ADAPTIVE CONTROL THROUGH ON-LINE SIMULATION THEORETICAL ASPECTS

Radu Bălan

Technical University Cluj-Napoca, Fac. Mecanică/ Mechatronics dep.
B-dul Muncii 103-105 Cluj-Napoca, tel 064-155967 e-mail: radubalan@yahoo.com

ABSTRACT

An algorithm, which uses on-line simulation and rule-based control, is proposed. The algorithm is a Model Based Predictive Control (MBPC) type. The basic idea is the on-line simulation of the future behaviour of control system, by using a few control sequences. Then the simulations are used to obtain the 'optimal' control signal.

KEYWORDS: adaptive-predictive control, on-line simulation, rule-based control.

1. INTRODUCTION.

Model Based Predictive Control (MBPC) designates a very ample range of control methods, which make an explicit use of a model of the process to obtain the control signal by minimizing an objective function. The ideas appearing in greater or lesser degree in all the predictive control family are:

-explicit use of a model to predict the process output in the future;

-on line optimization of a cost objective function over a future horizon;

-receding strategy, so that at each instant, the horizon is displaced towards the future, which involves the application of the first control signal of the sequence calculated at each step.

Performance of MBPC could become unacceptable due to a very inaccurate model, thus requiring a more accurate model. This task is an instance of closed-loop identification and adaptive control. The difficulty of closed-loop identification is that the input of process to be identified is not directly selected by the designer but ultimately by the feedback controller. Usually, the cost function is defined by using the output prediction error relative to the system setpoint and the weighted control signal, which can lead to a quadratic function as follows:

$$J(N_1, N_2) = \sum_{j=N_1}^{N_2} [y(t+j) - y_r(t+j)]^2 + \sum_{j=1}^{N_u} \rho(j)[\Delta u(t+j-1)]^2 \qquad (1)$$

where y[.] is the predicted values of output signal, y_r[.] is the future set-point, u[.] is the future control signal, N_1, N_2 are the minimum and maximum predicted horizon, N_u is the command horizon and ρ(j) is a control-weighting sequence.

In order to obtain the future values of command, it is necessary to minimize function J from relation (1). To do this, using the model of the process, there are calculated the values of predicted outputs as a function of past values of inputs and outputs and of future control signals. Then, using the minimization of the cost function, are obtained the optimal values of control signals. An analytical solution can be obtained for the quadratic criterion if the model is linear and there are not constraints, otherwise an iterative method of optimization should be used. But obtaining the solution is not easy because are more independent variables. In order to reduce this degree of freedom, a certain structure may be imposed on the control law. There are some 'classical' methods; for example see [1].

The main goal of this paper is to show another way to obtain the value of u(t). Let's consider that it is possibly to compute (for every sample period):

- the predictions of system output over a finite horizon (N)
- the cost of the objective function (1)

for each possible sequence (for simplicity, here N_u=N):

$$u(.)=\{u(t),u(t+1), \ldots ,u(t+N)\} \quad (2)$$

and then to choose the first element of the optimal control sequence. For a first look, the advantages of the proposed algorithm include the following:

-the minimum of objective function is global;
-it is not necessary to invert a matrix, so potential difficulties are avoided;
-it can be applied to nonlinear processes if a nonlinear model is available;
-the constraints (linear or nonlinear) can easily be implemented.

The drawback of this scheme is a very long computational time, because there are possibly a lot of sequences. If u(t) is applied to the process using a "p" bits numerical-analog converter (DAC), the number of sequences is 2^{p*N} [2]. Therefore, the number of sequences must be reduced. In figure 1, is presented a scheme of control that used "j" sequences.

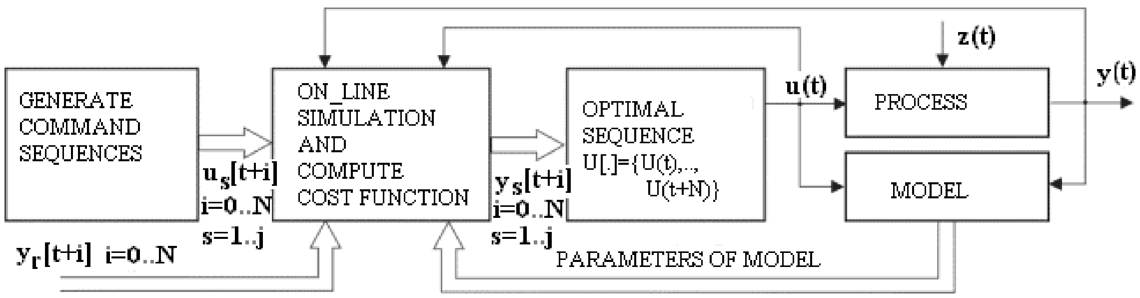

Fig.1. Adaptive control using on-line simulation.

