LASER INTERFEROMETER GRAVITATIONAL WAVE OBSERVATORY
- LIGO -
CALIFORNIA INSTITUTE OF TECHNOLOGY
MASSACHUSETTS INSTITUTE OF TECHNOLOGY

Document Type	**LIGO-G000005-00-**	**E**	Feb. 2000

End to End simulation
Talk given at Stanford Univ. 2/16/2000

Hiro Yamamoto / LIGO Lab

Giancarlo Cella / Pisa Univ

Distribution of this draft:

xyz

This is an internal working note
of the LIGO Project..

California Institute of Technology
LIGO Project - MS 51-33
Pasadena CA 91125
Phone (626) 395-2129
Fax (626) 304-9834
E-mail: info@ligo.caltech.edu

Massachusetts Institute of Technology
LIGO Project - MS 20B-145
Cambridge, MA 01239
Phone (617) 253-4824
Fax (617) 253-7014

WWW: http://www.ligo.caltech.edu/

End to End simulation

Hiro Yamamoto / LIGO Lab

Talk at Stanford University
February 16, 2000

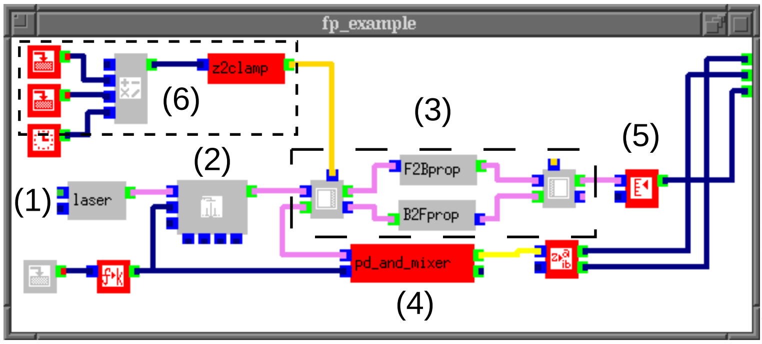
- Overview
  - » What is it
  - » Examples : FP, Lock Acquisition
- Simulation software - a generic tool
  - » Physics in the simulation
  - » Mechanics simulation
    - – Dr. Giancarlo Cella of Pisa University
  - » Graphical User Interface
- LIGO simulation
  - » Explicit implementation of LIGO hardware using the e2e framework
  - » Example : PSL
  - » Applications started
    - – 2K FP
    - – full LIGO

Overview
the End to End simulation package

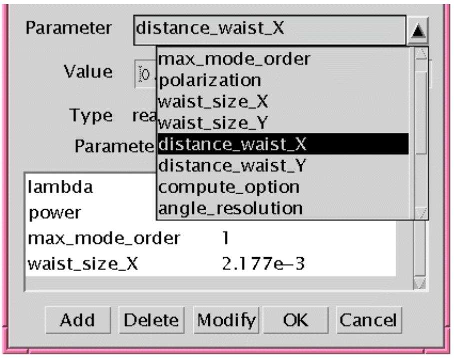
- General purpose GW interferometer simulation program
  - » Generic tool like matlab or mathematica
  - » Easy to simulate a wide range of configuration without modifying and revalidating codes
- Simulation program
  - » Time domain simulation written in C++
  - » Optics, mechanics, servo ...
  - » Easy to add new phenomena by concentrating on physics, not on programming
- Graphical user interface
  - » Define the setup to simulate
- Need to implement specific hardware and software configurations
  - » Subsystems
  - » Full LIGO

Example 1

FP cavity - step-by-step



(a) GUI front-end of e2e



Parameter distance_waist_X

Value 0

Type rea

Paramete

lambda	
power	
max_mode_order	1
waist_size_X	2.177e–3

(b) parameter setting windows

(a) e2e programming using GUI

(1) Laser : scalar/multi mode, power,...

(2) Pockels Cell : RF, Γ, #SBs, ...

(3) FP cavity : length, alignment, r, t...

(4) photo diode and mixer : RF, shot noise

(5) power meter : SB, mode

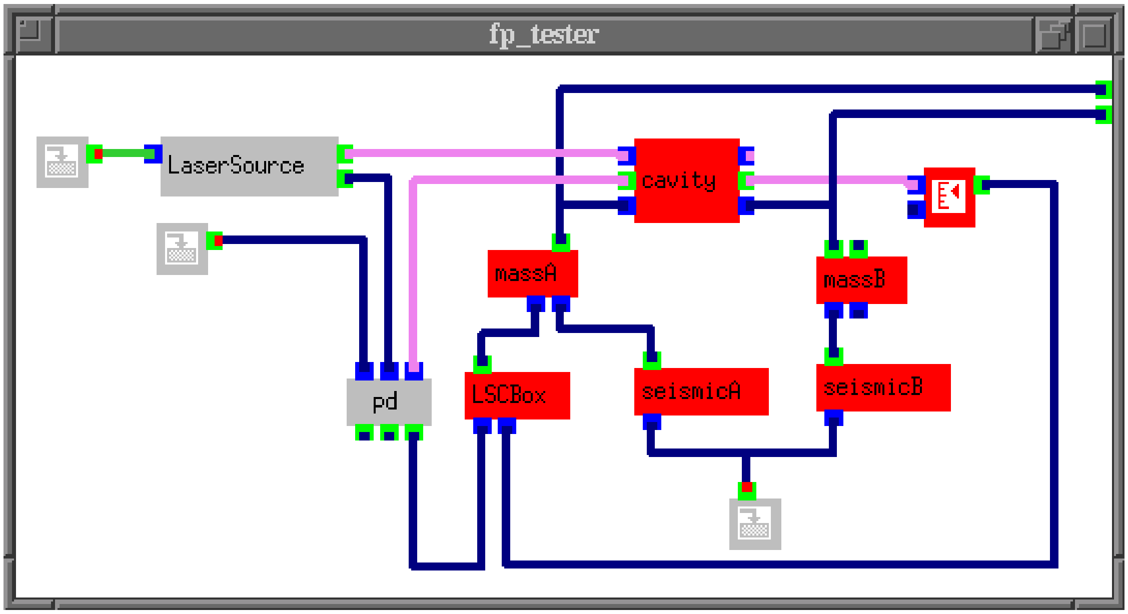
(6) $x = x_0 + v\ t$: mirror motion

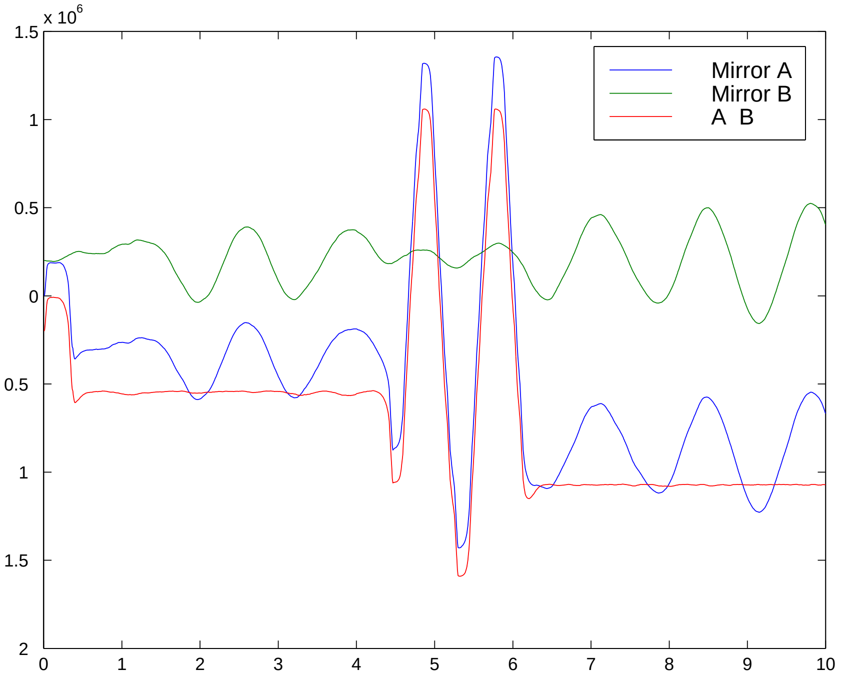
(b) setting of parameters

do not need to know all parameters

Example - 2

simulation of lock acquisition
(base of 2k FP simulation)

