



LIGO & VIRGO Controls Overview

Franco Carbognani, European Gravitational Observatory

Dennis Coyne, LIGO Laboratory, Caltech

Joint Gravitational Waves and CERN Meeting

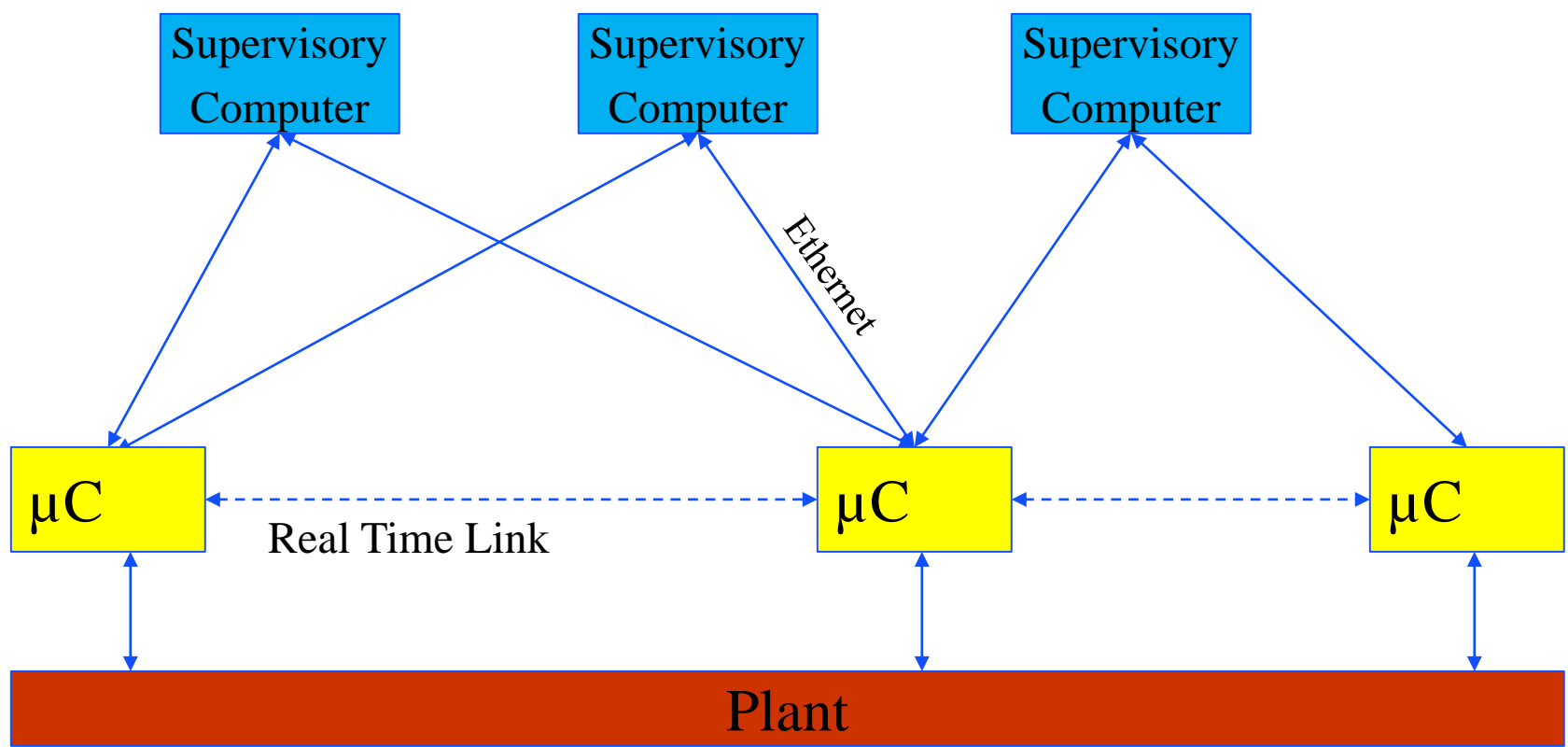
01 Sep 2017, Geneva

- ❖ **Distributed Control Systems**
- ❖ **Ligo Hardware architecture**
- ❖ **Ligo Software architecture**
- ❖ **Virgo Hardware architecture**
- ❖ **Virgo Software architecture**
- ❖ **“Plants” to be controlled**
- ❖ **Current/Future focus areas**

Overall references on the Advanced LIGO and Advanced Virgo detectors:

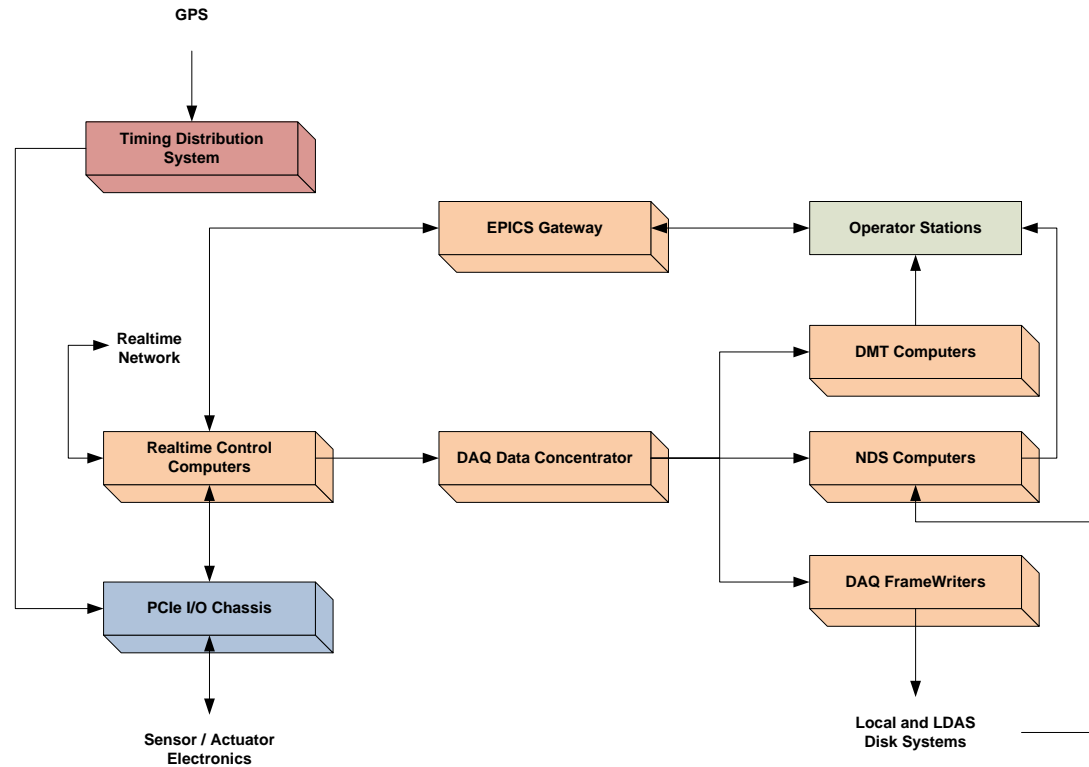
- *Advanced LIGO, [LIGO-P1400177](https://arxiv.org/abs/1411.4547), <https://arxiv.org/abs/1411.4547>*
- *The Sensitivity of the Advanced LIGO Detectors at the Beginning of Gravitational Wave Astronomy, LIGO-P1500260, <https://arxiv.org/abs/1604.00439>, Phys. Rev. D 93, 112004, June 2016.*
- *Advanced Virgo Technical Design Report, [VIR-0128A-12](https://arxiv.org/abs/1502.07588)*
- *Advanced Virgo: a second-generation interferometric gravitational wave detector, <https://arxiv.org/pdf/1408.3978.pdf>, F Acernese et al 2015 Class. Quantum Grav. 32 024001*

Distributed Control System(DCS)



