

How to Build a High-Performance Control System

Overview of the control package in XC05Edge

By

Gunnar Bengtsson, PhD, CEO First Control systems AB

Joakim Örnescans. M.Sc. Inventor of XC05Edge



Introduction

Industry still relies to a large extent on control technology that was fundamentally developed more than fifty years ago. The problem is not that better methods are lacking, but that most ICS/PLC systems still offer a narrow and outdated range of standard functions. In practice, this means that much of the powerful control technology that has been developed never becomes available where it is actually needed. Achieving better control therefore often requires custom programming, which makes the solutions more expensive, less efficient, and often less reliable in operation.

In XC05Edge, we solve this problem by providing pre-programmed objects and modules that engineers can use directly in their automation, based on their own process knowledge.

Together, they cover the central parts of designing a modern control system:

- ✓ **A general adaptive controller** with both feedback and feedforward for processes with high performance requirements. It can handle up to 20 control parameters, all of which adapt to changes in the process.
- ✓ **Classical PID controller** for simpler applications
- ✓ **Higher-order signal filters** that are significantly more effective than the simple filters normally available in ICS/PLC system
- ✓ **An adaptive optimizer** that in real time seeks to drive a measured variable toward a maximum or minimum.
- ✓ **An adaptive neural network** that identifies nonlinear relationships directly from real-time data. It can be used for predictive control during production transitions or to generate an offset to fixed optimal control curves.
- ✓ **Real-time process simulation** using dynamic models created in Modelica for commissioning support and process diagnostics. This function runs in a separate CPU core so as not to affect the automation part.
- ✓ **Real-time AI functions** operating directly on process data. They can be used for all types of AI functions developed in Python. These also run in a separate CPU core so as not to interfere with the automation part.
- **Real-time calculation of statistical data** according to the SPC standard in order to document the improvements achieved through control.

Below, an overview is given of the most important objects in XC05Edge and how they are used in practice.

Objects in XC05Edge

All the functions described below are pre-programmed and ready for use by the process engineer.

Controllers

PID				The PID controller	
- L1-	ON	HIL	-L1-	The technology behind the PID controller was developed more than 60 years ago and is still dominant in industrial control systems. It is simple and robust, but its functionality is limited by its simple structure. PID should mainly be used for: • simple control loops without high performance requirements • control loops that are repetitive and expected to behave in the same way over time The controller has fixed parameters that are normally adjusted manually. The tuning often has to be conservative in order to remain stable over a wider operating range without creating oscillations or instability. An alternative is to adapt the parameters to the operating point during operation, for example in position control with a hydraulic valve in a rolling mill, where the settings vary with position and pressure. For processes with more complex dynamics, such as dominant dead time or non-minimum-phase behavior, PID performs significantly worse because its structure is simply too limited	
- L2-	AUTO	LOL	-L2-		
- L3-	IHOLD	U	-R1-		
- L4-	IZERO	E	-R2-		
-R1-	MV				
-R2-	SP				
-R3-	UE				
-R4-	HI				
-R5-	LO				
-R6-	KP				
-R7-	1/TI				
-R8-	TD				
-R9-	K				
- I1-	MODE				

STREGX2

– L1	ON	U	– R1
– L2	AUTO		
– L3	ADAPT		
– L4	LOAD		
– L5	DUMP		
– I1	MODEL		
– R1	MV	Self	
– R2	SP	Adaptive	
– R3	UE	regulator	
– R4	FF1	Pole	
– R5	FF2	Placement	
– R6	HI		
– R7	LO		
– R8	DUP		
– R9	DUM		
I1	NA		
I2	NB		
I3	NC		
I4	MD		
I5	ALFA		
I6	SAMP		
R1	UMAX		
R2	UMIN		
R3	RESU		
R4	RESY		
R5	KINIT		
R6	BMPLVL		
R7	POLE		
I7	MODE		
I8	W		

The general adaptive controller

The adaptive controller is based on the self-tuning technology developed at Lund University under the leadership of Professor Karl Johan Åström. The purpose is to make modern model-based control practically usable by basing the control action on identified process models instead of physical models.

The adaptive controller is particularly suitable for:

- processes with high performance requirements
- processes with large and varying dead times
- processes where predictive control with feedforward signals is possible
- processes with varying dynamics

The greatest improvement is achieved through predictive feedforward control, where disturbances can be compensated before they affect the process. In many cases this has reduced variations by up to a factor of ten compared with conventional control. Feedforward requires a process model, which in practice makes the method difficult to use without adaptive technology, where the model is created automatically.

