Real Time Control Of Suspended Masses In Advanced VIRGO Laser Interferometer

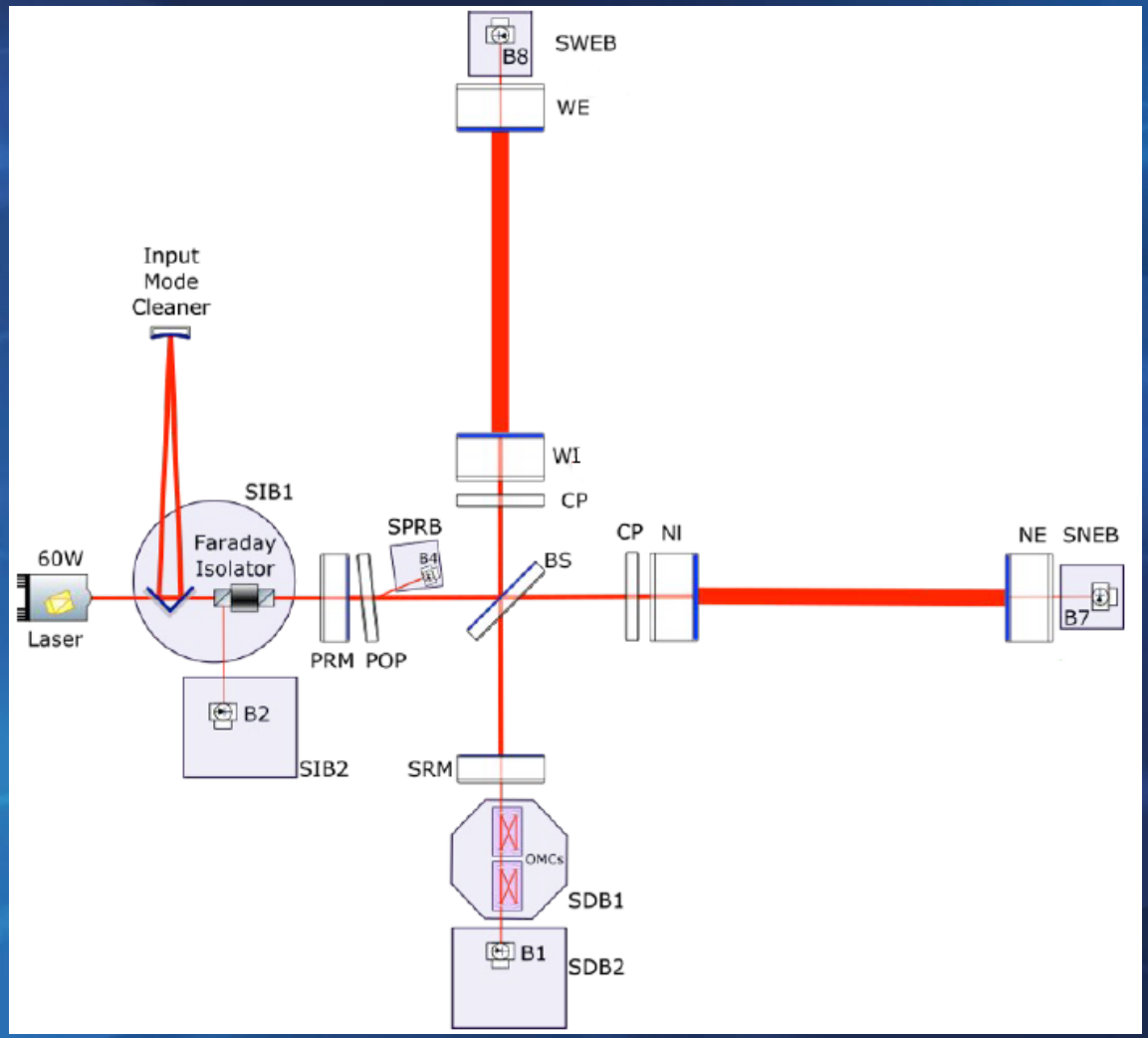
Alberto Gennai
for the VIRGO Collaboration

http://www.virgo-gw.eu/

20th Real Time Conference 5-10 June 2016 Padova, Italy

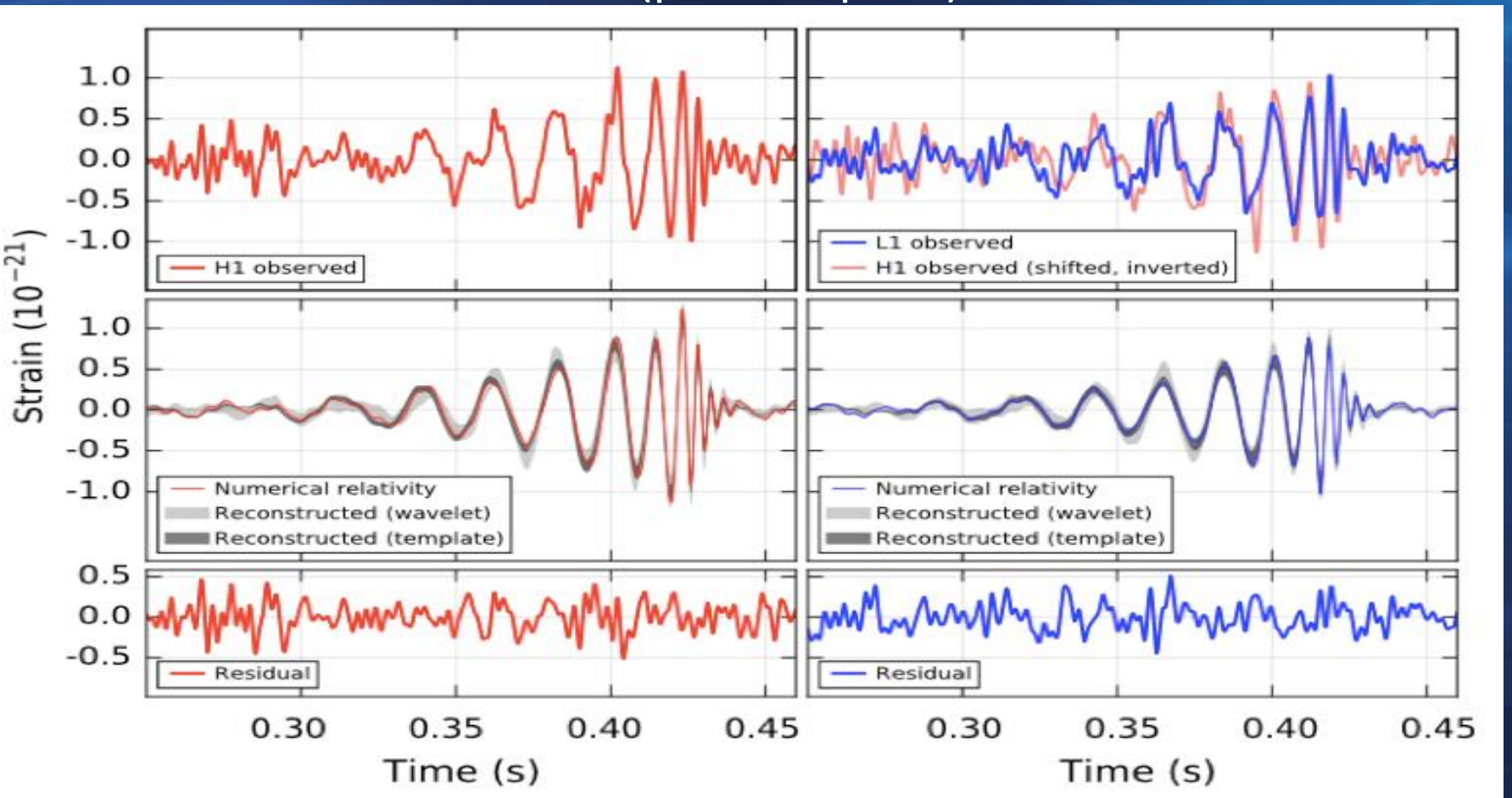
- Virgo is a laser interferometer designed to detect and observe gravitational waves. It is operated in Cascina, near Pisa on the site managed by the consortium European Gravitational Observatory (EGO), by an international collaboration of scientists from France, Italy, the Netherlands, Poland, and Hungary

VIRGO Interferometer



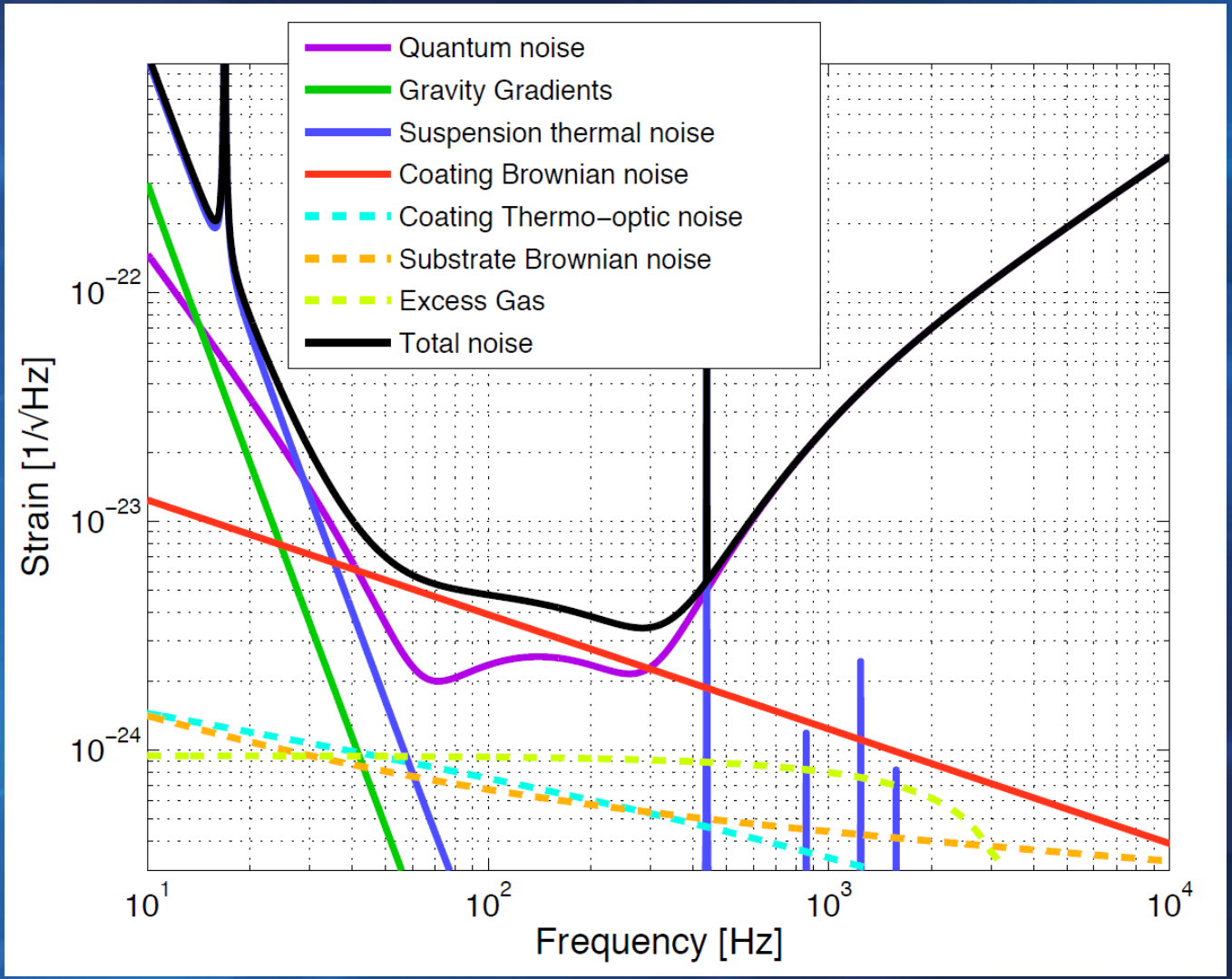
Gravitational Waves

- GW Signal is incredibly small: actual relative displacement is in the order of 6-8 x 10^{-18} m (peak-to-peak)

