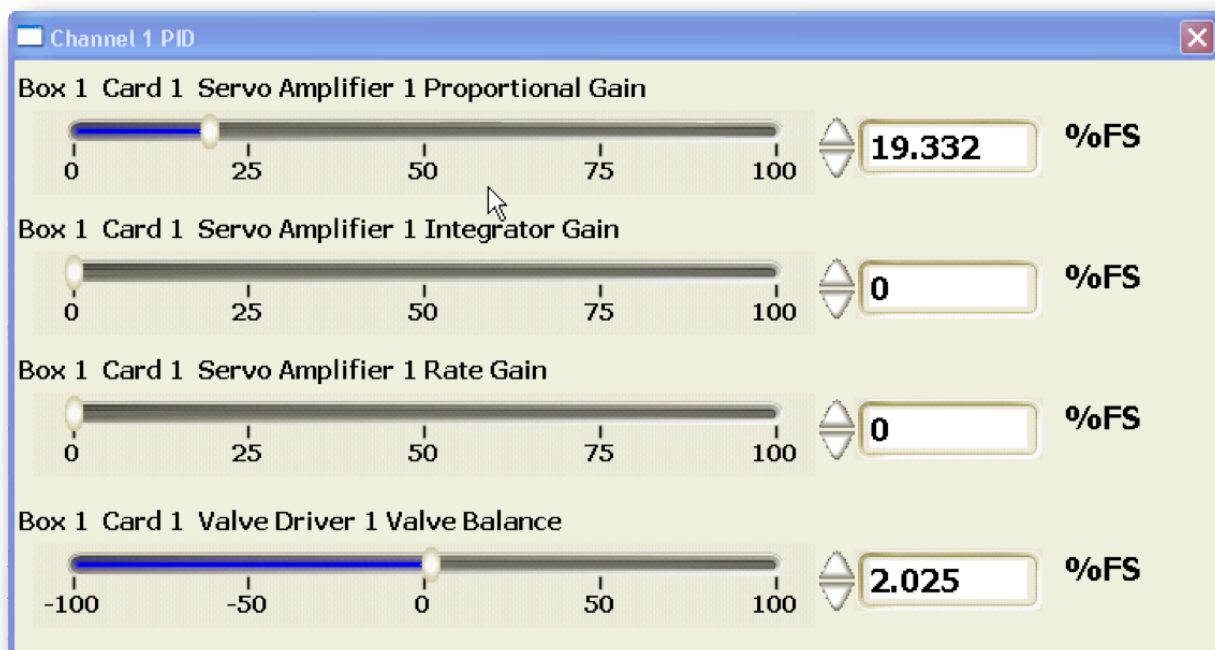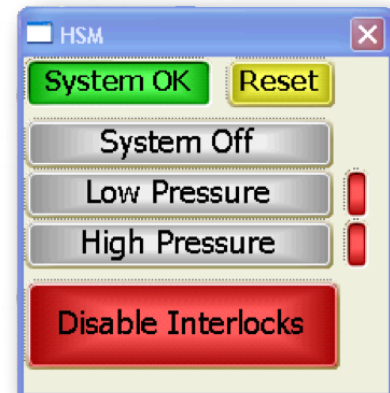
User Interface Customization

One of the reasons Block Programming is so powerful is that it can be linked in to a custom user interface. The user environment is set up in hierarchical layers: Controls, Panels, and Applications.

Custom Controls

Controls are individual groupings of buttons, lights, meters and sliders that can be invoked as floating windows, or embedded into Panels. Controls can be nested within each other too, so that a given layout can be used in many places throughout the user environment. For example, you may have a control for turning-on the hydraulics. One button runs a Block Program that checks the status of the system, asks the user a question or two, auto balances the servo loops, and turns the hydraulics to low pressure. Another button may turn from low pressure to high pressure. The panel may also include buttons to reset interlocks, indicate E-Stop status, and so on. The control can be created, and embedded anywhere in the software. It can be placed behind a button on another panel, so that it pops-up when the button is pressed, or it can be embedded directly into a Calibration panel, or a Multi Axis Function Generator for example.

All the elements in a custom control can be copied and pasted within the control, and also across controls. Controls can be saved to disk or imported from another source.

