Real Controls with a Simulated Plant at the 40m

LIGO Caltech

Joe Betzwieser
Rana Adhikari, Koji Arai, Jameson Rollins

LIGO - Caltech / April 26, 2011

LIGO Document G1100524-v3

Why Simulate?

- To help **design:**
  - Combine many simple laws to produce complexity
  - Make sure we don't miss something due to that complexity
  - Make sure a design is plausible

- To help **understand:**
  - Difference between reality and simulation lies new understanding
  - Toy models can help suggest methods of control

Previous Simulations of Interferometers

- E2E: End to End
  - Full Time domain simulation of the fields
  - Simulates optics, mechanics, servos, etc
  - Written in C++
  - Outputs text files with fields and other info

- Optickle and its derivatives:
  - Frequency domain simulation of the fields
  - Written in MATLAB
  - Outputs plots and text data

- GWINC:
  - Static simulation of noise spectra
  - Written in MATLAB
  - Outputs plots and text data

Other Simulations of Interferometers

- Siesta (time dimain, c, modal model, Virgo-Annecy/France)

- Twiddle(frequency domain, Mathematica, scalar field, LIGO)

- Finesse(frequency domain, c, modal model, GEO)

- MIST (frequency domain, matlab, modal model, Virgo-Pisa/Italy)

- SIS (stationary field, C++, FFT, aLIGO)

- MITFFT(static field, fortran, FFT, iLIGO)

- DarkF (static field, fortran, FFT, Virgo-Nice/France)

- Siesta (static field, c, FFT, Virgo-LMA/France) use tools of initial time domain siesta

- OSCAR(static field, matlab, FFT, Virgo-LMA/France)

So why do we need another Simulation?

- Because of output!

- Other simulations can't drive the real aLIGO servos and controls
As aLIGO controls evolve and change, update Simulated Plant just by SVN checkout.

- They don't use the same software tools to look at data as aLIGO

- Previous simulations don't work directly with the CDS

What is CDS?

- CDS stands for "Control and Data Systems"

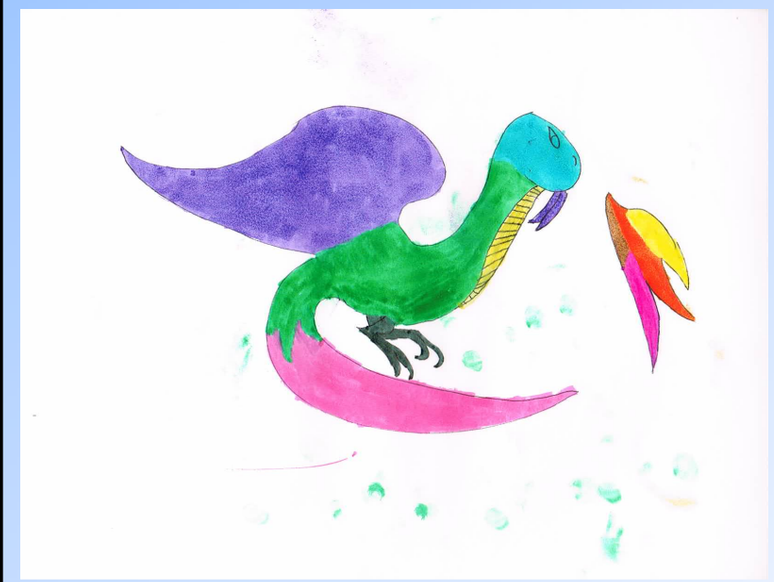
- Its is a real-time digital system

- Interface to the analog interferometer (IFO) systems

- Records data signals

- Contains many sub-systems and control loops

Physical structure of the aLIGO CDS



Analog Realm:
Here be dragons
and IFOs

ADCs

DACs

Digital Realm:
Here be bit noise
and computers

Front End Computer

DAQ network

Frame Builder

EPICS

NDS and
Raw Frames

User's Workstation

"Simulated Plant (SP)"

- An IFO emulator for the digital control system

- A time domain simulator realized by realtime codes for the digital control system itself

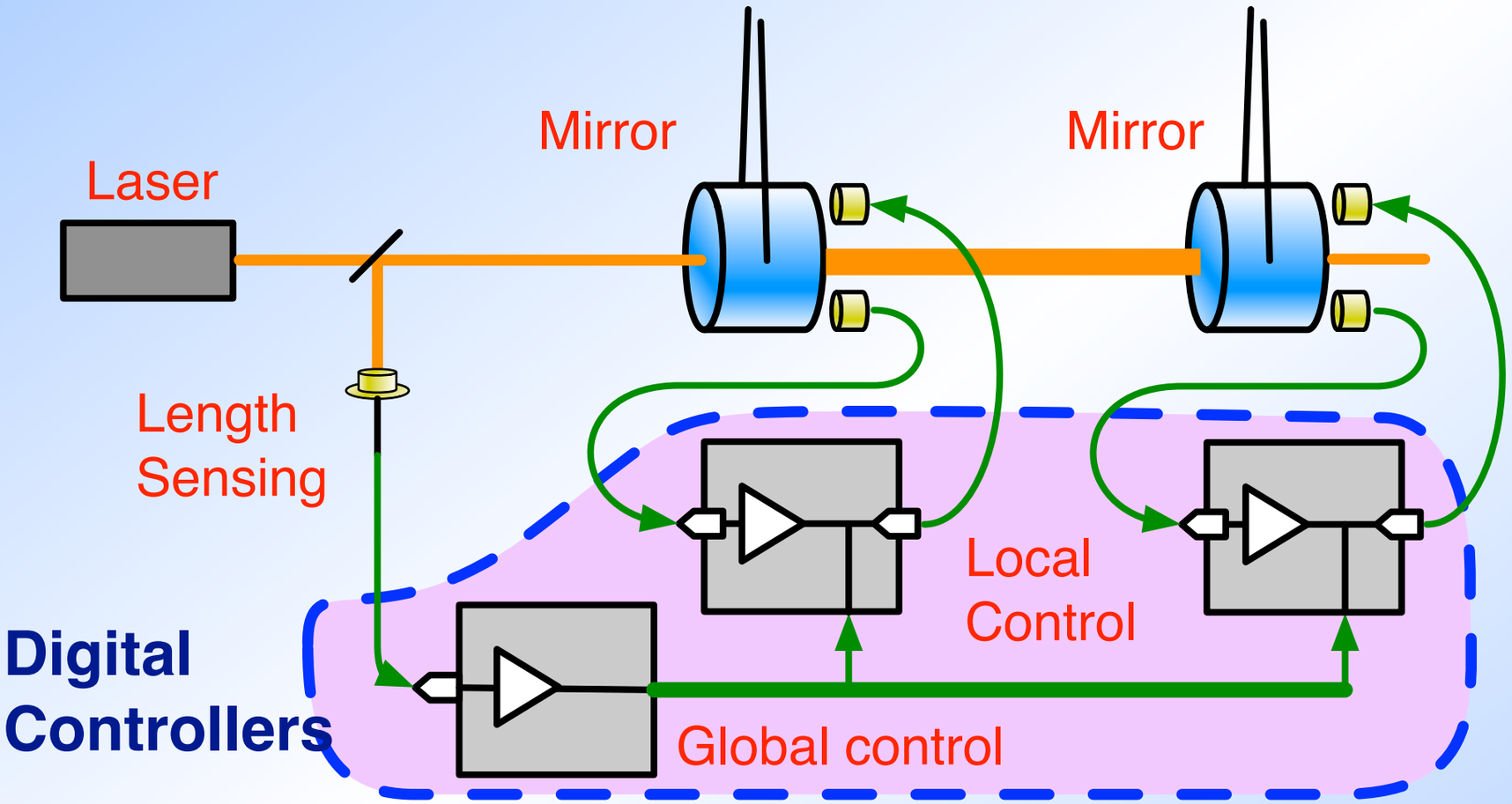
- Imitating responses of interferometer components

Basic Idea

Interferometer control:

Local control (suspension)

+ Global control (interferometric)

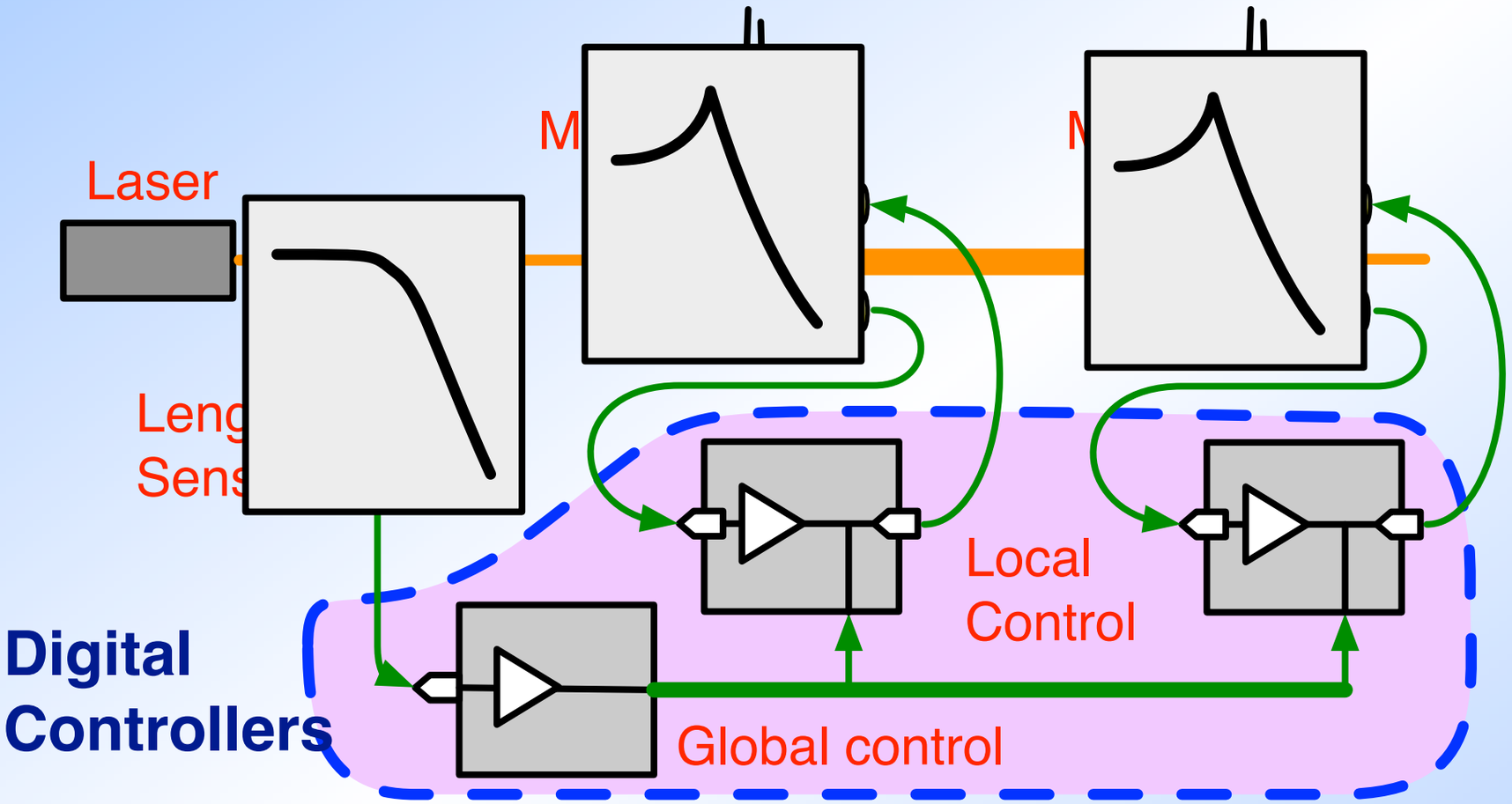


Basic Idea

Interferometer control:

Local control (suspension)

+ Global control (interferometric)

