Development of Remote Controls for the Motorized Polarization Controller in LIGO's Arm Length Stabilization System

Caroline Martin
Mentor: Daniel Sigg

Interim Report 2
LIGO SURF 2017

LIGO-T1700331-v1

August 4, 2017

Abstract

The arm length stabilization (ALS) system allows the arms of the interferometer to be locked separately, decoupling these two degrees of freedom from the Fabry-Perot and recycling cavities; this system faces issues with polarization drift along fiber optic cables, however, due to factors such as thermal stress, mechanical stress, and irregularities in the shape of the core. If this drift is not corrected, the mismatch in polarization can prevent the interferometer from observing. Currently, this drift is corrected by a motorized polarization controller (MPC) that must be adjusted manually on a regular basis. This project aims to develop comprehensive, user-friendly, and robust remote controls for the polarization controller to streamline the drift correction. The controller was connected through a serial port to TwinCAT Programmable Logic Controller software, on which the controls were written in IEC-1131 structured text. User input through a graphic interface is interpreted and written to this code, allowing the user to remotely control the polarization. In the future, this program could be used to automate the correction process.

1 Motivation

The gravitational wave detector at LIGO Hanford Observatory (LHO) is a specialized Michelson interferometer with 4 km long arms that convert spacetime perturbations into a measurable signal [1]. The difficulty in this measurement is due to the length scale of these perturbations; this requires both that the interferometer mirrors are isolated from disturbances and that the noise encountered is quantifiable and removable. Thus, arm length stabilization (ALS) is essential. The current ALS system, however, faces issues with the drift of the polarization of light along fiber optic cables. This drift is corrected by a motorized polarization controller (MPC) that is adjusted manually on a regular basis. As aLIGO progresses, it is important to maximize the time spent locked and observing; by streamlining and digitizing the process to correct this polarization drift, we will allow for greater efficiency. The goal of this project is to develop remote controls such that an operator familiar with the physical device will intuitively be able to perform the same functions remotely using the user interface. This code would allow for more than just remote controls, however; it would also allow for data acquisition (DAQ) storage of the numeric channels for analysis, easier monitoring of the MPC settings, hourly snapshots of the state of the system, and control and data system scripting with the ability to automate the control of the MPC [2].

2 Background

2.1 Arm Length Stabilization (ALS) System

In order for the interferometer to be functional, the Fabry-Perot cavities, power recycling cavity, and signal recycling cavity must be kept locked on resonance. For aLIGO, a stabilization system was devised to make locking reliable and repeatable using active feedback control. The arm length stabilization (ALS) system locks each arm cavity individually and separately from the central recycling cavities using lasers mounted behind each end test mass; they are doubled Nd:YAG lasers operating at 532 nm (green) deployed at each end station to distinguish them from the main laser. [3] By locking the arms separately, the ALS decouples these degrees of freedom.

Once the auxiliary laser is locked to the fiber transmission, the output of the laser is locked to the arm cavity. Unfortunately, the drift of polarization of the light over these fiber optic cables causes some of that input light to be rejected due to a mismatch of polarization. Figure 1 shows the trend of this drift over the course of a year by tracing the percent of light rejected due to polarization mismatch. This drift is inherent, and can be caused by factors such as thermal stress, mechanical stress, and irregularities in the shape of the fiber core. [5] To correct this drift, a polarization controller was installed to manually adjust the polarization across the fiber optic cables; this polarization controller is part of a larger fiber noise cancellation scheme. It is located in the corner station. [6]

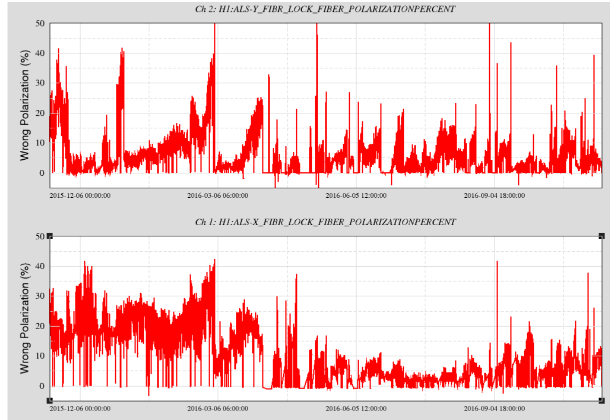


Figure 1: Percent of light rejected in ALS due to mismatch in polarization, observed over the course of a year. Ideally, this value is less than 5%; if this value gets too high, it must be corrected or observation is not possible. The sudden drops in polarization are due to active correction from the MPC. [4]

