Automatic differentiation techniques

Simulation of massively controlled space telescopes

Mauro Manetti, Pierangelo Masarati,
Marco Morandini and Paolo Mantegazza

Dipartimento di Ingegneria Aerospaziale – Politecnico di Milano

Inside ANTASME

1 (C)	LOM	Dipartimento di Ingegneria Aerospaziale, Politecnico di Milano	DIA
2	LOM	Dipartimento di Elettronica e Informazione, Politecnico di Milano	DEI
3	LOM	University of Bergamo	UNIBG
4	LOM	Carlo Gavazzi Space SpA	CGS
5	LOM	A.D.S. International	ADS
6	NBR	University of Eindhoven	TUE
7	CAT	CENTRE INTERNACIONAL DE MÈTODES NUMÈRICS EN ENGINYERIA	CIMNE
8	CAT	COMPASS INGENIERÍA Y SISTEMAS	COMPASS
9	CAT	QUANTECH ATZ	QUANTECH

WP number	Workpackage title	Start month	End month	Leading[1] participant n.
1	Project management	1	13	1
2	Automatic differentiation techniques	1	9	1
3	Innovative finite element methods for unsteady aeroelastic analysis	1	12	7
4	Advanced evolutionary algorithms for transonic drag reduction and high lift of 3D configuration using unstructured FEM	1	12	7
5	Simulation of massively controlled space telescopes	1	12	1
6	Object-oriented modelling of mechatronic electrohydraulic systems	1	12	2
7	Object-oriented modelling of spacecraft dynamics	1	12	2
8	SINDA/FLUINT simulation of the LHP prototype	1	12	3
9	Lumped and "distributed" analysis of the by-pass valve	3	11	3
10	Dissemination of results	1	12	1

WP2 - Automatic differentiation techinques

Implicit code:

- Set of nonlinear equations $\mathbf{f}(\mathbf{x}) = \mathbf{0}$
- Jacobian matrix $\mathbf{J} = \partial\mathbf{f} / \partial\mathbf{x}$ (cumbersome)

Rapid elements prototyping

- Code $\mathbf{f}(\mathbf{x}) = \mathbf{0}$
- Let the code compute $\mathbf{J}$

MBDyn:

- Multibody code

  http://www.aero.polimi.it/~mbdyn
- GPL
- C++

WP2 - Automatic differentiation techinques

Automatic differentiation techniques:

- Analysis of source code (Fortran/C)
- Tape of operations (operator overloading, C++)

Requirements:

- Instrumentation of the code (operations/loops/conditions)
- Use of custom data type
- Template (C++)

Result:

- Source code analysis:

  compiled **J**

- Callable subroutine:

  run-time **J**

WP2 - Survey of available libraries

Tool	Language	License	Source code	Method
ADF95	Fortran77/Fortran95	Custom non.profit	Yes $	Code analysis?
ADIC	C/C++	?	No	
ADIFOR	Fortran77	Custom non-profit	Yes	Code analysis?
AdiMat	MATLAB	To be defined	No	
ADOL-C	C/C++	CPL	Yes	Tape
AUTO_DERIV	Fortran77/Fortran95	Custom non.profit	Yes $	Code analysis?
CppAD	C/C++	CPL/GPL	Yes	Tape
FAD	C/C++	Teaching, non profit	Yes	Tape?
FADBAD/TADIFF	C/C++	Custom non.profit	Yes	Tape
GRESS	Fortran77	?	Yes	?
OpenAD	Fortran77/Fortran95	?	No?(EDG front end)	Code analysis
TAMC	Fortran77	Non-profit	No	Code analysis?
TAPENADE	Fortran77/Fortran95	?	No	Code analysis?
TaylUR	Fortran95	Custom non.profit	Yes $	Code analysis?
TOMLAB/MAD	MATLAB	?	$$	Tape