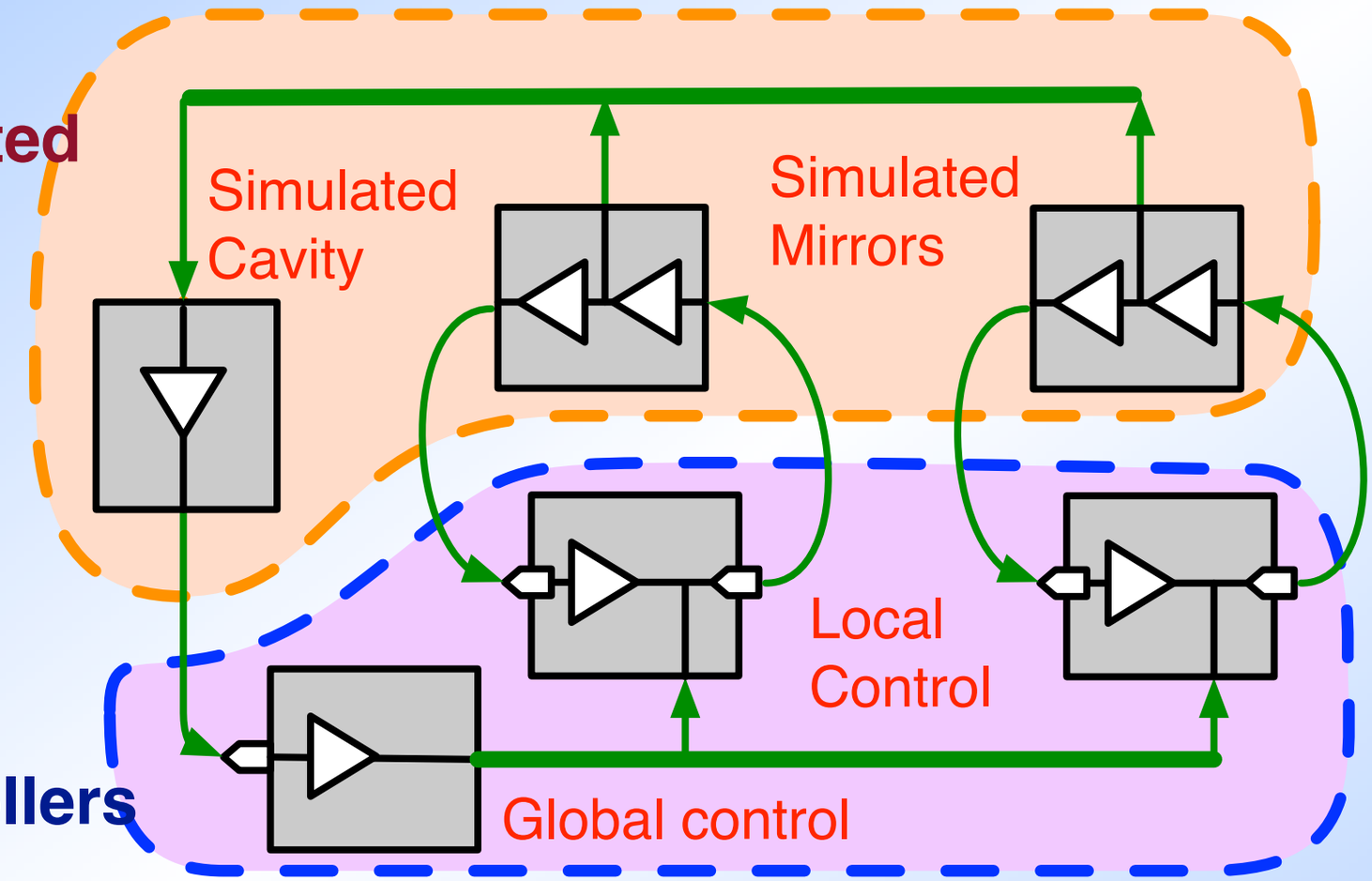


Basic Idea

Replaced hardware responses with digital filters

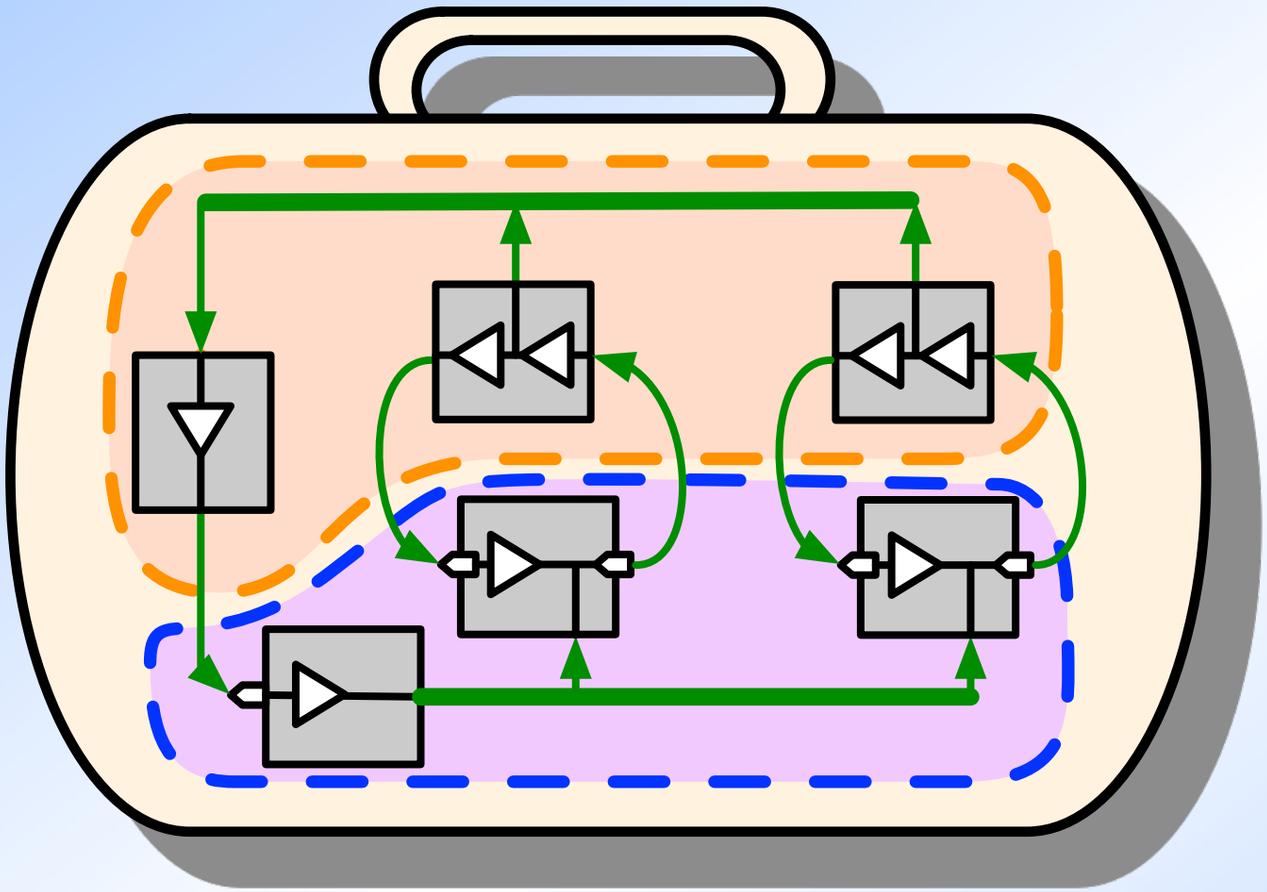
==> simulated plants

The servo loops remains stable



Basic Idea

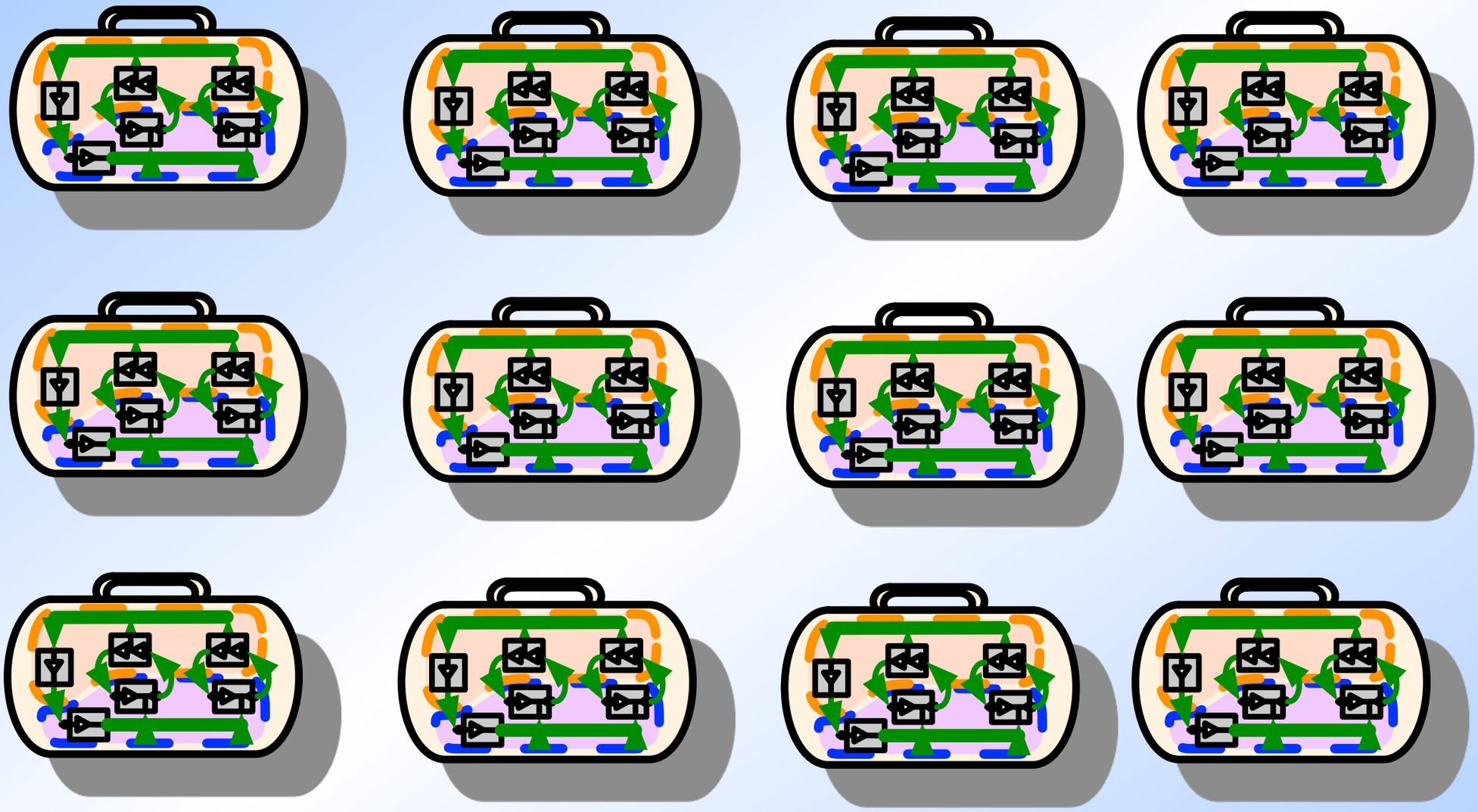
You can put them on a single machine!



Simulated Plant version of the 40m end suspensions

Basic Idea

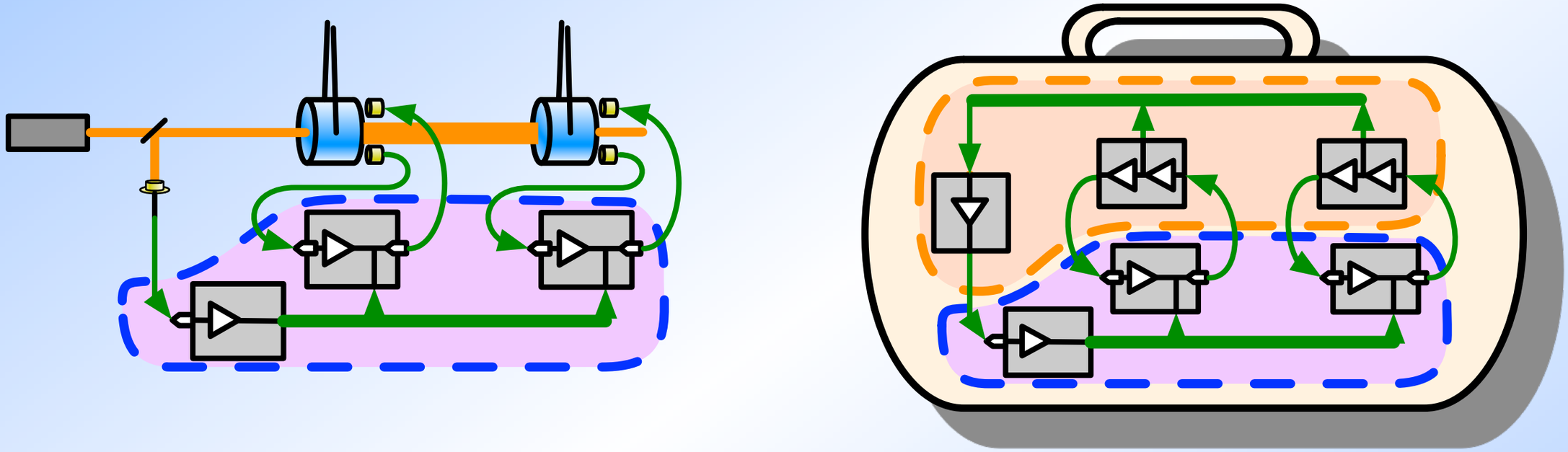
The real CDS is 25 computers and 100 models per IFO.
Full simulation could be done with about 12 computers.

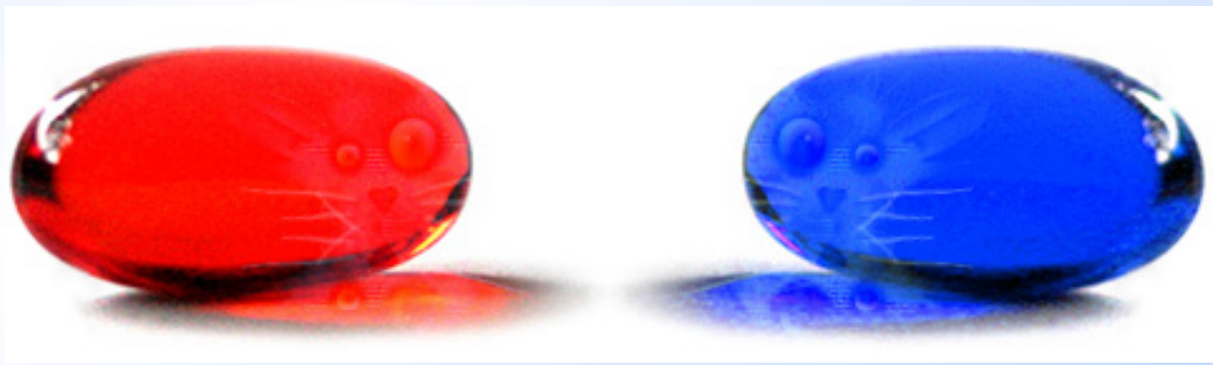


Simulated Plant version of a full LIGO IFO?

Basic Idea

Or you could run the simulation directly on those 25 IFO computers.
You can then switch between real and simulated inputs/outputs.





