AUTOMATED COMPUTING MACHINERY S.L.
c/Playa de Lagoa,1
28660 Boadilla del Monte Madrid
Phone: +34 605876613 Fax: (+34) 91 6323812
www.acm-sl.com

Document:

SHORT USER'S MANUAL

FOR THE

SIMULATOR-TRAINER

SIM1300

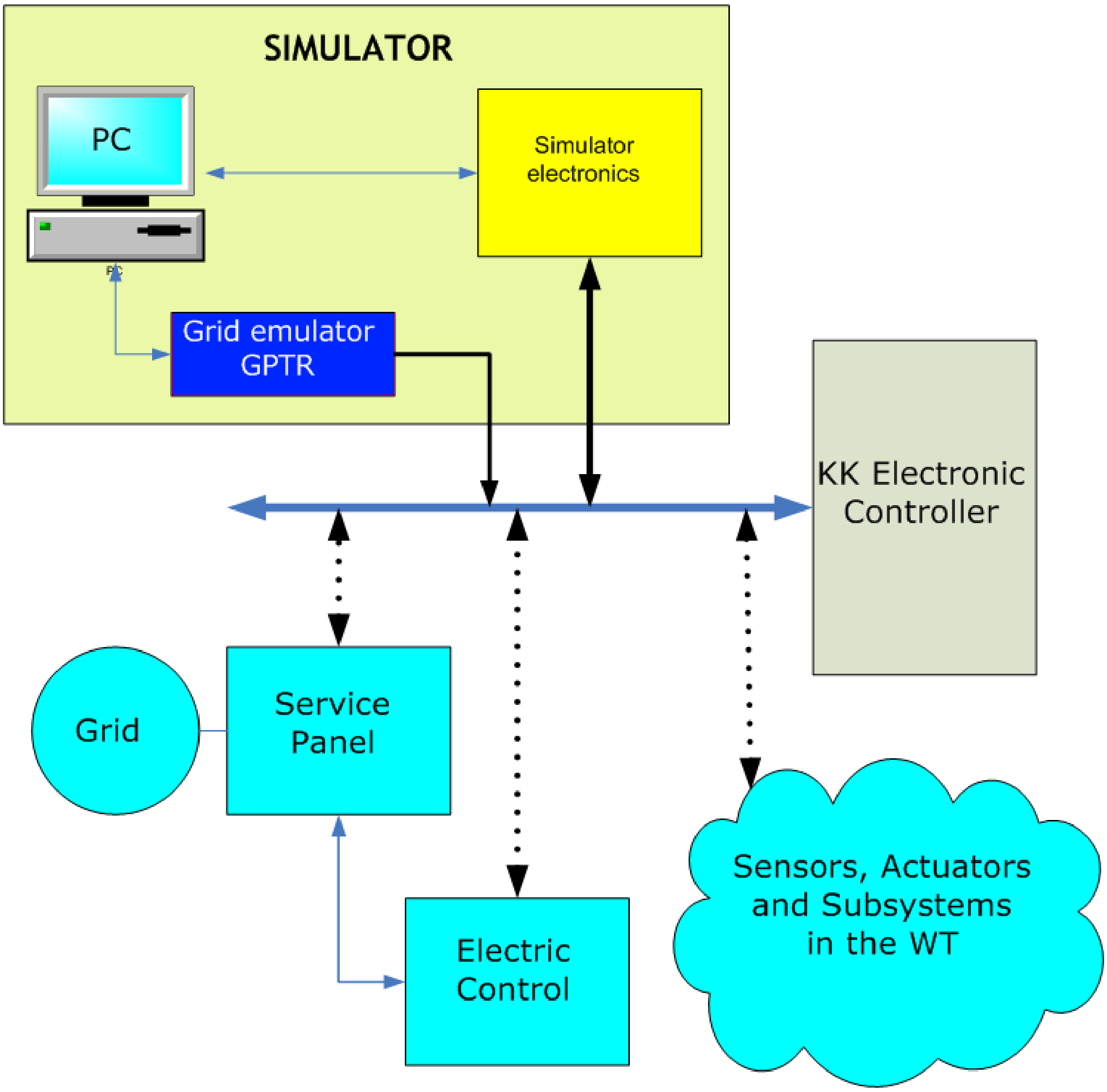
15/Octubre/2007

Contents

Photo

Front view of the Simulator Trainer (Monitor not shown)

Conceptual Building Block diagram



Dual view

The real components of the wind turbine (represented in blue) that are connected to the (two) KK Elektronic Controllers (bottom and top), are replaced by a set of electronic instruments (yellow and dark blue) and a control program running on a PC. This ensemble simulates the behavior of the real components. This is the Simulator. When executed, it read the outputs of the controllers KK Controllers and acts accordingly, maneuvering and setting the input of those Controllers, so that they "think" they are connected to the real components of the wind turbine.

Introduction

The purpose of this document is to serve as a brief reminder of the most typical operations that can be performed by the SIM1300, so that it is easy to start it up. This is, by no means, an exhausting listing of all its possibilities. Check please the “**SIM1300 User’s manual**” for a complete description.

Reminder of functionalities

The main functionalities of the Simulator are:

1. To read the output signals coming from the Controller.
2. To generate all the signals that are input into the Controller, it such a way that they mimic the corresponding signals from the real component. To do so it must take into account the output signals of the Controllers and the desired setting of the user of the Simulator.
3. To present the values of all of these signals to the user in a easy to understand way.
4. Generate the evolution of the signals following the instruction of user, the Controller and the associated behavior of every subsystem, so that the Controller “sees” a real wind turbine.

The program is fully configurable to accommodate for the hardware blocks in place as well as the user interface (HMI) definition. In this document the configuration used is prepared for a **BONUS 1300 wind turbine**.

Setting starting conditions

As shown in the photograph before, the system is host in a 19” rack and has two mains switches: one for the general power (located right most) to all instruments and subsystems, and another specifically for the instruments and the controllers (Instruments). The PC is switched off only with the general switch.

Additionally the special instrument GPTR (grid emulator) has its one security switch, located in the back of it. The PC also has its only switch located in its front panel.

So for starting the SIM1300, please follow the next steps:

1. Switch on the General Switch.

For that, uncover the control panel (using the supplied key) and press the “Start” button.

2. *Loggin.*

The operating system (Windows XP) will present the form for user identification. Use the *userid* and *password* supplied by the System Administrator of the PC.

3. *Power up the instruments and controllers.*

Check that the *Instruments power switch* is already connected: switch it on otherwise.

4. *Start the Terminal emulator*

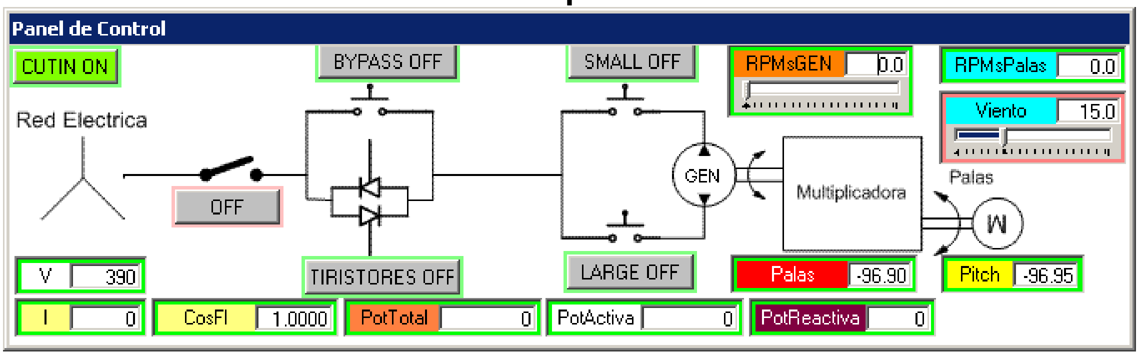
1. In the "Desktop" you can find a programs group named "ACM". Inside it you will find a set of icons.
   Look for and icon named "TerminalKK" and execute it. It will present a terminal emulator similar to the portable KK Electronics Terminal for the Wind turbine (it contains a display of 4 lines , 40 characters per line and a keyboard).
2. Check that this terminal shows the typical message shown in a real portable terminal connected to the wind turbine. Otherwise reset the Controllers.

5. *Execute the Simulation program*

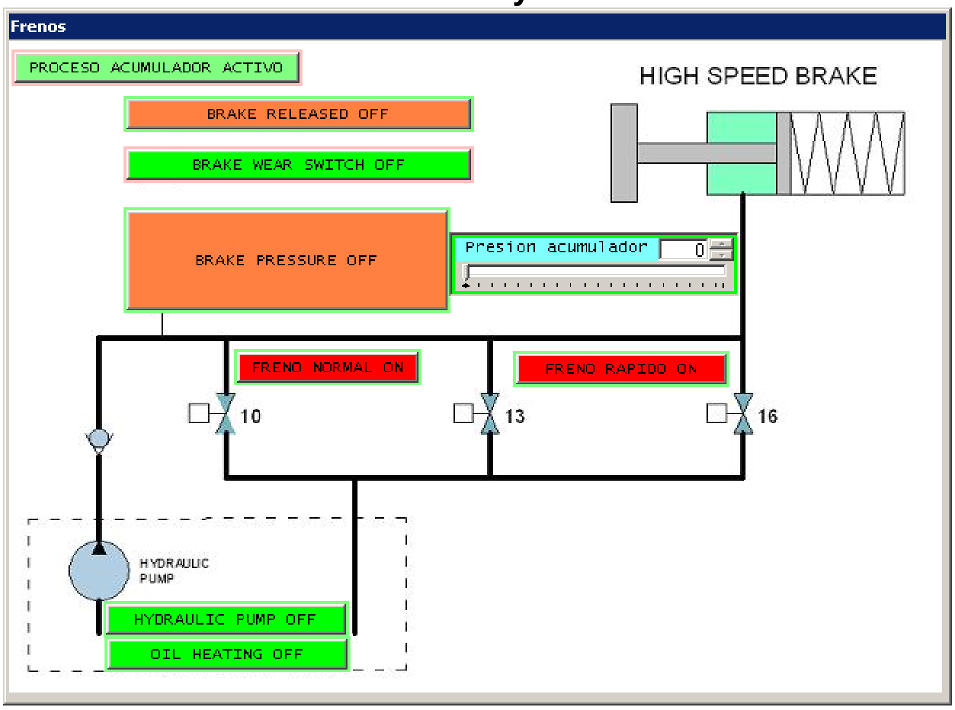
1. Look for an icon named "Simulador 1300" and execute it. This is the main program.
2. A form will appears that contains a menu where you can select the option for specifying a "configuration" file. Select the configuration file that the administrator gave to you. This configuration file is adapted to the set of forms (sinoptics), emulation algorithms and physical instruments in this installation.
3. You will be presented with a list of previously saved "sessions" in the form of a list. Sessions allow the user to save a set of values that will be used as starting setting for the new session. Choose the session that best suits your purpose.

