SOME APPLICATIONS FOR NONLINEAR PROCESSES OF A MODEL BASED PREDICTIVE CONTROL ALGORITHM

Radu Balan[*], Sergiu Stan[**], Ciprian Lapusan[*]

[*]*Technical University of Cluj-Napoca, Department of Mechanisms, Precision Mechanics and Mechatronics*
*** Technical University of Cluj-Napoca, Department of Mechanics and Programming*

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 function over a future horizon, subject to various constraints. This paper presents an MBPC type algorithm applied to nonlinear processes. The basic idea of the algorithm is the on-line simulation of the future behavior of the control system, by using a few candidate control sequences. Then, using rule based control these simulations are used to obtain the 'optimal' control signal. The efficiency and applicability of the proposed algorithm for nonlinear processes are demonstrated through applications.*

Keywords: *non-linear process, model based predictive control*

1. INTRODUCTION

The analyses and design of control systems are most of time based on linear systems. There are two reasons for this approach. First of all, there are relatively simple closed analytical solutions to many control problems like including LQR and pole-placement controller design, Kalman-filtering, model parameter and structure estimation, etc. On the other hand, practical applications are also based on linear or linearized models in most cases and handle nonlinearities only when it is absolutely unavoidable [1].

A common approach of controlling process systems with strong nonlinear character is to apply model-based predictive controllers where a detailed dynamic process model is used in an optimization framework. The popularity of model-based predictive control is partially explained by the fact that it uses traditional dynamic process models which are usually available for design and/or simulation purposes. At the same time, model-based predictive control is being criticized by control engineers because of its lack or weakness of theoretical background, having no guarantee of convergence, stability, robustness, etc. in the general case [1], [2].

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

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

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

The purpose of the controller is typically to force the output to follow the reference signal. If reference is a constant, the problem is commonly referred to as set-point regulation. When the reference is time varying (but is known in advance), defining a control law to force the output to follow the reference signal is called the positioning control. The remainder of this paper is organized as follows. In sections II and III are reviewed the proposed algorithm in two cases: set-point regulation and positioning control. Five nonlinear plants are presented as case studies in section IV.

2. SET-POINT REGULATION ALGORITHM (MBPC-A1)

In [3] it was proposed an algorithm (MBPC-A1) designed for set-point regulation problem (but set-point can be 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;
- this algorithm can be applied to nonlinear processes;
- the constraints can easily be implemented.

The drawback of this scheme is a very long computational time, because there are possible a lot of sequences. Therefore, the number of sequences must be reduced. For a first stage, there were proposed [3] 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 sequences results 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 extreme values y_{max0}, y_{max1}, y_{min0}, y_{min1} (fig. 1- d is dead time, t_1=N) of the output predictions. In the followings, considering processes with positive sign, it can be put in evidence four usual cases:

Case 1: If $y_{max0}<y_r$ (corresponding to u_1(t) sequence) and $y_{max1}>y_r$ (corresponding to u_2(t) sequence) Then:

$$u(t) = \frac{u_{max} - u_{min}}{y_{max1} - y_{max0}} y_r + \frac{u_{min}y_{max1} - u_{max}y_{max0}}{y_{max1} - y_{max0}} \quad (4)$$

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

$$u(t) = \frac{u_{max} - u_{min}}{y_{min1} - y_{min0}} y_r + \frac{u_{min}y_{min1} - u_{max}y_{min0}}{y_{min1} - y_{min0}} \quad (5)$$

Case 3: If: $y_{max0}>y_r$ (corresponding to u_1(t) sequence)

Then $u(t_0)=u_{min}$ (6)

Case 4: If: $y_{max1}<y_r$ (corresponding to u_2(t) sequence)

Then $u(t_0)=u_{max}$ (7)

In fig. 1, every output prediction curve is marked with a number which corresponds to the number of control sequence from relations (3). Analogous to case 3 and case 4, there are two similarly cases if $dy/dt<0$ for $t<t_0$.

