



 **EICASLAB™**

The professional software suite
for automatic control design and forecasting

EICASLAB in a nutshell

User objectives
and required
performance

System Concept
Functional architecture, H/W e S/W

Like
real-time

Algorithm and software development
in interaction with a simulated plant

Real-time

Rapid Control Prototyping
Test and tuning on the real plant

Like
real-time

Slow Motion
Final tuning

Like
real-time

Automatic Code Generation
for the target

Real-time

Hardware in the loop
hardware testing with simulated plant

Final Validation
Tests

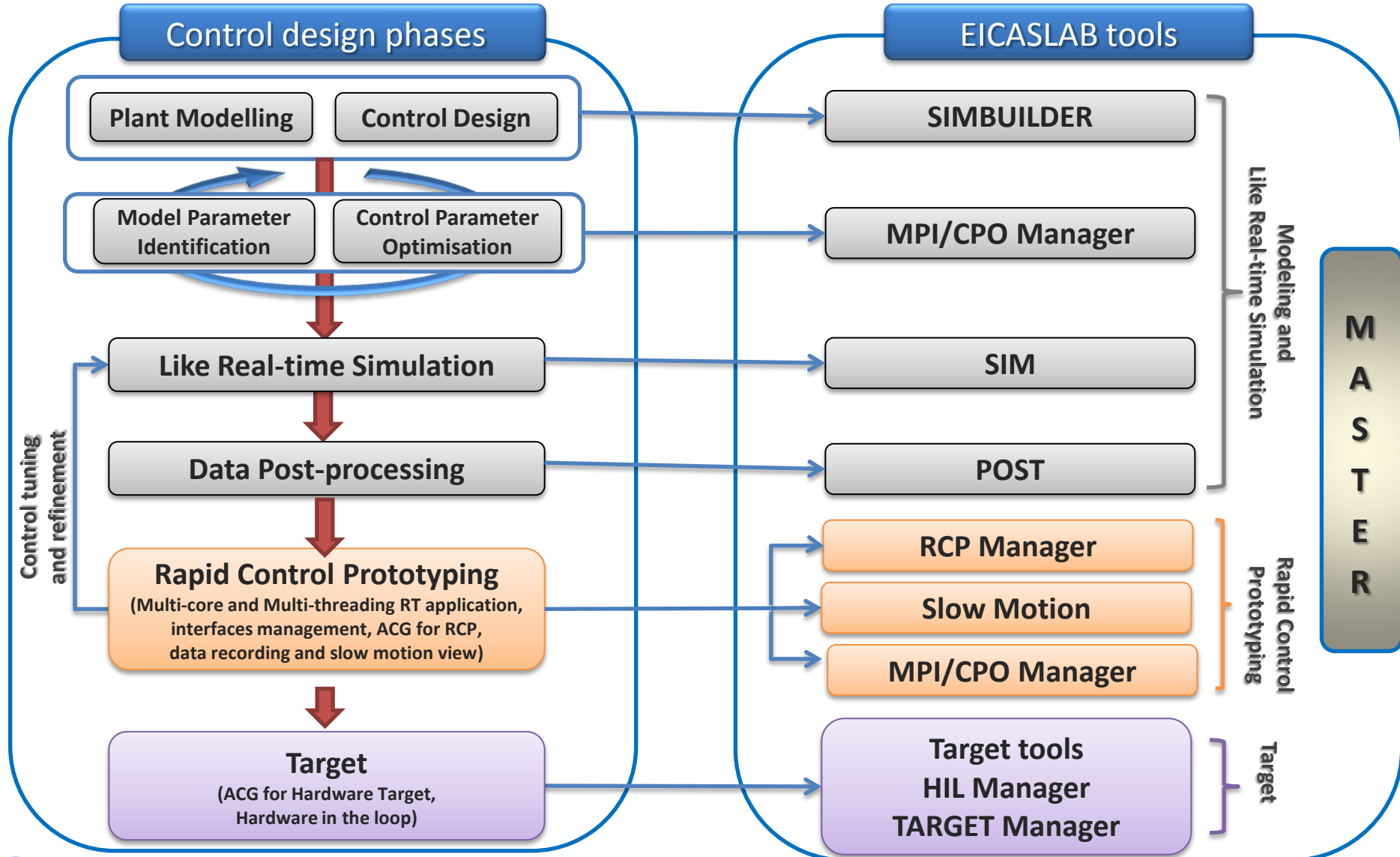
EICASLAB™
The professional
software suite
for automatic control
design and
forecasting



EICASLAB RCP Platform
Standard Multicore PC equipped
with a RTOS (Real-Time
Operative System) and the
EICASLAB Suite
RCP=Rapid Control Prototyping

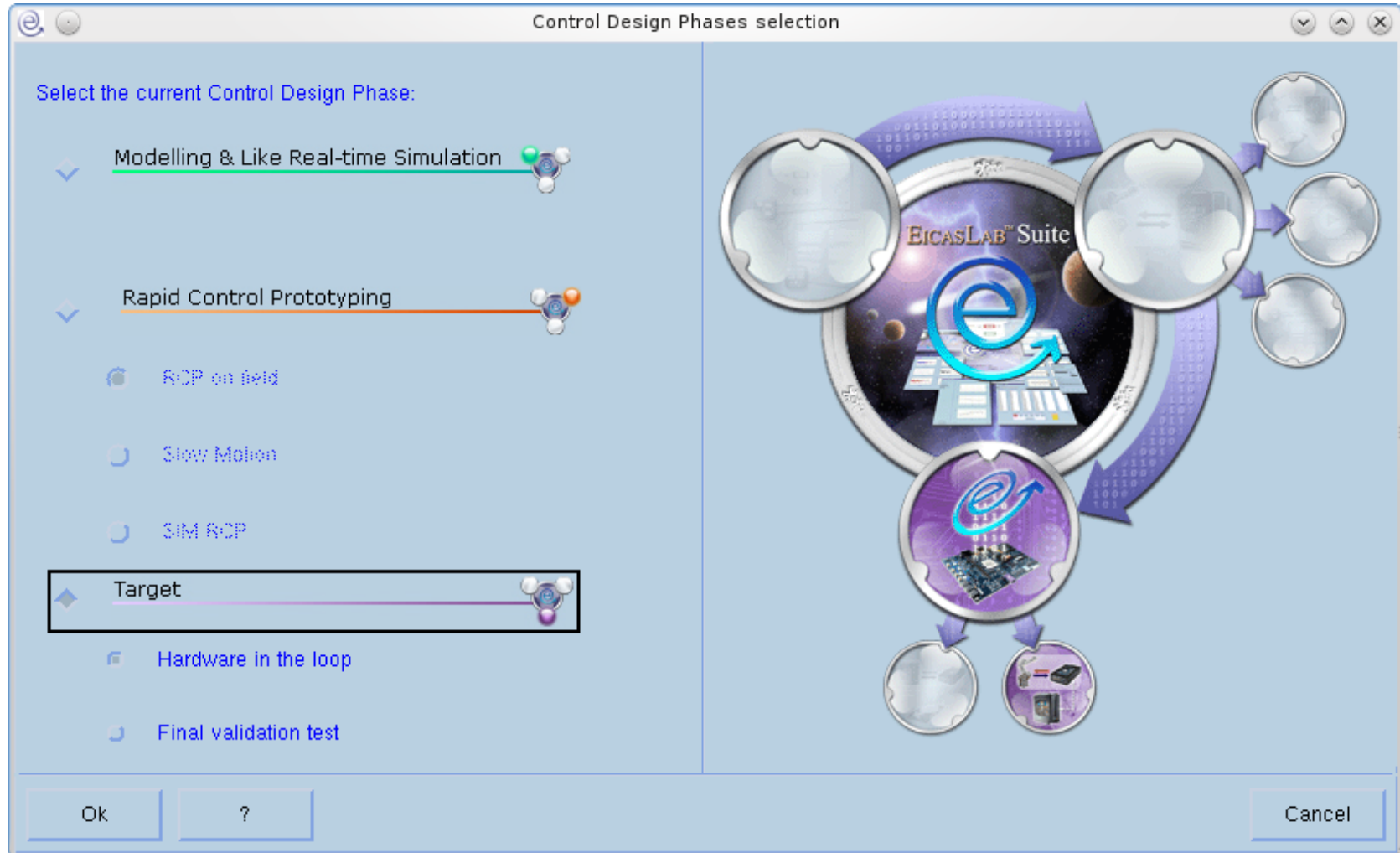


EICASLAB™ tools cover all the phases of the design process



Just one sw suite, Just one project!

