

A Self-Tuning Optimizer for Industrial Processes

A Simple Extremum-Seeking Controller

By

Gunnar Bengtsson, PhD, CEO First Control systems AB

Joakim Örnescans. M.Sc., Inventor of XC05Edge

Introduction

An optimizer can be regarded as a generalization of a conventional controller, with the difference that the objective is not to drive the process toward a fixed setpoint, but to increase or decrease a measured variable during operation so that a maximum or minimum is reached.

Such problems occur in several fields, including engines, chemical and biological processes, and energy and turbine systems. In many practical applications, fixed control curves are used to achieve high efficiency. These curves are normally determined by model tests and field measurements. The drawback is that they remain unchanged even when the process ages or is subjected to unexpected or unmeasurable disturbances.

Against this background, we early developed a general adaptive optimizer that is entirely data-driven and does not rely on any physical process model. It was originally designed to optimize the efficiency of a Kaplan turbine and an oil burner, but the principle is general and can be used in many different processes. The method is also easy to implement in a standard ICS/PLC system.

During the development work, extensive simulations were carried out using the methods that were available at the time, but none of them met the requirements we set regarding robustness, speed, and insensitivity to measurement noise.

Problem formulation

The purpose of the optimizing controller is to follow a moving optimum that changes during operation.

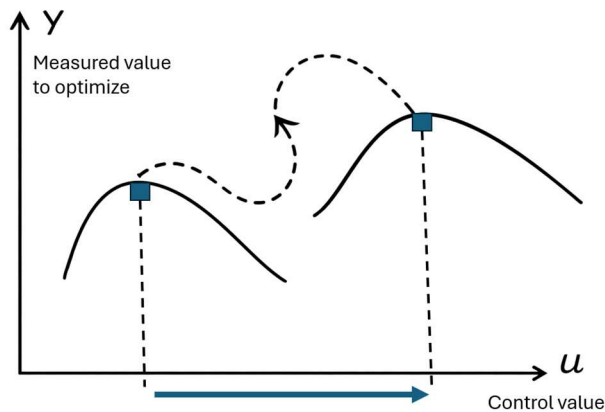


Fig. 1. Moving optimum to follow.

The controller must therefore drive the process during operation so that the measured variable y is moved toward the optimum. The requirements we imposed on the optimizer were the following:

- it must be able to follow a moving optimum that changes due to unexpected and unmeasurable disturbances in the process
- it must not depend on a physical model, but be entirely data-driven
- it must be robust, reliable in operation, and insensitive to signal disturbances
- it must be usable by a process engineer without specialist knowledge

The problem is more complex than for a conventional controller, because one must first determine in which direction the controller should move before the next step is taken. We solved this problem using a recursive identification method that we had previously used in the adaptive controller.

A simple feedback optimizer

The regulator algorithm

The process model is a simple parabola that does not take other measurable process variables into account. It functions as a simple controller with parameters that the user can adjust in order to adapt the function to the specific process.

$$Y(t + k) = a U^2(t) + b(t) U(t) + c(t)$$

$U(t)$ = process input

$Y(t)$ = process output (to be optimized)

where a is fixed and $b(t)$ and $c(t)$ are unknown.

The parabola is adapted so that it matches the tangent of the unknown optimum curve at the operating point as closely as possible.

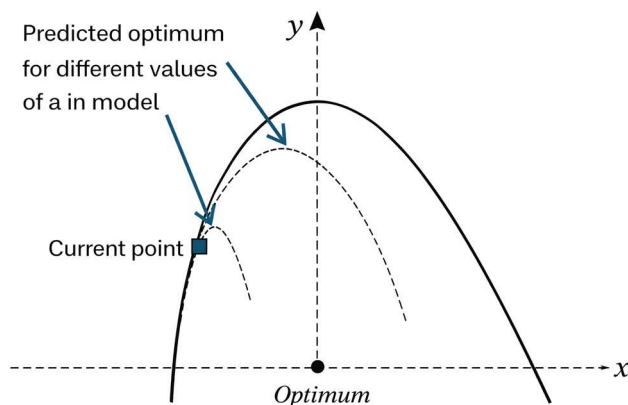


Fig. 2. The curvature a acts as a gain parameter that makes the optimization faster or slower.

We use the same robust recursive identification method that we previously used in the adaptive controller to estimate the unknown factors $b(t)$ and $c(t)$. The curvature a in the model is fixed and acts as a “gain factor” that is adjusted to the specific process, see the figure.

The forgetting factor is adjusted to the reliability of the measured signals $y(t)$ and $u(t)$ involved. Normal values are 0.7–0.9, which corresponds to 7–22 data points with weights greater than 10%. This may be interpreted as the equivalent of the “integral part” in a conventional PID.

Note: This approach is in fact a robust and disturbance-insensitive way of calculating the first and second derivatives at the operating point, so that the next step can be taken.

