CLOSED LOOP WIRE SCANNER ACTUATOR CONTROL FOR LANSCE ACCELERATOR BEAM PROFILE MEASUREMENTS*

Stanley Cohen, Sandeep Babel (BiRa, Albuquerque, New Mexico),

J. Douglas Gilpatrick, James D. Sedillo (LANL, Los Alamos, New Mexico),

David A. Bonal, Murali M. Ravindran (National Instruments, Austin)

Abstract

The design and test of a new beam-profile wire-scanner actuator for the LANSCE (Los Alamos Neutron Science Center) 800-MeV proton linear accelerator is described. Previous actuator implementations use open-loop stepper-motor control for position indexing. A fixed-frequency, fixed-duration pulse train is sent to the stepper motor driving the linear actuator. This has lead to either uncertainties in position due to mechanical resonances and electrical noise or slowing down actuator operation..

A real-time, closed loop control system is being developed and tested for more repeatable and accurate positioning of beam sense wires. The use of real-time controller allows one to generate a velocity profile for precise, resonance-free wire position indexing. High radiation levels in the beam tunnel, dictate the use of an electro-magnetic resolver, typically, used in servo applications, as the position feedback element. Since the resolver is an inherently analog device, sophisticated digital signal processing is required to generate and interpret the waveforms that the feedback mechanism needs for positioning. All of the electronic and computational duties are handled in one the National Instruments compact RIO real-time chassis with a Field-Programmable Gate Array (FPGA)

DESIGN CONTRAINTS

Timing and Physical Environment

Beam-profile wire scanners present a number of electrical and mechanical challenges for designers.. Achieving accurate, repeatable and rapid wire positioning is a key function that must be addressed to obtain reliable beam-intensity data in a timely manner. The high beam intensity, itself, limits the parameter space of what kinds of electronics and materials can be used in the wire-scanner design.

This report is focused on controlling wire-scanner position using the National Instruments cRIO system and getting the actuator movement in the FPGA of cRIO under close-loop conditions., For this first set of tests we are using a wire scanner assembly from the decommissioned LEDA project[1,2]. Operating the wire scanner with closed-loop control is one of the essential requirements for the beam diagnostics refurbishment of the 30-year-old LANSCE proton linear accelerator at Los Alamos [3, 4].

Incorporating closed-loop motion control for the positioning the wire will provide a more accurate picture of the proton beam profile than practice of using open-loop stepper-motor positioning.

Motion control in this design demands that the position for each intensity measurement be achieved during the 8ms between beam macropulses. That is the mechanism must move and settle, between beam pulses, before there is request to measure the beam intensity at that location.

ACHIEVING CLOSED-LOOP MOTION WITH A STEPPER MOTOR

Mechanically Quantized Motion

Using a stepper motor within a closed-loop motion control context introduces quantized motion into the system. The rotational motion of the motor cannot be moved to an arbitrary rotational angle. This can be mitigated somewhat, by microstepping the motor, but this slows down its rotational speed, hence the maximum linear velocity of the wire housing. Motion is, ultimatedly controlled by a computer. This has implications for the motor-control algorithm, since all I/O must be digitized. The wire cannot be positioned with arbitrary accuracy, using an analog to digital converter, (ADC) in conjunction with a linear encoder, may further limit the position accuracy. This limitation and the former quantization effects means that the wire can be positioned to the commanded position within, $\pm\varepsilon$, a small error. This will determine the convergence criterion for reaching a position setpont.

Wire Scanner Physical Environment and Constraints

Why use a stepper motor, at all, since servo motors are available? A stepper has the advantage of having detent torque or “holding torque”, when it’s stopped. This keeps the wire stationary without an explicit program for the stopped motion. It is assumed that the detent torque is great enough to overcome competing mechanical forces on the wire assembly. The stepper motor comes to a complete stop, when command signals are inhibited. This is advantageous for keeping the beam-sense wire steady for sensing the particle beam charge.

Linear Position Feedback Elements

The choice of feedback elements for the LANSCE-accelerator is limited, since radiation levels are high

---

* Supported by LANL Contract SBA 030608

enough to destroy some internal elements for an optical encoder. We want to close the feedback-control loop with a sensor that samples the linear motion of the wire carrier directly. We are limiting our choices to a resolver [5], linear variable differential transformer (LVDT) [6] or magnetostrictive linear displacement device (MLDT) [7]. We've chosen to postpone the decision on which transducer to use and focus on developing closed loop motion control software. For this proof-of-principle project, a linear potentiometer was chosen to close the feedback loop. We are developing all of the programming using relatively low-level modules available in Labview. National Instruments propriety "SoftMotion" package does not have the bandwidth required for the fast positioning performance we are demanding.

