COCOA

Cms Object-Oriented Code for Optical Alignment

Pedro Arce (CIEMAT), Michael Case (U. California Davis)

CHEP 2006, Mumbai, 13-17th February 2006

Outline

- Optical alignment
  - The CMS optical alignment system
  - The problem of optical alignment and how to solve it
  - COCOA
- Effect of misalignment in the reconstruction
- Alignment with tracks
- Commonalities and difference of the several alignment softwares

The problem

SIMULATION:

You are designing a new optical alignment system and you want to:

- **Propagate errors:**
  - Calculate how much the errors of the calibrated parameters and the measurements affect the errors of the parameters we want to measure
- **Study redundancies:**
  - How much errors changes if some measurement disappears
- **Study range:**
  - When a measurement will get out of the range of the measuring device is some objects move

The problem (II)

RECONSTRUCTION:

• Optical system takes measurements (2D sensors, 1D sensors, tiltmeters, distancemeters)

$\Rightarrow$ results are not what expected by extrapolating measured and calibrated parameters. **Why?**

- Wrong rotation / position of some objects
- Wrong internal calibration of some objects
    - wedge of a splitter
    - internal calibration of a distancemeter
    - deviation when traversing a sensor
    - ...

- **Software is the same for Simulation and Reconstruction**
    - Only difference: for Simulation measurement is ideal, for Reconstruction measurement is real

How to solve it

- Get the equations of how each measurement depends on all those parameters
  - positions, rotations, internal parameters

$M_1 = f_1(p_1, p_2,\ldots, p_m)$

$M_2 = f_2(p_1, p_2,\ldots, p_m)$ $\quad$ *$M_1,\ldots, M_n$ = Measurments*

… $\quad$ *$p_1,\ldots,p_m$ = parameters (known and unknown)*

$M_n = f_n(p_1, p_2,\ldots, p_m)$ $\quad$ f_i are non linear equations

- You know the measurements and some calibrated parameters, need to know the missing ones

$\Rightarrow$ Solve the system of equations: **Non-linear least squares fit**

- **To solve a system of equations, you do not have to know the equations**

How to solve it (II)

- **Only the derivatives are needed**

⇒ Get the derivatives with a numerical method

- Reproduce a measurement with initial parameters (e.g. propagate a laser until the sensor)
- Move a parameter and see how the measurement value changes
- Repeat n times moving $1/2^i$, until it converges

Total CMS alignment system: 30000 parameters ⇒ **big and sparse matrices ⇒ sparse matrix library (meschach C library)**

COCOA

♦ General purpose software to simulate and reconstruct optical alignment systems composed of any combination of

laser, x-hair laser, source, lens, pinhole, mirror, plate splitter, cube splitter, rhomboid prism, optical square, sensor2D, sensor1D, COPS, distancemeter, distance target, tiltmeter, 'user defined'

• Each object has internal parameters (planarity of a mirror, wedge between plates of a plate splitter, internal calibration of COPS...)

• User can define its own object: *Tell COCOA how much the light ray will be shifted and deviated for each measurement*

➢ Reconstructs positions, angles or internal parameters of each object from the measurement values

➢ Propagates the errors of the measurements and calibrations (including correlations)

COCOA

COCOA

- ♦ System description in simple files
  - Simple text files
  - XML (CMS format)
  - Tool available to trasalate automatically ASCII-XML
- ♦ Interface with DAQ measurements
  - Currently from text files
  - Soon from POOL-ORACLE database
- ♦ Interactive 3D view
  - VRML (Virtual Reality Modeling Language)
  - IGUANA visualisation of XML files

- Scan of a parameter and analysis of results

- Randomization of a parameter

COCOA

- **Documentation:**
  - Primer
  - User’s Guide
  - Advanced User’s Guide
  - Two examples explained with detail
  - dOxygen documentation

COCOA

- **How it works:**
  - Describe the system in an input ASCII file
  - Select which parameters are unknown and which are known
  - For the known one write the values
  - Input the measurements

