A&QT-R 2004 (THETA 14)
2004 IEEE-TTTC-International Conference on Automation, Quality and Testing, Robotics
May 13 – 15, 2004, Cluj-Napoca, Romania

MOTION CONTROL USING A MODEL BASED ADAPTIVE PREDICTIVE ALGORITHM

Radu Bălan, Vistrian Mătieş, Olimpiu Hancu

Technical University of Cluj-Napoca, Mechanical Faculty / Mechatronics dep.
B-dul Muncii 103-105 Cluj-Napoca, tel 064-155967 e_mail radubalan@yahoo.com

ABSTRACT

In the previous decades, model predictive control (MPC) has developed substantially, both in research and industry. MPC is, perhaps, the most general way of posing the process control problem in the time domain. This control based on on-line optimization, has long been recognized as the winning alternative for constrained systems. MPC is not a new approach, and has traditionally been applied to plants where the dynamics are sluggish enough to permit a sampling rate amenable to optimal input computations between samples, for example chemical process plants. These systems are usually also governed by strict constraints on states, inputs and/or combinations of both. The main limitation of MPC is, however, its on-line computational complexity but with the progress of faster modern computers, it has become possible to extend the MPC approach to systems governed by faster dynamics that warrant this type of solution.

This paper presents the application of an adaptive MPC algorithm in motion control. The basic idea of the algorithm is the on-line simulation of the future behavior of control system, by using a few candidate control sequences. Then, these simulations are used to obtain the 'optimal' control signal. The efficiency and applicability of the proposed algorithm in motion control are demonstrated through applications.

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

1. INTRODUCTION.

Today, many industrial systems are still controlled by simple PID algorithms, despite the better performances usually provided by systems developed following the modern control theory. This is probably not only due to the quite surprising efficacy of this simple control method, but also to the higher computational load and design effort required by most of the more sophisticated control techniques. PID controllers can be used to control a wide range of different processes, need only rough process models to be easily tuned and give pretty good set-point tracking performances. On the other hand it is clear that PID performances, though satisfactory, could be improved when dealing with highly nonlinear processes, or processes featuring unmodeled dynamics and external disturbances. This is specially needed in those applications where highest accuracy is required, like in robot manipulator joint position control. The closed mechanical chains make the dynamics of parallel manipulators coupled and highly nonlinear. To minimize the tracking errors, the dynamical forces need to be compensated by the controller. This control should ensure the best possible compliance of planned trajectory with taking into account the maximum available torques of the drives and effective cooperation between drives.

To overcome such restrictions, many advanced control [4], [5], [6] strategies have been developed in the past, showing that a good control schema could both ease the design and improve performances over process variations from the model.

The aim of this paper is the study of a model based adaptive predictive algorithm and applications in motion control.

2. CONTROL ALGORITHM

In [1], [2] it is proposed an algorithm which uses on-line simulation and rule based control. This algorithm is designed for applications with cvasiconstant set-point, arbitrary changed. Using a few candidate control sequences, for every sample period is computed the predictions of output over a finite horizon and the cost of an objective function. To use this algorithm in motion control, the rules must be changed. Figure 1, contains some examples of the evolutions of error (difference between future set point and predicted output, a_i(t), i=1...4) for candidate sequences (u_i(t),i=1..4). On every curve it is represented the term which is used by algorithm (***min**$_0$*, ***min**$_1$*, ***max**$_0$*, ***max**$_1$*) to compute control effort.