At that moment, the program will present you the set of forms (synoptic like), located in the screen places and with the variable's values, according to the session file selected. For example, you may be looking at the "Control Panel" (showing the electric connections and their status), the "Brake subsystem synoptic" and the Wind and Nacelle Orientation ("Viento y Orientacion" "Wind and Orientation"), as can be seen in the next figures.

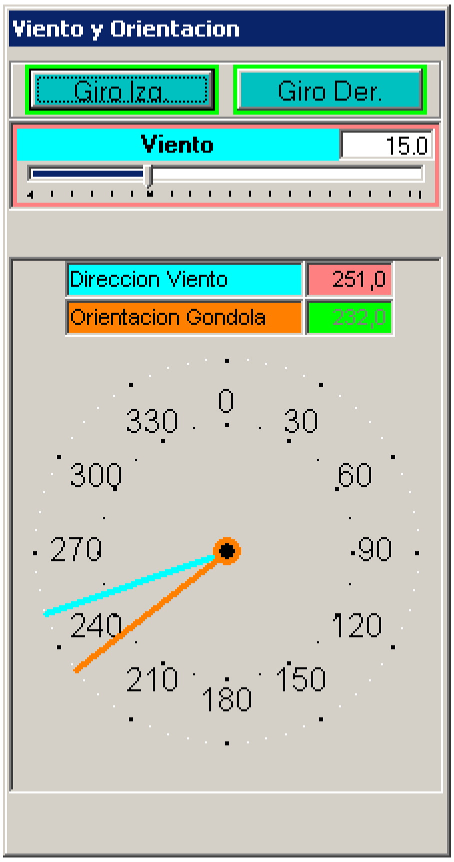
Control panel



Brake subsystem



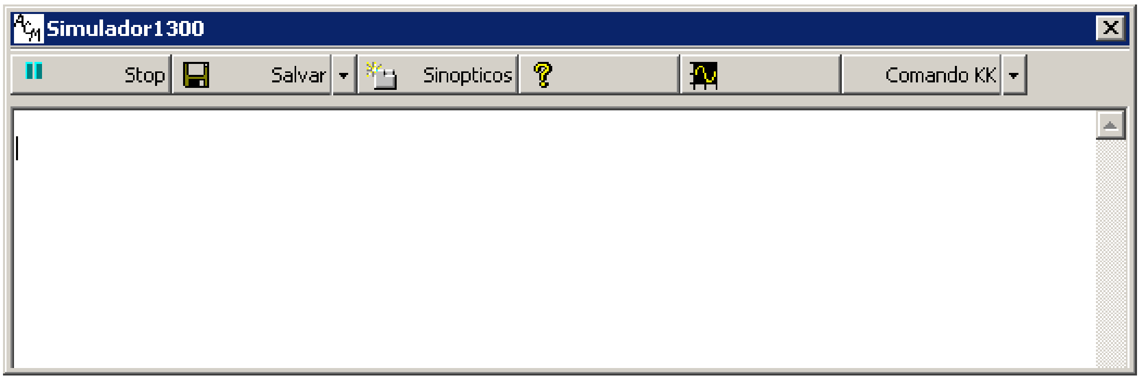
Wind and Nacelle orientation



6. Start the simulation

1. So far what happened is that all the outputs from the Controllers are reflected in the "connected" graphic widget in the synoptic, and that all the signals input to the Controllers are set to the values specified in the session file.

   You can now "start" execution: look to the form named "Simulador1300" and press the left most menu button (showing "*Ejecutar*"). Inmediatly this button will change to "Stop" (a multipurpose button) and all the simulation will actually start.



2. Next, you should turn on the (emulatred) Electrical Grid.

   The GPTR (Generador Programable de Trifásica) is the instrument, located in the bottom of the 19" rack, that actually creates a triphase grid. By default, when switched on, this equipment generates 0 V in each phase, for security reasons. So you must "connect" it to the Simulator (similar operation to connecting the Mains switch in a real wind turbine).

   This can be checked in various ways:

   a. Through the measurements done by the controllers, that can be seen through the Terminal Emulator (menu 15, submenu 3);

      or

   b. Going to the "Control Panel" to see that the widget (digital) below the switch connecting to the grid is showing "OFF". In this Control Panel you can see also the actually value of the average voltage in the (analog) widget mark "V": should be 0.

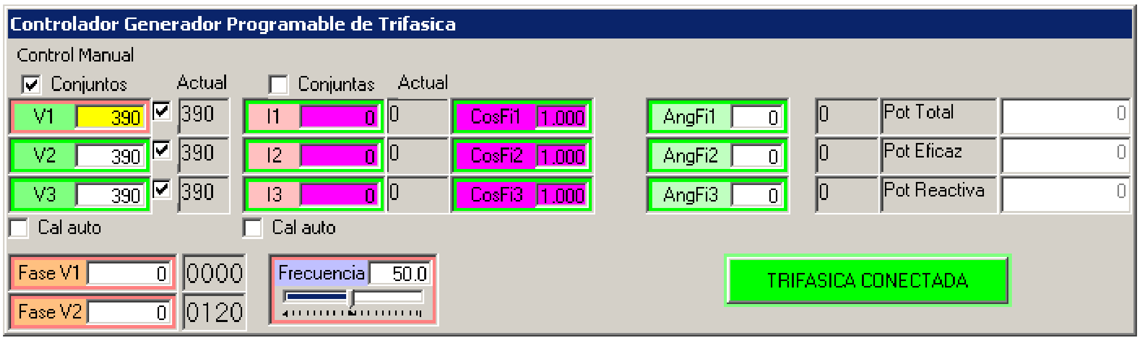
      or

   c. Checking in the "GPTR Control Panel"

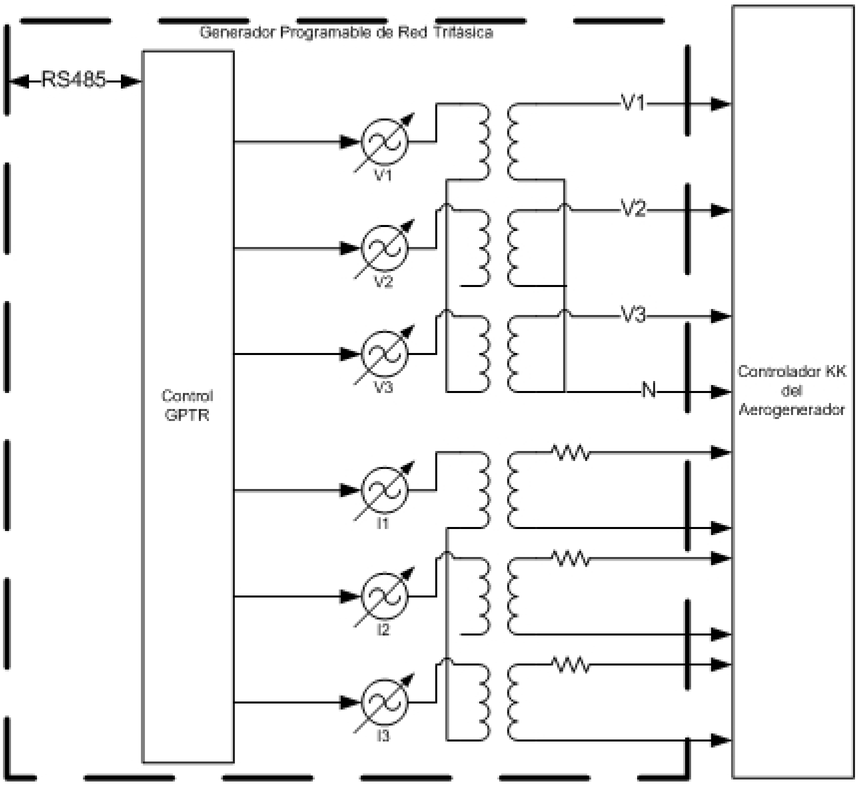
      In this synoptic you can see voltage that will be actually applied to measuring inputs of the controllers, as well as the remaining electrical variables (the simulated grid).

      To connect the grid to the simulator, simply press the digital widget marked OFF in the "Control Panel": automatically the GPTR will generate the voltages indicated in its "GPTR Control Panel", and this widget will change to "ON"

      In this stage, the Simulator should be applying a set of values to the Controllers coming from the selected former session. Values such as wind, orientation, etc.



Next figure shows the block diagram of the grid subsystem generator.