  ➢ software provides **best values for unknown parameters** (positions/rotations/internal parameters) compatible with measurements and **propagate the errors** from the measurements and the known parameters

An example input file

```
// system composed of one laser, one periscope that
holds a plate splitter and a mirror and two 2D
sensors.

GLOBAL_OPTIONS
report_verbose 2
save_matrices 0
length_error_dimension 2
angle_error_dimension 2

PARAMETERS
pos_laser 0
posZ_periscope 1
posZ_sensor 1.1
err_pos 100
err_ang 100
prec_sens2D 5

SYSTEM_TREE_DESCRIPTION
object system laser periscope 2 sensor2D
object periscope plate_splitter mirror
```

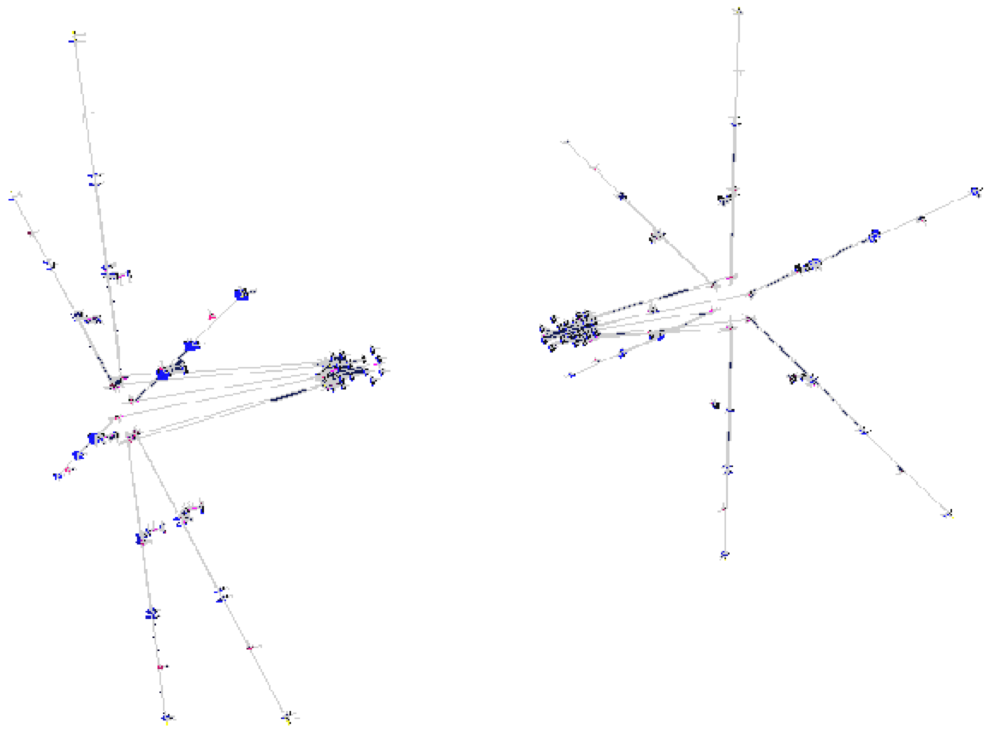
```
SYSTEM_TREE_DATA
system s
 laser laser // this is the laser
  centre
   X  pos_laser 1000  unk
   Y  pos_laser 1000  unk
   Z  pos_laser 0.    fix
  angles
   X   0  err_ang  unk
   Y   0  err_ang  unk
   Z   0  err_ang  cal
 periscope peri
  centre
   X   0  err_pos  cal
   Y   0.25  err_pos  cal
   Z posZ_periscope err_pos  cal
  angles
   X   0   err_ang  cal
   Y   0   err_ang  cal
   Z   0   err_ang  cal
```

```
plate_splitter spli
    ENTRY {
      length shiftX 0. 0. fix