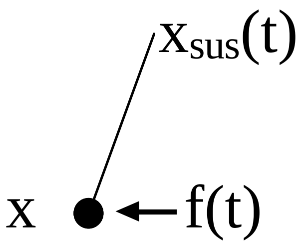




Physics in the simulation

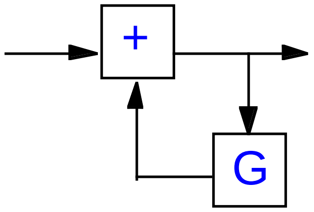
Time domain simulation

- Analog process is simulated by a discritized process with a very small time step (10^{-7}~ 10^{-3} s)
- Linear system response is handled using digital filter
  - Transfer function -> z trans -> digital filter
  - Pendulum motion



$$x = \frac{1}{s^2 + \gamma s + \omega_0{}^2}\left(\frac{f}{m} + \omega_0{}^2 x_{sus}\right)$$

  - Analog electronics
- Easy to include non linear effect
  - Saturation, e.g.
- A loop should have a delay

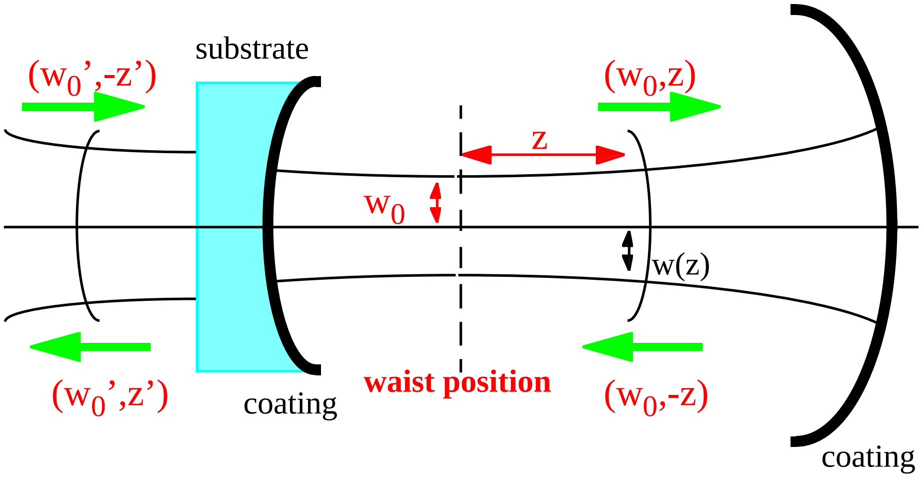


  - Need to put explicit delay when needed
  - Need to choose small enough time step
  - working to relax this constraint

Physics in the simulation

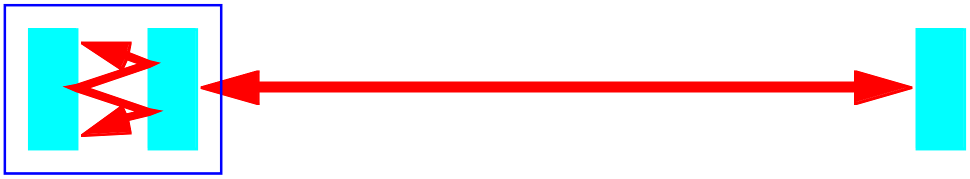
Optics - 1

- Time domain modal model
  - » field is expanded using Hermite-Gaussian eigen states, characterized by the waist size (w_0) and position (z).
  - » Tilt, shift and curvature mismatch are calculated using mode decomposition matrix as perturbation
  - » Lenses change the base (w_0 and z)
- Completely modular
  - » Build planar optics configuration by combining mirrors and propagators



Physics in the simulation
Optics - 2

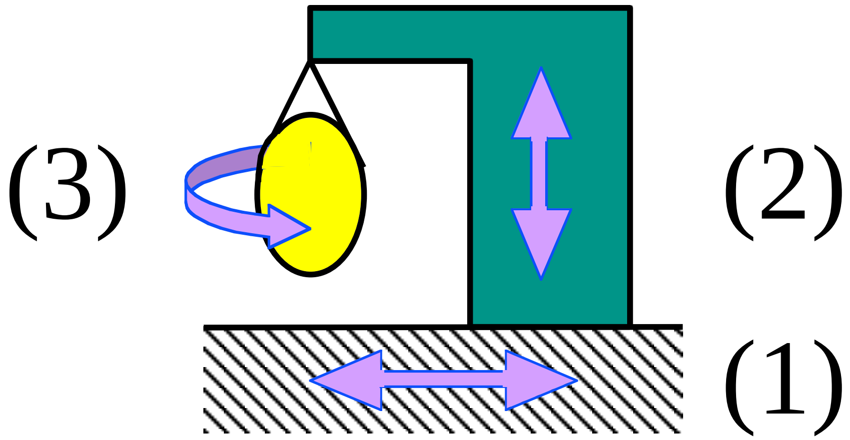
- Summation cavities for fast simulation



  - Short cavity response is calculated using linear approximation
  - FP, triangular cavity, recycling Michelson
  - $10^2 \sim 10^4$ faster
- Remodeling from the ground up
  - Multi mode, polarization, logical organization
  - Better expandability, easier maintenance

Physics in the simulation
Mechanics

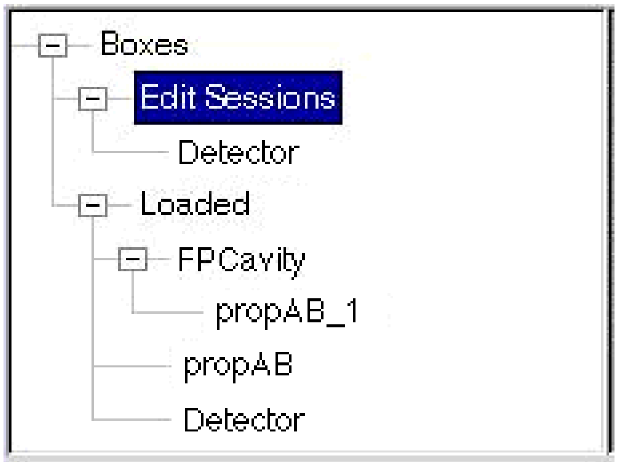
- Basic model
  - » Seismic motion from measurement
  - » Parametrized HYTEC stack
  - » Simple single suspended mirror
    - – 4 sensors and actuators
  - » Thermal noise added in an ad hoc way
- Mechanics Simulation Engine
  - » G.Cella of Pisa Univ.



