Electronic Design

ALAN R. EARLS, Contributing Editor

In an Uncertain World, Adaptable Process Control is the Smart Move

Supply chains are in flux and process operations need to be ready and able to adapt to both changing inputs and a need for changing outputs. This puts a premium on adaptability.

A certain amount of flexibility is always good to have in any process-control system. But with supply chain uncertainty increasing, it's more important than ever to build in flexibility and adaptability so that small or large changes in input or desired outputs can be most easily accommodated.

There are a few ways to design adaptability into a process-control system. For instance, you can implement adaptive control strategies that enable the controller to adjust its parameters in response to changing process conditions. This involves estimating process model parameters on-line and then modifying the controller settings accordingly. Such an approach, often called self-tuning control, can maintain optimal performance even when uncertainties or disturbances are present.

It's important to determine if the process being controlled has or might have characteristics that change over time, such as variations in load, temperature, or material properties. Those in process industries know only all too well how subtle or even completely undetected variations in inputs of materials can lead to unexpected and usually undesirable results. As supply chain uncertainty becomes more common with shifting trade patterns and tariffs, process industries must necessarily be ready to take on the unknown.

While much of this is an operational problem that operators and engineers will need to solve, the problem can undoubtedly be better managed with underlying control and instrumentation system that adapts and adjusts.

How Can Self-Tuning Support an Adaptive Control Strategy?

Self-tuning control automatically tunes the controller parameters based on real-time process data.

For example, proportional-integral-derivative (PID) controllers have been an industry standard for feedback control for decades. However, their effectiveness depends on proper tuning, which must be tailored to each application and periodically adjusted to maintain performance.

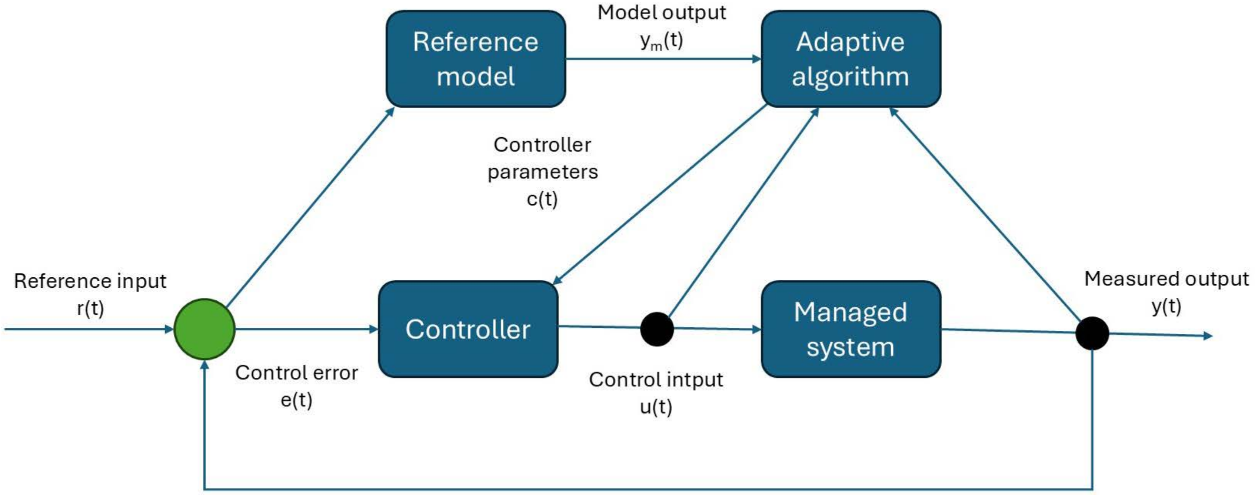
Auto-tuning controllers simplify this process by generating appropriate tuning parameters upon request. With control temporarily disabled, the operator initiates the tuning process, during which the controller analyzes system behavior to determine optimal settings.