      length shiftY 10. 0. fix
      angle wedgeX 0.0001 10 cal
      angle wedgeY 0.0001 10 cal
}
    centre
      X   0   err_pos  cal
      Y -0.25 err_pos  cal
      Z   0.  0.       cal
    angles
      X   0   err_ang  cal
      Y   0   err_ang  cal
      Z   0   err_ang  cal
  mirror mirr
    ENTRY {
      none planarity 0.1 0. cal
     }
    centre
      X   0   err_pos  cal
      Y 0.25  err_pos  cal
      Z   0.  err_pos  cal
    angles
      X   0  err_ang  cal
      Y   0  err_ang  cal
      Z   0  err_ang  cal
```

```
// now the two sensors
  sensor2D sens1
    centre
      X   0  err_pos  cal
      Y   0  err_pos  cal
      Z posZ_sensor err_pos  cal
    angles
      X   0  err_ang  cal
      Y   0  err_ang  cal
      Z   0  err_ang  cal
  sensor2D sens2
   centre
      X   0  err_pos  cal
      Y  0.5 err_pos  cal
      Z   0  err_pos  cal
    angles
      X   0  err_ang  cal
      Y   0  err_ang  cal
      Z   0  err_ang  cal

MEASUREMENTS
SENSOR2D
  s/laser & s/peri/spli:T & s/sens1
  H  0.1   prec_sens2D
  V -0.1   prec_sens2D
SENSOR2D
  s/laser & s/peri/spli:D & s/peri/mirr & s/sens2
  H  0.2   prec_sens2D
  V -0.1   prec_sens2D
```

