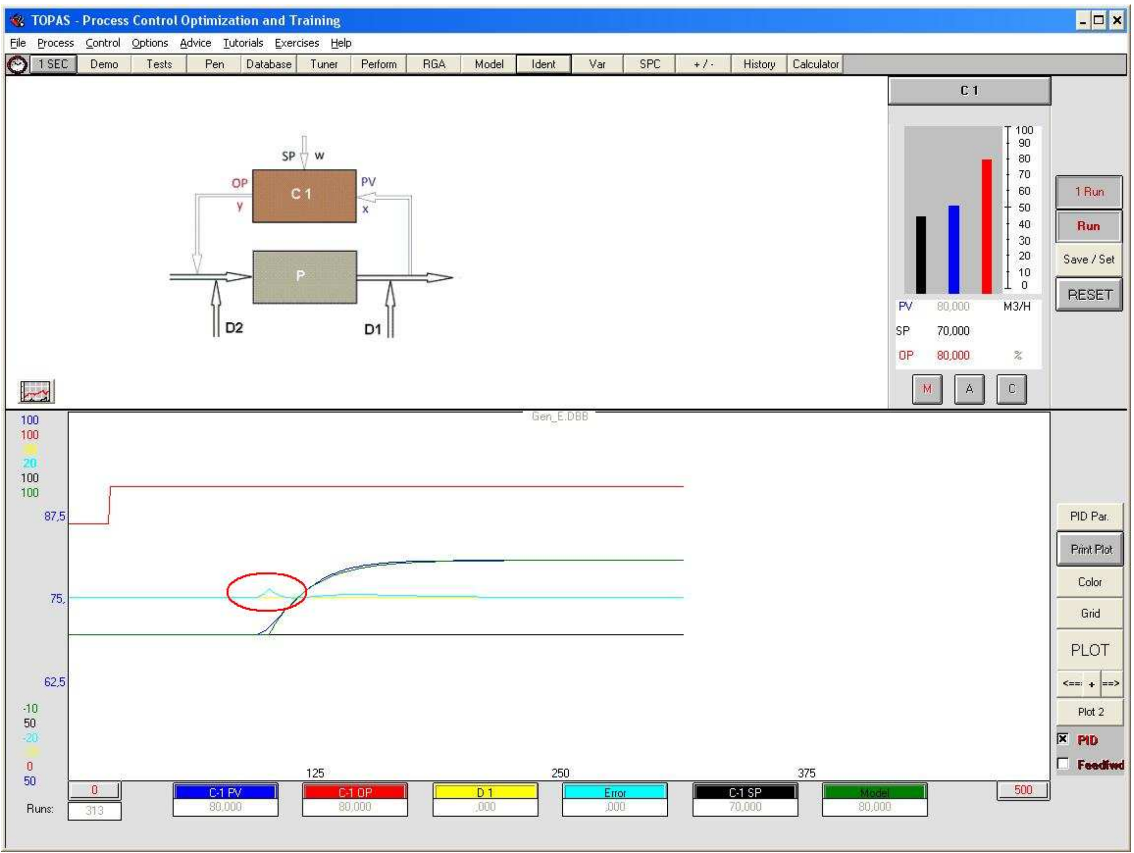
Dealing with long deadtime – a comparison between PID, PI, Smith Predictor and Model Based Control (MBC)

For this comparison we use a simple second order process. As a first step we get a first order approximation of the process parameters (here from an open loop test, but we could do this also from an closed loop test – ***not*** the Ziegler-Nichols test, just a “normal” setpoint step). In the case at hand the ratio deadtime / time constant = 4, a situation thus where the use of the standard PID is not recommended any more.