Ligo Control & Data System Hardware Architecture Overview

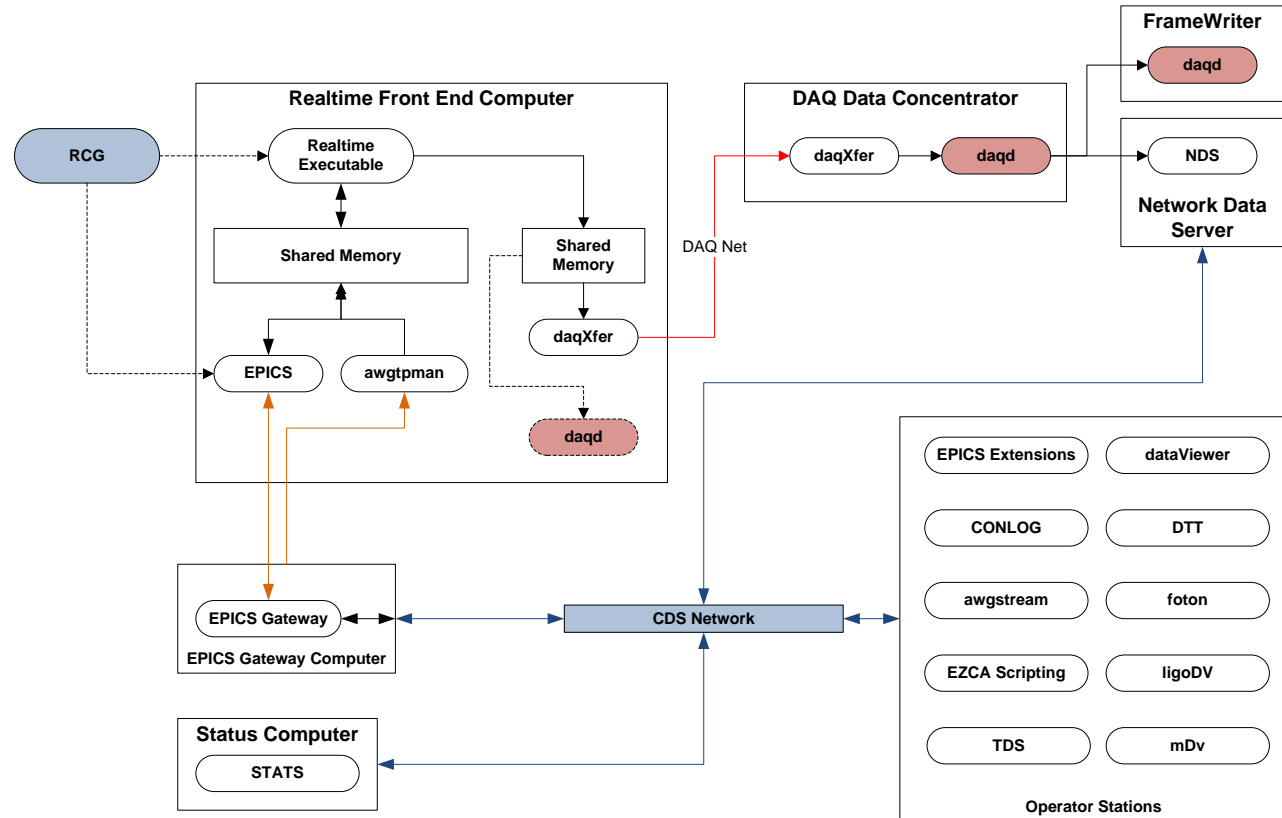
- ❖ **Timing derived from GPS**
- ❖ **Front-End Computers**
 - ❖ **hard, real-time**
 - ❖ **Linux real-time OS**
 - ❖ **multi-core, server class**
- ❖ **Fiber-linked PCIe I/O bus with 18-bit ADC/DAC**
- ❖ **Servo loop rates up to 65 kHz**
- ❖ **Synchronous, deterministic operation to within a few microseconds**



- *AdvLIGO CDS Design Overview, [LIGO-T0900612](#)*
- *New Control and Data Acquisition System in the AdvLIGO Project, [LIGO-P1100052](#)*

Ligo Control & Data System Software Architecture Overview

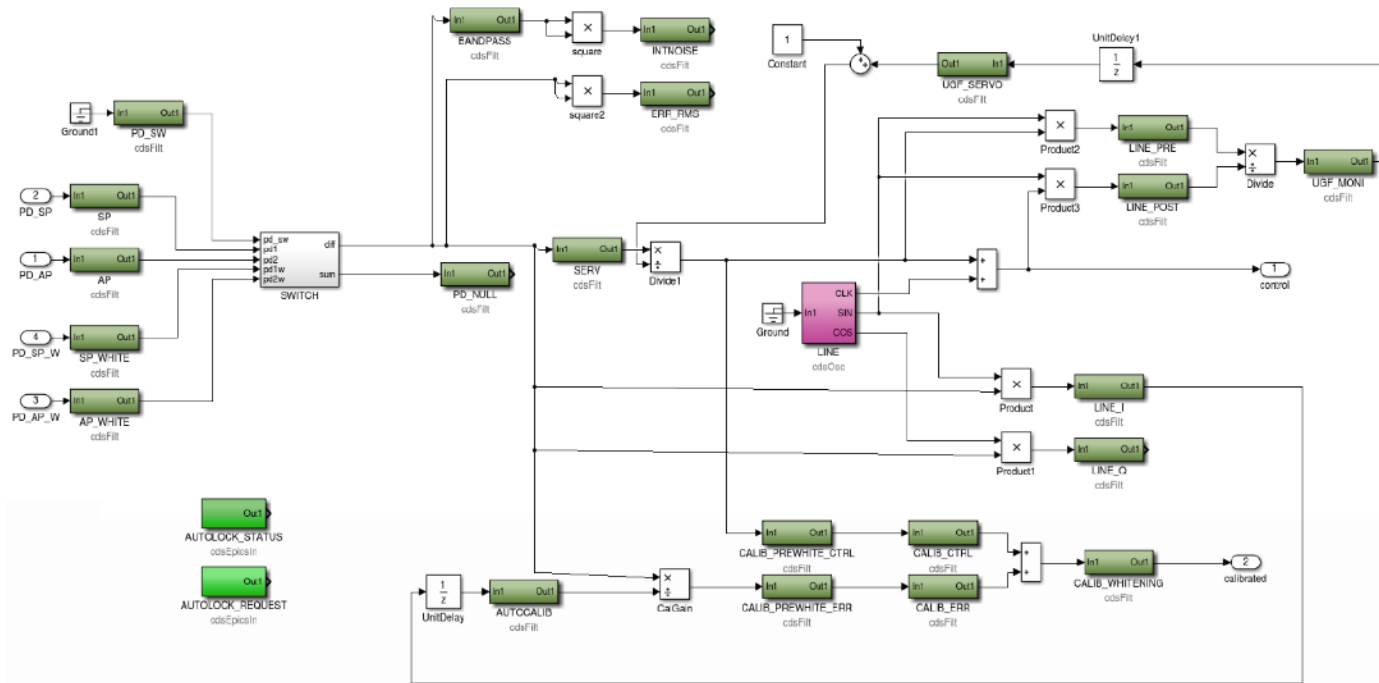
- ❖ **Real-Time Code Generator (RCG)**
 - ❖ Matlab Simulink graphical interface used to sketch control
- ❖ **EPICS**
 - ❖ Interface for setting parameters
- ❖ **Guardian**
 - ❖ State machine for sequencing



▪ *aLIGO, DAQ, Software Design Documentation, [LIGO-T1000625](https://www.ligo.caltech.edu/publications/LIGO-T1000625)*

Ligo real-time digital control

- ❖ Matlab/Simulink used as a graphical interface to sketch control system using standard blocks
- ❖ Generates real-time code to run on linux front-end machine