Playing with the system

You can now play with the system in many ways. Examples are:

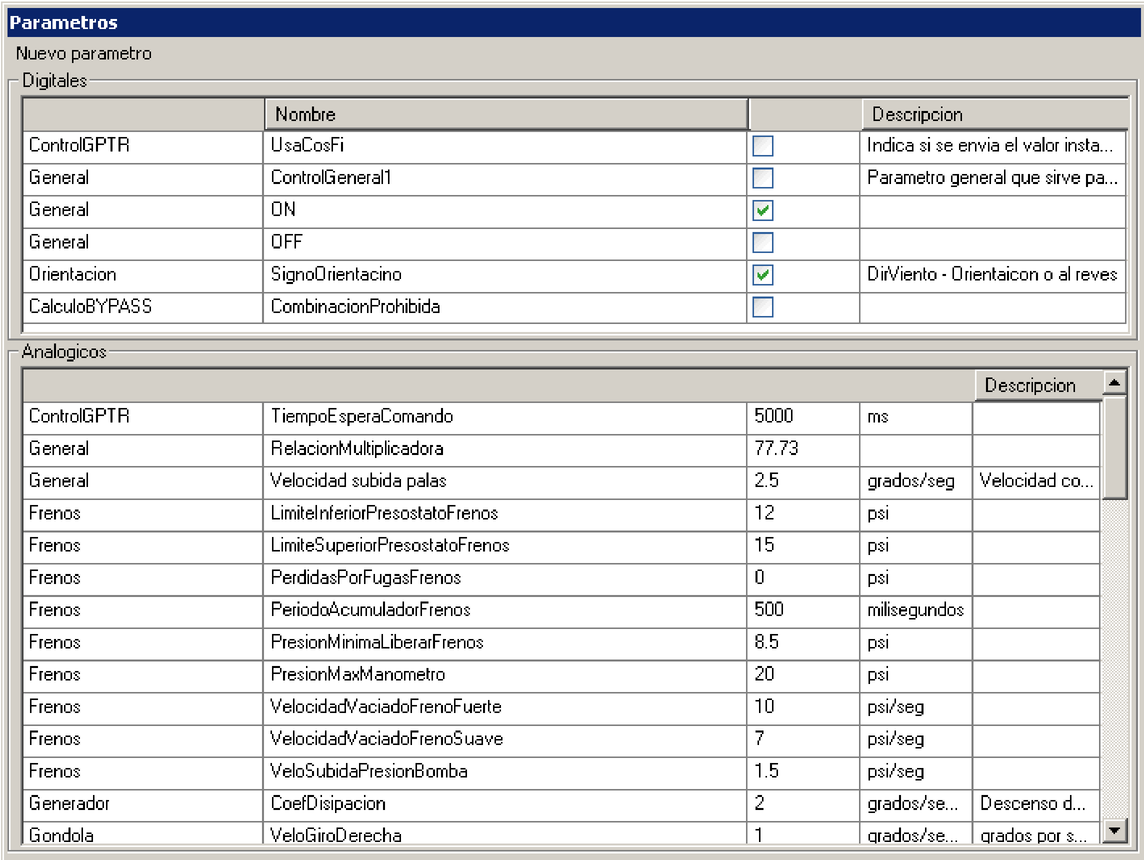
1. Checking the capability of self orientation of the wind turbine

Change the value of the wind speed using the "Viento y Orientacion" synoptic so that it is higher than the minimum value set up in the parameters as "minimum wind". You can change it with the slider or directly keying the desire value (the entry is error insensitive).

Change them the angle of the wind, using the blue needle in the same synoptic (place the mouse on top of it, press left button and drag), so that this angle is very different from the indicated nacelle orientation.

Wait to see how the Controller takes into consideration the preset time to command the yaw motors: the status of these signals are indicated by the two (digital) widgets mark "Giro derecha"(yaw CW), "Giro izquierda" (yaw CCW).

Check that the value of the nacelle orientation changes accordingly to the status of these signals and the values of the parameters "VeloGiroDerecha" (speed for yawing CW) y "VeloGiroIzquierda" (speed for yawing CCW). You can check the actual values for these two parameters as well as change them in the form "Parametros". Those parameters take into consideration the possibility (a faulty situation) that the two yaw motors behave different or badly (i.e. setting one of these parameters to 0, will prevent the nacelle to yaw in that direction, causing – hopefully- a fault).



Parametros

Nuevo parametro

Digitales

	Nombre		Descripcion
ControlGPTR	UsaCosFi	[ ]	Indica si se envia el valor insta...
General	ControlGeneral1	[ ]	Parametro general que sirve pa...
General	ON	[x]	
General	OFF	[ ]	
Orientacion	SignoOrientacino	[x]	DirViento - Orientaicon o al reves
CalculoBYPASS	CombinacionProhibida	[ ]	

Analogicos

				Descripcion
ControlGPTR	TiempoEsperaComando	5000	ms	
General	RelacionMultiplicadora	77.73		
General	Velocidad subida palas	2.5	grados/seg	Velocidad co...
Frenos	LimiteInferiorPresostatoFrenos	12	psi	
Frenos	LimiteSuperiorPresostatoFrenos	15	psi	
Frenos	PerdidasPorFugasFrenos	0	psi	
Frenos	PeriodoAcumuladorFrenos	500	milisegundos	
Frenos	PresionMinimaLiberarFrenos	8.5	psi	
Frenos	PresionMaxManometro	20	psi	
Frenos	VelocidadVaciadoFrenoFuerte	10	psi/seg	
Frenos	VelocidadVaciadoFrenoSuave	7	psi/seg	
Frenos	VeloSubidaPresionBomba	1.5	psi/seg	
Generador	CoefDisipacion	2	grados/se...	Descenso d...
Gondola	VeloGiroDerecha	1	grados/se...	grados por s...

NOTE:

There are many ways to *force* operational faults. The simplest one is to remember that many of the *widgets* in the synoptic has **dual purpose**:

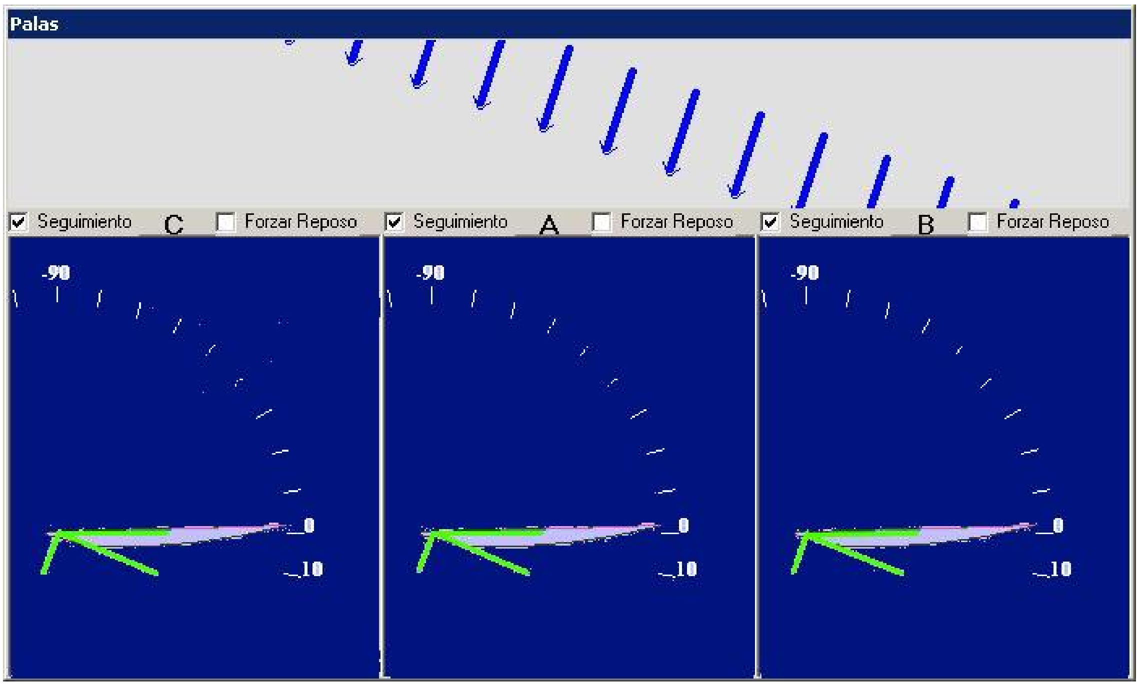
a) As an *indicator* (showing the value of a variable of the system) and

b) As a tool to change the value of that same variable (**indicator and** *control*).

Sometimes this dual behavior is forbidden by the internal settings; in any case you can see in which mode a given widget is, checking the color of the little border around it: if *green* then the mode is in “indicator” mode (the user can not modify it manually), if *red*, is in “control” mode. If not forbidden by the internal settings of the synoptic, you can change the behavior of any widget pressing “**F6**” on it: you will be prompted to provide a password to enter this “setting” mode. That can be done in any moment, even while the execution of an emulation.

In order to fully understand the influence of each widget, you have to know how the internal variables are connected among each other and with the widgets. Go to the "SIM1300 User's manual" for a complete description of this matter.

An interesting way of looking the values of the wind speed and the difference between the wind direction and the nacelle orientation is to look at the upper part of the "Blades" synoptic.



The *higher the wind speed*, the *bigger the density of the arrows*. The direction is the difference between the wind direction and the nacelle orientation (*disorientation* degree of the nacelle).

In this synoptic are represented (different for each blade) the actual angle of the blade and the incoming wind, the rotational speed wind and the resulting effective wind and the attack edge (lower part of the synoptic).