2.2 Motorized Polarization Controller (MPC)

The motorized polarization controller used at LHO in the ALS system is a dual channel MPC1-02 from FiberControl. It alters the state-of-polarization of single mode optical fibers using stress-induced birefringence. The optical fibers are formed into three Lefèvre loops, each mechanically connected to a paddle for independent motion [7]. These loops act as three effective fractional waveplates (or retarders); with these three degrees of freedom, they can transform any arbitrary waveform into any other. The first Lefèvre loop acts as a quarter waveplate ($\frac{\lambda}{4}$) to transform elliptically polarized light into linearly polarized light. The second acts as a half waveplate ($\frac{\lambda}{2}$) to rotate the linear polarization. The final acts as a quarter waveplate ($\frac{\lambda}{4}$) to transform back to elliptically polarized light, if required [5]. These three Lefèvre loops are labeled in Figure 2. As the paddles are physically rotated, the stress produced by this rotation changes the internal birefringence of the fibers, altering the polarization of the light.

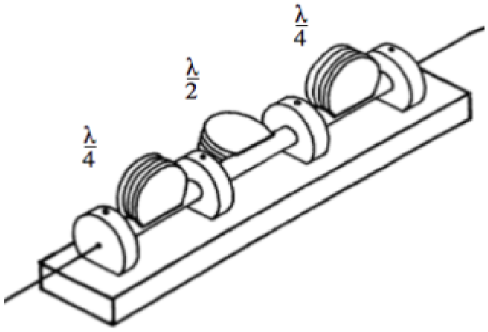


Figure 2: Fiber polarization controller consisting of three sequential Lefèvre loops. The orientation of these loops determines the state-of-polarization. [8]

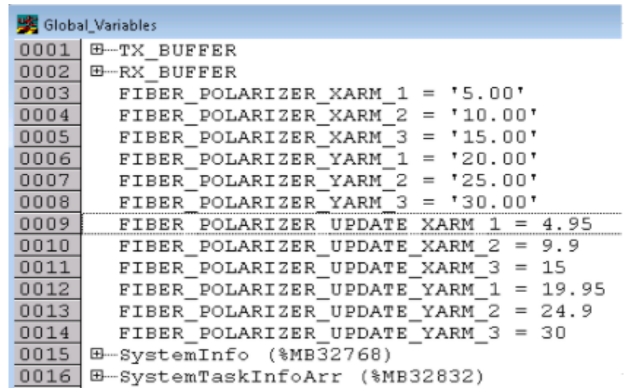


Figure 3: Command and response as shown in TwinCAT variables. These values will be shown in a user interface for the actual controls; for testing, they are manipulated through the software. The limited angular resolution of the MPC requires that the user be updated on the final angular position after each command is given.

3 Current Status of Project

In order to connect the MPC to existing software, the first goal was to build familiarity with programming industrial control systems using Structured Text and the TwinCAT 2 Programmable Logic Controller (PLC) software used by LHO. Next, the MPC was connected to the EtherCAT system, which is used to run slow controls hardware, using a serial connection. This system is controlled through TwinCAT, run on a rack-mounted PC and accessed remotely. [9] This TwinCAT software sets up a PLC in the IEC-1131 language standard and receives input and sends commands to the EtherCAT bus terminals, which in turn takes input and feeds output to the MPC.

Once the MPC was connected to the PLC software, the next goal was to establish communication and response. While the communication was easily established, the MPC did not initially respond as expected to simple commands. Instead, it echoed back any command that it was sent. After much trial and error, the proper syntax was discovered such that the MPC would implement any change requested.

For these digital controls, however, more than this simple command-and-response communication is required. Because the TwinCAT software will continuously loop through the code, there must a robust case state that is able to only move past start when a command is given, and return to start after it has been executed. The MPC also has limited angular resolution; thus, the case state must also inquire into the final angular position, and update the user with this information. This case state has proven successful for simple commands, as shown in Figure 3.

To mirror the functionality of the physical MPC, however, more complex commands were required. First, it essential to have the ability to scroll through angles, as the process for correction involves scanning each channel for the position with the lowest rejection. Since these positions are not known, it does not make sense to have remote controls centered only on direct angle input. This scrolling functionality was established by setting an angle step size, as well as buttons to scroll up and down that determine if this step size is added or subtracted from the current angular position. Other additional functions are rescan and center. While the center command mirrors the center button on the physical MPC, the rescan function is not necessary for physical controls. Because there is the possibility that someone could physically change the MPC without the knowledge of the user operating the remote controls, it is important to give that user the option to periodically rescan the MPC channels to ensure that the remote controls are accurate.