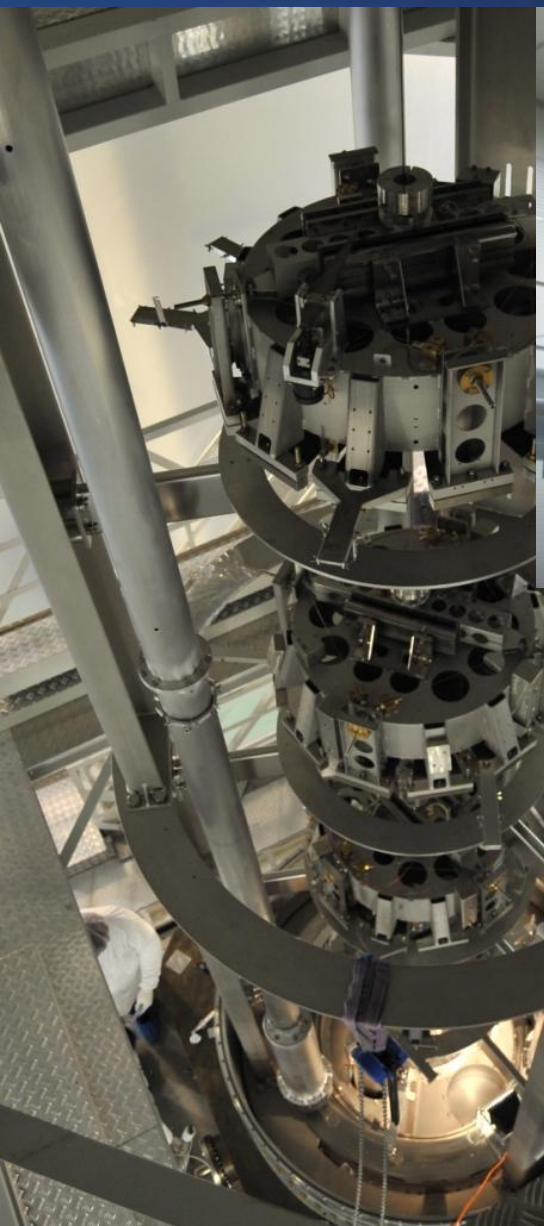


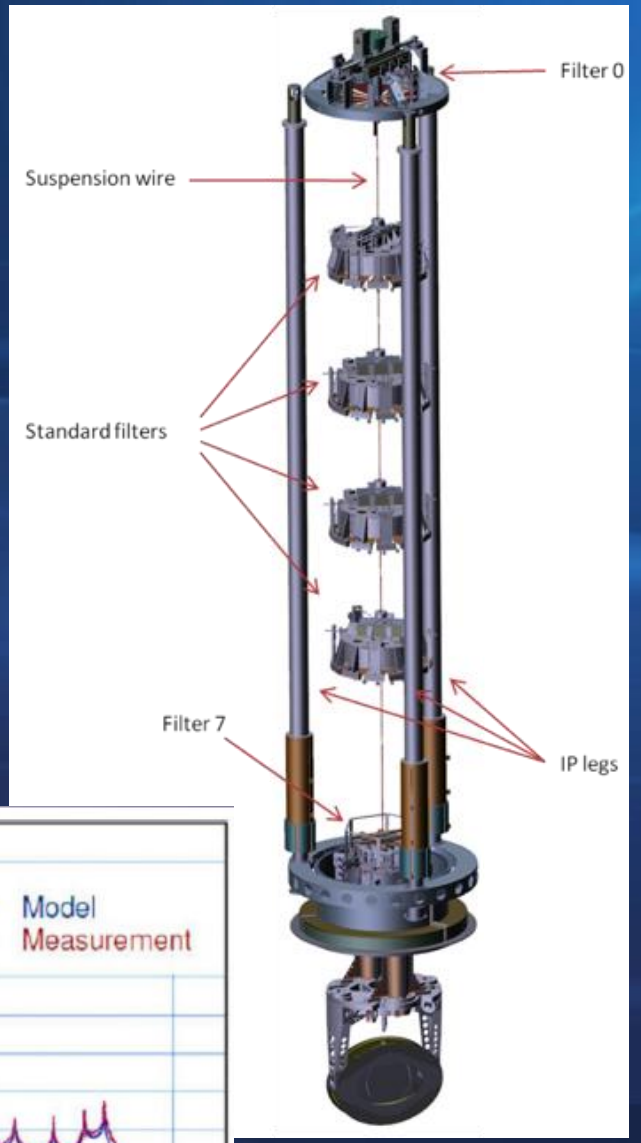
LIGO Hanford & Livingstone data: first direct detection