Self-tuning controllers go further by continuously adjusting PID parameters during operation to sustain optimal performance. In contrast, auto-tuning controllers perform this adjustment only once before switching to closed-loop control. Many commercial PID controllers offer both features, and the terms are often used interchangeably.

For example, in a temperature-control system for a reactor, you might use self-tuning control to adjust the controller parameters (P, I, D gains) based on changes in reactor temperature, pressure, or the composition of the reactants. Thus, the controller would maintain the desired temperature even if the process dynamics change due to variations in reactor load or ambient temperature. PIDs can also be applied to optimize linear servo motors.

Auto-tuning typically replicates the procedures that an experienced engineer would follow when initially configuring a control loop. Self-tuning is more complex, as it requires the controller to maintain system stability while learning and adapting in real-time. The right components can be helpful, for example, when implementing a temperature-control loop for precise thermal stabilization.

Most of these capabilities exist in available commercial products, but their capabilities must be matched to need,



Shown is a generalized configuration of a Model Reference Adaptive Control (MRAC).

with an eye to ensuring adequate adaptability.

How Can Model Reference Adaptive Control Improve Adaptability?

Model Reference Adaptive Control (MRAC) is a control technique where a controller adapts its parameters in real-time to track the behavior of a reference model, even when the system being controlled has uncertainties or variations *(see figure)*. The controller adjusts its gains or parameters to minimize the difference between the system's actual output and the desired output from the reference model.

Implementing MRAC requires a digital signal processor (DSP) or a microcontroller with sufficient processing power to execute the control algorithms. You'll also need sensors, actuators, and potentially other hardware components like a power supply and communication interfaces, depending on the specific application.

The DSP or microcontroller must be able to perform the complex calculations required by the MRAC algorithm, including parameter estimation, control law computation, and potentially signal processing from sensors. And it needs enough memory to store the control algorithm, reference model, and potentially the estimated parameters.

The reference model defines the desired behavior (e.g., response to a step input) that the controlled system should achieve. The controller is the part of the system that computes the control input based on the reference model, plant's current state, and the adaptation algorithm.

The adaptation algorithm adjusts the controller parameters (e.g., feedback gains, feedforward gains) based on the tracking error between the plant's output and the reference model's output.

The five steps in MRAC operation are:

1. **Define a reference model:** The desired behavior of the system is defined by a reference model.
2. **Observe the system's response:** The actual output is observed.
3. **Calculate tracking error:** The difference between the actual output and the reference model's output is calculated.
4. **Adapt controller parameters:** The adaptation algorithm adjusts the controller parameters to minimize the tracking error.
5. **Repeat steps 2-4:** The process continues until the system's output closely matches the reference model's output.

The Different Types of MRAC

There are two main types of MRAC: direct and indirect. Direct MRAC means the controller parameters (e.g., feedback gains) are directly adapted based on the tracking error. On the other hand, indirect MRAC is when the controller parameters are adapted indirectly by estimating the plant's parameters first and then using these estimates to calculate the controller gains.

In either case, the advantages of MRAC include robustness, simplicity, and ability to deliver performance. MRAC can provide robustness by handling variations in system parameters and uncertainties. It uses algorithms that can be relatively straightforward to implement. By adapting to the system's dynamics, MRAC can help achieve a desired level of performance.

On the other hand, standard MRAC will exhibit high-gain control, which could be problematic for stability. Moreover, MRAC algorithms can be non-linear, which makes design and analysis more complex.

What are Some General Design Considerations to Bear in Mind?

Ensure that the adaptive control algorithm is stable and robust to changes in the process and disturbances. The adaptive control algorithm should converge to a stable operating point quickly and reliably. Also, the system should be able to maintain its performance even in the presence of uncertainties or disturbances.

The adaptive control system should achieve the desired performance goals, such as accurate tracking of setpoints or disturbance rejection. Be mindful that an adaptive control algorithm can have a higher computational load than traditional controllers, so consider the available processing power and plan accordingly.

References

Autonomic Computing Now You See It, Now You Don’t

Bridging Model Reference Adaptive Control and Data Informativity