STREGX2 uses a prediction model of relatively high order, around 5, to cover a broad range of industrial processes. The model is continuously adapted to data through recursive identification with a forgetting factor, which allows both the model and the controller to follow changes in the process dynamics. The control strategy is then calculated by an algorithm based on pole placement

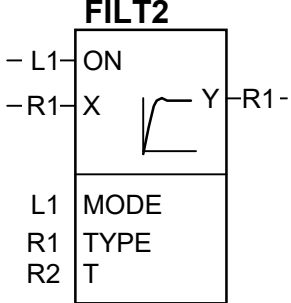
Proven in industry.

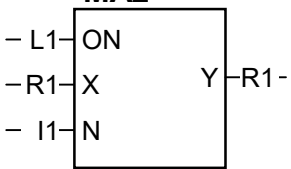
STREGX2 is robust and reliable in operation. The same controller object has been used in hundreds of installations and has delivered documented improvements of 50–75% in yield and efficiency across a large number of processes in widely different fields.

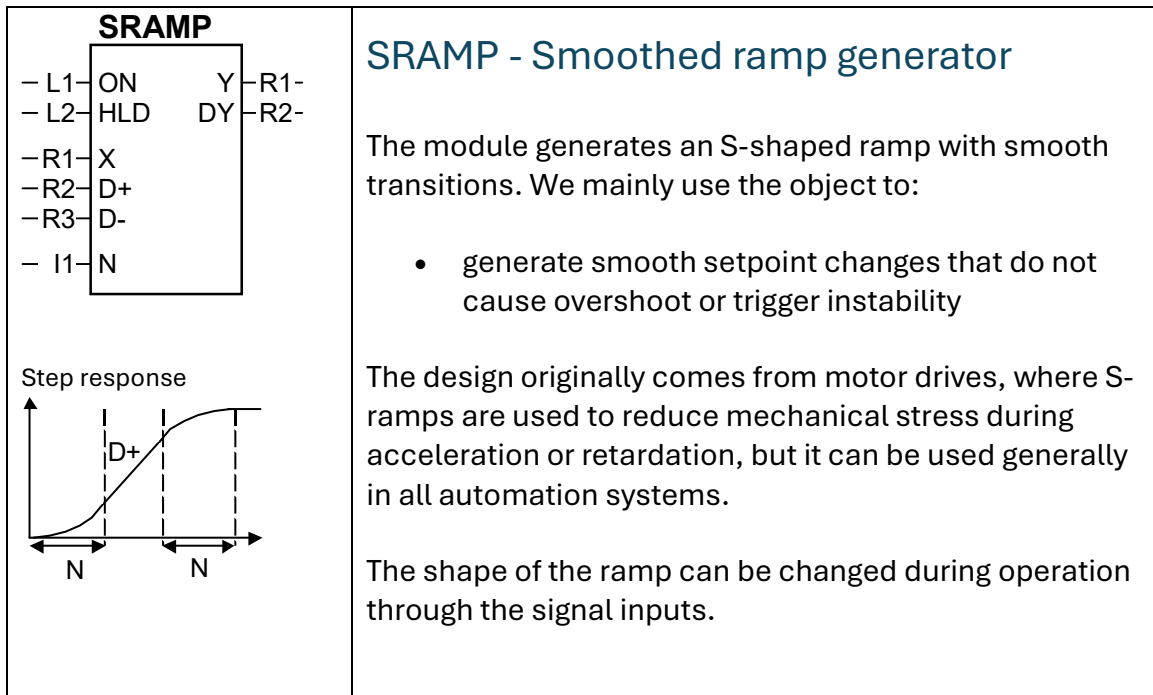
For more information, see

<https://www.firstcontrol.se/wp-content/uploads/2026/02/Self-tuning-as-automation-standard.pdf>