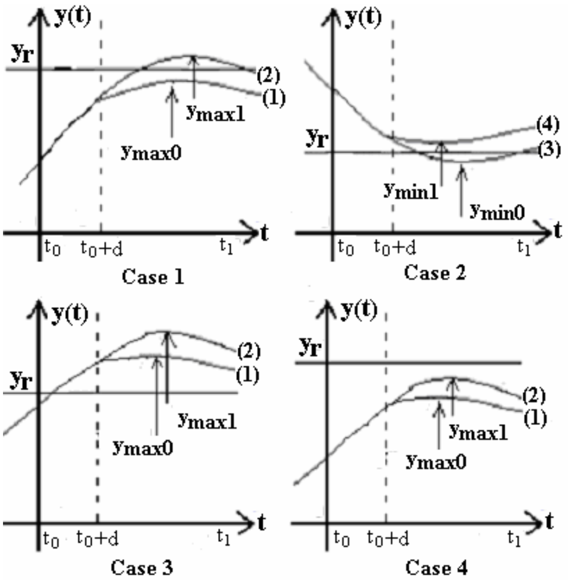

Fig. 1. Examples of output predictions

If the algorithm uses only these 6 rules, the variance of u(t) will be large [3]. So, for the second stage, depended by behaviour of the control system, are used next methods:

- an algorithm that modifies the limits of control signal:
-

$$u_{min} = u_{minst}(t) = u(t) = u_{maxst}(t) = u_{max}$$
$$? u_{min} = ? u = ? u_{max} \quad (8)$$

In relations (3).. (7), the values of u_{max}, u_{min} are replaced with $u_{minst}(t)$, $u_{minst}(t)$;

- using the “variable set-point“ [3]:
-

$$y_{r1}(t) = y_r(t) + k_{ref}[y(t) - y_r(t)] \quad (9)$$

where k_{ref} is a weight factor. The algorithm will try to reduce only a part of error;
- using a filter to compute control signal.

This algorithm was applied with good results, both for linear processes [3], and for nonlinear processes (heat exchangers, inverse pendulum on a cart) [4], [5].

3. POSITIONING CONTROL ALGORITHM (MBPC-A2)

For positioning control, it is used a specific algorithm. In this case, the rules (4)…(7) can not be applied directly. In fig. 2, are represented the evolutions of errors $e_i(t)_{i=1...4}$, versus sample time. Every output prediction curve is marked with a number which correspond to the number of control sequence from relations (3). Notations: t_0 is current time, N is the horizon of output, d is a parameter which is used for a fine-tuning (first, it is more simple to consider d=0).

It is used next five rules:
Case 1: The sequence u_3(t) leads to:

$$min_0 = \min_{t0+d<t<N}\{e_3(t)\}, \qquad min_0 > \delta \quad (10)$$

In this case $u(t)=u_{minst}(t)$.
Case 2: The sequence u_2(t) leads to:

$$max_1 = \max_{t0+d<t<N}\{e_2(t)\}, \qquad max_1 < -\delta \quad (11)$$

In this case $u(t)=u_{maxst}(t)$.
Case 3: The sequence u_4(t) leads to:

$$min_1 = \min_{t0+d<t<N}\{e_4(t)\}, \qquad min_1 < -\delta \quad (12)$$

and: $e_4(t_0+d+1)>0$. In this case $u(t)=u_{maxst}(t)$.
Case 4: The sequence u_1(t) leads to:

$$max_0 = \max_{t0+d<t<N}\{e_1(t)\}, \qquad max_0 > \delta \quad (13)$$

and: $e_1(t_0+d+1)<0$. In this case $u(t)=u_{minst}(t)$.

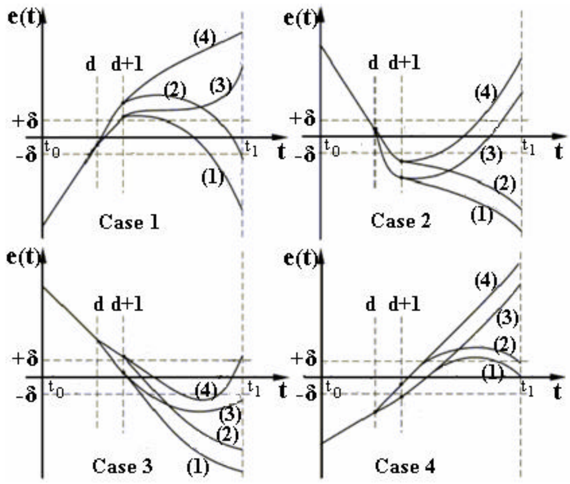

Fig. 2. Output predictions (Cases 1..4)

Case 5: In majority of the other situations, the predictions (for u_2 and u_3 sequences) are obtained like in fig.3. In this case it is used a linear relation:

$$u(t) = \frac{u_{minst}(t)\, max_1 - u_{maxst}(t)\, min_0}{max_1 - min_0} \quad (14)$$