Benefit: Parallel Work

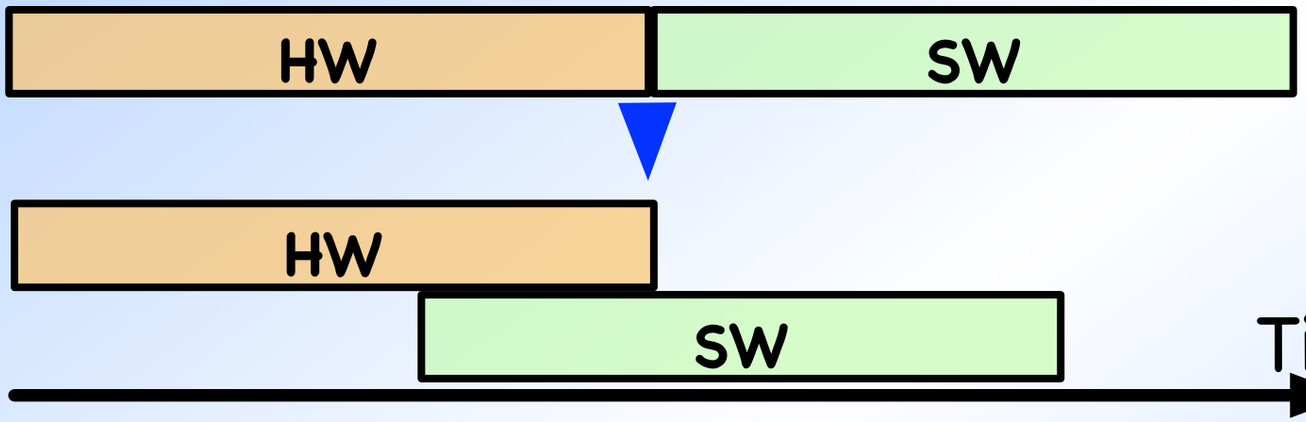
- Commissioning time is valuable

Make the installation and commissioning faster!

- Start building the control SW without actual HWs

While doing single arm test, work on dual recycled IFO code

Realtime codes / Scripts (auto locker / initial alignment)



- Separate controller (computer) problems and SW development from the real HW issues

Even a simple model is sufficient!

Benefit 2: Help understand the noise

- Eventually we face with noise hunting

Make it more convenient

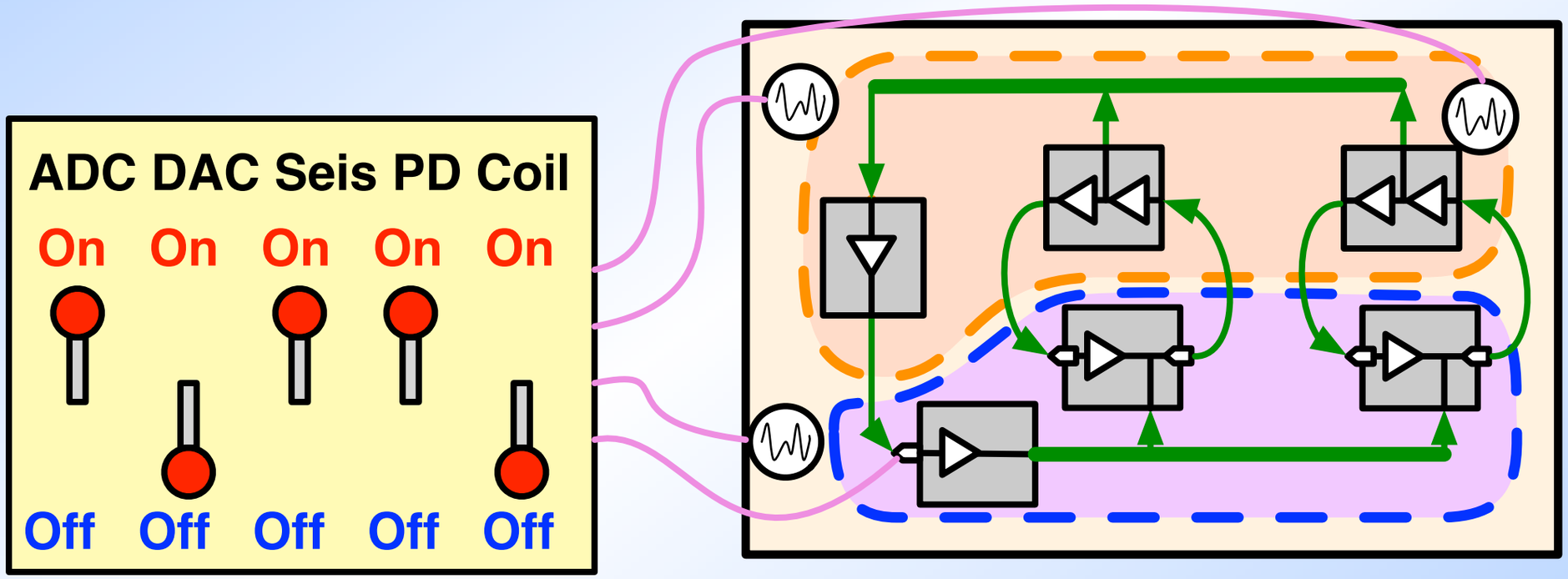
- Build diagnosis SWs offline

Calibration / Measurement templates / Auto noise budget

- Anticipate the noise spectra / saturations

Noise emulation: ADC / DAC / Electronics / Seismic / Laser

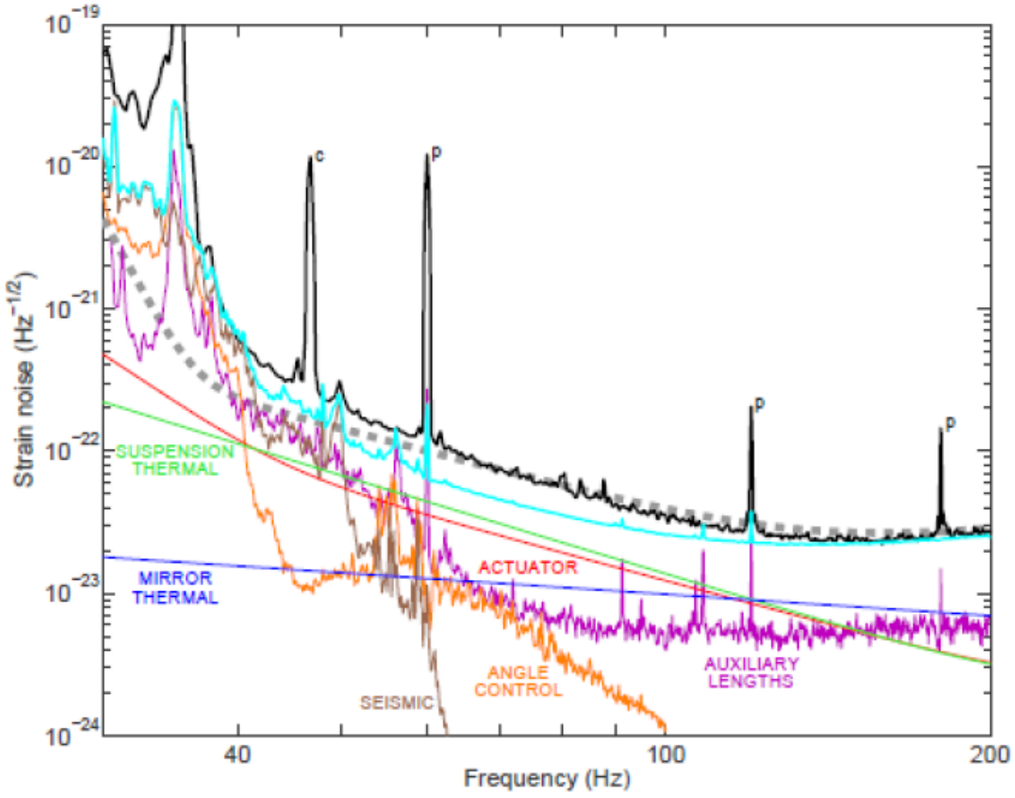
Timing errors / Thermal / Shot / Radiation Pressure



Benefit 2: Help understand the noise

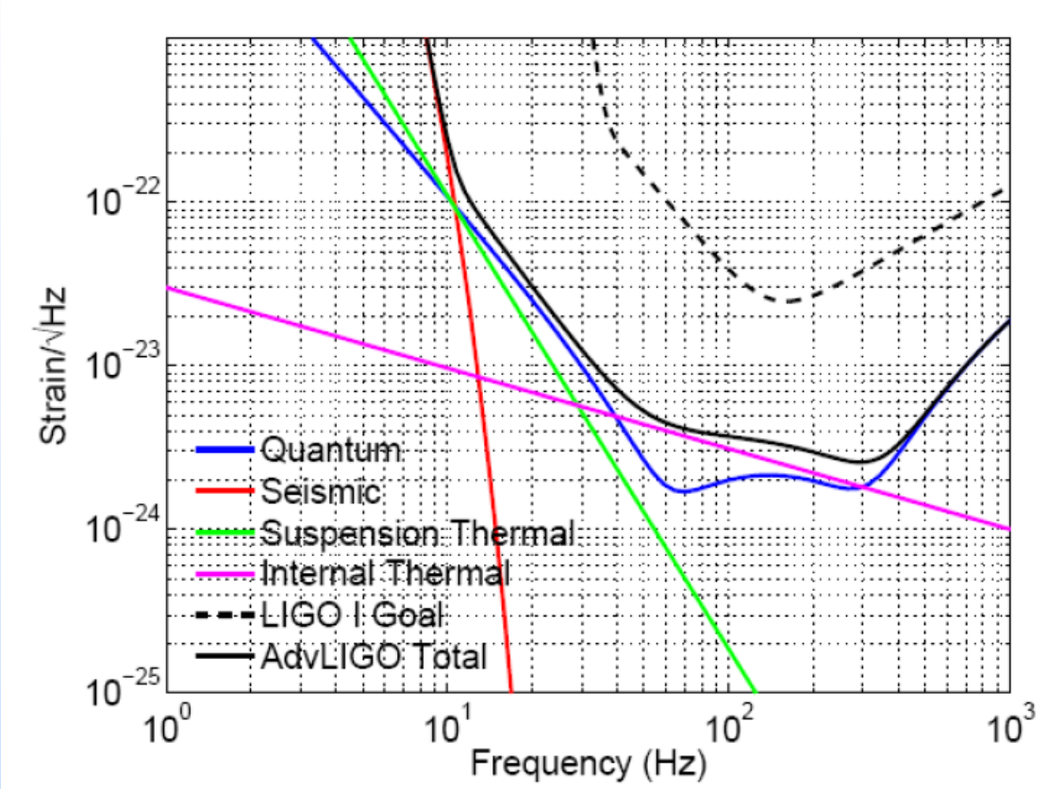
- Initial LIGO noise budgets are extremely complex
  Predicted aLIGO noise: a few lines with GWINC
- Simulated Plant can include:
  ADC saturations, electronics noise, control loops, seismic,
  thermal, shot, radiation pressure, timing errors
- Can produce a time series or spectra directly with DTT

Initial LIGO Noise Budget



?

aLIGO Noise Budget



Benefit 3: Data Analysis

- Does SP only help the commissioning guys?

No! It will be useful to the data analysis people too!

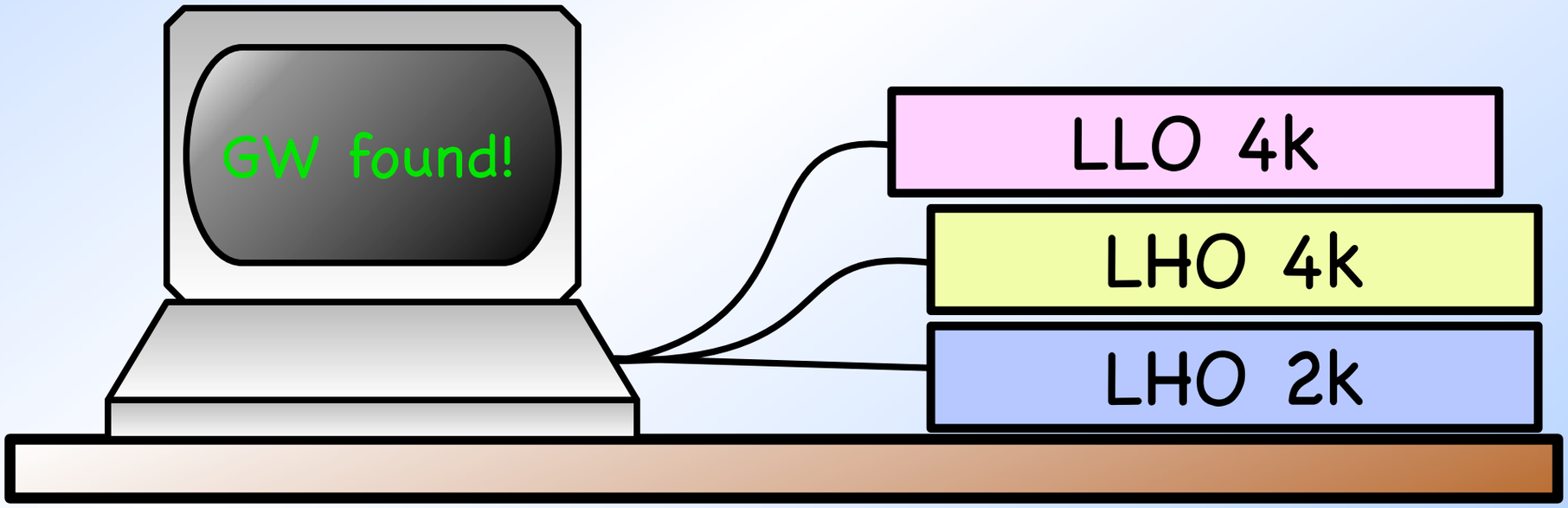
- Imagine a dedicated simulation cluster used for testing:

HW Injection system / DA pipeline / Online monitor

- From Front End to Frame builder to Computering Cluster

All the code will be identical to aLIGO!

IMAGINE your desk...