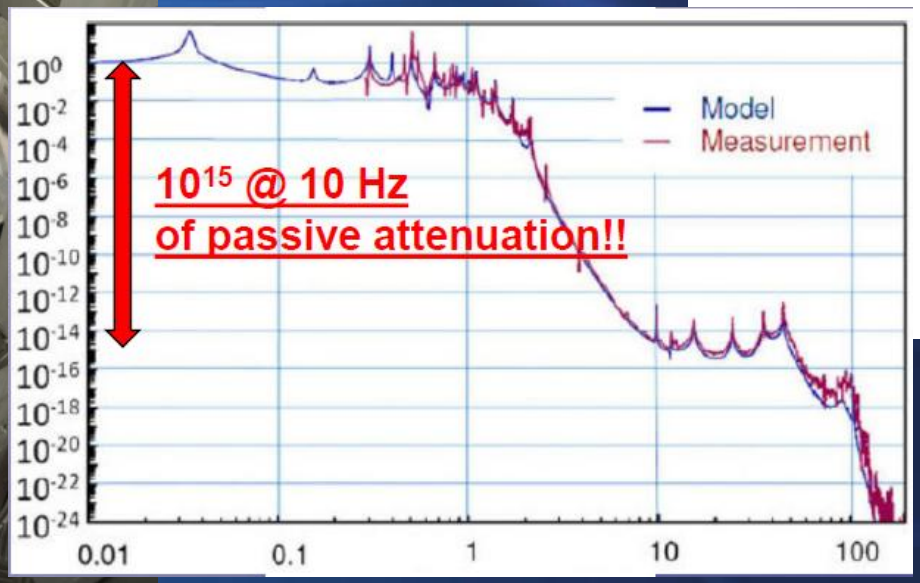
Adv VIRGO Design Sensitivity



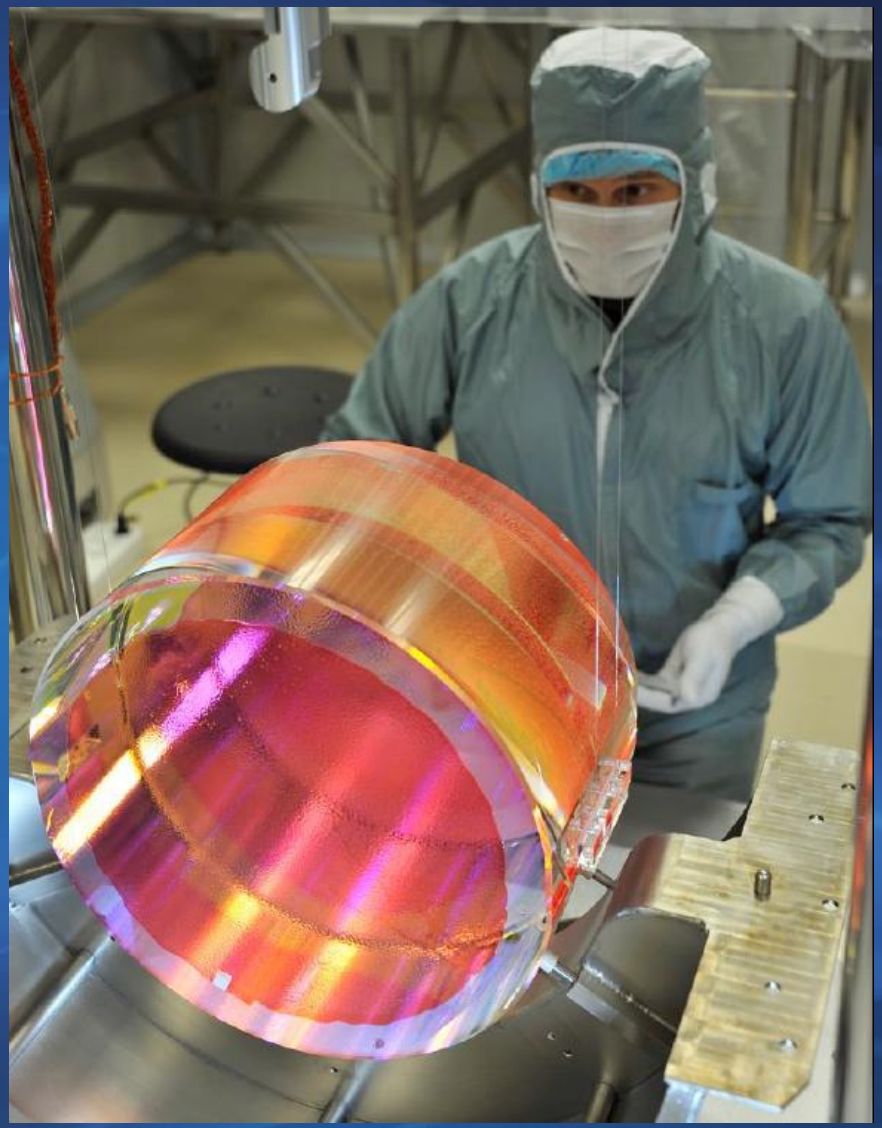
Seismic Isolation

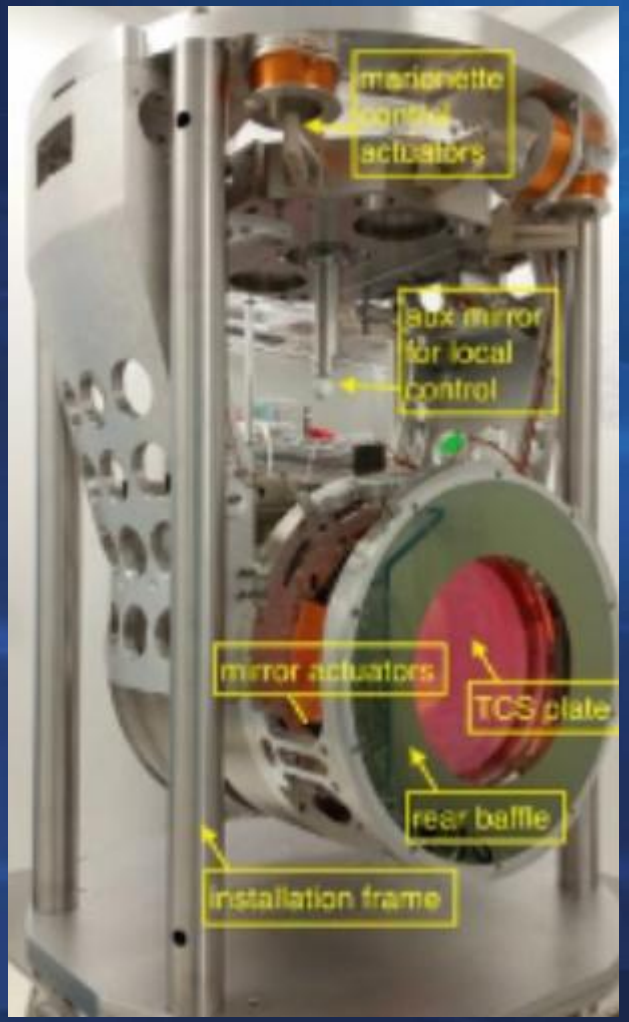




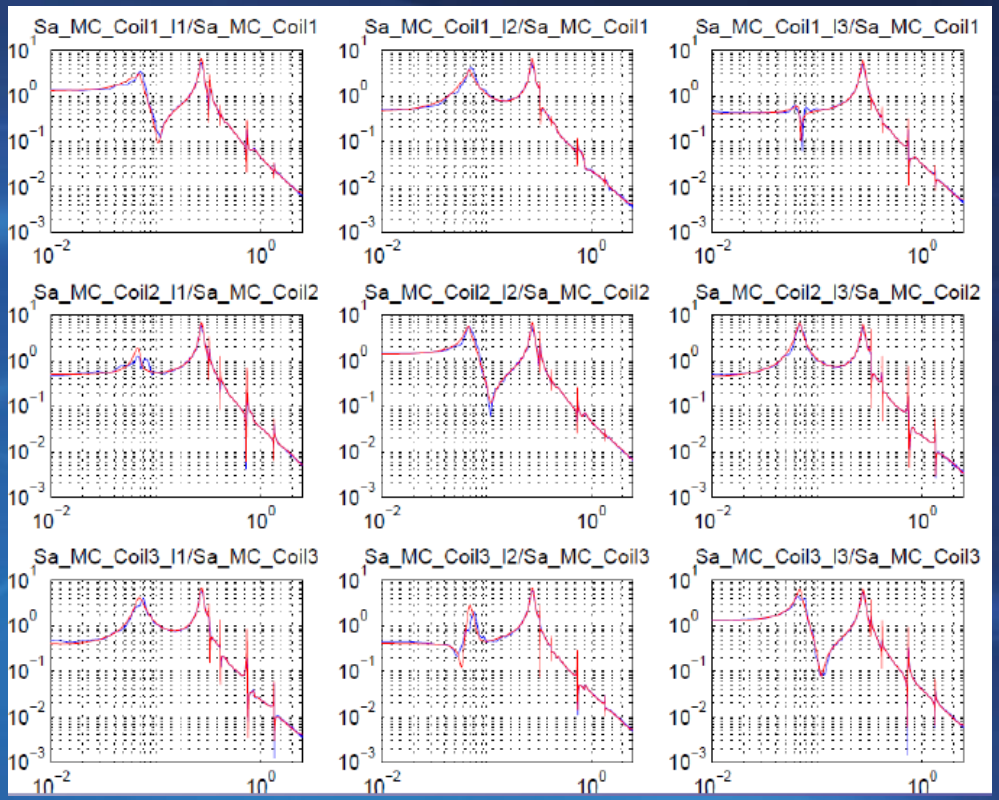


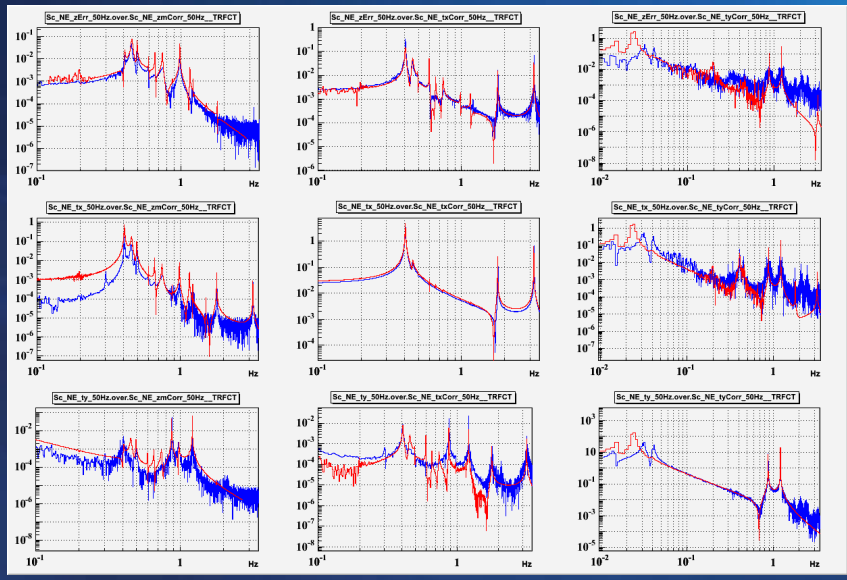
Mirrors

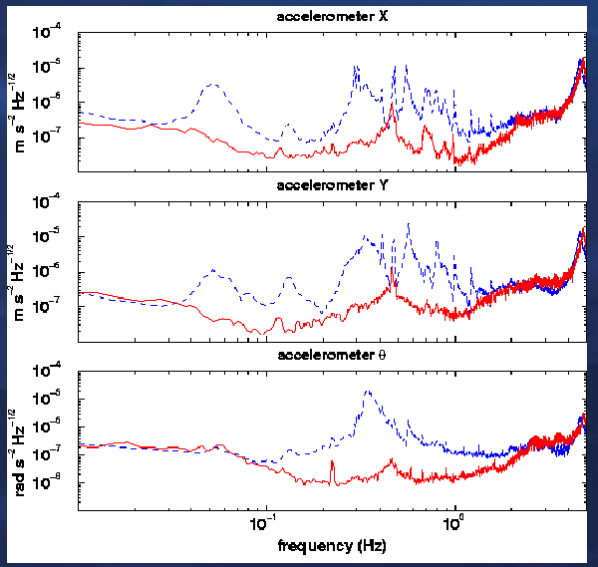



marionette
actuators
aux mirror
for local
control
mirror actuators
TCS plate
rear baffle
installation frame

Control







- Each of 10 seismic isolation systems is equipped with inertial sensors (accelerometers), displacement sensors (LVDT and optical levers), stepping motors and magnet-coil actuators

New Control System: Design Drivers

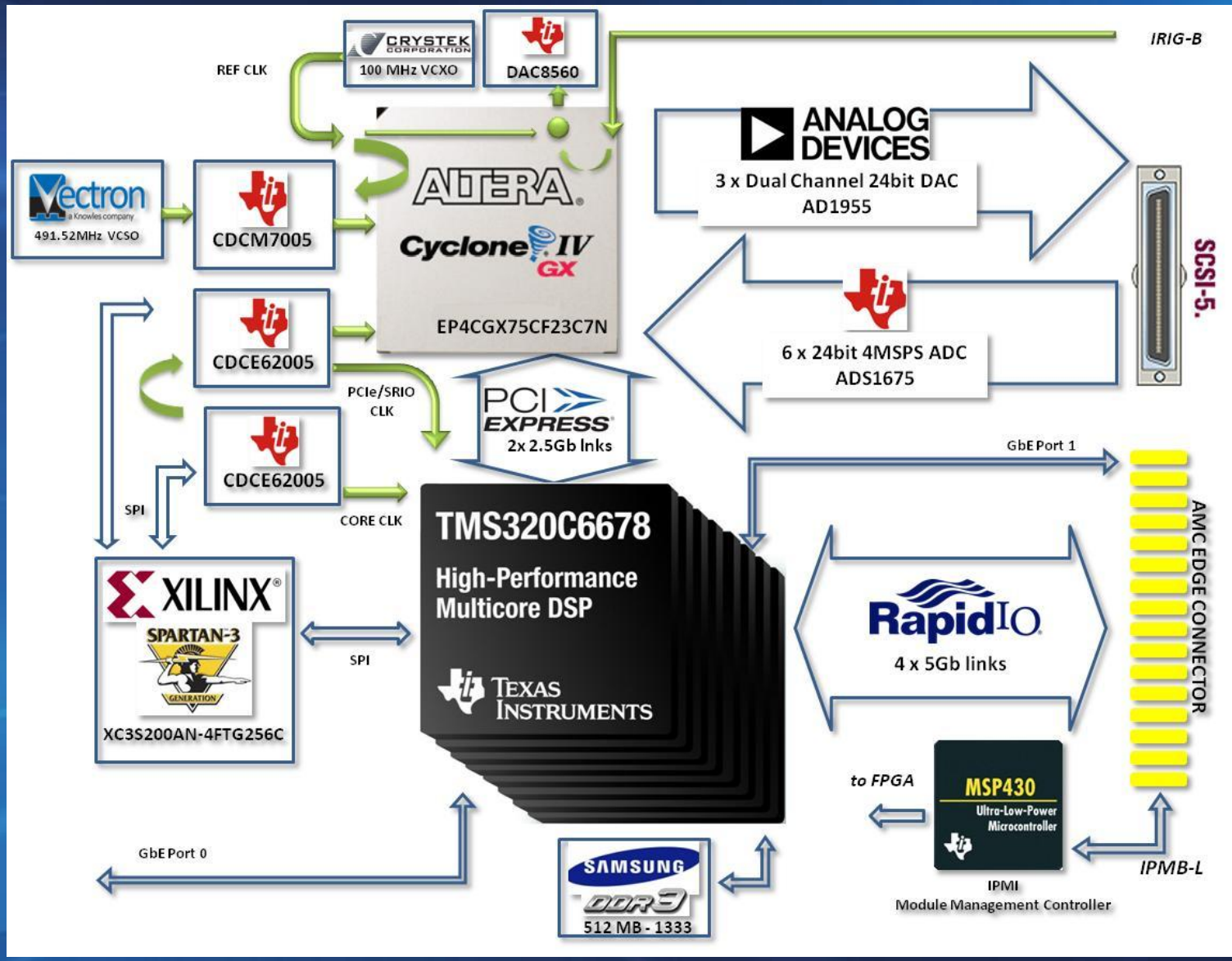
- Move out of VME standard (limiting board-to-board communication bandwidth)
- Eliminate physical gap between analog-front end and data converters (in average we had more than 40 meters STP cabling, multiplied by about 100 signals for each of the 10 suspensions)
- Get out of rigid single sampling frequency operation (usually 10 kHz) and move on a more performing and flexible multi-rate system
- Further increase of locally available computation power to increase data quality analysis capabilities and properly assist control design and implementation mainly for the last stage of suspension and new payloads

