Cahners
EUROPE

CONTROL ENGINEERING EUROPE

Control, Instrumentation and Automation in t dustries

February/March 2001

New SCR package drives industry's smallest soft starter, p34

Why Europeans are buying motor/controller combinations, p37

New power for PLCs: model-based predictive control, p44

The latest in factory floor operator interfaces, p50

A first look inside Dr. George Cheng's 'model free' adaptive controller, p25

‘Model free’ adaptive control

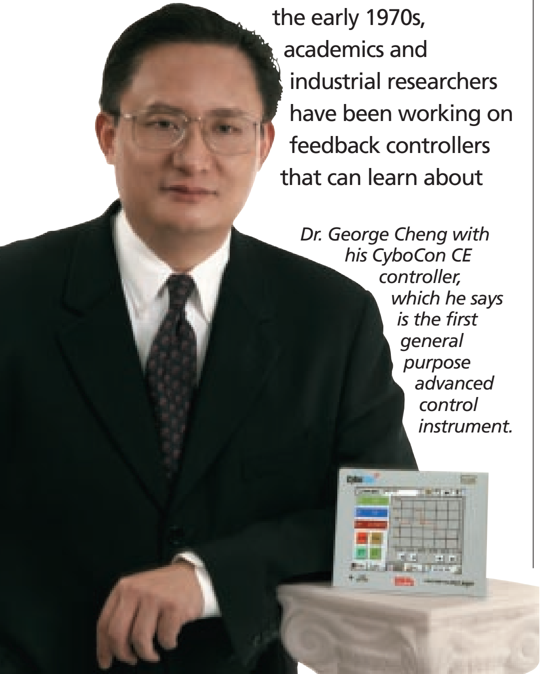
This new technique for adaptive control addresses a variety of technical challenges

PID loops control a majority of the automated processes in industrial facilities. The proportional-integral-derivative algorithm is both simple and reliable and has been applied to hundreds of thousands of control loops over the last 50 years.

However, not all industrial processes can be controlled with basic PID loops. Multivariable, non-linear, and time-varying processes, for example, all require more advanced control techniques. There was a time that such technology was available only in academic laboratories and in the aerospace industry, but the advent of affordable computing platforms has made even the most arcane algorithms accessible to industrial users.

Adaptive control

Adaptive control is a case in point. Since the early 1970s, academics and industrial researchers have been working on feedback controllers that can learn about and adapt to gradual changes in the behaviour of the controlled process.



Dr. George Cheng with his CyboCon CE controller, which he says is the first general purpose advanced control instrument.

Of course, all process controllers are “adaptive” in that they force a process to adapt its outputs to the values desired by the operators. However, most process controllers do so according to algorithms that are designed (or at least fine tuned) by the operators before the controller ever starts its work. The operators may periodically re-tune the parameters of a traditional controller, but this is generally a manual operation performed only after the controller’s performance has begun to deteriorate for some reason.

A truly adaptive controller can update its tuning parameters all the while it is in operation so that its performance remains optimal, even if the behaviour of the process changes. This amounts to automatically updating the controller’s entire strategy to accommodate the new behaviour of the process.

For example, an adaptive controller that is initially tuned to provide aggressive control for a sluggish process will substitute more conservative tuning parameters if it detects that the process has somehow become more responsive to control efforts. A traditional controller with fixed tuning parameters would continue to control the process aggressively, causing the process outputs to fluctuate excessively.

Expert systems

Adaptive controllers come in all shapes and sizes. Perhaps the most popular is the self-tuning PID loop capable of generating its own tuning parameters. First introduced in the early 1980s, some sort of on-demand or continuous tuning operation is now available in most stand-alone loop controllers.

Many self-tuners take an expert systems approach to the problem of updating their P, I, and D parameters. They try to emulate the thought process of an expert control engineer by tweaking the tuning parameters according to a complex set of empirical rules designed to improve closed loop performance.

This approach works well when the behaviour of the controlled process is simple and predictable. Many expert self-tuners rely on the assumption that the dynamics of the process can be completely quantified with just a gain, a deadtime, and a time constant. The controller may not need to know what those three values are, but it assumes that no other process dynamics affect the relationship between the inputs applied by the controller and the resulting outputs from the process.

This can be a problem when the process dynamics are more complicated. The expert systems’ rules may generate a spurious result simply because they do not embody the necessary expertise to handle the more challenging process.

Expert self-tuners can also have a hard time commissioning a new loop. Their rules are typically designed to account for gradual changes in the behaviour of the process or to correct the existing tuning parameters. Some sort of initial tuning must generally be implemented manually in order to get the controller started.

