Model Parameter Identification & Control Parameter Optimisation in EicasLab™

EicasLab™ Suite
SIM
POST
SIMBUILDER
RTAI
Linux RTAI Code Generation
AAG
Automatic Algorithm Generation
Identification & Optimization Tools
MPI
CPO
Automatic Code Generation
ACG

Welcome to Innovation

TABLE OF CONTENT

- Model Parameter Identification (MPI) and

  Control Parameter Optimisation (CPO) concept

- How to configure MPI/CPO

- The MPI/CPO process

Model Parameter Identification (MPI) & Control Parameter Optimisation (CPO) concept

The MPI/CPO concept

Model Parameter Identification and ***Control Parameter Optimisation*** are two important tasks to be addressed during the control algorithm development:

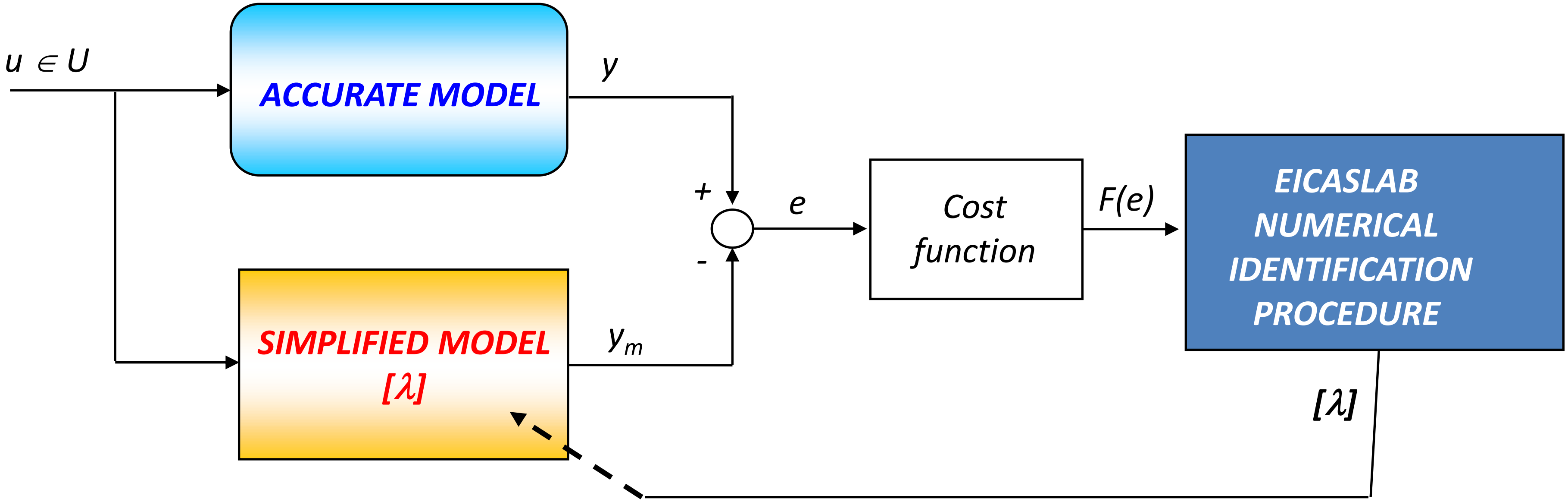
- the *Model Parameter Identification* (**MPI**) is specifically devoted to identify the parameters of the "plant simplified model" which is the base for the design of the "model based" controls.
- the *Control Parameter Optimisation* (**CPO**) is specifically devoted to optimise the control algorithm parameters in order to obtain the required performance: it allows to perform the control parameter tuning.

EICASLAB offers a professional support and advanced features
to address the overall
Model Parameter Identification & ***Control Parameter Optimisation*** process.

The Model Parameter Identification in EICASLAB

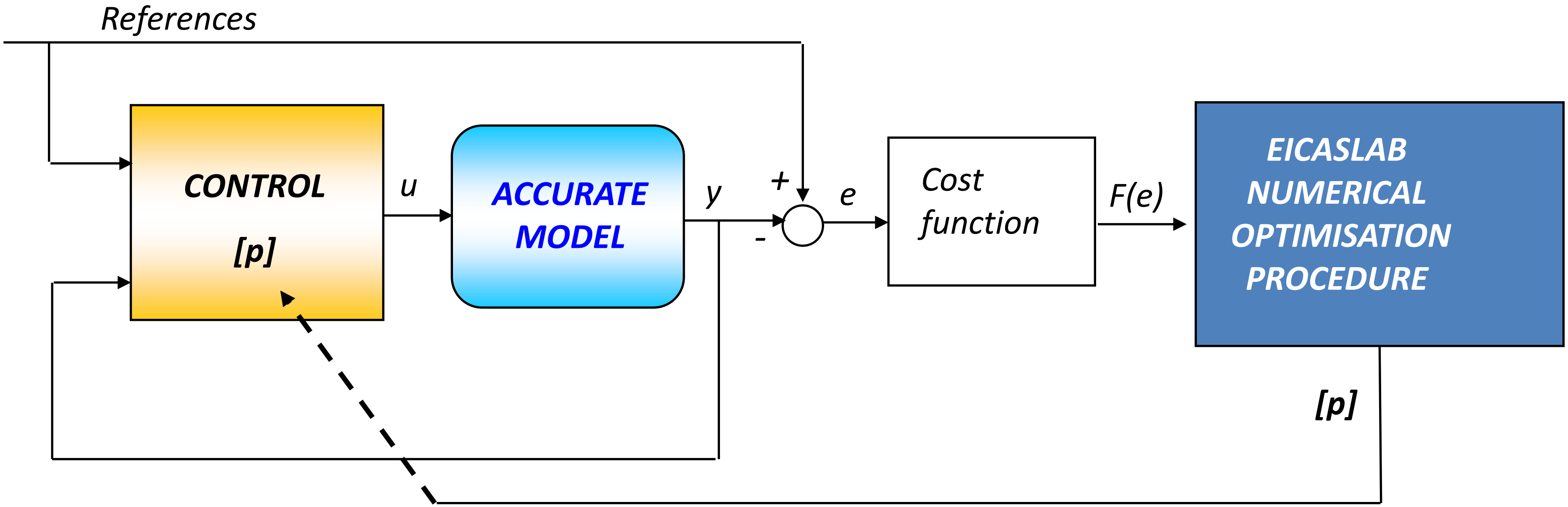
The parameter "true" value does not exist: the model is an approximate description of the plant and the parameter "best" values depend on a *cost function* adopted to evaluate the difference between model and plant.

An **original identification method** is available in EICASLAB, oriented to estimate the best values of the Plant simplified model parameters from the **point of view of the control design** starting from experimental data acquired on field or from an accurate Plant model (more detailed and sophisticated than the simplified one).


$u \in U$
ACCURATE MODEL
y
+
-
e
Cost function
$F(e)$
EICASLAB NUMERICAL IDENTIFICATION PROCEDURE
SIMPLIFIED MODEL $[\lambda]$
y_m
$[\lambda]$

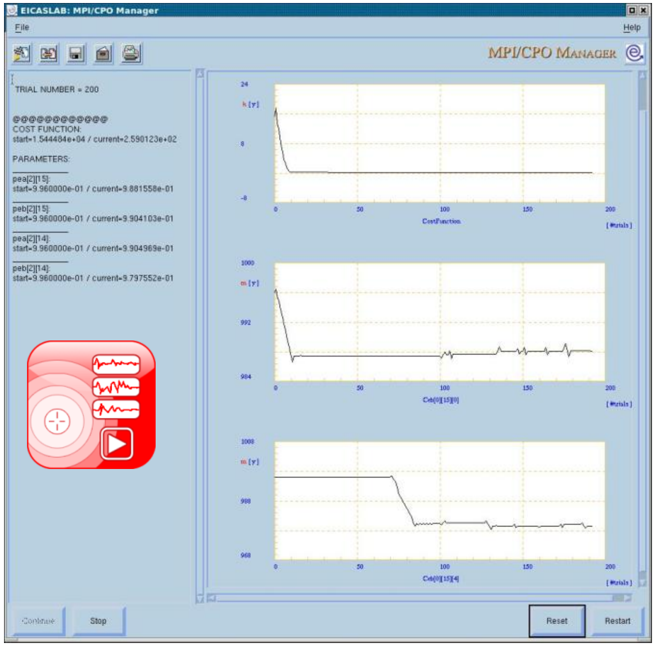
The Control Parameter Optimization in EICASLAB

EICASLAB uses a powerful numerical optimisation algorithm that allows to get the optimum value of a large number of parameters in a very reasonable computing time.
The optimisation is performed numerically over a predefined simulated trial, that is for a given mission (host command sequence and disturbance acting on the plant and any other potential event related to the plant performance) and for a given *cost function* associated with the plant control performance.



The MPI/CPO Manager tool in EICASLAB

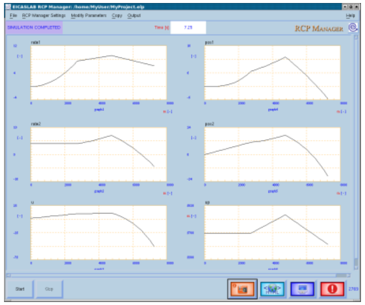
The *Model Parameter Identification* and *Control Parameter Optimisation* advanced features are managed in EICASLAB by the professional **MPI/CPO Manager tool**, available in the **MPI/CPO EICASLAB module.**



EICASLAB: MPI/CPO Manager

```
TRIAL NUMBER = 200

@@@@@@@@@@@@
COST FUNCTION:
start=1.544484e+04 / current=2.590123e+02

PARAMETERS:
__________
pea[2][15]:
start=9.960000e-01 / current=9.881558e-01
__________
peb[2][15]:
start=9.960000e-01 / current=9.904103e-01
__________
pea[2][14]:
start=9.960000e-01 / current=9.904969e-01
__________
peb[2][14]:
start=9.960000e-01 / current=9.797552e-01
```

The MPI/CPO Manager tool is an **add-on of the SIM tool** that simulates the project in an iterative way, by modifying the value of the parameters to be identified or optimised, in order to minimise the value of the cost function.



The MPI/CPO Manager tool is also an add-on of the **RCP Manager tool** allowing to perform MPI/CPO during the **Rapid Control Prototyping** phase.

How to configure MPI/CPO

The MPI/CPO process configuration

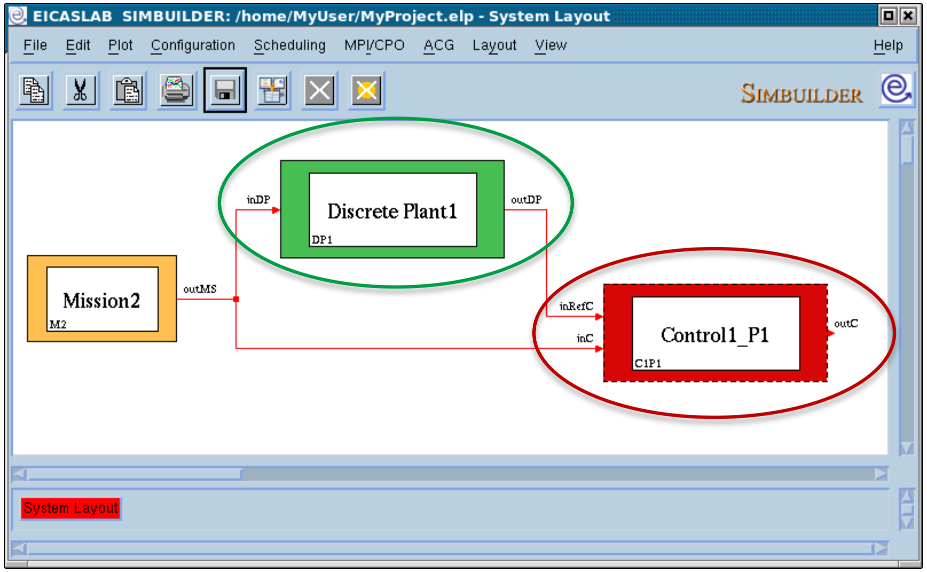
The MPI/CPO process can be easily **configured in SIMBUILDER**:

- defining the list of parameters subject to identification/optimisation;
- defining the related *cost function*;
- providing the process configuration (in particular the range of variation of the parameters to be identified/optimised and the accuracy with which the algorithm must obtain the parameter values);
- configuring suitable plotting areas for monitoring the MPI/CPO process during its execution.