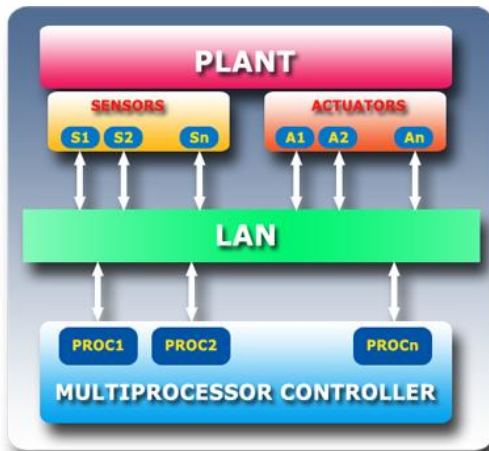
★ You can build all control design phases in a single EICASLAB project



System concept and design

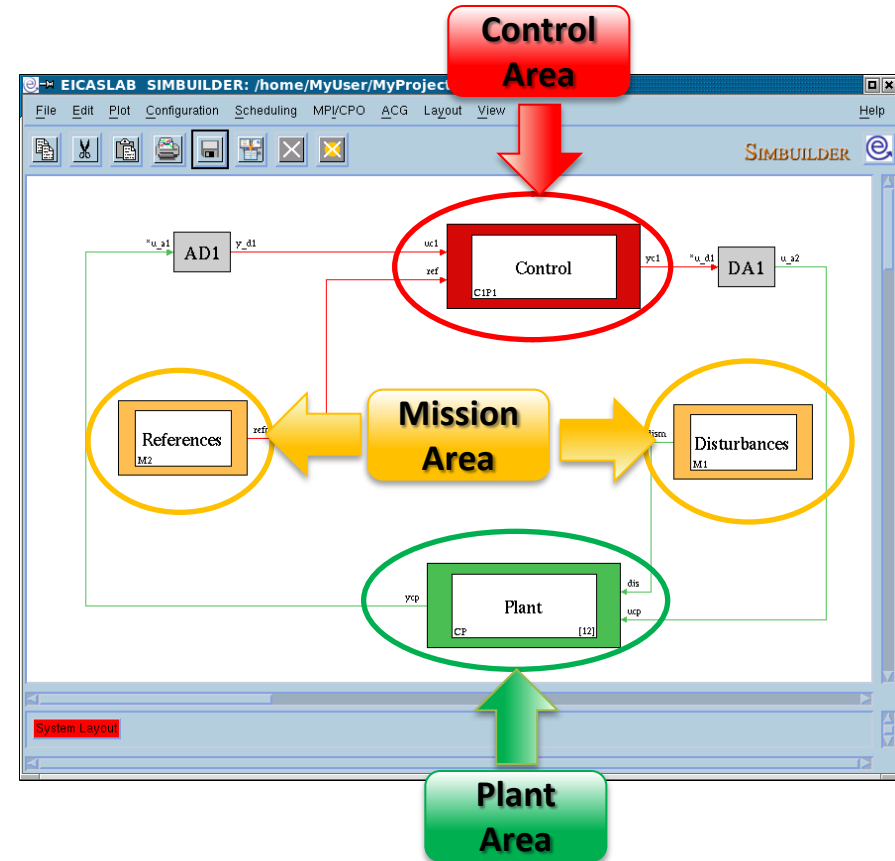


Starting from the requirements, the designer develops its own system concept



and define the architecture to be implemented

Multi processor control architectures at multi hierarchical levels



Pre-organized working areas
Plant, Control, Mission Areas

Modelling

The 2 model approach

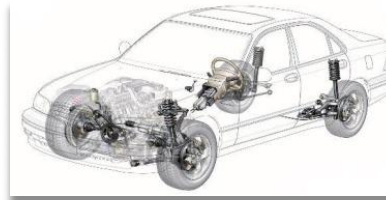
Accurate Model

Usually non linear,
suitable for
modelling the
plant to be
controlled

Plant Area

Model libraries for specific application
sectors:

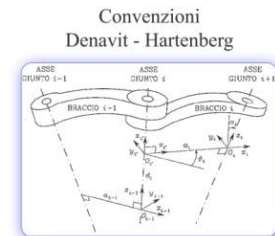
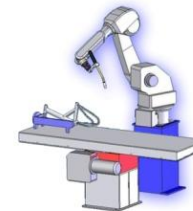
- Automotive
- Economy



- Space



- Robotics, machine tools



Simplified Model

Usually linear,
suitable for
designing the
control

Control Area

In EICASLAB you directly design digital controls

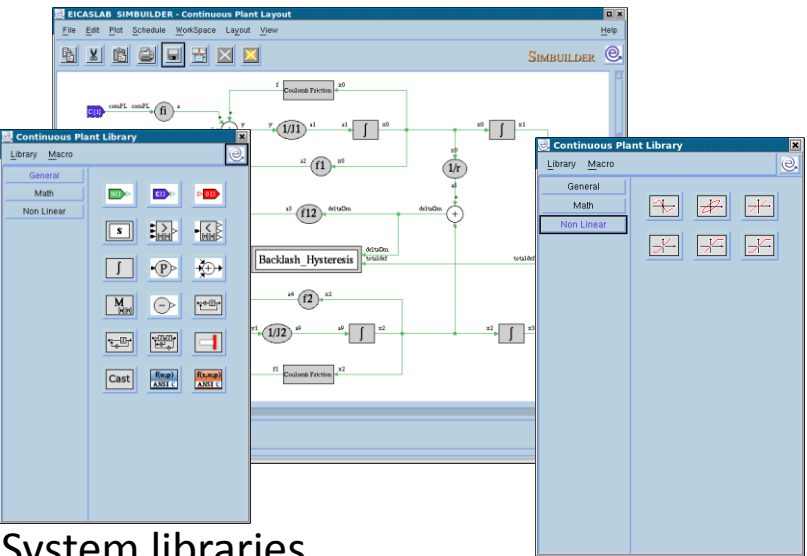
In EICASLAB you can implement your desired
control design methodology



Programming modes

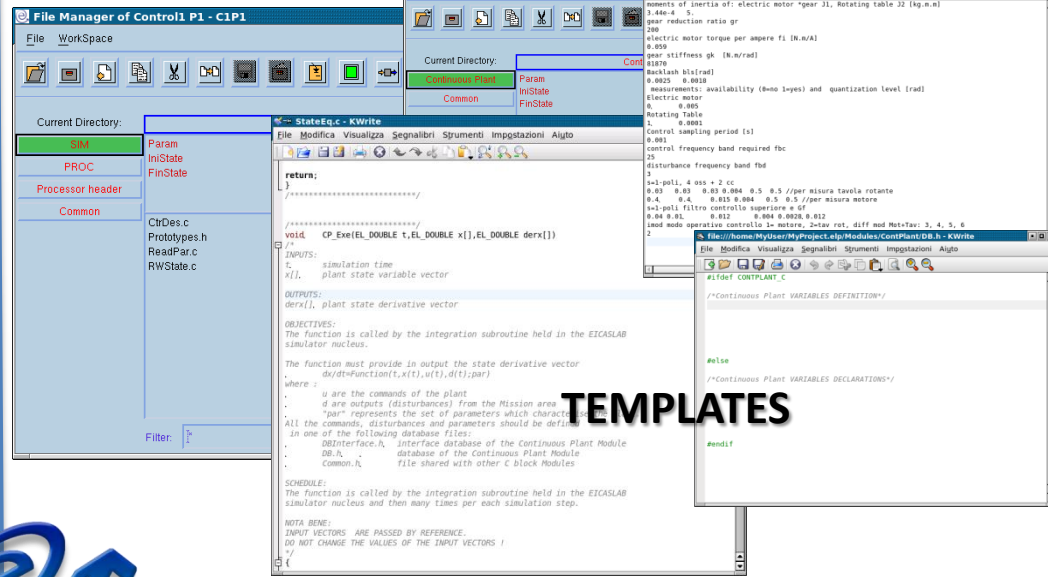


GRAPHICAL MODE



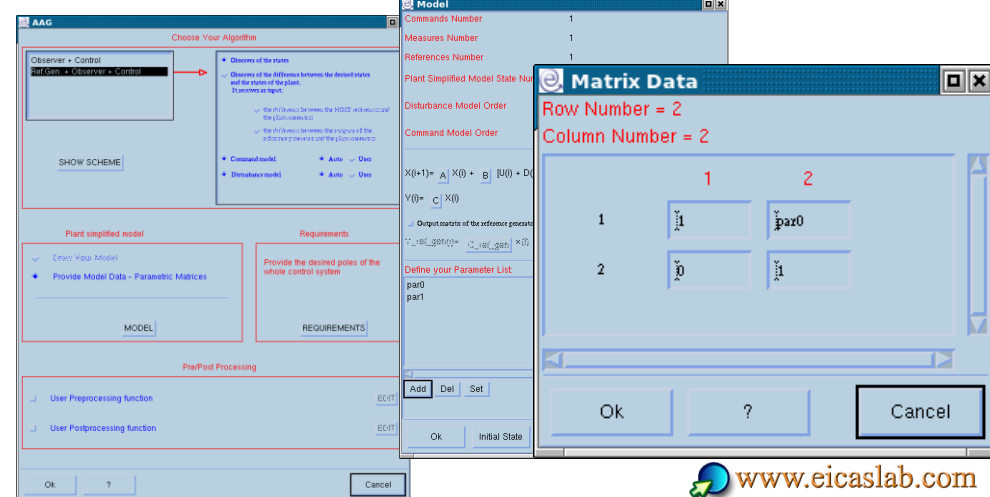
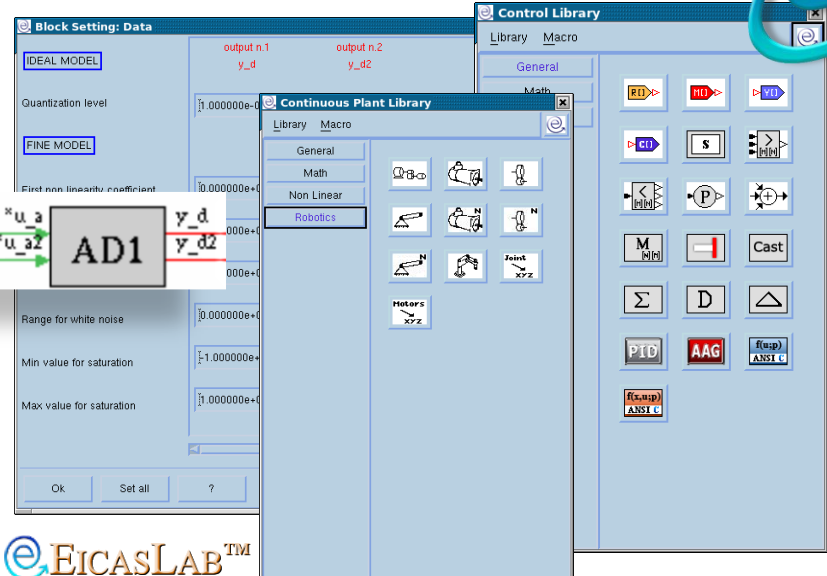
System libraries,
User libraries (macroblocks)

