The Jefferson Lab Generic Lock Server

Brian Bevins

Outline

- Background – The Slow Locks
- The Problem – Making Feedback Easy
- Our Solution – The Generic Lock Server
- Implementation Details
- The PID Locks
- Results
- Future Work

The "Slow Locks"

- A collection of programs on back-end hosts for closed loop feedback at speeds ≤ 1 Hz
- Stabilize various beam parameters:
  - energy
  - orbit
  - current
  - various helicity-correlated effects
- For historical reasons each flavor had its own server, GUI, and code base.

The Problems

- **We need to consolidate the slow locks into a unified framework to improve performance, extensibility, and maintainability.**
- **In control systems for experimental equipment and facilities, new problems and new ideas are always changing the requirements.**
- **It is necessary to allow for quick prototyping of new control ideas and easy accommodation to temporary operating conditions.**

Testing Options

- **To test new feedback loops in a control system you can:**
  - **Reboot the front end computers**
    highly disruptive
  - **Restart a back end lock server**
    somewhat disruptive
  - **Create new locks on the fly**
    minimally disruptive
- **Our operators and system experts would like to test new locks at will without rewriting the software.**

Requirements

- **Rapid: New locks can be created at runtime with no disruption to the underlying control system or the operating machine.**
- **Distributed: All the locks are accessible from any operations console. No information is hidden.**
- **Arbitrary: Any process variable from any front end computer can be used for input or output of a lock.**

The Solution

- A "Generic" Lock Server and GUI have been developed to consolidate the slow locks.
- Four lock types have been integrated thus far:
  - general purpose Proportional-Integral-Derivative (PID)
  - beam polarization-correlated asymmetries in position and charge
  - beam currents
- The GUI allows all the locks to be viewed from any X display and controlled by any authorized user.

Implementation

- The server is highly object oriented and is implemented in standard C++.
  - heavy use of templates and the standard library to minimize the required code
- Each lock is a virtual CDEV device whose attributes can be accessed by CDEV-aware clients like StripTool and MEDM.
- I/O is through the CDEV client library.
  - all EPICS channel access PV's are available
  - also values from other CDEV servers such as the on-line model server, BPM server, or the lock server itself

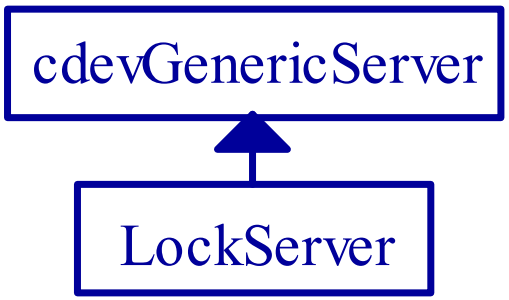
Implementation Cont'd

- All configuration information for the locks is stored in a human readable eXtensible Markup Language (XML) file.
- The XML configuration file is read and written using the Document Object Model (DOM) parser in the Qt toolkit from Trolltech, AS.
  - moving to Xerces from the Apache project
- The GUI is implemented in Tcl/Tk using the TclCdev package.

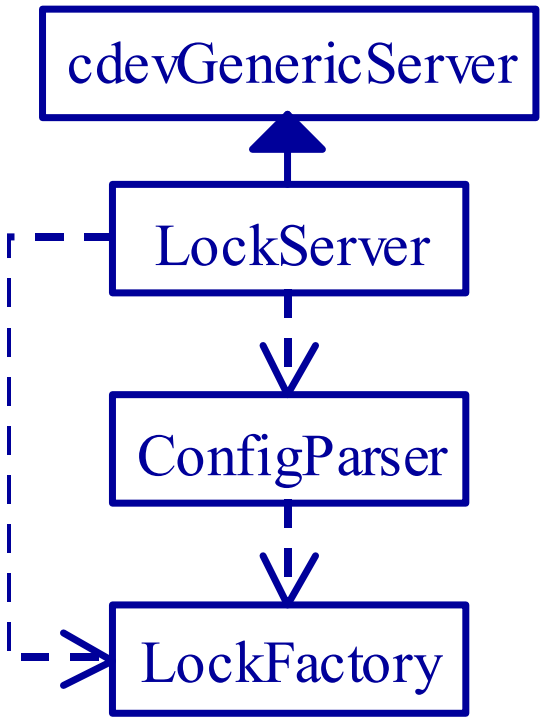
XML Configuration

```
<lockConfig>
    <Lock type="PIDLock" name="GangPhNL" />
    <device name="GangPhNL" >
        <attribute value="0" name="GainD" />
        <attribute value="0.2" name="GainI" />
        <attribute value="0.1" name="GainP" />
        <attribute value="ILI1L_PHASEerror" name="InputName" />
        <attribute value="4" name="Interval" />
        <attribute value="0.1" name="MaxChange" />
        <attribute value="0.1" name="MinChange" />
        <attribute value="30" name="MaxPos" />
        <attribute value="15" name="MinPos" />
        <attribute value="R1XXPSET" name="OutputName" />
        <attribute value="0" name="SetPoint" />
        <attribute value="North Linac First Pass Gang Phase"
          name="Description" />
        <attribute value="ILI1L_ERROR = 0" name="EnableString" />
    </device>
</lockConfig>
```