WP2 – Coding effort

- Once for all: enable the code

```
double Mat3x3::dDet(void) const
{
   double* p = (double*)pdMat;

   return p[M11]*(p[M22]*p[M33]-p[M23]*p[M32])
     +p[M12]*(p[M23]*p[M31]-p[M21]*p[M33])
     +p[M13]*(p[M21]*p[M32]-p[M22]*p[M31]);
}
```

```
template<class T>
T Mat3x3T<T>::dDet(void) const
{
   T* p = (T*)pdMat;

   return p[M11]*(p[M22]*p[M33]-p[M23]*p[M32])
     +p[M12]*(p[M23]*p[M31]-p[M21]*p[M33])
     +p[M13]*(p[M21]*p[M32]-p[M22]*p[M31]);
}

typedef Mat3x3T<double> Mat3x3
```

- Element level: **J** (requires **f**)

```
std::vector<T> y_dep(12);                        //declare dep variables
CppAD::Independent(x_indep);                     //declare indep variables
res_vec(p, x_indep, y_dep, pEl->GetLabel());     //compute f(x)
CppAD::ADFun<doublereal> f(x_indep, y_dep);

.....

J = f.Jacobian(xx);                              //Compute J(x)
```

WP2 - Status and future work

Integration almost finished

- Templatize MBDyn: double vs. CppAD<double>
- Elements

Issues

- Rotation:

    SO(3) -> nonlinear **x**/body orientation

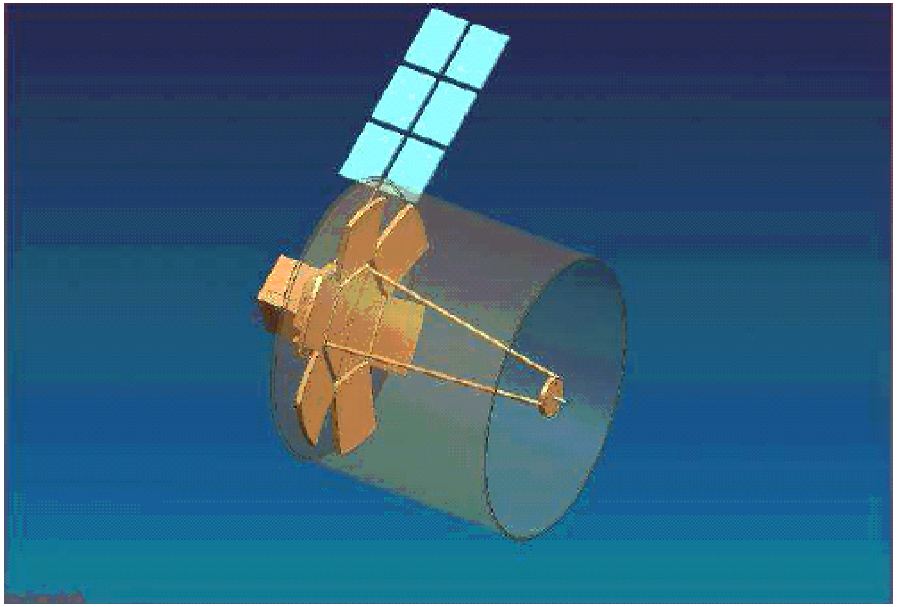
    handled outside elements

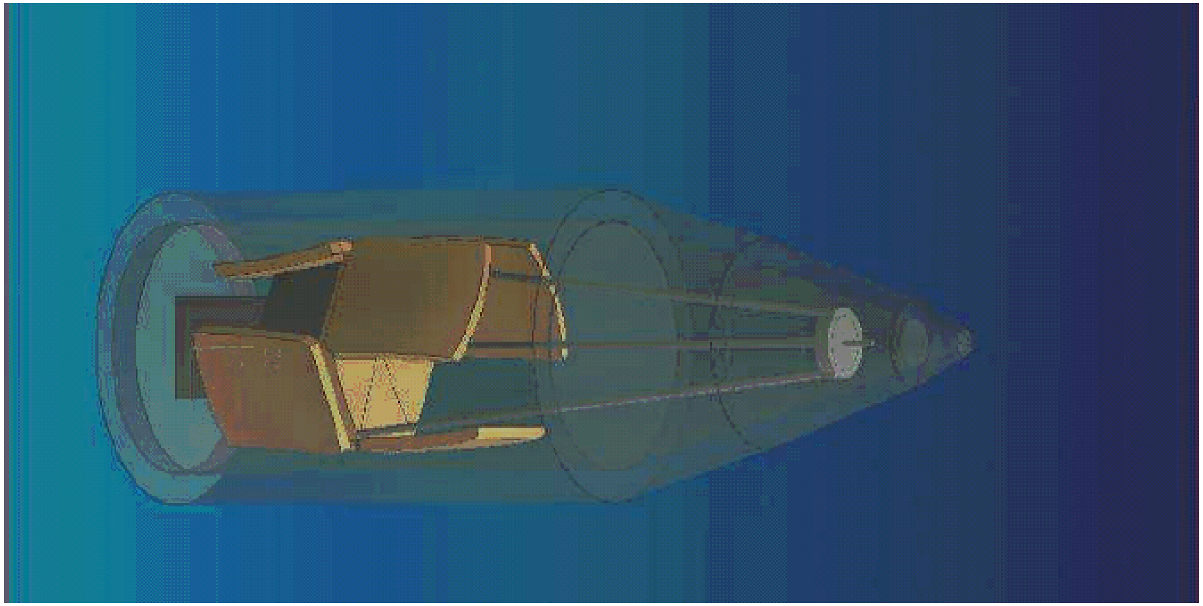
Automatic elements

- Wheel
- Complex joint (gimball?)

