SUPERCONDUCTING COIL WINDING MACHINE CONTROL SYSTEM*

J. M. Nogiec[†], S. Kotelnikov, A. Makulski, K. Trombly-Freytag, D. Walbridge
Fermi National Accelerator Laboratory, Batavia, IL 60510, USA

Abstract

The Spirex coil winding machine is used at Fermilab to build coils for superconducting magnets. Recently this machine was equipped with a new control system, which allows operation from both a computer and a portable remote control unit. This control system is distributed between three layers, implemented on a PC, real-time target, and FPGA, providing respectively HMI, operational logic and direct controls. The system controls motion of all mechanical components and regulates the cable tension. Safety is ensured by a failsafe, redundant system.

INTRODUCTION

The need to wind superconducting coils for LHC Accelerator Research Program (LARP) led to a project to equip the Spirex coil winding machine with a control system based on up-to-date technology and to provide the machine with a safety system that meets current safety policies [1].

The winder machine is about 11 m long with a bridge that moves along a track and supports a rotating boom holding a spool of cable and providing cable tension, and the mandrel supporting the coil with its winding fixture (Fig. 1).

The control system was designed to provide necessary functionality for winding coils, including automatically keeping the proper cable tension and cable spool elevation, synchronizing motors for mandrel tilt and bridge motion, and providing interlocks to prevent material or equipment damage.

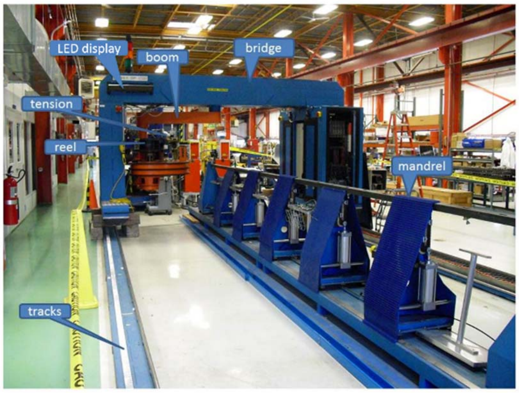


Figure 1: Coil winding machine.

COMPUTERS AND INSTRUMENTATION

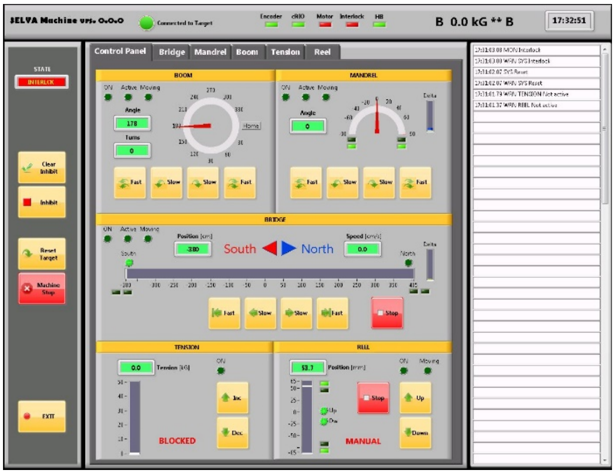
The computer and data acquisition (DAQ) hardware consists of a PC-based station connected via a private network to a CompactRIO (cRIO) [2] crate with I/O modules and a processor running a real-time operating system. An extension crate is connected via Ether CAT to the cRIO crate. All of the control signals to and from the machine hardware are routed through an interface box, which also contains a heartbeat watchdog circuit.

Two wireless CAN transmitters are connected to the cRIO crate, allowing for control of the machine from the remote console. Vital machine parameters are displayed on two large LED displays located on both sides of the bridge.

SOFTWARE ARCHITECTURE

The control system was designed with the following layered architecture:

- HMI Layer - modules residing on the HMI computer implementing the user interface and data storage.
- Operational Logic Layer - modules that provide operational logic residing on the real-time (RT) target computer.
- Direct Control Layer - FPGA code that interfaces I/O signals and implements fast and deterministic behavior, such as motion control, motion synchronization, tension and reel regulation and interlocks.
- Hardware Layer – actual hardware elements including motors, sensors, actuators, and switches.



17:31:03.08 MON Interlock
17:31:03.08 WRN SYS Interlock
17:31:02.07 SYS Reset
17:31:02.07 WRN SYS Reset
17:31:01.79 WRN TENSION Not active
17:31:01.37 WRN REEL Not active

Figure 2: Touch screen user interface.

