

A PRACTICAL PREDICTIVE CONTROL ALGORITHM FOR A LARGE CLASS OF PROCESSES

Vasile Cîrtoaje, Sanda Frâncu, Gabriel Rădulescu

Department of Automation & Computers

Oil & Gas University of Ploiesti, Romania

E-mail: vasc@csd.univ.ploiesti.ro

Abstract. In the last few years, the model based process control (MBPC) methodology was mainly situated at the upper control level (optimization level, product level or economical level), which requires more advanced control techniques. Many laboratories and industrial applications have demonstrated that MBPC has a well performance/cost ratio and it is not difficult to be implemented by means of digital equipment. Moreover, even non-specialists in control field easily understand the basic principles of MBPC. As compared to the classical PID strategy, the MBPC strategy uses an explicit process model to predict the controlled process output. However, due to the computational complexity and the requirement to own a suitable process model, the PID type algorithms continue to remain the most widely used in lower level control loops.

In this paper, we will try to prove that it is possible to obtain well control performances using some standard unconstrained predictive algorithms (like those PID), characterized by fewer calculus and robustness relating to the process model accuracy. In our control strategy, we propose for process model a second order continuous-time transfer function with dead time and having the time constants allocation coefficient in the range 0.6 - 1.0. At first, the user must specify three process parameters: the proportional constant, the main time constant, and the dead time. During the operation, the user may modify, like in PID strategy, three tuning parameters: steady state gain adjustment parameter, time constant adjustment parameter, and dead time adjustment parameter. We show how each of these parameters modifies the system dynamic performances. The increasing of the steady state gain adjustment parameter leads usually to a stable system. In order to ensure the stability and good performances of the system, the prediction horizon should completely overlap the transient time and the number of the sampling periods on the prediction horizon should be limited at maximum 20.

Key words: MBPC, predictive control, robust control, standard control algorithm, real time execution, sampling time.

1. Introduction

Model Based Predictive Control represents a *receding horizon* control strategy based on the explicit on-line use of a *suitable process model* to predict the effect of potential control action upon the plant future state over a long-range time horizon [1]. At each sampling instant, the updated plant information is used to solve an *open-loop optimal control problem*, but only the

first element of the optimal control vector is actually applied to the real process. All other elements of the optimal control vector can be either not calculated or forgotten because at the next sampling instant all calculus-sequences are performed again based on the new output measurement (“receding horizon” principle). The control vector or only its first element are calculated in order to

minimize a specified cost function [8], depending on the future *postulated* values of the control variable and the predicted control errors.

The availability of a suitable process model and the necessity to optimize the control objective *in real time* (at each sampling instant) are the most stringent requirements for the use of MBPC policy. Due to the mathematical convenience, in this paper we use the optimization quadratic criterion, which presents the advantages of an intuitive and reliable analytical solution. Instead of an accurate but less robust optimal algorithm, we will use an algorithm based on analytical solution [2], which allows the quick computation of suboptimal control actions.

2. The proposed MBPC algorithm

The proposed algorithm consists of two parts: one off-line computation part and other on-line computation part.

Off-line computation

In the proposed control strategy, we consider the process dynamic model as a second order continuous-time transfer function with dead time

$$H(s) = \frac{Ke^{-\tau s}}{(T_1s+1)(T_2s+1)}. \quad (1)$$

This off-line computation part consists in finding the following elements: the process model coefficients, the sampling period, the prediction horizon, and the unit step response of the process. The *initial data* are the following: K – process steady state gain (proportionality constant), T_0 – process dominant time constant, τ – process dead time, ξ - time constants allocation coefficient, T – sampling period. The time constant T_0 can be approximated by the process time

constants sum or by the ratio $(T_r - \tau)/3$, where T_r is the process transient time (for step response and ± 5 percent limits).

The sampling time is chosen so that the integer value of the ratio $(\tau + 3T_0)/T$ should be in the range 15 ... 20. This integer represents the *prediction horizon* and it is denoted by N .