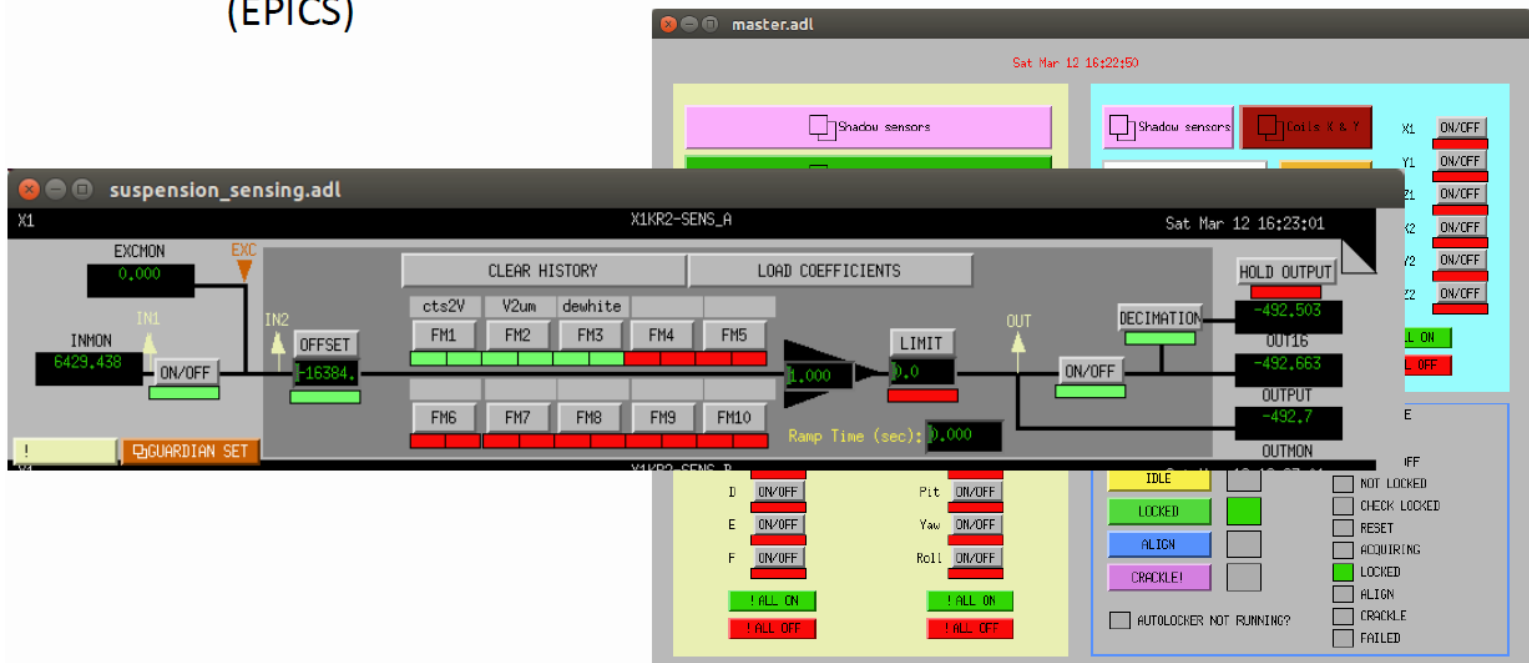


▪ Real-time Code Generator (RCG) Software Component Overview, [LIGO-T1200291](https://www.ligo.org/science/Gravitational-Wave-Observatories/VIRGO/Software/Real-time-Code-Generator-RCG-Software-Component-Overview)

Ligo real-time digital control

- ❖ Interface to the front-end, real-time “models” is via EPICS
- ❖ Change filters, gains, parameters
- ❖ Set Point Definition/Monitor software automates configuration control for the ~100k servo system parameters

(EPICS)

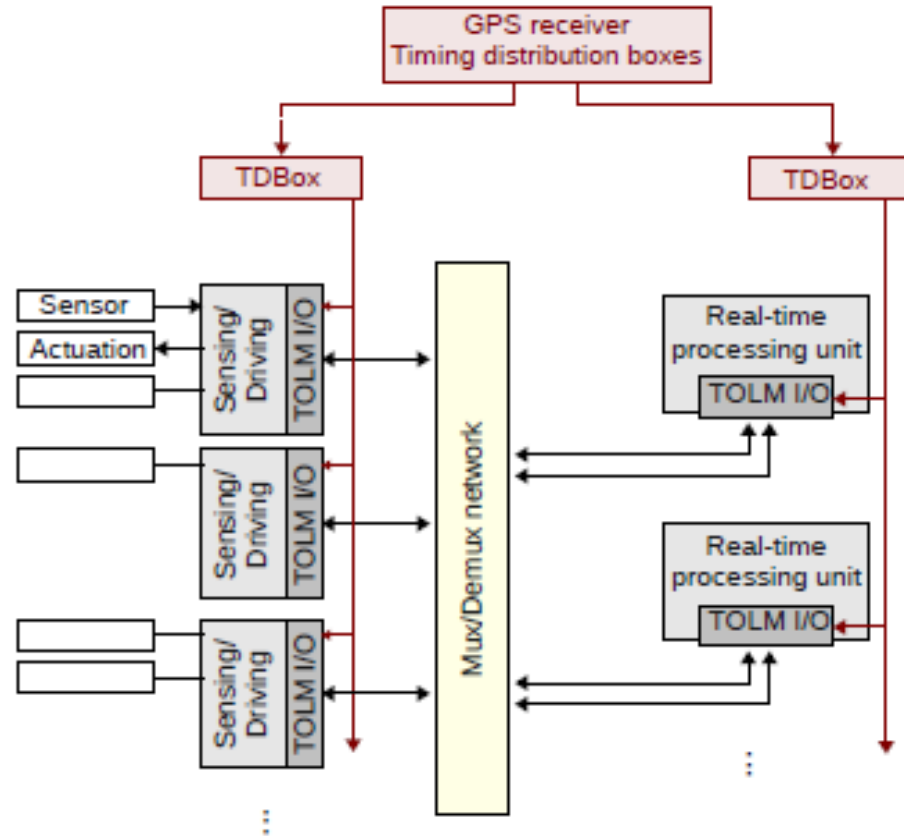


The screenshot displays two EPICS control windows. The foreground window is titled 'suspension_sensing.adl' and shows a control interface for 'X1KR2-SENS_A'. It includes various input fields like 'EXCHON' (0.000), 'IN1' (6429.438), and 'OFFSET' (-16384). A central section contains 'LOAD COEFFICIENTS' (FM1-FM10) and a 'LIMIT' field (0.0). The 'OUT' field shows values for 'DECIMATION' (-492.503), 'OUTPUT' (-492.663), and 'OUTMON' (-492.7). The background window is 'master.adl' showing a 'Shadow sensors' panel and a status column with 'ON/OFF' indicators for X1, Y1, Z1, X2, Y2, Z2, and E.

▪ Real-Time Code Generator (RCG) SDF Software, LIGO-T1500115

Virgo Control & Data System Hardware Architecture Overview

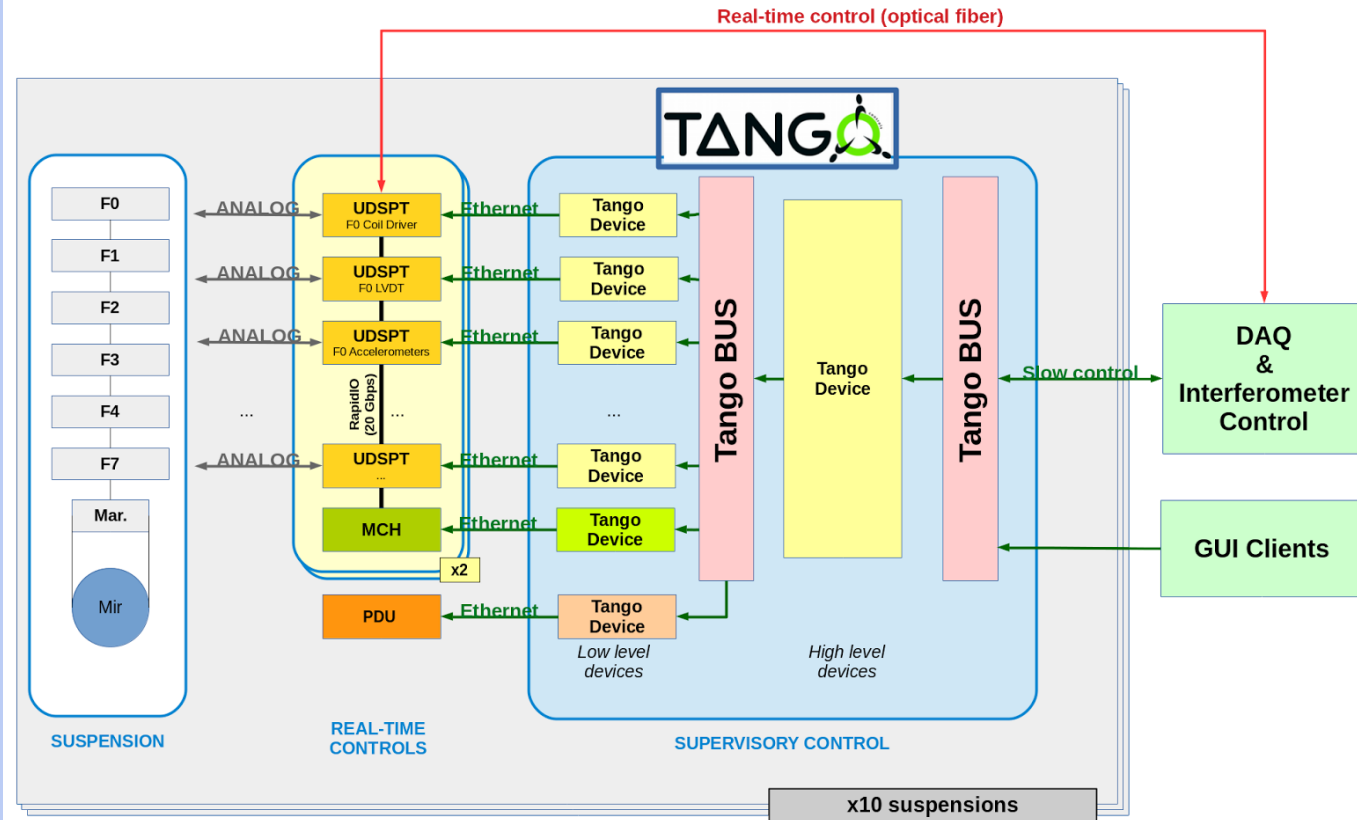
- ❖ Timing derived from GPS
- ❖ Front-End Computers (RTPC)
 - ❖ hard, real-time
 - ❖ Linux real-time OS
 - ❖ multi-core, server class
- ❖ Fiber-linked TOLM I/O bus, Mux/Demux network
- ❖ 18-bit ADCs, 24-bit DACs
- ❖ DaqBoxes: loops run up to 1MHz (SSFS)
- ❖ RTPC: standard loops run at 10 kHz.