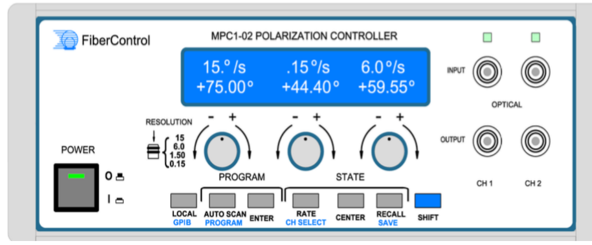
As the code expands, it also essential to have robust errors and error interpretation. If an error is encountered, the case state is returned to start, and the user is notified through a red box on the user interface around the break point. There are also associated error identifications to inform the user if the error is due to serial communication (error in sending or receiving), user input, or unexpected syntax from the MPC.

4 Future Work and Goals

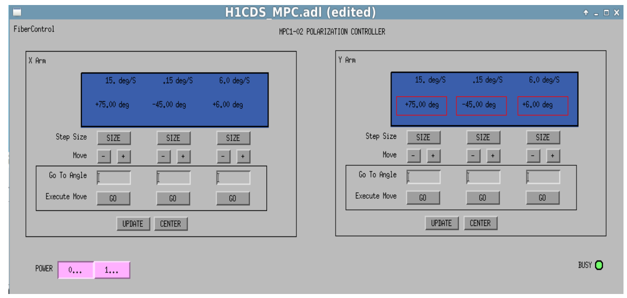
Although basic controls have been established for the MPC, and functionality increased, there are several problems that still need to addressed. First, because of the noise generated by continuously running the MPC, it is important that the remote controls include a power on/off switch, which will require external hardware as this command is not internal to the MPC. Another issue has arisen due to the translation of variable types across the user interface and the MPC. While the input through the MEDM is required to be a floating point number, the MPC interprets commands as strings. To translate, the TwinCAT function REAL-TO-STRING was used; this introduced error, however, as the function rounds exact decimal points in unpredictable ways, and produces a string with far too many decimals that is not equal to the floating point number input. To address this, the code must parse the resulting string by finding the decimal point and dropping the all but two of these digits, as the box only understands commands in the form of XX.XX. While it was determined that the loss of precision due to the angular resolution of 0.15 degrees of the box is more significant than the loss of precision of approximately 0.01 degrees from this rounding, it is still an issue that must be explored further.

After addressing these details of the code, the next goal will be to increase its usability; an interface for the controls will be created using MEDM (motif editor and design manager), which will write variables through the EPICS OPC server to the TwinCAT code. A prototype of these controls is shown in Figure 4, with a representation of the physical face of the MPC for comparison. Subsequently, the interface will be tested; after it is proven robust, it will be updated to the TwinCAT 3 system. Finally, the program will be installed and integrated into the LHO controls.

Another opportunity for the expansion of this program is the automation of finding the ideal settings for the MPC1. Currently, to achieve a polarization that results in less than 5% of the light rejected, the knobs of the MPC1 are adjusted at random until the percent of rejection is minimized. Now that basic computer controls have been established, it will be relatively simple to program a way to cycle through each knob until the minimum is reached, without manual adjustment. The storage of the remote control variables also makes it possible to examine the trend of the settings used on the MPC. Over time, it is possible that a pattern could emerge, and the root cause of the drift could be better understood.



(a)



(b)

Figure 4: (a) Graphic of the face of the MPC1-02. The two channels are controlled through the same face by switching channels manually. [7] (b) Prototype of user interface created using MEDM. Red boxes indicate values in error to the user. [2]

References

[1] B. P. Abbot et. al. *The Advanced LIGO Detectors in the Era of First Discoveries.* LIGO Document GW150914. 2016.

[2] David Barker. *Model of User Interface.* LIGO Wiki, Motorized Polarizer Controller EPICS Remote Control. 2017.

[3] A. Staley et. al. *Achieving Resonance in the Advanced LIGO Gravitational-Wave Interferometer.* LIGO Document P1400105. 2014.

[4] Jeffrey Kissel. *Trend of Polarization in ALS Fiber Transmission.* aLIGO LHO Logbook. 2016.

[5] LeFevre, H.C. *Fiber optic polarization controller.* Google Patents, US Patent 4,389,090. 1983.

[6] Matt Evens et. al. *Advanced LIGO Arm Length Stabilisation System Design.* LIGO Document T0900144-v6-D. 2015.

[7] FiberControl. *MPC-1 User and Programming Manual, Version 2-2.* LIGO Document T1200496-v1. 2003.

[8] Ajoy Ghatak and K Thyagarajan. *Introduction to Fiber Optics.* Cambridge University Press. 1998.

[9] *EtherCAT for Advanced LIGO.* LIGO Document G1200005-v1. 2012.