➤

Model-based adaptive control

A model-based adaptive controller offers a more mathematically exact alternative to the heuristic approach of an expert system. It bases its control decisions on an empirical model of the process that quantifies the input/output relationship as a differential equation. It also refines that model with recent input/output data as it continues to control the process.

Assuming that the latest model will remain valid in the near future, the controller then predicts where the process is heading and determines the control efforts required to steer it in the right direction. By constantly updating its process model, the controller attempts to account for unexpected or time-varying process behaviour that might affect the future results of current control efforts.

> All the information necessary to characterise the dynamic behaviour of a process is contained in the input/output data. There should be a way to compute control actions directly from the I/O without any model.

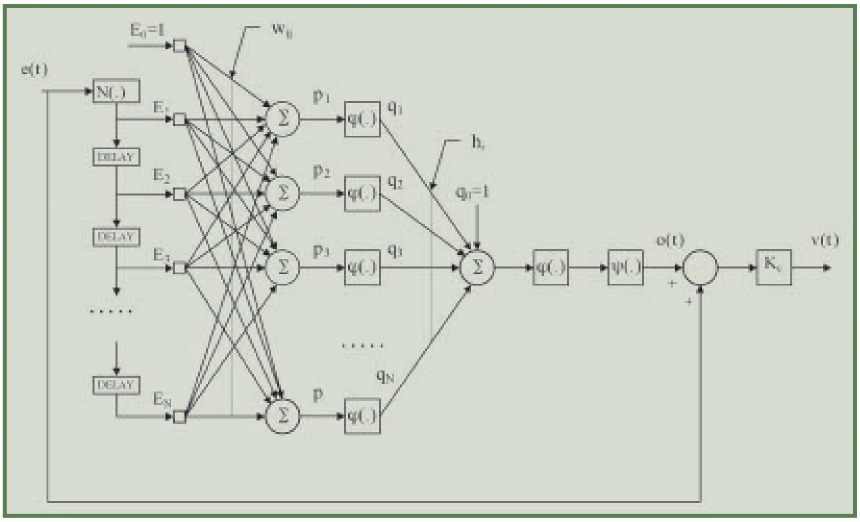
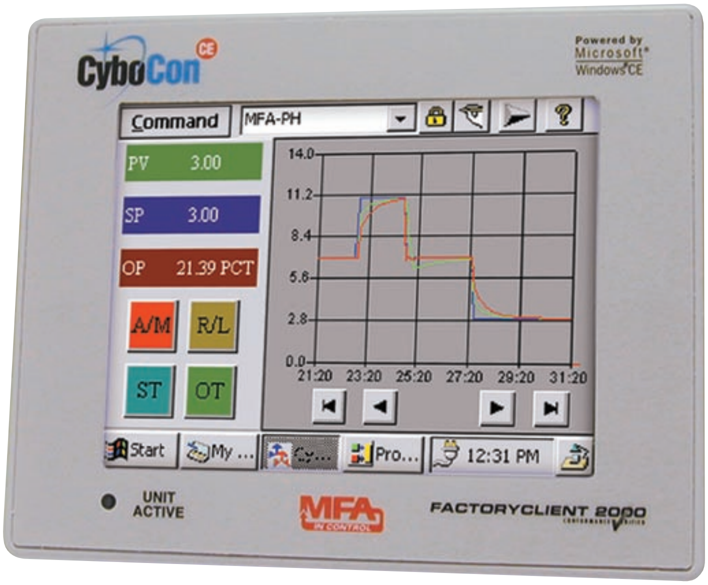


Figure 1 – The MFA controller computes each control effort directly from a history of process errors using this 3-layer neural network with time-delayed functions, activation functions, weighting factors and other components. To perform its function, the controller analyses error measurements recorded during the previous N sampling intervals (shown in the left column, E1, E2, ...), which allows it to observe and learn the dynamic behaviour of the process and compute the next control effort directly from the error history. The neurons and delay blocks are automatically configured; precisely how this is done remains a trade secret.

My own experience has shown that this procedure can accommodate a wider variety of process dynamics than rule-based controllers. Model-based adaptive controllers can also take advantage of numerous tuning formulas that have been developed to translate model parameters into controller parameters such as P, I, and D values. In fact, I suspect this is why on-line modelling has been so well accepted (in academic circles, at least). If an accurate process model can be determined from input/output data, the appropriate parameters for the controller are easy to compute.