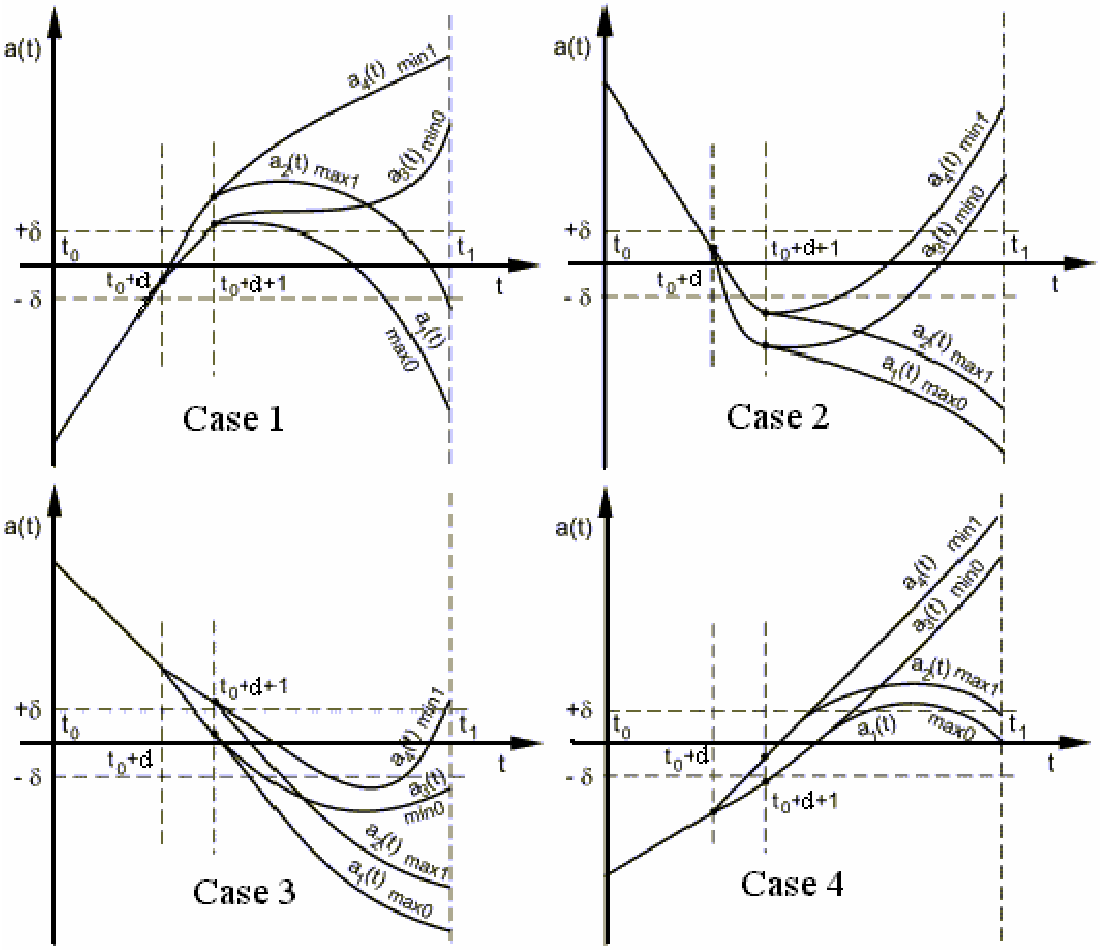

Fig. 1: Examples of output predictions

Notations: d is dead time, t_0 is current time, t_1 is the horizon of output, δ is a parameter which is used for a fine-tuning (first, it is more simple to consider $\delta=0$).
For a first stage, are used the next four control sequences:

$$u_1(t)=\{u_{\min},u_{\min},..,u_{\min}\}\quad u_2(t)=\{u_{\max},u_{\min},..,u_{\min}\}$$
$$u_3(t)=\{u_{\min},u_{\max},..,u_{\max}\}\quad u_4(t)=\{u_{\max},u_{\max},..,u_{\max}\} \tag{1}$$

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

In the second stage, depending by the behavior of control system, it is used an algorithm that modifies the limits of control signal:

$$u_{min} \le u_{minst}(t) \le u(t) \le u_{maxst}(t) \le u_{max} \tag{2}$$
$$\Delta u_{min} \le \Delta u \le \Delta u_{max} \tag{3}$$

In relations (1), the values of u_{max}, u_{min} are replaced with $u_{minst}(t)$, $u_{minst}(t)$. At every sample period, the steps of the algorithm are:

STEP 1: Output measuring y(t); up-date input u[.] and output y[.] vectors
STEP 2: Identification
STEP 3: Simulate the behaviour of system for the next control sequences and compute extremes of future errors:

$$u_1(t)=\{u_{\min st},u_{\min st},..,u_{\min st}\} \Rightarrow \max{}_0 = \max_{t0+d<t<t1}\{a_1(t)\}$$
$$u_2(t)=\{u_{\max st},u_{\min st},..,u_{\min st}\} \Rightarrow \max{}_1 = \max_{t0+d<t<t1}\{a_2(t)\}$$
$$u_3(t)=\{u_{\min st},u_{\max st},..,u_{\max st}\} \Rightarrow \min{}_0 = \min_{t0+d<t<t1}\{a_3(t)\}$$
$$u_4(t)=\{u_{\max st},u_{\max st},..,u_{\max st}\} \Rightarrow \min{}_1 = \min_{t0+d<t<t1}\{a_4(t)\}$$

STEP 4: Use simulated results and next rules:

