

# The Disturbance Model in Model Based Predictive Control

Robin DE KEYSER, Clara Mihaela IONESCU

**Abstract** -- Model Based Predictive Control (MBPC) is a control methodology which uses a process model on-line in the control computer; this model is used for calculating output predictions and optimizing control actions. The importance of the system model has been generally recognized, but less attention has been paid to the role of the disturbance model. In this paper the importance of the disturbance model is indicated with respect to the EPSAC approach to MBPC. To illustrate this importance, an example of this advanced control methodology applied to a typical mechatronic system is presented, to compare the performances obtained by using different disturbance models. It clearly shows the benefits of using an 'intelligent' disturbance model instead of the 'default' model generally adopted in practice.

**Index Terms** -- disturbance model, disturbance rejection, mechatronics, model, prediction, predictive control.

## I. INTRODUCTION

MODEL Based Predictive Control (MBPC) is nowadays one of the most important control strategies generously accepted in industry. This is due to its capability to deal with multivariable and non-linear processes, constraints and modeling errors, as well as unusual dynamic behavior of processes.

During the last decades, MBPC has become an important, distinctive part of control theory and application. A great interest has been shown for this methodology resulting in many excellent reviews and books [1-6].

The MBPC methodology is based on a model of the process, which is used for calculating the prediction of the controlled variables. It is characterized by:

- explicit *on-line* use of the process model to forecast the process output at future time instants;
- calculation of an optimal control strategy based on the minimization of one or more cost functions, possibly including constraints on the process variables.

The initial predictive algorithms were utilizing linear models and a large number of algorithms have appeared in the literature [7-10], mainly differing in:

- the type of model used to represent the process and its disturbances;
- the cost function(s) to be minimized, with or without constraints.

Taking into account that the PID controller is still the *bread and butter* of the instrumentation engineer in everyday industrial practice, it is important to realize that MBPC requires the extra-effort of identifying a *process model*. This involves a model for the dynamic system itself *as well as* a model for the disturbances. While PID generally deals with control loops situated at the lower level of the control pyramid, the more advanced MBPC covers the optimization level and it has an important role to play in future process and production industries.

The purpose of this paper is to emphasize the role of the *disturbance model* in the EPSAC approach to MBPC [9],[11],[12]. This role has generally been sub-estimated.

The content of this paper is as follows. The 2<sup>nd</sup> section describes the MBPC methodology used. The 3<sup>rd</sup> section presents in detail the role of the disturbance model and its importance, with 2 design solutions: the *default* design and the *optimized* design. A comparison between these 2 designs and the classical PID controller is presented in the 4<sup>th</sup> section, using a typical mechatronic control example.

## II. MBPC METHODOLOGY

Being one of the earlier predictive controllers, EPSAC [9],[11],[12] is based on a generic process model:

$$y(t) = x(t) + n(t) \quad (1)$$

which is illustrated in Fig. 1, with:

- $y(t)$ : (measured) process output;
- $u(t)$ : process input;
- $x(t)$ : model output;
- $n(t)$ : process ( $y$ ) vs model ( $x$ ) *disturbance*;
- $t$ : discrete-time index.

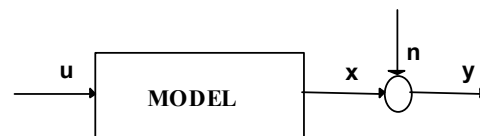


Fig. 1. Generic Process Model

The disturbance  $n(t)$  includes the effects in the measured output  $y(t)$  which do not come from the model input  $u(t)$  via

R. De Keyser is with the EeSA-department of Electrical energy, Systems & Automation, Ghent University, Technologiepark 913, B-9052 Gent, Belgium (telephone: +32-9-2645585, fax: +32-9-2645603, e-mail: rdk@autoctrl.rug.ac.be).

C. M. Ionescu is with the Department of Automatic Control and Industrial Informatics, University of Galati, Domneasca Street no. 47, 6200 Galati, Romania, currently on leave to Ghent University, Belgium.

the available model. These non-measurable disturbances have a stochastic character with non-zero average value, which can be modeled by a coloured noise process:

$$n(t) = \frac{C(q^{-1})}{D(q^{-1})} e(t) \quad (2)$$

with:

- $e(t)$ : uncorrelated (white) noise with zero mean value;
- $C(q^{-1})$ ,  $D(q^{-1})$ : monic polynomials in the backward shift operator  $q^{-1}$  of orders  $n_c$  and  $n_d$ .