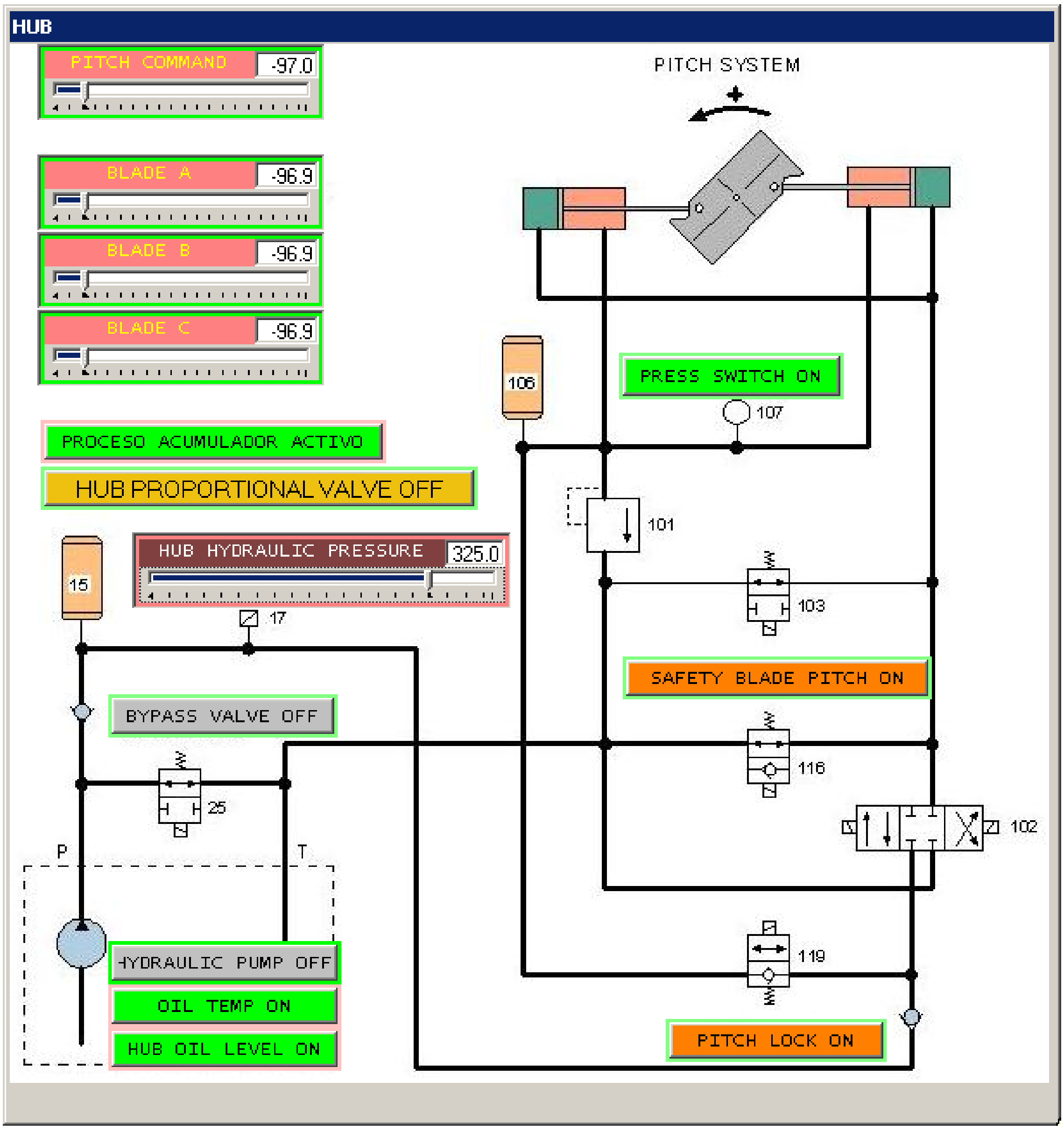
2. Start production

Assuming that there are no errors pending in the Controllers (that implies that the emulated grid is connected) and the wind speed is above the minimum limit, you can use the "Start" command in Terminal Emulator, to order the Controller to initiate a operation of connection to the grid and start generating power.

To achieve that, go the Terminal Emulator form, set it at menu 2 and check all the pending errors. Whenever they are no pending errors, you can issue the Start command with pressing "A" followed by "1" (immediate start) at the Terminal Emulator.

You can check the behavior of the simulated wind turbine looking, first, to the "Brake subsystem" synoptic. There, you will see that the Controller orders the brake pump to start, that produces a steady growth of the pressure, till a value is achieved, at which moment, brake is released.

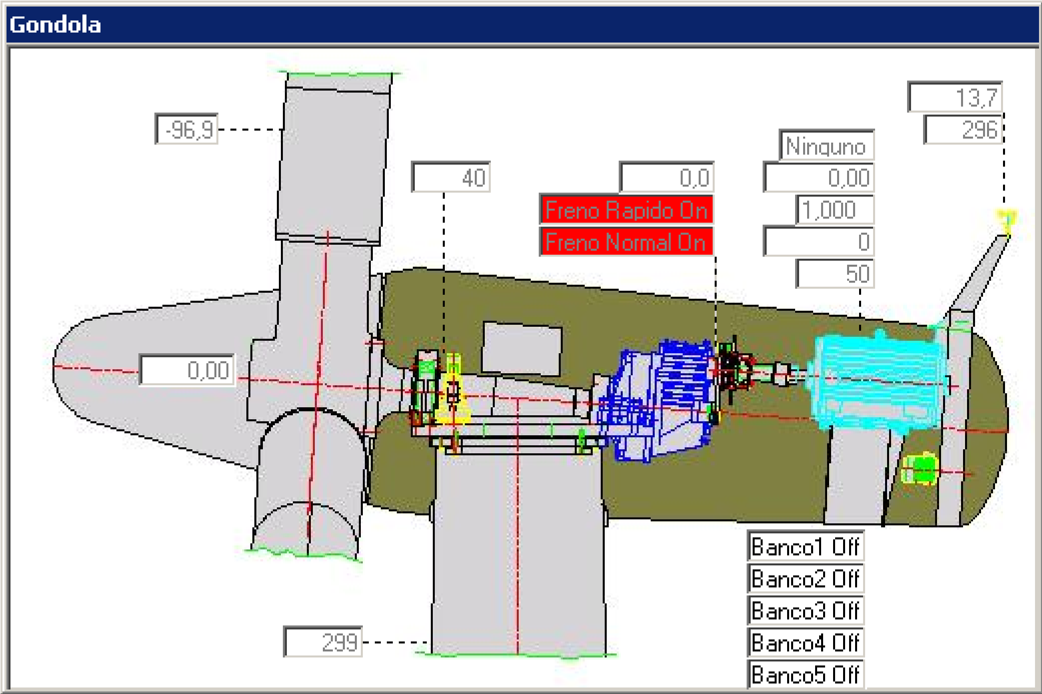
You might have a look also to the behavior of the "Pitch subsystem" and the "Blade synoptic".



In this synoptic is presented the status of all the variables related to the *Hub subsystem*. You can see the value of the Pitch angle "forced" by the Controller, as well as the "actual" angle of the each of the three blades. These three values are simulated by SIM1300 using independent linear control algorithms (one for each blade) and presented in the indicated widgets.

Another part of the synoptic shows the behavior of the hub oil pump, also implemented with its own process algorithm.

Of course, angle position of each of the blades is also presented in other synoptic, like the “Blade” synoptic, or the general view of the wind turbine (“Gondola” synoptic).



The behavior of every algorithm is controlled by a number of parameters. The value of these parameters can always be changed through the “Parametros” form. For a full description of each of them please check the “SIM1300 User’s manual”.

A typical behavior of the Controller, with adequate values for the parameters, and after the “Start” command issued in the Terminal Emulator, follows this pattern:

1. Look to the “Viento and Orientacion” synoptic, and see that the Controller actives the signals for “Giro Izq” or “Giro der”, forcing the orientation of the nacelle. You can check (in the TE, menu 3) that this situation correspond to the characters “<”, “>” or “-“ indicated in the 4th line of the Terminal Emulator.

2. Look also to the “Brake subsystem” synoptic, and you will see that after some time, the Controller set on the pump and that the (simulated) pressure grows to arrive to a value, feeded back to the Controller, indicating that the brakes are set free (widget “Brake pressure” change color).

3. Look now for the “Pitch” value issue by the Controller. You can find it in a number of synoptics: the “Control Panel”, the “Pitch subsystem” and the “Gondola view”. Typically these value will change in the range of 2, depending on the wind speed. This is the value the Controller uses

to “speed up” the blade speed in order to arrive, in a controlled manner, to the cut-in point.

4. If you looks now to the angle of the blades (in the “Pitch subsystem” or “Blade subsystem”) you will see that they attempt to arrive (they follow) the pitch value.
5. As a result of the (simulated) wind on the (simulated) blades and due to the fact that the (simulated) brakes are not acting, the blades start moving. The Controller uses the Pitch value to control the profile of the speed growing process. This will be more evident, if you look carefully, when the blade speed corresponds to around 600 rpms of the generator speed.

   Note that you can always change the value of the wind and/or its direction using the “Viento” synoptic, to see how those changes influence the behavior of the Controller.
6. Assuming that the wind is high enough (say, 13 m/s) you can watch that the Controller does activate the “Large” widget (“Control Panel”) when the generator speeds reach around 1000 rpms. That event indicates that it plan to use the Large generator for the connection to the grid.
7. If conditions continue stable you will see that the generator rpms RPMsGEN (“Control Panel” up right) will continue to climb, allways controlled by the Pitch value, till the reach and specific value (around 1490 rpms). At that moment starts the operation that will end up (if succesful) with the connection of the generator to the grid (*cut-in*).

   At that value of rpms, the Controller activates the conduction of the thyristor block. You will see that looking to the “TIRISTORES” widget in the “Control Panel”, that will change color to green. This typically last for around 4 seconds and, in reality, this conduction is not uniform.
8. When the RPMsGEN exceed the synchronous speed (1500 rpms for the Large connection of the generator), the Controller activates the BYPASS switch, short circuiting the Thyristors block and making a direct connection between the generator and the grid.