▪ Advanced Virgo Technical Design Report, [VIR-0128A-12](#)

Virgo Control & Data System Hardware Architecture Overview

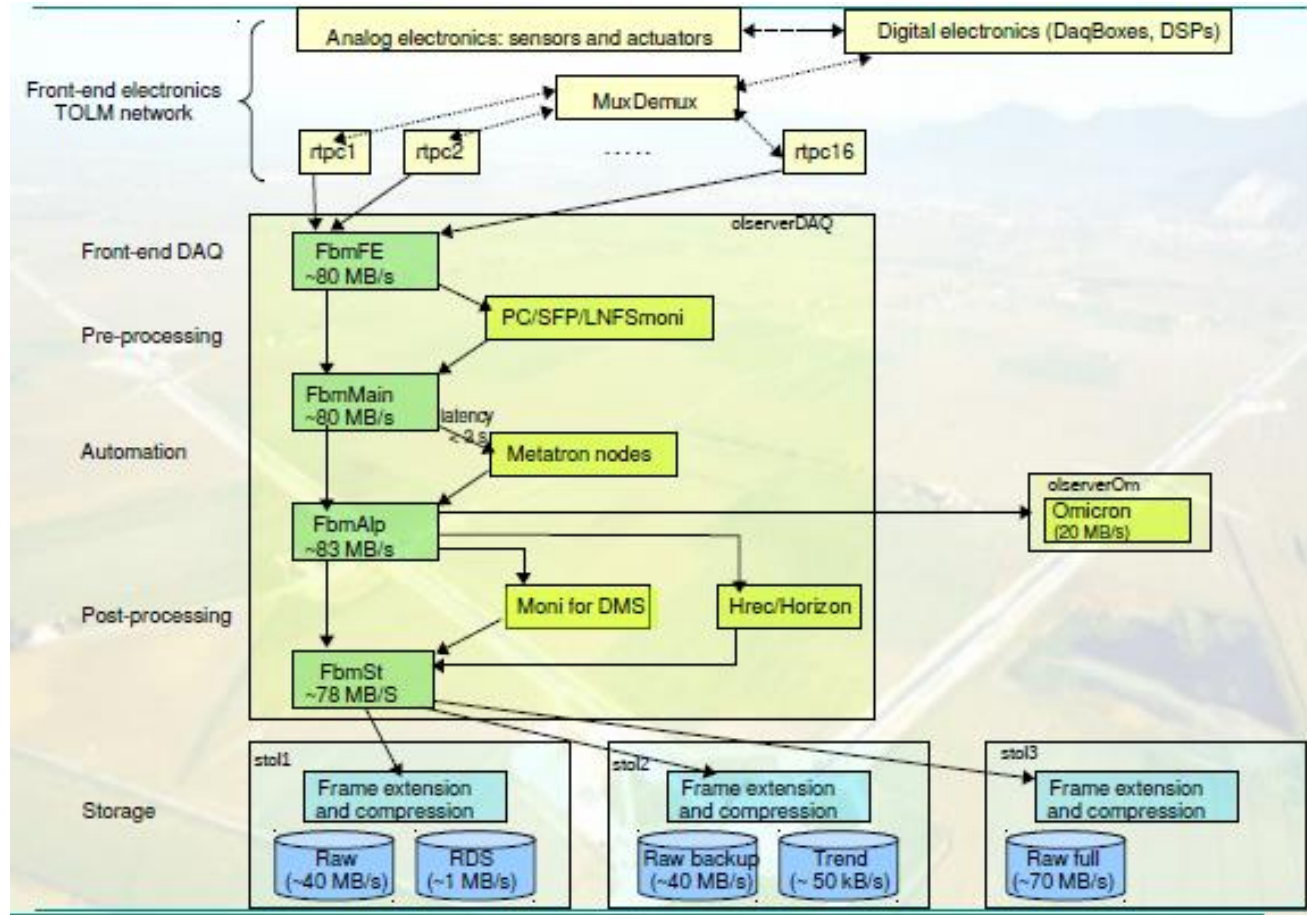
- ❖ UDSPT Boards
- ❖ 24bit ADC/ DAC
- ❖ Loops: 10 kHz (IP, F7, LC controls), 40 kHz (Global signals oversampling) and 320 kHz (Digital demodulation of sensors)



▪ Real Time Control Of Suspended Masses in Advanced VIRGO Laser Interferometer, [VIR-0299A-16](#)

Virgo Control & Data System Software Architecture Overview

- ❖ Damping & VPM interfaces for digital control development
- ❖ Tango & Cm/VPM
 - ❖ Interfaces for setting parameters
- ❖ Guardian/Metatron
 - ❖ State machine for automation



▪ DAQ sub-system status, [VIR-0343A-17](#)

Virgo real-time digital control

❖ Suspensions: Developed using the custom Damping interface

```

Level: Top
Virgo Inertial damping on [ 172.16.2.14 ] Page 1

Hardware implementation
BS_VLVDT_00.hrd

Ramp Time [100.00] Downsampling Factor [1]
Sampling Frequency [320000.00] Oversampling Factor [1]

Input Output Filename GUARD Gain Gname @Frequency When
ADC1 sc1 NULL no 1
ADC2 sc2 NULL no 1
ADC3 sc3 NULL no 1
ADC4 sc4 NULL no 1
ADC5 sc7 NULL no 1
SIG_GEN pr1 sine1 no 1
SIG_GEN pr2 sine2 no 1
SIG_GEN pr3 sine3 no 1
SIG_GEN pr4 sine4 no 1
SIG_GEN pr7 sine7 no 1
ADD phase1_1 no -83.44
ADD phase1_2 no 6.56
ADD phase2_1 no -85.32
ADD phase2_2 no 4.68
ADD phase3_1 no -83.76
ADD phase3_2 no 6.24
ADD phase4_1 no -82.06
ADD phase4_2 no 7.94
ADD phase7_1 no -75.95
ADD phase7_2 no 14.05
SIG_GEN mod_sin1 mod_sin1 no 1
SIG_GEN mod_cos1 mod_cos1 no 1
SIG_GEN mod_sin2 mod_sin2 no 1
SIG_GEN mod_cos2 mod_cos2 no 1
SIG_GEN mod_sin3 mod_sin3 no 1
SIG_GEN mod_cos3 mod_cos3 no 1
SIG_GEN mod_sin4 mod_sin4 no 1
SIG_GEN mod_cos4 mod_cos4 no 1
SIG_GEN mod_sin7 mod_sin7 no 1
SIG_GEN mod_cos7 mod_cos7 no 1
pr1 DAC1 NULL no 5
pr2 DAC2 NULL no 5
pr3 DAC3 NULL no 5
pr4 DAC4 NULL no 5
pr7 DAC5 NULL no 5
MIX mm1_sin mix_sin1 no 1.0
  