Custom Panels
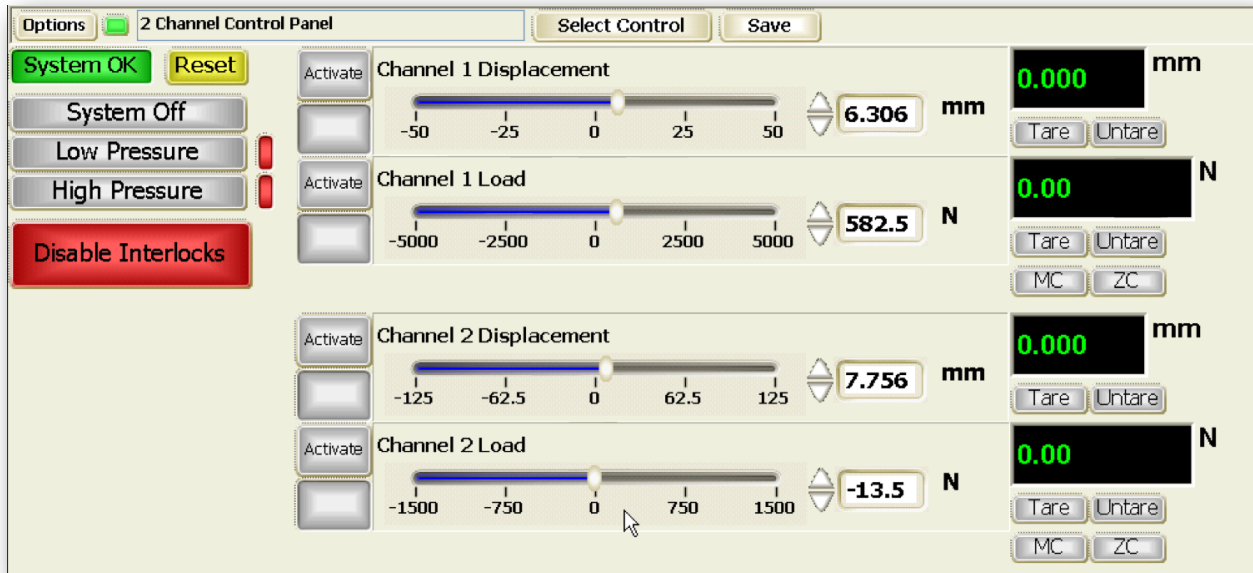
The user interface is divided into four quadrants. The quadrants can be split and resized, to create custom views. Any one quadrant can fill the window. The quadrants themselves contain panels. These may be predefined, such as the Multi-Axis Function Generator, or the Calibration screen, or they can be custom built. Custom buttons, lights, meters, sliders, and pop-up controls can be embedded into the Panels, as can Controls (above). This means that you can create a panel to perform PID tuning, for example, that takes an instance of the Multi-Axis Function Generator, adds hydraulic controls, and pop-ups for the PID tuning sliders. The panel includes a scope, and the function generator parameters, the settings of which can all be saved in the Application (see below). All the end-user needs to do, is select the tuning window from the Applications list, and everything is at their fingertips.

Panels can be saved to disk or imported from another source.

Note that Card Diagrams, the Block Program Editor, and Event Log are not panels that can be customized.

Custom Applications

Applications are a predefined group of panels. When you save a set of panels as an application, you also save all of the settings behind the panels, and their custom layout. Applications are selected from a simple pull-down at the top of the screen. Once a system has been fully customized, all the tasks required to set up and run a test, from calibration, tuning to execution, can be selected via the application pull-down menu. One single screen with all the controls required for calibration, one single screen with all the controls required for tuning, and so on. All the user needs to do, is follow step-by-step guides, or enter the parameters that are unique to his or her test.

The customizable user environment is another unique benefit of the SWCS, unmatched by any other controller in the industry.