Graphical User Interface

preparing inputs for the simulation

- The learning curve should be minimal for a new user
  - » Very successful
  - » Many happy users
- Complex configurations need to be constructed, edited and maintained
  - » Basic functionality ready
  - » Need better handling of hierarchical directory structure
    - – Different subsystems developed separately and merged together without interfering
  - » Need tree view implemented
    - – Easy access to deeply hidden target
  - » Find by name & type
  - » Copy and Paste
  - » Etc, etc
  - » Stability



partial list of

Other things to do

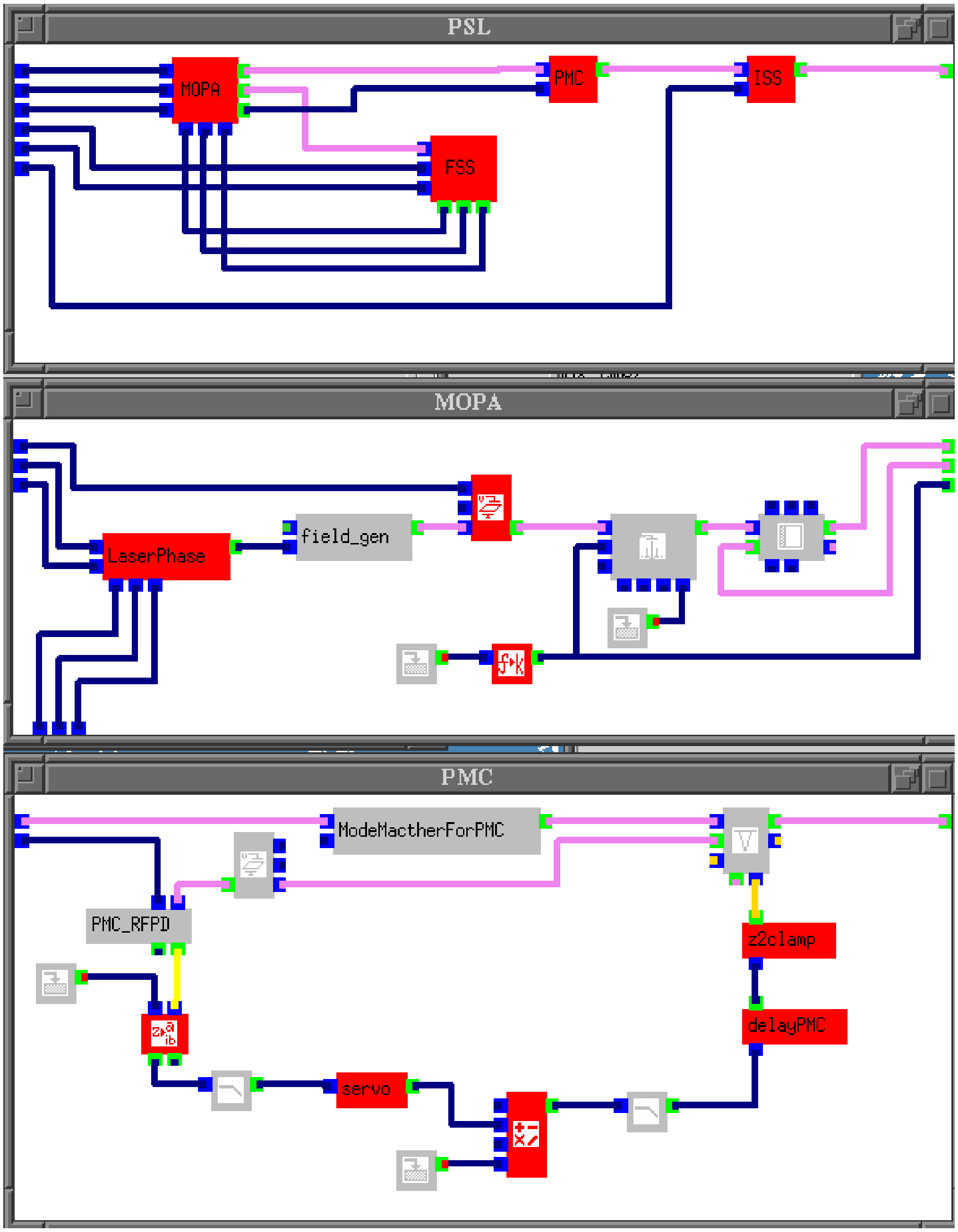
- Implement thermal lensing effect in the optics model
- Save intermediate state and load it later to resume simulation
- Macro support
- Improvements of the interface of the simulation engine program
- Different streams of inputs and outputs
- Include measured phasemap in the simulation
- Very fast simulation for long time psudo data production
- Support parallelization, possibly by thread
- Use of different time steps in different part of the simulation
  - » τ[PSL, IOO] < τ[COC] < τ[SUS,SEI]
- Automation of choosing the simulation time step
- Code improvements and validations
- Documentation
  - » Bare minimum necessary for our work
  - » Outdated tutorial

LIGO simulation

- Based on the End to End simulation package, the LIGO simulation code is created
  - » LIGO specific hardware and software setup
  - » Comparison between subsystem measurements and simulation
    - – Validation of the model
    - – Understanding the limitation of the simulation
- PSL, IOO, COC, LSC/ASC, SUS/SEI
- e2e programmers works with experts of each subsystem to build the subsystem model
  - » R.Savage for PSL
  - » P.Fritchel for LSC/ASC
  - » ...