Use of COCOA



- Several test benches

- Several design studies

- Simulation of full CMS Link alignment system (3000 parameters)

- Simulation of full CMS Muon Endcap align. system (6500 parameters)

- Reconstruction of ISR test: one full CMS muon alignment halfplane (3000 parameters)

- To be used for the reconstruction of displacements at the CMS Magnet Test (May 2006)

The CMS optical alignment system

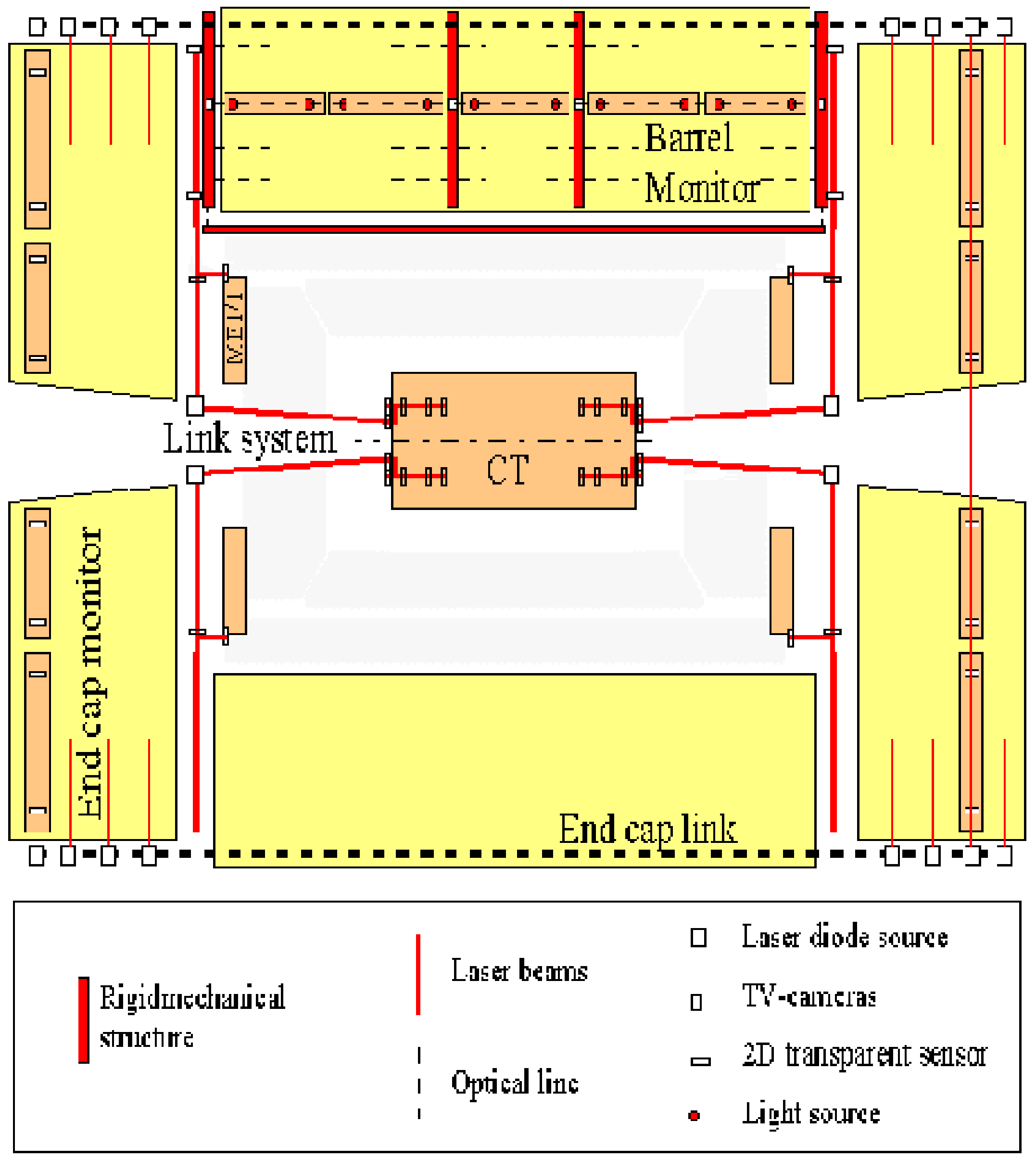
COCOA

Muon Chambers suffer movements and deformations from magnetic field, gravity and temperature
(≈ several mm)

⇒ Monitor Muon Chambers position relatively among them and with respect to the central Tracker
(≈ 150 µm)

***4 subsystems* with quite different hardware**

- Align. internal Muon Barrel
- Align. internal Muon Endcap
- Align. Muon ⇔ Tracker
- Align. Internal Tracker