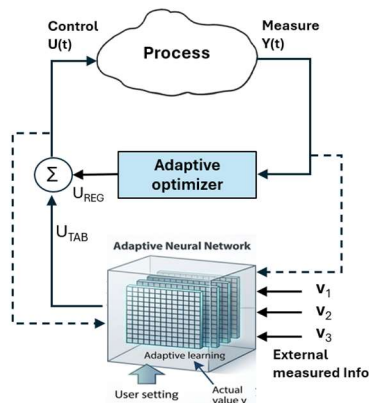
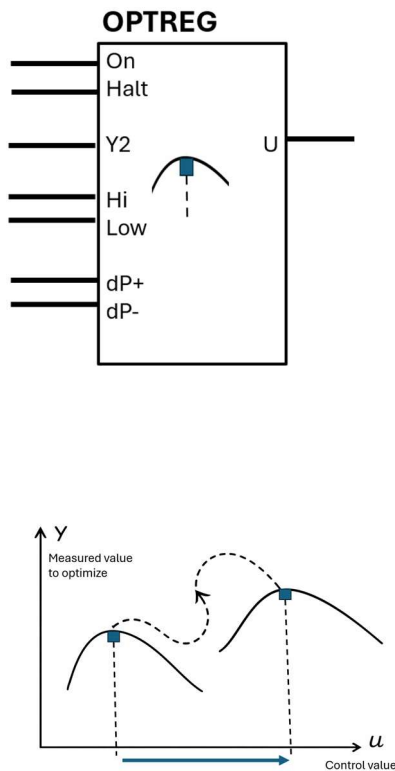
Signal conditioning

FILT2 	Higher order signal filter The choice of signal filter is a central part of a control system design. The purpose is to condition the signal so that only the useful and controllable part remains. Since most disturbances are high-frequency, low-pass filters are normally used, although band-pass filters may also be relevant in some cases. We use higher-order low-pass filters as standard on all analog inputs. In FILT2, the user can choose between 4th-order ITAE, Butterworth, and Bessel low-pass filters. These filters are far more effective than the simple low-pass filters normally found in ICS systems or built into sensors. This is particularly important in processes where external oscillations occur close to the controllable frequency range, for example from rotating electrical machines or out-of-round rolls. Incorrectly tuned filtering is one of the most common causes of poor control in practice. In addition, many sensors already contain heavy internal filtering to make the measured signal “look good.” In classical control terms, excessive filtering increases the phase lag in the control loop, which can cause oscillations and slower control.
---	---

MA2 	MA2 - Moving average filter (max 100) The MA2 object calculates the moving average of the latest N measured values, where $N < 100$. We mainly use the filter to: - Detect trends, for example to limit the control signal if it drifts outside the effective operating range - Detect sufficient signal variation for adaptive learning An example is also shown in the SPC object below.
---	--



Real-time optimization



Self-tuning extremum seeker

The controller drives a process during operation so that the measured variable Y2 is driven toward an optimum. The requirements we placed on the optimizer were the following:

- it must be able to follow a moving optimum
- it must not depend on a physical model
- it must be robust, reliable in operation, and insensitive to signal disturbances

The user controls the function of the optimizing controller by setting limits and by starting, stopping, or freezing the controller through the module inputs.

The controller has two parameters that are adjusted by the engineer during commissioning:

- the curvature **a** of the parabola (gain factor)
- the forgetting factor **w** (integration time)

There is a very large number of processes in which an optimizer can be used, either as a simple corrective function to improve efficiency or as a complement to an existing optimal control curve. We ourselves have used it to optimize operations in Kaplan turbines and oil burners.

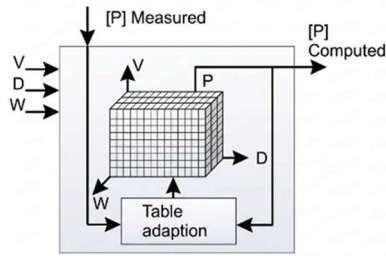
An extended predictive optimizer

The simple feedback optimizer described above can operate independently in any PLC/ICS system. However, it is often desirable to also “remember” the optimal setting and how it is affected by other measurable process signals. We solved this problem by using an adaptive neural network as a complement to the adaptive optimizer, according to the configuration shown in the figure to the left.

For more information, see

<https://www.firstcontrol.se/wp-content/uploads/2026/04/Self-tuning-optimizer.pdf>

Adaptive neural networks



Adaptive neural network

The starting point was the way engineers usually describe nonlinear relationships in a computer: by using a table. Our idea was to automate this method by replacing the traditional table with a multidimensional grid, with a linear static or dynamic model at each grid point that is continuously adapted to process data.

We use the adaptive table as an internal “memory” of the results from previous production. It can be used to:

- optimize production transitions, for example in a zinc coating line
- correct an optimal control curve in a Kaplan turbine or combustion engine
- implement data-driven MPC without requiring a physical model
- provide the best starting point in a rolling mill depending on production data and previous start-ups

The range of applications is enormous. Only the applications in which we ourselves have experience are mentioned here.

The adaptive neural network is designed as a module that the engineer can use without specialist knowledge. The engineer simply enters some basic parameters defining the network and connects the signals. The function of the network is then handled by XC05Edge and is controlled during operation through logical inputs.