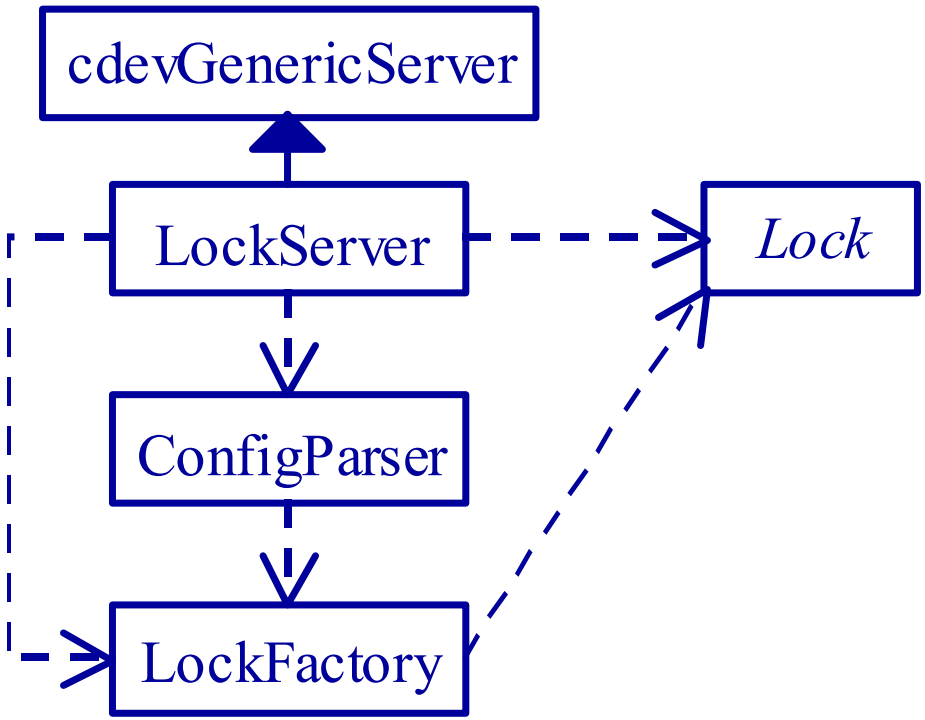
Server Class Diagram



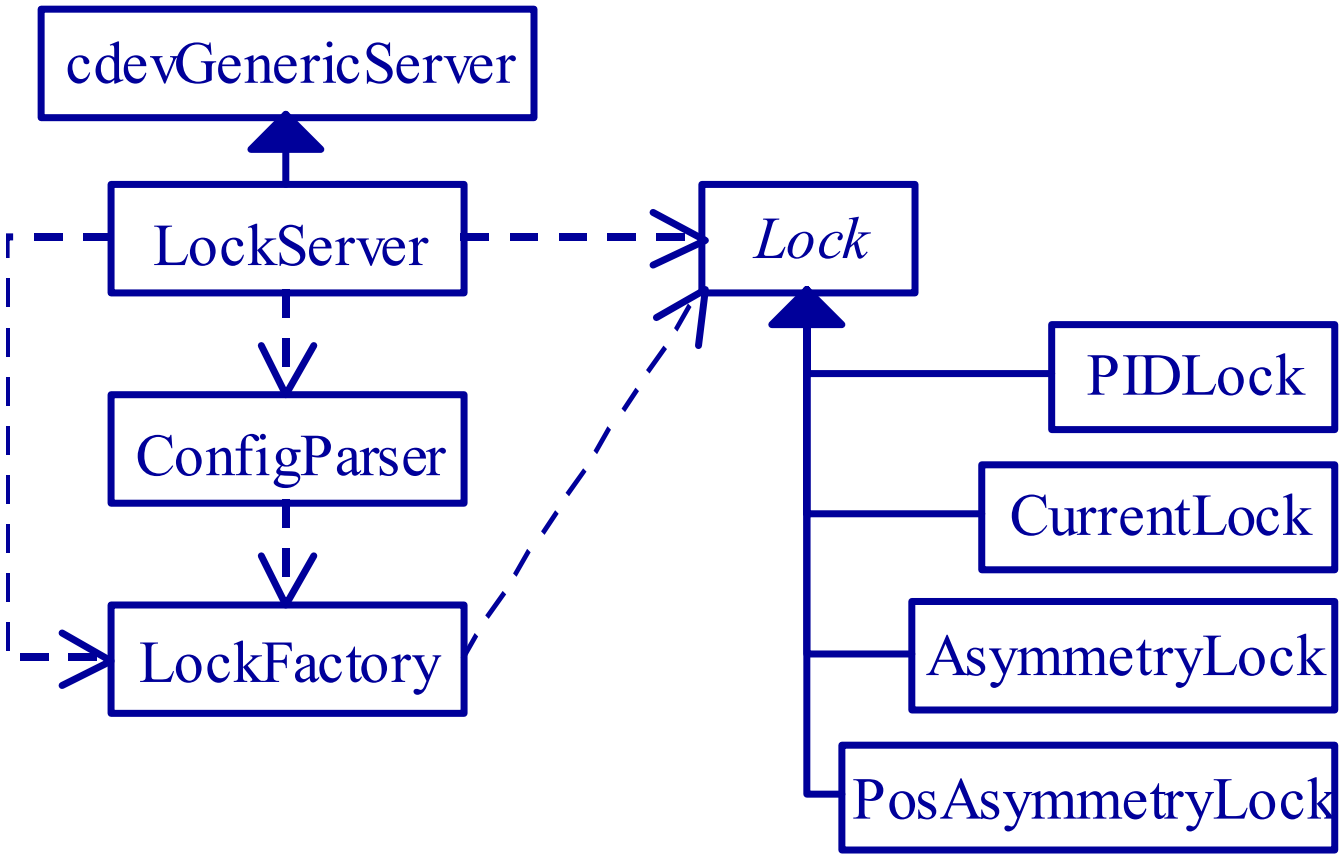
Server Class Diagram



Server Class Diagram



Server Class Diagram



PID Lock Features

- The server allows a user to create PID locks on the fly for any process variable using any other process variable.
- The PID functionality is based on the EPICS cpid record.
- Since the locks reside in a central server that enforces consistency, a new lock cannot be activated using an output variable that is in use by another lock.