2. CONTROL ALGORITHM

In this paper, will be tackled *only* the case when setpoint is constant for a long time. The setpoint is changed arbitrary. Using the model of process, it is possibly to compute the predicted output y(t+i) i=1..N. The expression of y(t+i) contains two terms: one (g(.)), which depends on old values of command and output signals g(.) and one, which depends on future values of command signal.

Therefore:

$$y(t+i)=g(t+i) \quad \text{for } i=1..d \quad (3)$$

$$y(t+i)=g(t+i)+f_{i-d-1}(i)u(t+i-d-1)+f_{i-d-2}(i)u(t+i-d-2)+\ldots f_0(i)u(t) \quad \text{for } i > d \quad (4)$$

where f_j, j=0..N-k-1 can be computed iteratively using the model of process.

In real case, it is necessary to consider that:

$$u_{min} \leq u(t) \leq u_{max} \quad (5)$$

where u_{min} and u_{max} are the minimum and the maximum of command signal.

Let's consider the next four command sequences and suitable output predicted sequences:

Case 1: $u_1(t) = \{u_{min}, \mathbf{u_{min},.., u_{min}}\}$ (6)

$y_1(t+i) = g_i + s_i u_{min} + f_0 u_{min}$ (7)

where $s_i = f_{i-d-1} + f_{i-d-2} + .. + f_1$ (8)

Case 2: $u_2(t) = \{u_{max}, \mathbf{u_{min},.., u_{min}}\}$ (9)

$y_2(t+i) = g_i + s_i u_{min} + f_0 u_{max}$ (10)

Case 3: $u_3(t) = \{u_{min}, \mathbf{u_{max},.., u_{max}}\}$ (11)

$y_3(t+i) = g_i + s_i u_{max} + f_0 u_{min}$ (12)

Case 4: $u_4(t) = \{u_{max}, \mathbf{u_{max},.., u_{max}}\}$ (13)

$y_4(t+i) = g_i + s_i u_{max} + f_0 u_{max}$ (14)

The sequences (6), (9), (11), (13) were selected after a lot of experiments on different processes. Using these four pair sequences, it is possibly to compute:

Case 1: $y_{max0} = \max_i \{y_1(t+i)\}_{i=d..N}$ (15)

Case 2: $y_{max1} = \max_i \{y_2(t+i)\}_{i=d..N}$ (16)

Case 3: $y_{min0} = \min_i \{y_3(t+i)\}_{i=d..N}$ (17)

Case 4: $y_{min1} = \min_i \{y_4(t+i)\}_{i=k..N}$ (18)

In figures 2,3 there are presented the predictions of output in some usual cases. Here, it was used a thermal process with dead time, but next assumptions are generally. The difference between $\{u_1(t)\}$ and $\{u_2(t)\}$ or $\{u_3(t)\}$ and $\{u_4(t)\}$, can lead to a considerable difference between output predictions. Also, the predictions depend on gain factor, time constants and dead time.