Multi-Axis Function Generator
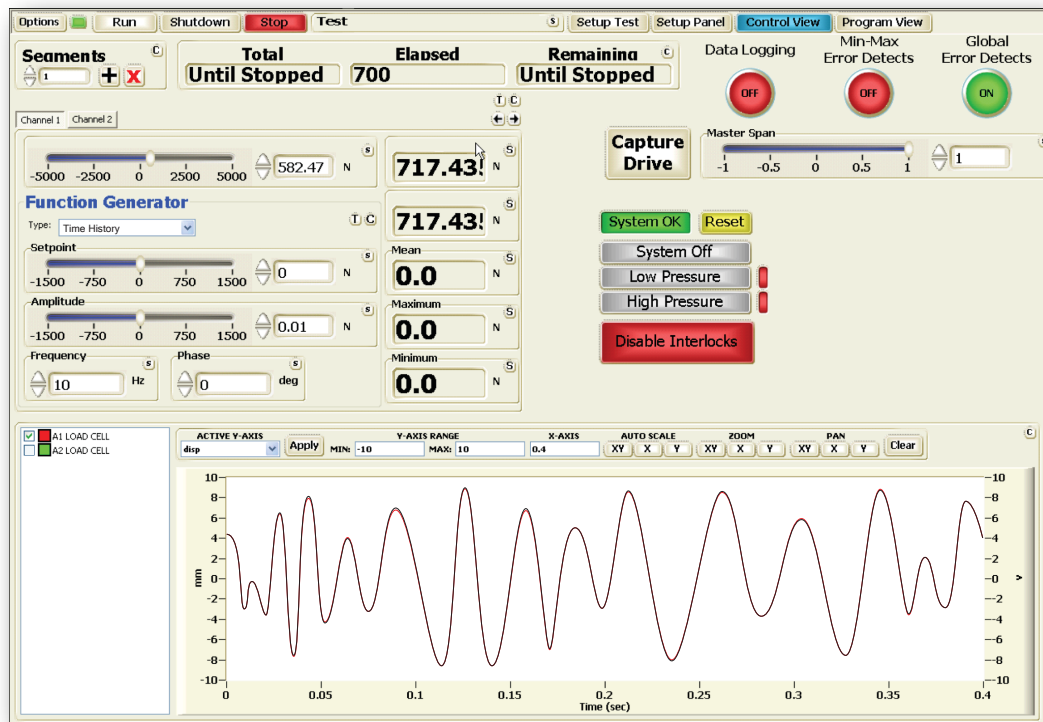
The centerpiece of the SWCS is the Multi Axis Function Generator. This is where users will spend most of their time. The function generator is a powerful test programming environment, complete with sophisticated Real Time Active Control (RTAC), that optimizes the commands, so that the system accurately responds with the signal you intend. The Multi-Axis Function Generator can reproduce the following signals:

- Sine
- Square
- Triangular
- Sawtooth
- Time Histories and Arbitrary Waveforms

In addition to the normal periodic signals, the Function Generator can reproduce any non-periodic time history, such as a recorded earthquake record, a road profile, or an arbitrarily defined waveform.

Multiple segments can be defined, where each segment is a signal with a given amplitude, frequency and number of repeats. A segment could also be a time history or arbitrary waveform with repeats. The user can define ramps between segments with or without compensation. The full sequence of segments can then be repeated as many times as necessary, to run a complete fatigue test.

Cycle counters are stored to disk in a file, and automatically backed-up to a second file, so that you never lose the status of your test.

Two types of compensation are available: Peak Mean and RTAC. Peak Mean is a simple algorithm for periodic signals that ensures that the peaks and valleys of the signal are faithfully reproduced, without regard to phase. It is the simplest to set up, and may be the best algorithm for single-channel fatigue tests. RTAC is a point-by-point reproduction that

ensures the entire waveform is reproduced, whether it is a periodic sine wave, or a time history. Phase and amplitude across channels are preserved, making a multi-channel real-time simulation possible.

What makes RTAC particularly powerful, is the fact that it can run in a "mixed mode." This means that you can run the digital optimization with a different transducer than the analog control loop. As an example, you can run the PID loop in displacement control, while commanding, optimizing, and reproducing a load signal. Or, for a vibration table, you can command, optimize, and reproduce acceleration, while running the actuator in displacement control.

Set Up of RTAC is relatively straight-forward. A step-by-step process is followed, that leads to a full simulation in minutes. Drive files (the files that are used to reproduce the target response) can be stored and recalled. When RTAC has converged, you can continue to adapt throughout the test, or you can freeze it, so that it does not adapt as the specimen's stiffness changes.

To perform this same task, our competitors use an arcane frequency domain based modeling technique that was invented in the 1970s. since then, little has changed in the basic technique, which is difficult and time intensive to set up, and takes highly experienced personnel. It also cannot compensate once the test has started. It is effectively open loop. RTAC is another industry first. It significantly reduces the required setup time, removing many of the difficulties of a time history simulation, while improving accuracy throughout the entire course of the test.