This filter  $C(q^{-1})/D(q^{-1})$  is considered to be a *design* filter. It plays an important role in MBPC (which has generally not been recognized appropriately). It will be further presented in detail in Section III.

#### A. Prediction Algorithm

The model output  $x(t)$  represents the effect of the control input  $u(t)$  on the process output  $y(t)$  and is also a non-measurable signal, and the relationship between  $u(t)$  and  $x(t)$  is given by the generic dynamic system model:

$$x(t) = f[x(t-1), x(t-2), \dots, u(t-1), u(t-2), \dots] \quad (3).$$

The fundamental step in MBPC methodology consists in prediction of the process output  $y(t+k)$  at time instant  $t$ , indicated by  $\{y(t+k|t), k=1 \dots N_2\}$ , over the *prediction horizon*  $N_2$ , and based on:

- measurements available at sampling time instant  $t$ :  $\{y(t), y(t-1), \dots, u(t-1), u(t-2), \dots\}$ ;
- future values of the input signal (postulated at time  $t$ ):  $\{u(t|t), u(t+1|t), \dots\}$ .

Using the generic process model (1), the predicted values of the output are:

$$y(t+k|t) = x(t+k|t) + n(t+k|t) \quad (4)$$

Prediction of  $x(t+k|t)$  and of  $n(t+k|t)$  can be done respectively by recursion of the process model (3) and by using filtering techniques on the noise model (2) [11],[12].

#### B. Control Algorithm

In EPSAC for linear models, the future response is then considered as being the cumulative result of two effects:

$$y(t+k|t) = y_{\text{base}}(t+k|t) + y_{\text{optimize}}(t+k|t) \quad (5)$$

The two contributions have the following origins:

$$y_{\text{base}}(t+k|t):$$

- effect of past control  $\{u(t-1), u(t-2), \dots\}$  (initial conditions at time  $t$ );
- effect of a *base* future control scenario, called  $u_{\text{base}}(t+k|t)$ ,  $k \geq 0$ , which is defined *a priori*; some ideas on how to choose  $u_{\text{base}}$  are presented in [11],[12];

for linear systems the choice is irrelevant, a simple choice being  $\{u_{\text{base}}(t+k|t) \equiv 0, k \geq 0\}$ ;

- effect of future (predicted) disturbances  $n(t+k|t)$ .

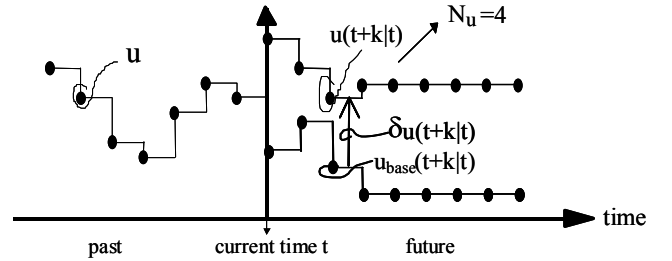
The component  $y_{\text{base}}(t+k|t)$  can be easily obtained using (2)(3)(4) taking  $u_{\text{base}}(t+k|t)$  as the model input for (3).

$$y_{\text{optimize}}(t+k|t):$$

- effect of the *optimizing* future control actions  $\{\delta u(t|t), \delta u(t+1|t), \dots, \delta u(t+N_u-1|t)\}$  with  $\delta u(t+k|t) = u(t+k|t) - u_{\text{base}}(t+k|t)$ .

Refer to Fig. 2 for the concepts of *base* and *optimizing* controls. Notice that  $u(t+k|t)$  is constrained to be constant from  $k=N_u$  on (and this is realized by selecting  $u_{\text{base}}(t+k|t)$  constant from  $k=N_u$  on and by imposing that  $\delta u(t+k|t)$  should be constant from  $k=N_u$  on). The *design* parameter  $N_u$  is called the *control horizon* (a well-known concept in MBPC-literature).