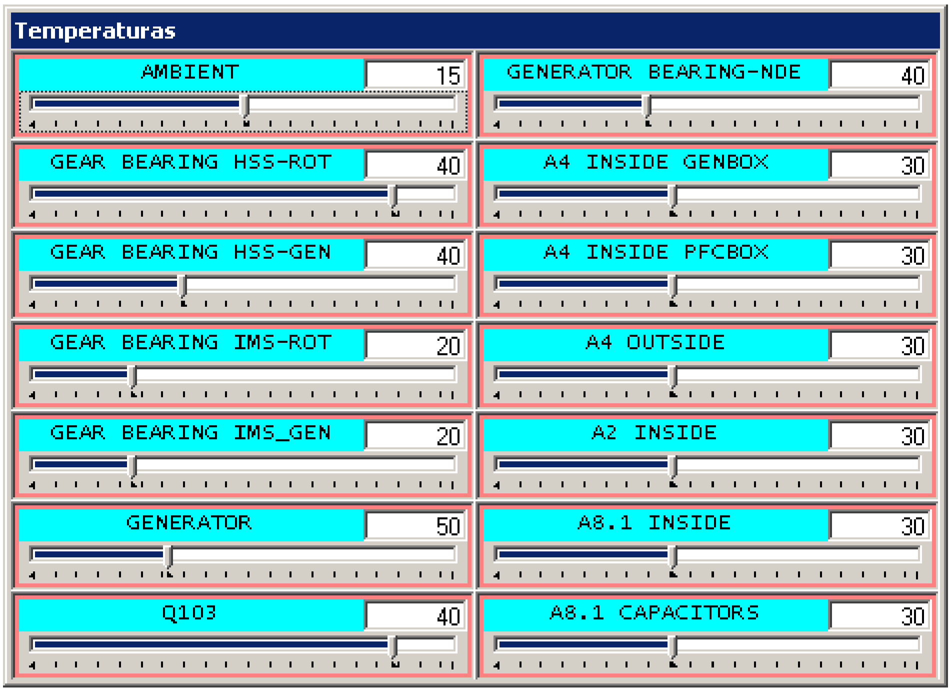
   Immediately after, the GPTR generates current values for each current measuring circuit of the grid, effectively “simulating” a drain (production) of power to the grid. This situation is presented in the “Control Panel” with changes in the “I”, “cosFi”, “PotTot”,”PotAct” and “PotReact” widgets.

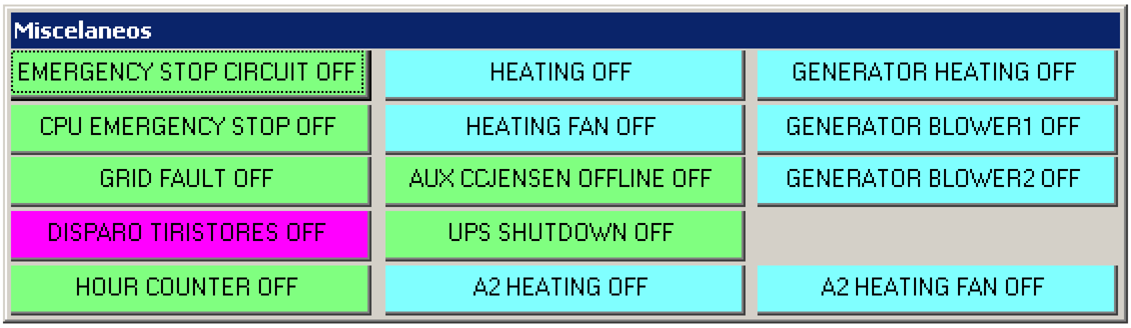
   The wind turbine is in production state.

3. Changes while in production

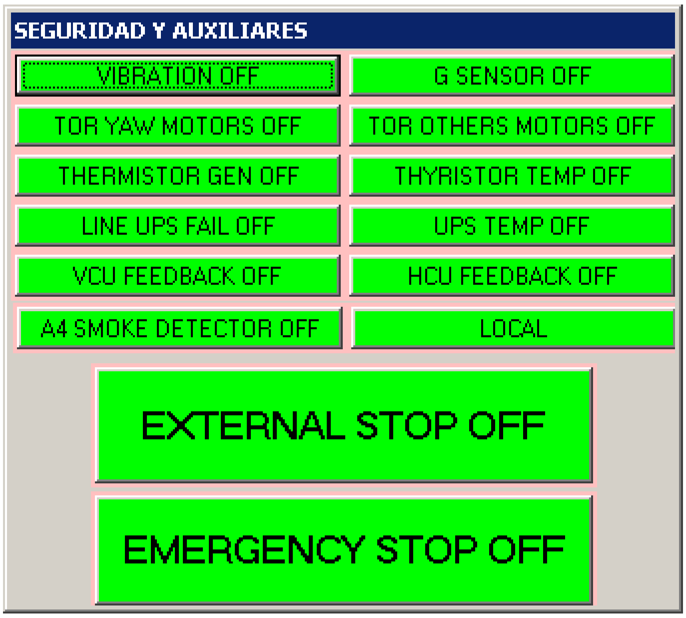
While the Controller + Simulator maintain a “Production” state, you can play with all the external variables (wind speed, wind direction, grid voltage) as well as with internal variables and see the induced changes in the whole set.

For example, you can change the temperature of the generator using the corresponding slider widget in the “Temperaturas” synoptic, to check how long takes the Controller to react connecting the corresponding fan, or any other reaction, in the “Miscelaneos” synoptic.

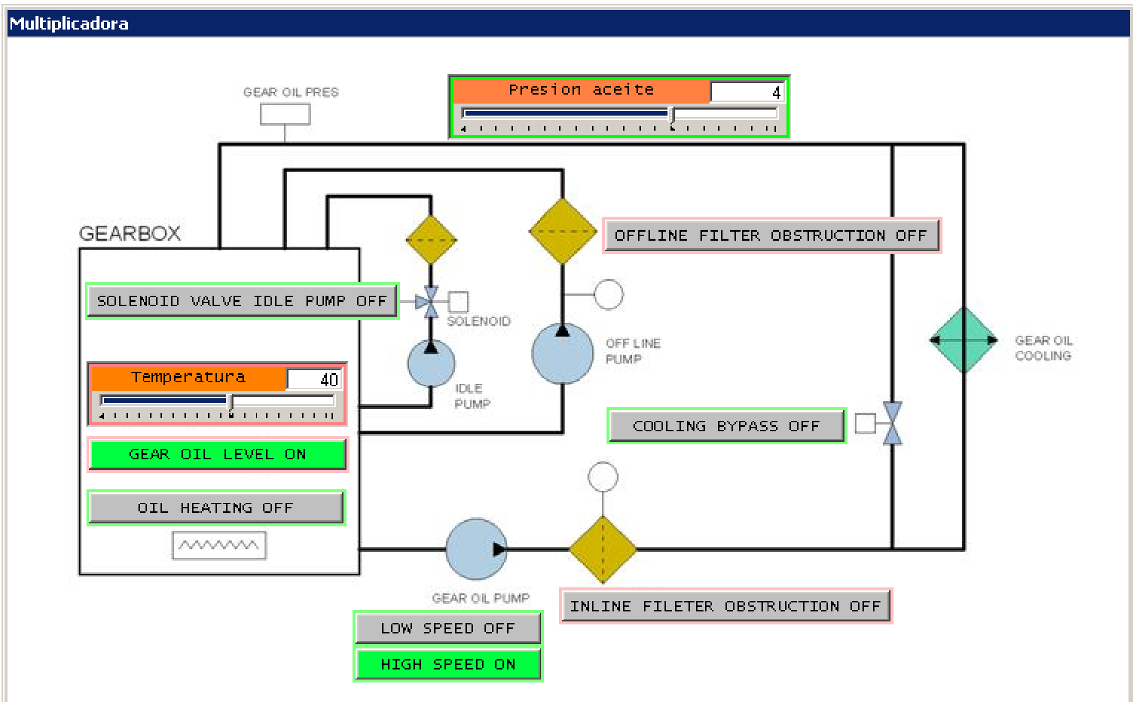




You can also exercise the reaction to some security transgression, using the signal monitor and/or controlled in the “Seguridad y Auxiliares” synoptic.



Or generate problems in the Gearbox subsystem (“Multiplicadora” synoptic). You can create a fault increasing the value of the temperature in the gearbox, or keeping the oil pressure low, or forcing a filter problem.



You can also generate **complex-to-create-in-reality** problems such as *setting one blade as non controllable,* to understand the behavior of the Controller and its timing. You can do that in a number of ways:

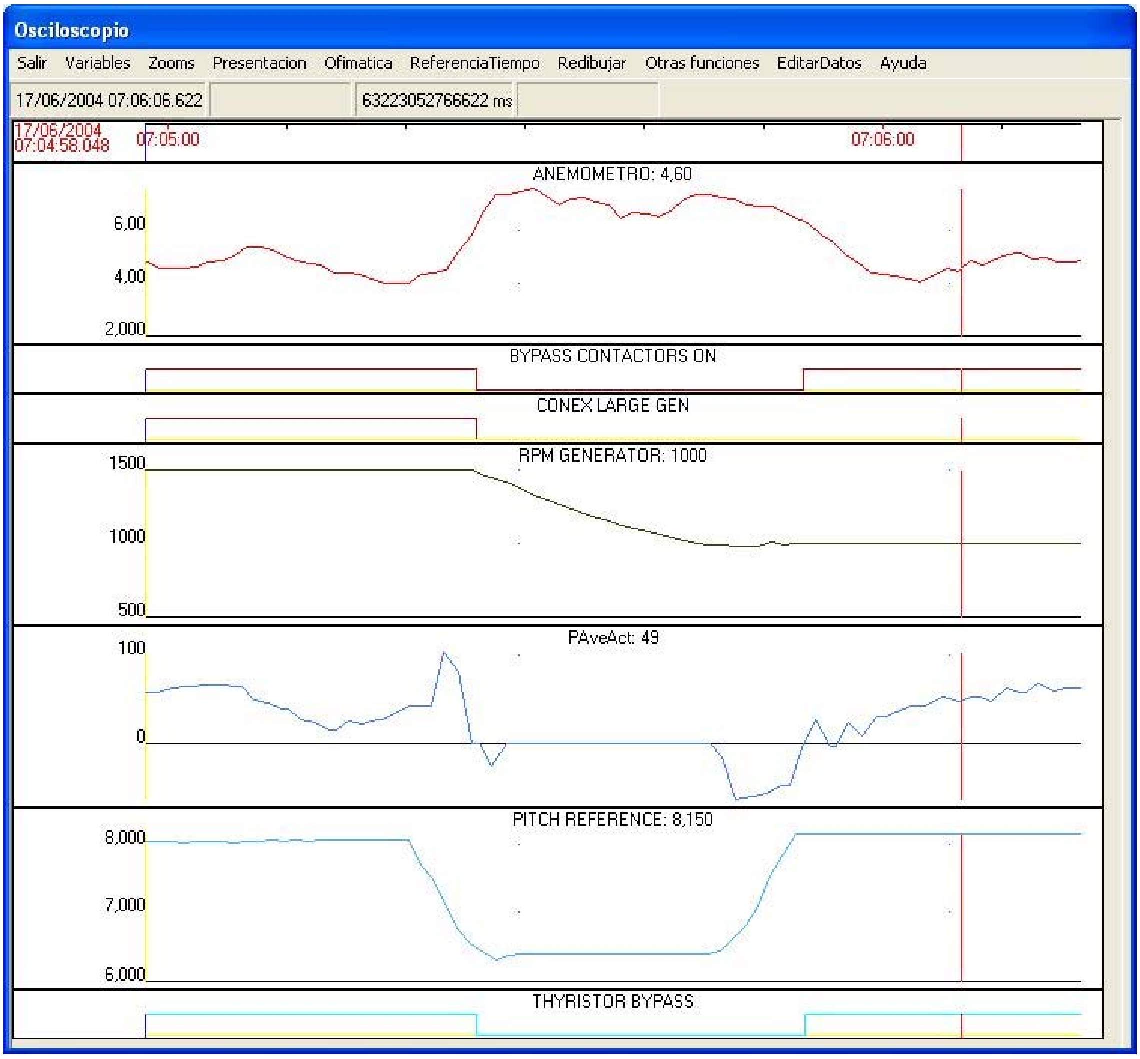
- Setting the blade angle as non controllable (“Pitch subsystem”, widget “Blade X” to manual control, pressing “F6”)

- Setting the “Seguimiento” (follow-up) checkbox of blade X to non checked, in “Blades synoptic” (follow up algorithm not active)
- Setting the “Forzar reposo” (resting position) checkbox of blade X to check , in “Blades synoptic” (force fixed position).

An interesting experiment is to see how is the transition from “Large” generator connection to “Small” generator connection and vice versa. These changes occur in “normal” behavior, due to changes in the wind speed.

You can check that the transitions are similar to the cases shown in the following figures.

1500 to 1000 transition



1000 to 1500 transition

These figures come from **actual field data, taken in a real wind turbine**.

Of course you can also test the reaction of the Controller to extreme wind speeds, just changing the wind speed above the preset level for that and keeping it for a long enough period of time.

Knowing the system

Basic principles

This simulator is developed with **LdP**, methodology and language specifically designed for the development of Simulation of Processes and Plants.

In a Simulator made with LdP, the physical devices (if any), the algorithms that express behaviors, and the graphical user interface (HMI) are treated in a common way, as “building blocks”. A building block has “inputs”, produces some “outputs” and have a given behavior. An algorithm that can be used to solved the equation that controls how the temperature a generator changes according to dissipated power, the thermal inertial, the dissipation coefficient and the external temperature is a building block that has four inputs and generates an output.

Similarly a synoptic (a form) that contains two switches and an analog meter has two digital inputs, two digital outputs, a analog input and an analog output.

A controlled waveform generator and its driver in LdP is a building block that has three inputs (frequency, wave form type and amplitude) and an output.

You construct a Simulator selecting the appropriated set of building blocks and “connecting” their inputs and outputs of so that they mimic the behavior of the real system to be simulated.

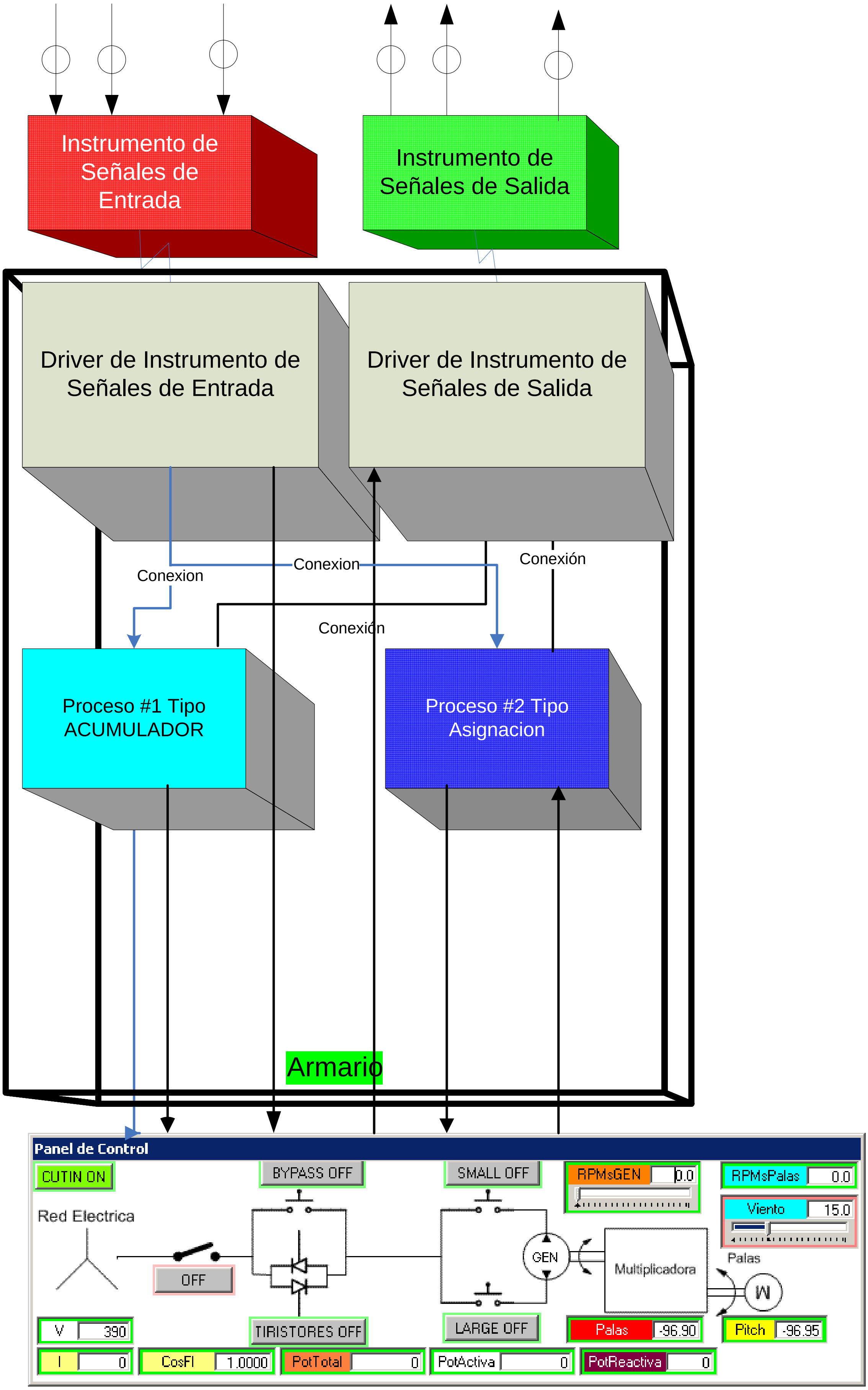
You can use the LdP for developing the building blocks and for testing them, before they are used in a particular Simulator.

The operation of making the connections can be done at any moment, even while executing the simulator.

So the whole Simulator that be viewed as similar to a electrical circuit.

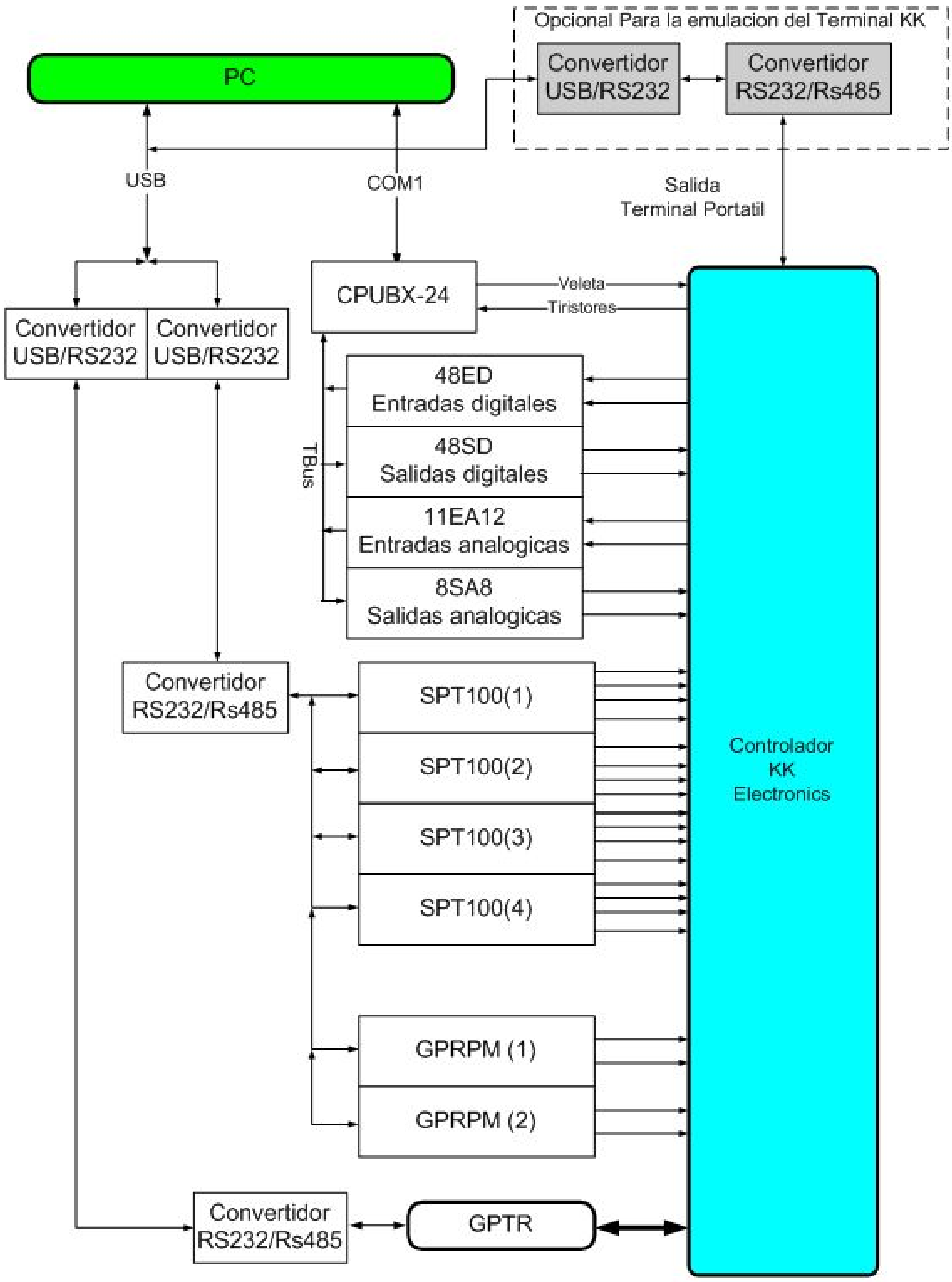
When executing, the values of the “output pins” are fed into the “input pins” of the building blocks to which each one is connected and the algorithms (behavior) of each building block is executed to produce the new values in their corresponding output pins.