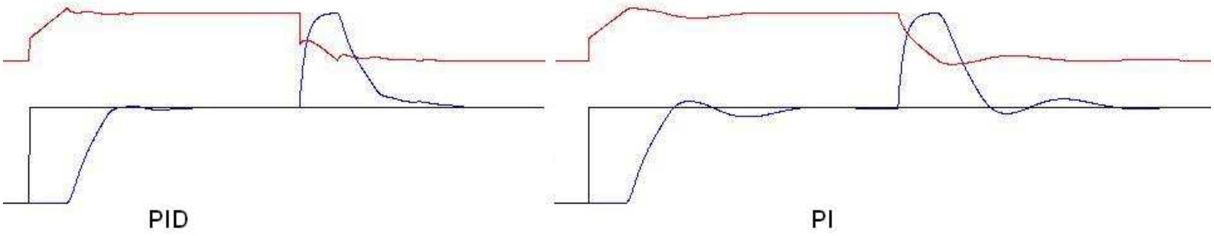


We can see the step made with the controller output (red), the process response - the PV (blue), the model based on the estimated process parameters (dark green) and the model error (light blue), that is the difference between the second order process and the first order representation, the model. Note the model error at the beginning of the response (red circle), it will become important when we are dealing with the Smith Predictor and the MBC: the PV already moves, but the model still represents this part as deadtime.

Next, we calculate the optimum PID settings. We will always use the result of the calculations without any rounding and without any further adjustment. The result:
Kc = 0.675, Ti = 0.88 min, Td = 0.298 min

Note that the D-action is quite strong. Now we make first a setpoint change and trigger afterwards an upset, a change in the disturbance variable D1.

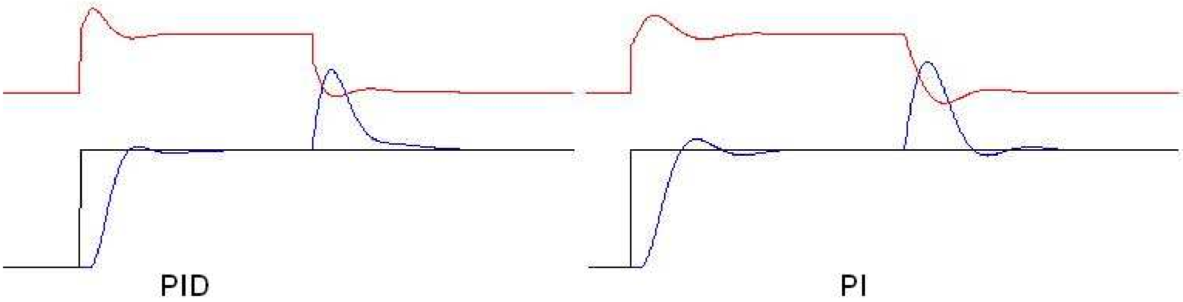
For comparison, we also calculate the settings for the PI controller (no D-action) and repeat the two tests. The tuning parameters are: Kc = 0.681, Ti = 0.938 min.



The setpoint is drawn in black, the PV in blue and the controller OP in red.

This case is next to ideal (with other words: a bit unrealistic) because neither trail and error tuning nor most tuning methods will deliver such good PID settings: First off, we are using a simulation, thus the data used for the process parameter estimation are ideal: no noise, no outliers, nothing. Also, our identification methods deliver very good estimates, here the regression coefficient was 0.998 (!), we have a perfect “picture” of the process (of course, both the Smith and the MBC later are also taking advantage of the quality of the process parameters). Finally, for the PID tuning we are using not always the same method, but the one which gives the best results for the specific situation.

For comparison sake let us make these tests again, but now with a process that has the same time constants but a much shorter deadtime und thus a ratio deadtime / time constant of 0.75 (instead of 4 before).



	Short deadtime				long deadtime			
Tuning	***PID***	***PI***			***PID***	***PI***		
Kc	1,852	1,146			0,676	0,682		
Ti	0,487	0,368			0,878	0,936		
Td	0,079	-			0,297	-		
Td/Ti	0,16				0,34			
			Diff	***%***			***Diff***	***%***
SP - ITAE	5464	8247	2783	50,93	75257	118291	43034	57,18
IAE	287	339,3	52,3	18,22	1162,3	1310,4	148,1	12,74
Dist - ITAE	7325	10130	2805	38,29	97709	127497	29788	30,49
Std. dev.	2,09	2,65	0,56	26,79	3,33	3,5	0,17	5,11
IAE	276,3	307,4	31,1	11,26	1169	1293,1	124,10	10,62

Note that:

- The setpoint is considered the optimum value for the variable in question. Thus, any deviation from the SP means an economical loss. The Integrated Absolute Error (IAE) is a measure of that loss. While for the fast process (short deadtime) the difference in the IAE between the PID and the PI was ~30 for the disturbance case (last row, green cell), it was 4 times higher for the slow process (orange cell): We are suffering a much higher economical loss in case of the slow process – as expected.