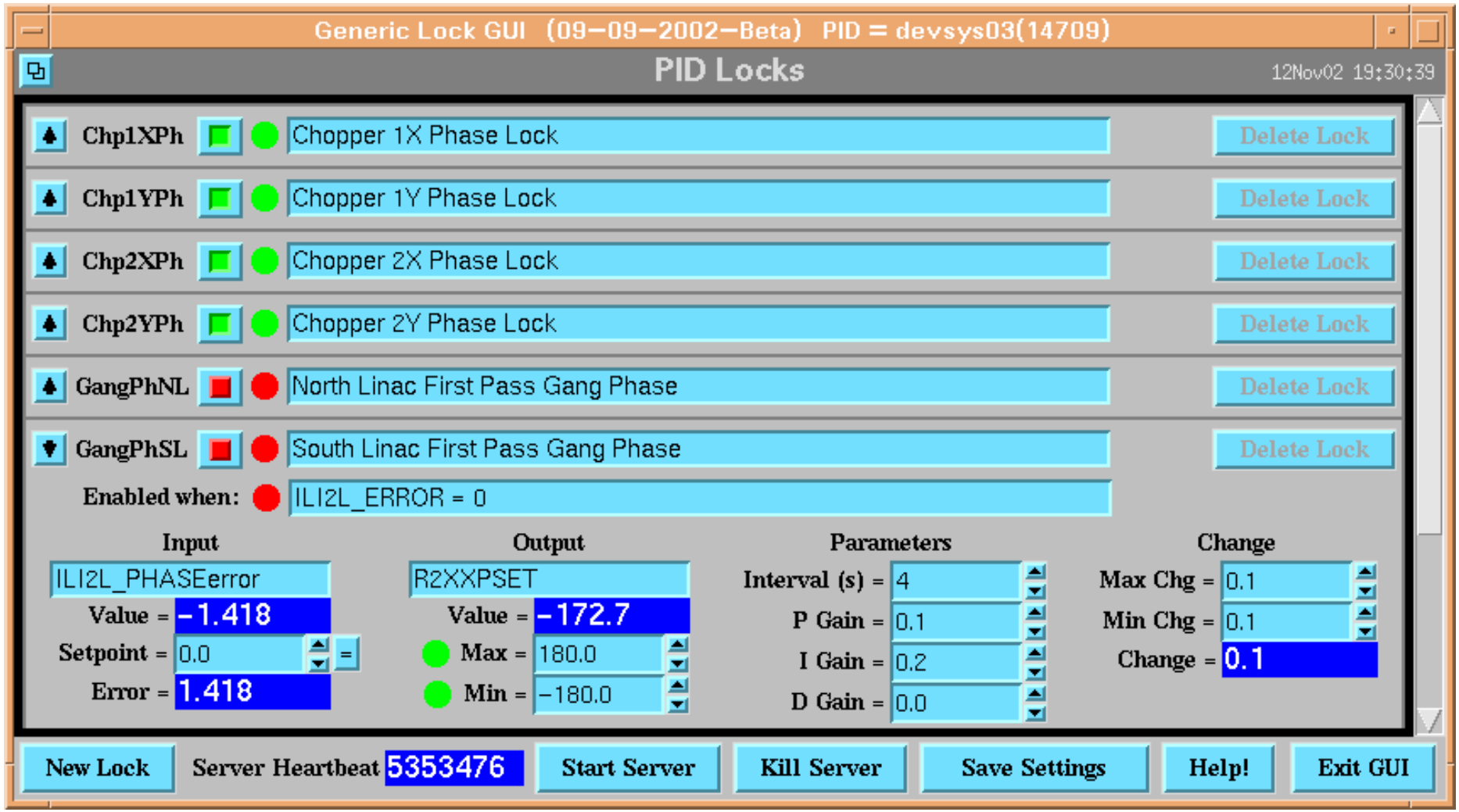
Input Expressions

- Instead of a PV name for the lock input, the user can enter an expression involving the values of up to 12 channels. For example:
  - `CTD1242.VAL - CTD1248.VAL`
  - `ATAN2(COS(D2R*(Chp1XPh.SetPoint-R011PMONphase)),SIN(D2R*(Chp1XPh.SetPoint-R011PMONphase)))*-R2D+Chp1XPh.SetPoint`
- All functions available to the EPICS calc record can be used except the C conditional operator "?:"
- A second expression can be entered that will disable the lock when it evaluates to false.