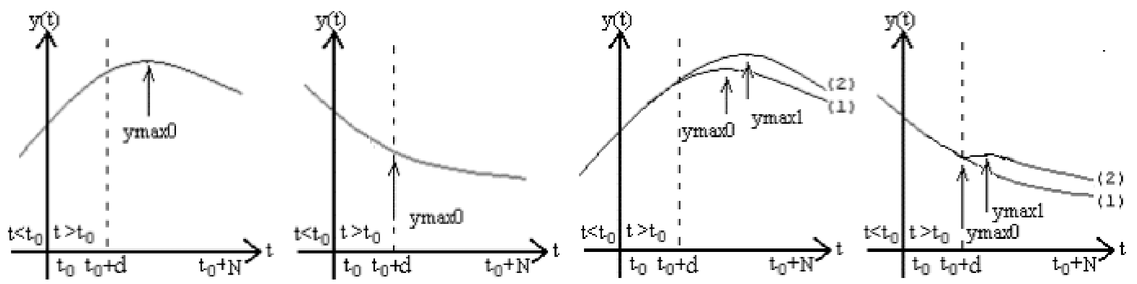

Figure 2: Cases for output predictions using $\{u_1(t), \{u_2(t)\}$

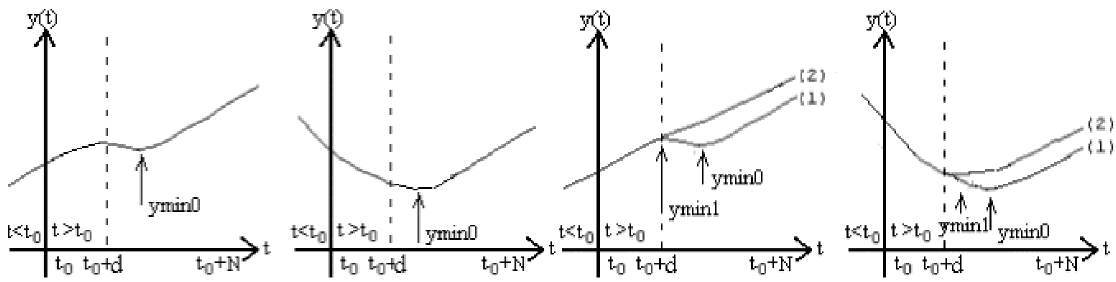

Figure 3: Cases for output predictions using $\{u_3(t), \{u_4(t)\}$

These four pair sequences $\{u_s(t+i), y_s(t+i);\ s=1..4, i=d..N\}$ and the extremes (15)..(18) can be used to compute the 'optimal' command $u(t_0)$. For a first stage, the quadratic cost function (1), will be replaced with minimizing of the future errors between output and setpoint. The control increments are not penalized.

Let's consider four usual situations:

Case 1: If: $y_{max0} < y_r$ (corresponding to $u_1(t)$ sequence)

$y_{max1} > y_r$ (corresponding to $u_2(t)$ sequence) (19)

Then 'optimal' command $u(t_0)$ is (using a linear approximation - figure 4):

$$u(t_0)=\frac{u_{max}-u_{min}}{y_{max1}-y_{max0}}y_r+\frac{u_{min}y_{max1}-u_{max}y_{max0}}{y_{max1}-y_{max0}} \tag{20}$$

Case 2:If: $y_{min0}<y_r$ (corresponding to $u_3(t)$ sequence)
$y_{min1}>y_r$ (corresponding to $u_4(t)$ sequence) (21)

Then 'optimal' command $u(t_0)$ is (using a linear approximation – figure 4):

$$u(t_0)=\frac{u_{max}-u_{min}}{y_{min1}-y_{min0}}y_r+\frac{u_{min}y_{min1}-u_{max}y_{min0}}{y_{min1}-y_{min0}} \tag{22}$$

Case 3:If: $y_{max0}>y_r$ (corresponding to $u_1(t)$ sequence)
Then $u(t_0)=u_{min}$ (figure 4) (23)

Case 4:If: $y_{max1}<y_r$ (corresponding to $u_2(t)$ sequence)
Then $u(t_0)=u_{max}$ (figure 4) (24)

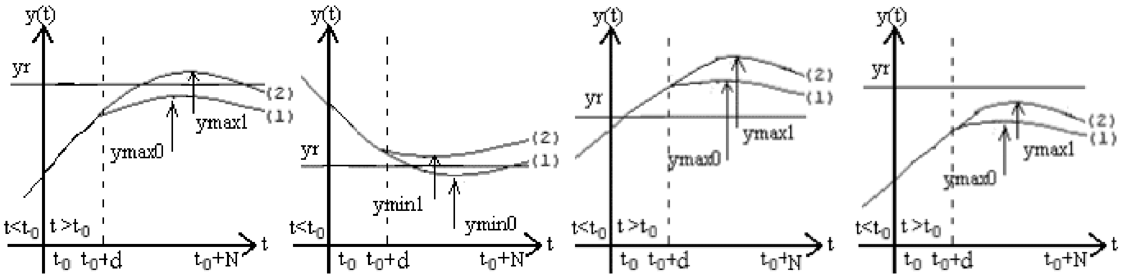

Figure 4: Cases for $u(t_0)$ computing

Similar to cases 3, 4 there are two cases when $dy/dt<0$ for $t<t_0$.