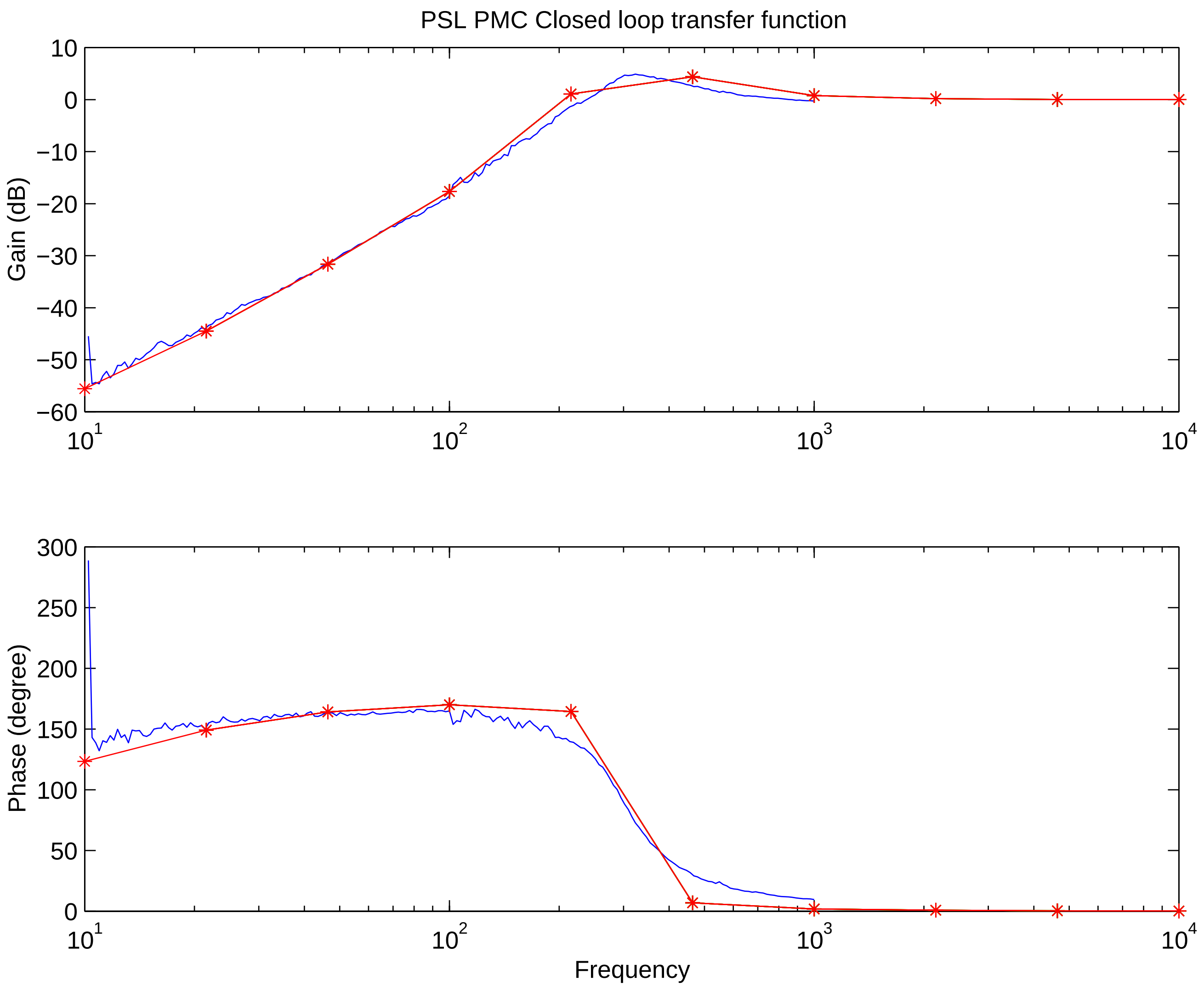
LIGO modeling

PSL - setup



LIGO modeling

PMC - transfer function



Applications

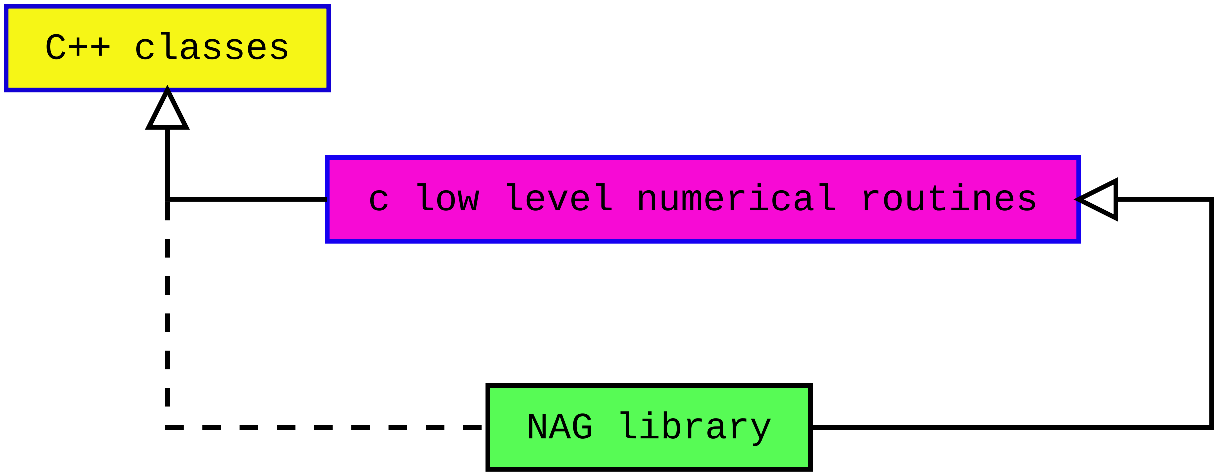
- 2K FP
  - » Error signal structures
    - – various resonance and interference patterns of higher order harmonics and higher order modes
  - » Lock Acquisition threshold velocity
  - » 60Hz and harmonics effects
    - – $d\varphi(t) = 450 \sin(2 \pi 60 t)$
  - » Misalignment effect when locked
  - » Help to understand the characteristics of FP
    - – Drifts (length or freq), SB resonance in arm
  - » LSC/ASC simulation
    - – Validation of the model
    - – Identify the problem of the simulation
- full LIGO
  - » Lock Acquisition re-design started using e2e
    - – full length and alignment degree of freedom
    - – realistic methods
    - – realistic noises
    - – ...

MSE- A mechanical simulation engine for the LIGO E2E model

Giancarlo Cella

Stanford,2/16/2000

1. **General structure of the library**
2. **How to use it**
3. **Modelization techniques and problems**
4. **Future developments**



C++ classes

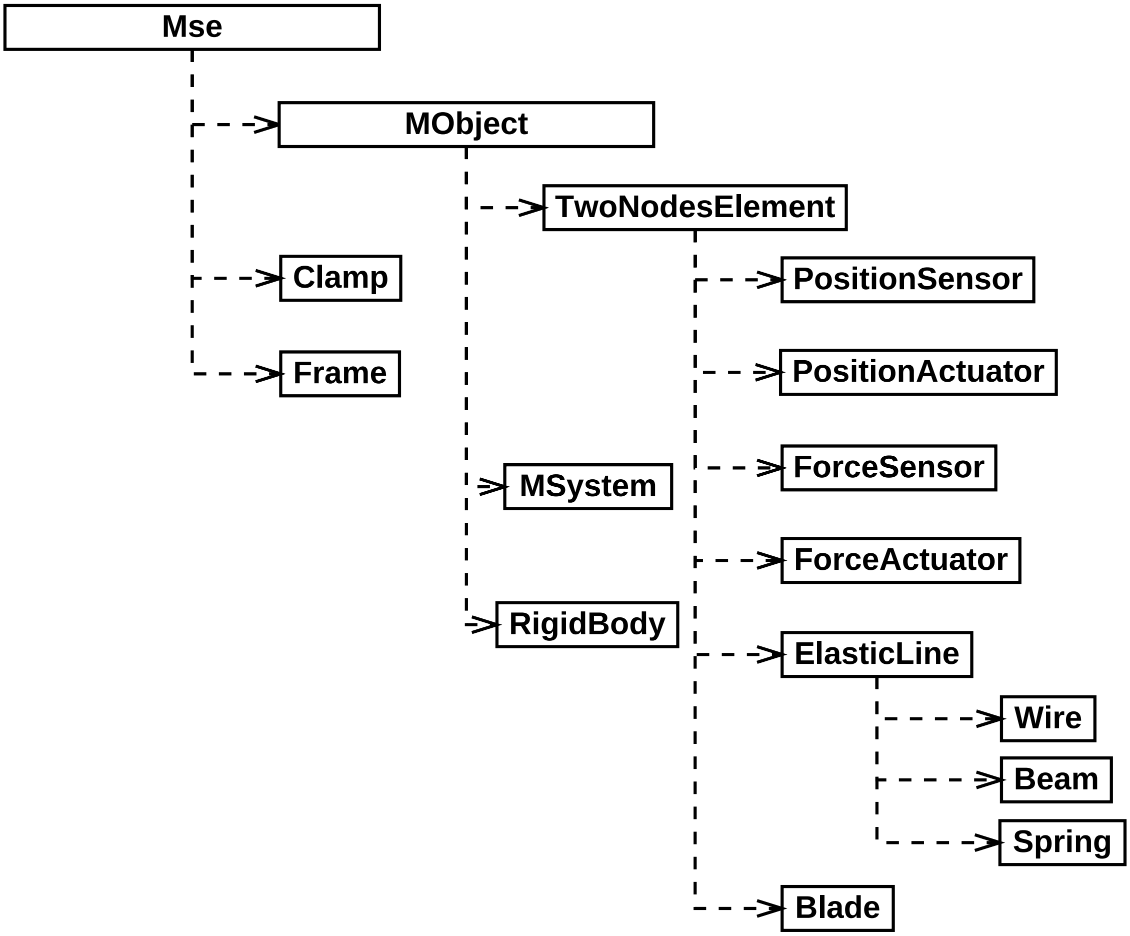
c low level numerical routines

NAG library

General principles

- ✔ It is a fully tridimensional simulation. In this way it is possible to give extimates on cross couplings connected to system asymmetries.
- ✔ It is a modular environment. The system is partitioned in subunities, and each of them can be modeled internally in an arbitrary way.
- ✔ A model can be selectively refined. For example it is possible to set the number of internal modes of a given mechanical component, or to choose different representation for it.
- ✔ The equilibrium working point for the system is automatically calculated. There is no need to insert a lot of geometrical positioning parameters, only the connections between elements must be specified.
- ✔ It is (hopefully) easy to use.