UDSPT Board

- Result of design effort is a high performance signal conditioning, signal conversion and processing platform that enables users to implement state of the art hard real time control system. Board can be operated in standalone mode or in cooperation with other boards.
- Form factor is a variation of a Double Compact Module PICMG® AMC.0 R2.0 AdvancedMC module (variation consists in a wider board than what specified for a Double Module)
- Key features of the UDSPT board include:
  - Texas Instruments' multi-core DSP – TMS320C6678
  - 512 Mbytes of DDR3-1333 Memory
  - Two Gigabit Ethernet ports supporting 10/100/1000 Mbps data-rate
  - 170 pin B+ style AMC Interface containing SRIO and Gigabit Ethernet
  - IRIG-B input
  - Module Management Controller (MMC) for Intelligent Platform Management Interface (IPMI)
  - Powered by DC power-brick adaptor (12V/3.0A) or AMC Carrier backplane
  - Two on board FPGA: one Xilinx Spartan3 dedicated to processing unit and one Altera Cyclone IV interfacing data converters
  - 6 x 24b 3.84 MHz ADC converters
  - 3 x 24b 320 kHz DAC stereo converters (6 channels individually addressable)
  - Converters sampling frequency DSP IRQs synchronous with IRIG-B signal
  - Fully differential input channels and balanced output channels

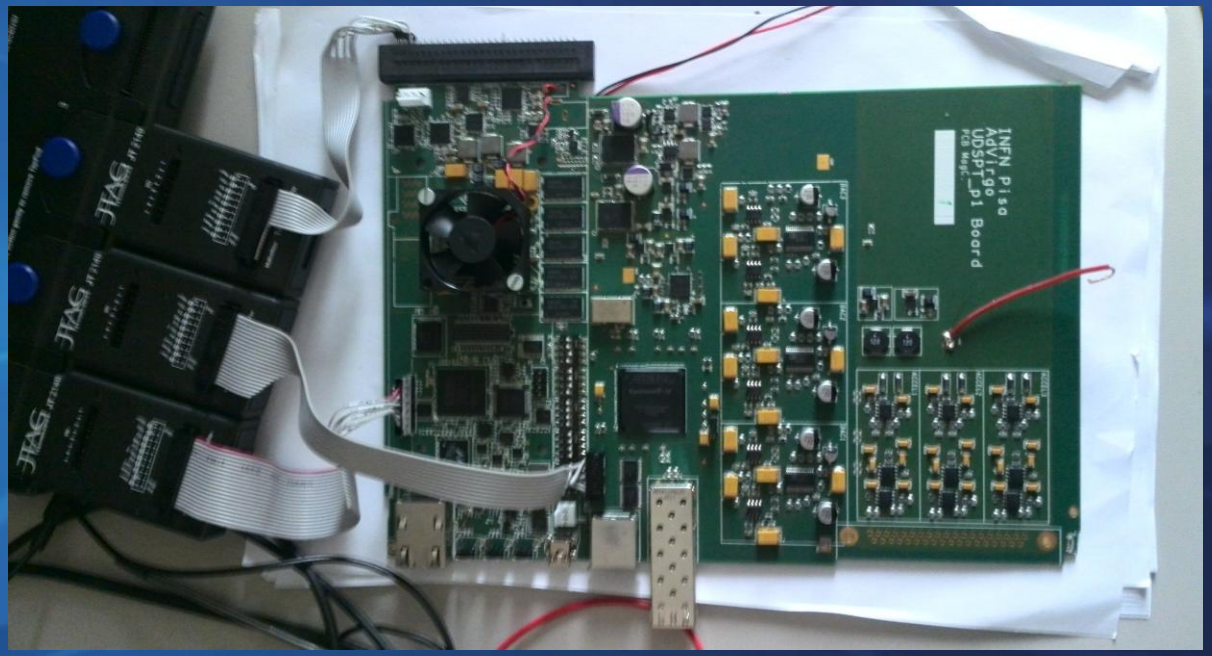
UDSPT Board


VIRGO
INFN
PISA
OPTICAL LINK
TIM
CONSOLE
RESET
LAN1


REF CLK
CRYSTEK CORPORATION
100 MHz VCXO
DAC8560
IRIG-B
Vectron
a Knowles company
491.52MHz VCSO
CDCM7005
ALTERA
Cyclone IV GX
EP4CGX75CF23C7N
ANALOG DEVICES
3 x Dual Channel 24bit DAC
AD1955
SCSI-5.
6 x 24bit 4MSPS ADC
ADS1675
CDCE62005
PCIe/SRIO CLK
PCI EXPRESS
2x 2.5Gb lnks
GbE Port 1
CDCE62005
SPI
CORE CLK
TMS320C6678
High-Performance Multicore DSP
TEXAS INSTRUMENTS
XILINX
SPARTAN-3
GENERATION
XC3S200AN-4FTG256C
SPI
RapidIO
4 x 5Gb links
AMC EDGE CONNECTOR
to FPGA
MSP430
Ultra-Low-Power Microcontroller
IPMB-L
IPMI
Module Management Controller
GbE Port 0
SAMSUNG
DDR3
512 MB - 1333

UDSPT Board

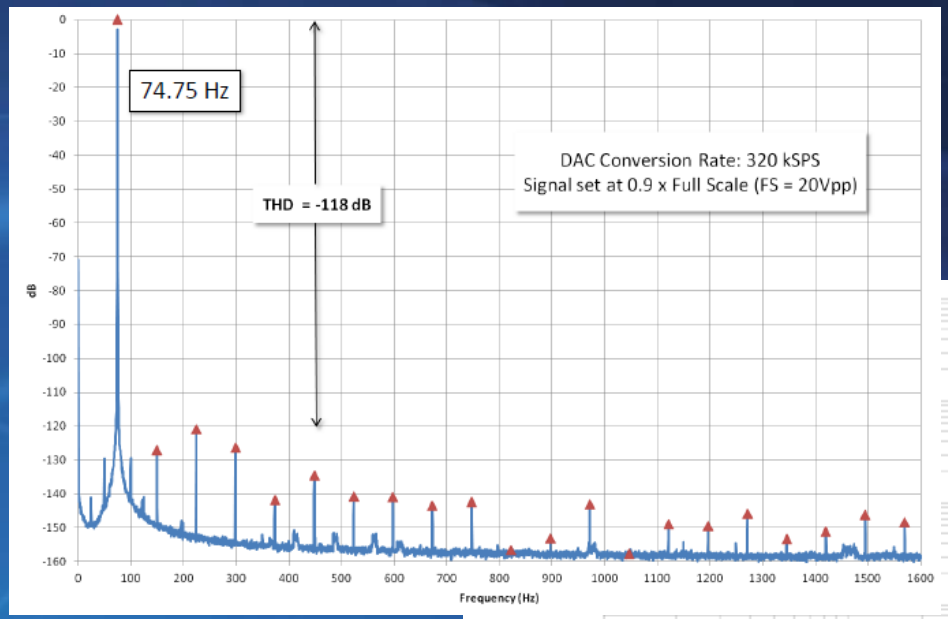
- Board complexity (250 different parts, a total of 2500 components) was compensated by a modular design. Processing, data converters and front-end occupy pre-defined areas. Strong effort in interfaces design