A conundrum

Unfortunately, that can be a mighty big “if.” A controller that has been successful in holding the process variable steady also leaves itself with very little useful information about the dynamic behaviour of the process. Conversely, a



Originally implemented as PC software interfaced to a PLC or DCS, CyboSoft has been repackaged to run on this dedicated Windows CE box.

controller that has sufficient information to generate an accurate model must be forcing or at least allowing the process variable to fluctuate.

A poor model leads to poor control efforts as the controller attempts to regulate a process that does not react the way it expects. Ironically, that tends to cause the process variable to fluctuate which in turn allows the controller to improve its model. On the other hand, as the model and the subsequent control efforts improve, the information from the process becomes less and less useful and the model can be refined no further.

In the end, the best result that a model-based adaptive controller can hope to achieve is reasonably accurate models and reasonably successful control efforts. In the worst case, however, the closed loop system can become unstable if the mismatch between the process model and the actual process is severe enough.

Model-free adaptive control

So why generate a model at all? Theoretically, all of the information that a feedback controller requires to regulate the process is contained in the input/output data whether a mathematical equation is ever derived from it or not. After all, an experienced control engineer can look at a strip chart and tune a PID manually without ever calculating the gain, deadtime, or time constant of the process model.

It is certainly convenient to have a model of the process that distils the process' behaviour into a set of compact mathematical equations. And in spite of its limitations, model-based adaptive control technology has been applied successfully to a wide variety of control problems, especially when parts of the model can be derived from available knowledge of the process (such as a deadtime computed from a known transport delay).

Nonetheless, it seems to me that there should be a way to compute control actions directly from the input/output data without first creating any model at all. All the information necessary to characterise the dynamic behaviour of the process is already there; it should just be a matter of crunching the numbers correctly.

'Model free' adaptive control

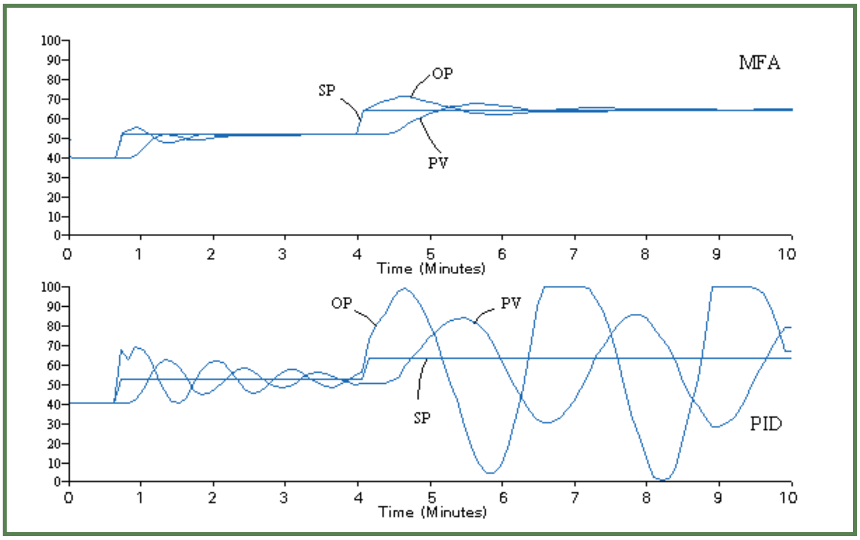


Figure 2 – Simulation results demonstrate the relative performance of MFA control (top) and manually-tuned PID control for a particularly difficult test case. Each controller's output (OP) and the resulting process variables (PV) are shown reacting to setpoint changes (SP) at times 0.6 and 4.0 minutes. The dynamic behaviour of the simulated process also changes at the 3.7 minute mark. In this case, the MFA controller is able to adapt to the new process dynamics while the PID controller becomes unstable.

The CyboCon product

Dr. George Cheng, president of CyboSoft, General Cybernation Group Inc., in Rancho Cordova, California shares that opinion. In fact, he claims to have designed a "dream controller" that can regulate time-varying, multivariable processes without any mathematical models or manual tuning procedures. He calls it MFA (Model-Free Adaptive) Controller.

CyboSoft originally implemented the MFA controllers in CyboCon, a software product to run on a PC interfaced to a PLC or DCS. The latest MFA product is CyboCon CE that runs under Windows CE on a dedicated control box. CyboSoft claims that it is the first general-purpose advanced control instrument.

Both products employ MFA controllers to minimise the variance of the process variable (the sum of the squared errors between the process variable and the setpoint). CyboSoft claims that it can effectively control various tough process loops and guarantee closed loop stability in most practical applications.