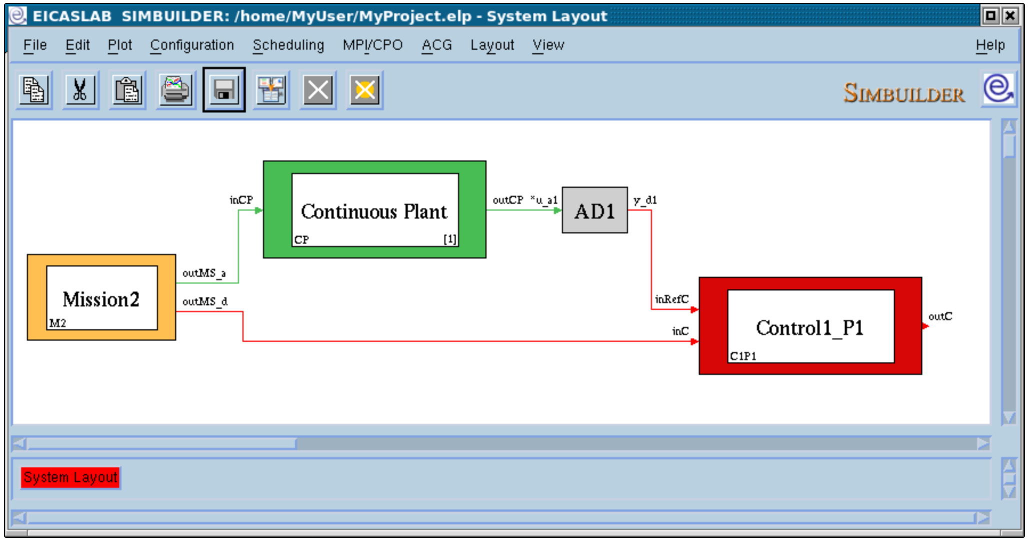
The MPI configuration

You need to build a project where you compare your plant simplified model (placed in the *Control Area*) with the accurate model or with data recorded on the actual plant (placed in the *Plant Area*).

Typically an EICASLAB project devoted to identification purposes is composed by a *Plant Area* and a *Control Area* with the same input signals.


EICASLAB SIMBUILDER: /home/MyUser/MyProject.elp - System Layout
File Edit Plot Configuration Scheduling MPI/CPO ACG Layout View Help
SIMBUILDER
Discrete Plant1
DP1
inDP
outDP
Mission2
M2
outMS
inRefC
inC
Control1_P1
C1P1
outC
System Layout

You can define your own *cost function*; generally it is related to the error between the output of the two models.
For instance:
$F=(outCP - outC)^2$


EICASLAB SIMBUILDER: /home/MyUser/MyProject.elp - System Layout
File Edit Plot Configuration Scheduling MPI/CPO ACG Layout View Help
SIMBUILDER
Continuous Plant
CP
[1]
inCP
outCP
*u_a1
AD1
y_d1
Mission2
M2
outMS_a
outMS_d
inRefC
inC
Control1_P1
C1P1
outC
System Layout

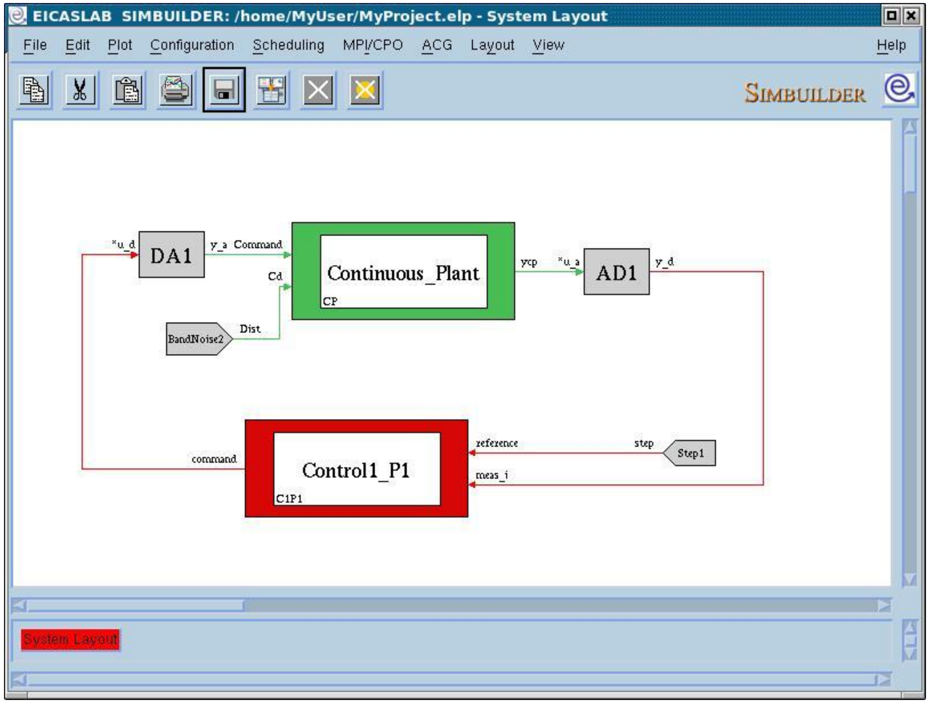
The CPO configuration

You build a project composed by a classic control architecture, with the plant model in the Plant Area and your control algorithm in the Control Area.

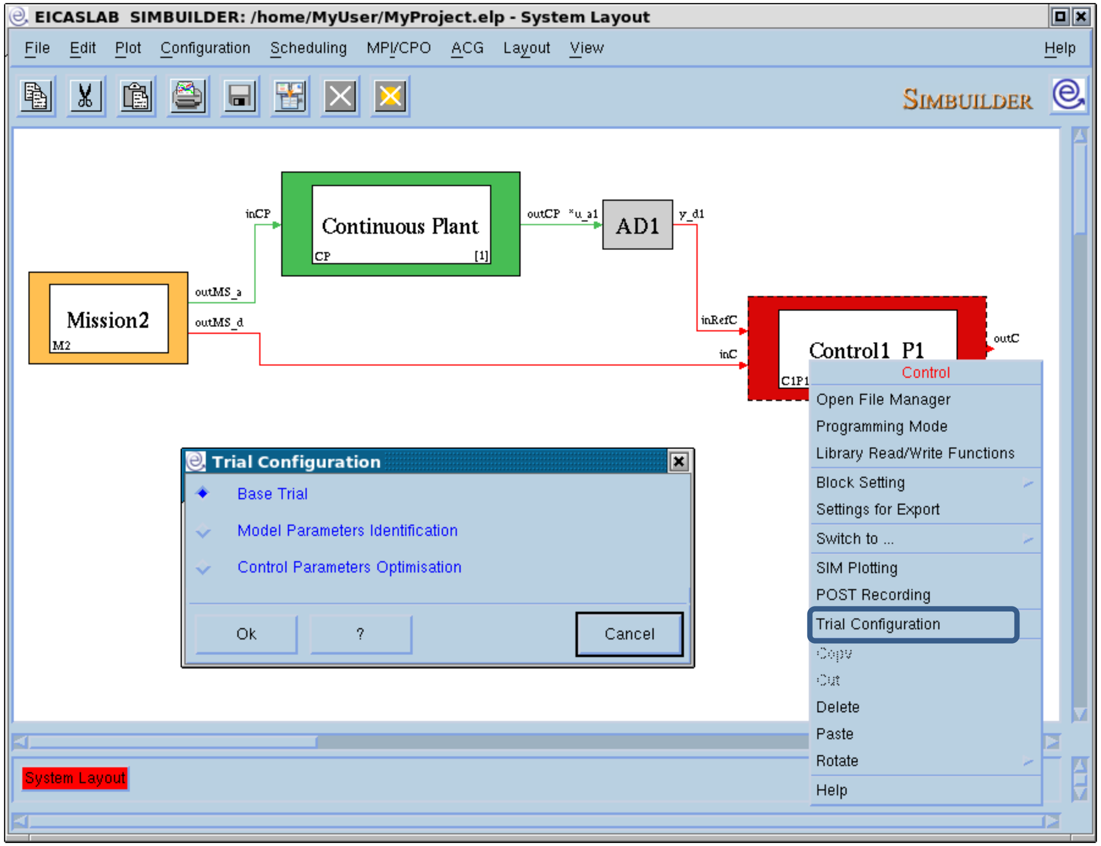
You can define your own *cost function*; it should be related to the performance indicators of the control.

For instance you can consider the error between the reference, provided by the *Mission Area*, and the measure obtained from the plant:

$F=(reference - meas\_i)^2$



The Trial Configuration setting



EICASLAB SIMBUILDER: /home/MyUser/MyProject.elp - System Layout

You can set the

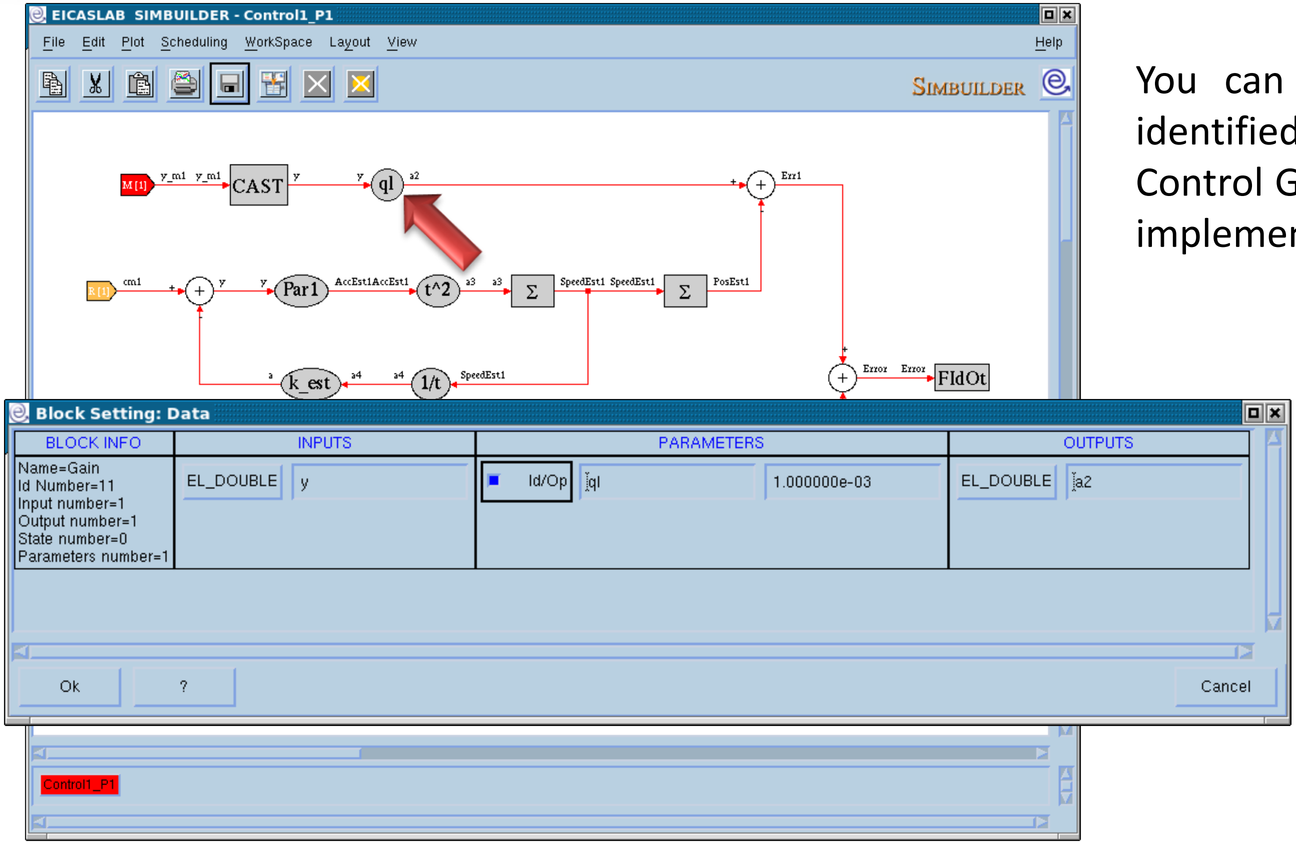
Trial Configuration

by selecting, for one Control function:

- *Base Trial,*
- *Model Parameters Identification,*
- *Control Parameters Optimization.*

MPI/CPO configuration for a Control programmed in Graphical (1/2)

Selection of the Parameters to be identified / optimised



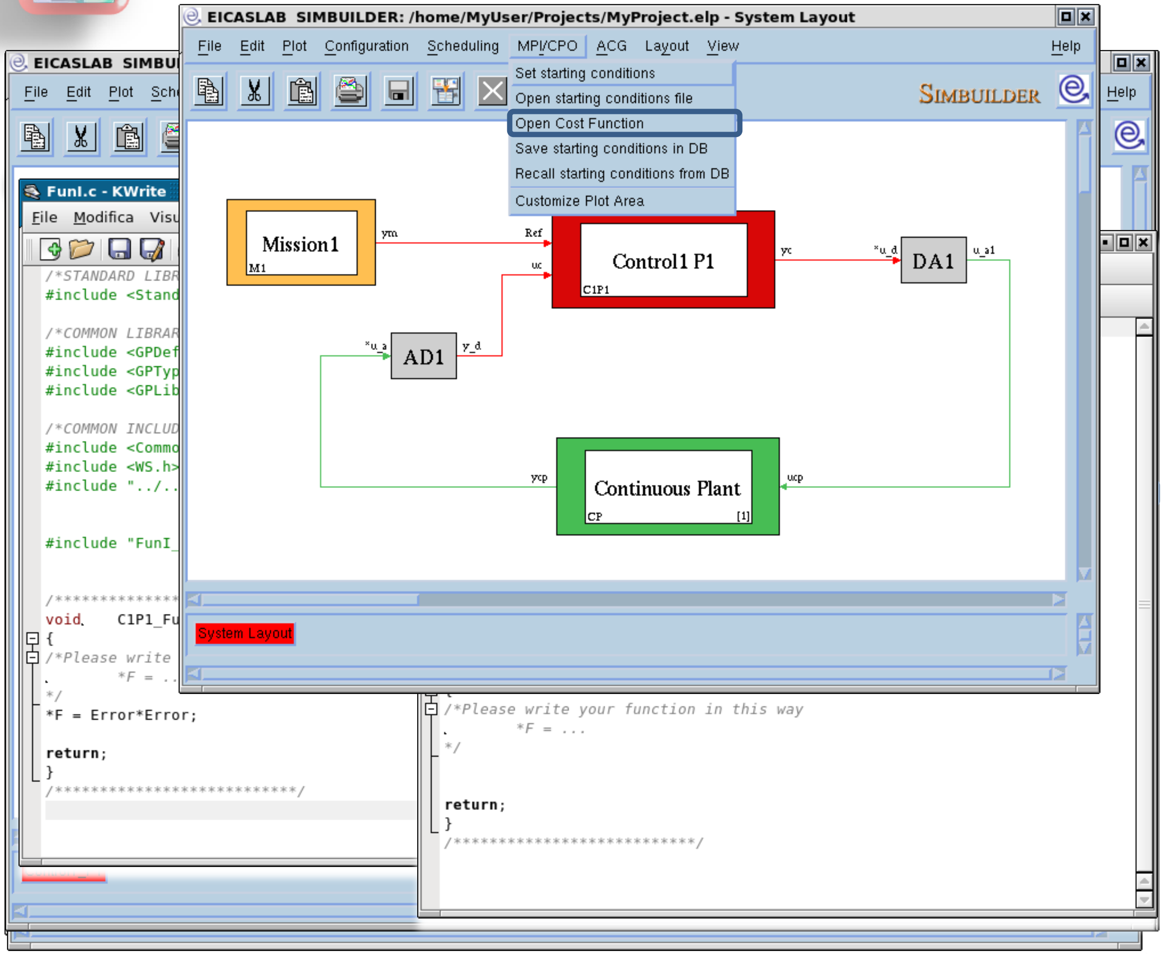
BLOCK INFO	INPUTS		PARAMETERS			OUTPUTS	
Name=Gain Id Number=11 Input number=1 Output number=1 State number=0 Parameters number=1	EL_DOUBLE	y	■ Id/Op	ql	1.000000e-03	EL_DOUBLE	a2

You can select the parameters to be identified/optimised directly in the Control Graphical Layout where you have implemented your control algorithm.

You select any block that has one or more parameters, open its 'Data' window and press on the “*Id/Op*” button corresponding to the parameters you want to identify/optimise.

MPI/CPO configuration for a Control programmed in Graphical (2/2)

Cost function



```
/*STANDARD LIBR
#include <Stand

/*COMMON LIBRAR
#include <GPDef
#include <GPTyp
#include <GPLib

/*COMMON INCLUD
#include <Commo
#include <WS.h>
#include "../..


#include "FunI_


/*************
void     C1P1_Fu
{
/*Please write
.        *F = ..
*/
*F = Error*Error;

return;
}
/****************************/
```

```
/*Please write your function in this way
.        *F = ...
*/


return;
}
/****************************/
```

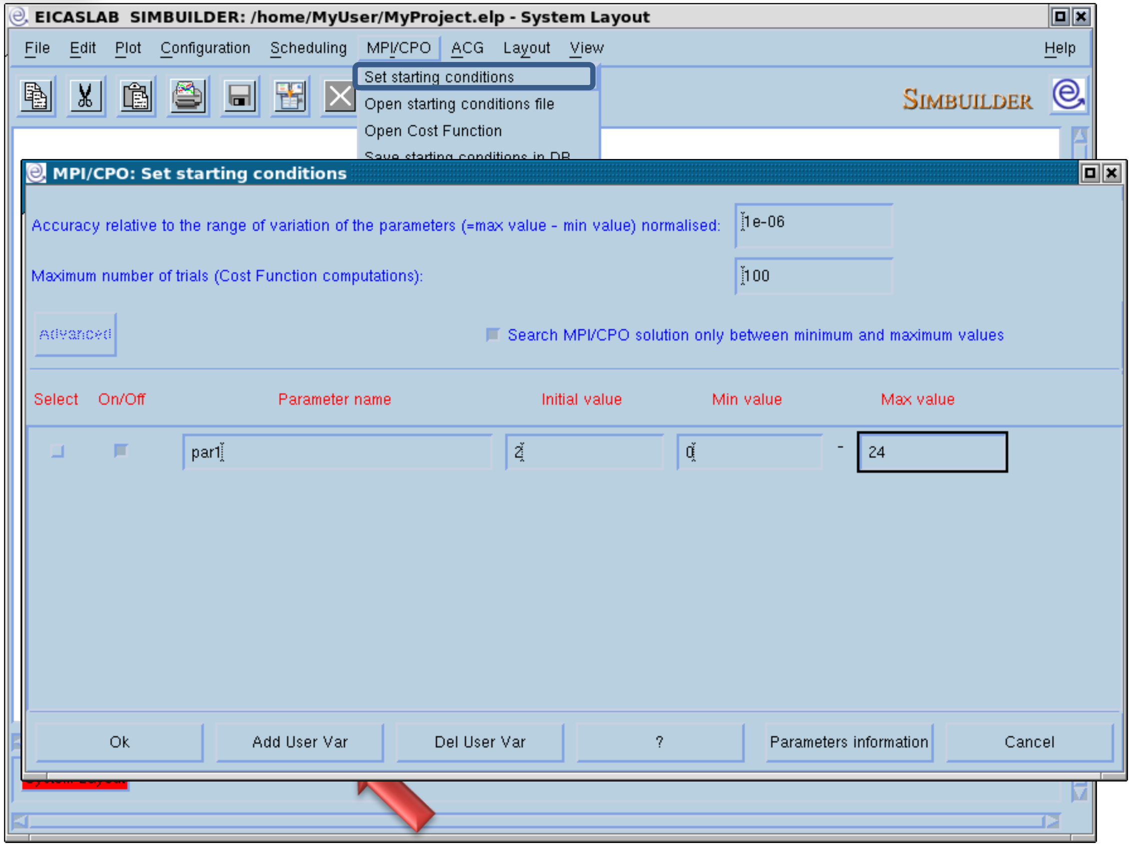
When you select the MPI/CPO for a Graphical Control, a special block, named “*FIdOt*”, is automatically inserted in the Control Graphical Layout.

From the “*FIdOt*” block you can open the file *FunI.c* (or *FunO.c*) and edit the *cost function* as a function of the inputs of the “*FIdOt*” block and of any variable of the *Workspace* of the Control Processor.

You can also edit the *FunI.c* (or *FunO.c*) file by selecting the “*Cost Function*” menu of the System Layout.

MPI/CPO configuration for a Control programmed in ANSI C (1/2)

Selection of the Parameters to be identified / optimised



Accuracy relative to the range of variation of the parameters (=max value - min value) normalised: 1e-06