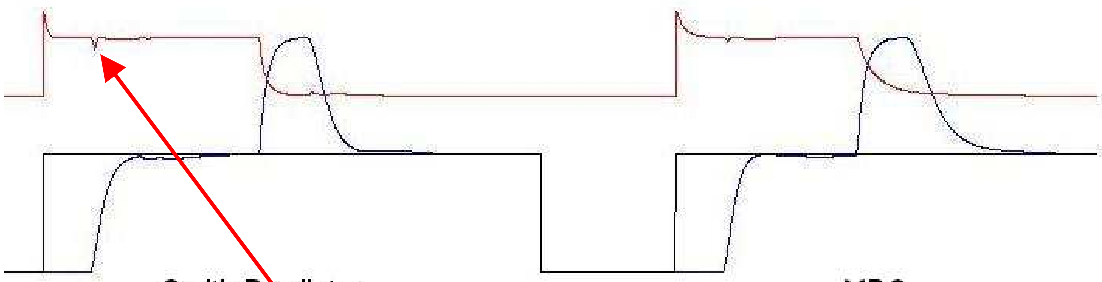
- To achieve its remarkable performance in the case of long deadtime, the PID needed quite some strong D-action, the ratio Ti/Td is twice as high as for the fast process
- For setpoint changes (long deadtime), the performance difference between the PID and the PI (measured by the ITAE) is more the 57%!

Some further observations regarding the tuning methods:

- The three best results for the fast process are quite a bit apart: The second best delivered a 24%, the third a 42% (!) higher ITAE
- The results of the various tuning methods differ also for the slow process, yet not as much as before (ITAE of second is 10%, of third best is 17% higher). Remarkable is, however, that none of the best three methods for the fast system delivers a top result for the slow case: In the order of the previous ranking they show ITAE values that are 260, 38 and 223% higher than the best
- Note that we use here methods that are open domain (except our own one) and take the results as calculated without any attempt to manually refine them further.

Coming back to the comparison between the PID and the PI: To achieve good performance and thus to minimize the economical loss, the D-controller is most helpful and needed.

As a next step we compare these results first with the Smith Predictor and then with a model based controller (here our own controller called AMC) for the process with long deadtime. In both cases we use the initially identified process parameters for the model.

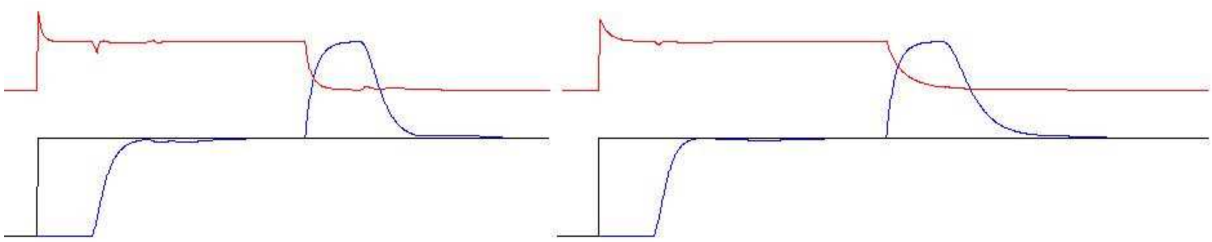


Smith Predictor MBC

Unlike as for the PID/PI case we spend here in this comparison no effort in optimising the tuning of these controllers, we just give it some reasonable figures. For the Smith Predictor we use a PI controller with Kc = 2 and Ti = 0.2 min, a quite aggressive setting. Also the MBC controller is tuned quite tightly. Note the effect of the model error mentioned in the beginning. The ITAE results for setpoint changes are:

Smith 56506 … 33% better than the PID
MBC 50355 … 49% better than the PID

We repeat these tests with a 10% difference in all tuning parameters, always in the less favourable direction:

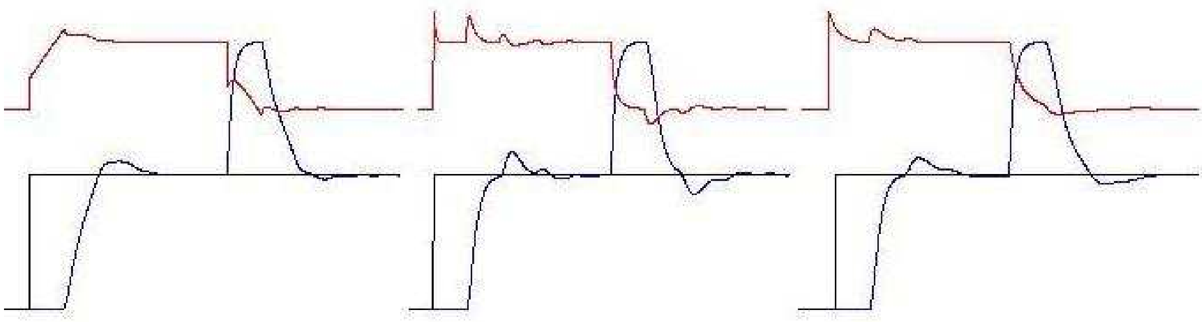


Smith Predictor MBC

As can be seen (and also measured), the tuning has much less influence as in the case of the PI(D). More important is the quality of the model(s).

Let us explore thus what would happen if the deadtime would increase by 15%, e.g. due to a reduction in the speed of a device or the flow rate of a product. All other process parameters are still unchanged. For all controllers (we skip the PI here) we use the same tuning as in the base case. The ITAE results for the setpoint tests are:

PID	108928
Smith	84311
MBC	84284

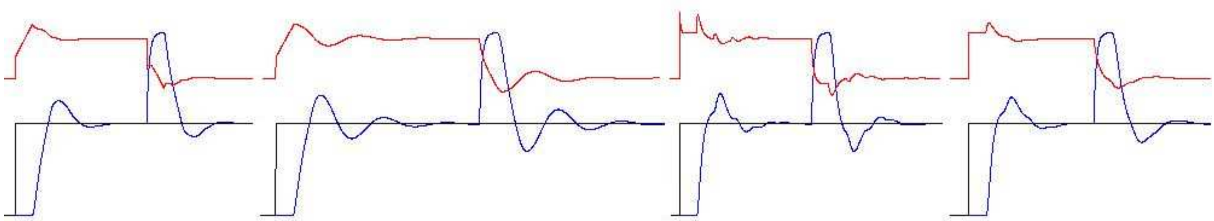


PID Smith Predictor MBC

The PID result suffers from the slow start, because the P-gain had to be set very small. Due to the aggressive tuning chosen, esp. the Smith shows quite some unsettled behaviour, the response to the model error. That could be smoothened out, but is on purpose not done: We want to see how a very tightly tuned controller would respond to changes in the process.

To make things worse, we explore now a case where all three process parameters, the process gain, the deadtime and the time constant, increase by 20%. With other words, we assume that the process is very non-linear and also that the dynamics are changing with the operating conditions, a situation we can e.g. see for a heat exchanger temperature controller when the product throughput is reduced. The ITAE results for the setpoint tests are:

PID	148110
PI	270470
Smith	136920
MBC	142321



PID PI Smith Predictor MBC

We can see that:
1) The PID is catching up because of its outstanding tuning. Again: We used here the best result of several tuning methods, the PID really got a head start
2) The PI controller is losing even more ground
2) The Smith Predictor acts quite unsettled due to the aggressive tuning, yet achieves the best ITAE
3) The MBC performance (ITAE) comes close to the Smith, with a much smoother behaviour

How much error in the parameters (coming from a poor estimate or changes in the process) can be tolerated depends for the Smith Predictor and the MBC very much on the tuning: A aggressively tuned controller – as in our case - will obviously be less tolerant to model errors then a moderately tuned one. Thus, one cannot make a general statement like "The xx controller tolerates only y% model errors".