Accuracy and timing tests

WP5 - Simulation of massively controlled space telescopes

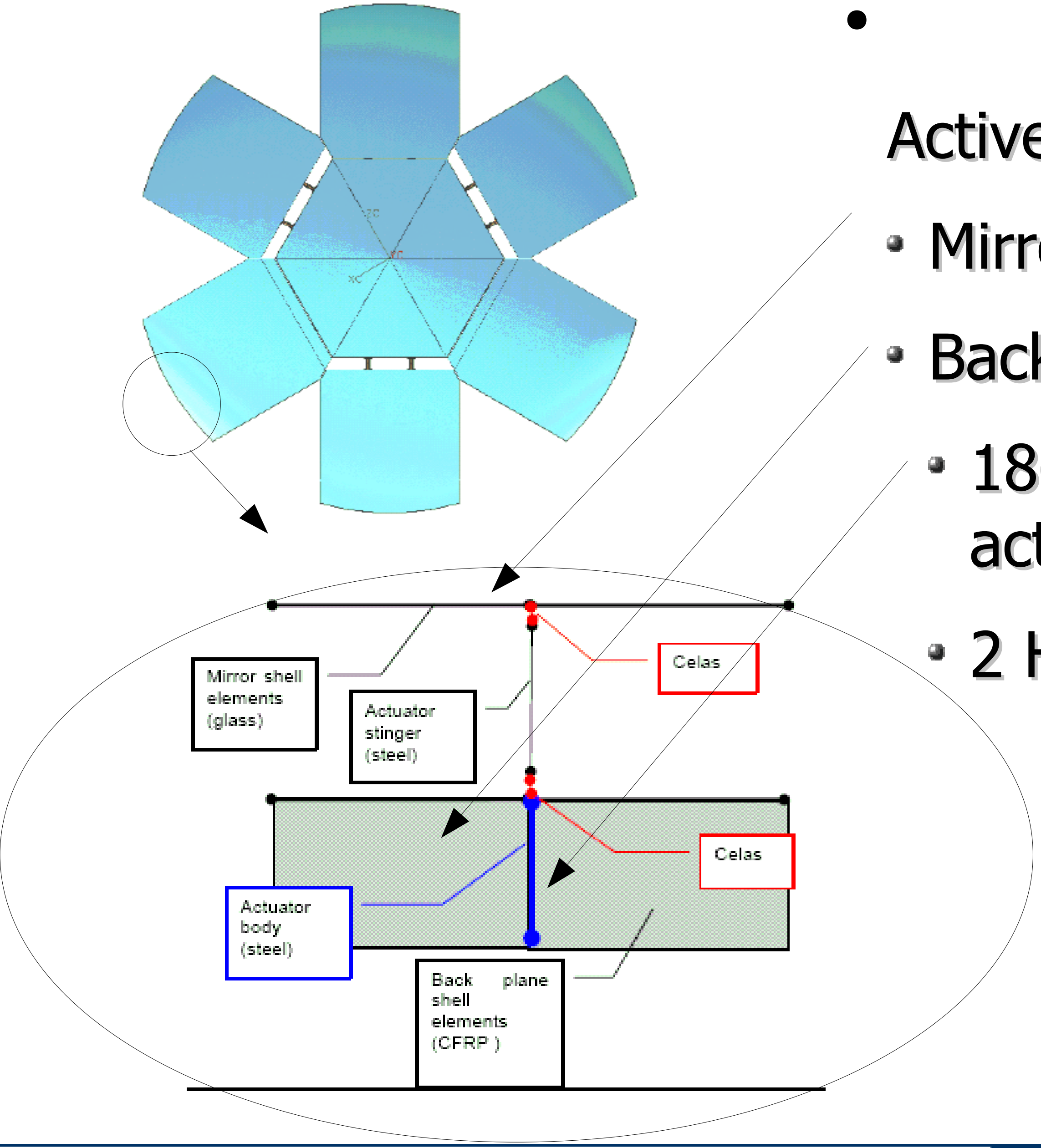




- Carlo Gavazzi Space SpA
  - Structural design
- A.D.S. International
  - Active mirror actuators
- DIA
  - Mirror deployment simulation
  - Active mirror control

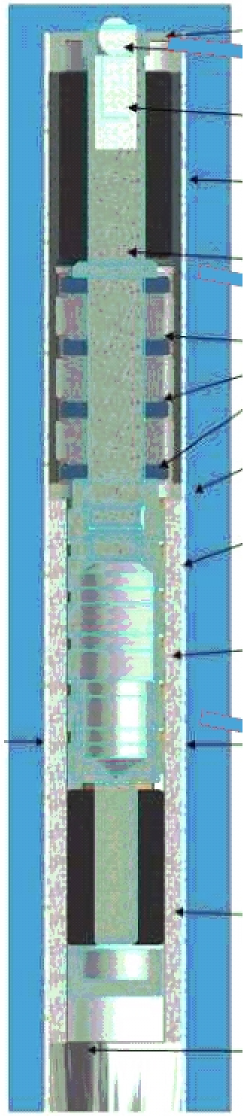
Images credit Carlo Gavazzi Space

WP5 – space telescope active control


Mirror shell elements (glass)
Actuator stinger (steel)
Celas
Celas
Actuator body (steel)
Back plane shell elements (CFRP)

Active mirror

- Mirror
- Back-plate
  - 186 co-located actuator-sensor pairs
  - 2 Hz



Images credit Carlo Gavazzi Space

WP5 – space telescope active control

- Background:
  - MMT active secondary mirror
    - Feed-forward
    - Decentralized PID2

- Simulation code:
  - Modal dynamic
  - Off-line mirror stiffness identification
  - Gain optimization

-
  - Rigid body movement?
  - Non-controllable mirror flapping?