CDS at the 40m

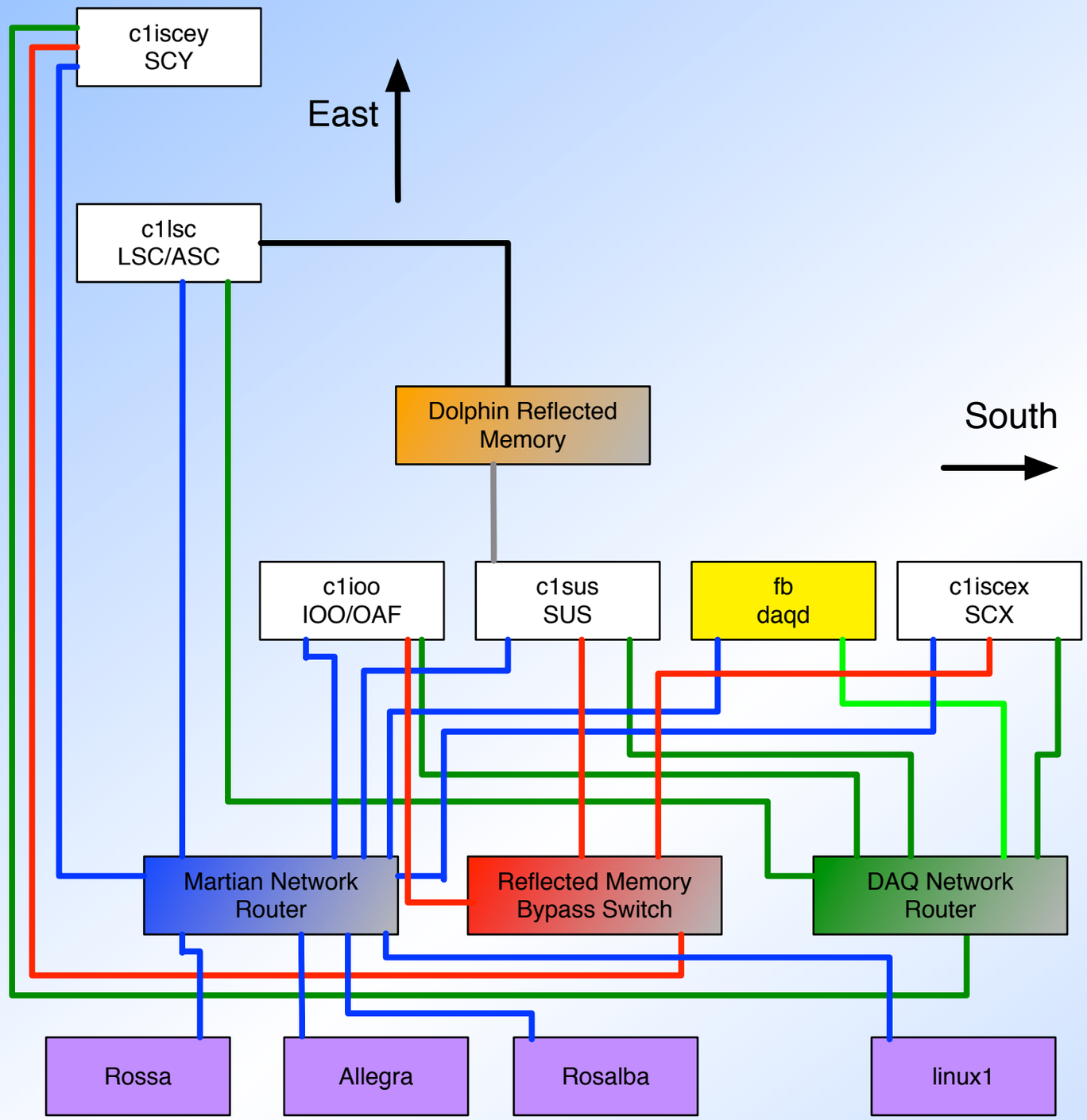
- 40m prototype interferometer upgraded to the aLIGO CDS

CDS still evolving, upgrading

- 5 front end computers running:
  - 16 control models
  - 2 simulated plant models (End suspensions)

- Frame builder collecting data from all models and recording

CDS computers and network at the 40m



Legend

Gentoo Diskless Boot Server

Gentoo Diskless Boot Front End

Control Room Centos Computer

10 Gb/s Fiber

2 Gb/s Copper

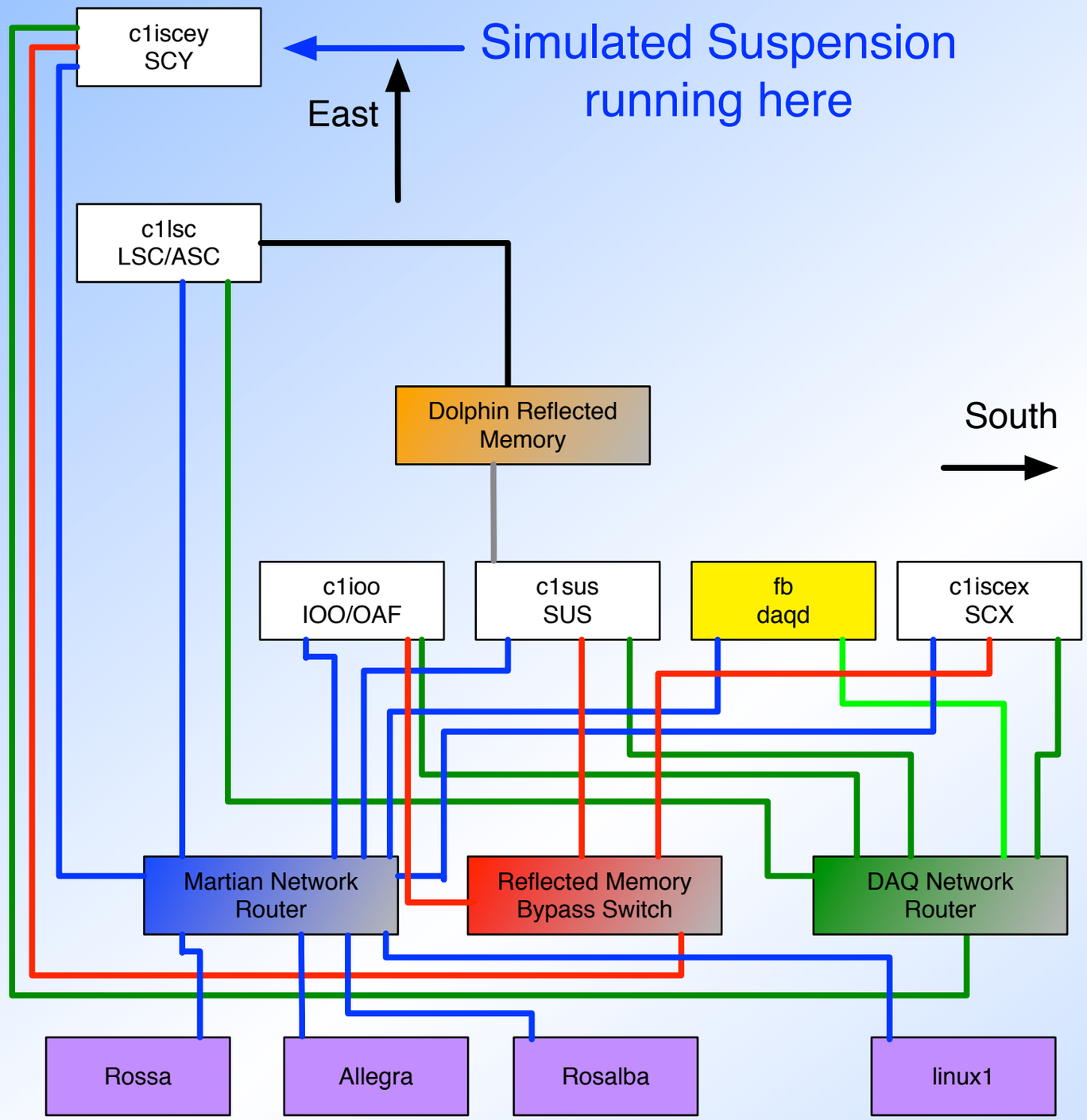
1 Gb/s Fiber

1 Gb/s Copper

2 Gb/s Copper

2 Gb/s Fiber

CDS computers and network at the 40m



Legend

- Gentoo Diskless Boot Server
- Gentoo Diskless Boot Front End
- Control Room Centos Computer

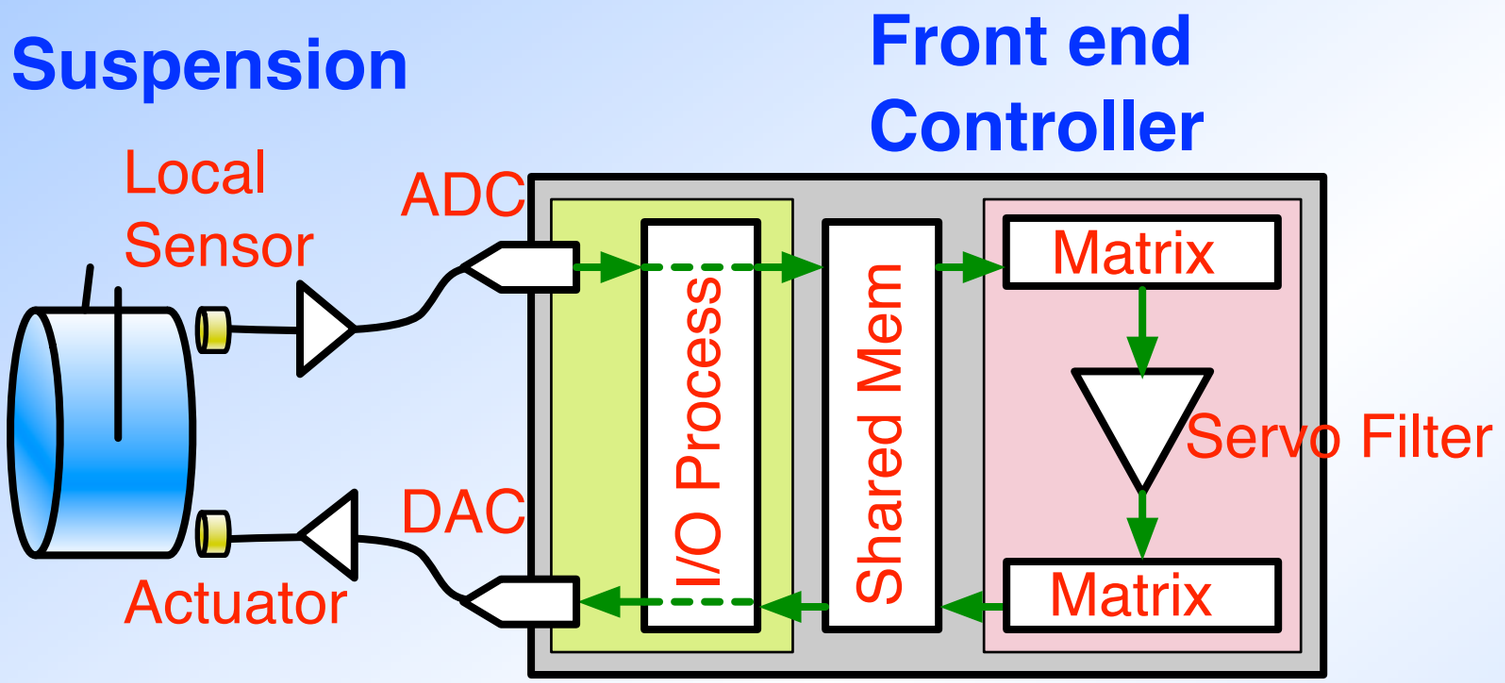
- 10 Gb/s Fiber
- 2 Gb/s Copper
- 1 Gb/s Fiber
- 1 Gb/s Copper
- 2 Gb/s Copper
- 2 Gb/s Fiber

Simulated Suspension
running here

Simulated Plant: Single Machine Case

- How do we do it? Here's an example.