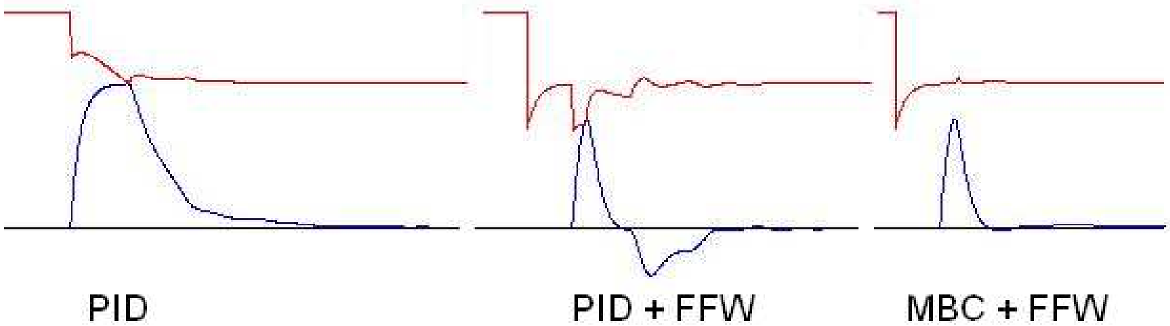
So far we concentrated on setpoint changes. Especially for processes with long deadtime the PV deviates far from the setpoint and it takes a long time before it returns. To reduce the associated economic loss, a feedforward is much more needed than for a fast process.

Feedback + feedforward

We want to explore specifically a situation where the deadtime of the disturbance is shorter than the deadtime of the manipulated variable, a difficult situation since the feedforward can never act in time. The difference chosen here is 25%.

In the first test series we use the process and the process parameters as initially estimated. To start, we repeat the test with the PID controller without the feedforward - just for comparison sake - then activate the feedforward (short FFW) and make first a test with the PID and then with the MBC. All tuning constants are still the same.

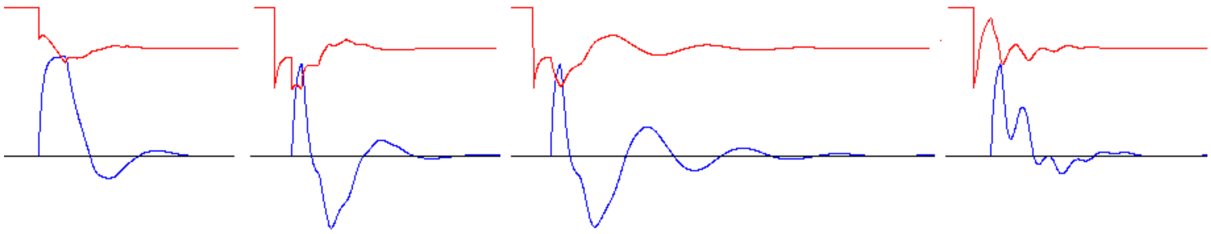
The Smith will not be used here any more since it has (in its original version) a problem with the feedforward: The response to the feedforward action is showing up as model error causing harsh reactions by the PID. Of course, one could expand the Smith, it is not a big effort, but this would not be the version used by the control community at large. The results (in terms of the IAE) speak for themselves:



PID (w/o FFW): 1169   PID + FFW: 432   MBC + FFW: 238
(PI + FFW 508, not shown)

We see here another fundamental difference in the behaviour of the PID and the MBC: Since the FFW cannot fully compensate the effect of the disturbance (the disturbance being too fast relative to the MV), the PV will always deviate from the SP for some time, but would eventually return to it if the FFW gain was correct. Thus no feedback action would be needed at all. Yet, the PID, being a "reactive" controller, just "sees" the deviation from the SP and acts – which leads to the "undershoot" seen in the second test. The MBC "knows" about the FFW action and therefore does not make things worse. That is the reason for the superior performance – about half of the IAE. The PI (no D-action) is again behind (this time 18%).

To make things worse, we will now investigate a situation where all process parameters have changed by 20% - always in the more critical direction. The difference in the deadtime between the disturbance variable and the MV is now 33.3% – a critical situation where the use of a FFW becomes questionable. The IAE results are:



PID (no FFW): 1469   PID+FFW: 1219   PI+FFW: 1701   MBC+FFW: 804

Despite its "ugly" behaviour (could be smoothened out by tuning), the MBC is delivering by far the best case - 50% better than the PID - and proofs an old myth wrong: "If the model parameters of an MBC deviate more the 10% the whole system will become totally instable".