Maximum number of trials (Cost Function computations): 100

☑ Search MPI/CPO solution only between minimum and maximum values

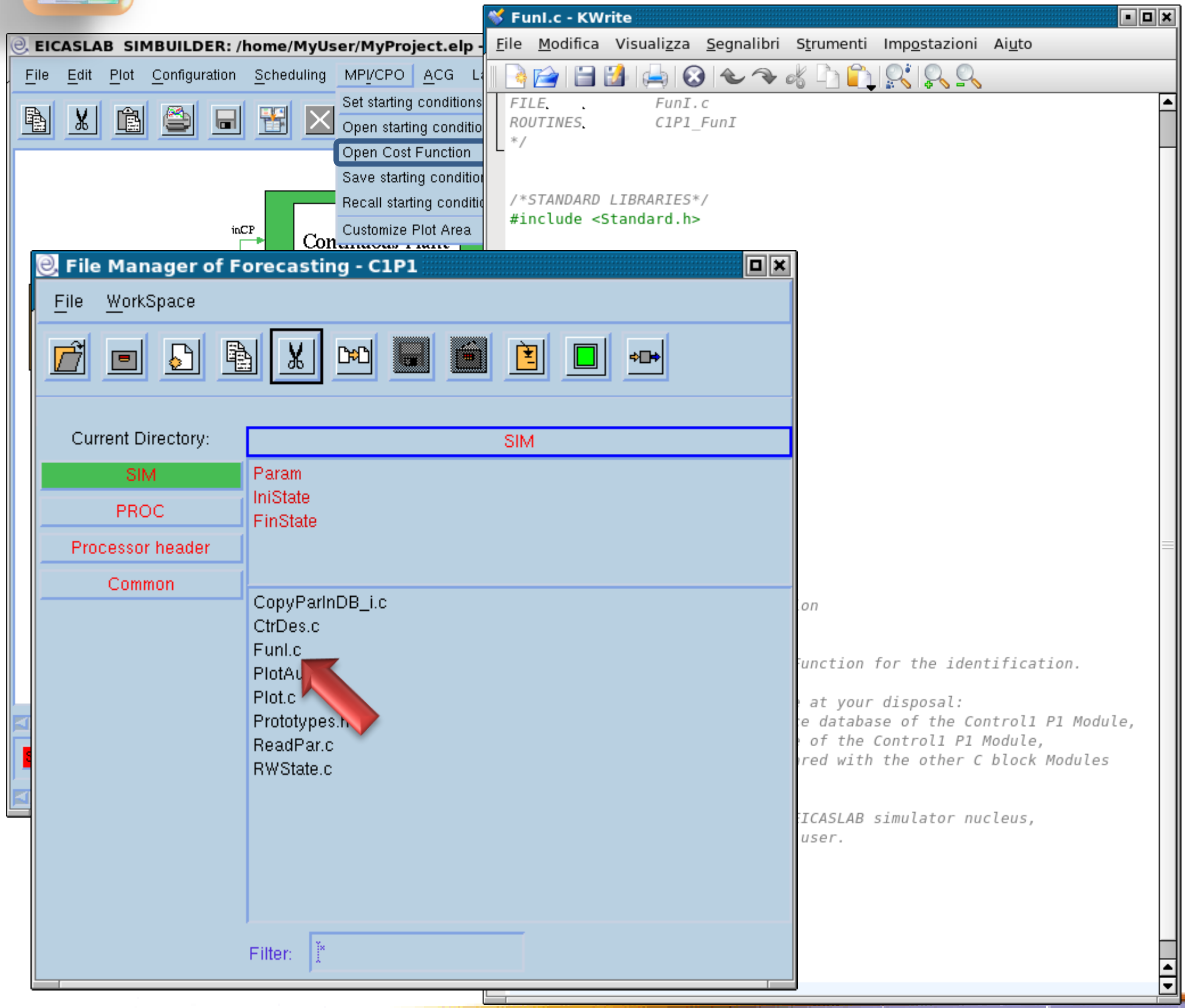
Select	On/Off	Parameter name	Initial value	Min value	Max value
□	☑	par1	2	0	24

You can select the parameters to be identified/optimised through the *"Set Starting conditions"* window, clicking on the "*Add User Var*" button.

Beyond the Parameter name you can indicate the range in which the parameter value should vary.

MPI/CPO configuration for a Control programmed in ANSI C (2/2)

Cost function



```
FILE.       .       FunI.c
ROUTINES.           C1P1_FunI
*/


/*STANDARD LIBRARIES*/
#include <Standard.h>
```

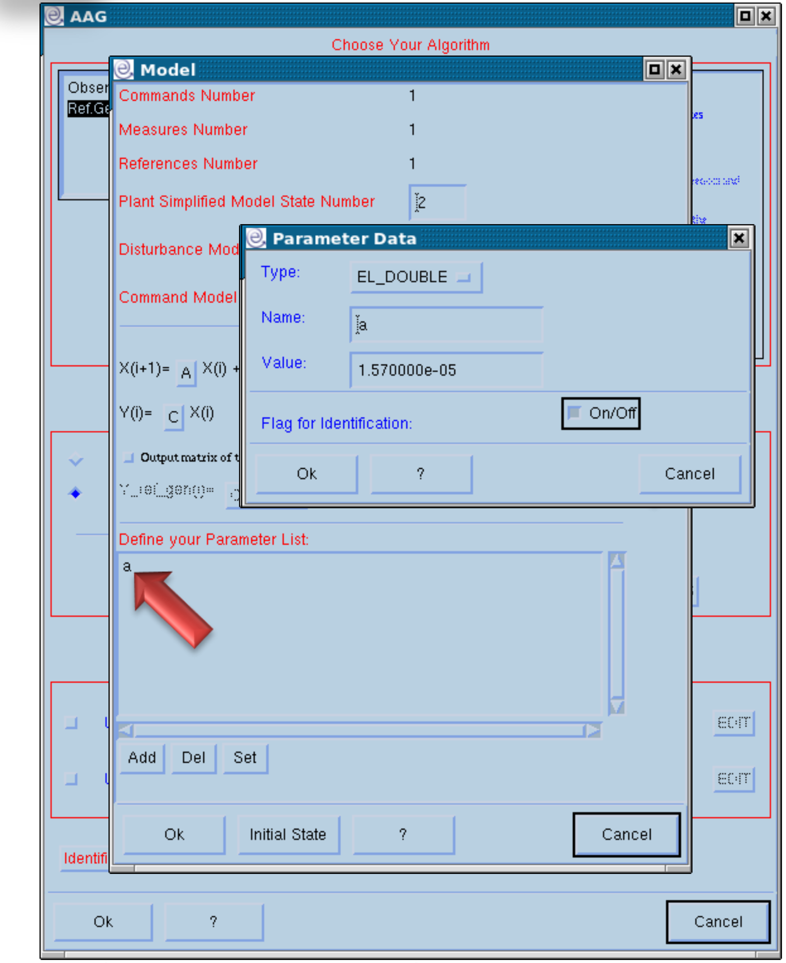
```
on

unction for the identification.

 at your disposal:
e database of the Control1 P1 Module,
 of the Control1 P1 Module,
red with the other C block Modules


ICASLAB simulator nucleus,
user.
```

For an ANSI C Control you can edit the *cost function* by means of the “Open *Cost Function*” menu of the System Layout which opens the *FunI.c* (or *FunO.c*) file containing a pre-defined function whose aim is to compute the *cost function* as a function of all the global variable available in your control.

The *FunI.c* (*FunO.c*) file is also available in the File Manager of the Control.

MPI/CPO configuration for a Control programmed in AAG (1/4)

Selection of the Parameters to be identified / optimised



AAG — Choose Your Algorithm

Model

Commands Number 1

Measures Number 1

References Number 1

Plant Simplified Model State Number 2

X(i+1)= A X(i) +

Y(i)= C X(i)

Define your Parameter List:

a

Parameter Data

Type: EL_DOUBLE

Name: a

Value: 1.570000e-05

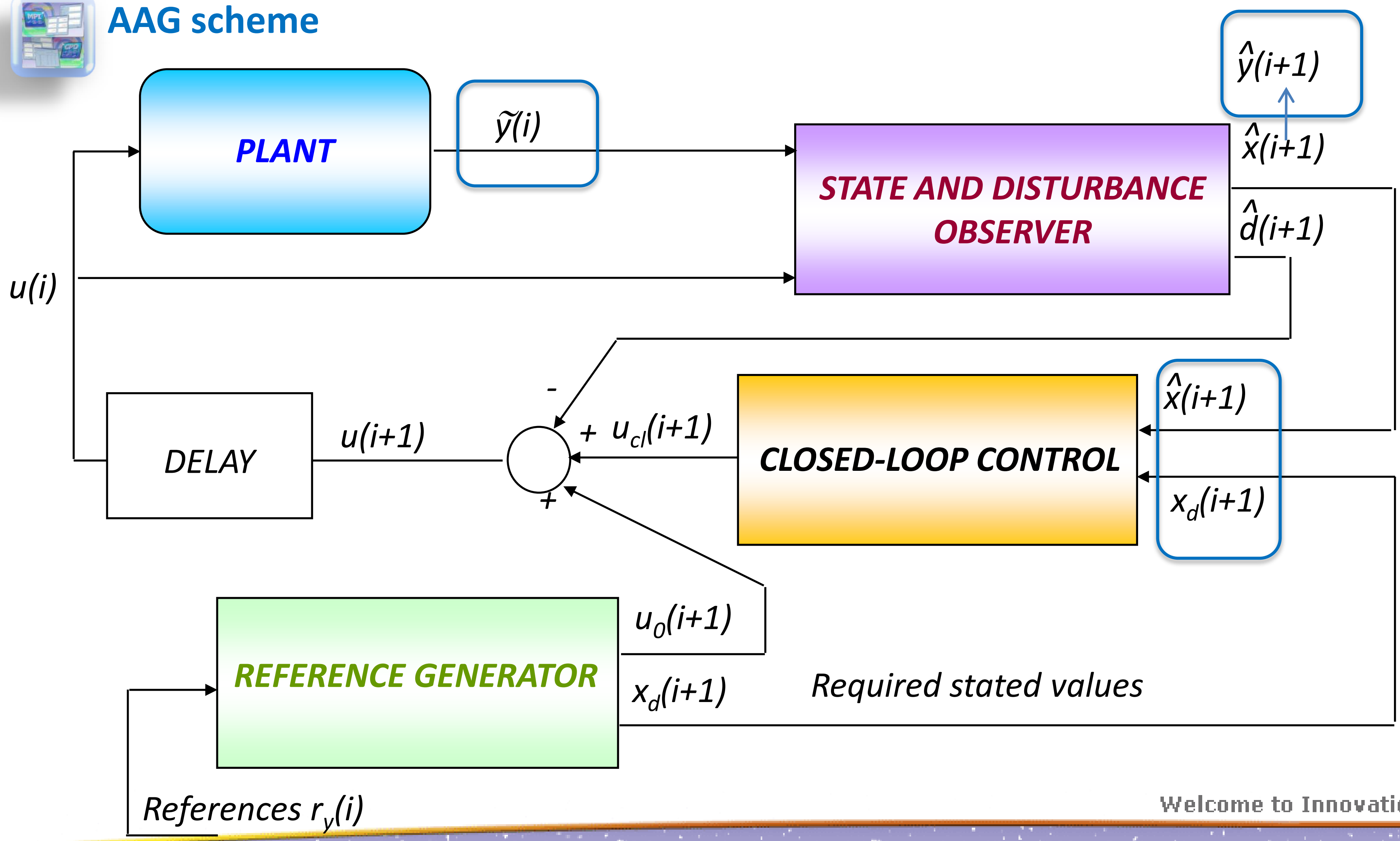
Flag for Identification: On/Off

MPI:
You can select the parameters to be identify in the "*Model*" window that is the window in which you define the plant model.
You can select any parameter of the model, open its "*Data*" window and press on the "*Flag for identification*" button.

CPO:
All the poles of the Control are automatically optimised.

