

EcosimPro

Modelling and Simulation Software



CONTROL LIBRARY

The CONTROL library facilitates the robust and fast design of control diagrams applied to many applications, interacting with subsystems in other libraries and creating multidisciplinary systems. Moreover, it provides ample functionality for handling signals.

EcosimPro

EcosimPro is a powerful **modelling** and **simulation** tool with a simple interface that makes the design of **multidisciplinary** dynamic systems easy and intuitive using graphic diagrams.

For users with specific needs, EcosimPro provides an object-oriented non-causal approach towards creating reusable component libraries and is based on very powerful symbolic and numerical methods capable of processing **complex systems** represented by differential-algebraic equations (DAE) or ordinary-differential equations (ODE) and discrete events. However, low-level problems such as programming calls with numerical solvers, equation handling, etc, are solved automatically or using simple wizards.

Features

CONTROL is a standard library supplied with EcosimPro. It has been designed as a basis for managing **signals** and creating control structures.

Using **drag & drop** methodology, the user can quickly create a diagram of the control system to be analysed, the representation of which is very similar to the physical system. This system can be modelled by means of transfer functions or in status spaces with the actual CONTROL library, or it can be used in modelling together with other disciplines, such as thermal, fluid, hydraulic or any other defined by the user.

Thanks to EcosimPro's features, the CONTROL library is very easy to **configure** and **extend**, adding any components and characteristics as needed. This can be done graphically through a simple, user friendly interface, or through EcosimPro's **object-oriented language** which makes it possible to re-use existing codes.

Applying the object-oriented methodology, a group of components can be encapsulated into a single subsystem used like a **black box**, with a defined number of inputs and outputs. This enables **modular** design of the overall system and makes it possible to share tasks among different specialised **users** or merely simplify the system.

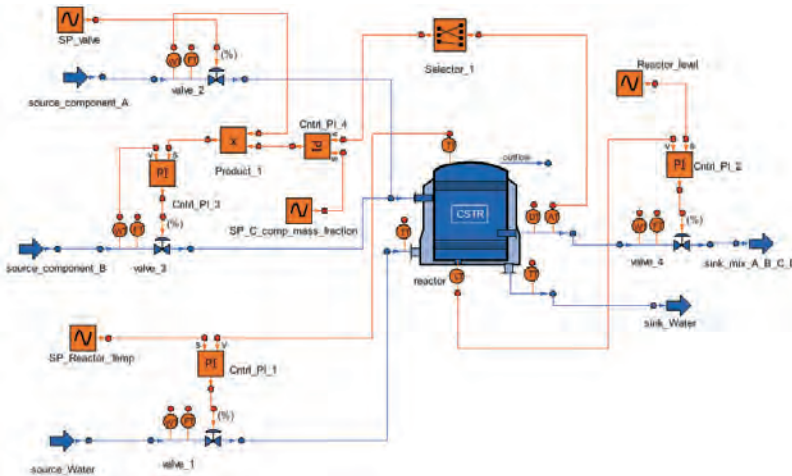
The library not only has a wealth of components with which to model any current control structure covering **analogue**, **digital** and **mixed** systems, it also contains specialised analogue and digital components and converters.

The components

In summary, different component groups will be established and included in the Control library with several representative examples:

- **Signal generators:** The input possibilities cover classic forms of excitation such as stepped input, ramp input, staggered input, exponentials, frequency variators... including stochastic signals based on different distributions (normal, Gaussian, Poisson,...) and different data reading functions and their interpolation into tables defined by the user via files so that any pattern can be established
- **Signal handling:** There are different functionalities for handling signals by means of switches, comparators, multiplexers, demultiplexers, or they can be handled through operations such as add, inversion, integration, bypass, trigonometry functions
- **Signal transformation:** Transformation of the type of variable (Boolean to real), samplers, holders, filters, delays





- **Logic gates:** To create Boolean logic diagrams, we have AND, OR, NOT, XOR gates
- **Controllers:** The components P, PI, PID are highly configurable contemplating the phenomena experienced with current controllers, such as anti-windup, deadband, saturation to different internal levels of the component, filtered
- **Modellers:** Finally, the plant to be controlled can be modelled by external libraries, or using their representation by means of transfer functions, status spaces, zero-pole

Example

In the above diagram we have a CSTR reactor in which two products A and B react to obtain C and D. The reaction is not complete so the resulting product will be a mix of the four components.

The reactor has a tube sleeving because the process temperature must be controlled. This temperature will be one of the values introduced by the operator and controlled with the component Cntrl_PI_1 which calculates the valve opening for the flow of water through the reactor tube sleeving. The variable measured is the temperature of the product. The graph illustrates a stepped change in the setpoint from 60°C to 65°C at instant 500 seconds with a temperature settling time of approximately 100 seconds.

The reactor level will also have to be controlled; in this case, a constant level is maintained throughout the simulation and it can be seen that the value measured does not exceed a variation of 0.5%.

This is performed with the following control structure:

- **Control PI:** the component Cntrl_PI_4 calculates the relationship that should exist between the mass flows of A and B based on the C concentration readings at the outlet and associated setpoint
- **Control ratio:** represented by the component Product_1. Based on the relationship calculated in the previous component and on the mass flow reading of A, we obtain the necessary flow of B
- **Cascaded control:** represented by the component Cntrl_PI_3, it calculates the valve aperture necessary to obtain the calculated flow of B

We can see in the graph how in the first instant we have a mass fraction of C of 0.444 and at instant 150 the setpoint is raised to 0.5. System behaviour is suitable with slight oscillation and a settling time of 100 seconds.

We can also see the need for greater cold water flow when the setpoint of the component C concentration rises, and on the other hand the lower demand by component C when we want to raise the reactor temperature.

The CONTROL library has been used both to establish the control structure and to impose the necessary setpoints and manage the signals, as in the case of the selector (the meter output is an array with the mass fraction of the 4 components and that corresponding to component C is selected as PI input).