3. COMMENTS

The manner of control sequences choosing is very simply and intuitive. A main advantage is the fact that the extremes u_{max}, u_{min} are used directly in control algorithm. It is possibly to choose another type or/and number of sequences. The main question is: how can be used the extremes (15)..(18), or, more generally: how can be used the predictions of output to obtain the 'optimal' command u(t)? The problem becomes very complex if consider the constraints on input and/or output, variable setpoint, perturbations, etc.

The command sequences (6), (9), (11), (13) respect two kind of demands:

1. On line simulations of future behaviour of system using sequences that contain only one of extremes of control signal u_{max} or u_{min} ($u_1(t)$, $u_4(t)$). These on line simulations are useful in transitory regime.

2. On line simulations using sequences which differ only for u(t) - the pairs of sequences $u_1(t)$, $u_2(t)$ respectively $u_3(t)$,$u_4(t)$. These on line simulations are useful especially in stationary regime and they show the effects of u(t) choosing.

The algorithm was tested (in simulations and practice) using different processes. These simulations and experiments led to next remarks:

1. Using only rules (19)..(24), the algorithm leads to a large variance of control signal, usual between u_{max} and u_{min}, especially in stationary regime. It is possibly a stabilization of control signal; this thing depends on setpoint, time constants, dead time etc. But it is difficult to maintain control signal stabilization, especially for important noise or/and identification errors. If it is necessary, the algorithm must be modified to obtain control signal stabilization. However, if it is used an on/off actuator (a usual case

in practice), the stabilization of control signal is not necessary. For example, a thermal process with electric heat; in this case, the control is better, comparatively with an on/off control.

2. After an important change of setpoint or after a considerable perturbation, the algorithm builds one of next sequences: $\{u_{max}, u_{max},..u_{max}\}$ or $\{u_{min}, u_{min},..u_{min}\}$. When the stationary regime begins, even for a accurate model, after a first small override, it is possibly to obtain a larger override. This phenomenon will be repeated but with dumping; it is amplified at the beginning of parameters identification or if the process parameters are modified. This phenomenon appears because in relations (15)..(18) the values of extremes $(y_{max0}, y_{max1}, y_{min0}, y_{min1}\}$ are resulted due to old (in time) control signals and the actual control signal is not computed correctly.

So, the algorithm must be modified in a view to minimize the two effects.

4. THE MODIFIED ALGORITHM (A1)

The behaviour of system is considerable influenced by the selected desired trajectory. For example, if the setpoint is changed considerable, it is possibly that algorithm to build a sequence $\{u_{max}, u_{max}, ... u_{max},\}$ or a sequence $\{u_{min}, u_{min}, ... u_{min},\}$ for a few sample periods. It is possibly to obtain a better behaviour if the desired trajectory is modified:

$$y_{r1}(t)=y_r(t)+k_{ref}[y(t)-y_r(t)] \qquad (25)$$

where $y_{r1}(t)$ is the desired trajectory and k_{ref} is a control parameter. A cautious value used in many applications is k_{ref}=0.3..0.5. If difference between process and model is significantly, k_{ref} must rise (k_{ref}>0.5); this parameter can be modified using a function of identification quality.

For control signal stabilisation there are used next methods:

1. An algorithm that modifies the limits (u_{minst},u_{maxst}) of control signal.

It is defined average of error using equate:

$$a_{med}(t) = k_{amed}a_{med}(t-1)+(1-k_{amed})|y_r(t)-y(t)| \qquad (26)$$

where k_{amed} is a weight factor.

It is defined C_{sta} -a counter of stationary regime and C_{tr} a counter of transitory regime. In transitory regime, $C_{sta} = 0$; in stationary regime, at every sample period, C_{sta} is incremented. The stationary regime begins after a sequence $\{u_{maxst,}u_{maxst,..,}u_{maxst}\}$or $\{u_{minst,}u_{minst,..,}u_{minst}\}$ when the algorithm build $u(t) \neq u(t-1)$.