Implementation

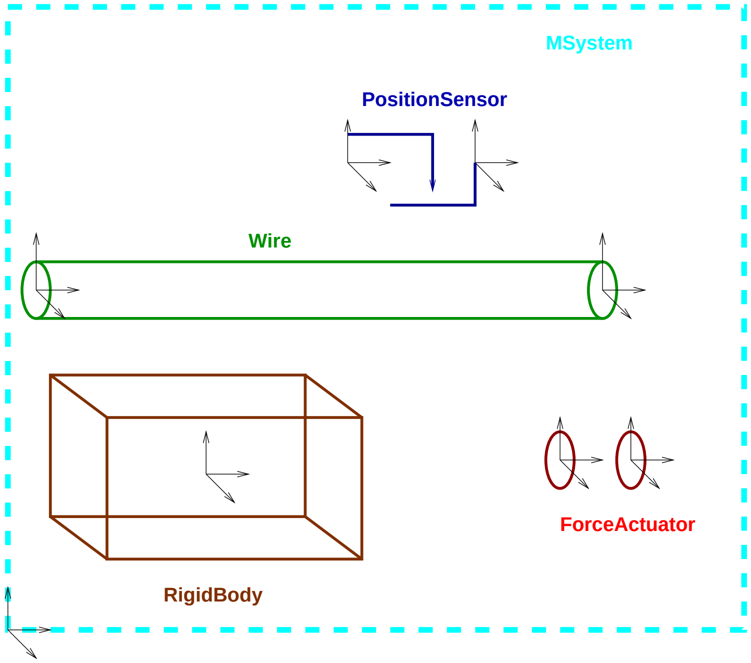


- ✔ **Frame:** is a specification of **position** and orientation of a point (6 d.o.f.)
- ✔ **Clamp:** is a rigid connection between two frames
- ✔ **MObject:** is a collection of frames with some dynamic defined among them
- ✔ **MSystem:** is an inertial frame, and also a container of MObjects

A simple example: suspended mirror

First we declare the system and the objects which compose it:

```
MSystem pendulum;
RigidBody mirror;
Wire wire1,wire2;
ForceActuator coil1,coil2,coil3,coil4;
PositionSensor sensor;
```

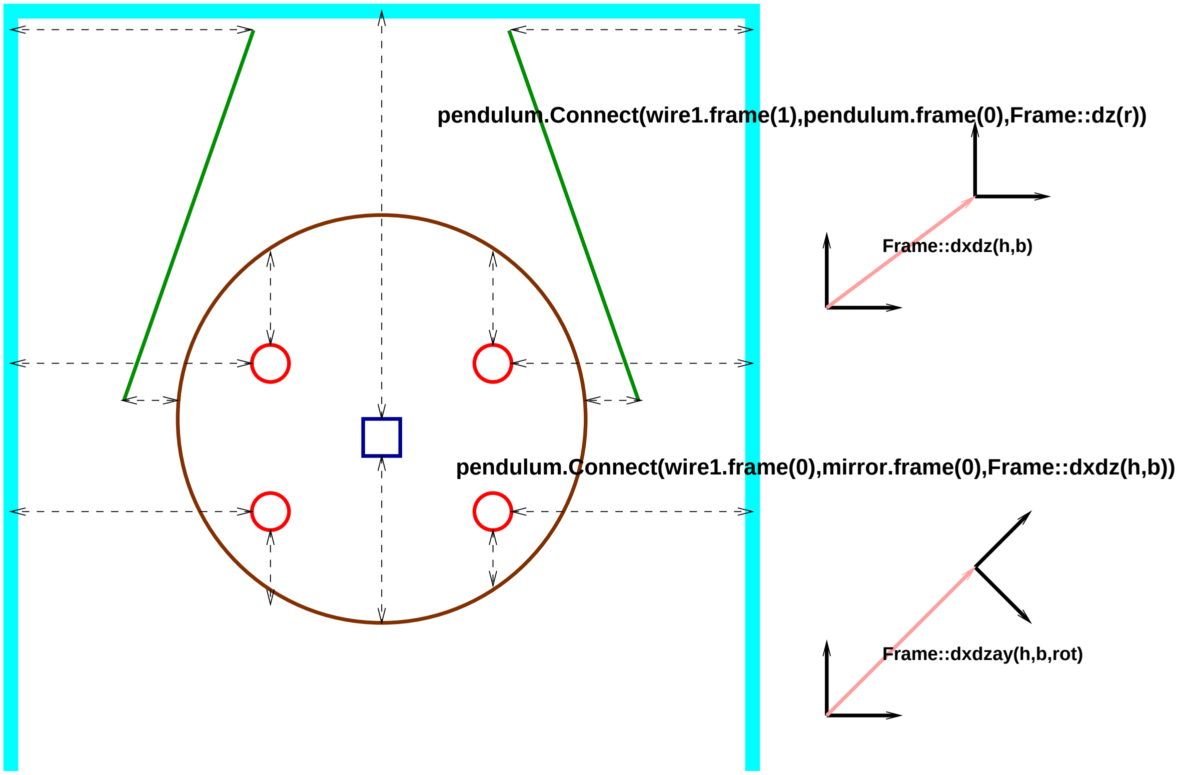


And we set the relevant parameters (mass, inertia tensor components, wire diameter etc.)

Now the system must actually be constructed. This is obtained clamping frames together.

System construction

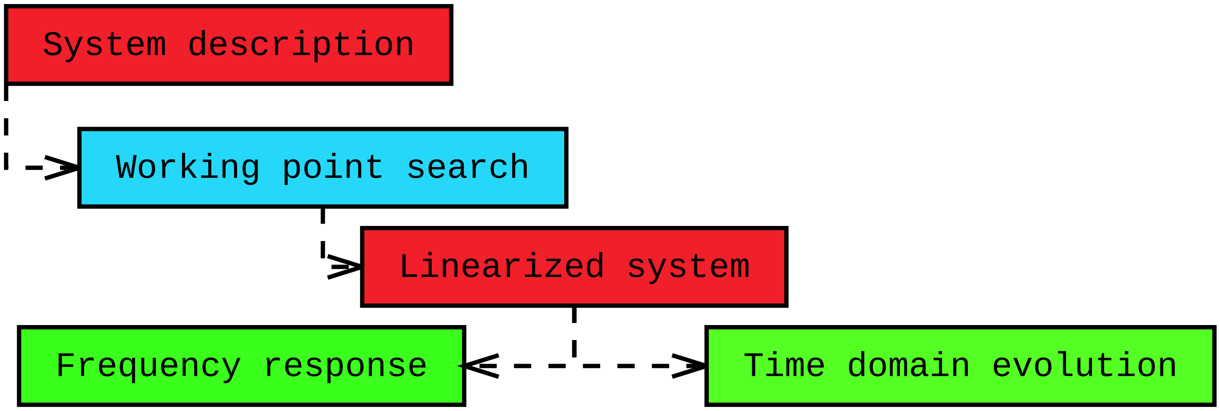
Each clamp fixes completely the relative displacement and orientation of frames.