The time constants of the process model are given by the relations:

$$T_1 = (1 - 0.5\xi)X_t T_0, \quad T_2 = 0.5\xi X_t T_0. \quad (2)$$

In these relations, X_t is *the adjustment coefficient of T_0* (default value equal to 1).

We define the integer dead time m as being equal to the integer value of the ratio $[\frac{X_d \tau}{T}]$, where X_d is *the adjustment coefficient of τ* (default value equal to 1). When the user modifies τ , he must observe if the integer m is also modified.

Introducing the coefficients

$$\begin{aligned} c_1 &= e^{-T/T_1}, \quad c_2 = e^{-T/T_2}, \quad a_1 = -(c_1 + c_2), \\ a_2 &= c_1 c_2, \quad b = X_k K (1 + a_1 + a_2), \end{aligned} \quad (3)$$

the process discrete model has the form

$$y_t + a_1 y_{t-1} + a_2 y_{t-2} = b u_{t-m}. \quad (4)$$

In (3), the parameter X_k is *the adjustment coefficient of K* and, in order to ensure the system stability, it must be greater than 1 (default value equal to 1.5).

The unit step response g of the process can be computed by the recursive equations:

$$\begin{cases} h_0 = b, \quad h_1 = b - a_1 h_0 \\ h_{i+2} = b - a_1 h_{i+1} + a_2 h_i, \quad i \geq 0 \\ g_i = 0, \quad i = \overline{1, m-1} \\ g_{i+m} = h_i, \quad i \geq 0 \end{cases} \quad (5)$$

The last off-line computation is the expression

$$E = g_m^2 + g_{m+1}^2 + \dots + g_N^2. \quad (6)$$

When the operator changes one of these three adjustment parameters, again the off-line computation set must be done.

On-line computation

One linear form of the predictive control law is the following

$$u_t = u_{t-1} + \frac{X_p}{E} \sum_{i=m}^N g_i e_{t+i} \quad (7)$$

where t is the current discrete time, X_p - the control weighting coefficient, E - the expression (6), g - the process unit step response, and e - the predicted error

$$e_{t+i} = w_{t+i} - y_{t+i} \quad (8)$$

In (8), w denotes the setpoint and y denotes the predicted free response of the process (fig. 1). The free response $y(t)$ represents the effect of the past control [8], because

the input remains constant for the future time, i.e. $U_t = U_{t-1}, U_{t+1} = U_{t-1}, \dots$

The prediction can be achieved by means of the process model (4), using the previous measured values Y_t, Y_{t-1} and Y_{t-2} , as well as the previous values of the input $U_{t-1}, U_{t-2}, \dots, U_{t-m}$.

The free response can be calculated by the recursive equations:

$$\begin{aligned} y_{t-2} &= Y_{t-2}, \quad y_{t-1} = Y_{t-1}, \quad y_t = Y_t \\ y_{t+k} &= y_{t+k-1} - a_1(y_{t+k-1} - y_{t+k-2}) - \\ &\quad - a_2(y_{t+k-2} - y_{t+k-3}) + bz_{t+k}, \quad k \geq 1 \end{aligned} \quad (9)$$

where

$$z_{t+k} = \begin{cases} u_{t+k-m} - u_{t+k-m-1}, & k < m \\ 0, & k \geq m \end{cases} \quad (10)$$