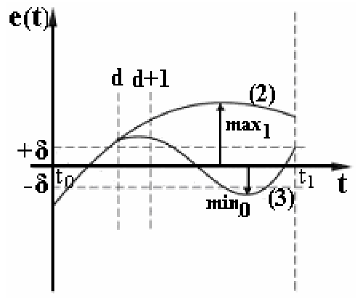

Fig. 3. Output predictions (Case 5)

If the algorithm uses only these 5 rules, the variance of $u(t)$ will be large. There are some solutions to reduce this variation. One of them is to use an algorithm that modifies the limits of control signal based on relations (8). As a result, the difference between u_{maxst} and u_{minst} decreases. On the other hand, in some cases, it is necessary to limit or to increases this difference. A good behaviour of the control algorithm leads to a prevalence of case 5.

4. APPLICATIONS

There are some well-known nonlinear control system design techniques: Lyapunov control design, input-output linearizing control design, input-state linearizing control design and integrator backstepping control design.

In [6], [9] are presented some examples where these methods failed and it's proposed a hybrid method as an alternative nonlinear control system design method.

The algorithms presented in previous sections can not be directly applied to nonlinear processes. For example, in the case of the inverted pendulum on a cart [5], it is necessary to use a supplementary rule which approximates the sign of the process.

In the following will be used the examples from [6] for testing the algorithms presented in previous sections. There will be denoted with (P) – the case of MBPC algorithm and with (H) – the case of the hybrid algorithm.

Example 1

Consider the system:

$$\begin{aligned} \dot{x}_1 &= x_2 + tanh(u) \qquad & x_1(0) &= 0 \\ \dot{x}_2 &= x_1 + x_2^2 + u \qquad & x_2(0) &= 1.15 \\ y &= x_1 \end{aligned} \tag{15}$$

Using an accurate model, the results obtained by two methods are similar. Though, due to the fact that (P) actions in first place on the state x_1 (the output y), the results obtained for the output signal are better. In fig. 4 at step 45 it is noticed the four predictions of the output signal.

Also, some tests of robustness were realized. In the first test (fig. 5) the control signal u(t) is replaced in equations (15) with 0.5·u(t), in the second test (fig. 6) with 2·u(t).

In the first test, (H) becomes unstable, while (P) succeeds to stabilize the system.

In the second test both algorithms succeed to stabilize the system. The dominant nonlinearity is the quadratic term x_2^2 in the second state equation. Let us consider this equation under the form $\dot{x}_2 = x_1 + a \cdot x_2^2 + u$. For a=2.1 (H) is still stable (fig. 7) but for a=2.2 (H) becomes unstable. (P) succeeds to stabilize the system even for a=3 with the condition of increasing the limits of the control signals (u_{max}=4, u_{min}=-4).

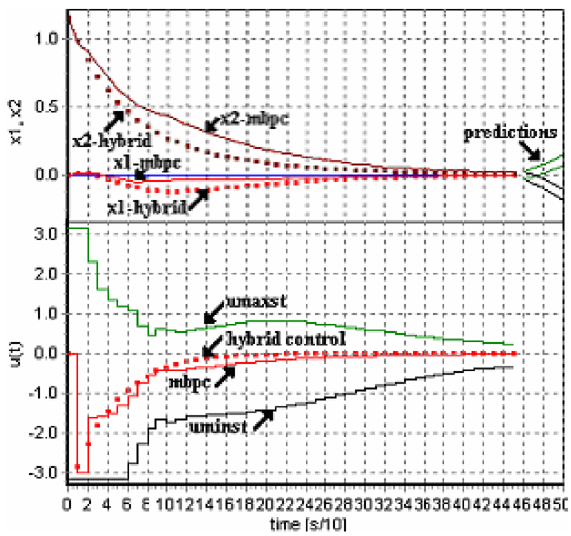

Fig. 4. Example 1. Accurate model.

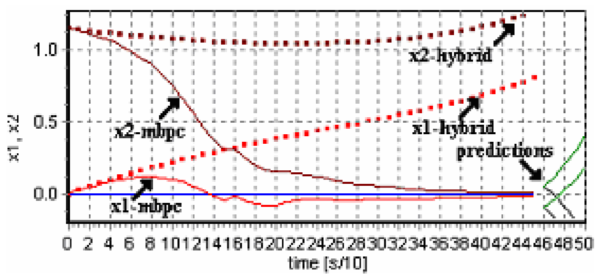

Fig. 5. Example 1. Robustness test 1.