HMI Computer

The HMI computer's software starts the tension control, and provides a graphical user interface (GUI) for controlling and monitoring the machine (Fig. 2). Although the primary method of controlling the machine is via the remote console, the GUI on the HMI computer can perform a larger set of functions. The current state of the machine, including its position, tension, mandrel angle etc. can be viewed on the HMI computer display, even when the machine is being controlled by the remote console. All software exceptions as well as important events are both

* Work supported by the U.S. Department of Energy under contract no. DE-AC02-07CH11359

† nogiec@fnal.gov

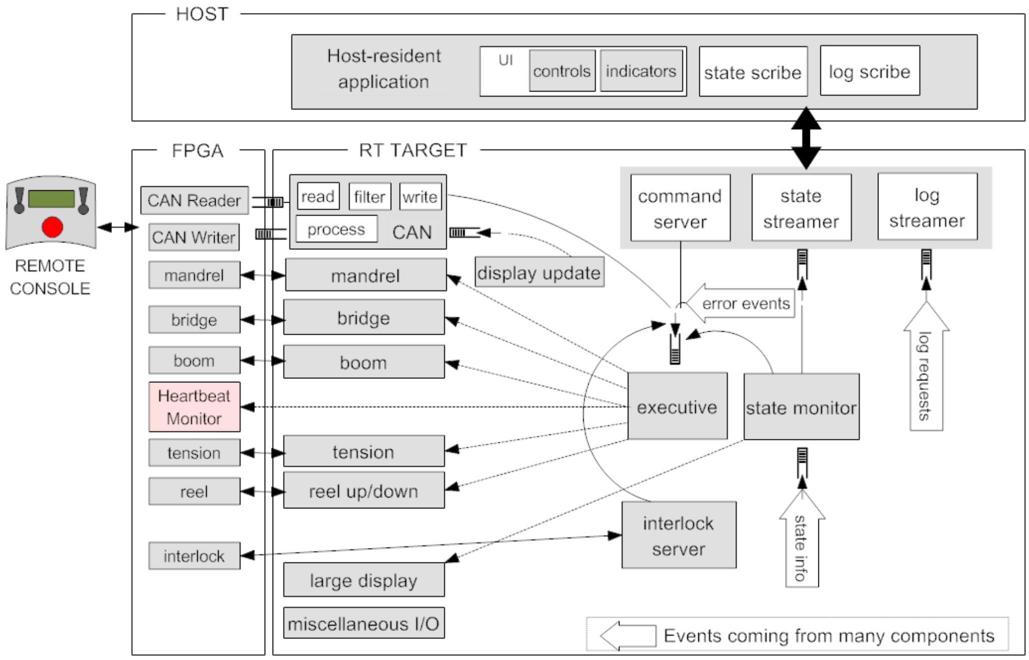


Figure 3: Software architecture.

logged to files and appear on the GUI, including the history of all motions performed by the machine.

The HMI computer's software communicates with the RT target over TCP/IP, with separate streams used for logging events, saving state and sending control requests.

Real-Time Target Computer

The software on the real-time target computer consists of several functionally distinct modules (Fig. 3):

- Executive, a module implementing operational logic as a state machine that receives events (commands) on its input queue and processes them depending on its current state. The commands may be sent by the CAN bus communication module (commands from the remote console) and/or the TCP/IP Communicator server (commands from the HMI computer). It also periodically places information about the current state in the input queue of the TCP/IP Communicator server.
- State Monitor, a server that receives data from all the components in the system. It assembles the data into a status data structure and places the data in the input queue of the TCP/IP Communicator server. That server will then send the structure to the registered client application running on the HMI computer. The state monitor also receives the heartbeat signals from all vital components and passes these signals to the executive server.
- TCP/IP Communicator, a server implementing communication with the HMI computer, capable of concurrently sending state, log messages, status information and receiving control commands.
- CAN Bus Communicator, a module that implements communication with the remote control panel and the auxiliary control unit (a small remote allowing for disabling all movement) over the wireless CAN bus. The software for the CAN communication is divided between the FPGA and RT computer, which communicate using queues. The RT portion receives a stream of CAN transmissions, which is filtered and checked for changes in individual signals. Detected signal transitions are then translated into events, which are placed in the executive's input queue. Interruptions in transmissions are also communicated to the executive module. Using a separate queue, the RT portion receives short messages regarding tension or state that are displayed on the control panel.
- Motor Control, a proxy for the FPGA implementation of motor control, which communicates with the FPGA motor control implementation.
- Interlock Service, a module monitoring the system and reacting to situations when the interlock logic detects an abnormal/exceptional situation. It then automatically disables the machine and communicates this event to the executive.
- Large Display, a module responsible for handling the two large LED showing cable tension and cable reel position.
- Utilities packages that are available for use by other modules, including logging, error reporting and property services, which provide standard methods for storing parameters, calibration values and constants.