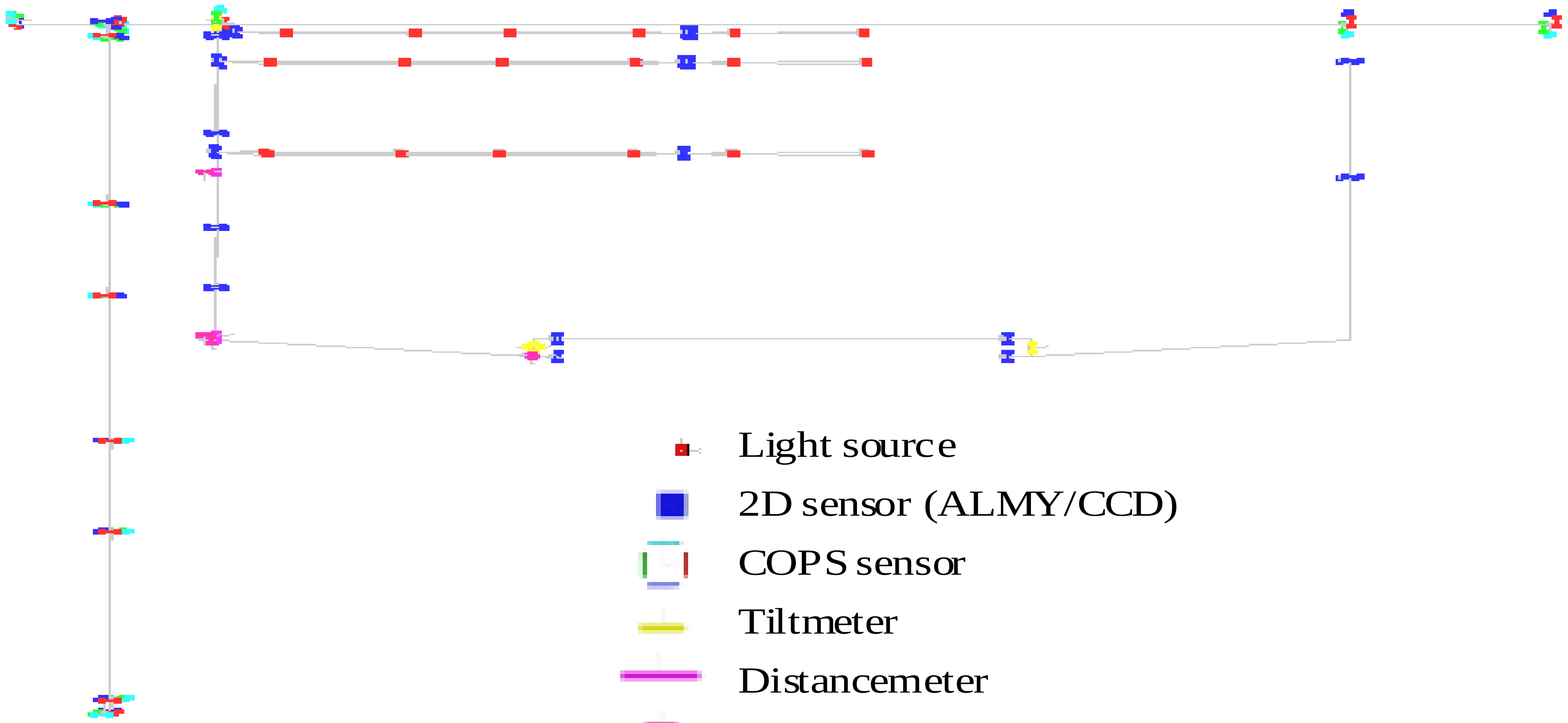


Reconstruction of ISR test

- ‘Proof of concept’ test of CMS alignment system: one full half-plane
  - Barrel
    - 18 forks (4 light sources each)
    - 3 double cameras
    - 3 single cameras on MAB+z
    - 120 measurements
  - Endcap
    - 2 x-hair lasers
    - 7 COPS
    - transfer plate with 2 COPS
    - 1 COPS on MAB +Z
    - 1 COPS on fake MA -Z
    - 47 measurements
  - Link
    - 2 laserboxes
    - laser level
    - 10 2D sensors
    - 2 tubes
    - 4 distancemeters
    - 4 tiltmeters
    - 312 measurements

- Input object parameters from calibrations
- Input object positions from survey
- Input measurements collected during August and September 2003