Compact RIO and FPGA loop control

The heart of the motion control system we are developing runs on a National Instruments(NI) hardware chassis (See Figure 1) based upon NI's Compact RIO (reconfigurable I/O) architecture. The system consists of five major parts. See Figure 2.

1. Real-time controller
2. Chassis-mounted signal conditioning modules
3. A Virtex II FPGA (field programmable gate array) that is the intermediate processor between the RT and the I/O modules.
4. Ethernet interface to the rest of the world.
5. Shared variable engine

The power of this configuration is contained in the FPGA which can run multiple, simultaneous processes. This is a true parallel process environment at a minimum of 40 MHz

The beauty of this arrangement is that the FPGA is programmed in Labview, not VHDL. Three pieces of software must be written, but the all use the common graphical programming paradigm of Labview, the virtual instrument (VI).

The FPGA allowed us to create our own stepper-motor-pulse train generator, signal conditioners, and digital I/O that are running as independent processes on the FPGA.

Figure 1 Compact RIO Hardware

The first control algorithm for our LEDA test bed consists of an FPGA module with stepper-motor speed and direction pulse control analog input from the linear-encoder potentiometer and a number of digital inputs whose state can be passed to the real-time controller.

The control loop is closed in the real time controller a simple PD algorithm is used which has been tweaked to compensate for the quantized motion and position data. See details below.

A shared-variable engine passes start stop and set point information to a VI running on a Windows XP, personal computer on a TCP/IP network. This mimics the topology of the future implementation of the system. Our goal is to pass the exact process variables used by the EPICS control-system drivers. See Figure 3.

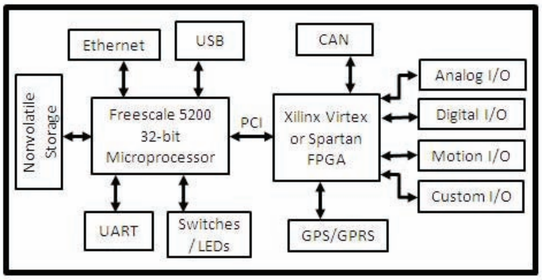

Figure 2. National Instruments cRIO architecture

Control Algorithm

The requirement that the wire assembly moves every 8 ms during operation implies that the response to a step input overrides long-term stability of the control loop. A successful control scheme is one that will move quickly and accurately to the commanded setpoint (linear position).

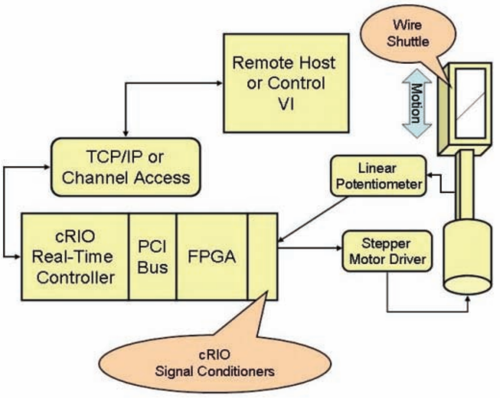

Figure 3. Wire Scanner Control Architecture

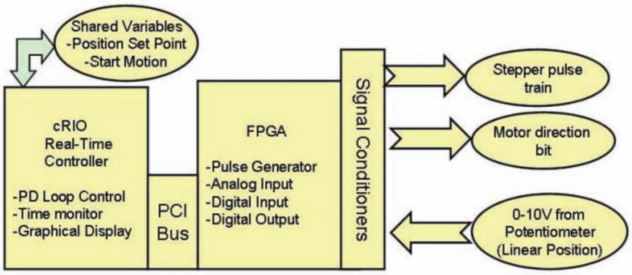

Figure 4. Distribution of processing tasks

Using the conventional model of proportional, integral, differential (PID) control, the integral part becomes irrelevant, compared to the mechanical time constants of the wire positioning system. The loop bandwidth dictates a fast feedback loop. Preferably, “closing the loop” in the FPGA. That is, placing the entire closed-loop control algorithm in hardware. No data required for loop convergence will need to be passed to the PCI bus to the real-time portion of cRIO during the 8 ms of wire motion.

These first tests, whose results are described here, sacrificed speed for understanding the system behavior. Therefore, loop convergence calculations were done in the real-time portion of the cRIO system. See Figures 4 and Figure 5.

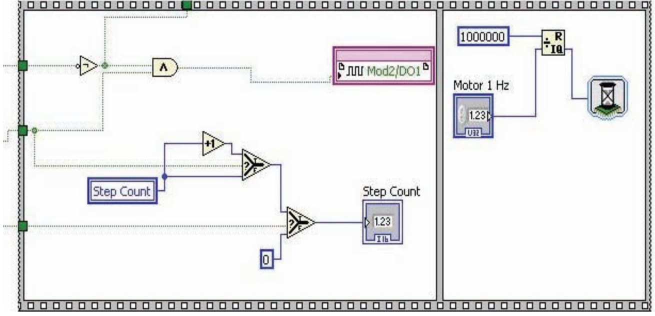

Figure 5. FPGA programming is done with Labview.