ANSI C



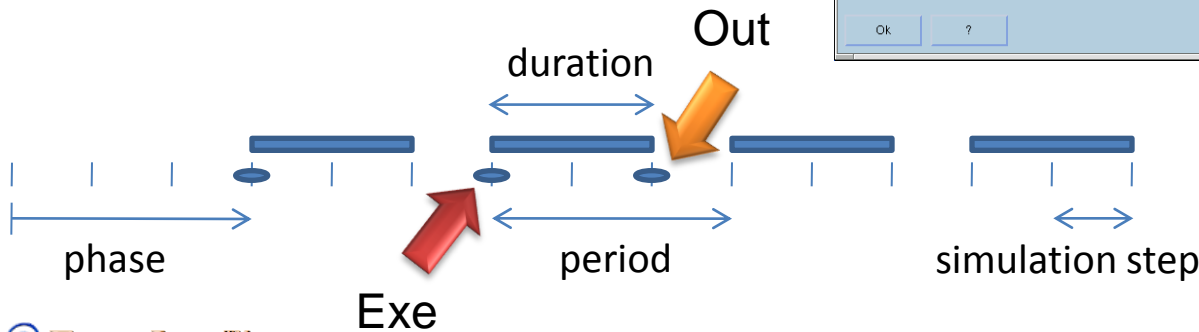
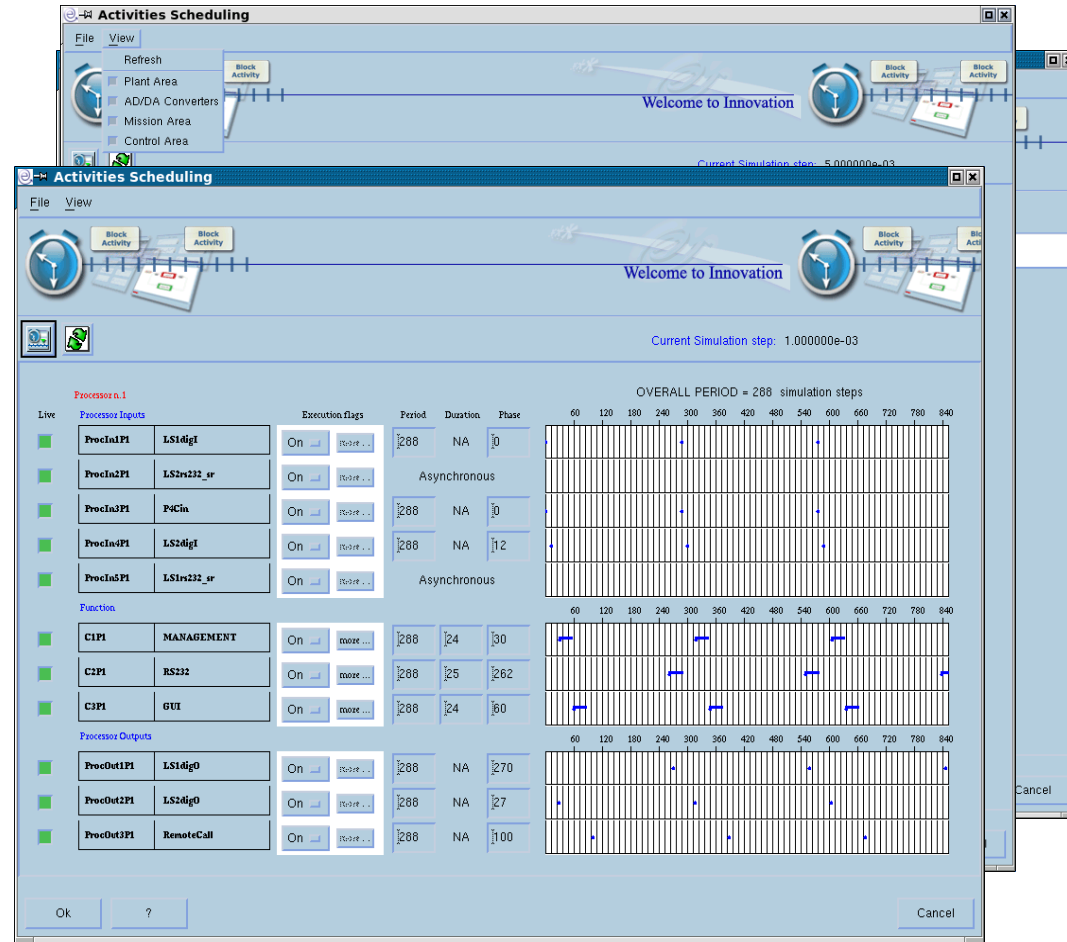
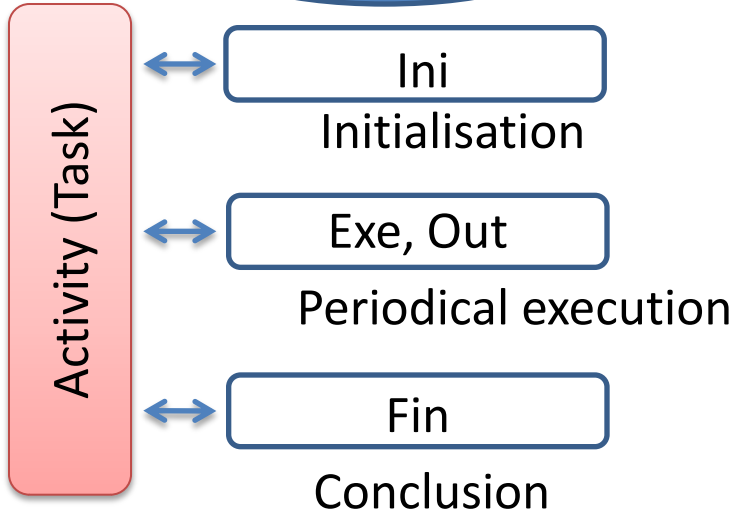
TEMPLATES

AAG Automatic Algorithm Generation



Activities scheduling

Like real-time



Like Real Time Simulation



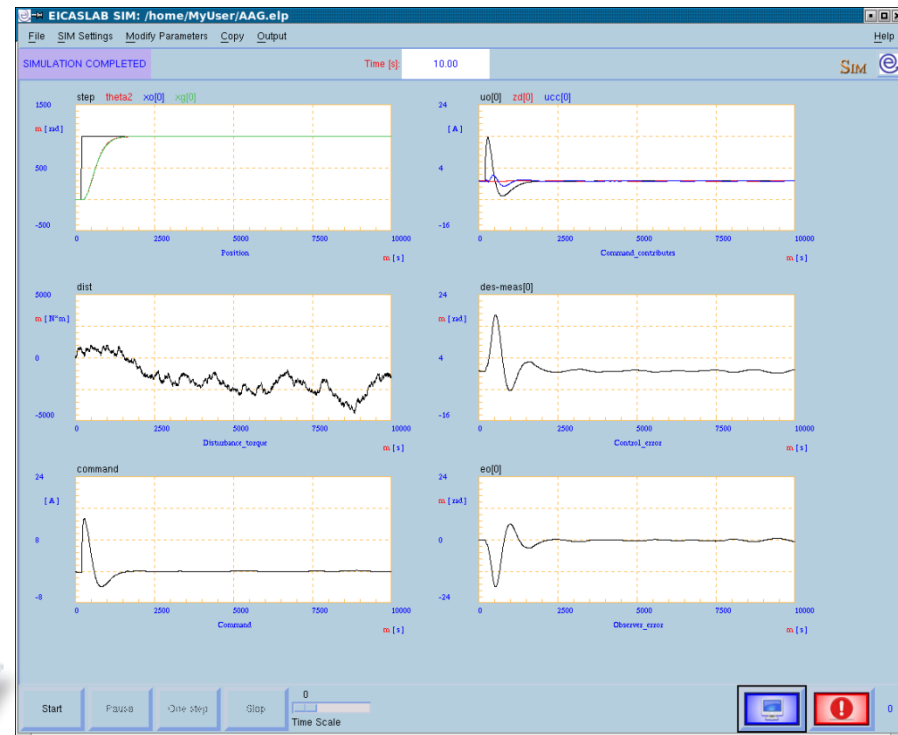
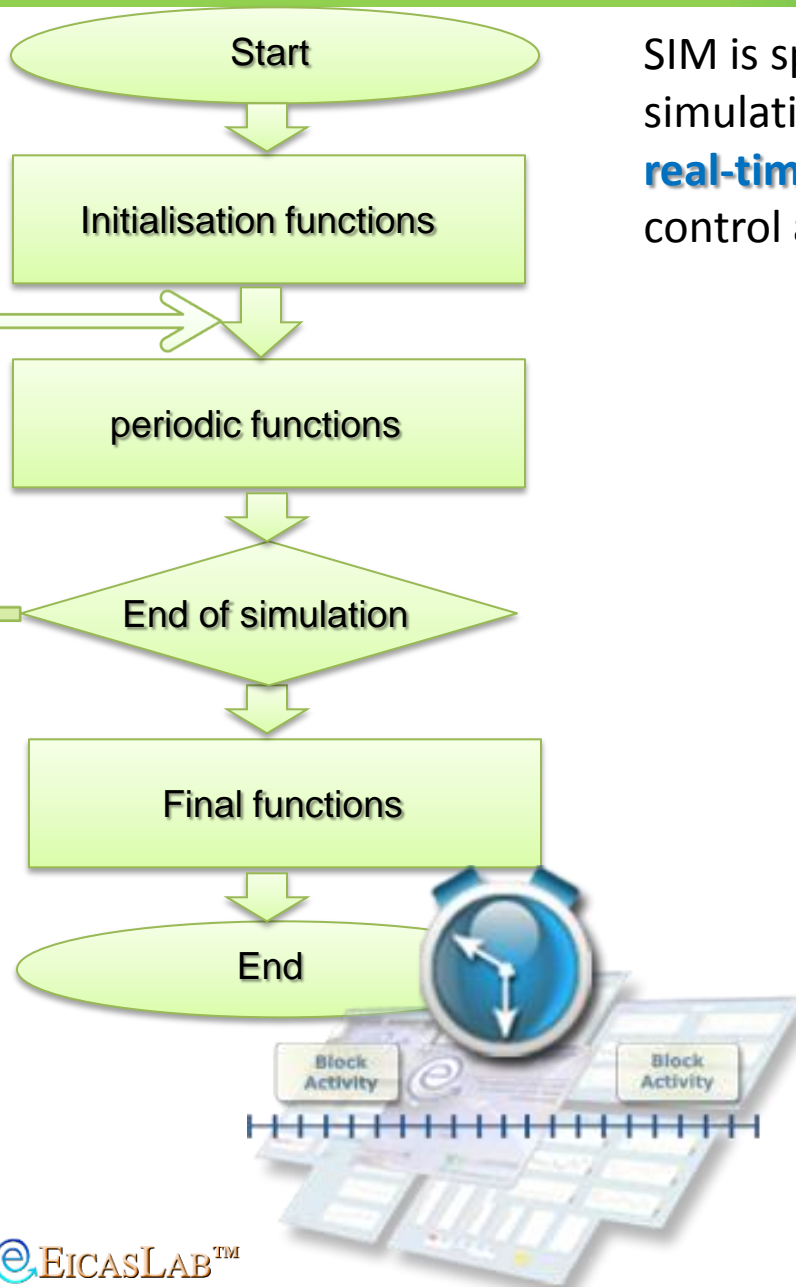
**SIM
TOOL**



excellence and passion
in automatic control design

EICAS Automazione S.p.A.