There are numerous MBC approaches out there, but only some of the very early ones (like e.g. the Dahlin or esp. the deadbeat controller) were really that sensitive to model error. Even in the late 1970's, when I developed my first MBC applications and the process parameter estimation methods and tools really left a lot to be desired, I did not have severe problems with the model quality, yet could handle situations unmanageable for the PID.

Adaptive control

One more remark concerning controller performance: If conditions are changing frequently and significantly, then it becomes necessary to resort to adaptive control. It should be noted that with MBC (and also with the Smith to some extent) this is easier to do than with the standard PID. Let us assume that the process parameters are a function of the throughput – which is often the case. Then we can measure the current throughput and estimate the resulting changes in the process parameters.

For MBC and Smith Predictor we just need to plug these new values into the models – that's it. Adaptation of the tuning is only needed upon truly drastic changes. For the PID, however, we need one more step to do, namely to calculate the tuning based on the new process parameters. Since experience shows that no tuning method gives optimum results over a wide range, we better use more than one, consequently the implementation in the DCS can get a bit tricky.

There are some more, less obvious advantages of model based control over the PID and also over the Smith Predictor. Just to mention one: Concerning setpoint changes, the performance of the PID (also the Smith) is governed by the strength (height) of the initial P-kick. Thus, everything that eliminates or mitigates this kick like ramping the SP or putting speed (ROC) limits on the output, greatly impairs the performance.

Especially for ramping we face the dilemma that either with smooth tuning the PV follows the SP in quite some distance – we lose precious time – or with tighter settings it overshoots the final SP - which is often not allowed. With MBC, the PV can follow the SP in the shortest distance as dictated by the deadtime without overshooting at the end.

Comparison of the needed effort

After all these performance comparisons let us finally take a brief look at the necessary effort. First we need the process parameters, which takes with a suitable tool about five minutes for data import and inspection and about a minute or two (regardless whether we use the results of an open or closed loop test) for the identification.

This step should be done in all cases, also for the PID, because calculating the tuning afterwards is much faster than trial and error tuning. Knowing the process parameters allows also to calculate the Controllability Ratio und thus to decide about the controller type to be used on a solid basis. It also enables us to simulate the loop and test it e.g. for disturbance handling, which is typically at least difficult if not impossible to do on the real process.

In the case of the standard PI(D) we next need to calculate the tuning parameters and to plug these 2 or 3 values into the controller. That takes another two or three minutes.

For the Smith, we have to enter the process parameters into the two models (five values in total) and to give it some reasonable tuning parameters (2 values). We would not even need to calculate them, just estimate them directly from the process parameters. Also here, the effort is just a couple minutes.

Also for the MBC (with the model formulation used here, many different model types are possible) we just have to enter the 3 process parameters and then supply some reasonable tuning (2 to 4 parameters), that can be easily estimated from the process dynamics or calculated. We could even run the MBC without any tuning but would lose a bit on performance. Again, the total effort is two to three minutes.

The bottom line is that, once all these controllers are installed and ready for use in the DCS, the needed effort to use them is about the same in all cases and thus not a criterion for use. The criteria are the process behavior and the required performance.

These controllers should indeed make part of our Technology Set in the system, yet unfortunately only the PID is normally available. The effort to install a Smith Predictor or a simple MBC like the one used here (incl. a feedforward) in the DCS depends of course on the system but should in most cases not exceed half a day.

Summary:

We have seen that:

- It absolutely pays out to determine the process parameters first – today we have the methods and tools - and then to decide about the controller type, based on the Controllability Ratio
- For the PID, it pays out to calculate the tuning rather than applying trial and error, however, no single tuning method covers the whole range of situations satisfactorily
- The longer the deadtime, the more the help of the D-controller is needed – properly dosed, of course
- The Smith Predictor and esp. the MBC will outperform even a perfectly tuned PID for a deadtime dominant process
- Additional significant improvement in the disturbance rejection can be achieved by a feedforward, also here better in combination with an MBC than with the PID
- The original Smith Predictor has problems accommodating a feedforward, thus when a FFW is needed, MBC is the only alternative to the PID
- Both the Smith and modern MBC’s are much less sensitive to model errors than widely assumed
- Adaptive control is easier to accomplish with the Smith and MBC than with the PID
- Where a Smith Predictor or an MBC controller is already available in the DCS, the effort for setting up these controllers is about the same as for the PID.