Fig. 2. The EPSAC concept of *base/optimizing* controls



From Fig. 2 it is obvious that the component  $y_{\text{optimize}}(t+k|t)$  is the cumulative effect of a series of *impulse* inputs and a *step* input:

- an *impulse* with amplitude  $\delta u(t|t)$  occurring at time  $t$ , resulting in a contribution  $h_k \delta u(t|t)$  to the process output at time  $t+k$  ( $k$  sampling periods later);
- an *impulse* with amplitude  $\delta u(t+1|t)$  occurring at time  $t+1$ , resulting in a contribution  $h_{k-1} \delta u(t+1|t)$  to the predicted process output at time  $t+k$  ( $k-1$  sampling periods later);
- etc;
- finally a *step*  $\delta u(t+N_u-1|t)$  at time  $t+N_u-1$ , resulting in a contribution  $g_{k-N_u+1} \delta u(t+N_u-1|t)$  to the predicted process output at time  $t+k$ .

The cumulative effect of all impulses and the step is:

$$y_{\text{optimize}}(t+k|t) = h_k \delta u(t|t) + h_{k-1} \delta u(t+1|t) + \dots + g_{k-N_u+1} \delta u(t+N_u-1|t) \quad (6)$$

The parameters  $g_1, g_2, \dots, g_{N_2}$  are the coefficients of the *unit step response* of the system, i.e. the response of the system for a stepwise change of the input (with amplitude 1). The parameters  $h_1, h_2, \dots, h_{N_2}$  are the coefficients of the *unit impulse response* of the system and can be easily calculated

from the step response coefficients and vice versa:  $h_k = g_k - g_{k-1}$  (and  $h_0 = h_{-1} = \dots = g_0 = g_{-1} = \dots = 0$ ).

Using (5) and (6), the **key EPSAC-MBPC equation**:

$$\mathbf{Y} = \bar{\mathbf{Y}} + \mathbf{G}\mathbf{U} \quad (7)$$

is obtained, where:

$$\begin{aligned} \mathbf{Y} &= [y(t+N_1|t) \cdots y(t+N_2|t)]^T \\ \bar{\mathbf{Y}} &= [y_{\text{base}}(t+N_1|t) \cdots y_{\text{base}}(t+N_2|t)]^T \\ \mathbf{U} &= [\delta u(t|t) \cdots \delta u(t+N_u-1|t)]^T \\ \mathbf{G} &= \begin{bmatrix} h_{N_1} & h_{N_1-1} & h_{N_1-2} & \cdots & g_{N_1-N_u+1} \\ h_{N_1+1} & h_{N_1} & h_{N_1-1} & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ h_{N_2} & h_{N_2-1} & h_{N_2-2} & \cdots & g_{N_2-N_u+1} \end{bmatrix} \end{aligned} \quad (8)$$

The controller output is then the result of minimizing the cost function:

$$V(\mathbf{U}) = \sum_{k=N_1}^{N_2} [r(t+k|t) - y(t+k|t)]^2 \quad (9)$$

with  $r(t+k|t)$  the desired *reference trajectory* and the horizons  $N_1, N_2$  being design parameters.

It is now straightforward to derive the solution. The cost function (9) is a quadratic form in  $\mathbf{U}$ , having the following structure using the matrix notation from (8) and with  $\mathbf{R}$  defined similarly to  $\mathbf{Y}$ :

$$V(\mathbf{U}) = [\mathbf{R} - \bar{\mathbf{Y}} - \mathbf{G}\mathbf{U}]^T [\mathbf{R} - \bar{\mathbf{Y}} - \mathbf{G}\mathbf{U}] \quad (10)$$

which leads after minimization w.r.t.  $\mathbf{U}$  to the optimal solution:

$$\mathbf{U}^* = [\mathbf{G}^T \mathbf{G}]^{-1} \mathbf{G}^T (\mathbf{R} - \bar{\mathbf{Y}}) \quad (11)$$

The matrix  $\mathbf{G}^T \mathbf{G}$  which has to be inverted has dimension  $N_u \times N_u$ . For the default case  $N_u=1$ , this results in a simple *scalar* control law. Only the first element  $\delta u(t|t)$  in  $\mathbf{U}^*$  is required in order to compute the actual control input applied to the process:

$$u(t) = u_{\text{base}}(t|t) + \delta u(t|t) = u_{\text{base}}(t|t) + \mathbf{U}^*(1) \quad (12)$$

At the next sampling instant  $t+1$ , the whole procedure is repeated taking into account the new measurement information  $y(t+1)$ . This is called the principle of *receding horizon control*, another well-known MBPC-concept.