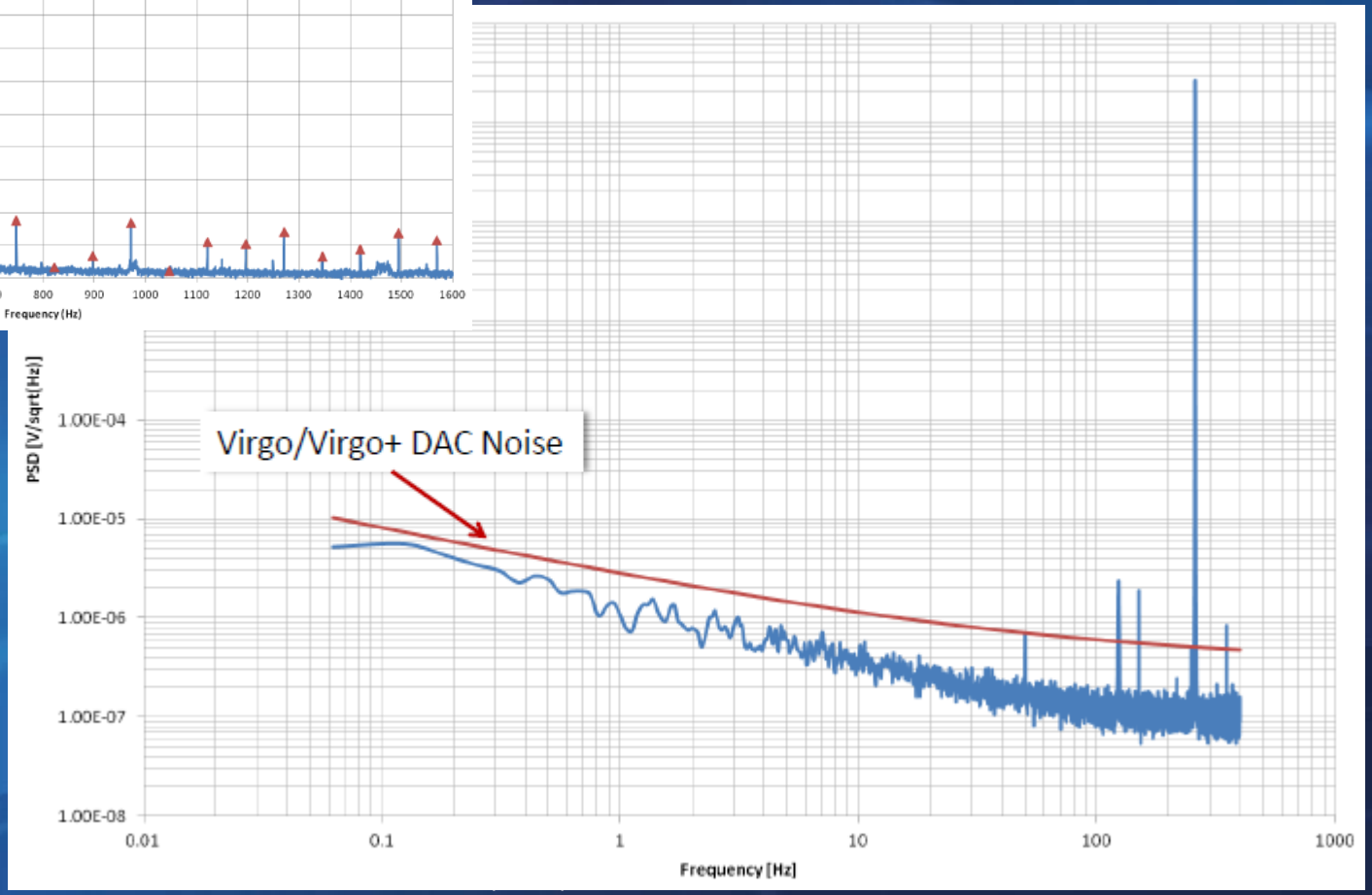


Performances

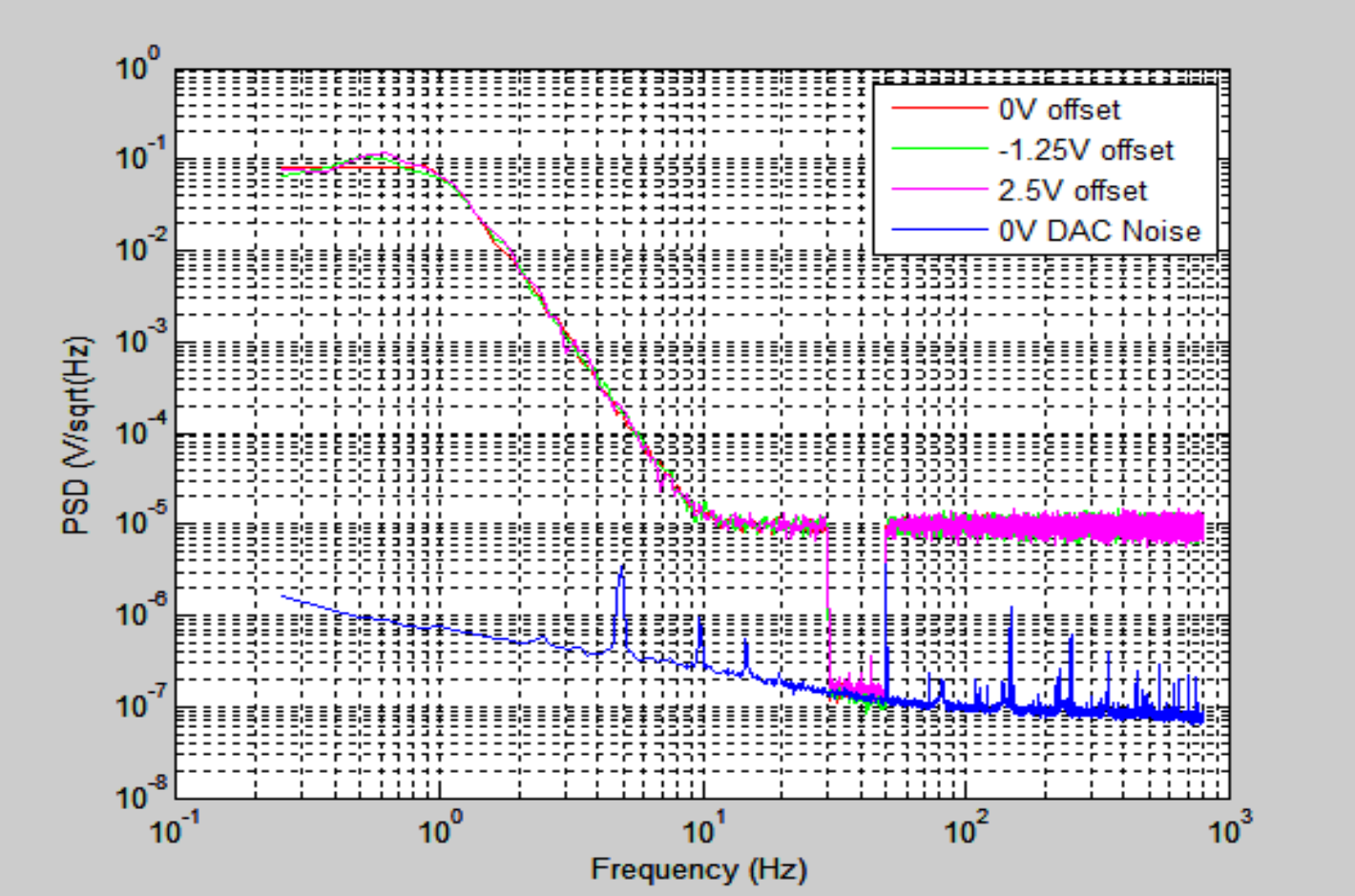
- Digital I/O
  - Gb Ethernet Board- Rest of The World communication
  - PCIe (2 x 2.5 Gbps) On-Board communication
  - SRIO (4 x 5 Gbps) Board-to-board communication
- DSP Software
  - Operation up to 320 kHz sampling rate (3.125 usec interrupt request repetition cycle)
  - Matrix(n,m)*Vector(m,1) double precision multiplication requires 0.5*nm+n+m nanoseconds:
    - State space with 3 inputs, 3 outputs and 12 states requires less than 200 nsec