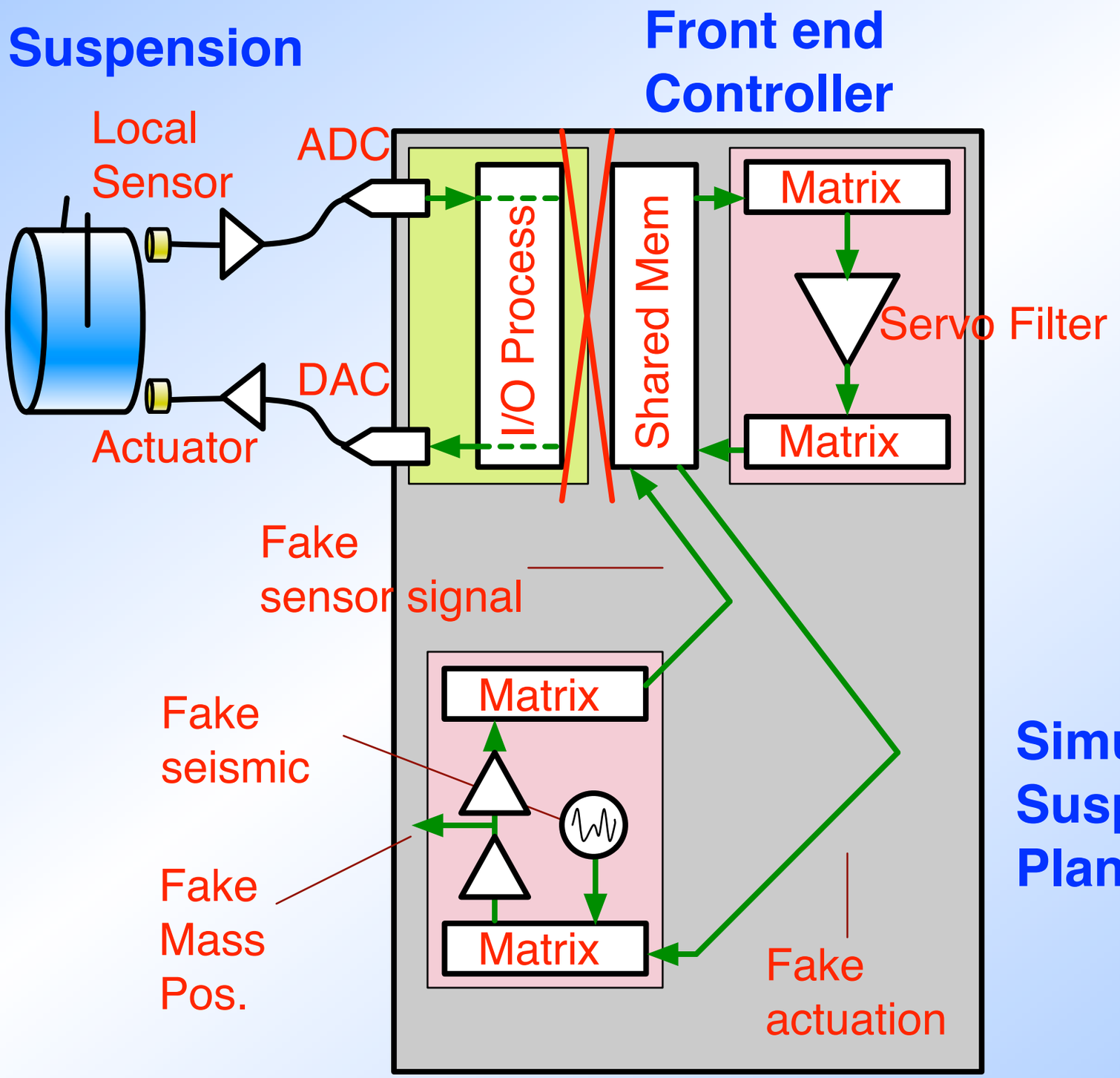
Suspension Controller



Simulated Plant: Single Machine Case

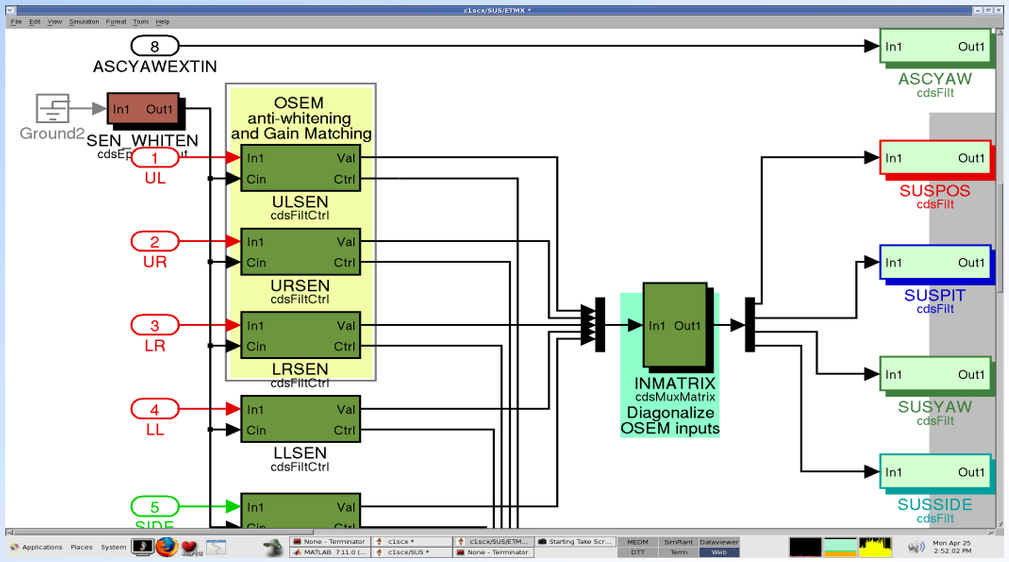
- How do we do it? Here's an example.

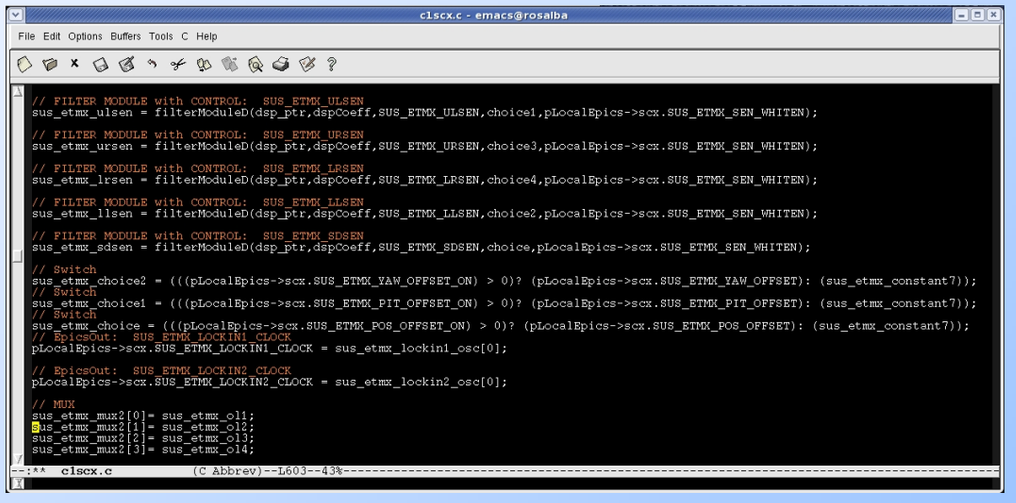
Suspension Controller



How to build the models?

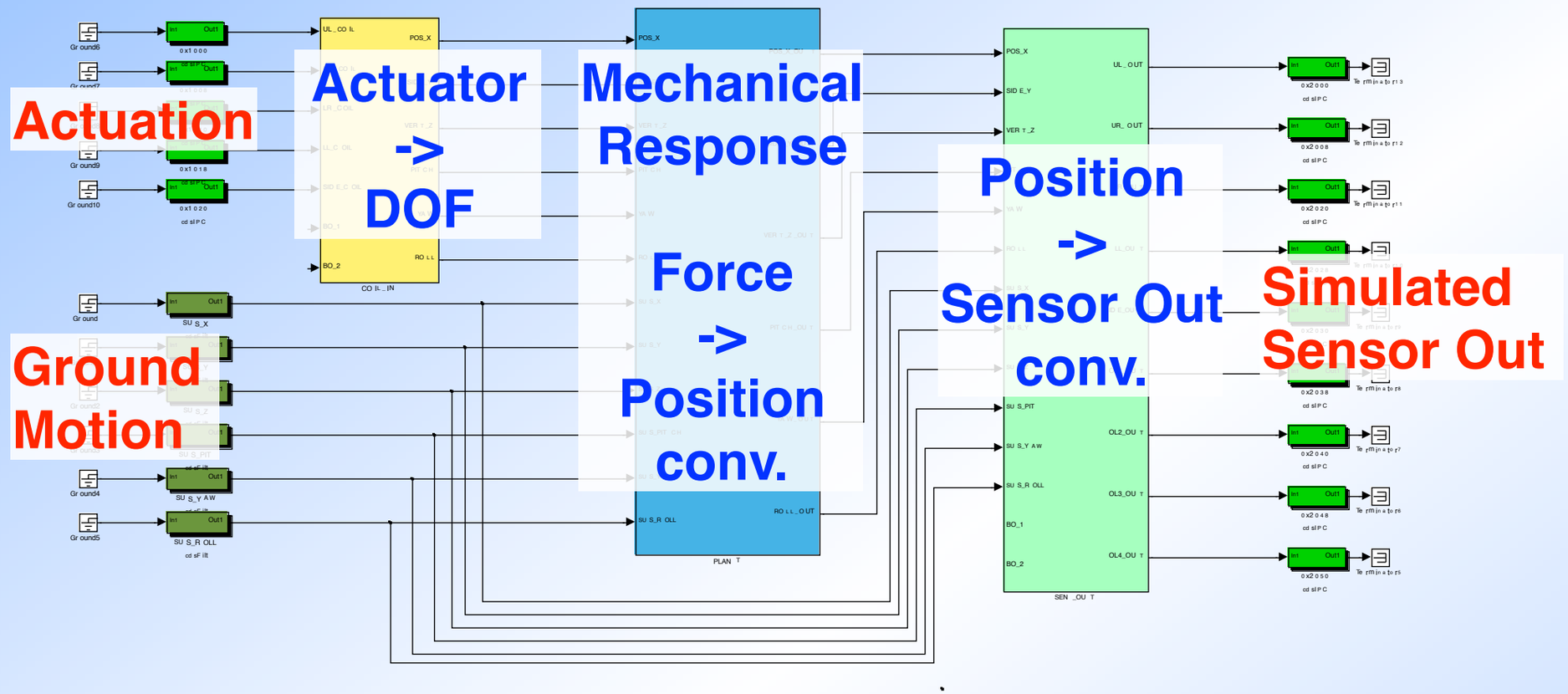
- **RCG:** Real-time Code Generator
  - Use **Simulink** as a GUI to design controls
  - RCG then parses the models to produce **C++** code
  - Code is compiled into **kernel modules** to run on **Front Ends**
  - **Frame builder** is also built to collect and record data





Actual Suspension Simulated Plant

Formed by matrices and arrays(or matrices) of filter modules



Actual Suspension Simulated Plant

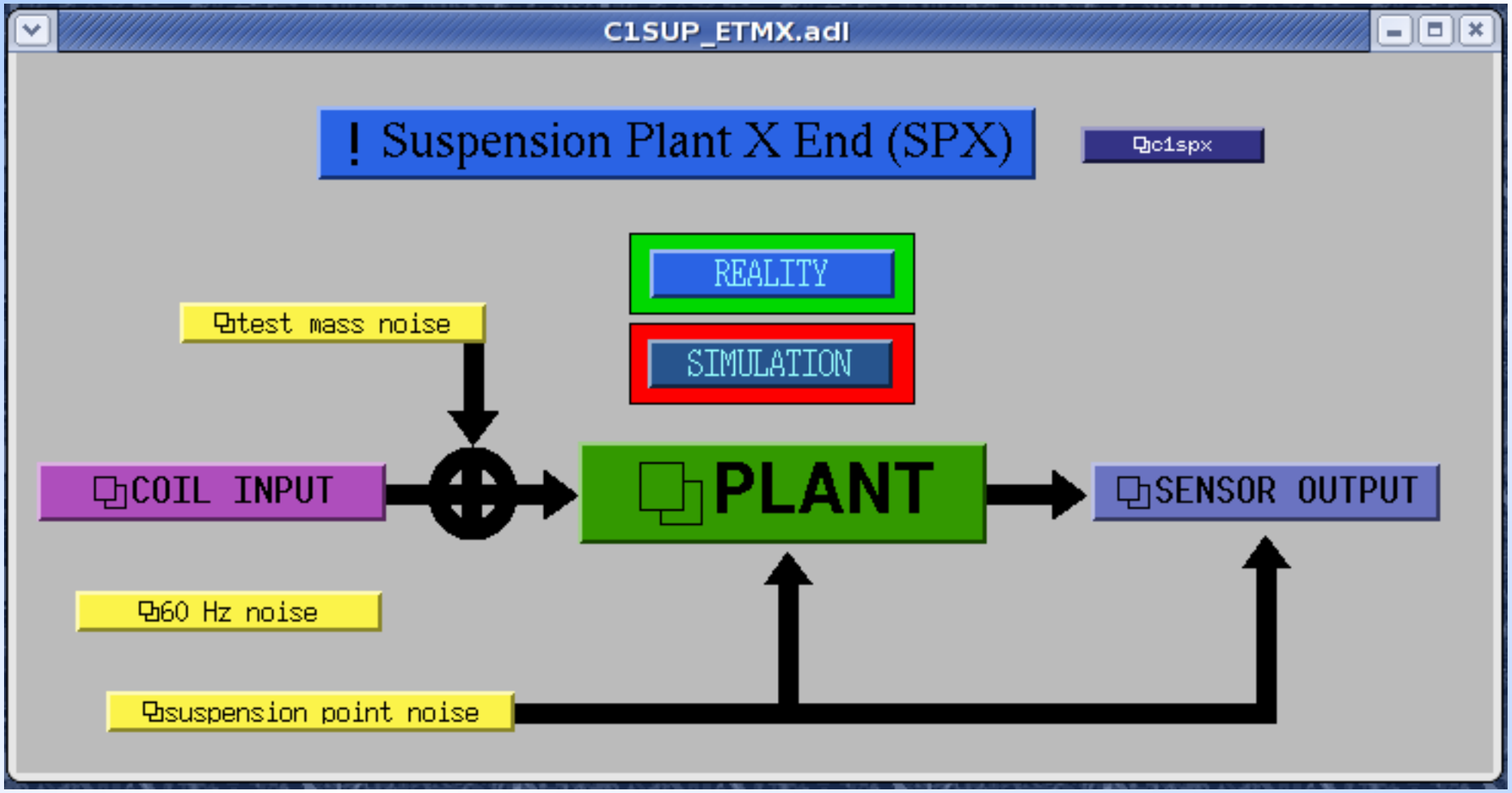
Formed by matrices and arrays(or matrices) of filter modules

Physics:
Mech. responses = transfer functions

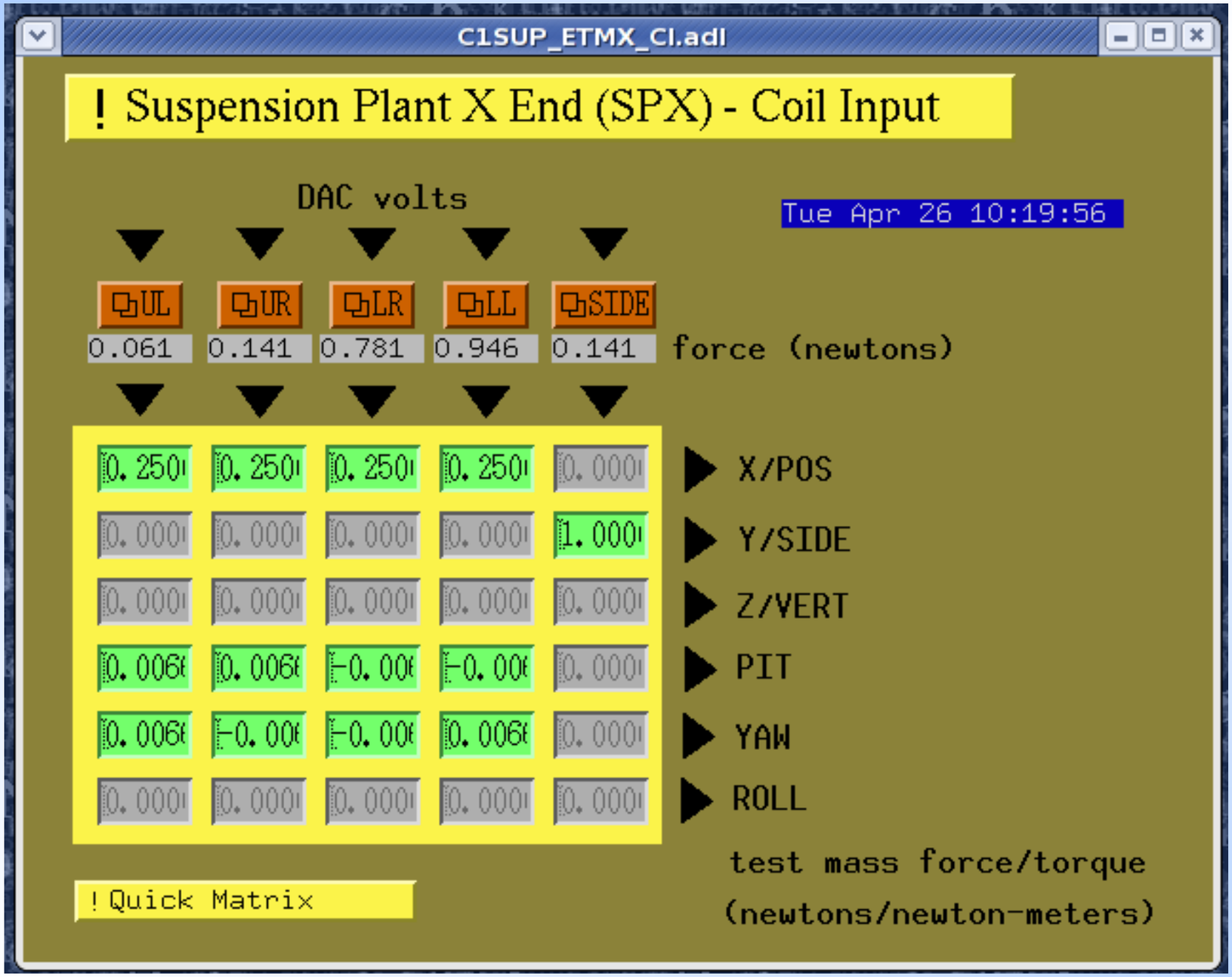
==> the responses are realized
by the filters and matrix elements

Once the model is built, it is flexible
i.e. single pendulum, quad pendulum,
etc...