```

```

Level: Top
Virgo Inertial damping on [ 172.16.2.14 ] Page 2

Hardware implementation
BS_VLVDT_00.hrd

Ramp Time [100.00] Downsampling Factor [1]
Sampling Frequency [320000.00] Oversampling Factor [1]

Input Output Filename GUARD Gain Gname @Frequency When
MIX mm1_cos mix_cos1 no 1.0
MIX mm2_sin mix_sin2 no 1.0
MIX mm2_cos mix_cos2 no 1.0
MIX mm3_sin mix_sin3 no 1.0
MIX mm3_cos mix_cos3 no 1.0
MIX mm4_sin mix_sin4 no 1.0
MIX mm4_cos mix_cos4 no 1.0
MIX mm7_sin mix_sin7 no 1.0
MIX mm7_cos mix_cos7 no 1.0
mm1_sin lvd11 lp.flt no 600 0.0 after
mm1_cos lvd11_cos lp.flt no 600 0.0 after
mm2_sin lvd2 lp.flt no 600 0.0 after
mm2_cos lvd2_cos lp.flt no 600 0.0 after
mm3_sin lvd3 lp.flt no 600 0.0 after
mm3_cos lvd3_cos lp.flt no 600 0.0 after
mm4_sin lvd4 lp.flt no 600 0.0 after
mm4_cos lvd4_cos lp.flt no 600 0.0 after
mm7_sin lvd7 lp.flt no 600 0.0 after
mm7_cos lvd7_cos lp.flt no 600 0.0 after
lvd11 PROBE F1_VLVDT 1.0
lvd2 PROBE F2_VLVDT 1.0
lvd3 PROBE F3_VLVDT 1.0
lvd4 PROBE F4_VLVDT 1.0
lvd7 PROBE F7_VLVDT 1.0
  
```

■ *AdVIRGO Suspension Control, [VIR-0662A-17](#)*

Virgo real-time digital control

- ❖ Algorithms for Control and Locking (Acl)
- ❖ The startup parameters are read from a configuration file.
- ❖ Some parameters can be changed at run-time, either modifying the configuration file and reloading it or via Cm commands (VPM)

SSFS_Ctrl - Configuration

/virgoData/VirgoOnline/SSFS_Ctrl.cfg



```

215 ACL_SNS SSFS_ noise_driving
216
217
218 ▶ ## Apply the SSFS demodulation slope↔
234 ▶ ## Perform DC normalization↔
261 ▶ ## Photodiode selection (B2 or B4)↔
267 ▶ ## UGF calculation, this could be moved to a separate SSFS_moni process in case of trouble with computation time↔
278 ▶ ## calculation of demodulation phase mistuning↔
288 ▶ ## Compute the Error signal for MC
289 ACL_SUM_CH B4_Err "" 1 0 B4_Err_pre 1 B4_Err_post
290 ACL_CONST_CH SSFS_GAIN_NORM "" 1.0 SSFS_DSP_FREQ 0.75
291
292 ACL_OP_CH B4_56MHz_I_NORM "" B4_Err * SSFS_GAIN_NORM
293
294 ACL_FILTER_CH B4_MC_I_NORMflt "" 1 LOOP_FREQ B4_56MHz_I_NORM 1 DAQ_FILTER_10KHZ
295 ACL_LF_CH B4_MC_I_NORM_10KHz 10000 B4_MC_I_NORMflt "last"
296 ACL_DAQ "SSFS_" "" B4_MC_I_NORM_10KHz
297
298 ACL_DAQ_FILTER "SSFS_" "_10kHz" 10000 DAQ_FILTER_10KHZ B4_selectFilter ;
299
300
301 TOLM_PROCESSOR_OUTPUT_PACKET_WITH_FREQUENCY "SSFStoINJ" 10000. 0 "" 2 0x07 0x07 0x1F 0
302 SSFS_B4_MC_I_NORM_10KHz SSFS_B4_selectFilter_10kHz ;
303
304 ▶ ## Compute the SSFS I and Q signals (for ISC)↔
328 ▶ ## Channels to be used by the SSFS_DSP↔

```

Virgo real-time digital control

- ❖ Algorithms for Control and Locking (Acl)
- ❖ The startup parameters are read from a configuration file.
- ❖ Some parameters can be changed at run-time, either modifying the configuration file and reloading it or via Cm commands (VPM)

SSFS_Ctrl ▲ GPS: 1188157847-000012100(50000) - SSFS_ON 1, LNFS_56MHz_phi 2.37, B4_Err_pre 0.00849, B4_Corr -0.0436, UGF_FILT_ON 1, B4_GAIN 750, B4_selectFlt0 0, B4_selectFlt1 1, noise_sensing -0.000122, noise_driving 5.55e-06, B4_PD_select 0,

Application: VirgoApp/Acl/v0r42p11/Linux-x86_64-SL6-2.6.32.11-x86_64-hal-2.6-03-rtai-3.8.1/Acl.exe
 Host: rtpc12.virgo.infn.it Pid: 29095 Process uptime: 33 days and 12h50m53s

🔗 Configuration: VirgoData/VirgoOnline/SSFS_Ctrl.cfg
 🔗 Log: VirgoLog/VirgoOnline/SSFS_Ctrl/SSFS_Ctrl_2017-07-28-06h59m37-UTC.log
 🔗 Documentation: VirgoApp/Acl/v0r42p11/doc/Acl.pdf

Reload config Reload config and synchro Reload config and reset Get ADC channel offset... ACL_FILTER_CH - reset the filter...

Photodiode: Use B4 DC normalization Use fixed normalization Use B4 PD1 Use B4 PD2

SSFS filter: Reload Filters B4_GAIN... B4: reset all DSP filters

B4 DSP filter0: Reset Select

B4 DSP filter1: Reset Select

Noise generation: offset... white noise... coloured noise... sinewave... ramp... select noise...

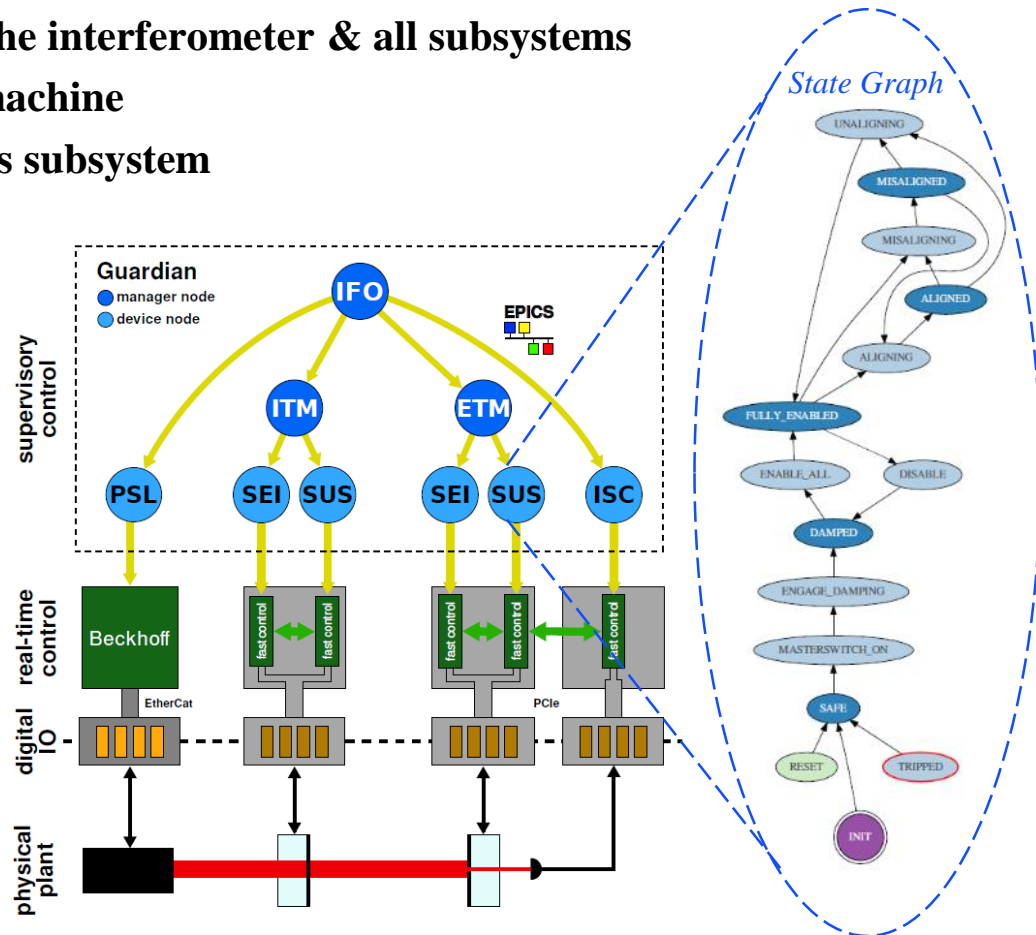
Sensing noise: set noise ampl... UGF line...