DAC Perfomances



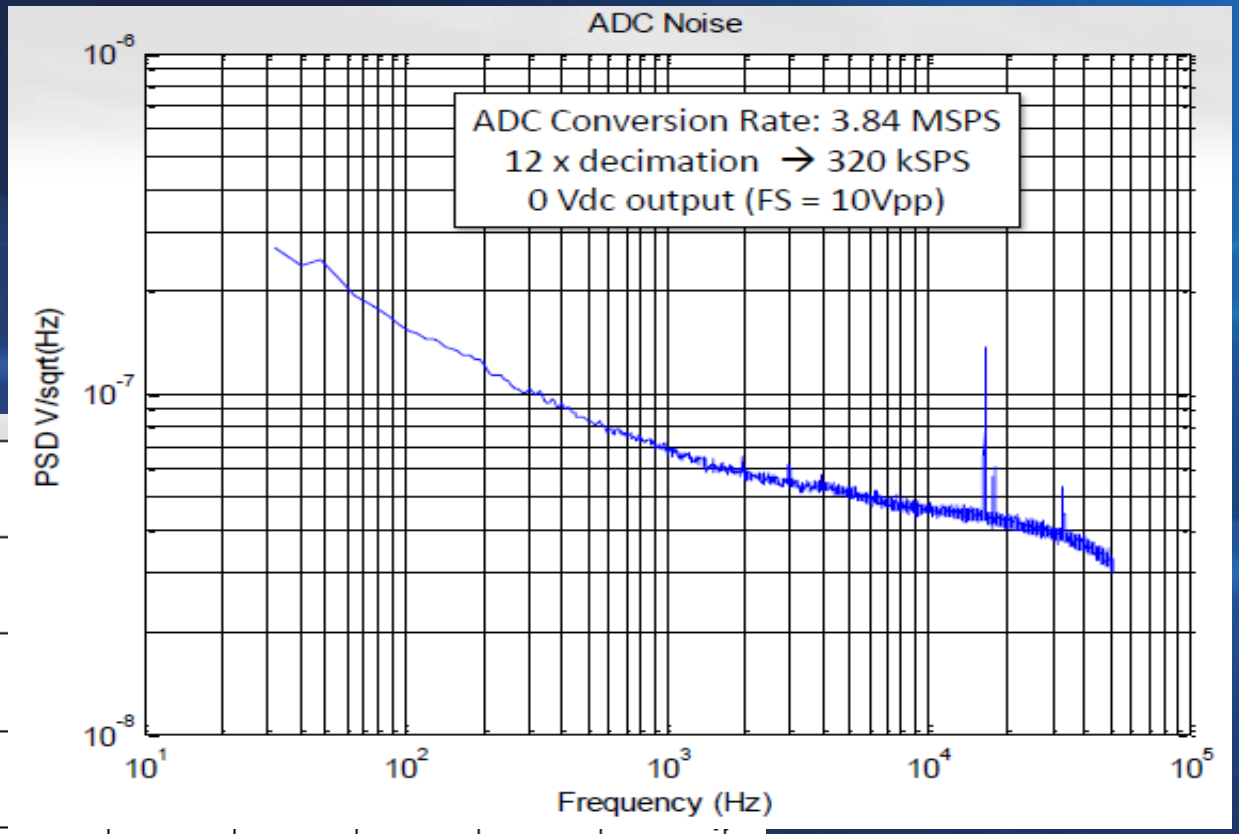


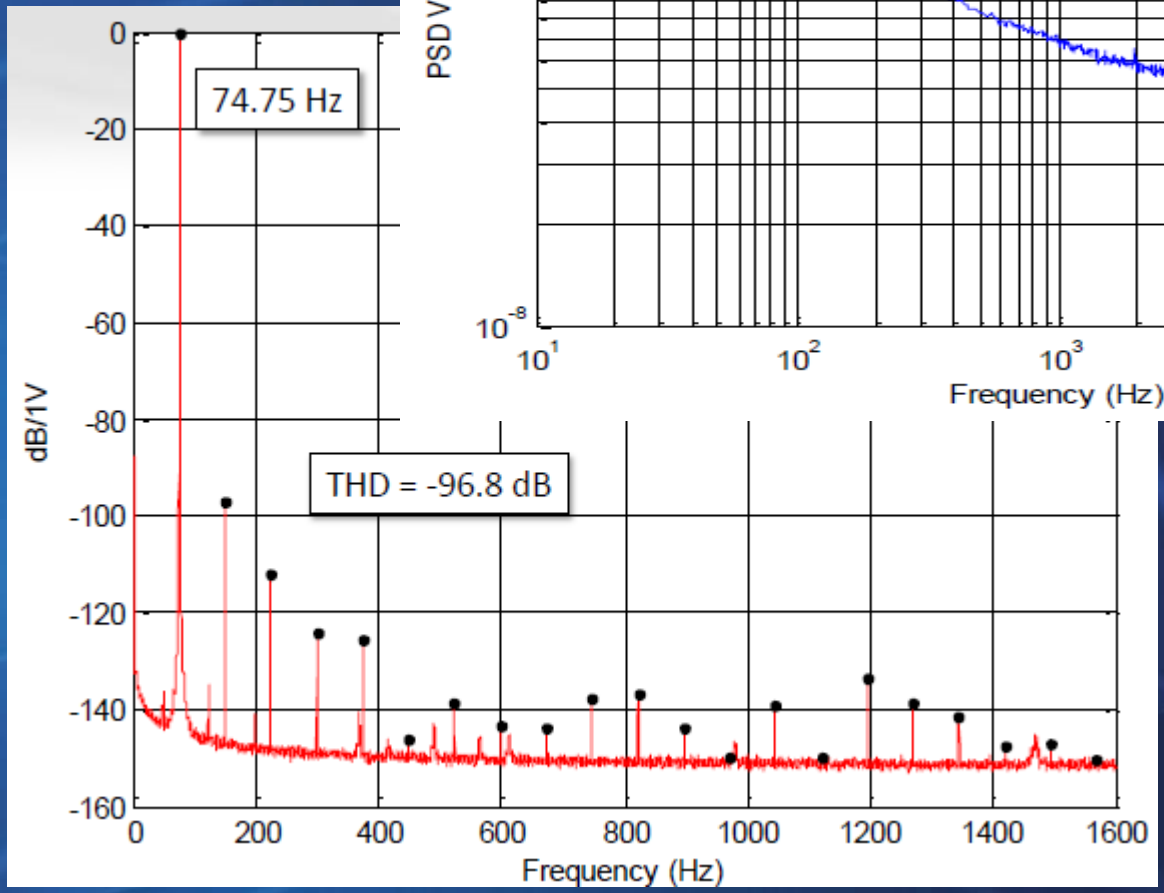
DAC Perfomances



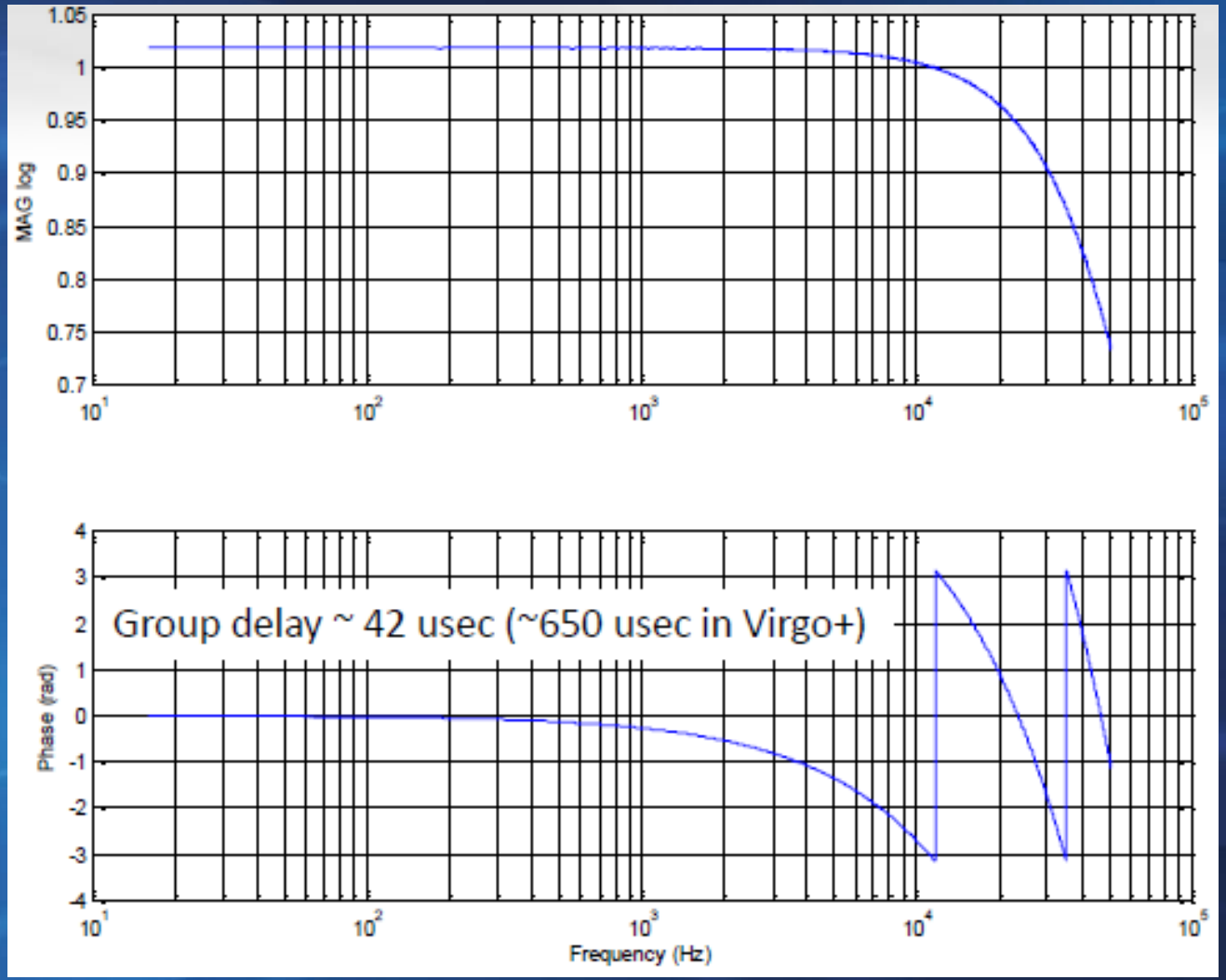
- DAC output spectrum when converting a white spectrum signal low pass filtered (4 poles @ 1Hz + 4 zeroes @ 10 Hz) and stopband (30-50 Hz, Bessel 12th order,0.1dB ripple, -60 dB stopband). Signal amplitude was set to 0.6Vpp

ADC

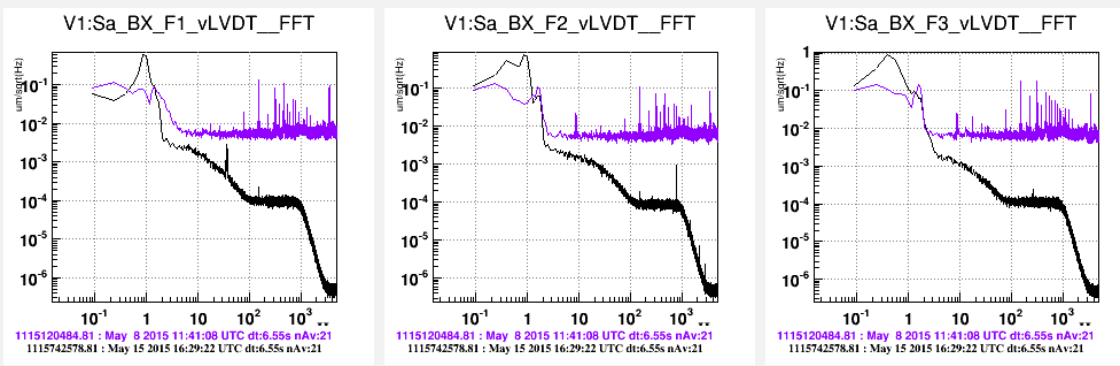

ADC Noise
ADC Conversion Rate: 3.84 MSPS
12 x decimation → 320 kSPS
0 Vdc output (FS = 10Vpp)
PSD V/sqrt(Hz)
Frequency (Hz)