PRELIMINARY TESTS AND RESULTS

Loop Parameters and Stepper Motor Control

A minimum number of channels were used with the cRIO hardware for this series of tests. An NI 9215 cRIO analog input module read the linear potentiometer. A single 16 bit channel was set up to read at 50 µs intervals. Pulses and direction digital an NI 9474 handled information, 1 MHz digital output module.

The inclusion of the FPGA in the cRIO system allowed us to create and modify stepper-motor control signals at that were, essentially hardware. For these tests, we used an IMS MForce motion controller [8] that takes a square wave pulse train to control the motor’s rotational speed and a single bit to specify direction of rotation. The motor will move one step (a specific rotational angle) with each transition of the pulse train. MForce is a micro stepping controller that allows one to change the number of steps per one complete rotation of the motors shaft. This can be adjusted “on the fly” via an Serial Peripheral Interface (SPI) [9].. For these tests the motor rotates 0.45 degrees for each pulse. If required, one could step this up to 1.8 degrees to facilitate a rapid retraction of the wire shuttle.

The linear potentiometer supplied absolute position as feedback which made loop calculations very straightforward. Input parameters for control consisted solely of specifying a target position (setpoint) and a start command. The motor moves toward the new position and stops its position is within ε of the target. Optimization of the motion was achieved by varying sample time, maximum motor speed (which is analogous to proportional gain), and differential gain. A further enhancement is a threshold parameter that reduces the proportional gain linearly as the position error approaches the target value.

By changing the parameters one can get the loop to exhibit ringing and milder overshoot, as well as critically-damped behavior. See Figure 5

Preliminary results achieved performance of 87 ms to move 1 mm with an accuracy of 0.1 mm. We estimate that, at least, 50 ms of that time is communication overhead from the real-time processor and the host monitor computer.

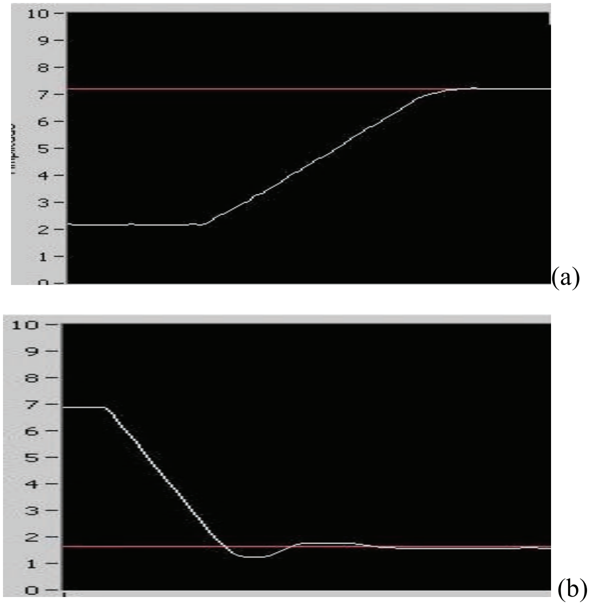

Figure 6. Actual recorded motion of wire shuttle (a)Critically and (b)Under-damped motion.-- Red - Set point White - Shuttle Motion

REFERENCES

[1] R. Valdiviez, , et. al.,The Final Mechanical Design, Fabrication, and Commissioning of a Wire Scanner and Scraper Assembly for Halo-Formation Measurements in a Proton Beam. PAC2001, Chicago, IL

[2]. D. Barr, et.al.,” Design and Experience with the WS/HS Assembly Movement Using LabVIEW VIs, National Instrument Motion Controllers, and Compumotor Electronic Drive Units and Motors., PAC2001 Chicago, IL

[3] K. W. Jones, et. al., The LANSCE Refurbishment (LANSCE-R) Project, PAC07, Albuquerque, NM

[4] J. D. Gilpatrick, et.al., Proposed Beam Diagnostics Instrumentation for the LANSCE Refurbishment Project, PAC07, Albuquerque, NM

[5] http://www.inductosyn.com/linear_transducers.htm

[6] http://www-ssrl.slac.stanford.edu/ble/lvdt.htm

[7]._http://www.directindustry.com/prod/novotechnik/absolute-magnetostrictive-linear-position-sensor-13745-210991.html

[8]_http://www.imshome.com/Product%20Datasheets/MFM_Micro.pdf

[9]http://www.embedded.com/story/OEG20020124S0116