Driving noise: set noise ampl... Disable NNS Enable NNS

🔗 SSFS filters: VirgoData/VirgoOnline/SSFS_Filters.cfg

the Guardian

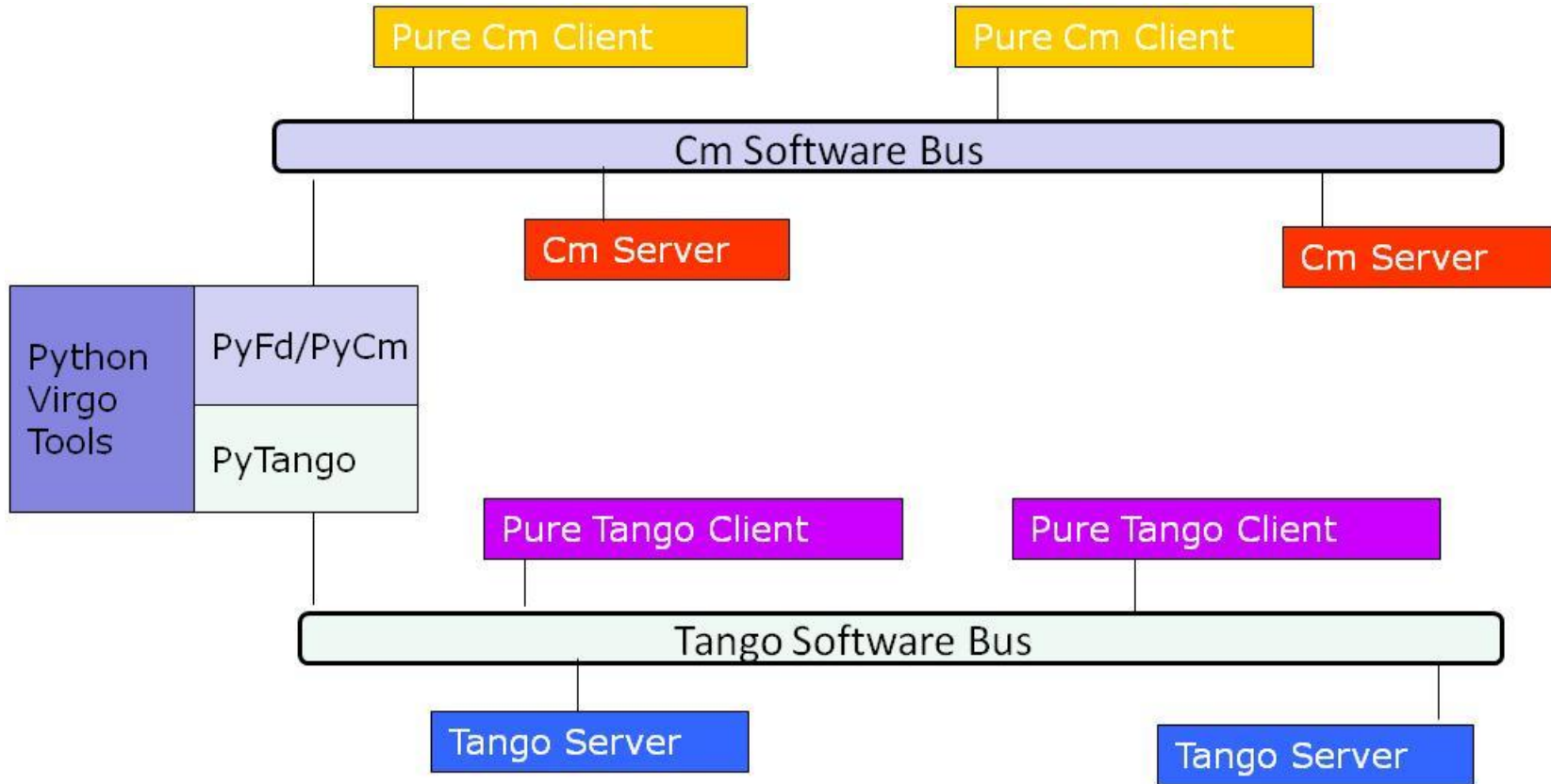
- ❖ Robust framework for automation of the interferometer & all subsystems
- ❖ Hierarchical, distributed, finite state machine
- ❖ Each node executes a state graph for its subsystem
- ❖ Supports commissioning & operation
- ❖ EPICS interface
- ❖ Python code
- ❖ Adopted & adapted by Virgo



- *Advanced LIGO Guardian Documentation, [LIGO-T1500292](#)*
- *Distributed State Machine Supervision for Long-baseline Gravitational-wave Detectors with the Guardian Automation Platform, [LIGO-P1600066](#), <https://arxiv.org/abs/1604.01456>, Rev. Sci. Instrum. 87 (2016) 094502*

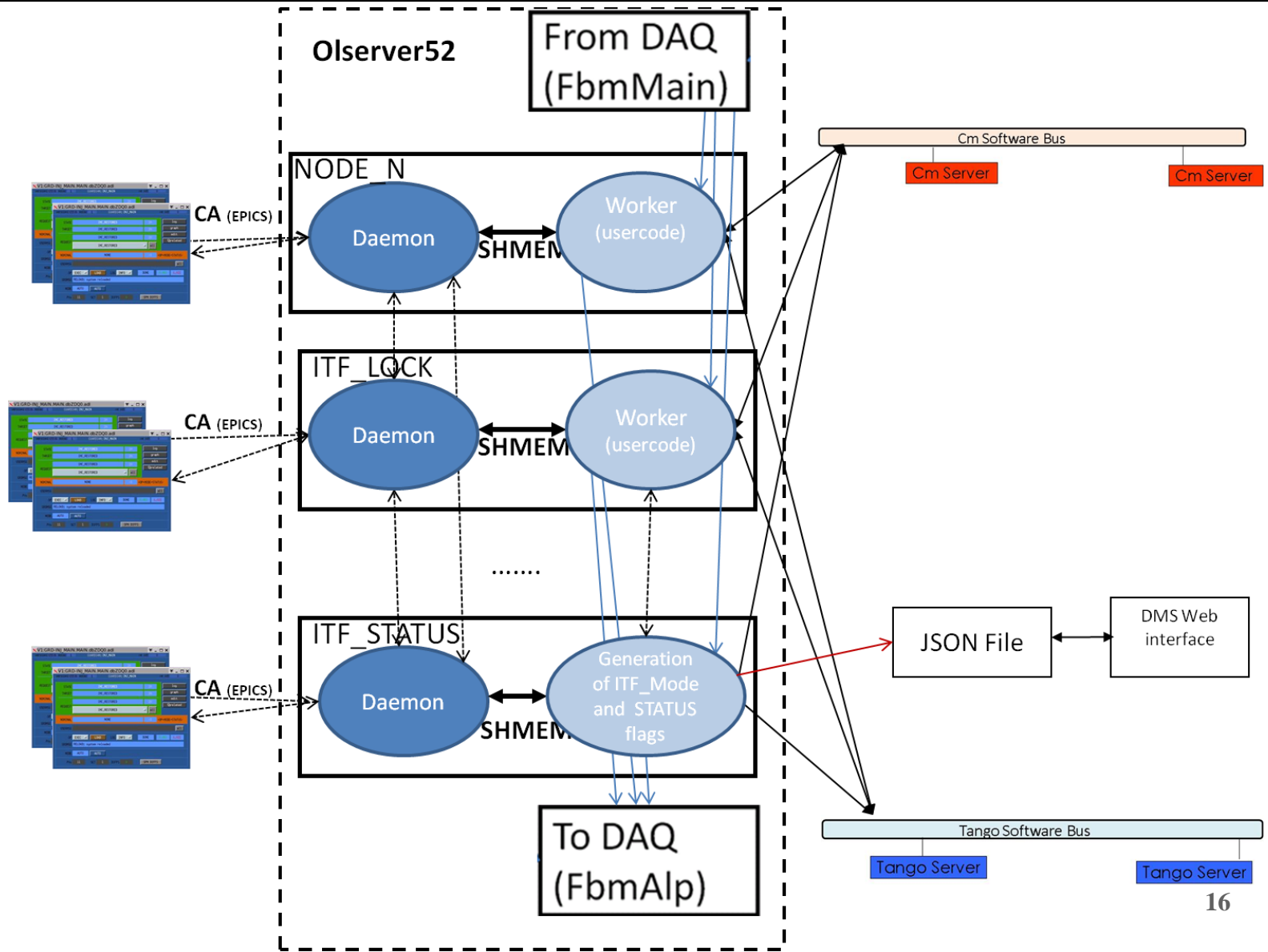
the Virgo Guardian: Metatron

Dealing with an hybrid system using both  and our legacy communication middleware *Cm*