74.75 Hz
THD = -96.8 dB
dB/1V
Frequency (Hz)

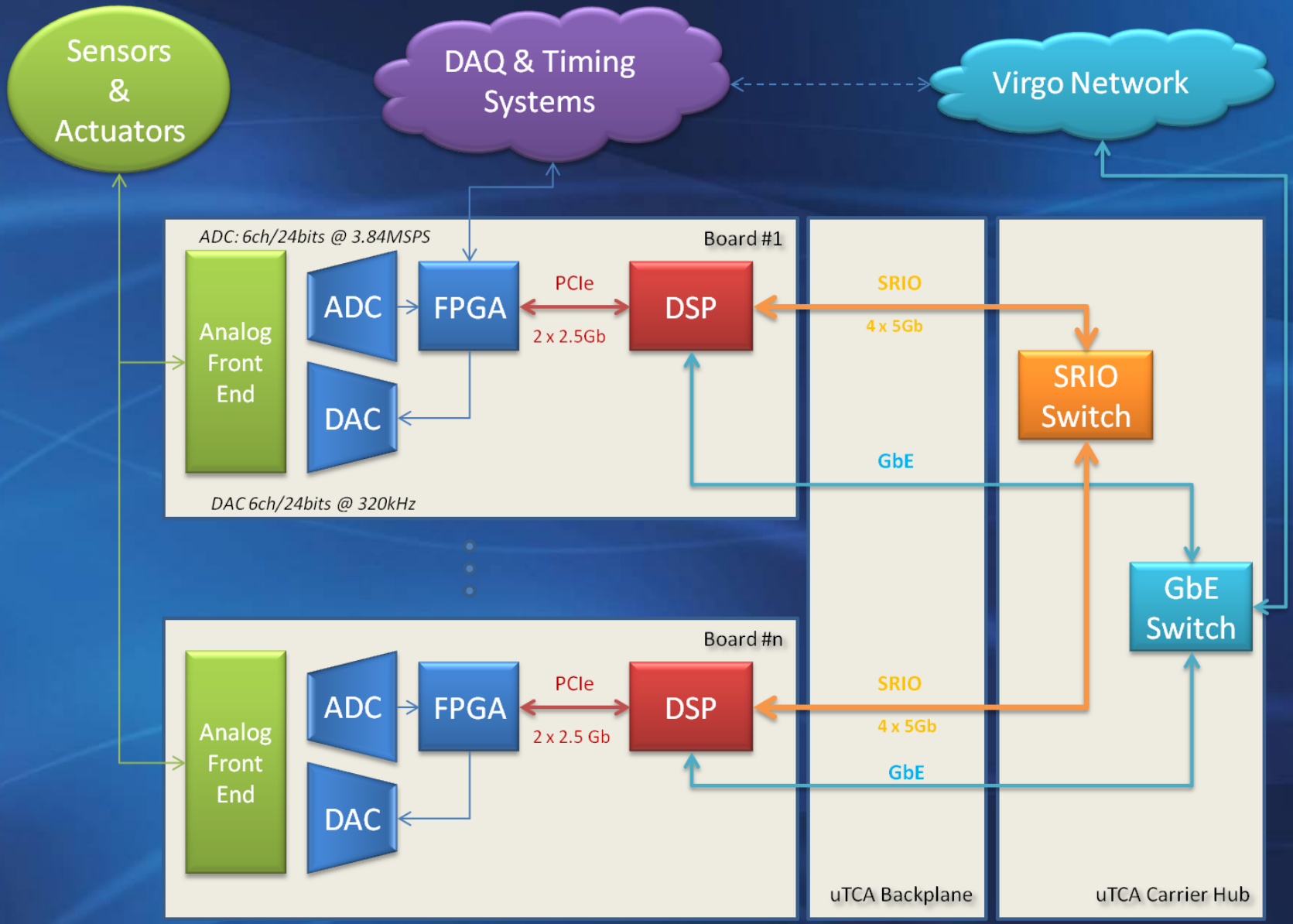
DAC-DSP-ADS Transfer Function



Performances: ADC Noise & EMI



- Performances tests on suspensions went beyond our expectation. Placing converters at front-end electronics level, together with introduction of digital demodulation, drastically improved noise immunity. Following picture shows improvement in vertical position sensors readout.


Sensors & Actuators
DAQ & Timing Systems
Virgo Network
ADC: 6ch/24bits @ 3.84MSPS
Board #1
Analog Front End
ADC
DAC
FPGA
PCIe
2 x 2.5Gb
DSP
SRIO
4 x 5Gb
SRIO Switch
GbE
DAC 6ch/24bits @ 320kHz
GbE Switch
Board #n
Analog Front End
ADC
DAC
FPGA
PCIe
2 x 2.5 Gb
DSP
SRIO
4 x 5Gb
GbE
uTCA Backplane
uTCA Carrier Hub

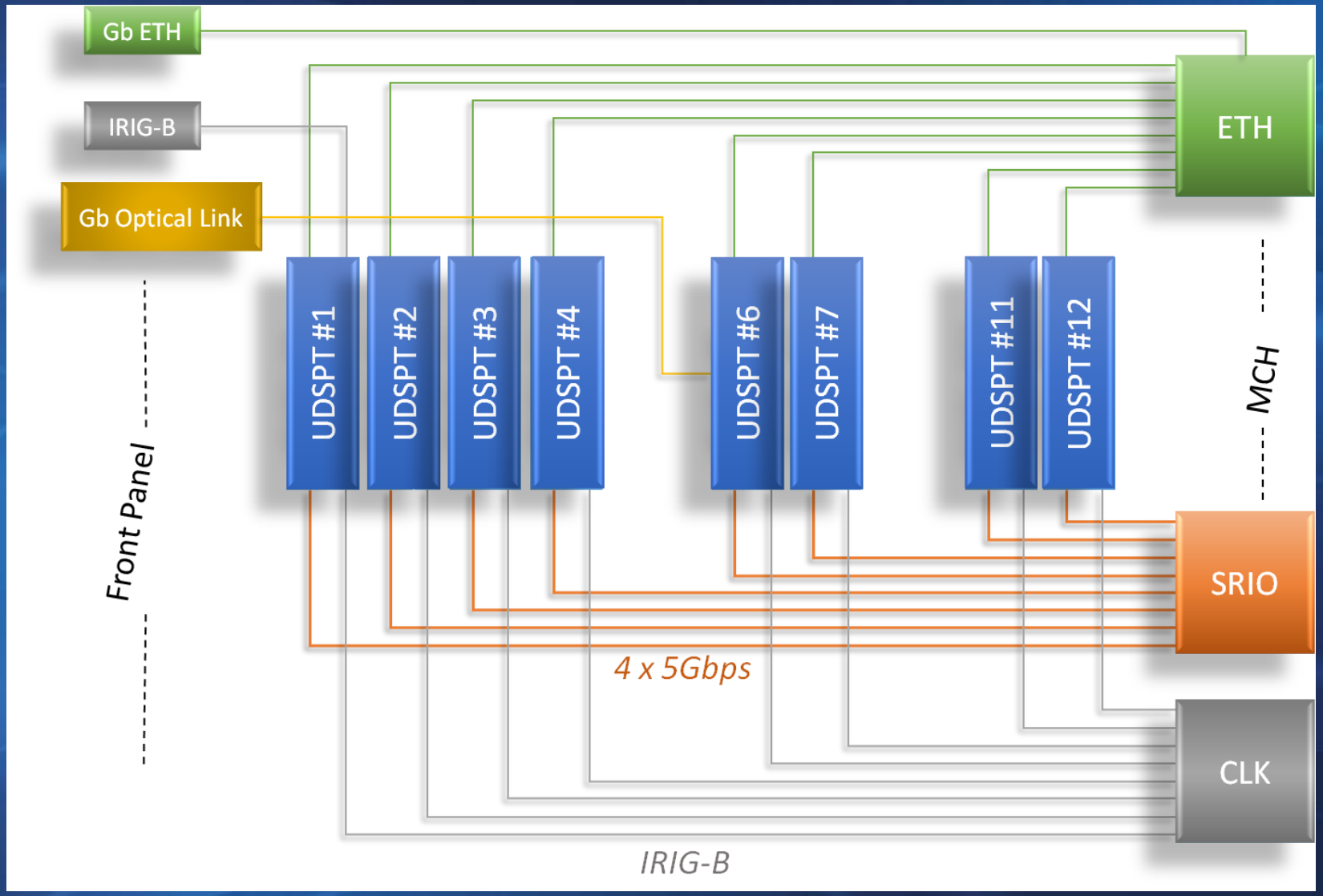
Local Control Unit

- In a single 6U x 19” crate we can concentrate 72 ADC + 72 DAC channels supported by 720 GFLOPs, 12 x 1 Gb Ethernet ports + 12 x 1 Gb Optical Link for digital IO and a total power consumption less than 500 Watts

Deployment

- Recently we completed the installation of 20 Local Contro Units for a total of 150 boards, 900 ADC channels and 900 DAC channels

Digital Signals Path



Software Supervisor

- For initial AdV, software shall support about 150 boards. At system startup each board loads boot code via tfpt from a dedicated boot server. This code is actually a sort of basic OS. In a second phase software supervisor (based on software toolkit Tango Controls) downloads in each processor the hard real-time code.

SAT Supervisor (su cmpl3.virgo.infn.it)

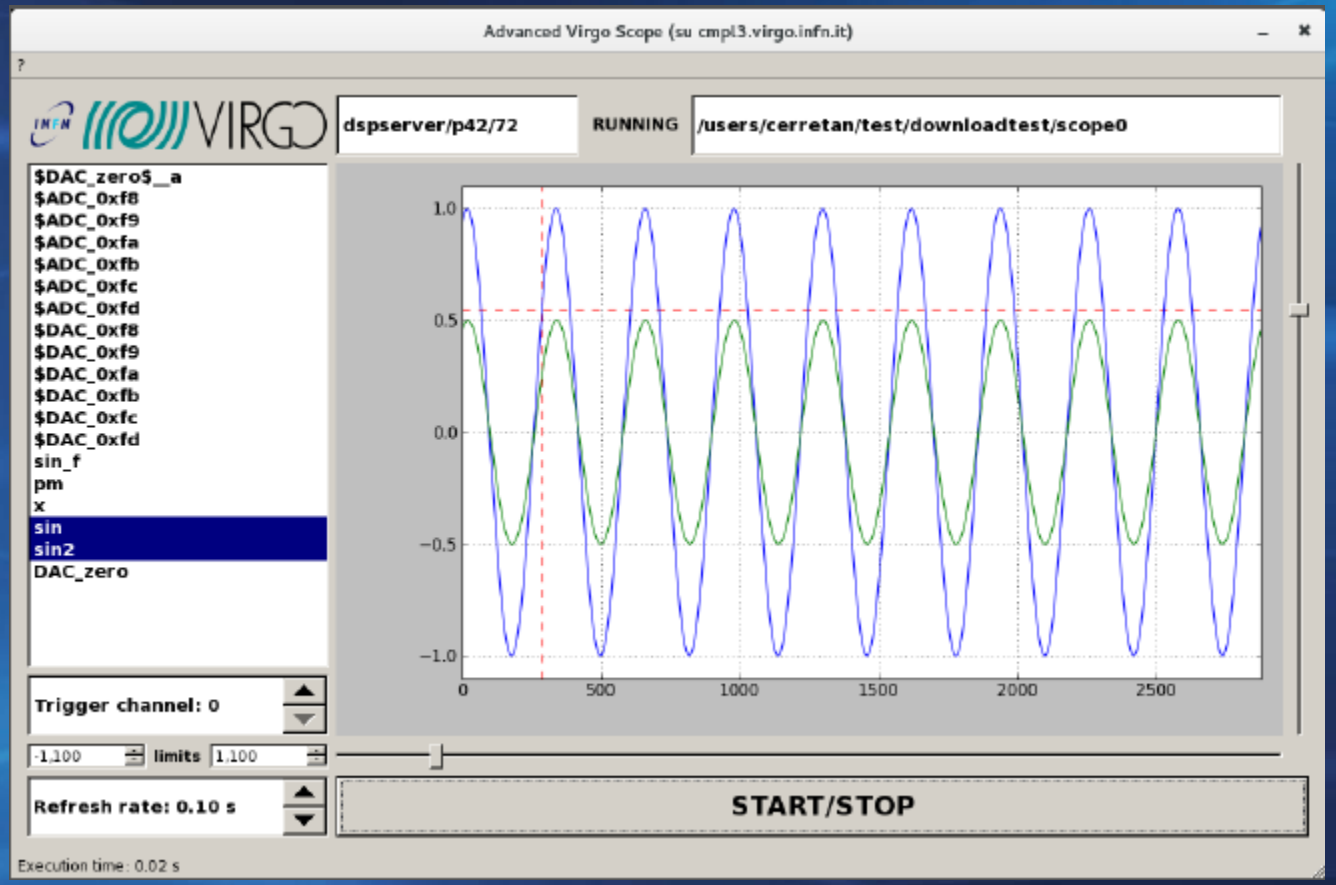
File

Sa_BS | Sc_BS | EE | Sa_IB | Sc_IB | Sa_GA | Sc_GA | Lab | Sa_MC | Sc_MC | Sa_NE | Sc_NE | Sa_NI | Sc_NI | Sa_OB | Sa_PR | Sc_PR | Sa_SR | Sc_SR | Sa_WE | Sc_WE | Sa_WI | Sc_WI