- ✔ Only the relative constraints must be specified
- ✔ Optionally some (temptative) absolute frames positions can be given. This can help the search for the equilibrium position
- ✔ Force actuator apply forces between two objects (3 forces + 3 torques).
- ✔ Position actuator measures the relative position of two object (3 displacements + 3 rotations)

Simulation

A prerequisite is the search for the correct working point:


System description

Working point search

Linearized system

Frequency response

Time domain evolution

Coming back to the example.

```
while(t=pendulum.CurrentTime()<1.0) {
  coil1.set_fx(force(t));
  pendulum.TimeDynamics();
  cout << sensor.get_x();
}
```

or, in the frequency domain,

```
coil.set_fx(1.0,0.0);
coil.set_fy(0.0,1.0);
pendulum.FrequencyDynamics(f);
cout << sensor.get_x_mod();
```

Simulation - system description

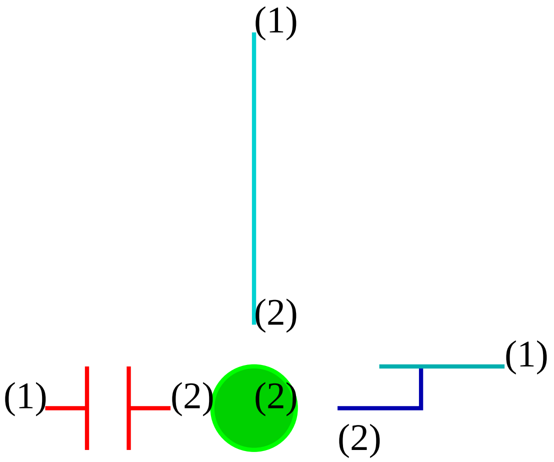
Each MObject must be able to provide:

1. A way to calculate the **static** forces on the frames, given their positions. This will be used in the working point search.
2. A **linearized** dynamic equation like

$$M\frac{d^2\vec{x}}{dt^2} + \Lambda\frac{d\vec{x}}{dt} + K\vec{x} = \vec{f}$$

   M,Λ and K are the mass,damping and stiffness arrays.

3. **Linear** relations between $\vec{x}$ and I/O variables (for an actuator or a sensor)



The system is partitioned in a collection of connected frames groups.

A reference frame is choosen in each group. This is optimized for numerical accuracy.

In this case:
Group 1: fixed inertial frame, no d.o.f.
Group 2: mass etc., 6 d.o.f.

Now we have a set of independent coordinates to describe system configuration.

Simulation - Nonlinearities

It is important to provide a non linearized static description. Look at the GAS blade:

$$W = \int_0^L \left[\frac{\gamma(l)}{2} \left(\frac{d\theta}{dl} \right)^2 - F_x \left(\cos\theta - \frac{x_0}{L} \right) - F_y \left(\sin\theta - \frac{y_0}{L} \right) \right] dl$$

The forces (and $\theta(l)$) are solution of the nonlinear problem

$$\frac{\delta W}{\delta \theta(l)} = 0, \ \frac{\delta W}{\delta F_x} = 0, \ \frac{\delta W}{\delta F_y} = 0$$

which can be, for example, discretized and solved numerically. The Blade object can do that, and the search for the working point can be schematized as follows:

1. Fix consistently the position of each frame
2. Ask each MObject to compute forces on frame
3. Compose these to find the ones coniugate to d.o.f.
4. Update d.o.f. (and frames) using some appropriate search algorithm
5. Go to point 2 until equilibrium is found

We end with F_x^{eq},F_y^{eq} and $\theta(l)^{eq}$

Simulation - time & frequency domain

When we know the working point, we get also a linearized description of each objects:

$$W^{lin} = \frac{1}{2}\int\int \left.\frac{\delta^2 W}{\delta\theta(l)\delta\theta(l')}\right|_{\theta^{eq}} \delta\theta(l)\delta\theta(l')dldl'$$

✔ This is only a static description. We need dynamics. This problem can be solved adding a kinetic energy contribution. In the frequency domain this can be written as

$$T^{lin} = \int\int \frac{\omega^2}{2} M(l,l')\delta\theta(l)\delta\theta^\star(l')dldl'$$

✔ We want to write an expression which depends only on frames (boundary conditions). If we stay in the frequency domain this is simple. We can find the minimum of $T^{lin} - V^{lin}$, $\delta\theta^{lin}_{min}(l) = \delta\theta_0 f(l,\omega) + \delta\theta_L g(l,\omega)$,and substitute.

$$T^{lin} - W^{lin} = \frac{1}{2}\left(\delta\theta_0^\star \delta\theta_L^\star\right) \begin{pmatrix} L_{00}(\omega) & L_{01}(\omega) \\ L_{10(\omega)} & L_{11}(\omega) \end{pmatrix} \begin{pmatrix} \delta\theta_0 \\ \delta\theta_L \end{pmatrix}$$

We obtain our result, but L_{ij} are generally nonlinear functions of ω, and cannot be used easily in the time domain.

Low frequency approximation

We can expand last results in power of ω^2,

$$T^{lin} - W^{lin} = \frac{1}{2}\delta\vec{\theta}^{+} \cdot (-K + \omega^2 M + O(\omega^4)) \cdot \delta\vec{\theta}$$

defining our best candidate for the stiffness and mass array of the object.

- ✔ The same procedure can be applied to every mechanical objects.
- ✔ A natural classification of objects based on the number of specified (=not free) boundary conditions
- ✔ We expect accuracy at low frequencies

Now the mass,stiffness and damping array of the system can be constructed:

- ✘ Frequency domain: we solve $(-\omega^2 M - i\omega\Lambda + K)\vec{x} = \vec{f}$
- ✘ Time domain: the library provides several solvers:
  1. Standard Runge Kutta and Adams methods and st
  2. Methods based on state transition matrix

$$x(t+T) = e^{AT}\left(x(t) + \int_t^{t+T} e^{A(t-\tau)} f(\tau) d\tau\right)$$

  3. Methods for stiff systems

Longitudinal wire modes

We want an analytical solution, so we consider a straight wire. The longiudinal motion decouples:

$$T = \frac{1}{2}\int_0^L \rho S \left(\frac{dx}{dt}\right)^2 dl, \quad W = \frac{1}{2}\int_0^L ES \left(\frac{dx}{dl}\right)^2 dl$$

Neglecting T the potential energy is given simply by

$$W = \frac{1}{2}\frac{ES}{L}[x_L - x_0]^2$$

which is a simple spring. The $O(\omega^2)$can be obtained simply substituting the zero order solution in T. We get

$$T = \frac{1}{2}\frac{\rho S L}{3}[\dot{x}_0^2 + \dot{x}_0\dot{x}_L + \dot{x}_L^2]$$