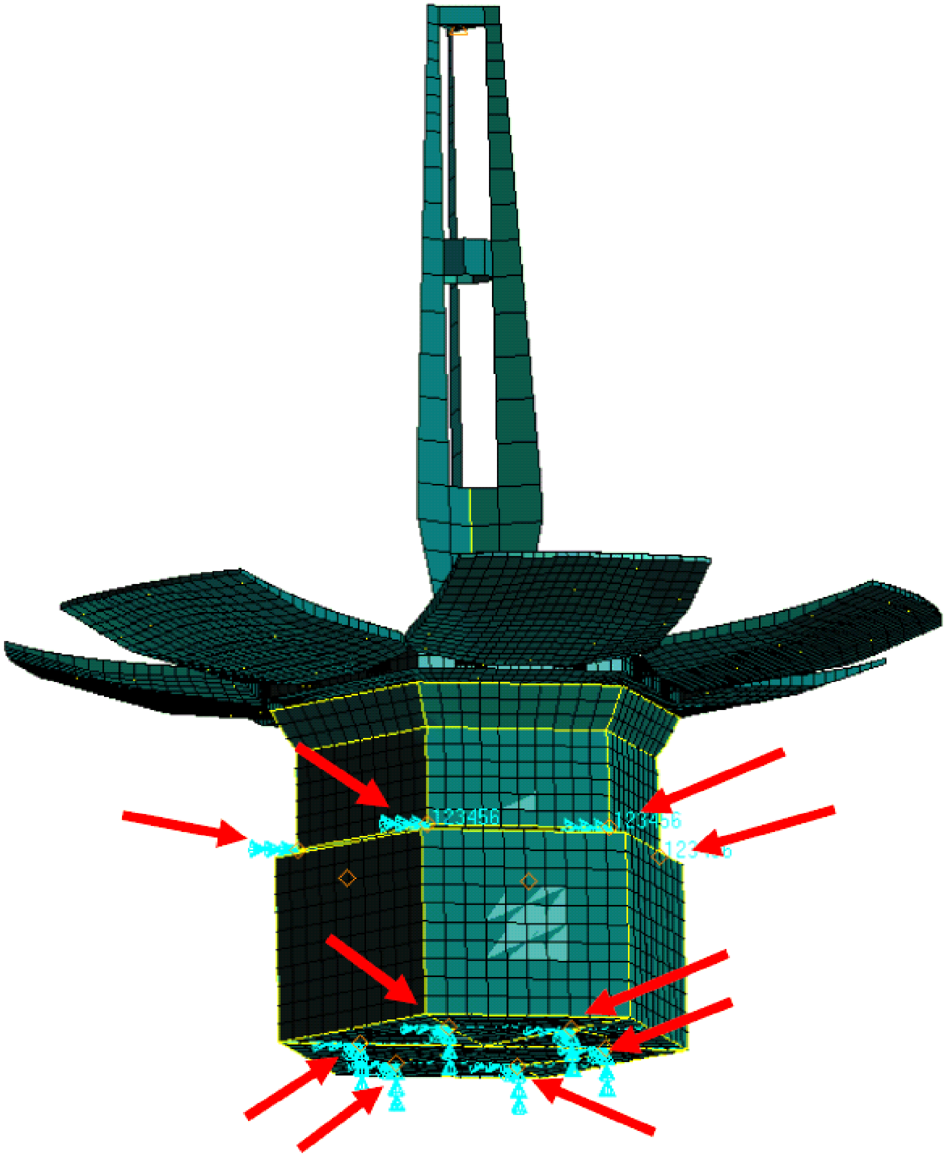
WP5 – space telescope feed-forward concept

-
    - Static feed forward: 2 Hz
    - Computed recursively

      $\Delta \mathbf{f} = \mathbf{K}(\mathbf{x}^{k+1} - \mathbf{x}^{k}_{avg})$
    - Robust vs. **K**
    - Requires: estimate of **K**
        - Identification
    - Stability: PID2 (500 Hz)