FPGA

The FPGA is directly connected to the I/O modules for high-performance access and guarantees almost no control latency for system responses. Because of the FPGA speed and determinism, it is used not only to interface I/O signals, but to implement PID control loops for the servo motors, the RT software heartbeat protection, CAN bus communication and the interlock logic.

The FPGA code is distributed between the main (master) chassis and the EtherCAT [3] slave extension chassis.

HEARTBEAT

The heartbeat signal is used to increase the dependability of the machine. All major components responsible for the correct behavior of the machine, including the motors, the interlock logic, state monitoring and the operational logic must function properly in order for the heartbeat signal to be generated. The state monitor collects the heartbeat signals from the motors and the interlock server, and when all have been received it passes the heartbeat signal to the executive, which sends it to the heartbeat monitor implemented in the FPGA. If the heartbeat monitor times out waiting for the signal, it will perform the heartbeat lost procedure, which stops and disables the machine.

MOTOR AND TENSION CONTROL

The machine has several servo motors controlling the motion of its elements, including the bridge, mandrel, boom, tension system and reel. These controls involve not only regulating the motor speed, but also monitoring the distance based on encoder readouts, monitoring the end switches, and controlling the brakes. The servomotors are controlled by PID regulators implemented in the FPGA. For motion requiring multiple motors (both the bridge and the mandrel) synchronization of motion on the same axis is also done in the FPGA.

The motion of the bridge is accompanied by a horn alert and requires blowers to be activated (to remove small objects from the tracks).

All motion can be inhibited by the operator via the HMI GUI, remote control or the auxiliary control unit (the system enters the inhibit software state) or disabled/terminated by removing power to all the motors (using the hardware push buttons).

Keeping the cable tension constant, a non-trivial problem in situations where the length of the cable changes with varying speeds, has been addressed by a PID controller with feed forward augmentation and low-pass filters.

SAFETY SYSTEMS

For a machine with several large and powerful moving parts, both training and experience are necessary for machine operation. In addition, the system has several safety features designed and built in to protect both the operators and the equipment against such situations as accidental over tension, or fast movement of the cable due to operational errors. The safety system has been constructed to be both redundant and failsafe.

During the operation of the machine, each operator must always be aware of location of the safety devices, which include Emergency Stop buttons, protective safety bumpers, the mandrel safety rope, interlocks, lights and signage. To facilitate operator awareness, two light trees are mounted on the system, with various combinations of colors and lights representing different actions or conditions of the winding machine.

CONCLUSION

The Spirex magnet coil winder has been equipped with a new control system, which allows operation from both a computer and a remote control unit. The control system has a layered architecture with software distributed between a PC, a real-time target, and FPGA, providing respectively the HMI, operational logic and controls. Regulation of servomotors propelling the bridge, mandrel, and boom and their synchronization is implemented in FPGA, which also hosts a PID controller with feed forward augmentation to control the tension.

Extra attention was given to safety aspects resulting in the development of a failsafe, redundant safety system with interlocks, including protection for the operator, the machine, and the superconducting cable.

This system has been in operation since mid-2016, winding coils for superconducting magnets as part of the LARP project. Although the initial need is for winding quadrupole coils for the LHC IR upgrade magnet MQXF, the machine is capable of winding coils of various lengths and types.

ACKNOWLEDGMENT

The authors would like to express their appreciation to everyone who helped this project to be a success, especially F. Levis, F. Nobrega and their technicians from Fermilab's Technical Division for support in electrical work and testing of the machine.

REFERENCES

[1] J. Nogiec *et al.*, "SELVA Operating Manual", Fermilab, Batavia, IL, USA, Rep. TID-N-728, Jan 2015
[2] NI LabVIEW for CompactRIO Developer's Guide, `http://www.ni.com/compactriodevguide/`
[3] Programming EtherCAT I/O with FPGA Intelligence, `http://www.ni.com/white-paper/10618/en/`