R1: IF $\min_0>\delta$ THEN $u(t)=u_{minst}(t)$ ELSE
R2: IF $\max_1<-\delta$ THEN $u(t)=u_{maxst}(t)$ ELSE
R3: IF $a_4(t+k+1)>0$ AND $\min_1<-\delta$ THEN $u(t)=u_{maxst}(t)$ ELSE
R4: IF $a_1(t+k+1)<0$ AND $\max_0>\delta$ THEN $u(t)=u_{minst}(t)$ ELSE

$$\text{R5:}\quad u(t)=\frac{u_{\min st}(t)\max{}_1-u_{\max st}(t)\min{}_0}{\max{}_1-\min{}_0}$$

STEP 5: Up-date $u_{maxst}(t)$, $u_{minst}(t)$:
IF R5 THEN IF $u_{maxst}(t)-u_{med}(t)>u_{med}(t)-u_{minst}(t)$
THEN
$u_{maxst}(t)=u_{maxst}(t)-K_{st}(u_{maxst}(t)-u_{med}(t))$
$u_{minst}(t)=2u_{med}(t)-u_{maxst}(t)$
ELSE $u_{minst}(t)=u_{minst}(t)-K_{st}(u_{minst}(t)-u_{med}(t))$
$u_{maxst}(t)=2u_{med}(t)-u_{minst}(t)$
IF $u_{maxst}(t)-u_{med}(t)<D_{ust}$ THEN $u_{maxst}(t)=u_{med}(t)+D_{ust}$
IF $u_{med}(t)-u_{minst}(t)<D_{ust}$ THEN $u_{minst}(t)=u_{med}(t)-D_{ust}$
IF R2 OR R3 THEN $u_{maxst}(t)=(1+K_{st})u_{maxst}(t)$
IF R1 OR R4 THEN $u_{minst}(t)=(1-K_{st})u_{minst}(t)$
IF $u_{maxst}(t)>u_{max}$ THEN $u_{maxst}(t)=u_{max}$.
IF $u_{minst}(t)<u_{min}$ THEN $u_{minst}(t)=u_{min}$.

STEP 6: u(t) filter: IF R5 then $u(t)=K_u u(t)+(1-K_u)u_{med}(t)$
STEP 7: Compute the average of controller's output:

$$u_{med}(t+1) = K_{umed}u_{med}(t)+(1-K_{umed})u(t).$$

Control parameters: K_{st} reduces the difference u_{maxst}-u_{mnist}, D_{ust} is accepted minimum of $u_{maxst}(t)$-$u_{minst}(t)$, K_u, K_{umed} are weight factors.

3. EXPERIMENTS

For experiments, it is considered a position control system based on a DC motor. The model of the process was obtained using experimental identification:

$$H(z) = \frac{0.005(z^2 - 0.26122z + 1.025)}{(z-1)(z+0.40347)(z-0.19248)} \tag{4}$$

For identification, the experiments use the recursive least squares (RLS) algorithm. The limits of control signal and output: u_{min}= -250, u_{max}=250, y_{min}=0, y_{max}=250 units and noise (if exists) is $\sigma = 10^{-3}$. In the next figures, there are represented two or more functions versus sample point, so both axis label only with units.

Example 1.

In this example (fig. 2), the setpoint has a trapeze shape. The parameters of model are constant but unknown. The process and the model have the same structure. Control error e(t), control signal and limits u(t), $u_{minst}(t)$, $u_{maxst}(t)$, are represented in two cases: without noise (1), with noise (2). To view control error, it is used the relation: e(t)←50+50*e(t). In the second case (2), the variance of u(t) is larger, but it can be reduced using desired values for K_u and K_{umed} (steps 6, 7 of algorithm). Initially, the parameters of model and the output are 0; then the parameters are identified using RLS algorithm. The position of poles and zeros after 160 steps is presented in fig. 3 (process and model).

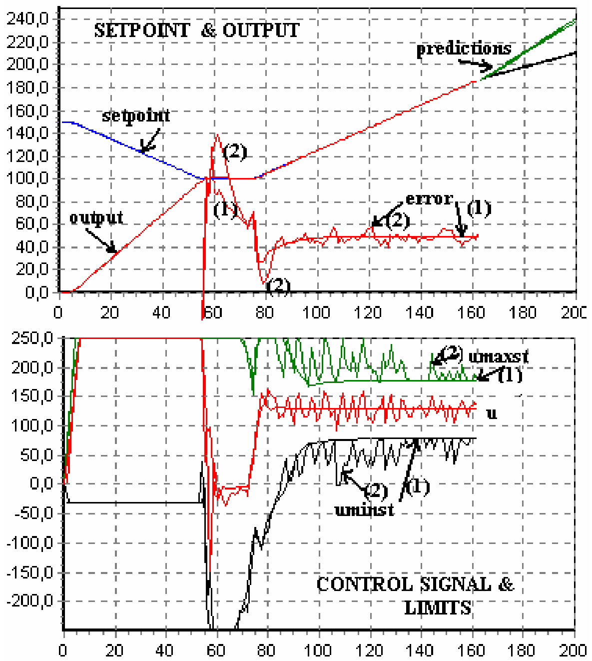

Fig. 2: Example 1: Control system (non-adaptive)

To compare the behaviour of control system in different conditions, a solution is to use a histogram chart. In fig. 5, there are represented the histograms of control error for two cases: (1)-without noise, (2)-with noise $\sigma = 10^{-3}$. The maximum of error is limited to ± 2 units.