1	dspserver/p42/80	RUNNING	/virgoDev/Sa/DSPCode_Adv/BS/LC/BS_PSDi	OFF	ON	Supervisor	Scope	Properties
2				OFF	ON	Supervisor	Scope	Properties
3	dspserver/p42/108	RUNNING	/virgoDev/Sa/DSPCode_Adv/BS/LC/BS_PSDm	OFF	ON	Supervisor	Scope	Properties
4	dspserver/p42/110	RUNNING	/virgoDev/Sa/DSPCode_Adv/BS/LC/BS_PSDt	OFF	ON	Supervisor	Scope	Properties
5	dspserver/p42/84	RUNNING	/virgoDev/Sa/DSPCode_Adv/BS/LC/BS_PSDf	OFF	ON	Supervisor	Scope	Properties
6	dspserver/p42/119	RUNNING	/virgoDev/Sa/DSPCode_Adv/BS/LC/BS_Mir1	OFF	ON	Supervisor	Scope	Properties
7	dspserver/p42/115	RUNNING	/virgoDev/Sa/DSPCode_Adv/BS/LC/BS_Mir2	OFF	ON	Supervisor	Scope	Properties
8	dspserver/p42/72	RUNNING	/virgoDev/Sa/DSPCode_Adv/BS/LC/BS_Mar1	OFF	ON	Supervisor	Scope	Properties
9	dspserver/p42/85	RUNNING	/virgoDev/Sa/DSPCode_Adv/BS/LC/BS_Mar2	OFF	ON	Supervisor	Scope	Properties
10	dspserver/p42/120	RUNNING	/virgoDev/Sa/DSPCode_Adv/BS/F7/BS_F7_CD	OFF	ON	Supervisor	Scope	Properties
11	dspserver/p43/139	RUNNING	/virgoDev/Sa/DSPCode_Adv/BS/LVDT/BS_F7_LVDT	OFF	ON	Supervisor	Scope	Properties
12				OFF	ON	Supervisor	Scope	Properties
MCH	mchserver/bs/2						Reboot chassis	Properties
PDU	pduserver/bs/1	#8					OFF ON	Properties

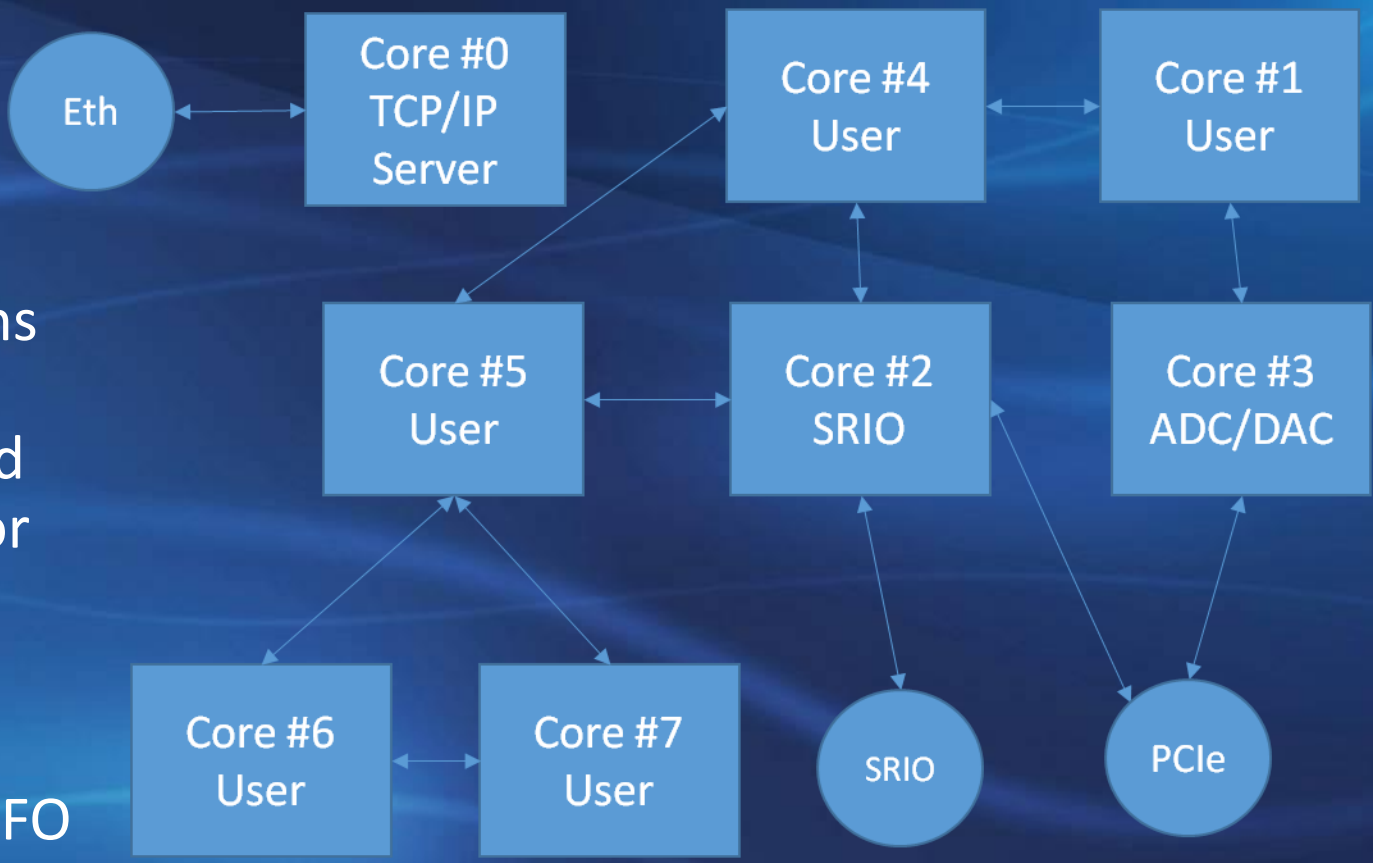
Real Time Scope

- A nice tool for system debugging is a real-time scope that can access any single variable declared into the source code. Work is in progress to provide real-time fft and few more functionalities



DSP Cores

- Each DSP core, excluding C#0, runs a single task activated at a fixed frequency lower or equal to 320 kHz
- Communication is memory mapped using a 2 stages FIFO (ping-pong table)
- 5 out of the 8 cores available are user programmable



DSP Code

```
#include <asm/header.net>
NETLIST {

  START {
    SAMPLING_FREQ = 10000;
    OVER_SAMPLING = 1;
    USE_ADC       = 1;
    DOWN_SAMPLING = 1;
    W_COEFF = internal;
    SOURCE = parent;
    DOL_ADDR = f94c0000;
    GRD_ADDR = Guard4_to_Core0;
    DICT_NAME = Sc_NI;
    KEEP_VALUES = no;
    DECAY_TIME  = 0x00000000;

    TOLM_OUT_N = 1;
    TOLM_OUT = {
      {
        PAGE = 0x00000000;
        NAME = PRB;
        ROUTING = 0xf94c0000;
        TOLM = TOLM1;
      }
    }
  }

  ASM_START {}

  DAC_SETUP {
    NDAC = 6;
    OUT = ($DAC_0xf8, $DAC_0xf9, $DAC_0xfa, $DAC_0xfb,
           $DAC_0xfc, $DAC_0xfd);
  }

  ASM_CONTINUE {}

  CALL {
    NAME   = NI_F7_LVDT_Demod.hrd;
    COMM   = " ";
    NINP   = 0;
    NOUT   = 9;
    OUT    = (lvdt1, lvdt2, lvdt3, vlvdt1, vlvdt2, vlvdt3, ratio_l1, ratio_l2,
              ratio_l3);
  }
```

```
    .tab 6
DAQ_header1   .equ  00055045h
DAQ_header2   .equ  00055046h
DMA7_chain    .equ  000517e8h
tolm_in_dpm_buffer  .equ  14831a00h
tolm_in_ser_buffer  .equ  14831c00h
tolm_in_glb_buffer  .equ  14831800h
tolm1_out_buffer    .equ  00058300h
tolm2_out_buffer    .equ  00058700h
Guard1_to_Core0 .equ  1087f000h
Guard4_to_Core0 .equ  1483c400h
        .sect       .textasm         ; program section

SP      .set        b15

        .global dsp_irq
        .global i_list
        .global c_list
        .global w_list
        .global noise_table
        .global cosine_table
        .global input_list
        .global output_list
        .global Guard4_to_Core0
dsp_irq:
        .asmfunc

        mvc     CSR, B2                     ; save old interrupt status
||      add     -8,SP,SP
        DINT                                ; disable interrrupts
        stdw        A11:A10, *SP--[1]       ; a10 = call_in
        stdw        A13:A12, *SP--[1]       ; a12 = tolm_out_address
        stdw        A15:A14, *SP--[1]
        stdw        B11:B10, *SP--[1]       ; b10 = call_out
        stdw        B13:B12, *SP--[1]       ; b12 = tolm_in_address
        stdw        B9:B8, *SP--[1]         ; b8 = From_Slot buffer address
        stdw        B3:B2, *SP--[1]
        stw         A8, *SP--[1]            ; save To_Slot buffer address
        stw         B6, *SP--[1]            ; save signals buffer address
        stw         A6, *SP--[1]            ; save probe buffer address
        stw         B4, *SP--[1]            ; save DAC address
```

- From a source code written in a simple object-oriented language we generate asm code for the cross-compiler that produces DSP binary code

Conclusions

- A new system to control suspensions and mirrors was developed and installation is now completed.
- Extensive use of supervising software, mysql databases and third parties tools will simplify operation and maintenance
- Simple user interfaces allow writing real-time DSP code without having specific training