SIM is specifically devoted to perform a professional simulation of the control system. Through the “**like real-time**” technique, SIM realistically emulates the control architecture designed in SIMBUILDER.

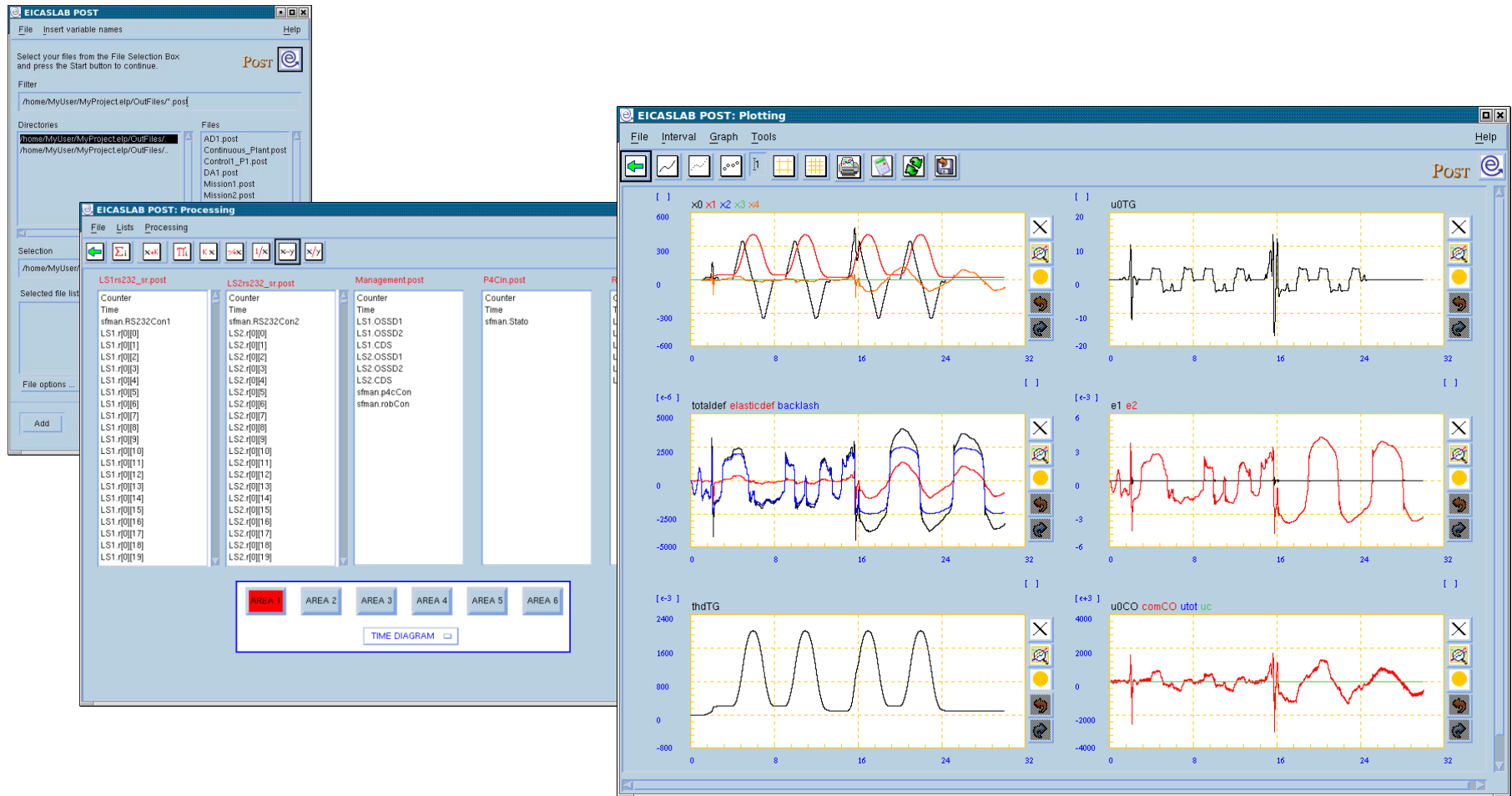


SIM allows to save on the PC disk all the desired variables for a further post-processing.

Post processing



POST is the professional tool of the EICASLAB suite specifically conceived to perform the post-processing of data recorded during the like-real time simulation phase or during the experimental trial on field.



Rapid Control Prototyping in a nutshell



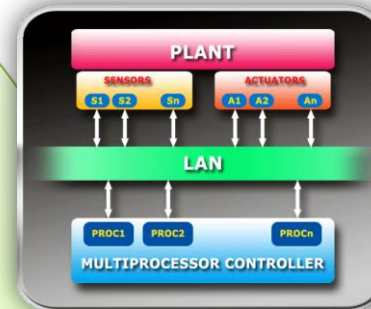
EICASLAB™



Slow Motion View



Multiprocessor
Architecture



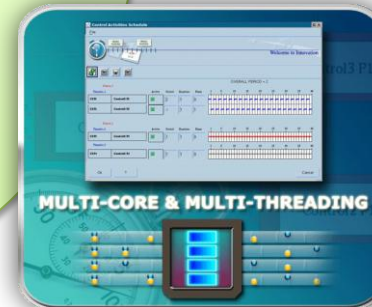
RAPID CONTROL PROTOTYPING



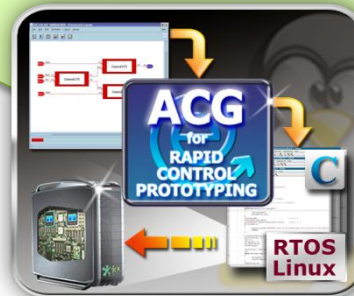
MULTI-CORE PC PLATFORM



Rapid Control Prototyping
on Field



Real-Time Scheduling

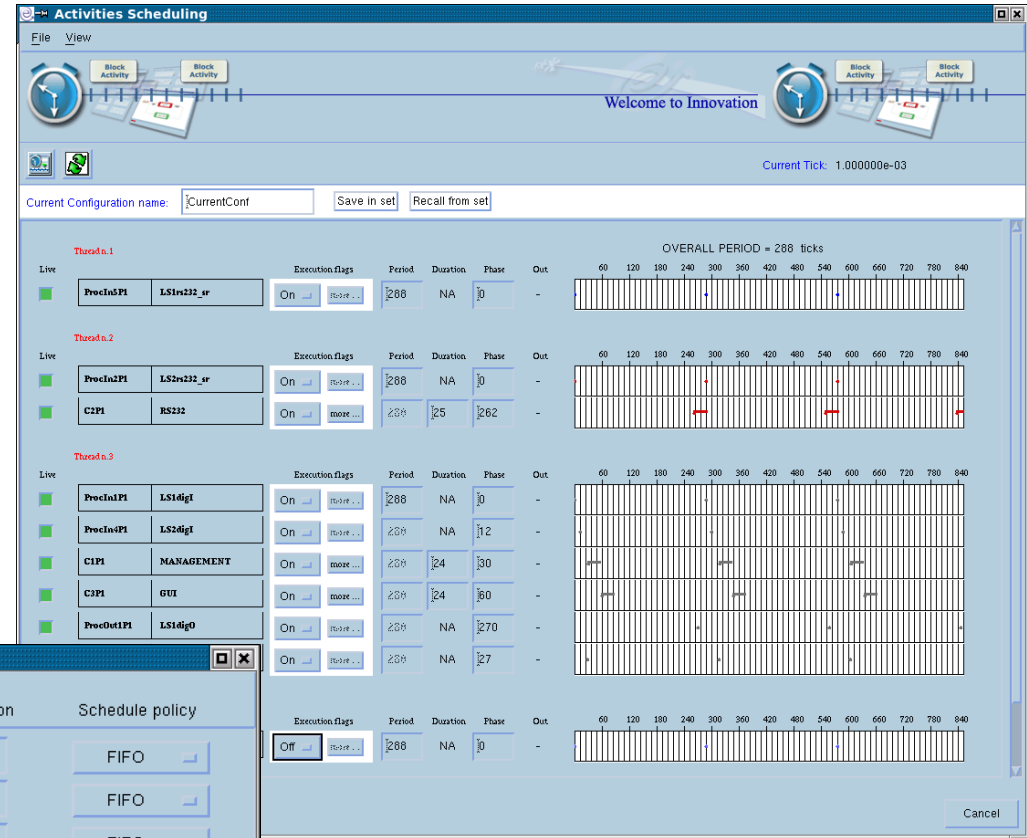
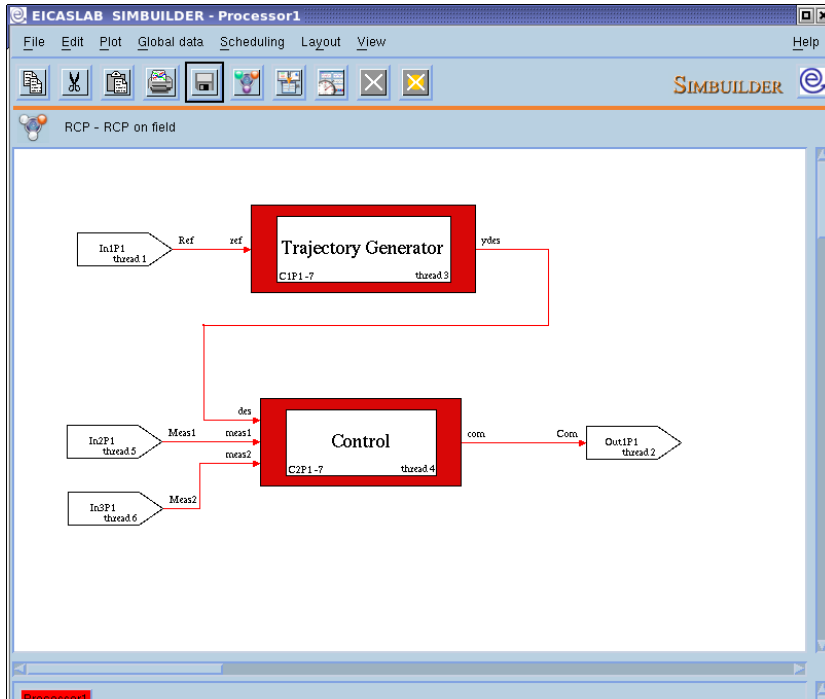