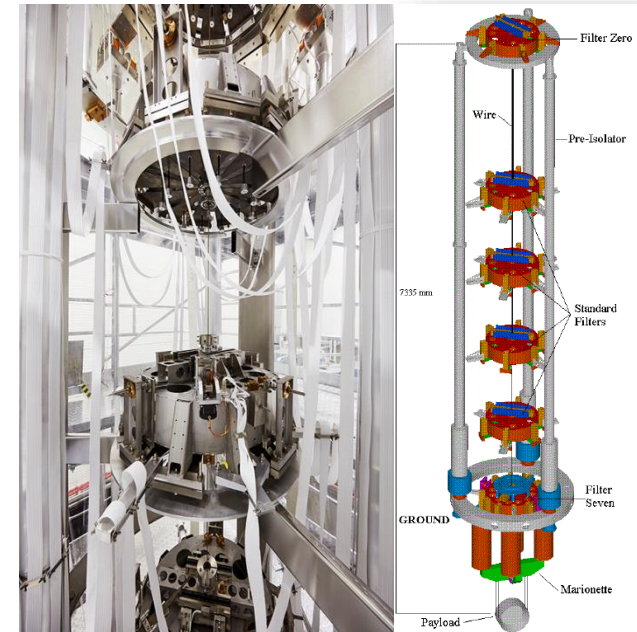
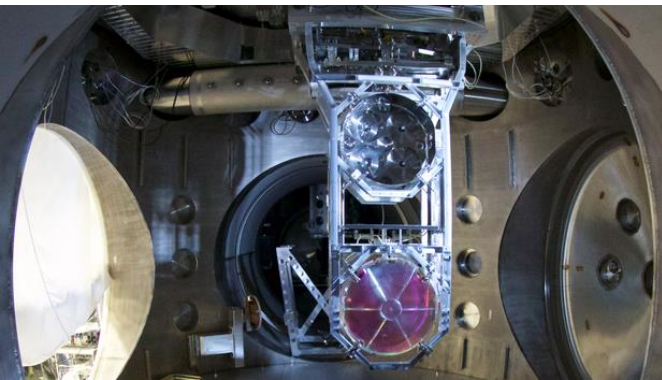


Problem managed by exploiting the existing PyTango and generating a Cm (and Fd) C library bindings using ctypesgen (PyCm and PyFd)

the Virgo Guardian: Metatron

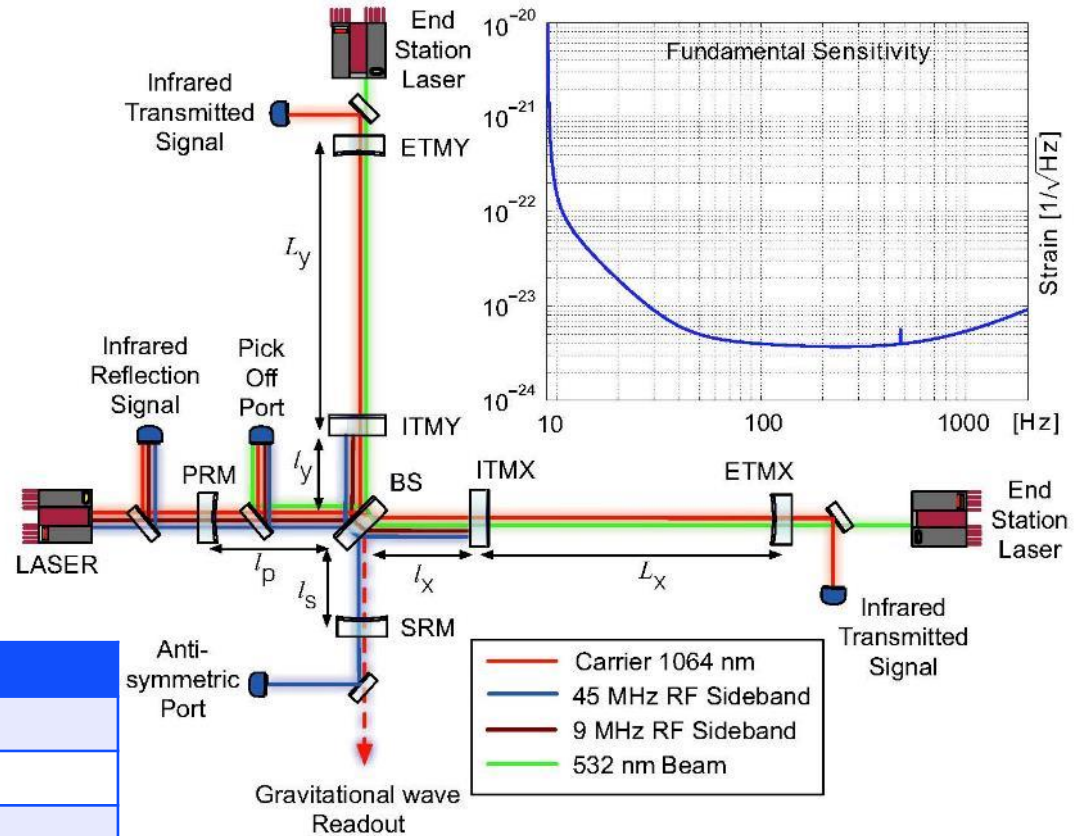


- ❖ **Pre-Stabilized Laser (PSL)**
 - ❖ Frequency, pointing & intensity stabilization
- ❖ **Ligo**
 - ❖ (SEI)
 - ❖ Isolated platforms for optics
 - ❖ **Suspensions (SUS)**
 - ❖ Single, double, triple & quadruple pendulum suspensions
- ❖ **Virgo**
 - ❖ **Suspensions (SUS)**
 - ❖ Superattenuators + Payloads
- ❖ **Interferometer Sensing & Control (ISC)**
 - ❖ Length
 - ❖ Angle



Ligo Interferometer Length Sensing & Control

- ❖ Nonlinear cavity lock acquisition control
- ❖ Length derived from RF demodulated signals
- ❖ Five resonant cavity lengths
- ❖ Arm Length Stabilization (ALS)
 - ❖ Acquire lock with lower finesse at doubled frequency (green wavelength) first