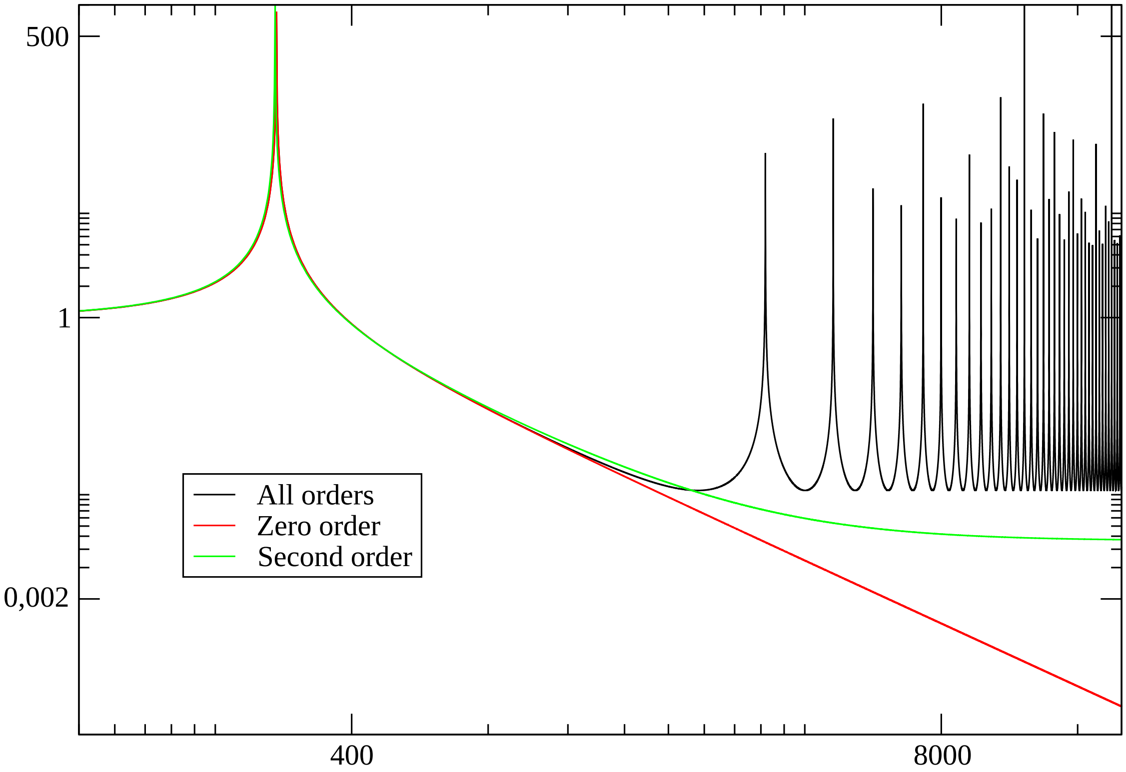
We can attach a mass and calculate the transfer function:

$$\frac{x_L}{x_0} = \frac{\frac{ES}{ML} + \omega^2\frac{\rho SL}{6M}}{\frac{ES}{ML} - \omega^2\left(1 + \frac{\rho SL}{3M}\right)}$$

A consistent mass matrix

In this case we can also calculate the exact transfer function

$$\frac{x_L}{x_0} = \frac{1}{\cos\frac{\omega^2\rho L}{E} - \frac{M}{\rho S}\sin\frac{\omega^2\rho L}{E}}$$



- ✔ The first kinetic correction can be important in many pratical cases
- ✔ It can be introduced without changing the d.o.f.

Cross coupling: the Beam case

We can apply the same procedure to a Beam.

- ✔ Two frames, so we expect a 12×12 mass, damping and stiffness array.
- ✔ In principle we must start from the appropriate potential, which is a quadratic function of the nonlinear stress tensor

$$u_{ij} = \frac{1}{2} \left(\partial_i u_j + \partial_j u_i + \partial_i u_l \partial_j u_l \right)$$

In a case of pratical importance (small bending) it is possible to use a simpler approach.

1. We know 6 eigenvectors of the stiffness array (global traslation and rotation)
2. The reduced 6×6 array factorize in 2 2×2blocks (y, $-\theta_x$and x, θ_y) and 2 1×1blocks (θ_z,z).
3. For z we can use the longitudinal equation of the wire. The equation for θ_z is similar.
4. The 2×2 blocks can be calculated starting from

$$W_{trans} = \frac{1}{2} \int_0^L \left[EI \left(\frac{d^2 y}{dl^2} \right)^2 + T \left(\frac{dy}{dl} \right)^2 \right] dl$$

Spontaneous stiffness generation...

We obtain a 2×2 reduced stiffness array which can be written as

$$K(T) = \frac{T}{L(k^2 - 2\tanh(k/2))} \begin{pmatrix} 1 & -\frac{\tanh(k/2)}{k} \\ -\frac{\tanh(k/2)}{k} & \frac{1}{k}\left(\frac{1}{\tanh k} - \frac{1}{k}\right) \end{pmatrix}$$

where $k^2 = TL^2/EI$.

- ✔ This expression depends explicitly on T, because it is the result of a linearization around a stressed state
- ✔ We can solve this problem adding the longitudinal degree of freedom

  $$W = \frac{1}{2} q^T K\left(\frac{ES}{L}(z-L)\right) q + \frac{1}{2}\frac{ES}{L}(z-L)^2$$

  Note that this potential is nonlinear. But we know that

  $$ES(z_{eq} - L)/L \simeq T$$

  so the beam “get stiffnes” at the working point and we obtain an accurate linear approximation.
- ✔ The mass matrix can be evaluated without problems
- ✔ The general case is much more complicated. All the 6 independent degrees of freedom can be coupled together.

Internal modes

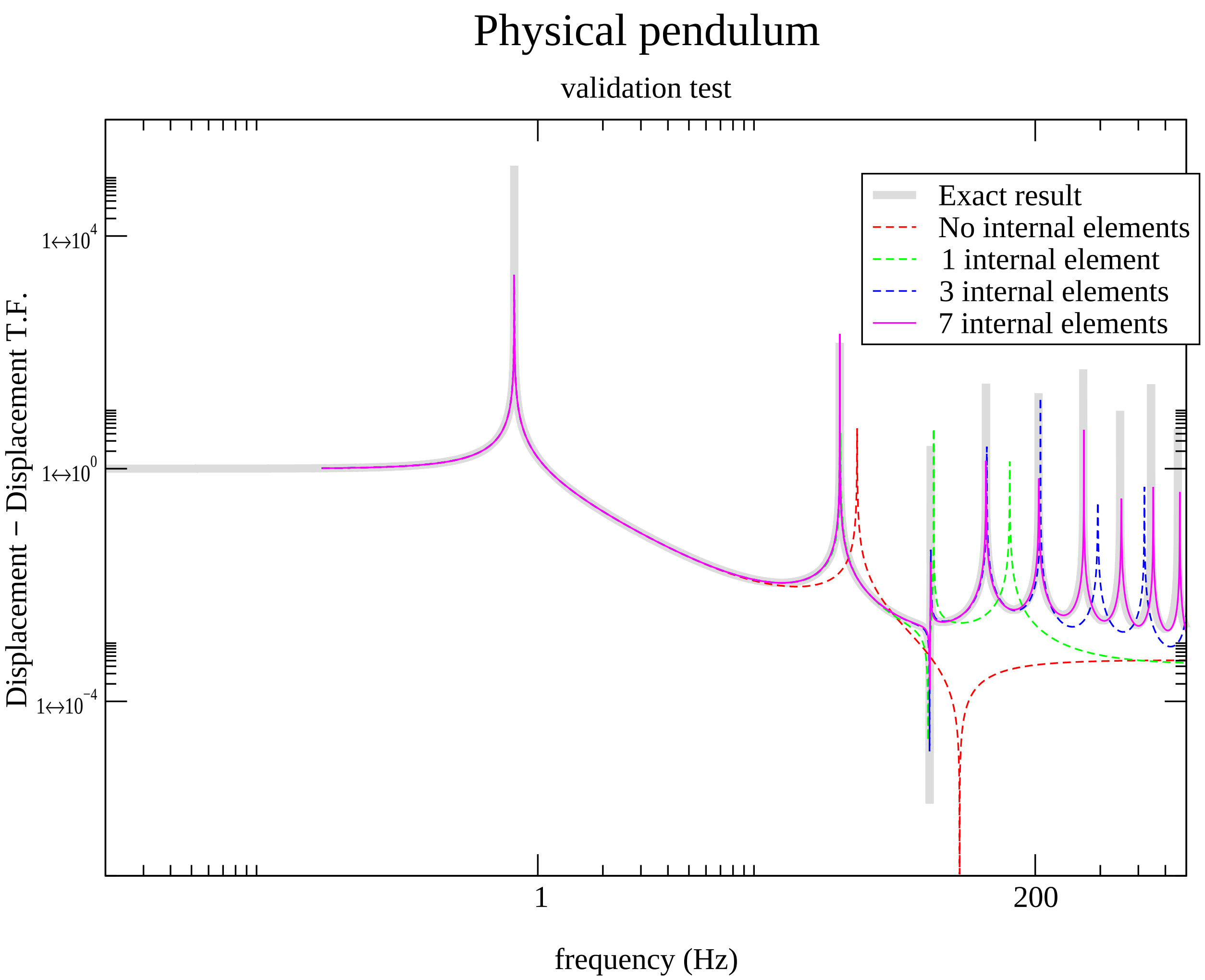
How to introduce internal modes, which are relevant in the high frequency region? There are many ways, the general idea is to add variables.