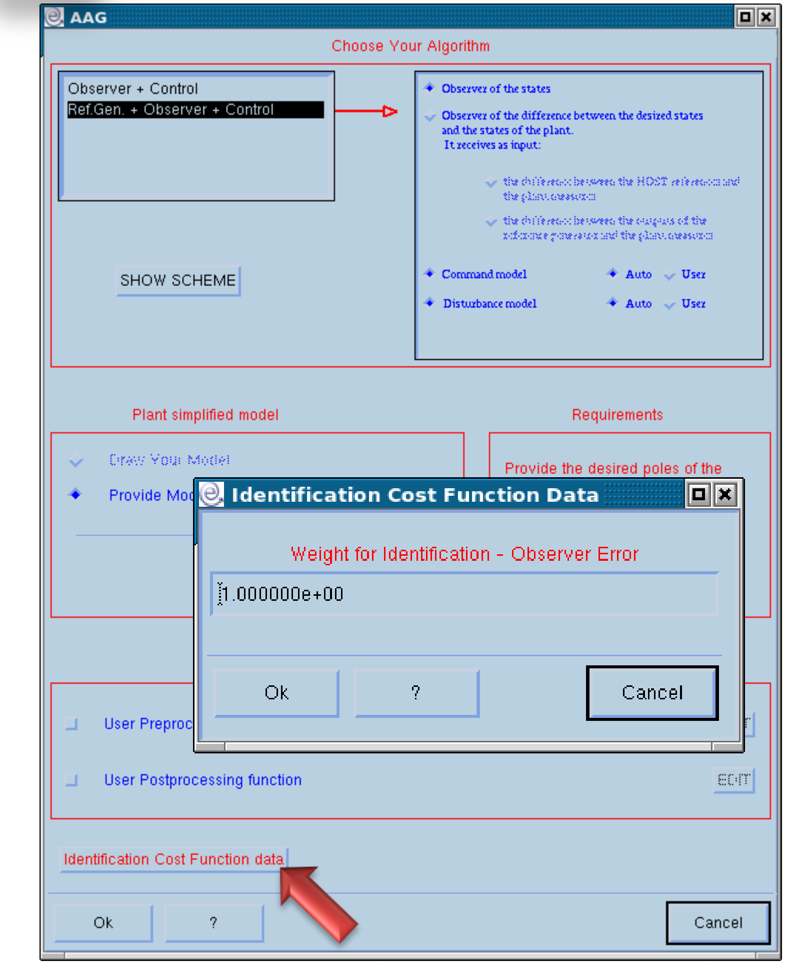
MPI/CPO configuration for a Control programmed in AAG (2/4)

AAG scheme



MPI/CPO configuration for a Control programmed in AAG (3/4)

Cost function for MPI



AAG

Choose Your Algorithm

Observer + Control
Ref.Gen. + Observer + Control

- ◆ Observer of the states
- ◇ Observer of the difference between the desired states and the states of the plant.
  It receives as input:
  - ◇ the difference between the HOST reference and the plant measures
  - ◇ the difference between the outputs of the reference generator and the plant measures
- ◆ Command model — ◆ Auto ◇ User
- ◆ Disturbance model — ◆ Auto ◇ User

SHOW SCHEME

Plant simplified model

- ◇ Draw Your Model
- ◆ Provide Mod

Requirements

Provide the desired poles of the

- □ User Preproc
- □ User Postprocessing function — EDIT

Identification Cost Function data

Ok | ? | Cancel

Identification Cost Function Data

Weight for Identification - Observer Error

1.000000e+00

Ok | ? | Cancel

For the AAG Controls EICASLAB manages automatically the *cost function* used in the MPI process.

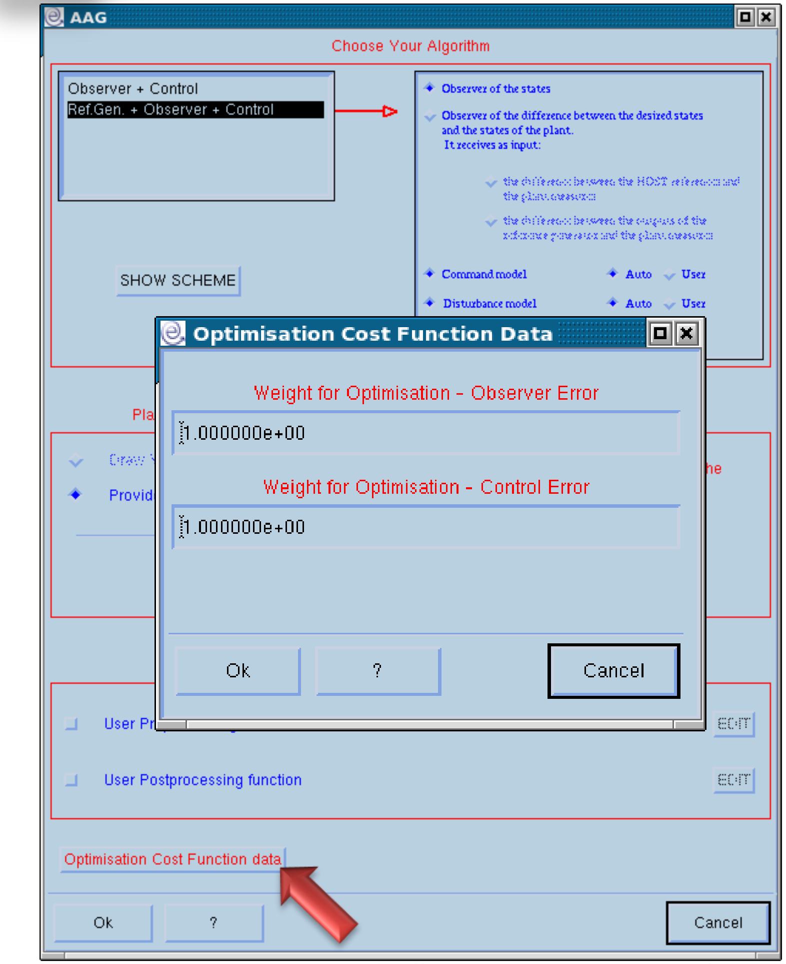
The AAG identification *cost function* is based on the ***Observer Error***, that is the difference between the estimated outputs ($\hat{y}$) and the measures coming from the plant ($\tilde{y}$):

$$F = \sum_{all\ outputs} Weight\ (\hat{y} - \tilde{y})^2$$

The *“Identification Cost Function Data”* window allows you to fix the weights corresponding to all the measures.

MPI/CPO configuration for a Control programmed in AAG (4/4)

Cost function for CPO



AAG

Choose Your Algorithm

Observer + Control
Ref.Gen. + Observer + Control

- ● Observer of the states
- ○ Observer of the difference between the desired states and the states of the plant. It receives as input:
  - ○ the difference between the HOST reference and the plant measures
  - ○ the difference between the outputs of the reference generator and the plant measures
- ● Command model ● Auto ○ User
- ● Disturbance model ● Auto ○ User

SHOW SCHEME

Pla

○ Draw

● Provid

he

□ User Pr

□ User Postprocessing function

Optimisation Cost Function data

Optimisation Cost Function Data

Weight for Optimisation - Observer Error

1.000000e+00

Weight for Optimisation - Control Error

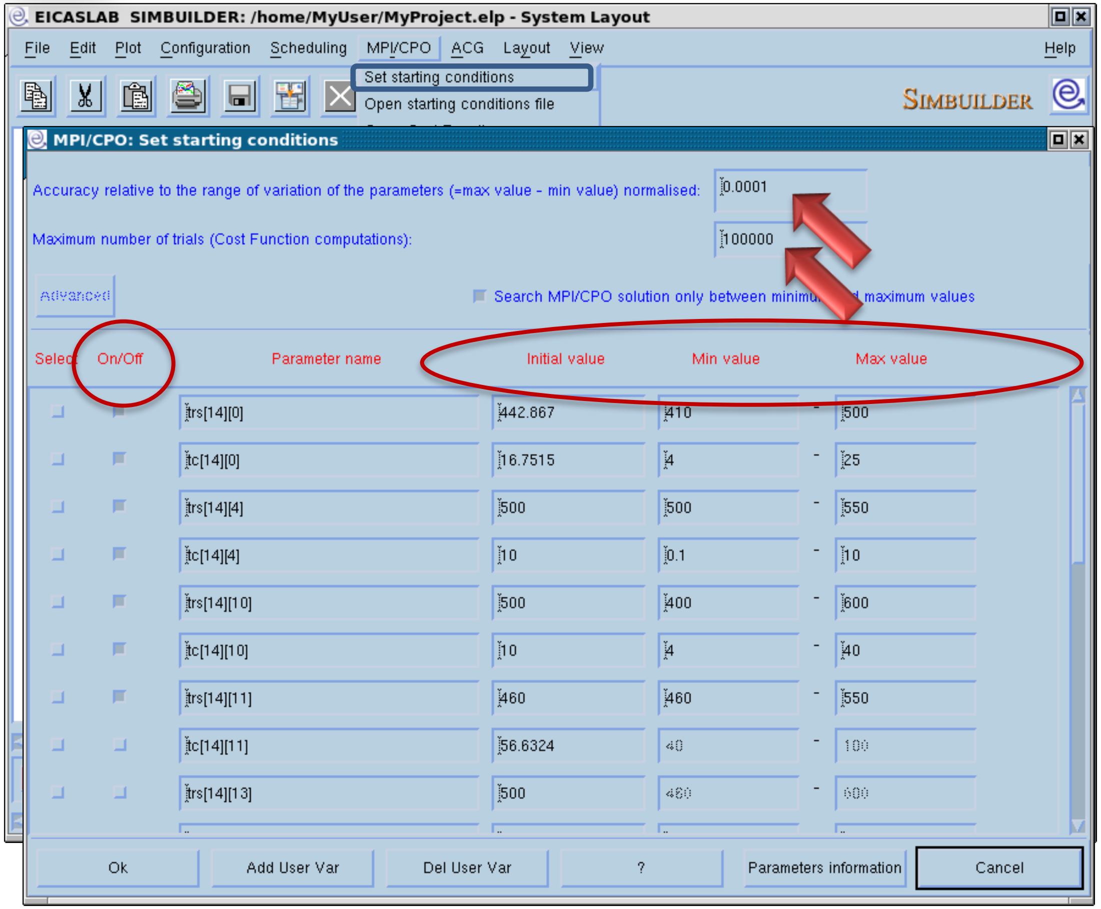
1.000000e+00

The AAG optimisation *cost function* is based both on:

- the ***Control Error***, that is the difference between the outputs of the reference generator (desired outputs, x_d) and the outputs of the state observer ($\hat{x}$)
- the ***Observer Error***, that is the difference between the estimated outputs ($\hat{y}$) and the measures coming from the plant ($\tilde{y}$):

$$F=\sum_{all\ outputs}(CtrWeight\ (x_d-\hat{x})^2+ObsWeight\ (\hat{y}-\tilde{y})^2)$$

The MPI/CPO starting conditions


EICASLAB SIMBUILDER: /home/MyUser/MyProject.elp - System Layout
File
Edit
Plot
Configuration
Scheduling
MPI/CPO
ACG
Layout
View
Help
Set starting conditions
Open starting conditions file
SIMBUILDER
MPI/CPO: Set starting conditions
Advanced

Accuracy relative to the range of variation of the parameters (=max value - min value) normalised: 0.0001

Maximum number of trials (Cost Function computations): 100000

■ Search MPI/CPO solution only between minimum and maximum values

Select	On/Off	Parameter name	Initial value	Min value		Max value
□	■	trs[14][0]	442.867	410	-	500
□	■	tc[14][0]	16.7515	4	-	25
□	■	trs[14][4]	500	500	-	550
□	■	tc[14][4]	10	0.1	-	10
□	■	trs[14][10]	500	400	-	600
□	■	tc[14][10]	10	4	-	40
□	■	trs[14][11]	460	460	-	550
□	□	tc[14][11]	56.6324	40	-	100
□	□	trs[14][13]	500	460	-	600

For any Control the “*Set Starting conditions*” window, allows you to configure the starting conditions of the MPI/CPO process:

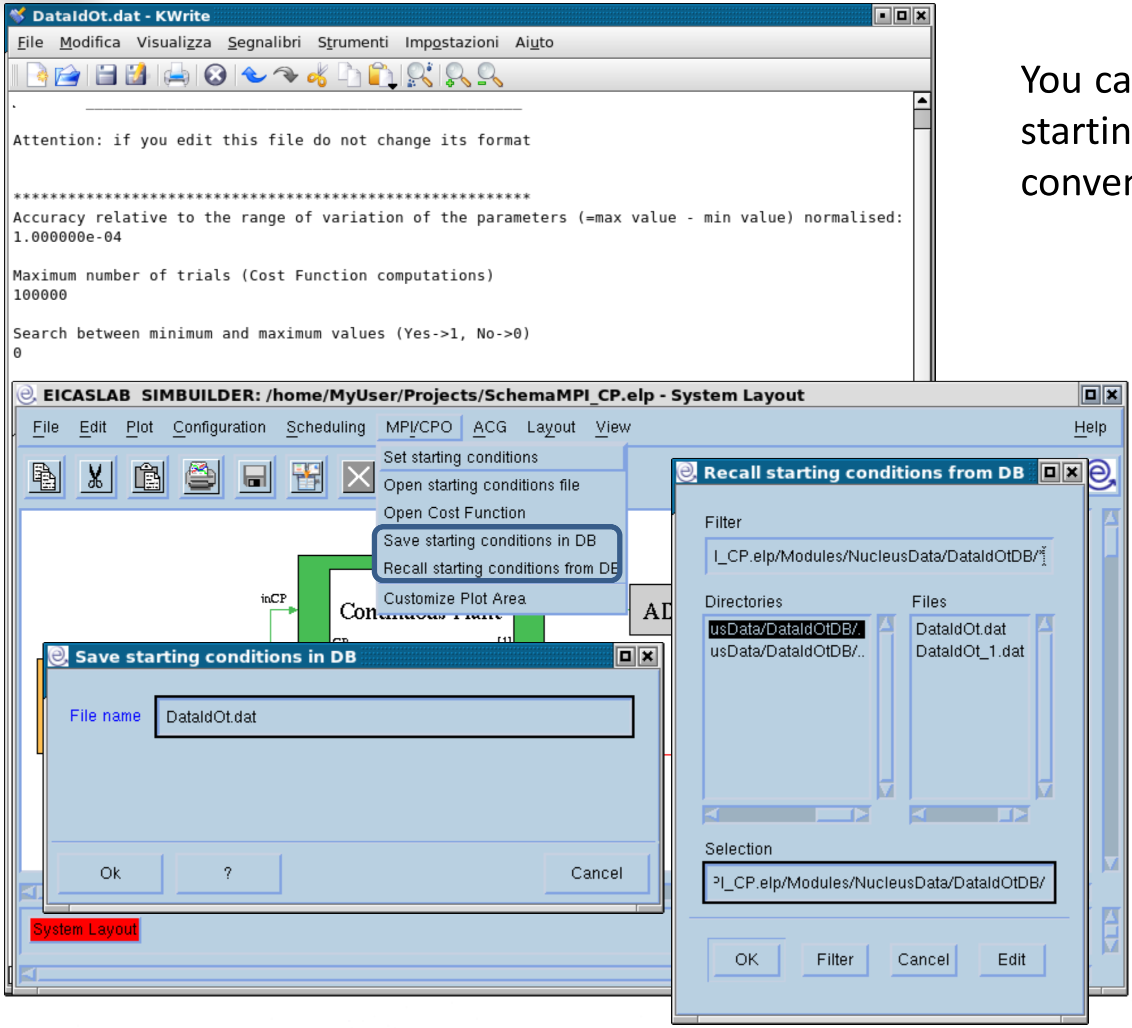
- the initial value and the range of variation of the parameters to be identified/optimized;

- the accuracy with which the algorithm must obtain the parameter values;

- the maximum number of trials to perform;

- the parameters to enable or disable: the disabled parameters will not be considered and then will not vary.

The MPI/CPO starting conditions management


DataIdOt.dat - KWrite
File Modifica Visualizza Segnalibri Strumenti Impostazioni Aiuto

```
Attention: if you edit this file do not change its format

*********************************************************
Accuracy relative to the range of variation of the parameters (=max value - min value) normalised:
1.000000e-04

Maximum number of trials (Cost Function computations)
100000

Search between minimum and maximum values (Yes->1, No->0)
0
```

EICASLAB SIMBUILDER: /home/MyUser/Projects/SchemaMPI_CP.elp - System Layout
File Edit Plot Configuration Scheduling MPI/CPO ACG Layout View Help
Set starting conditions
Open starting conditions file
Open Cost Function
Save starting conditions in DB
Recall starting conditions from DB
Customize Plot Area
inCP
System Layout
Save starting conditions in DB
Ok
?
Cancel
Recall starting conditions from DB
Directories
Files
OK
Filter
Cancel
Edit

File name: DataIdOt.dat

Filter: I_CP.elp/Modules/NucleusData/DataIdOtDB/*

Directories: usData/DataIdOtDB/. usData/DataIdOtDB/..

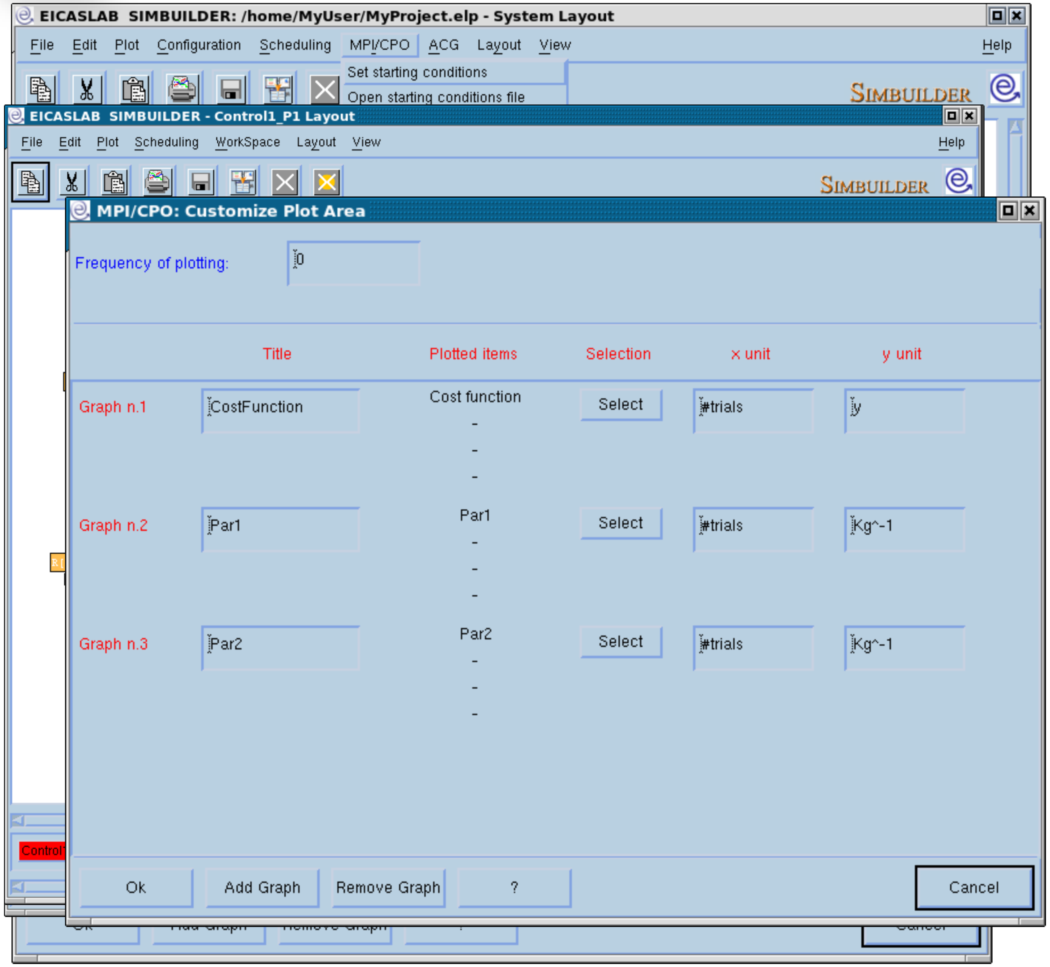
Files: DataIdOt.dat DataIdOt_1.dat

Selection: PI_CP.elp/Modules/NucleusData/DataIdOtDB/

You can directly edit the file containing the starting conditions (this is generally convenient for big amount of parameters).

You can save all your starting conditions in a Database managed by EICASLAB.

The MPI/CPO Plot Area customisation


EICASLAB SIMBUILDER: /home/MyUser/MyProject.elp - System Layout
File Edit Plot Configuration Scheduling MPI/CPO ACG Layout View Help
Set starting conditions
Open starting conditions file
SIMBUILDER
EICASLAB SIMBUILDER - Control1_P1 Layout
File Edit Plot Scheduling WorkSpace Layout View Help
SIMBUILDER
MPI/CPO: Customize Plot Area
Ok
Add Graph
Remove Graph
?
Cancel

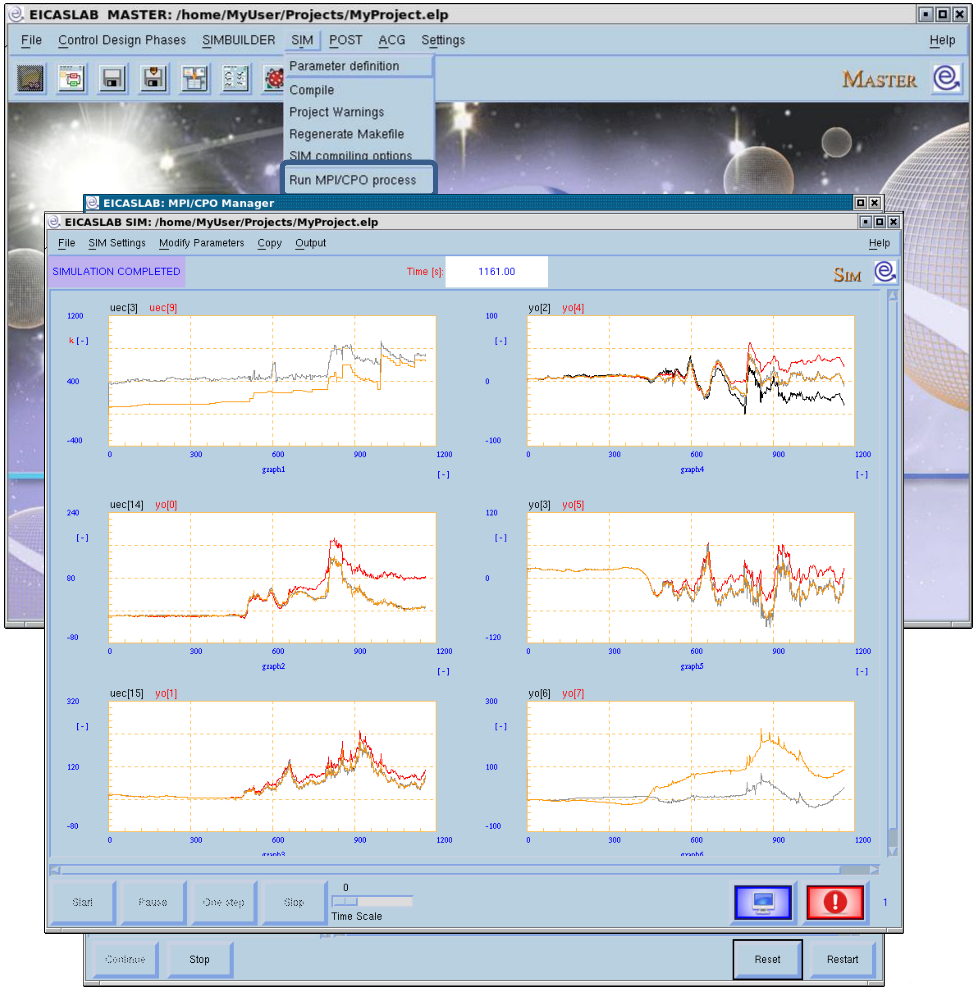
Frequency of plotting: 0

	Title	Plotted items	Selection	x unit	y unit
Graph n.1	CostFunction	Cost function - - -	Select	#trials	y
Graph n.2	Par1	Par1 - - -	Select	#trials	Kg^-1
Graph n.3	Par2	Par2 - - -	Select	#trials	Kg^-1

Before running the MPI/CPO process, you can customise the related plotting windows by means of the "*Customise Plot Area*" menu which allows you to quickly select which parameters to display during the MPI/CPO process (you can also display the *cost function*) .

The MPI/CPO process

How to start the MPI/CPO process



When you select the MPI/CPO trial configuration (for a single Control function) you can run the MPI/CPO process which is managed by the **MPI/CPO Manager tool**, which is an add-on of the **SIM tool**.

The MPI/CPO Manager and the SIM tool are both created during the Assisted Compiling Process of your project.