There are several variations on the basic MFA control algorithm to accommodate specific control problems. The standard algorithm works for a majority of processes, the "anti-delay" algorithm with its delay predictor is best for processes with more significant deadtime, the non-linear MFA is designed to control non-linear processes such as a pH loop, and the robust MFA to control process with significant disturbances.

Exactly how MFA manages to provide effective feedback control in all these situations with no model of the process and limited operator intervention has until recently been a carefully guarded secret. However, now that Dr. Cheng has successfully patented his technology, a few more critical details have emerged.

How it works

Like any feedback controller (adaptive or not) MFA looks at the error between the setpoint and the process variable to decide how best to manipulate the process. But unlike conventional process controllers, MFA analyses a whole string of error measurements recorded during the previous N sampling intervals. This is what allows the controller to observe the dynamic behaviour of the process.

However, MFA does not attempt to create a fixed model of the process from those observations. Instead, it computes the next control effort directly from the error history using the neural network shown in Figure 1.

Artificial intelligence experts will recognise Figure 1 as a traditional *multilayer perceptron network* with various *filters* (the φ's and ψ's) and *weighting factors* (the w's and h's) that give the network the ability to recognise or learn patterns embedded in a set of measurements. The unique feature of MFA's neural network is its dynamic nature. By continuously updating the values of the weighting factors, it repeatedly helps generate adaptive control signals at the end of every sampling interval towards the objective of minimising the error between the setpoint and process variable.

For those who are not as familiar with artificial intelligence theory, Dr. Cheng points out the key elements of his neural network that makes it applicable to process control. First is the mechanism for storing the history of error measurements. That is accomplished by the delay blocks at the input end of the network (on the left). The most recent error measurement enters the top block and pushes each of the previous error measurements further back (down) into the history file.

Second, the network multiplies each historical error measurement by the weighting factors, then sums and filters the results to produce its output o(t). Finally, o(t) is added to the current error measurement e(t) and that sum is multiplied by a user-selected gain K_c to produce the actual control effort v(t).

So what?

It is not at all obvious (to me, at least) how these mathematical manipulations produce a control effort with all the properties that Dr. Cheng claims. However, his research and various field applications have shown that the MFA controllers do have the capability that a regular PID controller does not possess. Simulations such as those shown in Figure 2 bear out this conclusion.

What I can see from Figure 1 is that MFA works something like a PI controller with a variable integral action and an overall gain of K_C. The proportional control action is provided by the feedforward path that adds the current error measurement to the output of the network. The neural network provides the integral action by summing all the previous error measurements. The only difference between this operation and simple integration is that each element of the network’s sum is weighted and filtered differently and the weighting factors change over time.

So, assuming that the MFA control algorithm for computing those all-important weighting factors works as claimed, the next question is why a practising control engineer with practical control problems would care. Or more to the point, what makes MFA a superior alternative to other forms of adaptive control, model-based methods in particular?

MFA corrects a “PID problematic” loop

Air Liquide America, a specialty gases manufacturer headquartered in Houston, needed to optimise production of air separation units operating around the world. The company originally installed a model-based predictive control system which led to considerable improvements in plant stability and product yields/ production.

However, plant engineers were still having problems maximising argon production due to poor regulatory control in the main column. According to Dave Seiver, Air Liquide’s advanced control engineer, and Brian Keene (Air Liquide’s plant manager in McMinnville, Oregon), ineffective regulatory control in the air separation columns had been causing oscillations throughout the process. Attempts to re-tune the PID loops that had been valiantly attempting to control the process had failed.

Enter CyboCon. In less than one day Mr. Seiver’s team was able to convert the “PID problematic” loop to MFA control and dramatically reduce the process variations. That in turn allowed an immediate increase in argon production to record-setting rates, increased plant stability, and plant operations buy-in to the technology.

Advantages

Dr. Cheng cites three principle advantages of his approach to adaptive control: robustness, speed, and stability. Believe it or not, he can prove that so long as the process is *passive, controllable,* and *does not change sign,* the closed-loop MFA system stability is guaranteed whether the process is linear/non-linear, time-invariant/time-varying, or single/multivariable.

The sign-change prohibition is perhaps the most obvious requirement. I don’t know of any controller that could stabilise a loop where the process suddenly switches from being direct acting to reverse acting. Controllability is also a fairly straightforward requirement. It means, in essence, that the actuator being manipulated by the controller must actually have an influence over the process variable. The temperature of a 100,000 litre tank of molasses being heated by a 10 watt light bulb in the next building would not be controllable (by MFA or any controller).