- ✔ The object (say, the wire) is seen as a collection of smallest ones. To each of them the low frequency approximation is applied.
- ✔ There are alternative approach, for example the finite element method
- ✔ In the MSE there is not a prescribed method to represent internal modes, it depends on the particular class implementation. The final result is always an “enlarged” stiffness (mass, damping) matrix of the following form

$$\begin{pmatrix} K_{00} & K_{0I} & 0 \\ K_{I0} & K_{II} & K_{I1} \\ 0 & K_{1I} & K_{11} \end{pmatrix}$$

  where the $0, 1$ indexes labels the frame variables, and I the internal,hidden ones.
- ✔ In this way we can avoid a complete three-dimensional representation of the internal modes when it is irrelevant (say, the torsional modes of a wire).

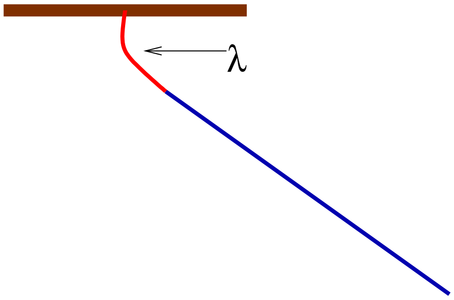
Internal modes: example



Rigid body suspended to a thick wire:

- Wire internal modes only
- Good convergence with a small number of elements
- Better tha usual Finite Element approach

Singular perturbative problems



In the MSE the level of discretization of each mechanical object can be specified.

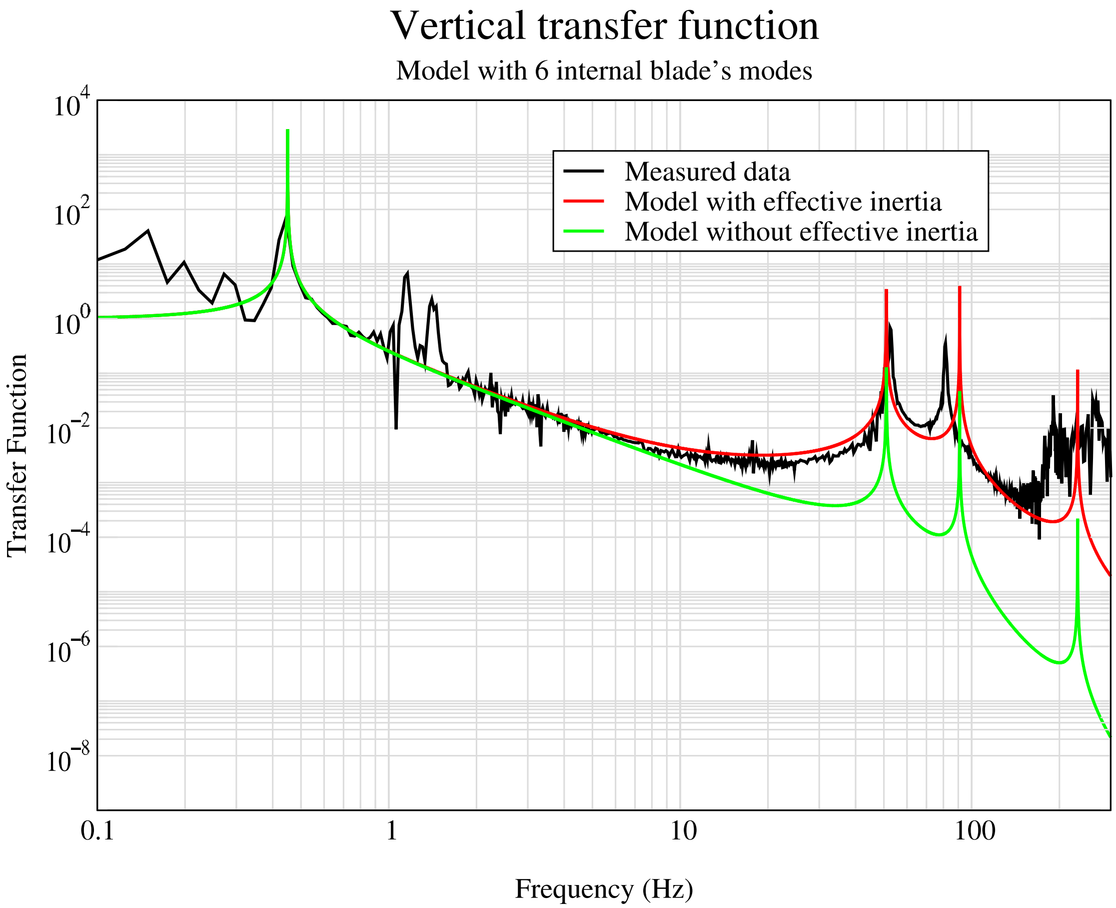
- ✔ Many elements give us many and more accurate internal modes. However the computational cost grows rapidly.
- ✔ In some cases it pays to use a more accurate procedure. Coming back to the beam equation, we can write

$$\frac{EI}{T}y'''' - y'' = 0$$

- ✔ Now suppose that $\lambda = \sqrt{EI/T}$ is small (as for a tensioned wire). The solution of $y'' = 0$ is not an accurate approximation around the suspension point.
- ✔ We have to adapt the discretization to this problem, in order to avoid an excessive number of elements, or inaccuracies.
- ✔ Also in this case the model performs better that Finite Element ones

An example: GAS filter

One of the first results:



- ✔ Single GAS filter with internal modes for the blade
- ✔ The addiction of the first kinetic correction give to the model a big improvement
- ✔ With 6 internal modes the first two internal resonances are in good agreement.

MSE integration in E2E

- ✔ The library can be used (and tested) independently, but its main purpose is the integration in the Ligo E2E model.
- ✔ The first stage is the construction of separate models, hard-wired to the E2E environment.
- ✔ In this way it will be possible to test the interaction between optics and mechanics
- ✔ In a future the library will be fully integrated. It will be possible to construct a mechanical system with the E2E GUI.

Actually the release 0.3 of MSE is avaliable.

1. It contains some basic mechanical objects that will be specialized in the near future
2. We are making extensive testing and debugging.
3. In a short time new classes for the description of composite system (a Ligo stack, a Geo triple pendulum, a GAS filter, an inverted pendum) will be implemented and tested

Future developments

1. **Thermal noise generation. This will be possible setting the temperature of the mechanical system. There are at least two possible approaches:**
   - (a) **the addiction of stochastic forces of Langevin type [Nearly finished]**
   - (b) **the use fluctuation-dissipation theorem to evaluate the relevant noise spectral densities.**
2. **Accurate modeling of structural damping. This will be done in two way:**
   - (a) **using internal, hidden variables**
   - (b) **projecting a viscous-type damping on the resonances [Nearly finished]**
3. **Internal modes for massive bodies (mirror)**
4. **Adaptive meshing facilities**
5. **A “black box” class that will be used to introduce experimental data (say, a mechanical object with known transfer functions).**