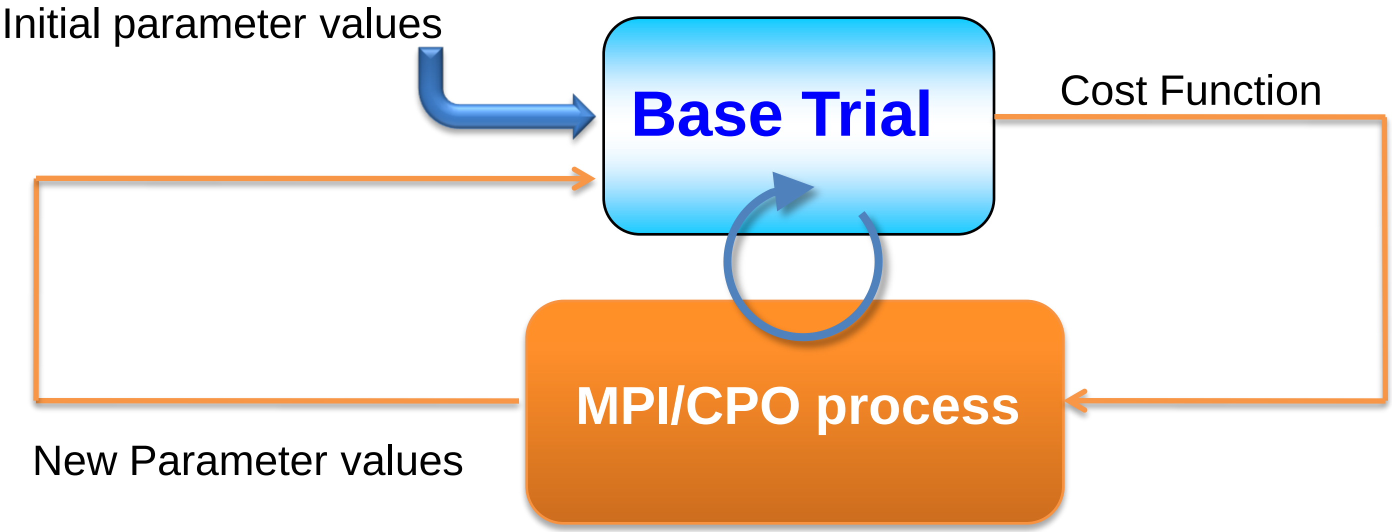
The MPI/CPO process is composed by the following three main steps

1. preliminary simulation trial, or **initial Base Trial**, performed within the SIM tool;

2. iterative process, where you have at disposal the **RCP Manager GUI**;

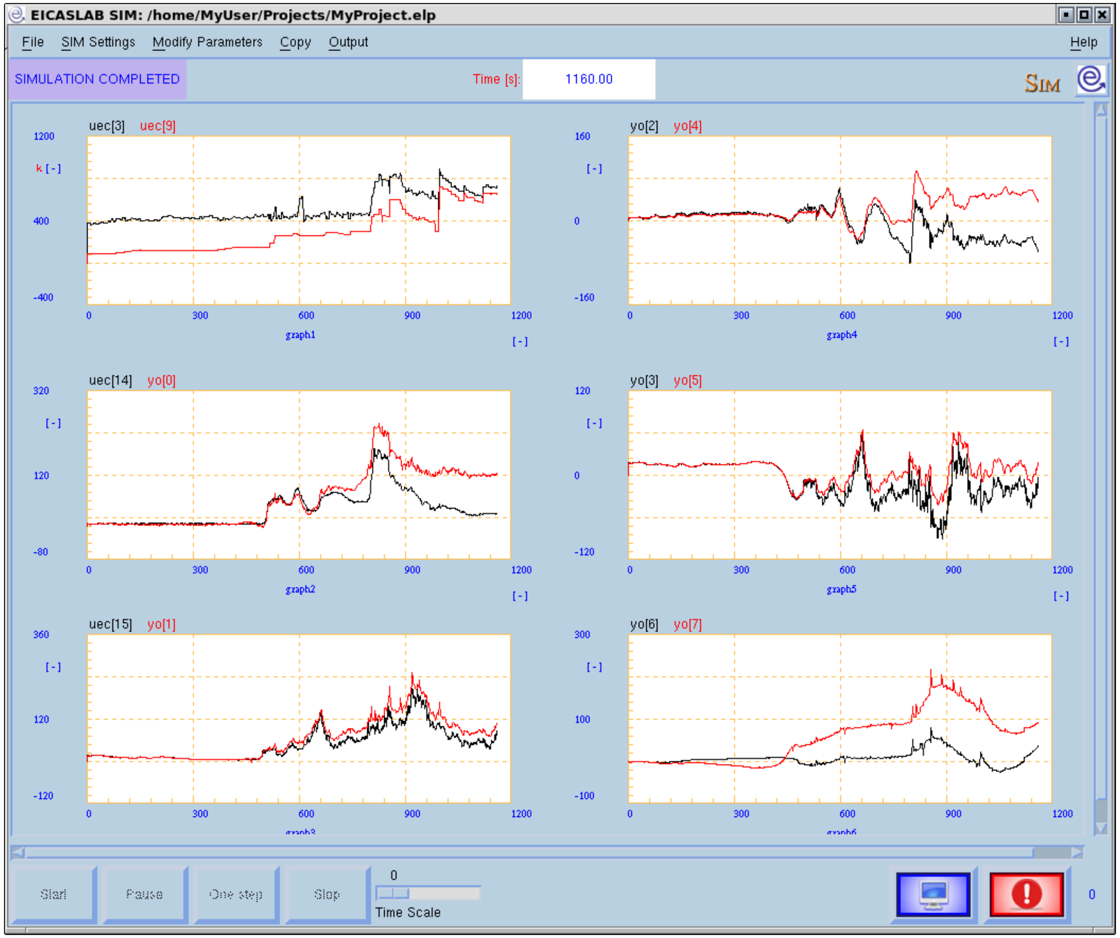
3. **final Base Trial execution**, performed within the SIM tool.

The main steps of the MPI/CPO process



1. During the initial Base Trial a preliminary simulation is performed starting from the initial parameter values provided by the user and a preliminary evaluation of the *cost function* is made.
2. Then the iterative process starts: the project is simulated in an iterative way, by modifying the value of the parameters to be identified or optimised, in order to minimise the value of the *cost function*.
3. Finally a last simulation is performed using the parameter values that minimize the *cost function*.

The initial Base Trial



EICASLAB SIM: /home/MyUser/Projects/MyProject.elp

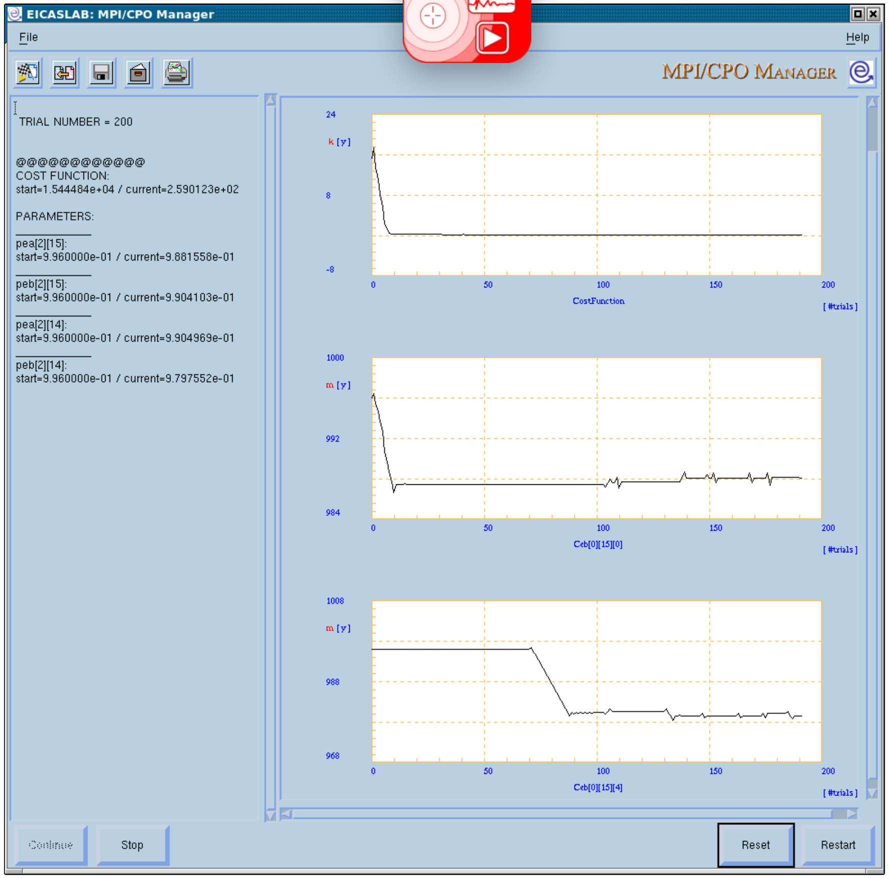
SIMULATION COMPLETED

Time [s]: 1160.00

In the first step, a preliminary simulation - called **initial Base Trial** - is performed starting from the initial parameter values provided by the user and a preliminary evaluation of the *cost function* is made.

During the initial Base Trial, the trends of the variables selected by the user may be displayed in the classical SIM plotting window.

The iterative process



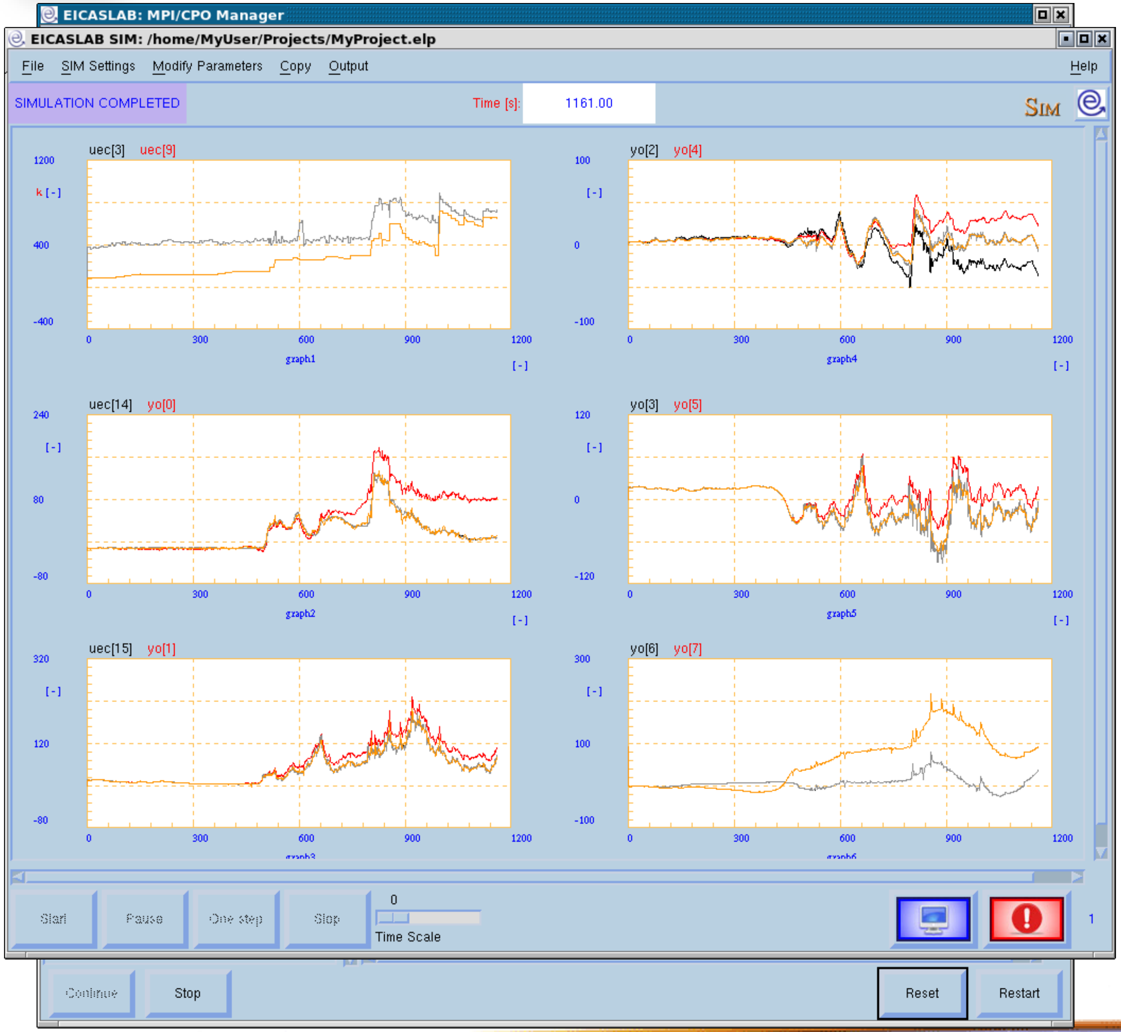
```
TRIAL NUMBER = 200


@@@@@@@@@@@@
COST FUNCTION:
start=1.544484e+04 / current=2.590123e+02

PARAMETERS:
____________
pea[2][15]:
start=9.960000e-01 / current=9.881558e-01
____________
peb[2][15]:
start=9.960000e-01 / current=9.904103e-01
____________
pea[2][14]:
start=9.960000e-01 / current=9.904969e-01
____________
peb[2][14]:
start=9.960000e-01 / current=9.797552e-01
```