A passive process is one that may store but does not generate energy. The tank of molasses would be passive, but a reactor would not. Dr. Cheng has proved in his Model-Free Adaptive control theory that MFA itself is a passive circuit. Non-linear stability theory states that a passive controller working on a passive process will produce a stable closed-loop system. Fortunately, most control problems in the process industries involve processes that are in fact passive.

Convergence

Dr. Cheng also claims that MFA is faster than most model-based adaptive controllers. Since MFA does not include an identification mechanism like most model-based control methods do, it can start applying effective control actions much sooner than a model-based controller can.

The difference is a matter of *convergence*. A model-based controller will attempt to create a mathematical model of the process that will remain fixed for at least the near future. Its estimates of the model’s parameters must therefore converge on some constant values that best represent the dynamics of the process. That can take a long time, especially when useful input/output data is sparse because the process is quiescent.

In contrast, the parameters that MFA computes – the weighting factors – need not converge at all. As conditions in the process change due to disturbances or variations in the process dynamics, so do the weights. They actually have no fixed values to converge on.

Furthermore, a quiescent process does not slow MFA’s learning operation the way it does a model-based controller. Or more precisely, MFA need not learn anything at all when nothing is happening in the process. That’s because a quiescent system does not require any control effort (it’s already in a steady state) and a zero control effort is exactly what MFA computes in the absence of any useful information from the process. However, as soon as the process begins to react to a disturbance or a setpoint change, MFA will start generating proper control output to minimise the error.

Robustness

A robust controller can regulate a process with uncertain behaviour or a process that changes its behaviour over time. However, unlike a true “black box” controller that needs no hints about the process whatsoever, a robust controller generally requires estimates of the process’ gain, deadtime, and time constant(s). The more robust the controller, the less accurate those estimates need to be.

Although it may seem that MFA tries to be a black box controller, it is actually better characterised as robust since it does need some qualitative information about the process. The user needs to come up with at least a rough estimate of the process time constant. For the Anti-delay MFA, an estimate of the process deadtime is also required. And by adjusting the controller gain K_c to obtain an overall gain of 1, the user implicitly defines the process gain as K_c^{-1}.

Fortunately, the user need not provide particularly accurate estimates of the process parameters. Dr. Cheng claims that MFA will still provide reasonably good control of the process if the user's estimate of the time constant is as much as 300% larger or smaller than the process' actual time constant. The deadtime estimate can be off by as much as 200% or so.

Anti-delay control

MFA's ability to deal with that much uncertainty in the process deadtime makes it particularly robust. A conventional controller will become impatient if the process exhibits a deadtime that is longer than what the controller was designed to handle. Its control efforts will appear to have no effect until after the real deadtime has elapsed. By that time the controller will have applied additional control efforts in a futile attempt to force the process variable to change sooner than it possibly can. All those extra efforts will eventually move the process variable, but by then it may be too late. The process variable may start to swing wildly as the controller keeps trying to correct previous mistakes by making more and more.

The Anti-delay MFA uses a special delay predictor to avoid this problem. The predictor produces an artificial error signal before the deadtime has elapsed. This allows the controller to "feel" the effects of its control efforts almost immediately and thus avoid second guessing itself. Unlike a traditional Smith Predictor, however, the Anti-delay MFA's predictor does not require a precise model of the process, only the user's rough estimates of the process deadtime and time constant. If those estimates don't match the actual process parameters, the MFA adaptive algorithm will take up the slack.

Disadvantages

If MFA has an Achilles heel, it would have to be its complexity. Traditionalists familiar with linear control theory like to know in advance what a controller is going to do with a given set of inputs. That's not possible with the convoluted calculations performed by a neural network. Not even Dr. Cheng himself knows exactly what goes on inside the network during each step of the calculation. Only the final result is at all predictable.

At the other extreme, users with no process control experience at all will be disappointed if they expect MFA to function completely without operator intervention. It is not a true black box controller because it does require some information from the user. Some of the required operator inputs, such as which sensor to poll for process variable measurements and which actuator to use for applying the control effort would have to be supplied to any controller. But there are other inputs, such as the time constant estimate, that a completely adaptive controller would be able to figure out on its own. Nonetheless, MFA has been successfully applied to a wide variety of process control problems. It may be unconventional and complex, but it's worth a try when traditional control schemes fail. ❖ *— Dr. Vance VanDoren*

For more information on the CyboCon controller, please enter number **200** *on the Reader Service card or contact CyboSoft, 2868 Prospect Park Drive, Suite 300, Rancho Cordova, Calif. 95670 USA; e-mail GCGroup@cybosoft.com; tel +1 916 631 6313; fax +1 916 631 6312; www.cybosoft.com*