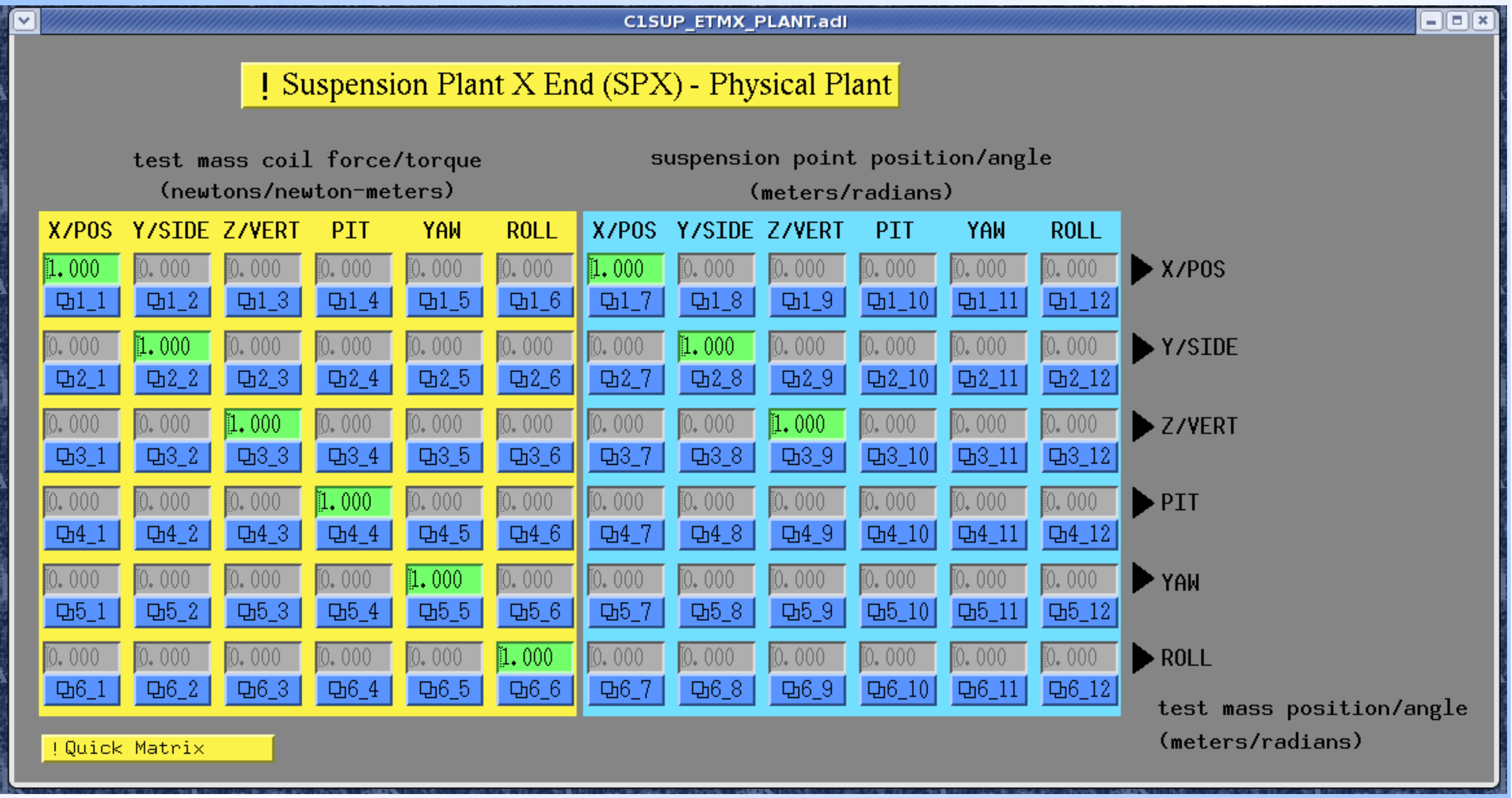
40M Simulated Plant Control Screens



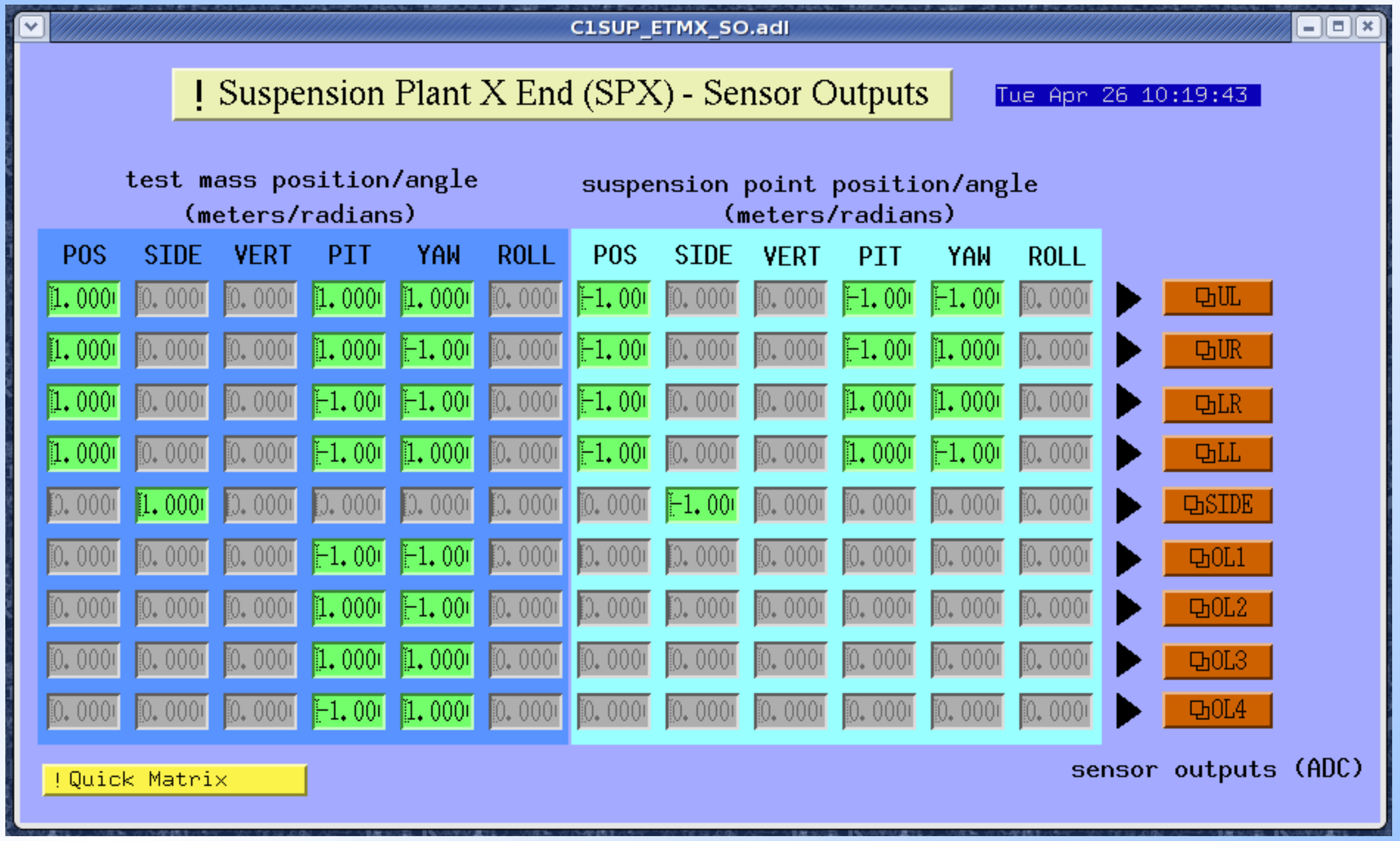
40M Simulated Plant Control Screens



40M Simulated Plant Control Screens

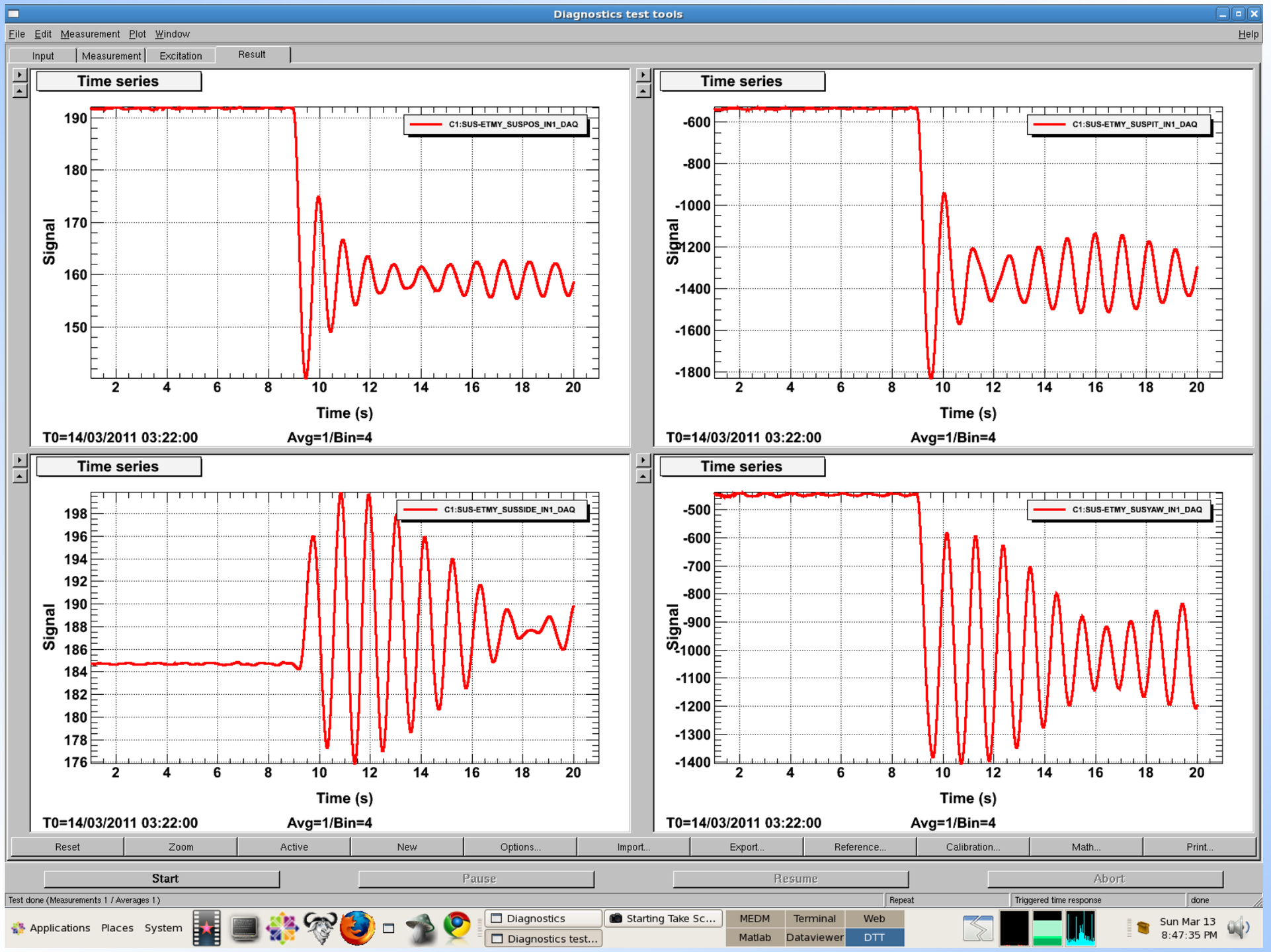


40M Simulated Plant Control Screens



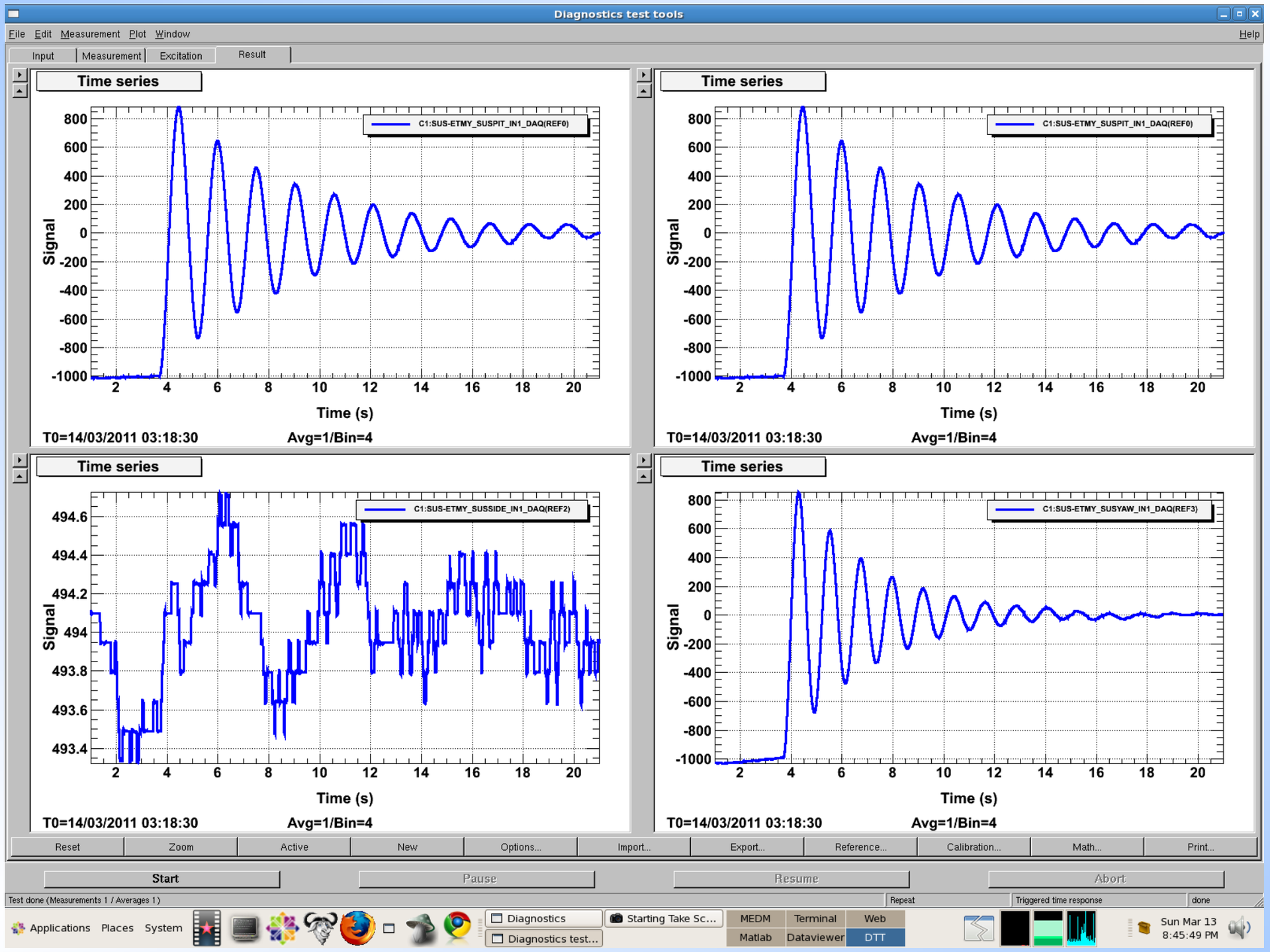