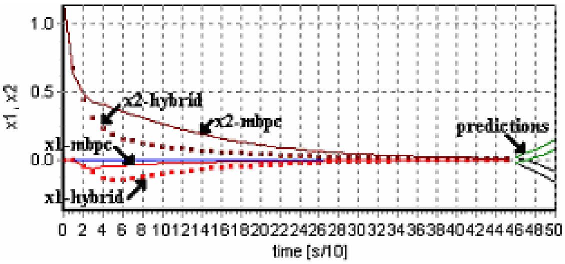

Fig. 6. Example 1. Robustness test 2.

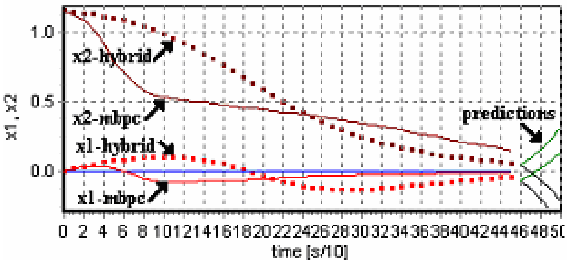

Fig. 7. Example 1. Robustness test 3.

In fig. 8 it is presented the behaviour of (P) in the case of the positioning system. The output y (state x_1) follows a trapeze reference.

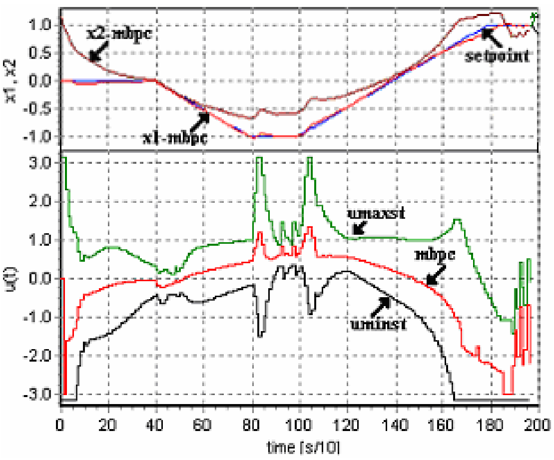

Fig. 8. Example 1. Positioning case.

Example 2

Consider the system:

$$
\begin{aligned}
&\dot{x}_1 = x_1 + x_2 + x_2^2 \qquad && x_1(0) = 0.12 \\
&\dot{x}_2 = x_1 + x_2^2 + u \qquad && x_2(0) = 0 \\
&y = x_1 && \qquad (16)
\end{aligned}
$$

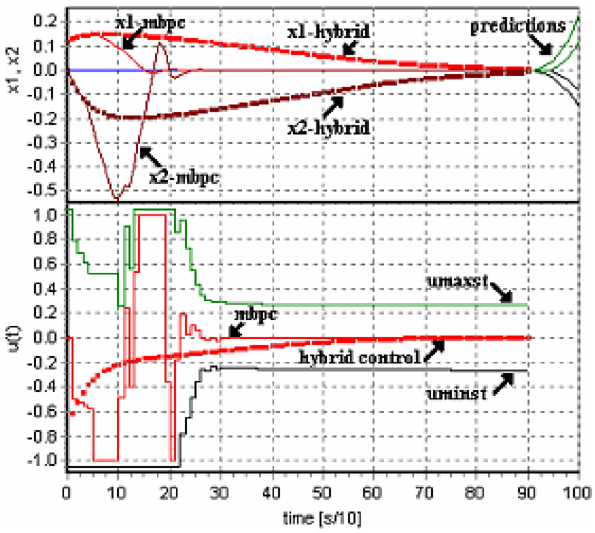

Fig. 9. Example 2. Accurate model.

If the model is accurate, the output reply is better in the case (P), but the variance of the control signal and the variance of the state x_2 increases (fig. 9). For testing the robustness it is modified the equation 2 thus $\dot{x}_2 = x_1 + x_2^2 + a \cdot u$. For a=0.8 (H) becomes unstable while (P) has a good behaviour (fig. 10).