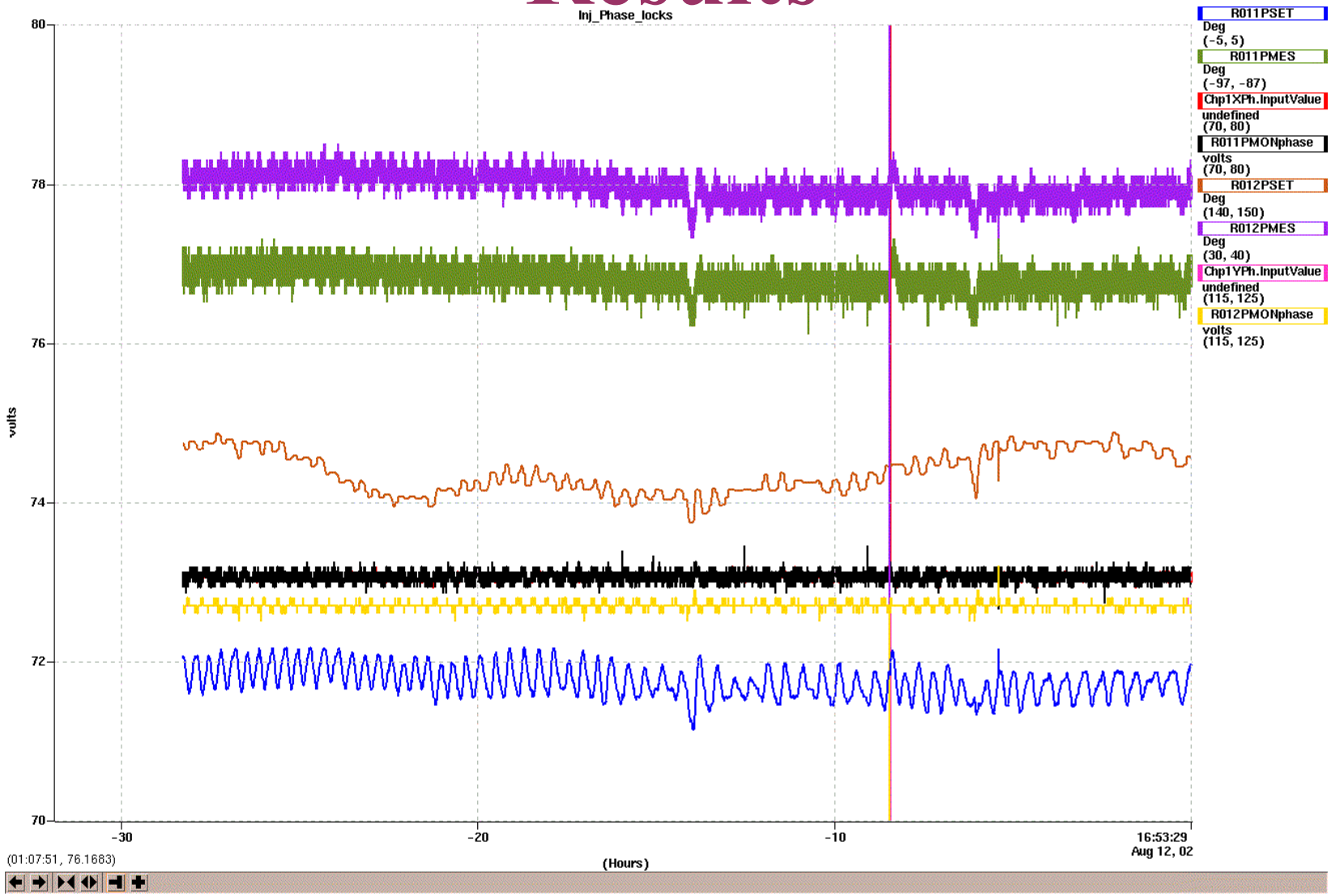
PID Lock Creation

1. Have a clever idea.
2. Start the PID Lock GUI and button to create a new lock.
3. Enter a name for the output (control) variable.
4. Enter a name or expression for the input (locked) variable.
5. Optionally enter an enabling expression.
6. Enter the desired set point, min/max output, etc.
7. Turn the lock on and adjust the gains as needed.

PID Lock GUI


Generic Lock GUI (09-09-2002-Beta) PID = devsys03(14709)
PID Locks
12Nov02 19:30:39
Chp1XPh
Chopper 1X Phase Lock
Delete Lock
Chp1YPh
Chopper 1Y Phase Lock
Delete Lock
Chp2XPh
Chopper 2X Phase Lock
Delete Lock
Chp2YPh
Chopper 2Y Phase Lock
Delete Lock
GangPhNL
North Linac First Pass Gang Phase
Delete Lock
GangPhSL
South Linac First Pass Gang Phase
Delete Lock
Enabled when:
ILI2L_ERROR = 0
Input
ILI2L_PHASEerror
Value = -1.418
Setpoint = 0.0
Error = 1.418
Output
R2XXPSET
Value = -172.7
Max = 180.0
Min = -180.0
Parameters
Interval (s) = 4
P Gain = 0.1
I Gain = 0.2
D Gain = 0.0
Change
Max Chg = 0.1
Min Chg = 0.1
Change = 0.1
New Lock
Server Heartbeat 5353476
Start Server
Kill Server
Save Settings
Help!
Exit GUI

Results


Inj_Phase_locks