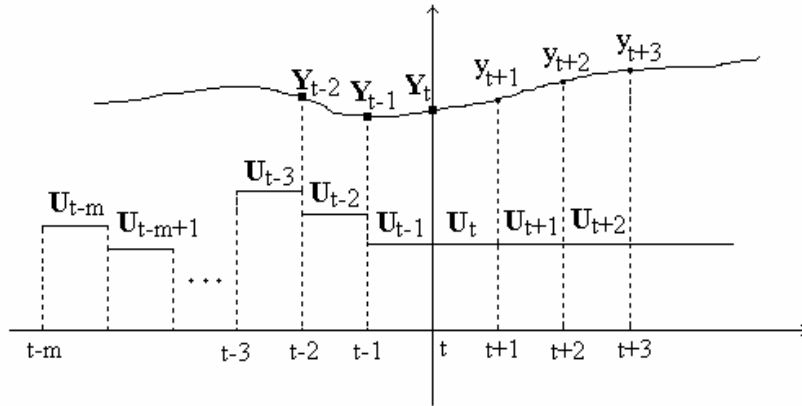


Figure 1. The predicted free response $y_{t+i} = f(Y_{t-2}, Y_{t-1}, Y_t; U_{t-1}, U_{t-2}, \dots, U_{t-m})$

The weighting coefficient X_p must be less than 1 (its default value is 0.6) and it is used to filter the control variable u .

The control law (7) can be obtained by minimizing the quadratic objective function

$$F = \sum_{i=m}^N (w_{t+i} - \bar{y}_{t+i})^2, \quad (11)$$

where \bar{y} represents the predicted response of the system to the input u having a single postulated value \bar{U} over all prediction horizon N (fig. 2). If there are two postulated values for u , one \bar{U}_1 for the first interval $[t, t+1]$ and the other one \bar{U}_2 for the interval $[t+1, t+N]$, then the predictive control law has the following form:

$$u_t = u_{t-1} + \frac{X_p}{E_1} \sum_{i=m}^N (g_i - a g_{i-1}) e_{t+i} \quad (12)$$

where

$$E_1 = E - aF_1, \quad E = \sum_{i=m}^N g_i^2, \quad (13)$$

$$F_1 = \sum_{i=m}^{N-1} g_i g_{i+1}, \quad a = \frac{F}{E - g_N^2}. \quad (14)$$

Furthermore, considering that u has the value \bar{U}_1' for the interval $[t, t+2]$ and the value \bar{U}_2' for the remaining interval

$[t+2, t+N]$, then the predictive control law becomes

$$u_t = u_{t-1} + \frac{X_p}{E_2} \sum_{i=m}^N (g_i - a g_{i-2}) e_{t+i} \quad (15)$$

where

$$E_2 = E - aF_2, \quad E = \sum_{i=m}^N g_i^2, \quad (16)$$

$$F_2 = \sum_{i=m}^{N-2} g_i g_{i+2}, \quad a = \frac{F_2}{E - g_{N-1}^2 - g_N^2}. \quad (17)$$

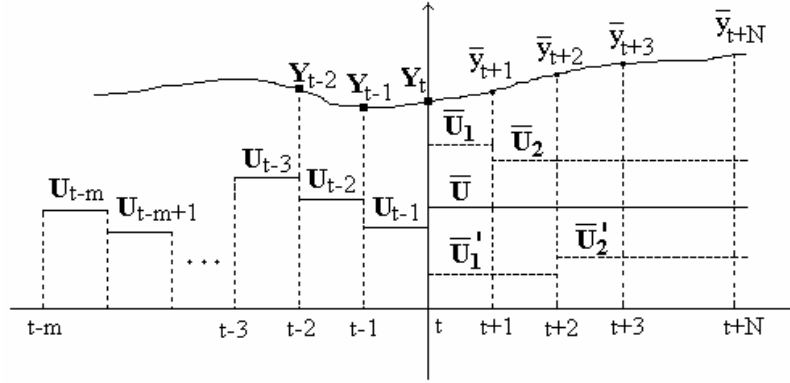


Figure 2. The predicted response \bar{y} to the input having one or two future values

3. The simulation results

The previous predictive algorithms have been implemented in a real time multitasking application. We have considered that the process has the model

$$H(s) = \frac{1.2(1 - T_4 s) e^{-90s}}{(90s+1)(60s+1)(50s+1)}.$$

We have chosen the *starting data*:

$$K=1.2; T_0=200 \text{ s}; \tau=90 \text{ s}; \xi=0.6; T=40 \text{ s}$$

and

$$X_k=1.5; X_t=1.0; X_d=1.0; X_p=0.6.$$

From these data, we obtain

$$m = \left\lceil \frac{X_d \tau}{T} \right\rceil = 2, \quad N = \left\lceil \frac{\tau + 3T_0}{T} \right\rceil = 17.$$

In figures 3...10, the simulation results for the minimum phase process ($T_4=0$) and the control law (7) are shown.