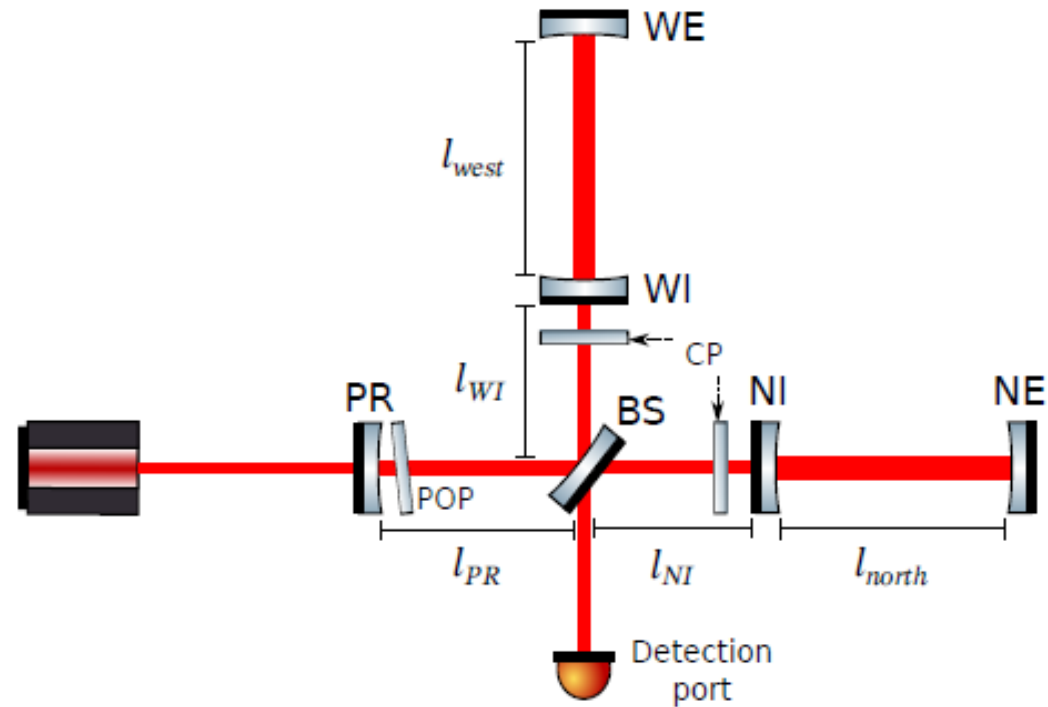


Mode	Definition
Common arm length (CARM)	$(L_x + L_y)/2$
Differential arm length (DARM)	$L_x - L_y$
Power recycling cavity length (PRC)	$l_p + (l_x + l_y)/2$
Signal recycling cavity length (SRC)	$l_s + (l_x + l_y)/2$
Michelson length (MICH)	$l_x - l_y$

■ Achieving Resonance in the aLIGO Interferometer, [LIGO-P1400105](#), Class. Quantum Grav. 31 (2014) 245010
 ■ CARM/ALS Electro-Optical Controls Diagram, [LIGO-G1500456](#)

Virgo Interferometer Length Sensing & Control

- ❖ Four resonant cavity lengths
- ❖ 6MHz Sidebands used to lock the arm cavities
- ❖ 56MHz used for CARM, DARM, MICH
- ❖ 8 MHz used for PRCL



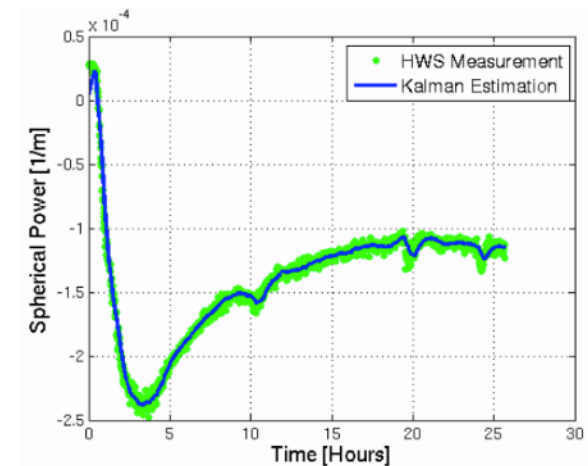
Mode	Definition
Common arm length (CARM)	$(l_{west} + l_{north})/2$
Differential arm length (DARM)	$(l_{north} - l_{west})/2$
Power recycling cavity length (PRCL)	$l_{PR} + (l_{NI} + l_{WI})/2$
Michelson length (MICH)	$l_{NI} - l_{WI}$

▪ *Development of a New Lock Acquisition Strategy for the Arm Cavities of Advanced Virgo,*
www.infn.it/thesis/thesis_dettaglio.php?tid=10879

Auxiliary Loops

Many additional, essential loops, many of which are not completely independent of the global interferometer controls:

- ❖ **Earth tidal correction**
- ❖ **Output Mode Cleaner (OMC) alignment**
- ❖ **Wavefront Sensor (WFS) centering**
- ❖ **Input Mode Cleaner (IMC) alignment**
- ❖ **Thermal Compensation System (TCS)**
- ❖ ...

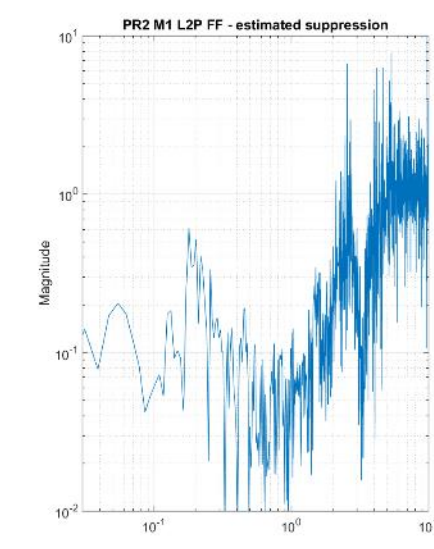
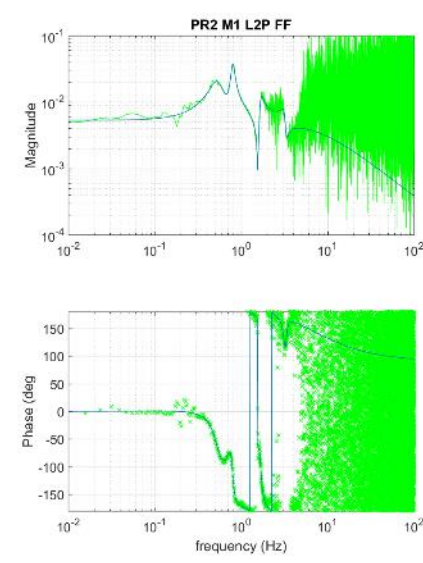
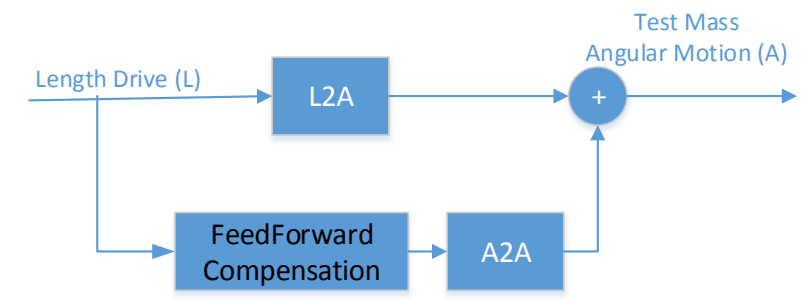


Wavefront sensor measurement and Kalman estimator

- *The Adv. LIGO Input Optics, LIGO-P1500076, <http://dx.doi.org/10.1063/1.4936974>, Rev Sci Instrum vol. 87 pg. 014502.*
- *Locking the Advanced LIGO Gravitational Wave Detector: with a focus on the Arm Length Stabilization Technique, LIGO-P1500273, <http://dx.doi.org/10.7916/D8X34WQ4>*
- *Kalman Filter for the Thermal Compensation System, LIGO-G1501532*
- *LSC-Virgo August 2017 Meeting @CERN, Thermal Compensation in Advanced Virgo, [VIR-0670A-17](http://dx.doi.org/10.1063/1.4936974)*

Ligo: Current/Future Focus Areas

- ❖ Applications of Machine Learning (ML) to Controls
 - ❖ Lock Maintenance
 - ❖ Lock Acquisition²
- ❖ Length to Angle (L2A) decoupling
- ❖ Feedback optimization (esp. applied to angular controls)
- ❖ System Identification
- ❖ Interferometer robust configuration for earthquakes³
- ❖ State space control for the Real-Time Code Generator (RCG) Software
- ❖ *More generally, we are working to inject more modern control techniques to improve performance & robustness*



1) CSWG wiki page: <https://wiki.ligo.org/viewauth/CSWG/WebHome>
 2) LSC-Virgo August 2017 Meeting @CERN, Deep leaning applied to lock acquisition, LIGO-G1701589
 3) LSC-Virgo August 2017 Meeting @CERN, Earthquake early warning & response, LIGO-G1701593

Virgo: Current/Future Focus Areas

- ❖ **SSFS upgrade to 1 MHz**
- ❖ **Mux-Demux and TOLM network upgrade**
- ❖ **Timing system upgrade**
- ❖ **SUS: Global Inverted Pendulum Control (GIPC) is a technique already used in VIRGO in which common and differential error signals are used to control the IP top stage instead of the local LVDTs and Accelerometers**
- ❖ **SUS: Optimal Control (ex: Kalman estimator and an associated optimal controller for vertical control)**
- ❖ **Converge on a unique Communication Middleware / Control Framework solution**
- ❖ *Applications of Machine Learning (ML) to Controls*

Thanks for your attention!