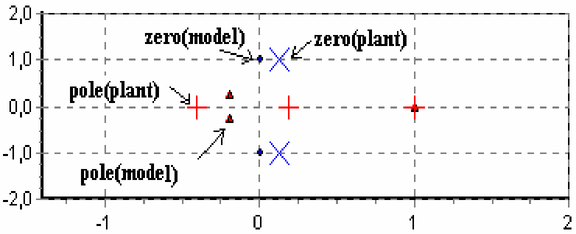

Fig. 3: The poles and zeros after 160 steps

Example 2.

In this example, the setpoint has a trapeze shape (fig.4), the gain factor is time variable: it rises from 0.005 to 0.2, and then decreases to 0.005 with step 0.0005. Although the gain factor has a large variation, the control system has a good behavior. For 360..400

steps, the control signal have a large variation because the shape of setpoint will be changed after step 400.

Example 3.

To improve the quality of control system, especially if the variation of parameters is large, a method is to use multiple models. This method is simple but time consuming. It is possible to use a few models and, at every sample period, to choose the better of them (based on a performance criterion). In fig. 6, using conditions from example 2, the behavior of control system is better in case 1 (multiple model) comparatively with case 2 (one model).

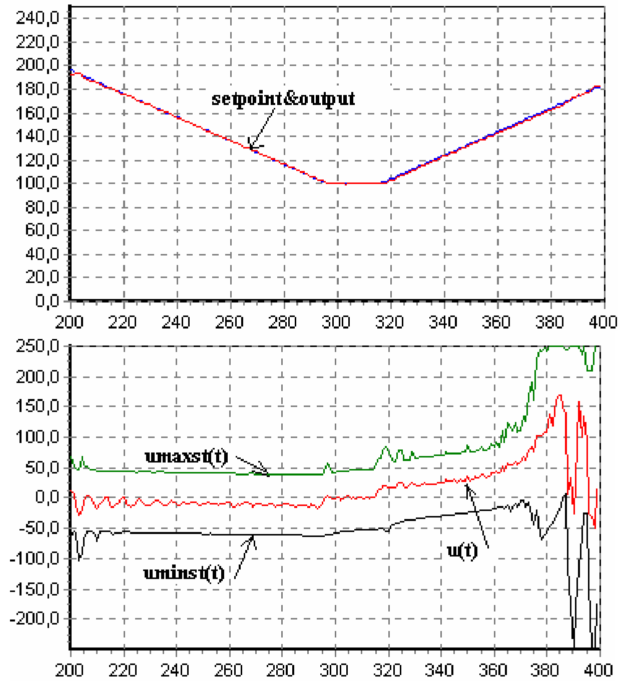

Fig. 4: Example 2: Control system (adaptive case)

Example 4.

The setpoint has a trapeze shape (similarly with example 2,3); a real pole is variable from 0.1 to 0.9 with step 0.001. In fig.7, the histograms of error are: (1)-the parameters of model are unknown but constant, (2)-adaptive case (variable pole).

Example 5.

The setpoint has a sine shape, gain factor is variable from 0.005 to 0.2 with step 0.0005. In fig. 8, the histograms of error are: (1)-the parameters of model are unknown but constant, (2)-adaptive case without noise, (3)-adaptive case with noise $\sigma = 10^{-3}$.

Example 6.

For adaptive case, the setpoint has a sine shape, a real pole is variable from 0.1 to 0.9 with step 0.001. In fig. 9, the histograms of error are: (1)-the parameters of model are unknown but constant, (2) adaptive case with noise $\sigma = 10^{-3}$.

Example 7.

In this example, the setpoint has a sine shape, non-adaptive case. The histograms are (fig. 10): (1)-MPC algorithm, (2)-PID algorithm.

4. CONCLUSIONS

This paper presents the study of a model based predictive control algorithm applied to motion control. The algorithm uses on-line simulation and rule-based control. A (non)linear model of the process, is used directly in control algorithm. The control algorithm is able to maintain better set point tracking performance in various conditions: variable parameters, variable setpoint etc.