The next figure shows a simple case.


Instrumento de Señales de Entrada
Instrumento de Señales de Salida
Driver de Instrumento de Señales de Entrada
Driver de Instrumento de Señales de Salida
Conexion
Conexion
Conexión
Conexión
Proceso #1 Tipo ACUMULADOR
Proceso #2 Tipo Asignacion
Armario
Panel de Control
CUTIN ON
BYPASS OFF
SMALL OFF
RPMsGEN
0.0
RPMsPalas
0.0
Viento
15.0
Red Electrica
OFF
GEN
Multiplicadora
Palas
W
V
390
TIRISTORES OFF
LARGE OFF
Palas
-96.90
Pitch
-96.95
I
0
CosFI
1.0000
PotTotal
0
PotActiva
0
PotReactiva
0

Hardware of the SIM1300

A block drawing of the *hardware* of the Sim1300 simulator is shown in the next figure.



Every device together with its associated software drivers behaves as a building block

Synoptics

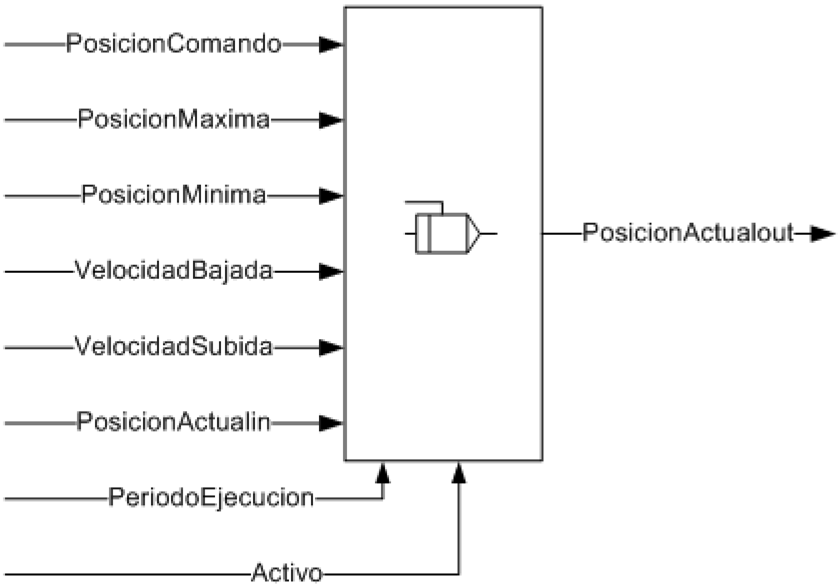
Every synoptic (form) presented so far, is considered also a building block, with signals coming in and out of widgets, with no functional difference from a hardware block.

Algorithms

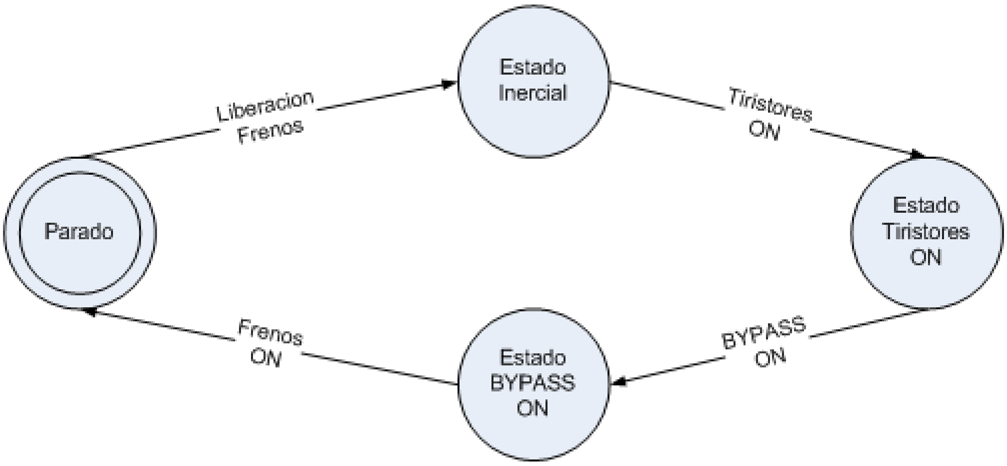
The behavior associated with some signals comes from the execution, in real time, of a number of instances of several algorithms. Those algorithms admit input signals as well as parameters and produced one or more output signals.

Algorithms are also treated as blocks. In this configuration for the BONUS 1300 we use a number of algorithms: linear following (blades), circular integration (nacelle yawing), thermal inertia, state-machine for the wind turbine, table driven (generator curves) and complex ones, such the computing of available power related to with speed and blade angle.

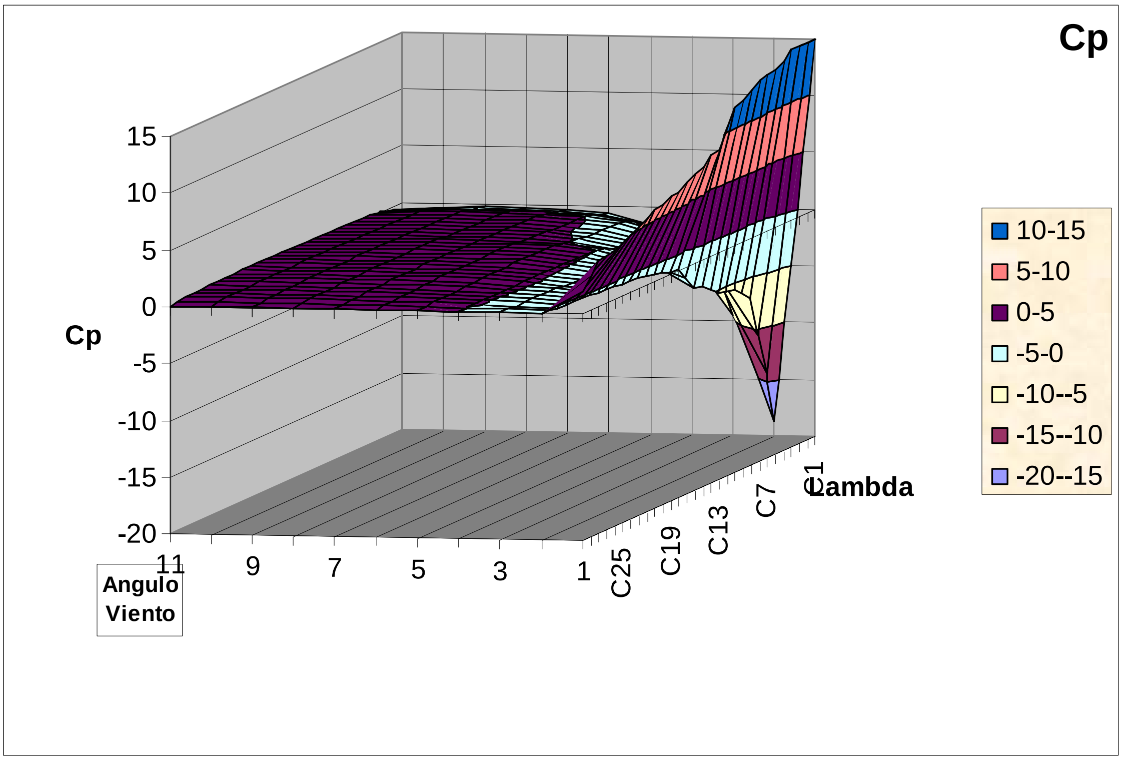
Linear Control



State Machine



Available power from wind speed, wind angle, blade angle.

