Operational State and Servo Instability DMT Softw are

Keith Riles

University of Michigan

LIGO Scientific Collaboration Meeting

LIGO Livingston Observatory

March 18, 2000

Outline:

- Operational State Condition
  - Motivation
  - Definition of interface
  - Progress & expected completion time
- Serv o Instabilit y Detection
  - What we wan t to detect
  - Definition of interface
  - Progress & expected completion time

Operational State Condition Software

Before undertaking an analysis, one often needs to specify required machined conditions (Operational State)

Examples:

- Before seeking instability in Recycling Mirror (RM) servo:
  - Require lock of RM and Beam Splitter (BS) servos
  - Require wave front sensing engaged (or not engaged)
- Require laser intensity above certain threshold
- Require seismic RMS below ceiling (or above threshold)

Want to specify Boolean combinations of conditions:

Cond1 = "RM servo locked"

Cond2 = "BS servo locked"

Cond3 = Cond1 & Cond2

Want convenience of standard conditions,
i.e., "Full_Lock" or "IFO_quiet"

Also want run-time convenience of ascii configuration file

Operational State Condition Software

Sample Configuration File
(To require quiet seismic motion at Hanford LVEA)

```
[name]       [type]  [definition]          [double parms]  [int parms]

x_quiet        3     "H0:PEM-LVEA_SEISX"   0.    150.
y_quiet        3     "H0:PEM-LVEA_SEISY"   0.    150.
z_quiet        3     "H0:PEM-LVEA_SEISZ"   0.    300.
all_quiet      0     "x_quiet & y_quiet & z_quiet"
```

Types (so far – will expand with time)

- 0 – Boolean condition
- 1 – All values / channel / frame within Min / Max params
- 2 – Average value within Min / Max params
- 3 – Root-mean-square within Min / Max params

Operational State Condition Software

Internally, conditions defined by OperStateCond class with methods not (normally) accessed by users.

User interacts via a linked list (class OperStateCondList) of related condition objects

Example

Initialization:

```
OperStateCondList osclist;

osclist.readConfig(``myconfig.file'');
```

When processing frames:

```
if (osclist.satisfied(``all_quiet'') ) {
    do analysis
}
```

Operational State Condition Software

Progress to date:

- Learned rudiments of C++
- Most of code infrastructure in place
  (classes defined, most implementations done)
- Biggest item missing: Boolean parsing / evaluation code

Expected completion date: April 8

(First public release with documentation)

Servo Instability Detection Software

Servo "instability" refers here not only to real instability (runaway behavior), but also to any servo state with too high gain, giving excess noise just below the unity gain frequency.

Another worry is excitation of out-of-band resonances, *i.e.*, internal test mass normal modes

Want an early-warning system on all vulnerable servos to inform operator of impending problems

Signatures:

- Rapidly increasing band-limited rms in servo channel (broad band for gain peaking, narrow band for resonance excitation)
- Significant deviation from nominal spectral shape

Servo Instability Detection Software

Sample Configuration File
(Monitor gain peaking and out-of-band resonance

```
[channel]          [type] [report code] [real parms]

"H2:LSC-GW_TO"      1          10         100.   800.  1000.
"H2:LSC-GW_TO"      1          1000       10.    3002. 3003.
```

Types (needs more thought...)

1 Band-limited power (Fmin/Fmax) above threshold

2 Band-limited power (Fmin/Fmax) above long-term running average + n sigma

3 Time derivative of band-limited power (Fmin/Fmax) above threshold (over n samples)

Report codes - decimal bit pattern
(needs more thought and coordination with J.Z.)

0 – Do nothing

1 – Write report to local log file

10 – Write entry into meta-database

100 – Notify control room operator

1000 – Notify control room operator urgently!

Servo Instability Detection Software

Progress to date:

- Not much – focussed on operational state code
- But most of the coding is straightforward
- Coming up with reasonable default parameters and specifying default spectral contents will require real thought / work

Expected completion date: April 30

(First public release of infrastructure with documentation)

Testing on real servos will require more time and coordination with on-site physicist (or an extended visit to LHO)