Activity scheduling

Rapid Control Prototyping

The activities are grouped on **threads**

The threads are assigned to the processors **core**



Thread	Assigned core	Priority	Hard/soft real time	Stack dimension	Schedule policy
Thread n.1	1	1	hard soft	1024	FIFO
Thread n.2	1	1	hard soft	1024	FIFO
Thread n.3	2	1	hard soft	1024	FIFO
Thread n.4	3	1	hard soft	1024	FIFO
Thread n.5	4	2	hard soft	1024	Round Robin

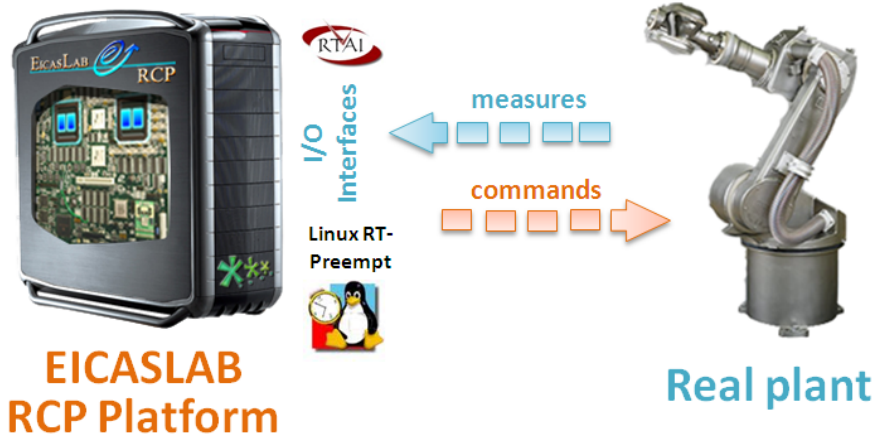
Rapid Control Prototyping Scenarios

STANDARD RCP

Classical RCP scenario

Rapid Control Prototyping scenario

Control algorithms



EXTENDED RCP

RCP with Target Bridge

RCP with Bridge Target scenario



Slow Motion View



**SLOW MOTION
TOOL**



excellence and passion
in automatic control design

EICAS Automazione S.p.A.

Slow Motion View analyses, step by step and variable by variable, the real-time control software run in experimental trials on the actual plant.