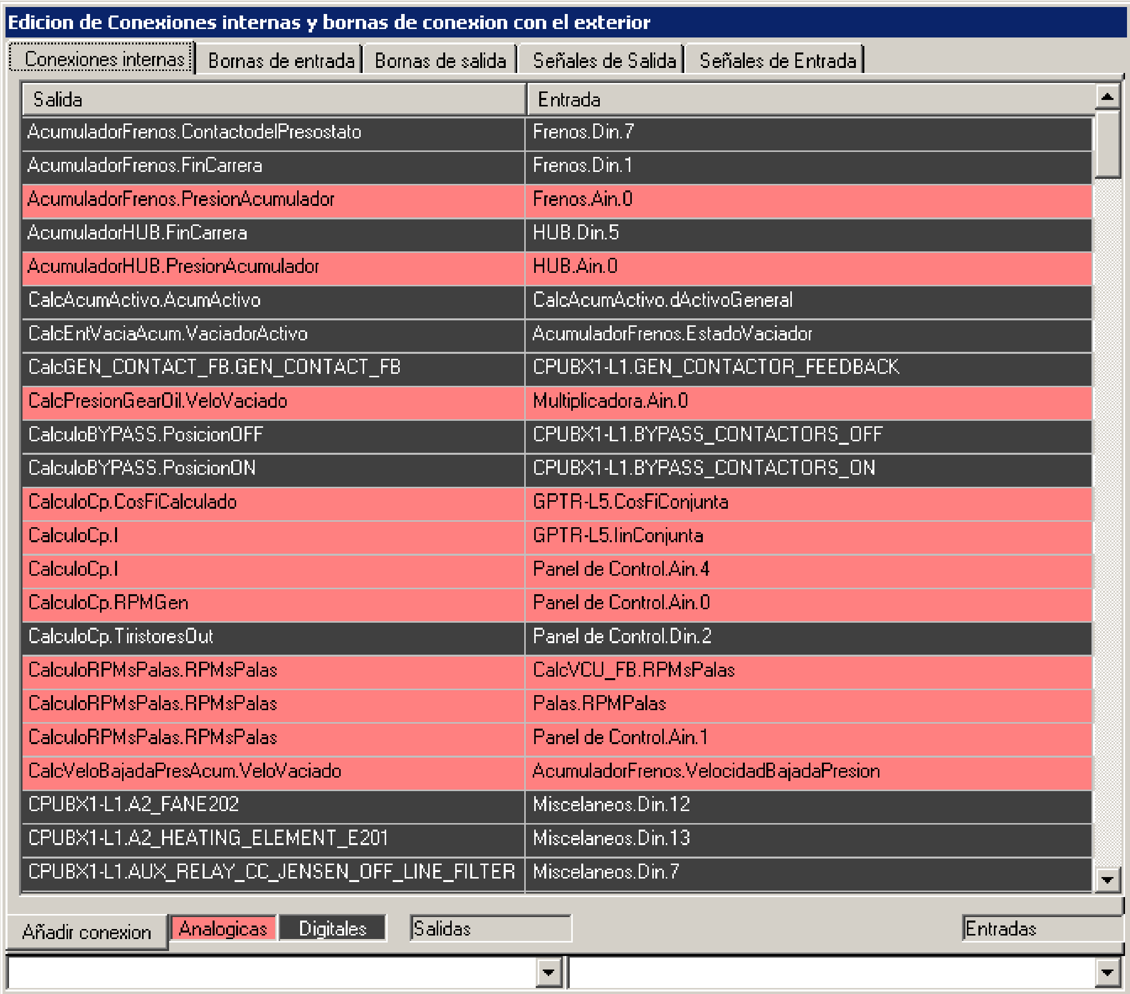


The design of this Simulator allows the replacement of any of these algorithms by others, with no need to replace other parts, such as the synoptic that shows/control its signals. That freedom of design is achieved with the concept of “Connectivity” among the blocks, through their signals.

Connectivity

The behavior of the Simulator, as a whole, depends not only of its block components (hardware, synoptics or algorithms) but also of the way they are interconnected

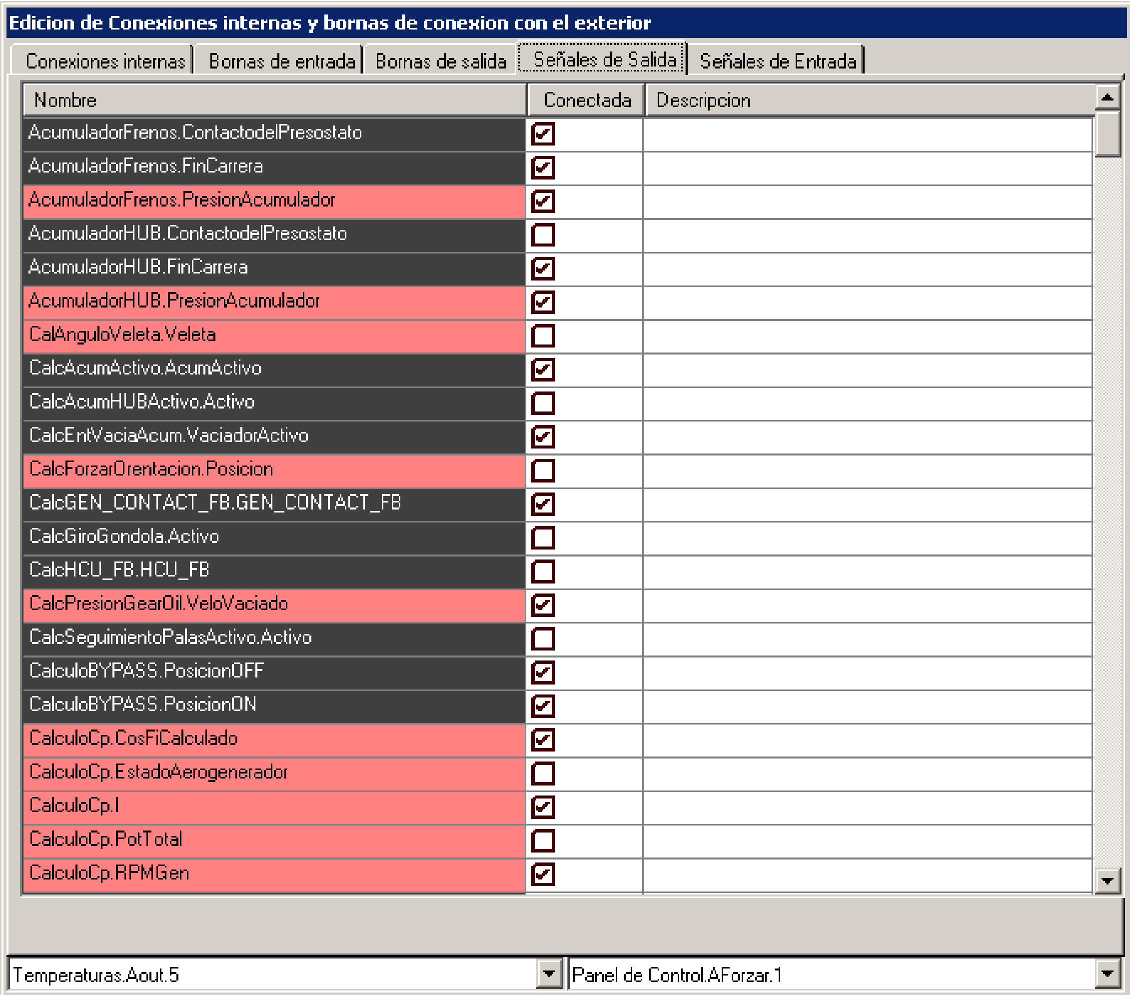
These connections can be seen (and <u>changed</u> during executing of a simulation!) through the following form.



Salida	Entrada
AcumuladorFrenos.ContactodelPresostato	Frenos.Din.7
AcumuladorFrenos.FinCarrera	Frenos.Din.1
AcumuladorFrenos.PresionAcumulador	Frenos.Ain.0
AcumuladorHUB.FinCarrera	HUB.Din.5
AcumuladorHUB.PresionAcumulador	HUB.Ain.0
CalcAcumActivo.AcumActivo	CalcAcumActivo.dActivoGeneral
CalcEntVaciaAcum.VaciadorActivo	AcumuladorFrenos.EstadoVaciador
CalcGEN_CONTACT_FB.GEN_CONTACT_FB	CPUBX1-L1.GEN_CONTACTOR_FEEDBACK
CalcPresionGearOil.VeloVaciado	Multiplicadora.Ain.0
CalculoBYPASS.PosicionOFF	CPUBX1-L1.BYPASS_CONTACTORS_OFF
CalculoBYPASS.PosicionON	CPUBX1-L1.BYPASS_CONTACTORS_ON
CalculoCp.CosFiCalculado	GPTR-L5.CosFiConjunta
CalculoCp.I	GPTR-L5.IinConjunta
CalculoCp.I	Panel de Control.Ain.4
CalculoCp.RPMGen	Panel de Control.Ain.0
CalculoCp.TiristoresOut	Panel de Control.Din.2
CalculoRPMsPalas.RPMsPalas	CalcVCU_FB.RPMsPalas
CalculoRPMsPalas.RPMsPalas	Palas.RPMPalas
CalculoRPMsPalas.RPMsPalas	Panel de Control.Ain.1
CalcVeloBajadaPresAcum.VeloVaciado	AcumuladorFrenos.VelocidadBajadaPresion
CPUBX1-L1.A2_FANE202	Miscelaneos.Din.12
CPUBX1-L1.A2_HEATING_ELEMENT_E201	Miscelaneos.Din.13
CPUBX1-L1.AUX_RELAY_CC_JENSEN_OFF_LINE_FILTER	Miscelaneos.Din.7

The left most tab in this form shows the connections among the different blocks (both hardware blocks, synoptics and algorithms). Black rows correspond to digital signals, red ones to analogic signals. Each output signal (left: "Salida" – output -) can be connected to 0, one or more input signals (right: "Entrada" – input -).

The tab branded "Señales de Salida" (Output signals) show all the signals going out of the simulator hardware (to the Controllers). The column "Conectada" (connected) shows the actual "virtual" connection status of the signal: if not checked, the changes of that signal are not transmitted to the Controllers. <u>This is another way to force faulty conditions to check the reaction of the Controller</u>.



The last tab ("Señales de Entrada") (Input signal) shows the list of signals coming in the Simulator hardware, and theirs "virtual" connection status.