The Multi Axis Function Generator panel is preconfigured with the controls you need, but it can be completely customized with your own buttons, lights, meters, sliders, and pop-up controls. It also has Block Programming capability, so that you can define the events before and after the test, along with what happens if there is a warning, emergency stop, soft stop, or hard stop.

Displays

Three different data displays are available in the software, a time history oscilloscope, a frequency domain analyzer, and an X-Y plotter. The displays feature the following capabilities:

- Multiple Axes for plotting different variables (scope only)
- Auto Scaling
- Manual Scaling
- Zoom Scaling
- Filtering and decimation of displayed signals (scope and X-Y)
- Setting colors and line widths
- Saving, exporting, and importing
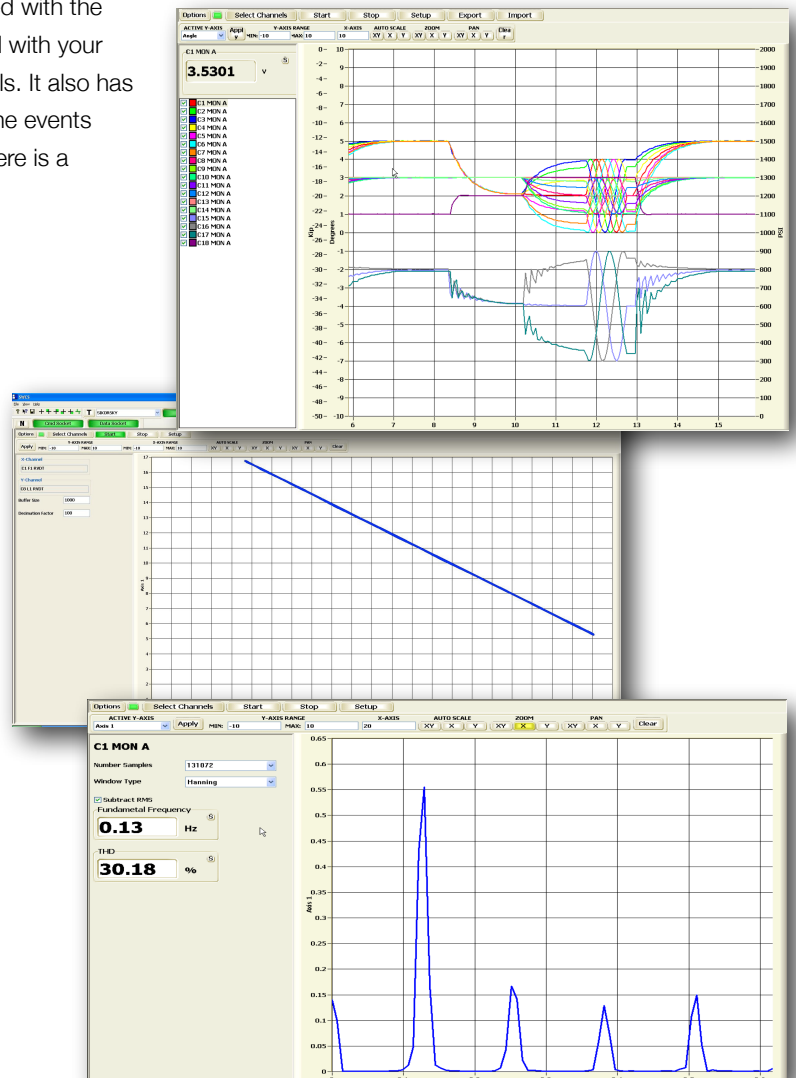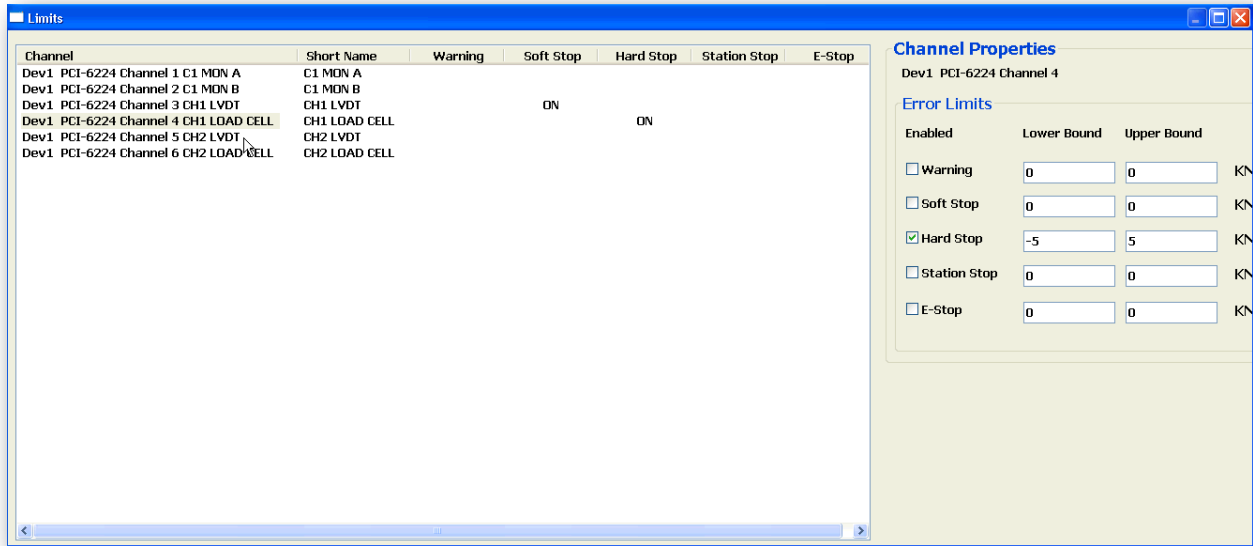
Limits and Interlocks