•Record I/O plant

**REAL-TIME RAPID
CONTROL
PROTOTYPING**

Host
commands



input
output

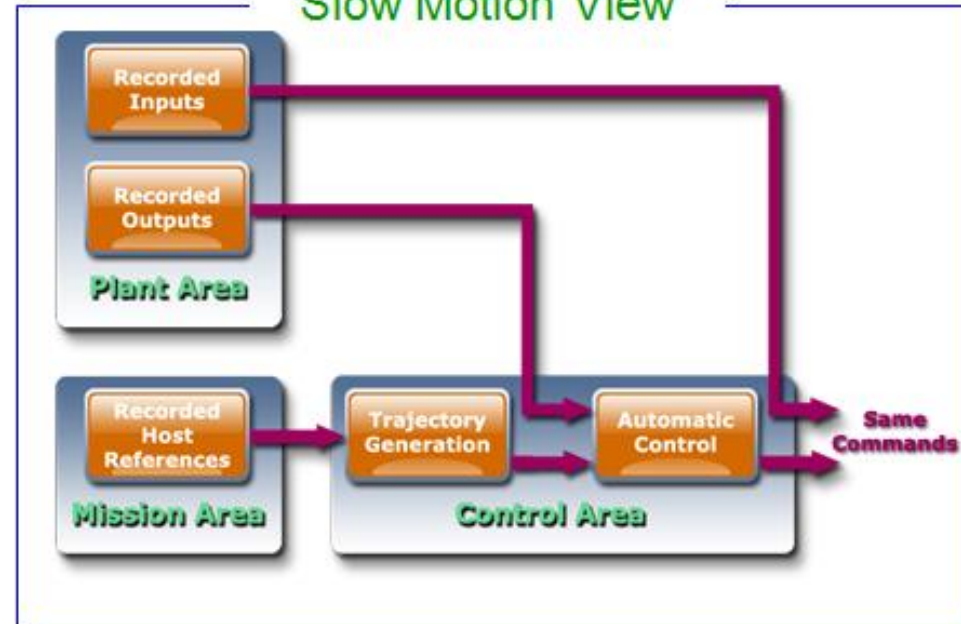


Repeat the trial as a
MOVIOLA with

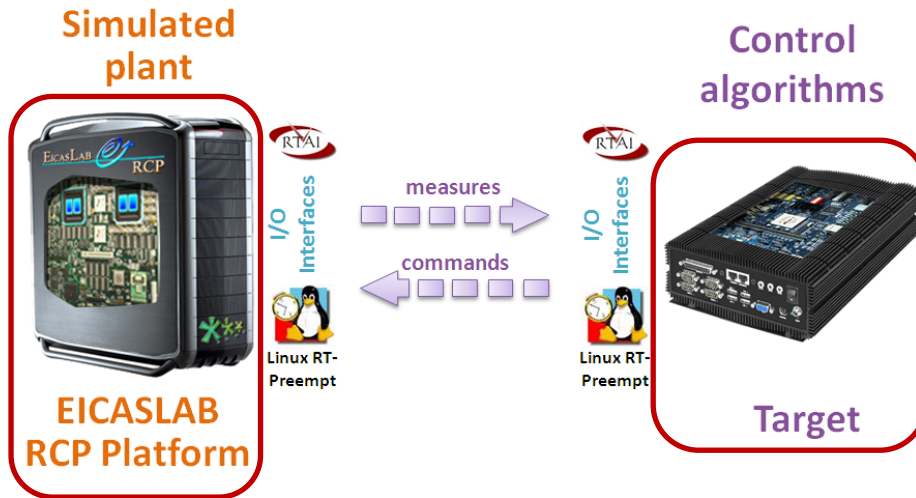
EICASLAB™



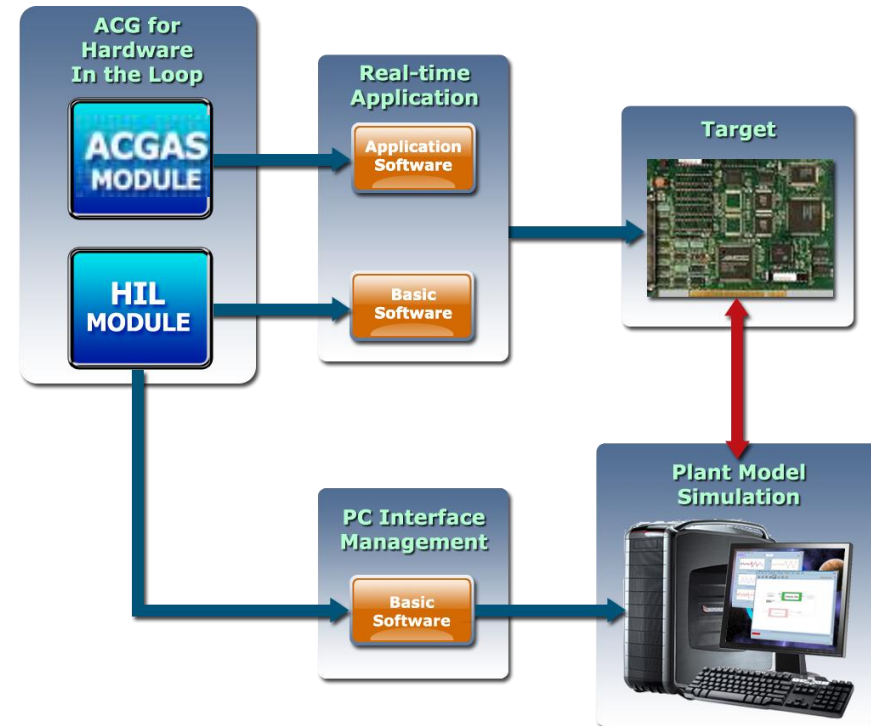
Slow Motion View



Hardware-in-the-loop

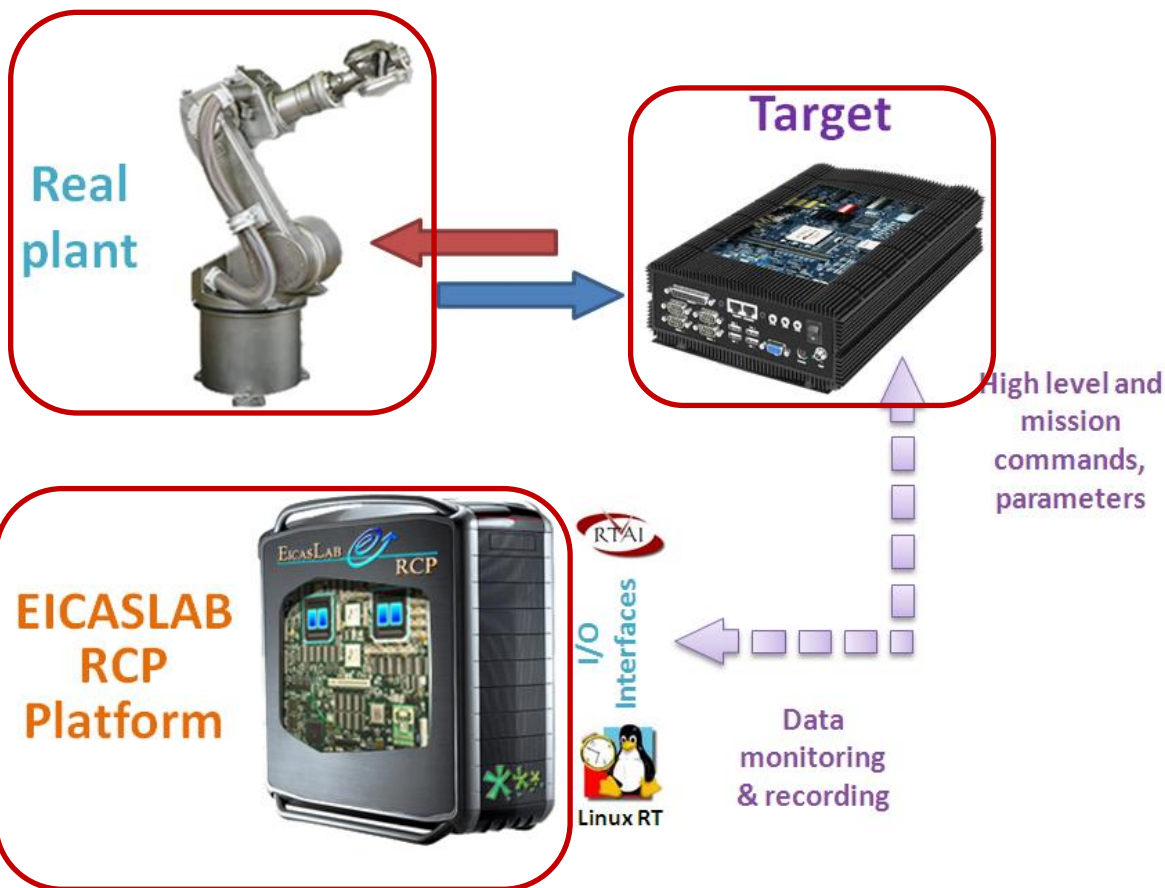


Once the control software code is installed in the final Hardware Target (ACG for specific Hardware Targets), the Hardware In the Loop (HIL) tests may be performed, consisting in piloting – instead of the actual plant - the plant simulated in **EICASLAB™** and running on your PC, suitably configured and connected through the necessary hardware interfaces with the final Hardware Target.





Final Validation Test scenario



Final Validation Tests can be performed, by connecting again your target on the RCP Platform.

In this case the EICASLAB RCP Platform can be used to provide commands to the target and for monitoring the target variables.

The MPI/CPO module is specifically conceived to enable

Model Parameter Identification



EICASLAB adopts an original identification method, oriented to estimate the optimal values of the simplified model from the control design point of view.

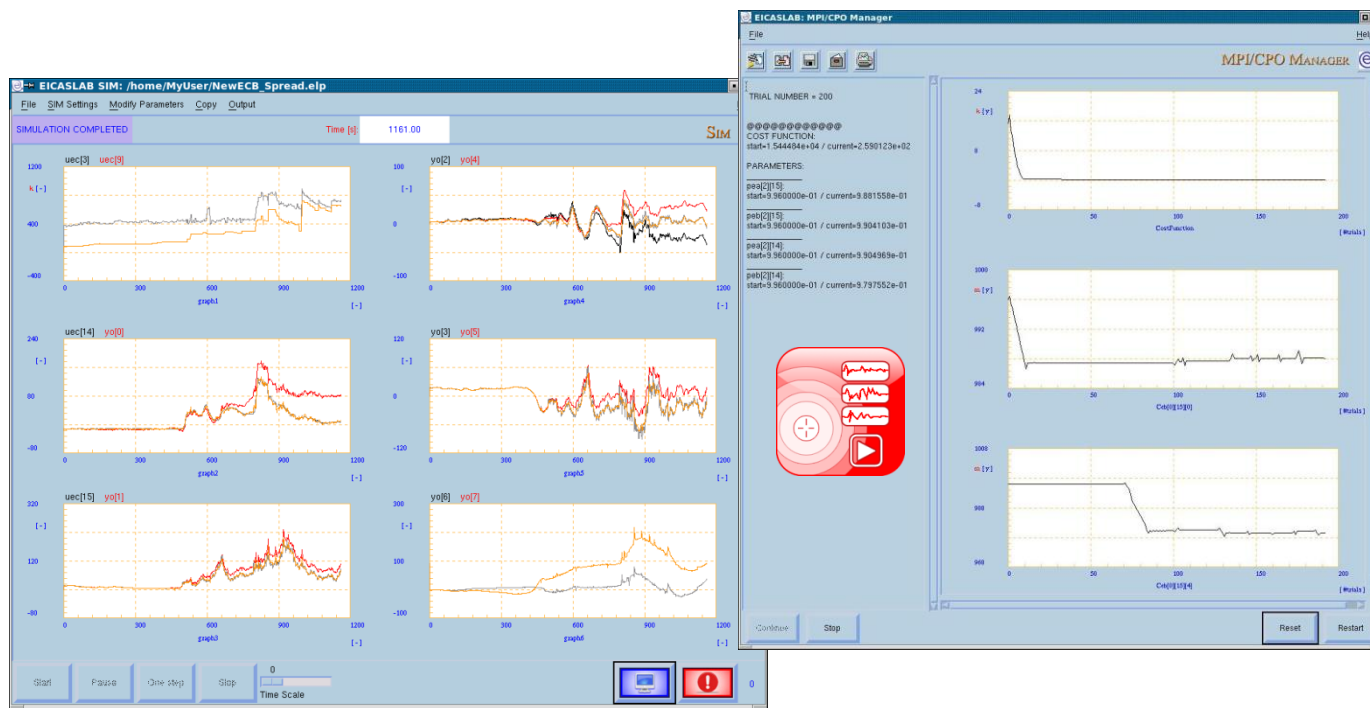
Control Parameter Optimisation

EICASLAB adopts a powerful algorithm of numerical optimisation that allows in short time to achieve the optimal values of a large set of parameters.



MPI/CPO Manager

is the EICASLAB tool that manages this phase



Identification and optimisation



The MPI/CPO module is available for being used in the following EICASLAB Operative Modes:

MPI/CPO MODULE

Modelling and
like Real Time
Simulation



➤ As Add-On of the SIM Tool

 Rapid Control
Prototyping



➤ As Add-On of the RCP Manager Tool

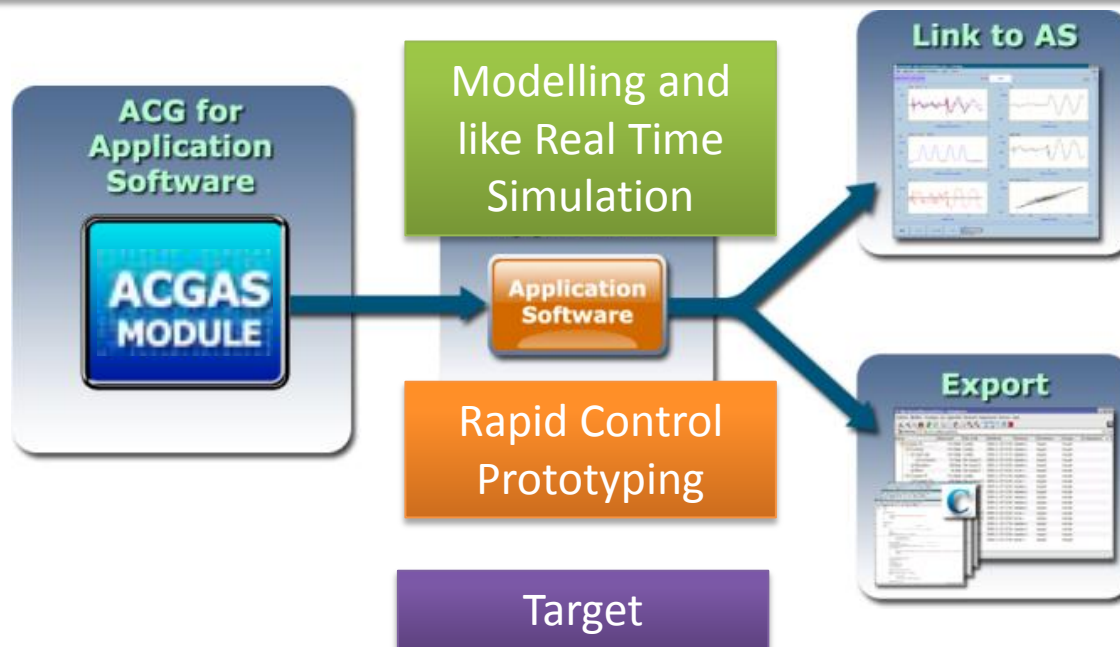
➤ As Add-On of the Slow Motion Tool

IT INCREASES YOUR
CAPABILITY TO OPTIMISE
YOUR CONTROL THROUGH
DATA ACQUIRED FROM THE
TRUE PLANT



ACG for Application Software (AS)

It allows to generate the AS, that can be tested in a simulated environment using the SIM tool. The AS, running in like-real time simulation (phase Modelling & Like Real-Time Simulation), **is the same one** that will run during the tests of Rapid Control Prototyping and of hardware in the loop and Final Validation Tests on the final HW target.