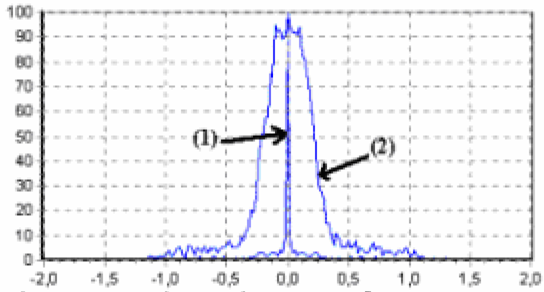

Fig. 5. Example 1: Histogram of errors

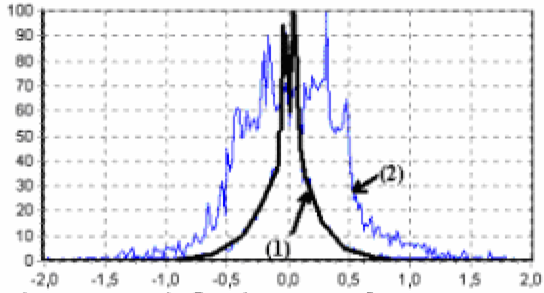

Fig. 6. Example 3: Histogram of errors

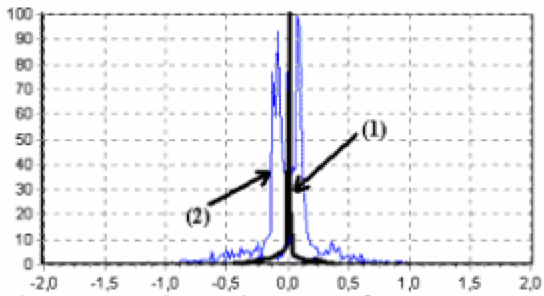

Fig. 7. Example 4: Histogram of errors

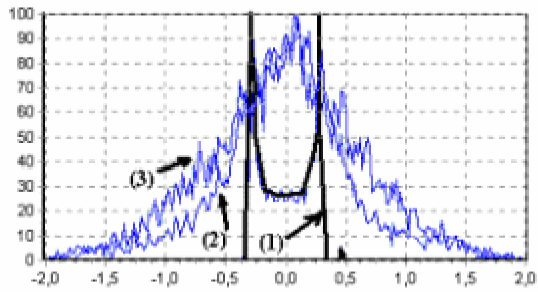

Fig. 8. Example 5: Histogram of errors

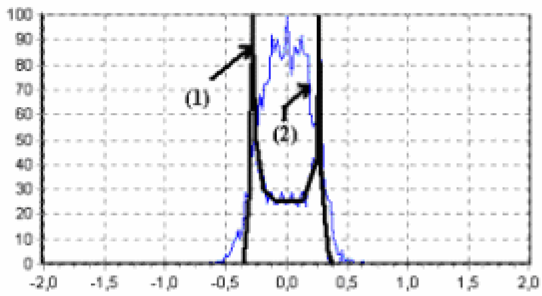

Fig. 9. Example 6: Histogram of errors

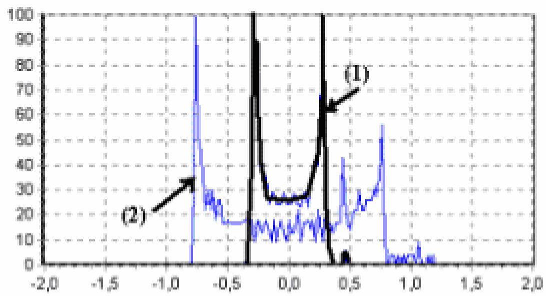

Fig. 10. Example 7: Histogram of errors

5. REFERENCES

[1] Balan R. [2002]: "Adaptive control through on-line simulation. Theoretical aspects." AQTR –Theta 13 Cluj-Napoca, Romania

[2] Balan R. [2001] "Adaptive control systems applied to technological processes", Ph.D. Thesis, T.U. of Cluj-Napoca, Romania.

[3] Dougherty D., Cooper D., [2003], "A practical multiple model adaptive strategy for single-loop MPC", Control Engineering Practice 11

[4] Belda K., [2002], "Generalized predictive control for real-time use in area of parallel robots", Advances in Supervision and Control Systems, Oct.1-4, Strunjan, Slovenia.

[5] Honneger M., "Nonlinear Adaptive Control Of A 6 Dof Parallel Manipulator",

[6] Zilouchian, A., Jamshidi, M.: [2001] Intelligent Control Systems Using Soft Computing Methodologies CRC Press