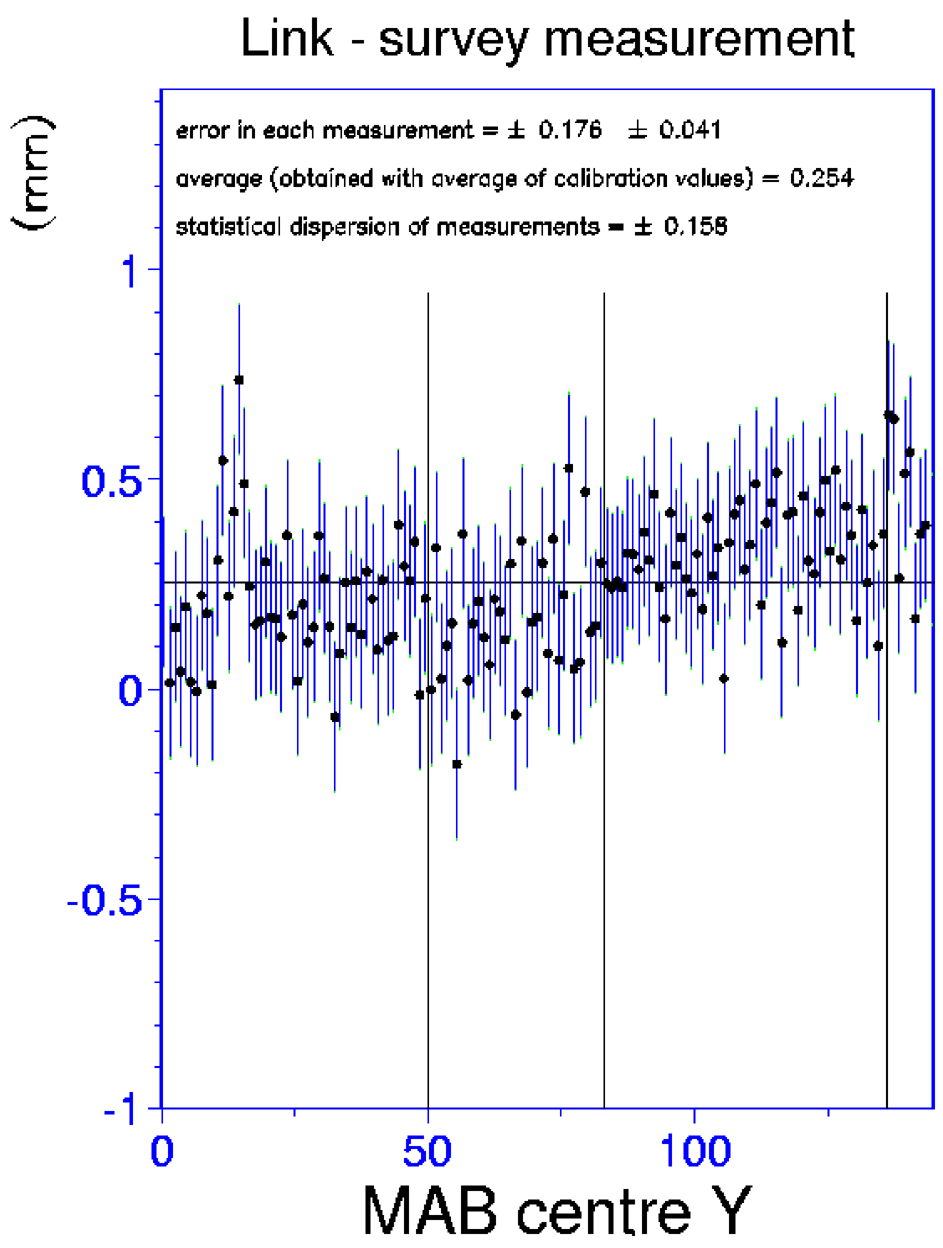
Full ISR setup in COCOA
(interactive 3D VRML view)