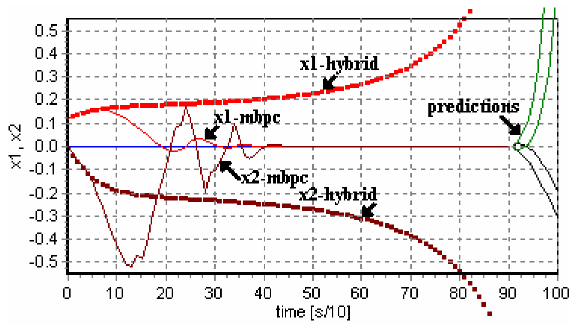

Fig. 10. Example 2. Robustness test 1.

To observing the effect of noise it is considered the measured value of the output signal under the form: $y = x_1 + 0.001 \cdot (random(50) - 25)$. It is noticed that for (P) the output signal it follows much better the reference (fig. 11), with a larger variation of state x_2.

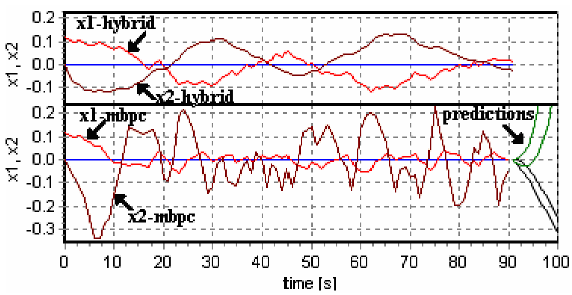

Fig. 11. Example 2. Noise test.

In fig. 12 it is presented the behaviour of (P) in the case of the positioning system. The output $y=x_1$ follows a trapeze reference. It is noticed that for $y_r(t)>0.25$ the setpoint can not be followed because equation $x_2^2 + x_2 + x_1 - \dot{x}_1 = 0$ does not have a real solution if $x_1 - \dot{x}_1 > 0.25$.

Example 3
Consider the system:

$$\begin{aligned} &\dot{x}_1 = \sin(x_2) && x_1(0) = 10 \\ &\dot{x}_2 = u && x_2(0) = 0 \\ &y = x_1 \end{aligned} \tag{17}$$

In the case of the predictive algorithm, for examples 3, 4, 5, it is necessary the introduction of new control sequences or/and some supplementary rules. The reason is the fact that system may change its sign. Possible solutions:

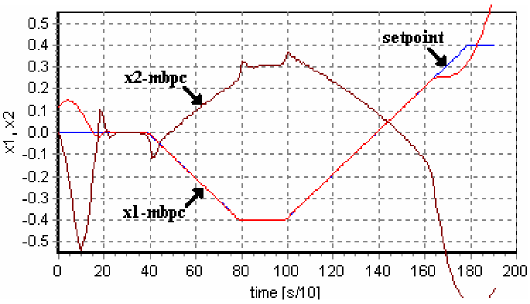

Fig. 12. Example 2. Positioning case.

- approximation of the actual sign of the system; if the sign is negative, it is necessary to use supplementary rules but similar to the rules defined for positive sign [5].
- usage of some supplementary sequences [7]:

$$\begin{aligned} u_5(t) &= \{k \cdot u_{\min}, k \cdot u_{\min}, .., k \cdot u_{\min}\} \\ u_6(t) &= \{0, 0, .., 0\} \\ u_7(t) &= \{k \cdot u_{\max}, k \cdot u_{\max}, .., k \cdot u_{\max}\} \end{aligned} \tag{18}$$

where k <1 is a parameter of the control algorithm.

In the case of usage of accurate model (fig. 13), the reply is more rapid in case of (P). In figure are represented also the form of predictions to the sampling steps 1, 5, 20, 100, 120. Used notations: (a) correspond to the sequences u_3 and u_4, (b) correspond to the sequences u_1 and u_2, (c), (d), (e) correspond to the sequences u_5, u_6, u_7, (a1) and (b1) correspond to the sequences u_3,u_1 respectively u_4,u_2. It was used k=0.2.

Comparatively with examples 1 and 2 it was produced supplementary rules that permits the choosing of the most rapid way to the reference. For example, for sampling steps 1 and 5 it is chosen u(t)=u_{minst} but for the sampling step 20 it is chosen u(t)=0.