At the end of the execution of the initial Base Trial, the second step, named **iterative process**, starts. It is the iterative part of the MPI/CPO process in which the parameter values are changed in order to reduce the *cost function* value, updated at any iteration, until the minimum has been reached, with the requested accuracy.

The MPI/CPO process is performed by a powerful numerical algorithm, developed by EICAS, which allows in a very reasonable computing time to get the optimum value of a large number of parameters (it has been already used to optimise the control system with up to about 500 parameters).

During the iterative process the MPI/CPO Manager GUI gets opened on the screen allowing the user to control the overall iterative process.

The final Base Trial



EICASLAB: MPI/CPO Manager

EICASLAB SIM: /home/MyUser/Projects/MyProject.elp

SIMULATION COMPLETED

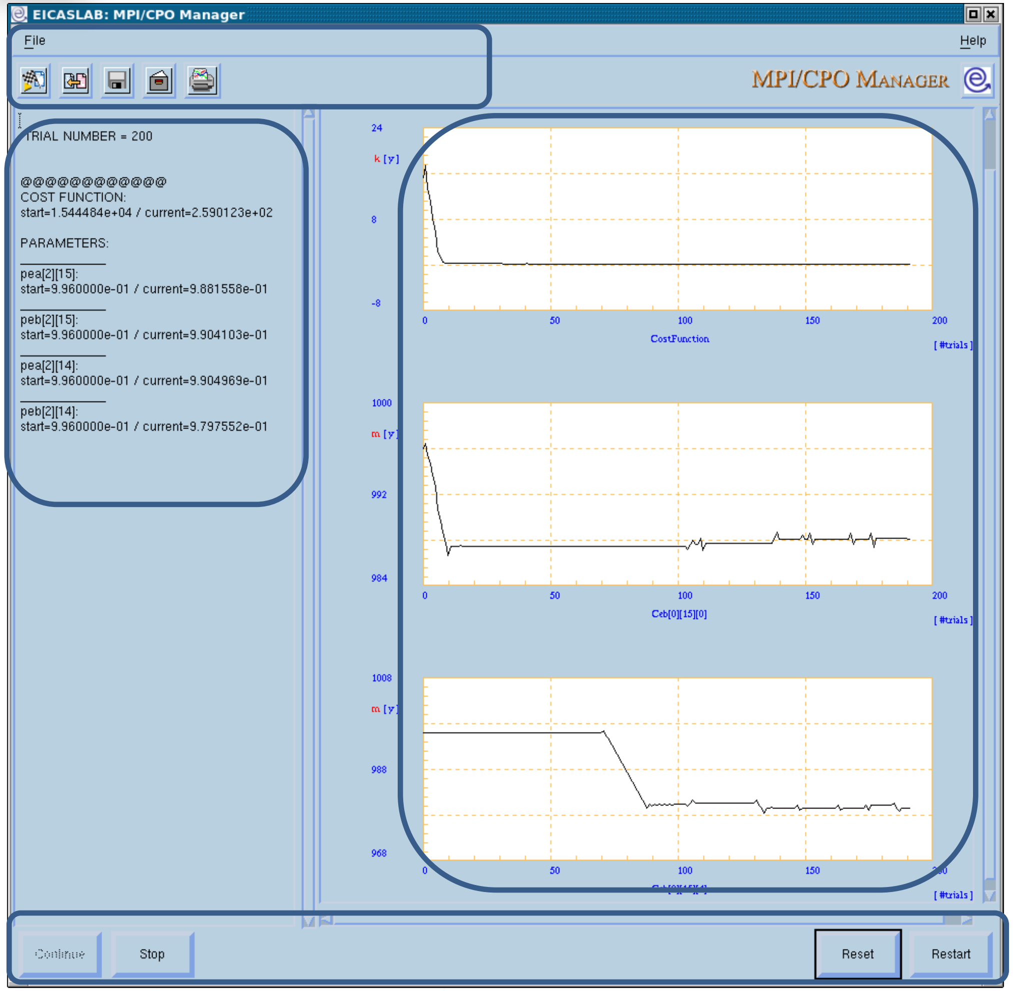
Time [s]: 1161.00

When the iterative process is completed the third step, named final Base Trial, starts.

Once achieved the minimum value of the *cost function*, the final Base Trial is executed with the related parameter values obtained from the previous iterative process.

The results are graphically displayed again in the SIM GUI: SIM plots the dynamic behaviour of the variables selected for SIM plotting, both in the initial and final Base Trials, for showing a comparison between them in order to immediately show the benefits obtained with the MPI/CPO algorithm.

The MPI/CPO Manager GUI



```
TRIAL NUMBER = 200


@@@@@@@@@@@@
COST FUNCTION:
start=1.544484e+04 / current=2.590123e+02

PARAMETERS:
________
pea[2][15]:
start=9.960000e-01 / current=9.881558e-01
________
peb[2][15]:
start=9.960000e-01 / current=9.904103e-01
________
pea[2][14]:
start=9.960000e-01 / current=9.904969e-01
________
peb[2][14]:
start=9.960000e-01 / current=9.797552e-01
```

The MPI/CPO Manager tool has its own Graphical User Interface which includes:

- a control panel for a full control of the MPI/CPO process allowing to start, continue or stop the MPI/CPO process, as well as to restart it;
- a set of plotting areas for displaying the parameters and *cost function* values;
- a text area for showing the results of the MPI/CPO process, including the numerical values both of the parameters and of the *cost function* as they are changed during the MPI/CPO process with respect to the initial values used in the initial Base Trial;
- a menu toolbar for customising the algorithm and for configuring the parameter values.

EicasLab™

The Professional Software Suite
for Automatic Control Design
and Forecasting

for Linux & for Windows

www.eicaslab.com

Identificazione e ottimizzazione

MPI/CPO MODULE

Il modulo MPI/CPO è specificatamente concepito per effettuare

Model Parameter Identification

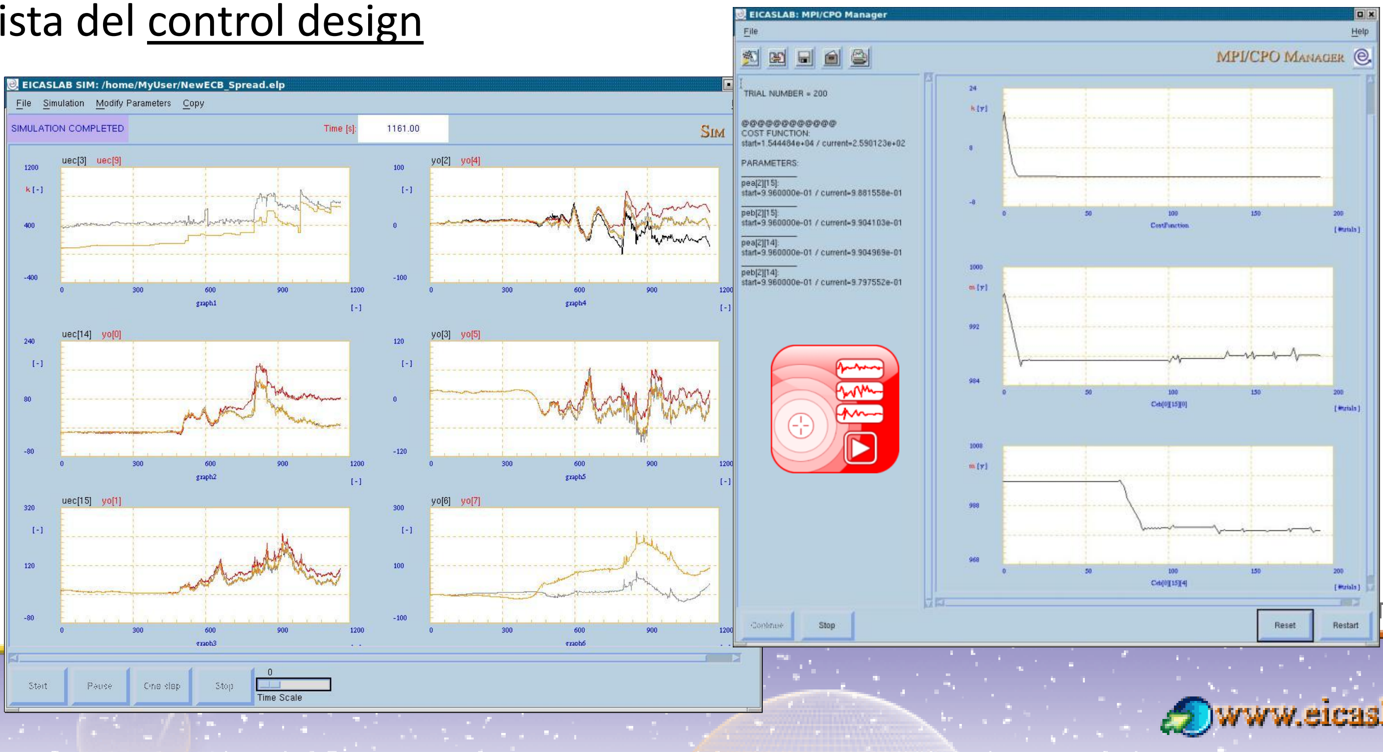
Adotta un metodo originale di identificazione, orientato a stimare i valori migliori dei parametri del modello semplificato dell'impianto dal punto di vista del control design

Control Parameter Optimisation

EICASLAB adotta un potente algoritmo di ottimizzazione numerica, che consente di ottenere in tempi ridotti il valore ottimo di un largo set di parametri.

MPI/CPO Manager

è il tool di EICASLAB che gestisce questa fase



```
TRIAL NUMBER = 200

@@@@@@@@@@@@@
COST FUNCTION:
start=1.544484e+04 / current=2.590123e+02

PARAMETERS:
________
pea[2][15]:
start=9.960000e-01 / current=9.881558e-01
________
peb[2][15]:
start=9.960000e-01 / current=9.904103e-01
________
pea[2][14]:
start=9.960000e-01 / current=9.904969e-01
________
peb[2][14]:
start=9.960000e-01 / current=9.797552e-01
```