Five sets of system-wide limits are available:

- Warning. An entry is made in the event log, the Global Error Detect changes state to yellow and the Block Programmer branches to the Begin Warning block (if defined).

- Soft Stop. The program is gracefully ramped to zero, the Block Programmer branches to the Begin Soft Stop block (if defined), an entry is made into the event log, and the Global Error Detect changes state to amber

- Hard Stop. The program stops wherever it is, the Block Programmer branches to the Begin Hard Stop block (if defined), an entry is made into the event log, and the Global Error Detect changes state to red

- Station Stop. All the relays tied to a Station Stop event are opened. These are typically for the HSMs, but a pump could also be configured to shut off. The program immediately stops, the Block Programmer branches to the Begin Station Stop block (if defined), an entry is made into the event log, and the Global Error Detect changes state to to flashing red and yellow.

- Emergency Stop. A hardware interlock is performed, shutting off the output relays[2], the program immediately stops, the Block Programmer branches to the Begin E-Stop block (if defined), an entry is made into the event log, and the Global Error Detect changes state to flashing red and yellow.

The user can enter different limits for each state, and a persistence. This is the number of data points that must fall outside the limit before action is taken.

These same actions can also define the response to an input relay's change of state.

Data Acquisition

Two types of data acquisition can be defined for a test: continuous logging, or cycle-based logging. You can trigger continuous data collection manually, by using the start button on the Multi-Axis Function Generator, or via a block in the Block Programmer.

---

[2] In special cases, you may need an output relay to close rather than open in the event of an E-Stop. This can be done with a combination of hardware jumpers, and software definition.

For long-term fatigue tests, you do not want to fill the disk by collecting data continuously. However, you also do not want to miss an important event when it occurs. You can define cyclic logging where, for example, you can collect 10 cycles of data every 1000 cycles of testing. At the same time, we collect data into a circular buffer, typically 200 cycles long (user defined). If something happens in the test, you can step back through the buffer, and examine each cycle in turn, to see where the change occurred.

Pseudo Dynamic and Hybrid Simulation

The open architecture of the Shore Western Control System makes it straightforward to interface with analytic models, so that physical structures can be tested from within a virtual environment.

Using the NEES grid, several components can be tested together in various labs across the world, to simulate a large integrated structure. Two types of protocols have been developed for this "Pseudo Dynamic" testing: SIMCOR (from UIUC) and OpenFresco (from UC Berkley). Shore Western is able to work with both protocols. For full dynamic hybrid simulation, we interface with an xPC running Simulink.