### III. ROLE OF THE DISTURBANCE MODEL

#### A. Default Design

As described in the beginning of section II, the disturbance  $n(t)$  includes all effects in the measured output  $y(t)$  which do not come from the model output  $x(t)$ . This is a *fictitious* (and thus non-measurable) signal and it includes effects of process disturbances, effects of other (un-modeled) process inputs, measurement noise, model errors, etc.

The net effect of all these unknown disturbances has a stochastic character with non-zero average value and can be modeled by a colored noise process as in (2):

$$n(t) = \frac{C(q^{-1})}{D(q^{-1})} e(t)$$

where the filter  $C(q^{-1})/D(q^{-1})$  is the *disturbance model*.

It is common practice in the MBPC approach to consider this filter as a *design* filter. It can be used - in order to improve the quality of the control performance - to “supply” information to the controller about the type of disturbances that can be expected.

The simplest way to design this filter is to neglect it, thus make it equal to 1. In doing this, *not any* information about the disturbance is given to the controller! In fact, this results in ‘telling’ to the MBPC-controller that the disturbance  $n(t) = e(t)$ , defined as uncorrelated noise with zero-mean average value. As a consequence then, the controller will not take any specific action to remove *non-zero*-mean disturbances. Usually, the disturbance has in practice a non-zero average component, and a steady-state control error can thus be expected as the result of a permanent disturbance.

A better choice for the disturbance model might be:

$$\frac{C(q^{-1})}{D(q^{-1})} = \frac{1}{1-q^{-1}} \quad (13)$$

resulting in a disturbance signal  $n(t)$  with *non-zero* average value. In this case the MBPC-controller will *intrinsically* take action to remove steady-state errors, similar to the effect of the integrator in a PID-type controller. Notice that (13) is the ‘default’ disturbance model that is usually applied in practice.

#### B. Intelligent Design

The ‘default’ disturbance model (13) still does not supply too much useful information to the MBPC-controller about the type of disturbance that is acting upon the process. Its main advantage is that it is easy-to-design (in fact, there is no design at all!).

In practical applications it is generally not difficult to obtain supplementary information about the kind of disturbance acting on the process. The disturbance signal  $n(t)$  can be reconstructed using the generic model (1):  $n(t) = y(t) - x(t)$ , by *measuring* the process output  $y$  and *calculating* the model output  $x$  with the system model (3).

As  $n(t)$  has usually the character of a correlated (colored) random signal, a useful and simple approach is then to calculate its PSD using a spectral analysis software, to detect around which frequency (frequencies) the main disturbance energy is situated.

Assume as an example that the spectrum of these disturbances leads to the conclusion that the main energy is around a certain frequency  $f_0$ . Then a more ‘intelligent’ filter  $C(q^{-1})/D(q^{-1})$  can be designed so that it has a band-pass characteristic around this frequency.

One way to design such filter is to use special digital filter design techniques (e.g. a Butterworth-filter), which may not be necessarily the simplest and most straightforward approach. Moreover it will lead to a rather complex disturbance filter, which introduces some drawbacks from the MBPC-point-of-view (drawbacks which are not further elaborated here). Moreover, an *accurate* filter design is not really necessary from the control point of view (which is shown experimentally in Section IV).

### C. Disturbance Filter

An alternative simple, effective and straightforward method to design the disturbance filter will now be presented. In order to indicate the presence of disturbance energy around the frequency  $f_0$  it is sufficient to place one of the poles in its transfer function near the unit circle (around frequency  $f_0$ ), resulting in a peak in the frequency response:

$$\frac{C(q^{-1})}{D(q^{-1})} = \frac{(\dots)(\dots)}{(1 - ae^{j\alpha}q^{-1})(\dots)(\dots)} \quad (14)$$

with  $\alpha = 2\pi f_0 T_s$  and  $a \cong 1$  ( $T_s$  is the sampling period and  $a \leq 1$  for stability).

Indeed, the filter frequency response is obtained by:

- interpreting  $q$  as the complex variable  $z = e^{sT_s}$
- replacing the Laplace-operator by  $s = j\omega = j2\pi f$ .

This then leads to a factor  $1 - ae^{j2\pi(f_0 - f)T_s}$  in (14), which approaches zero around the frequency  $f = f_0$ . As this factor is in the denominator of expression (14), it will result in a peak in the modulus of the frequency response.

