A model-based adaptive-predictive algorithm applied in tracking control

Radu Bălan
Department of Mecatronics
Technical University of Cluj-Napoca
B-dul Muncii 103-105 Cluj-Napoca
Romania
radubalan@yahoo.com

Vistrian Măţieş
Department of Mecatronics
Technical University of Cluj-Napoca
B-dul Muncii 103-105 Cluj-Napoca
Romania
matiesvistrian@yahoo.com

Olimpiu Hancu
Department of Mecatronics
Technical University of Cluj-Napoca
B-dul Muncii 103-105 Cluj-Napoca
Romania
ohancu@yahoo.com

Abstract – **Model Based Predictive Control (MBPC) is a class of computer algorithms that explicitly use a process model to predict future plant outputs and compute an appropriate control action through on-line optimization of a cost objective over a future horizon, subject to various constraints. The cost function is defined in terms of the tracking error (the difference between the predicted output and set-point). Using this scheme, many different MBPC algorithms have been proposed in the literature. This paper presents an adaptive-predictive control algorithm designed for tracking case, which uses on-line simulation and rule based control. Simulated examples are given in two cases of mechanical processes.**

I. INTRODUCTION

Model Based Predictive Control (MBPC) designates a very ample range of control methods that make an explicit use of a model of the process to obtain the control signal by minimizing an objective function. The main ideas of MBPC are basically:

-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. A solution to increase the performances is to use multiple models [2].

Usually, the cost function is defined by using the output prediction error relative to the system setpoint and the weighted control signal:

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

where:

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

In [1] it was proposed an algorithm (MBAPC-A1) designed for applications with constant set-point (but arbitrary changed). The main idea of the algorithm is to compute for every sample period:

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

for all (theoretically case) or a few (practically case) possible control sequences:

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

and than 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, therefore potential difficulties are avoided;

-this algorithm 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 possible a lot of sequences. For example, if $u(t)$ is applied to the process using a "p" bits numerical-analog converter (DAC), the number of sequences is 2^{p*N}. Therefore, the number of sequences must be reduced.

For a first stage, there were proposed the next four control sequences:

$$\begin{aligned}u_1(t)&=\{u_{\min},u_{\min},..,u_{\min}\}\\ u_2(t)&=\{u_{\max},u_{\min},..,u_{\min}\}\\ u_3(t)&=\{u_{\min},u_{\max},..,u_{\max}\}\\ u_4(t)&=\{u_{\max},u_{\max},..,u_{\max}\}\end{aligned} \quad (3)$$

where u_{min} and u_{max} are the limits of the control signal.

Using these candidate control sequences result four output sequences $y_1(t)$, $y_2(t)$, $y_3(t)$, $y_4(t)$. The control signal is computed using a set of rules based on the extremes y_{max0}, y_{max1}, y_{min0}, y_{min1} (fig. 1) of the output predictions. It is considered four usual cases (y_r is the set point, d is dead time, $t_1=N$):

Case 1: If:

$y_{max0}<y_r$ (corresponding to $u_1(t)$ sequence)
$y_{max1}>y_r$ (corresponding to $u_2(t)$ sequence) (4)

Then:

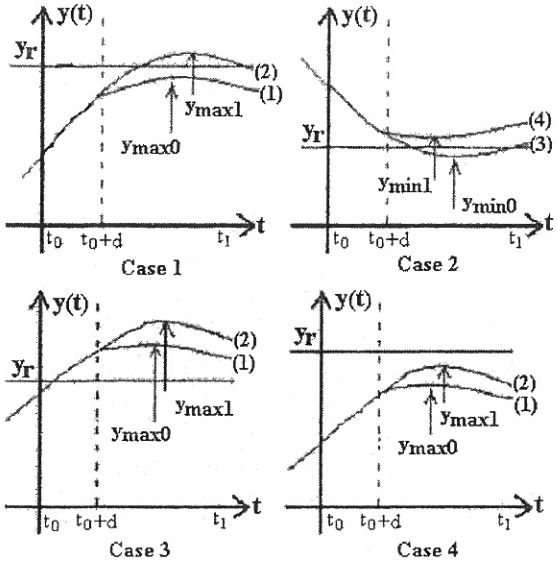
$$u(t)=\frac{u_{\max}-u_{\min}}{y_{\max 1}-y_{\max 0}}y_r+\frac{u_{\min}y_{\max 1}-u_{\max}y_{\max 0}}{y_{\max 1}-y_{\max 0}} \quad (5)$$

Case 2: If:

$y_{min0}<y_r$ (corresponding to $u_3(t)$ sequence)
$y_{min1}>y_r$ (corresponding to $u_4(t)$ sequence) (6)

Then:

$$u(t)=\frac{u_{\max}-u_{\min}}{y_{\min 1}-y_{\min 0}}y_r+\frac{u_{\min}y_{\min 1}-u_{\max}y_{\min 0}}{y_{\min 1}-y_{\min 0}} \quad (7)$$


y(t)
yr
(2)
(1)
ymax1
ymax0
t
t0
t0+d
t1
Case 1
y(t)
yr
(4)
(3)
ymin1
ymin0
t
t0
t0+d
t1
Case 2
y(t)
yr
(2)
(1)
ymax1
ymax0
t
t0
t0+d
t1
Case 3
y(t)
yr
(2)
(1)
ymax1
ymax0
t
t0
t0+d
t1
Case 4

```
ERROR: ioerror
OFFENDING COMMAND: flushfile

STACK:

-filestream-
-filestream-
-filestream-
-filestream-
-savelevel-
```