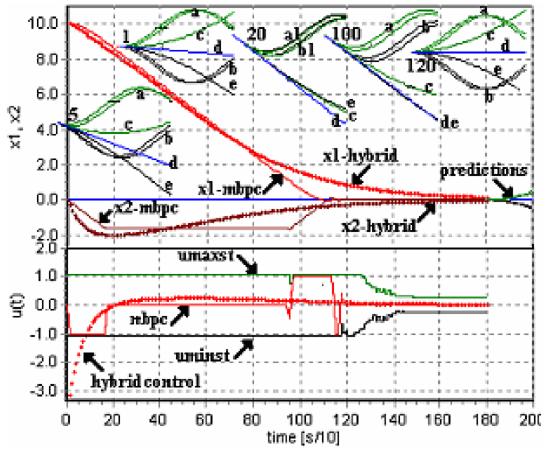

Fig. 13. Example 3. Accurate model.

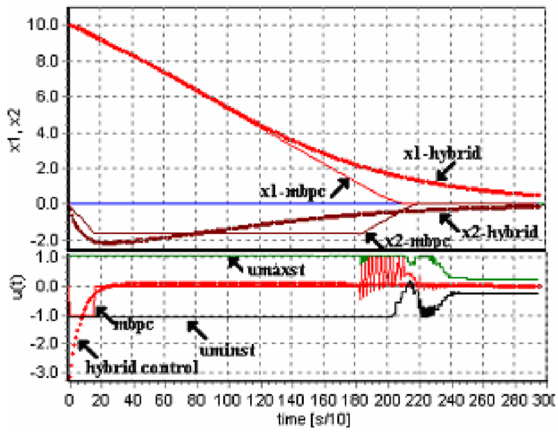

Fig. 14. Example 3. Robustness test.

For the study of robustness, the equation $\dot{x}_2 = u$ was replaced by $\dot{x}_2 = 2 \cdot u$ respectively $\dot{x}_2 = 0.5 \cdot u$. In both cases, the hybrid system is very small affected. The predictive algorithm is very small affected in the first case but in the second case appear dumping oscillations.

Let us consider now the first equation under the form $\dot{x}_1 = \sin(0.5 \cdot x_2)$. It can be noticed the sensible increasing of the response time and, in case of MBPC algorithm a certain trend of oscillation of the control signal in the moment in which the error tends to zero (fig. 14).

Another test was realized modifying the initial state. For $x_1(0)$=15 the hybrid system diverges; the predictive algorithm having a good behaviour.

In fig. 15 it is presented the behaviour way for (P) in the case of a positioning system. The output y (state x_1) follows a trapeze reference.

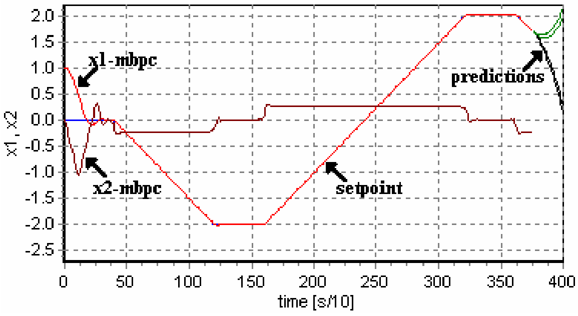

Fig. 15. Example 3. Positioning case.

Example 4

Consider the system:

$$
\begin{array}{ll}
\dot{x}_1 = x_2 & x_1(0) = 3.75 \\
\dot{x}_2 = x_2 + \sin(x_3) & x_2(0) = 0 \\
\dot{x}_3 = u & x_3(0) = 0 \\
y = x_1 &
\end{array} \qquad (19)
$$

The behavior of the two algorithms is similar (fig. 16). In the case of (P) algorithm, using of the relations (8) hasn't lead to favorable behavior.

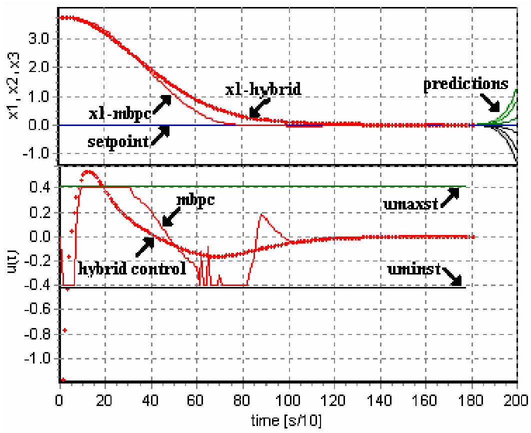

Fig. 16. Example 4

Example 5

Simplified dynamics of the ball-on-beam problem are modeled by the following fourth order system:

$$
\begin{array}{ll}
\dot{x}_1 = x_2 & x_1(0) = 1.15 \\
\dot{x}_2 = x_1 \cdot x_4^2 + \sin(x_3) & x_2(0) = 0 \\
\dot{x}_3 = x_4 & x_3(0) = 0 \\
\dot{x}_4 = u & x_4(0) = 0 \\
y = x_1 &
\end{array} \qquad (20)
$$