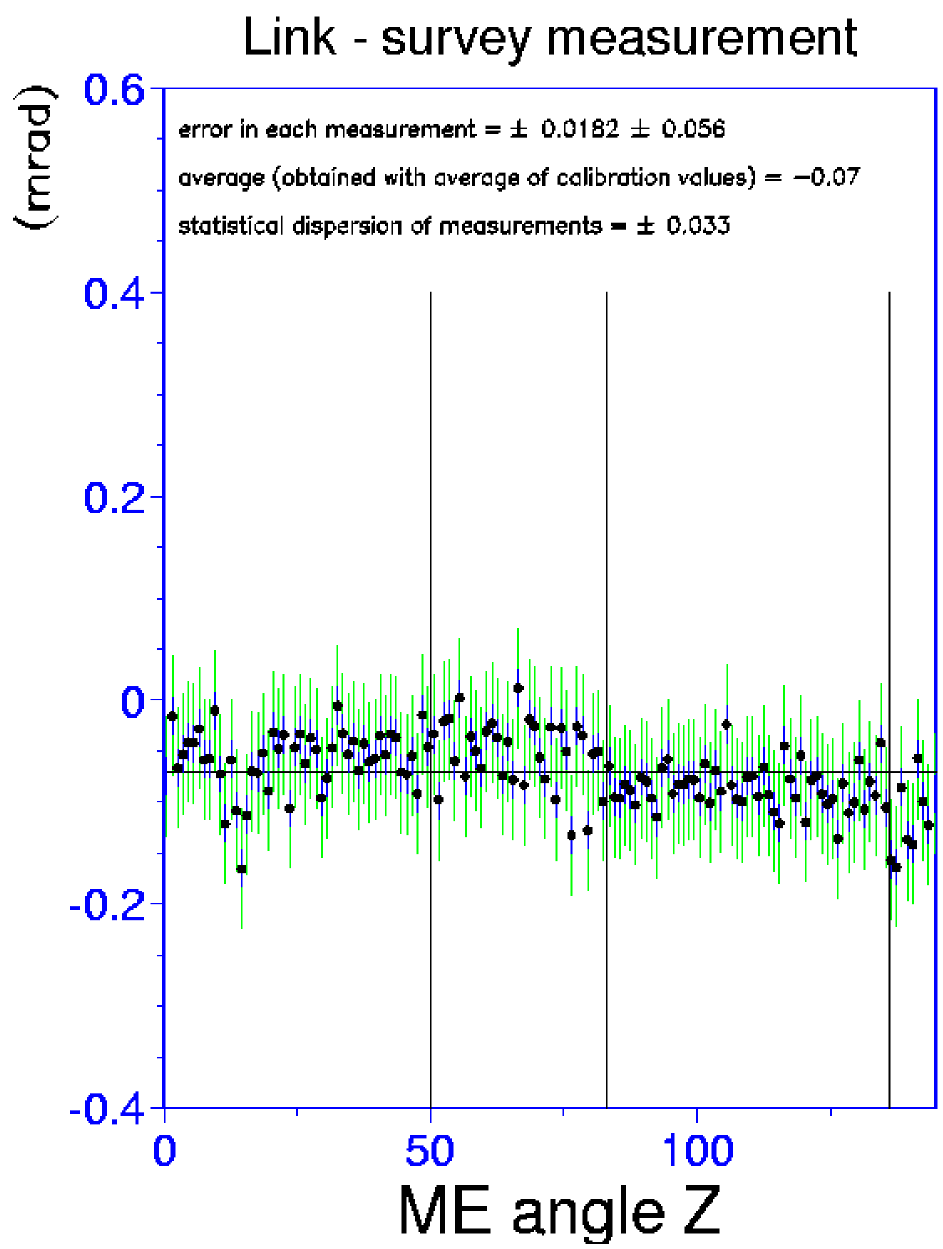
🟥	Light source
🟦	2D sensor (ALMY/CCD)
▢	COPS sensor
▬	Tiltmeter
▬	Distancemeter
🟥	Distance target

06/06/01

August



September



COCOA

Time and memory consumption

COCOA

Full CMS Link alignment system (2865 parameters):

- **20 minutes** in Athlon 1.3 GHz
- Memory: **590 Mb**
  - Due to the size of matrices (long double precision)
- Time and memory scales as ~(#param)^2 !

$\Rightarrow$ we cannot simulate full CMS (40k params)

- **Split the system in N parts**
  - Time and memory will diminish by a factor N
  - COCOA gives you the correlations and you can input the correlations for the next job
- **Diminish the number of parameters**
  - Many parameters have a negligible effect in the final result
  - Needs a thorough testing to avoid biasing
- **Try other sparse matrix libraries**
  - Many parameters have a negligible effect in the final result

Summary

- Optical alignment systems are used in several experiments
- Need a flexible software to simulate and reconstruct:
  - Design ideas
  - Design prototypes
  - Test benches
  - Full system

- COCOA is a general purpose alignment software developed as a Software Engineering project
  - User just describes its system in ASCII files
  - COCOA reconstructs the unknown parameters and propagate the errors

- COCOA soundness has been stressed during several years of use in CMS