There are defined $u_{maxst}(t)$ and $u_{minst}(t)$ the limits of control signal that are accepted at sample period 't'. The stationary regime is divided by using of two parameters $0< V_{sta1} < V_{sta2}$:

- for $0 < C_{sta} \leq V_{sta1}$ is 'initial part of stationary regime' and:

$$a_{med}(t)=0; \; u_{maxst}(t)=u_{max} \; ; \; u_{minst}(t)=u_{min} \; ; \qquad (27)$$

- for $V_{sta1} < C_{sta} \leq V_{sta2}$ is 'stabilization part of stationary regime'
- for $V_{sta2} < C_{sta}$ is 'final part of stationary regime'

If $V_{sta1} < C_{sta}$ then:

$$a_{med}(t)=K_{amed}a_{med}(t-1)+(1-K_{amed})|y_r(t)-y(t)| \qquad (28)$$

$$u_{maxst}(t)= u_{maxst}(t-1)-K_{st}[u_{maxst}(t-1)-u_{st}(t)]+K_a a_{med}(t) \qquad (29)$$

$$u_{minst}(t)= u_{minst}(t-1)+K_{st}[u_{st}(t)- u_{maxst}(t-1)]-K_a a_{med}(t) \qquad (30)$$

where K_{st} is a weight factor which controls the decrease of difference $u_{maxst}(t)- u_{minst}(t)$ and $u_{st}(t)$ is the prediction of control signal in stationary regime:

$$u_{st}(t) = \frac{1+a_1+a_2+..+a_n}{b_1+b_2+..+b_m} y_r \qquad (31)$$

The difference $u_{maxst}(t)$- $u_{minst}(t)$ must be greater than a minimal value d_{ust}. In expressions (6)..(14), (19)..(24) the value of $u_{maxst}(t)$, $u_{minst}(t)$ are used instead of u_{max}, u_{min}. The control parameters V_{sta1}, V_{sta2}, k_{amed}, k_{st}, k_a, d_{ust} can be choose in large limits. In expression (31), the value of u_{st} can be changed with the average of signal control:

$$u_{med}(t)=k_{umed}u_{med}(t)+(1-k_{umed})u(t) \quad (32)$$

where k_{umed} is a weight factor.

K_a is a parameter that controls the increase of difference $u_{maxst}(t)$- $u_{minst}(t)$ if $a_{med}(t)$ increases. The effect of this parameter is important if there are significant differences between process and model.

The parameter V_{sta1} controls the number of sample periods when the algorithm builds $u_{maxst}(t)=u_{max}$ and $u_{minst}(t)=u_{min}$. This larger variance of control signal has a positive effect for identification algorithm.

2. In stationary regime, for $V_{sta2}<C_{sta}$, it is used the average of control signal:

$$u(t)= k_u u^*(t)+(1-k_u)u_{med} \quad (33)$$

where u*(t) was choose using (19)..(24) and k_u is a weight factor. Another possibility (especially for an accurate model) is:

$$u(t)= k_u u^*(t)+(1-k_u)u_{st} \quad (34)$$

If $C_{sta}>V_{sta2}$ and the error |y(t)-yr(t)| is greater then a accepted value Δp, this is the case of perturbation regime, or a very inaccurate model; the solution is to make $C_{sta}=0$ (or anther value lesser than V_{sta1}). This permits to algorithm to build a control signal with large variance (is useful for identification algorithm).

5. CONCLUSIONS

This paper presents some principles of an adaptive-predictive algorithm that uses on-line simulation and rule based control. The on-line simulations for a few command sequences, allow the obtaining of the “optimal” control signal u(t). The parameters of algorithm can be optimized using a supervisor algorithm.

5. REFERENCES

[1] Camacho E., Bordons C. [1999], “Model Predictive Control” Spriger-Verlag

[2] Balan R. [2001] “Adaptive control systems applied to technological processes”, Ph.D. Thesis 2001, Technical University of Cluj-Napoca, Romania.

[3] Control Engineering Europe February/March [2001] pp. 25..28

[4] Balan R. [2000], “Adaptive Control for Thermal Processes described by Strejc Equation”, Q&A-R 2000 THETA 12

[5] Balan R. [2000], “A Method for Heat-Exchanger Simulation and Control” Q&A-R 2000 THETA 12

[6] Nikolau M., S.Alper Eker [1999], “Adaptive control through on-line optimization. The MPCI paradigm and variants. Int. J. Appl. Math and Comp. Sci, 1999 Vol 9, No. 1, pp 25-52