As it can be seen (fig. 17), the system (P) has a faster response comparatively with the system (H). Let us consider now the forth equation under the form $\dot{x}_4 = 1.5 \cdot u$.

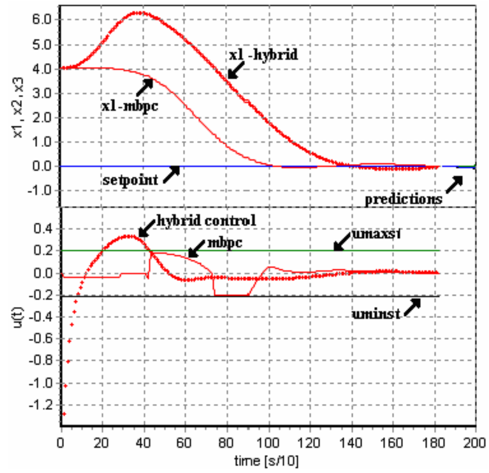

Fig. 17. Example 5

In this case, (P) has a better behavior, (H) being unstable. If $\dot{x}_4 = 0.5 \cdot u$ then the system (H) has a good behavior, meanwhile, in case of system (P), for obtaining a good behavior, the k parameter from relations (18) had to be increased.

5. CONCLUSIONS

The paper presents a simple and intuitive algorithm applied in the case of some nonlinear process. Using the process model and a reduce number of the sequences control, it's simulated the future behavior of the process and based on a set of rules it is chosen the signal control considered optimum at the actual moment. Of course there are some difficulties such as the proof of the stability, the way of choosing of the control sequences and the set of rules which will lead to a better result, choosing some parameters etc. Although, taking into account the simplicity of this algorithm the obtained results in the case of the presented examples by nonlinear systems are remarkable. A demo application that implements the proposed algorithm can be downloaded from reference [8].

REFERENCES

[1] Hangos, K.M., Bokor J., Szederkeny G., "Analysis and control of nonlinear process systems", Springer Verlag ISBN 1-85233-600-5, 2004

[2] E. Camacho and C. Bordons, "Model Predictive Control" Spriger-Verlag, ISBN 3-540-76241-8, 1999

[3] Balan R., Maties V., Hodor V., Zamfira I. "Some issues in the design of adaptive-predictive controllers based on on-line simulation", International Conference OPTIM 2002, Brasov, Romania, pp 447-452 ISBN 973-635-012-6

[4] Balan R., Maties V., Hancu O., "Model predictive control of nonlinear processes using on-line simulation" in Proceedings of International Conference on Automation, Quality and Testing, Robotics, AQTR 2004-Theta 14, Cluj-Napoca, Romania, 13-15 May 2004, pp. 201-207

[5] Balan R., Maties V., Hancu O., "A control algorithm for non-linear processes using on-line simulation and rule-based control", in Proceedings of International Conference of Intelligent Engineering Systems, INES 2004, 19-21 Sept 2004, Cluj-Napoca, Romania, pp. 497-502

[6] A. Simmons, J. Hung, A. Scottedward Hodel, "A hybrid improvement to traditional nonlinear control", IEEE ISIE 2005, June 20-23, 2005, Dubrovnik, Croatia, pag. 49-56, ISBN 0-7803-8739-2, 2005

[7] Balan R., Vistrian Maties, Olimpiu Hancu, Sergiu Stan, "A Predictive Control Approach for the Inverse Pendulum on a Cart Problem", IEEE-ICMA 2005 pag. 2026-2031, July 29 - August 1, Niagara Falls, Ontario, Canada, 2005.

[8] Available: http://zeus.east.utcluj.ro/mec/mmfm/download.htm

[9] A. T. Simmons and J. Y. Hung, "Hybrid control of systems with poorly defined relative degree: The ball-on-beam example," in IECON Proceedings, Conference on Industrial Electronics. Busan, Korea: IEEE, November 2004.