40M Simulated Plant Real Damping

- Data taken at the 40m with the normal commissioning tools

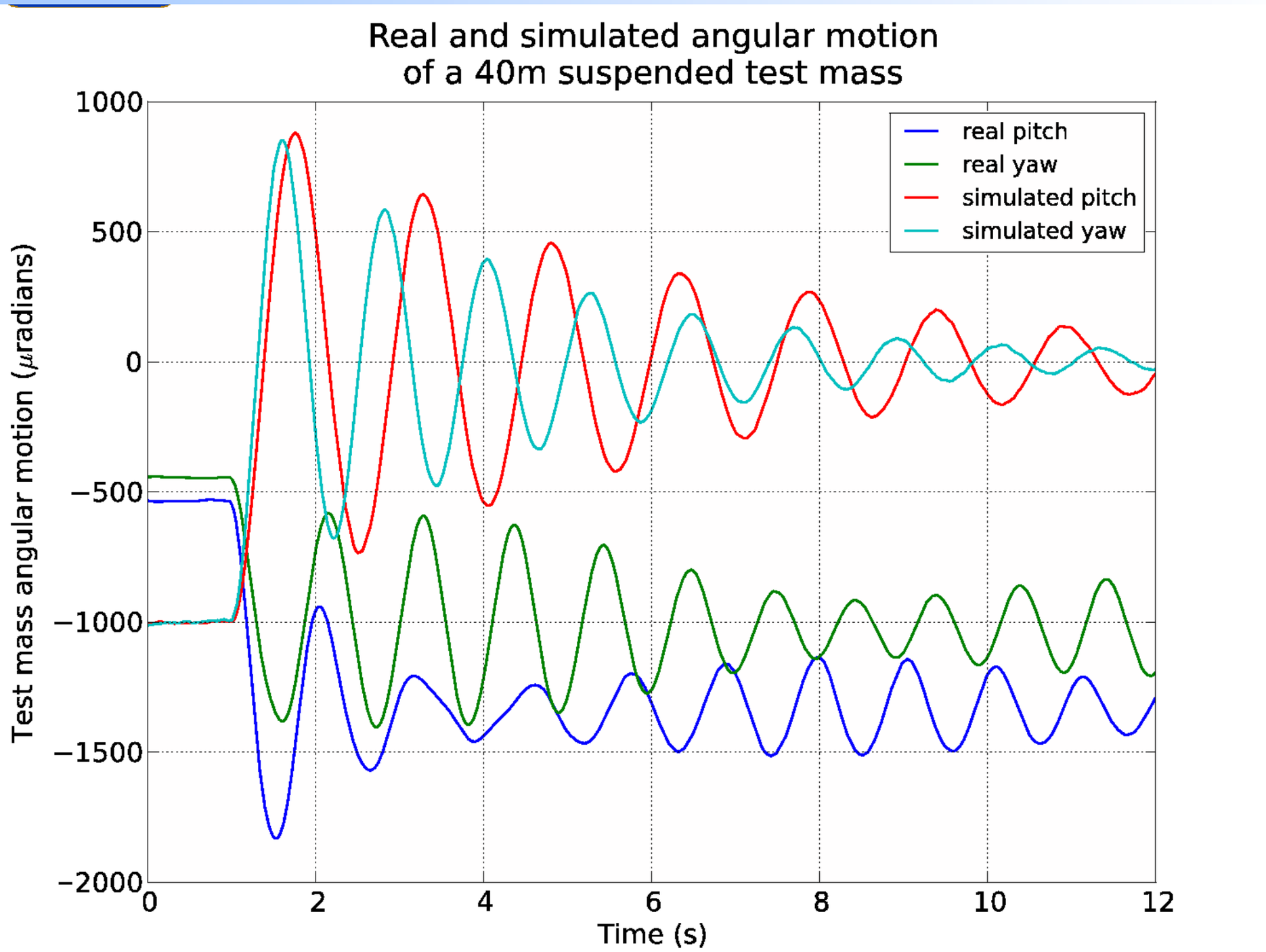


40M Simulated Plant Simulated Damping

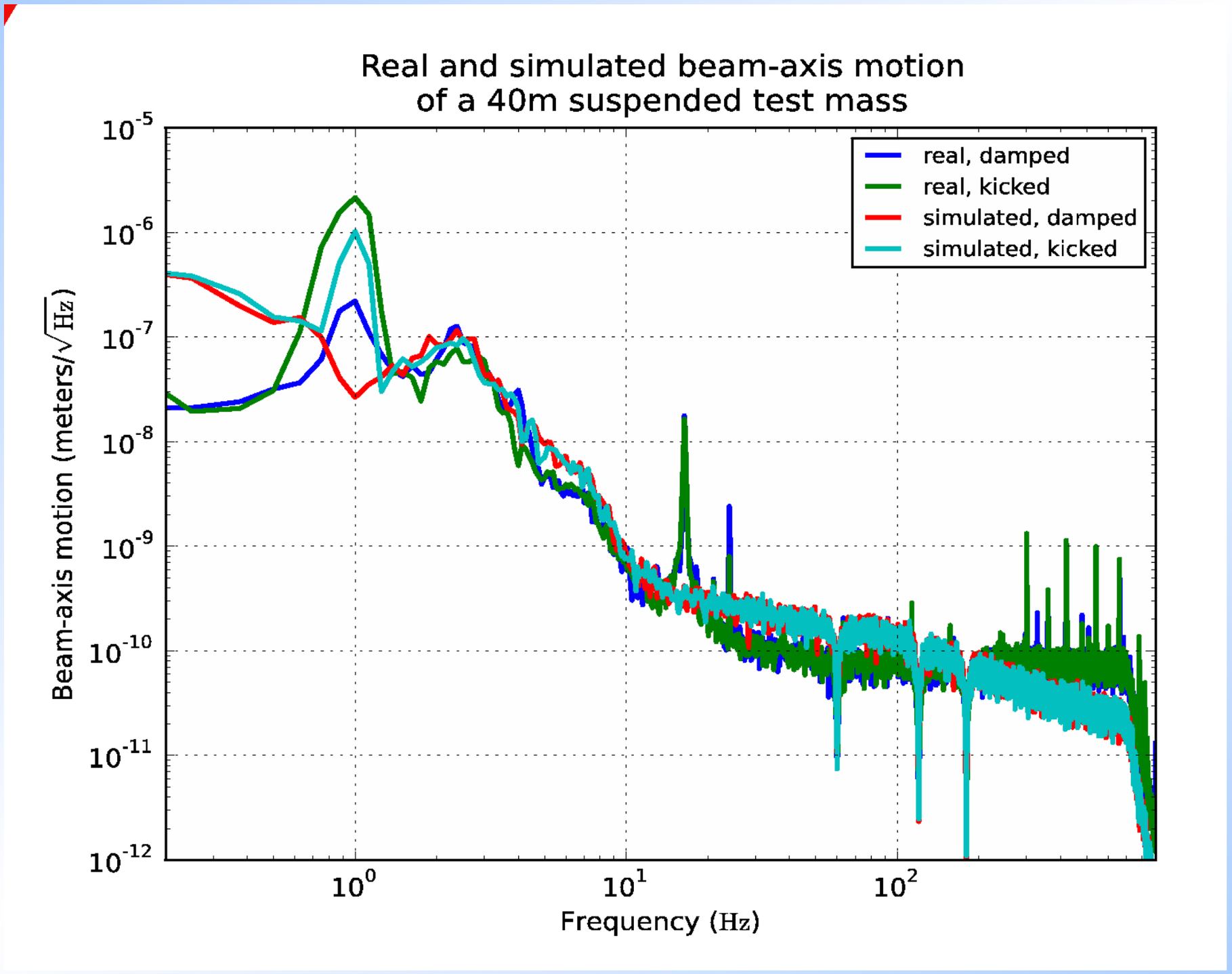
- Simulated data taken a few moments later with the same tools