For more information, see:

www.firstcontrol.se/wp-content/uploads/2026/03/Selftuning-AI.pdf

Real-time process simulation

MODELICA - L1 - ON ERR - L1 - L2 - HOLD TIME - R1 - = L3 = L L - L2 : = I1 = I I - I1 : = R1 = R R - R1 : R1 - STEP R2 - RTS T1 - MODELNAME	MODELICA – Real-time model simulator XC05Edge runs any dynamic model created in Modelica in real time directly on the device. Just load the Modelica file and define the model's signal inputs and outputs. The model is then executed in real time, essentially as a digital twin of the process. Modelica can be used in several ways: • verification and testing of the control system before installation • process diagnostics by detecting deviations between the real process and the model • training of customers and students using realistic process models The module is stored in a dedicated Modelica library that the programmer can use and connect to real-time data in the application. The simulation model runs in a separate CPU core and cannot interfere with the automation in any way, even if it runs out of control. The simulation capacity is comparable to that of a normal laptop with the electronics we use
---	--

Python with AI functions

PYTHON		2
v0-ON	DEF	
v0-HOLD	STATE	
v0-L.1	VER	
v0-I.1	ERR	
v0-R.1	EXTIME	
	L.1	
	I.1	
	R.1	
MODE		0
A_TMO		10
S_TMO		1.4
I_TMO		20
SPARE		0
MNAME	mymod	
FNAME	myfunc	
TEXT.1		T

Python scripts with AI

XC05Edge runs AI and other advanced Python functions directly on the device. By adding a simple wrapper for inputs and outputs, the Python function can be loaded into XC05Edge, where it appears as an object in the Python library and is connected directly to real-time data in the application.

The following standard libraries are included
SciPy, NumPy, Matplotlib, Pandas, Scikit-learn, and GEKKO.
Additional libraries can be included when needed.

The object can be used in several ways:

- AI functions running with real time access to data.
- Custom control functions beyond the standard library
- Storage and analysis of process data in real time

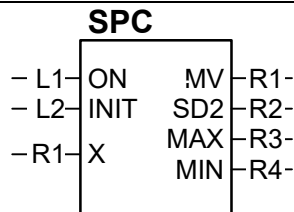
Most AI functions for automation can be prepared offline and then run locally on the device in real time without any internet connection. That means OT intrusion through external connections is not possible.

The device capacity is comparable to that of a normal laptop, and the internal storage can be expanded to 500 GB. Since the Python function runs in a separate CPU core, software errors cannot affect the automation. Program errors are reported to the automation system and the operator through the ERR output.

We have used the Python module to optimize the heating of a building using nonlinear MPC control. The automation part handled communication both with the local PLC systems at 10 ms and with SMHI weather forecasts at 1 hour. The test installation was in operation for about six months without any disturbance or any stop.

This installation has been described in another LinkedIn post

Production statistics



SPC – Statistical process control

It is often difficult to document the improvements achieved through better control. Conventional data acquisition systems are usually useless when it comes to showing how well a control system actually performs.

We therefore use a special SPC module that is widely used in the automotive and steel industries. Statistical data is recorded periodically through fast sampling of measured values:

- mean value
- standard deviation
- maximum value
- minimum value

The data is then stored in a local database after each time period for later analysis. In this way, all signal deviations are captured, including the fast ones, and can also be displayed to the operator in real time.

The following example from a continuous casting machine shows the importance of statistical methods. The diagram shows the frequency of large mould-level disturbances in 3,000 castings before and after we installed adaptive control. Did it improve?

It resulted in 75% fewer large disturbances. This becomes very clear if we filter the data using a **moving average** over about 50 batches. Note that the old PID control system was used during some longer holidays, resulting in heavy deterioration in performance

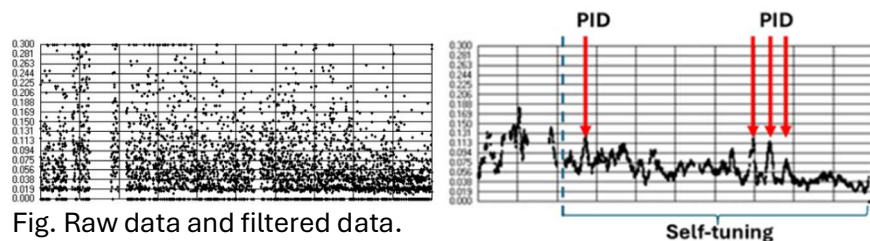


Fig. Raw data and filtered data.