The control value $U(t)$ is then calculated using the estimated value of $b(t)$:

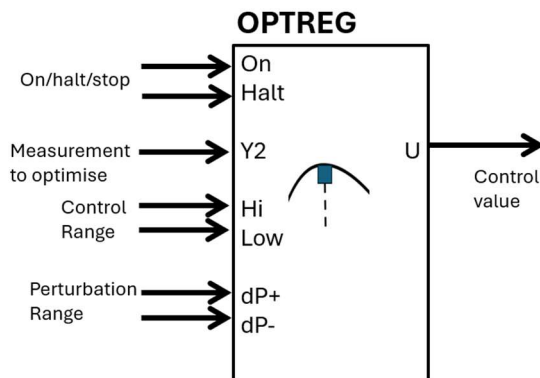
$$U(t) = -\frac{\hat{b}(t)}{2a}$$

Perturbation signals

We discovered at an early stage that the normal signal variation in the process was not sufficient for a model to be created. Therefore, we added a small perturbation signal in the form of a triangular wave to the control value. The signal was very small and was masked by the normal process variation, and it did not affect the control performance.

Standard module in an ICS/PLC system

A standard module that the user connects to the signals in the application.



The user controls the function of the optimizing controller by setting limits and by starting, stopping, or halting the controller via the module inputs. The user can adjust the controller by two parameters:

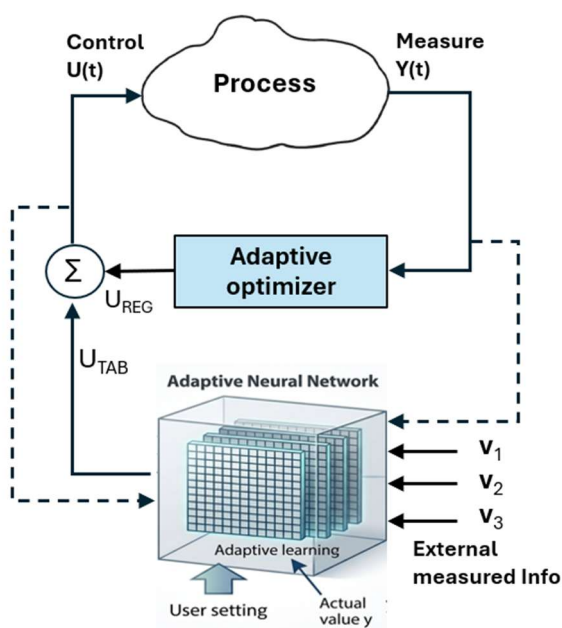
- the curvature a of the parabola, shown above
- the forgetting factor, which determines how data are used in the parabolic model

The user employs these parameters to tune the controller during commissioning.

An Extended Predictive Optimizer

The simple feedback optimizer described above can operate independently in any PLC/ICS system. However, it is often desirable also to be able to “remember” the optimal setting and how it is affected by other measurable signals in the process, for example in order to correct a fixed control curve during operation.

We solved this problem by using an adaptive neural network as a complement to the adaptive optimizer according to the configuration below.



For the adaptive neural network, see:

<https://www.firstcontrol.se/wp-content/uploads/2026/03/Selftuning-AI.pdf>

In this case, the adaptive neural network describes how the optimal setting U is influenced by other measurable variables, here denoted V_1 , V_2 , and V_3 , although there may be more. The model created by the network is:

$$U(t + k) = f(U(t), Y(t), V_1(t), V_2(t), V_3(t))$$

This is illustrated in the diagram below. The adaptive neural network thus stores how the optimal setting is affected by other measurable variables beyond those used directly by the adaptive optimizer.

The optimizing control value therefore consists of two parts:

- U_{TAB} , which is calculated by the network as the estimated optimal setting
- U_{REG} , which is calculated by the optimizer and is used for the final adjustment

The stored optimization curve in the network can thereby be used as a correction curve to a fixed control curve, for example in a Kaplan turbine or an internal combustion engine, or as a predictive control of a moving optimum.

Applications

There are a very large number of processes in which an optimizer could be used, either as a simple corrective function to improve efficiency in a process or as an addition to an existing optimal control curve.

The concept described above has been installed in oil burners and in a Kaplan turbine, in both cases at the direct request of suppliers.

The installation in the Kaplan turbine was permanent and was intended to correct the so-called CAM curve for optimal operation, which had previously been implemented as a fixed control curve. The supplier was convinced after the optimizer identified a correction curve, particularly evident at one operating point.

Two test installations were also carried out in oil burners. The purpose was to maximize efficiency by controlling the air/fuel mixture so that the total energy loss was minimized. This loss consisted partly of heat losses through the chimney and partly of chemical losses in the fuel. Both installations were operated for 2–3 weeks and gave very promising results.