40M Simulated Plant Damping Comparison



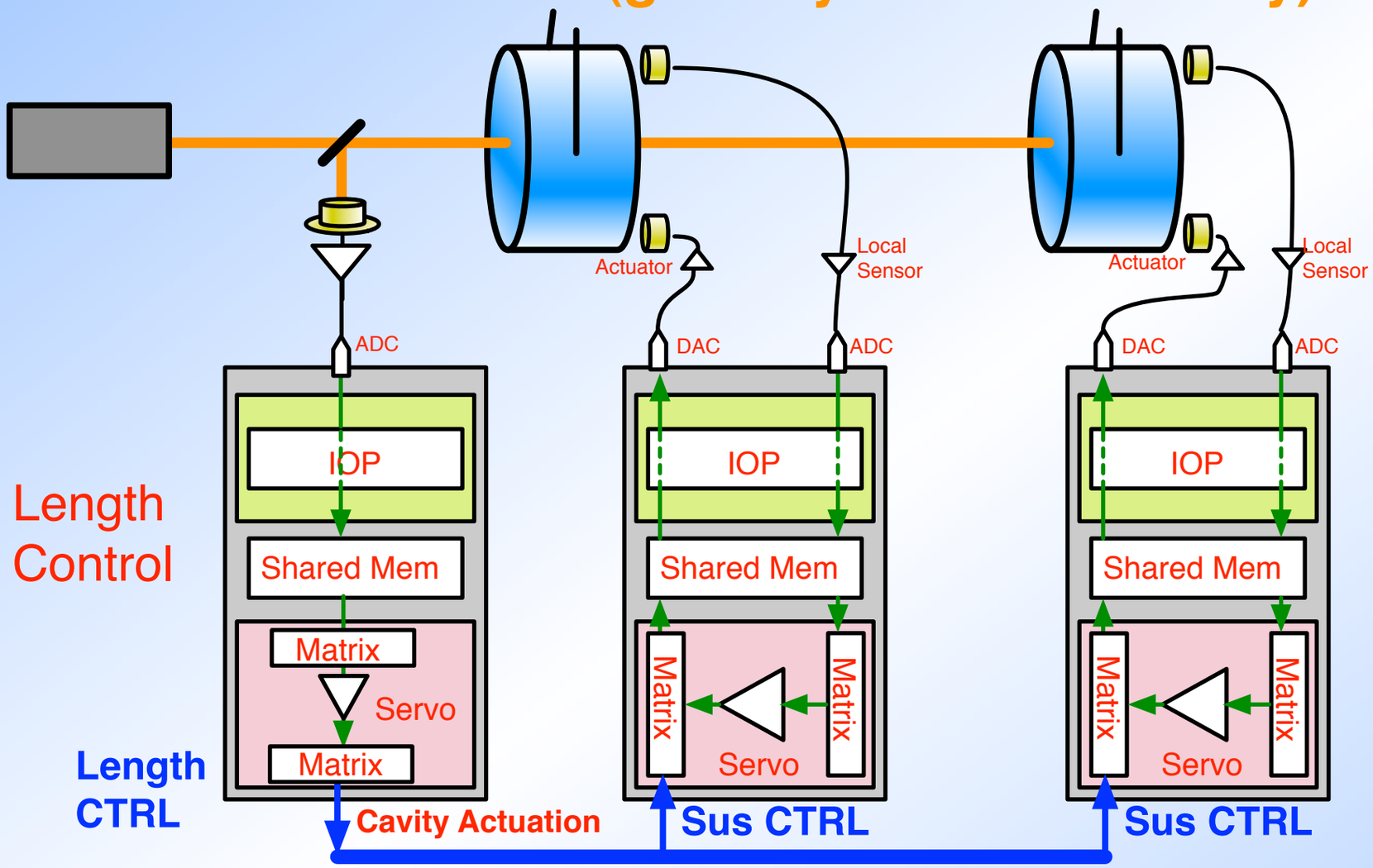
40M Simulated Plant Suspension Noise



Simulated Plant: multiple machine case

Length Controller ~ Multiple machine case

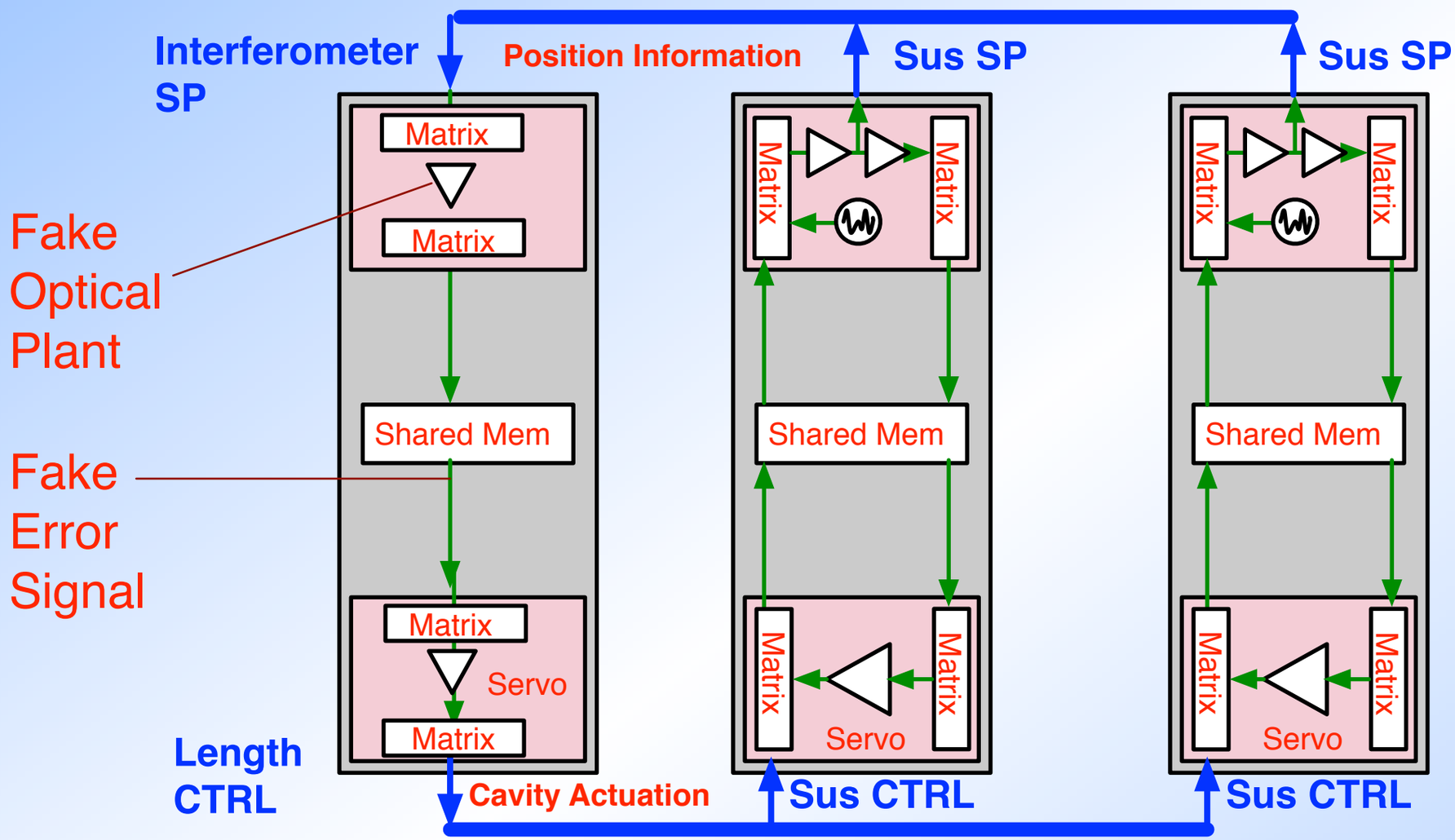
RFM/PCIe communication (globally-shared memory)



Simulated Plant: multiple machine case

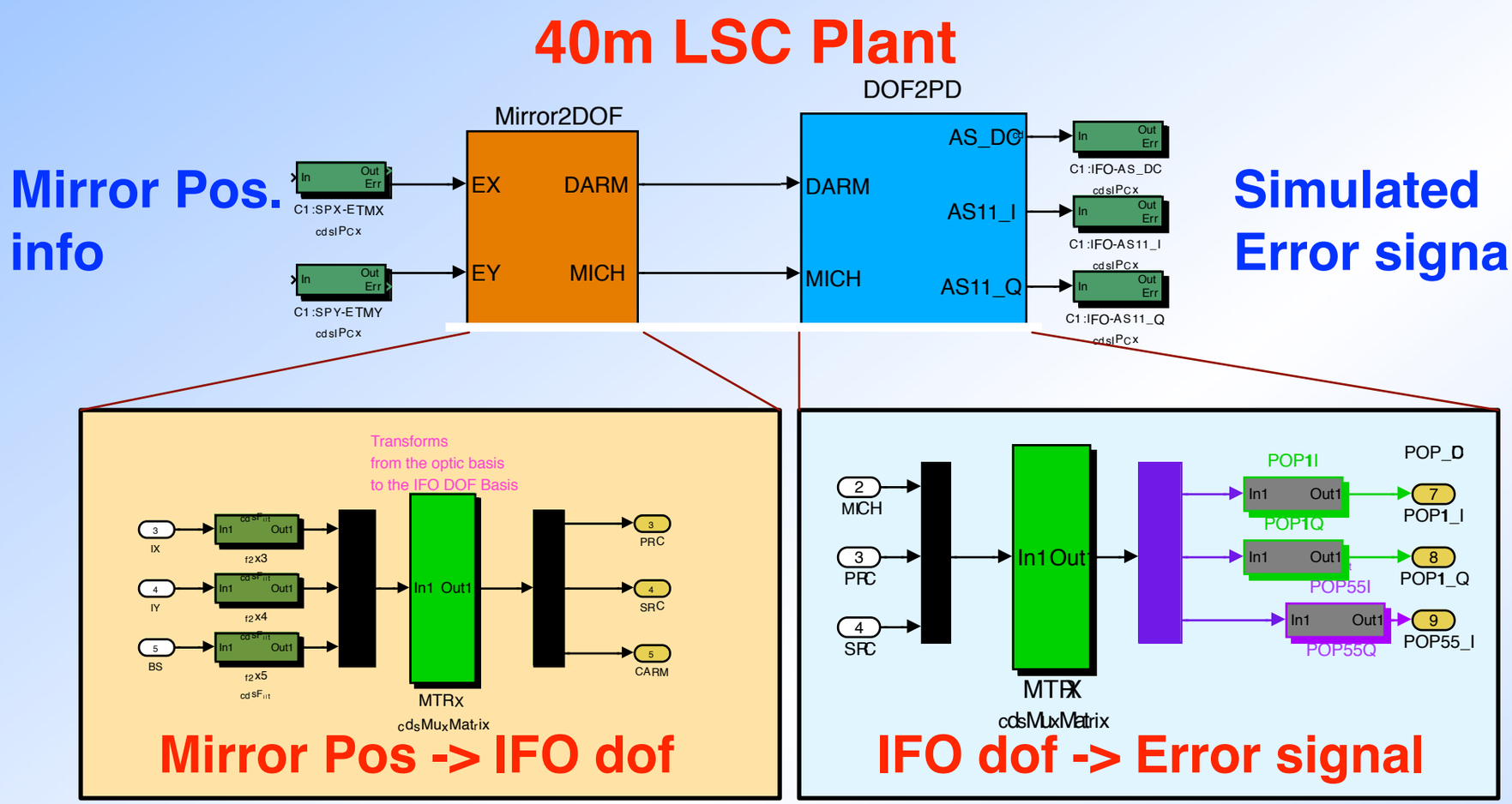
Length Controller ~ Multiple machine case

RFM/PCIe communication (globally-shared memory)



IFO Simulated Plant

Formed by matrices and arrays(or matrices) of filter modules



IFO Simulated Plant

Formed by matrices and arrays(or matrices) of filter modules

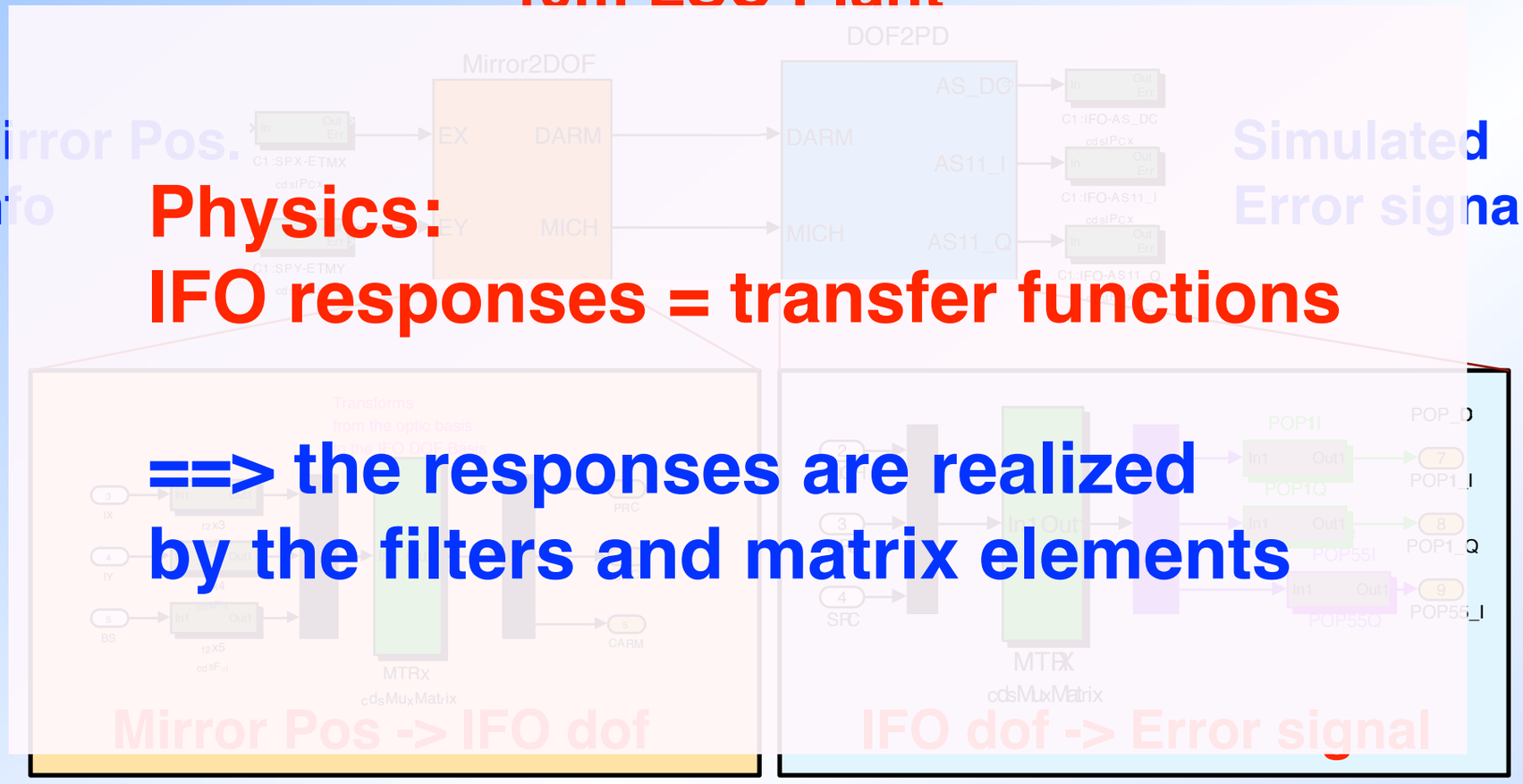
40m LSC Plant

Mirror Pos. info

Simulated Error signal

Physics:
IFO responses = transfer functions

==> the responses are realized by the filters and matrix elements



Future Plans

- More Noise!
  - Add thermal noise, shot noise, radiation pressure
  - Improve seismic, add option to use real seismometer data
  - Add earthquake stops (to prevent runaway suspensions)

- Finish IFO Plant:
  - Allow for the suspension models to talk to each other
  - Demonstrate near resonance conditions can be simulated

- Library:
  - Yesterday Alex finished a RCG recursive library parser
  - Build a library of all 40m control models
  - Allows changes to controls to propagate to simulated plant
    Just checkout and buld

- Statespace RCG block:
  - Allow for Statespace simulation in addition to using filters
  - Allow for import from currently existing SS models of Quads

Summary

- Simulated plant:

an IFO emulator for commissioning

Realized by the digital control system itself

Could run in parallel on the IFO hardware

- Enables to run IFOs only with the SW

Will make the commissioning easier & faster

Will make the noise hunting easier & faster

Will help the DA development / tests

- Basic suspension simulation at the 40m

Will continue to add noise and improve the plant

- Looking aLIGO quadruple pendulum models

Planning for when we apply these to aLIGO

- Thanks to many people for their contributions:

Rolf Bork, Alex Ivanov, Valera Frolov, Chris Wipf