Because the system model does not describe exactly the real system dynamics, the controlled system can be unstable for small values of X_k . Increasing X_k , the system becomes usually stable, but slower (fig. 3 and fig. 4). The introduction of the reference trajectory has not an important impact on the loop dynamics (fig. 5). The system may become unstable for both large and small values of the X_d and X_t respectively (figures 6, 7, 8). The change of the time constants allocation coefficient ξ does not reveal major effects upon the system response form (fig. 9). However, it

can be observed that the best response is obtained for maximum value of ξ . In fig. 10 is presented the effect of the control weighting constant X_p upon the system response. For a small X_p , the response is too slow; it becomes faster as X_p increases to 1. In consequence, for many processes X_p can be fixed in advance at a value in the range 0.5 - 0.8.

In the figure 11 is shown that for $T_4=60$ s (non-minimum phase process), the predictive control system may be stabilized using a large value of X_k (fig. 11).

4. Conclusions

In this paper, we have proved that it is possible to obtain well control performances using some standard unconstrained predictive algorithms (like those PID), characterized by fewer calculus and robustness relating to the process model accuracy.

We have shown that a *second order transfer function with dead time* and having the time constants allocation coefficient ξ in the range 0.6 – 1.0 can be successfully used to describe the process dynamics.

The user must firstly specify three *process parameters*: K, T_0, τ .

In order to get the right performances, the user may modify during the operation time only three *tuning parameters*: X_k, X_t, X_d (which are initialized at the values 1.5, 1.0, 1.0 respectively). Usually, the control weighting constant X_p (initialized at a value in the range 0.5-0.8) is not necessary to be adjusted during the operation time. For both large and small values of the tuning parameters X_d and X_t respectively, the system may become unstable. But the system may be usually stabilized by increasing the steady state gain adjustment parameter X_k .

In order to ensure good performances it is necessary that *the prediction horizon should completely overlap the transient time and the number of the sampling periods on the prediction horizon should be chosen in the range 15 - 20*.

References

- [1] De Keyser R., *A Model Based Predictive Control Tutorial*, In Proceedings of the 12th International Conference CSCS, Vol.1, pp. 1-12, Bucharest, 1999.
- [2] Boucher P., Dumur D., *La commande predictive*, Editions Technip, Paris, 1996.
- [3] Marlin T., *Process Control – Designing Processes and Control Systems for Dynamic Performance*, McGraw-Hill, Inc., pp. 604-630, 1995.
- [4] Lazar C., Arabolu C., *Model Predictive Control with Amplitude Constraints*, In Proceedings of the 9th SIMSIS '96, pp. 77-82, Galati, 1999.
- [5] Richalet J., *Industrial Applications of Model Based Predictive Control*, Automatica, Vol. 29, No. 5, pp. 1251-1274, 1993.
- [6] Tsang T., Clarke D., *Generalised Predictive Control with Input Constraint*, IEE Proceedings, Vol. 135, No. 6, pp. 451-460, 1988.
- [7] Chwn H., Allgower F., *Nonlinear Model Predictive Control Schemes with Guaranteed Stability*, NATO Advanced Study Institute Proceedings, pp. 465-495, Antalya, Turkey, 1997.
- [8] Camacho E, Bordons C., *Model Predictive Control*, Advanced Textbooks in Control and Signal Processing, Springer-Verlag, London Limited, 1999.