Automatic Code Generation: AS



excellence and passion
in automatic control design

EICAS Automazione S.p.A.

The screenshot displays the EICASLAB SIMBUILDER interface. On the left, a control block diagram shows a signal flow from an input 'u1' through a 'Cast' block, a summing junction, a gain block 'p1', an integrator 'Σ', and another 'Cast' block to an output 'c1'. A 'Block Setting: Data' dialog is open, showing parameters for the 'Cast' block. In the center, the 'GPDefines.h - KWrite' window shows the generated C code, including preprocessor directives and variable definitions. On the right, the 'Control1_P1.c - KWrite' window shows the main C code, including an attention notice, includes, static variable definitions, and the main function logic.

```
#define DEF_CTR1P1
#include <Standard.h>
#include <GenHead.h>
#include <Common.h>
#include "../Interface/DBInterface.h"
#include "DBInterface.h"

#include "../Interface/Prototypes.h"
#include "ACGPrototypes.h"
#include "DBP.h"

/* Definition of the static variables */
static EL_DOUBLE y0 1;
static EL_DOUBLE x0 1;
static EL_DOUBLE xn0 1;
static EL_INT y1 1;
static EL_DOUBLE y2 1;
static EL_DOUBLE y3 1;
static EL_DOUBLE y4 1;
static EL_DOUBLE p4 1;

/* Initialization of the parameters of the block p1 (id=4) */
p4_1=1.000000e+00;
```

Block Setting: Data

BLOCK INFO	INPUTS	OUTPUTS
Name=Cast Id Number=0 Input number=1 Output number=1 State number=0 Parameters number=0	EL_INT	EL_INT

GPDefines.h - KWrite

```
#ifndef GP_DEFINES_H
#define GP_DEFINES_H

/* Type of variables */
#undef EL_INT
#undef EL_FLOAT
#undef EL_DOUBLE
#undef EL_CHAR
#undef EL_UCHAR
#undef EL_UINT
#undef EL_INT8
#undef EL_UINT8
#undef EL_INT16
#undef EL_UINT16
#undef EL_INT32
#undef EL_UINT32
#undef EL_INT64
#undef EL_UINT64
#undef EL_INTLL
#undef EL_UINTLL

#define EL_INT int
#define EL_FLOAT float
#define EL_DOUBLE double
#define EL_CHAR char
#define EL_UCHAR unsigned char
#define EL_UINT unsigned int
#define EL_INT8 char
#define EL_UINT8 unsigned char
#define EL_INT16 short
#define EL_UINT16 unsigned short
#define EL_INT32 int
#define EL_UINT32 unsigned int
#define EL_INT64 long long
#define EL_UINT64 unsigned long long
#define EL_INTLL long long
#define EL_UINTLL unsigned long long

#define FLOATING double
```

Control1_P1.c - KWrite

```
ATTENTION:
THIS FILE IS AUTOMATICALLY GENERATED BY EICASLAB:

#define DEF_CTR1P1
#include <Standard.h>
#include <GenHead.h>
#include <Common.h>
#include "../Interface/DBInterface.h"
#include "DBInterface.h"

#include "../Interface/Prototypes.h"
#include "ACGPrototypes.h"
#include "DBP.h"

/* Definition of the static variables */
static EL_DOUBLE y0 1;
static EL_DOUBLE x0 1;
static EL_DOUBLE xn0 1;
static EL_INT y1 1;
static EL_DOUBLE y2 1;
static EL_DOUBLE y3 1;
static EL_DOUBLE y4 1;
static EL_DOUBLE p4 1;

/* Initialization of the parameters of the block p1 (id=4) */
p4_1=1.000000e+00;
```

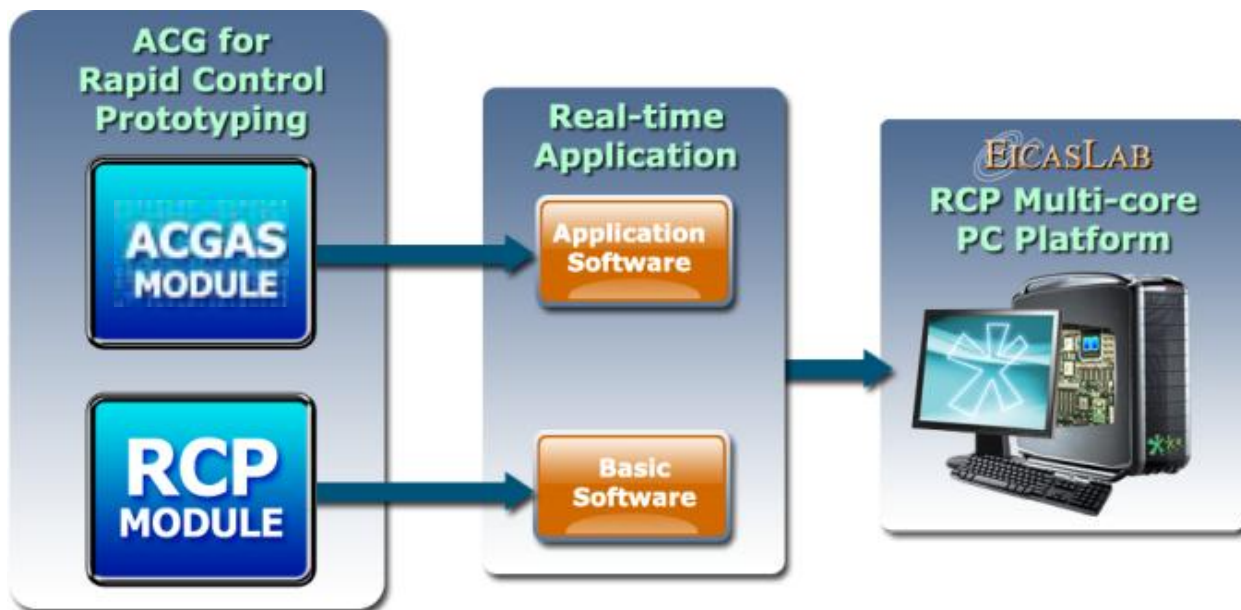
Example of
generated
templates





ACG for Rapid Control Prototyping (RCP)

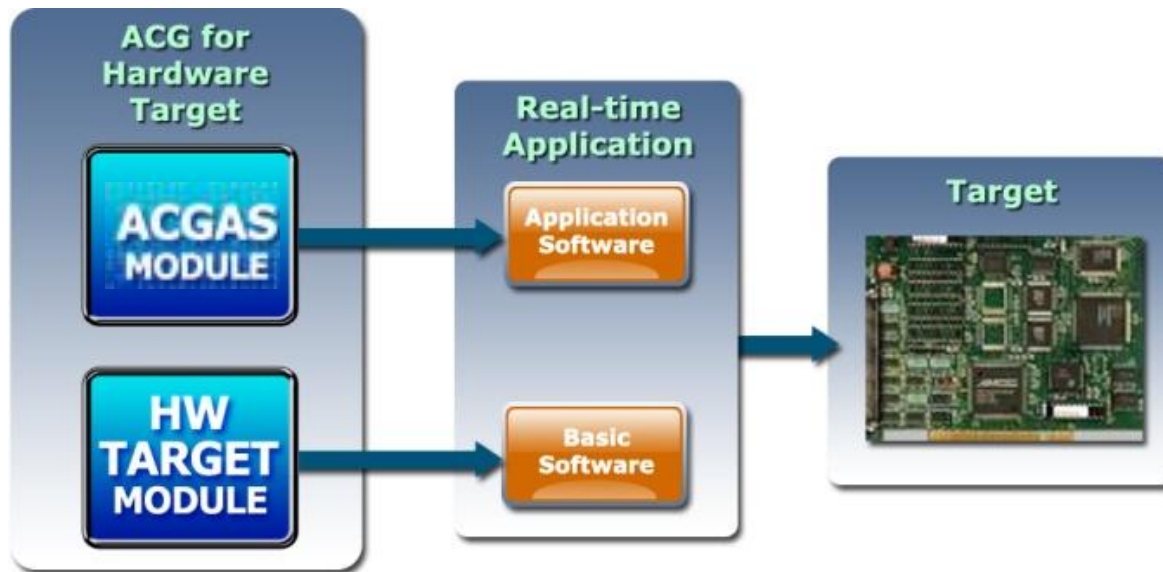
It allows to generate the complementary source code (BS=Basic Software) that, together with the AS, provides a Multi-core and Multi-threading RT application able to perform a complete RCP test.





ACG FOR THE FINAL TARGET

It allows to generate the target-dependent source code that, together with the AS, provides the real time program for desired HW target.