For Pseudo Dynamic integration, the SWCS controller includes a control panel that can accept commands from either a UI SIMCOR based network connection, or from an OpenFRESCO network (see Figure 1 below). Note that, in addition to this integration, the SWCS is able to interface directly with any third-party outer loop controller via analog IO. Since the
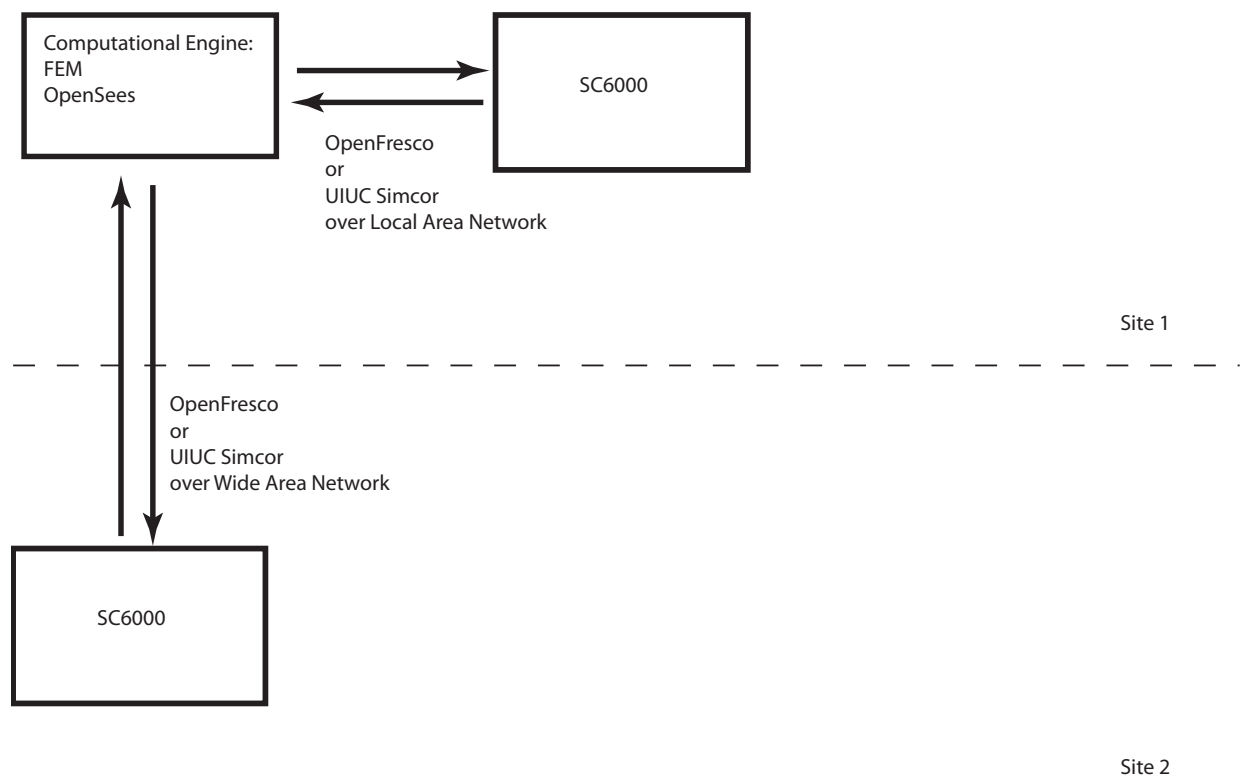
SWCS is a digitally optimized analog controller, there is no latency, clocking, or resolution issues with this approach. This is the only controller on the market that provides such seamless integration (see below).

Note that, in addition to this integration, the SWCS is able to interface directly with any third-party outer loop controller via analog IO. Since the controller is a digitally optimized analog control, there is no latency, clocking, or resolution issues with this approach. This is the only controller on the market that provides such seamless integration (see below).

Computational Engine: FEM OpenSees	**Controls and** **Data Acquisition Engine** dSpace or xPC Target with speedgoat front end Command Optimization (State-Space Modeling) Matlab Simulink	**SC6000**

OpenFresco
or
UIUC Simcor
over Local Area
Network

Analog I/O

Site 1

OpenFresco
or
UIUC Simcor
over Wide Area
Network

Controls and **Data Acquisition Engine** dSpace or xPC Target with speedgoat front end Command Optimization (State-Space Modeling) Matlab Simulink	**SC6000**

Analog I/O

Site 2