WP5 – space telescope work

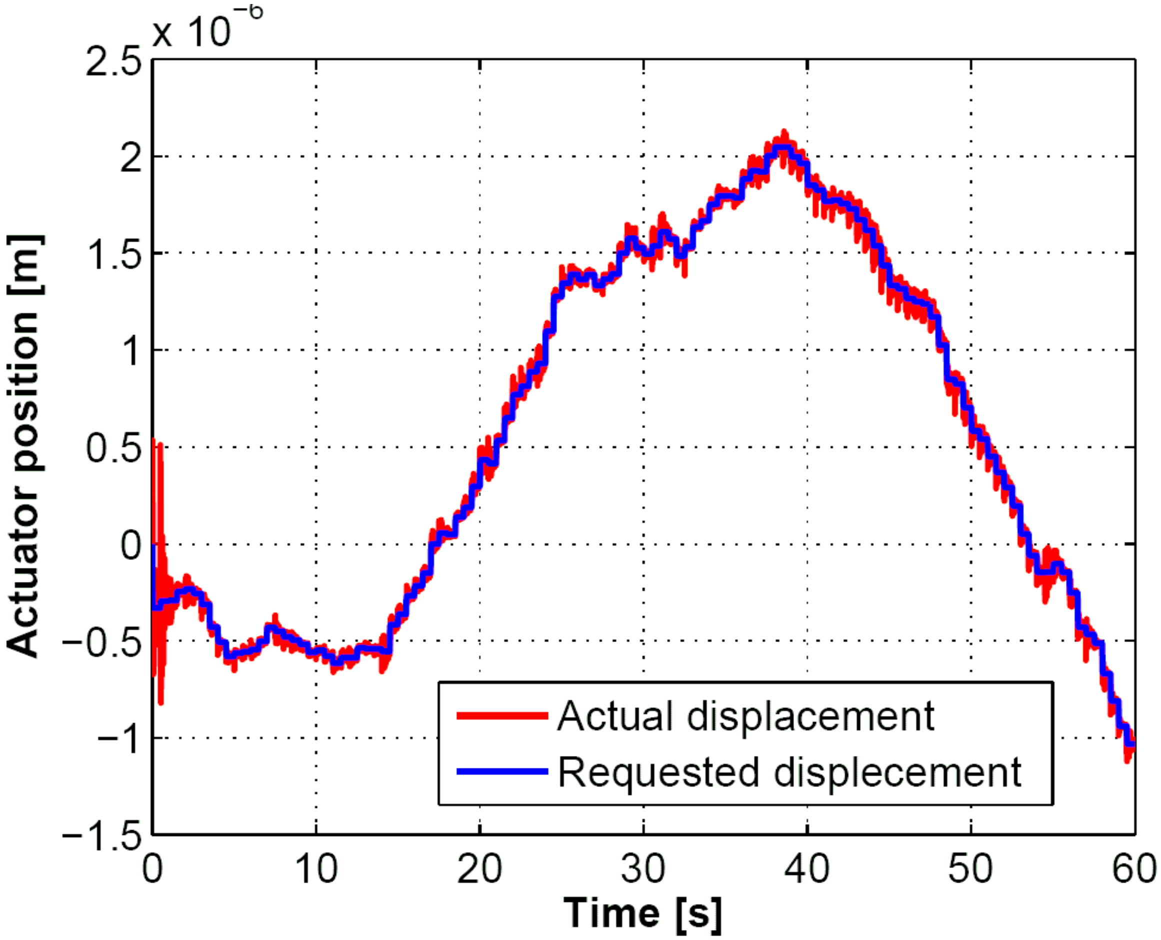
- Done:
-
  - Modified preliminary FEM model (from CGS)
  - Use actuators response function (from A.D.S.)
  - Data extraction
    - **K** identification



Images credit Carlo Gavazzi Space

WP5 – space telescope work

- Ongoing:
  - Gain optimization
- ToDo:
  - Improve actuator model
  - Attitude control
  - Attitude and deformable mirror controls
  - Simulation of mirror deployment
  - Disturbance rejection



WP9 - Dissemination

http://www.aero.polimi.it/Antasme

Send us your

- Presentations
- Documents
- WPs