R011PSET
Deg
(-5, 5)
R011PMES
Deg
(-97, -87)
Chp1XPh.InputValue
undefined
(70, 80)
R011PMONphase
volts
(70, 80)
R012PSET
Deg
(140, 150)
R012PMES
Deg
(30, 40)
Chp1YPh.InputValue
undefined
(115, 125)
R012PMONphase
volts
(115, 125)
80
78
76
74
72
70
volts
-30
-20
-10
16:53:29
Aug 12, 02
(Hours)
(01:07:51, 76.1683)

Future Work

- Use the Proxy IOC (PIOC) to host lock parameters as channel access PV's so that clients need not be CDEV aware.
- Other types of control loops will be added:
  - multiple inputs/outputs
  - model based feedback
- Automatic calibration and auto-tuning of PID loop gains
- A user specified function could be applied to the lock output before writing to the output channel.

Future Work

- **Security: The CDEV Generic Server engine does not have a built-in security model. A security layer will be added using Access Security.**

Reference

- ***A Distributed Feedback System for Rapid Stabilization of Arbitrary Process Variables***
  - **Brian Bevins and Alicia Hofler**
  - **Presented at ICALEPCS 2001**
  - **http://www.slac.stanford.edu/econf/C011127/WECT004.shtml**

Thank You!

The Jefferson Lab Generic Lock Server

Brian Bevins