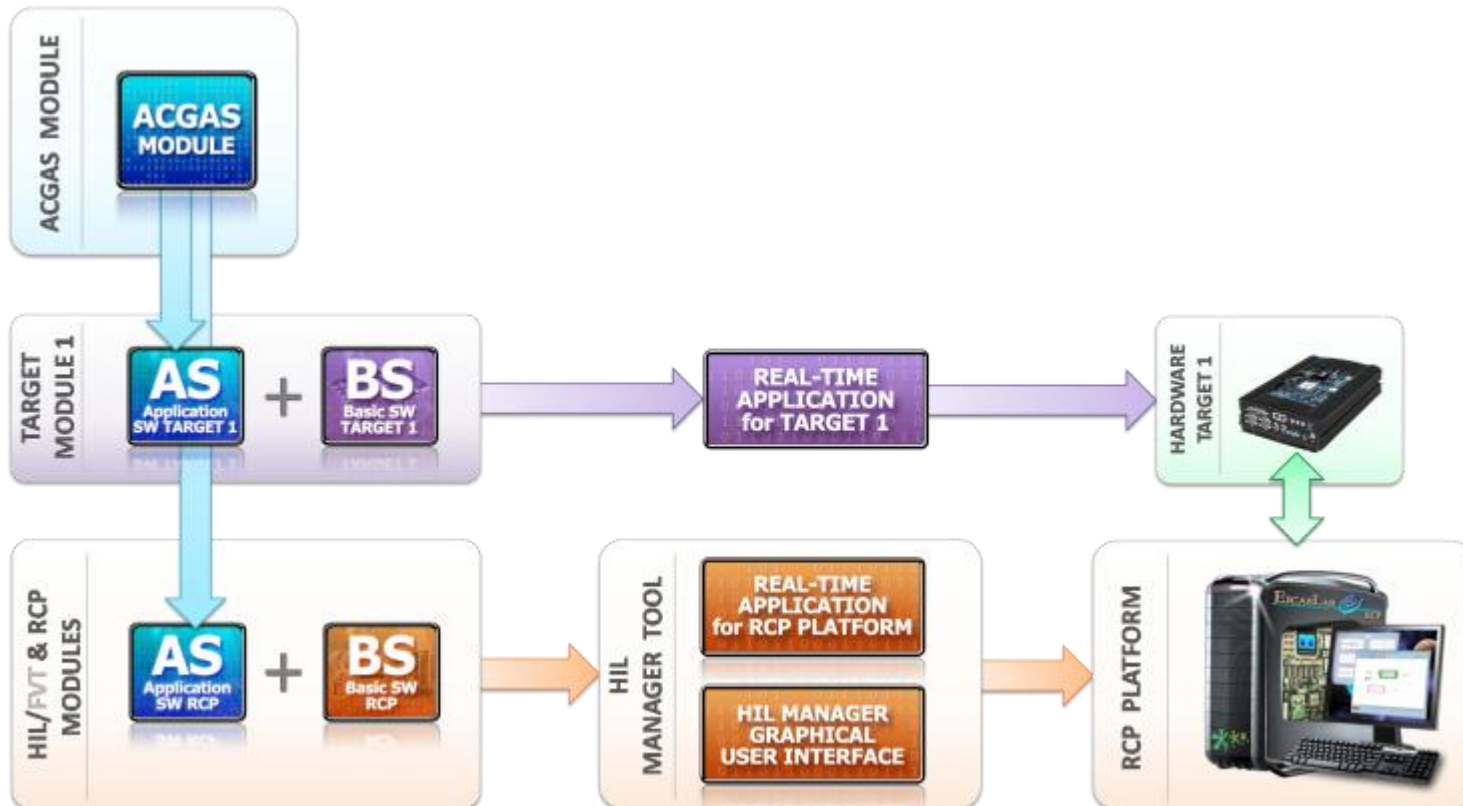


Automatic code generation for HIL



ACG FOR HARDWARE-IN-THE-LOOP

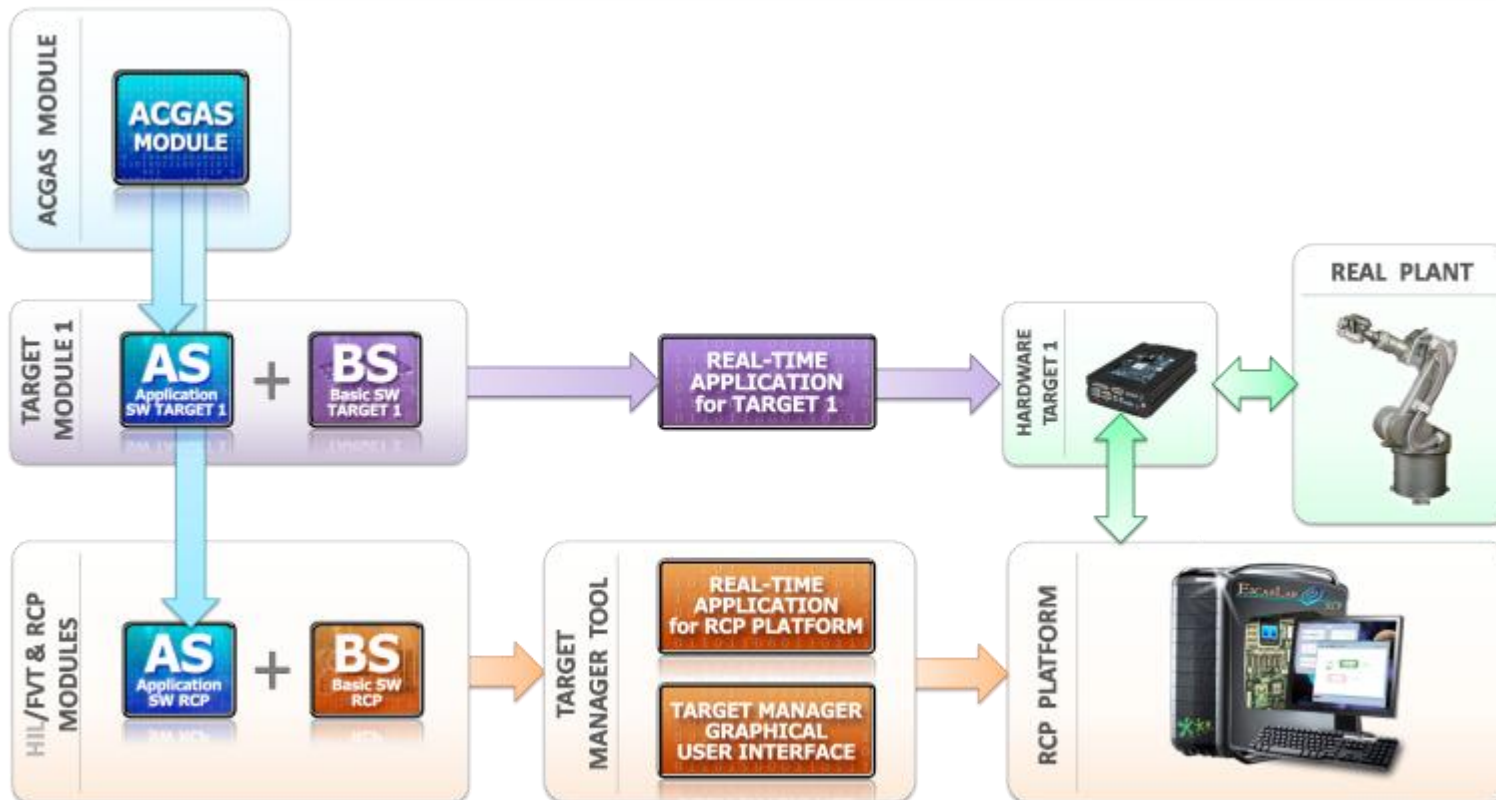
It allows to generate the BS which, together with the AS, allows to perform the Hardware-in-the-loop activities.



Automatic code generation for FVT

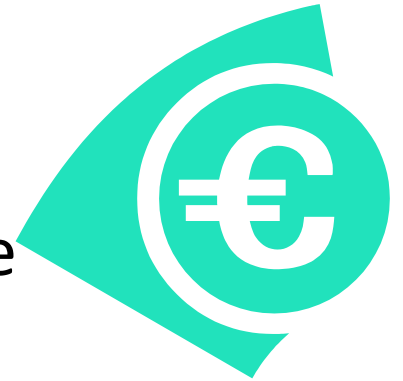
ACG FOR THE FINAL VALIDATION TESTS

It allows to generate the BS which, together with the AS, allows to perform the Final Validation Test activities.



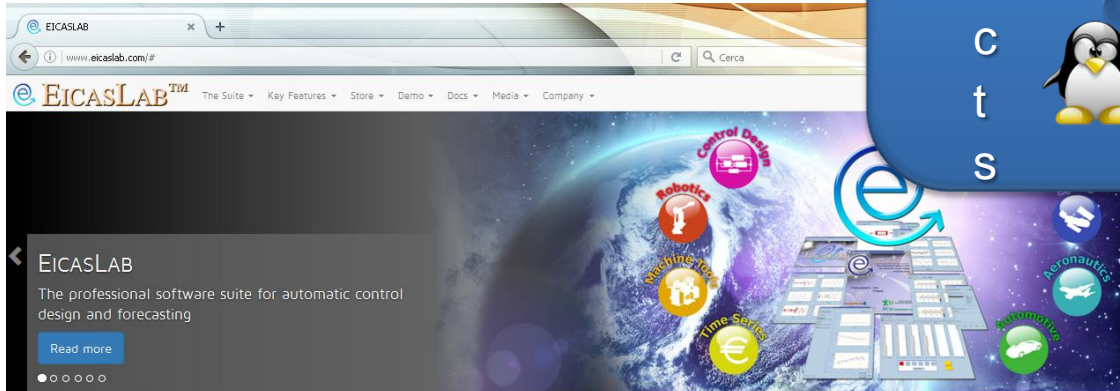
EICASLAB™

- ✚ Just one suite, just one project!
- ✚ No need for deep mathematical knowledge
- ✚ Minimise the time to develop the control algorithm.
- ✚ Reduce costs of the control design.
- ✚ Design starting from plant datasheet only.
- ✚ No set-up in field.
- ✚ Larger freedom to the designer.
- ✚ Increase performance specially in complex control cases.



EICASLAB™ Commercial web site

 www.eicaslab.com



P
r
o
d
u
c
t
s



for Linux



&



for Windows






Rapid Control Prototyping

Enjoy a smart, professional, fast, easy-to-use and low cost rapid control prototyping through EICASLab.

You just need your PC with installed a Linux RTOS, the EICASLab suite and suitable HW interfaces, directly available in your PC or by using your final target as a bridge!

[View details >](#)



Hardware-in-the-loop

Automatically generate the software code for your final target and perform professional hardware-in-the-loop tests through EICASLab.

In this way you can check the correct installation of your software code in the target.

[View details >](#)

Tweets by @eicaslab

 **EICASLAB** @eicaslab
Easy and fast #ControlSystem #Prototyping on a @rasberry_pi with #EICASLAB. Try our demo: eicaslab.com/DemoEICASLab...