The design parameter  $a$  can be used:

- to create a sharp and high peak (by taking  $a$  close to 1, e.g.  $a=0.99$ ; in the limit it can be equal to 1); this can be done in case the location  $f_0$  of the main disturbance energy is well-known;
- or to flatten-out the shape of the peak (by making  $a$  somewhat smaller than 1, e.g.  $a=0.90$ ); this allows to express some uncertainty about the exact location of the disturbance energy.

In conclusion, an effective disturbance filter would thus be:

$$\frac{C(q^{-1})}{D(q^{-1})} = \frac{1}{(1 - q^{-1})(1 - ae^{+j\alpha}q^{-1})(1 - ae^{-j\alpha}q^{-1})} \quad (15)$$

## IV. MECHATRONIC APPLICATION

The method can be applied to most kind of processes. As an example we will focus on the field of mechatronic systems, by applying it to a system with typical transfer function:

$$H(s) = \frac{K}{s(1 + \tau_1 s)(1 + \tau_2 s)} \quad (16)$$

This transfer function is indeed typical for many electro-mechanical position servos, the smaller time constant  $\tau_1$  representing the dynamics of the electrical actuator, the bigger time-constant  $\tau_2$  representing the dynamics of the mechanical load, and the pure integrator being the link between velocity and position. Practical examples are manifold, e.g. a parabolic antenna positioning system, a robot arm, an active suspension system, a laser beam positioning system, ...

As an example let us further focus on an antenna position system, where the objective is to control the antenna position towards a fixed target (setpoint is zero) despite severe wind disturbances with main energy around 2Hz.

The control loop block scheme is then given in Fig.3. In the simulation the wind disturbance is generated as band-pass filtered noise by means of a 4<sup>th</sup>-order Butterworth filter with pass-band [1.75 ... 2.25]Hz. Also notice that the wind acts as a disturbing torque on the mechanical structure, thus being an *input-disturbance*. The sampling period was  $T_s=20$ ms.

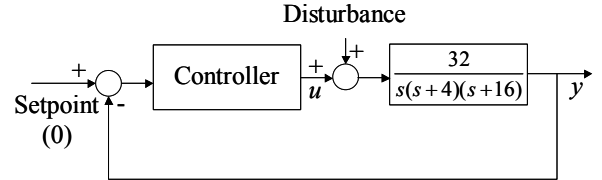


Fig. 3. Mechatronic Control Loop

As a reference for the MBPC-results, a PID-controller has also been designed:

$$R(s) = K_p \left( 1 + \frac{1}{T_i s} + T_d s \right) \quad (17)$$

using a frequency-domain design software with the specification: phase-margin=50°, and resulting in the PID-parameters:  $K_p=22.7$ ;  $T_i=0.60$ ;  $T_d=0.15$ .

The MBPC design parameters were  $N_u=1$ ;  $N_1=1$ ;  $N_2=2$  and the ‘intelligent’ disturbance filter was here (with  $a=1$  and  $\alpha=2\pi \cdot 2 \cdot 0.020=0.25$ ):

$$\frac{C(q^{-1})}{D(q^{-1})} = \frac{1}{(1 - q^{-1})(1 - e^{+0.25j}q^{-1})(1 - e^{-0.25j}q^{-1})} \quad (18)$$

The comparison between PID, *default* and *intelligent* MBPC control performance is depicted in Figs.4-6 below.

As expected, MBPC leads to better performance compared to PID (at the expense of a more complicated control algorithm). The *default* MBPC reduces the disturbance amplitude to about 1/3 compared to PID (Figs.4/5). However, the 'real' improvement is obtained when switching from *default* to *intelligent* MBPC (Figs.5/6).

Finally, it might also be interesting to evaluate and compare the control energy required by the several methods. In Figs.7-9 the controller output is depicted for each method.

The conclusion is that the MBPC requires about 2-times the effort of PID. However, the real interesting thing is that *no extra* control effort is required by the *intelligent* MBPC compared to the *default* MBPC! (when comparing Figs. 8/9 in detail, the 2 signals are however *not* identical).

## V. CONCLUSIONS

The first and the most important step in applying MBPC is the identification of the system model. The second step is usually the tuning of the design parameters (prediction and control horizons).

However, MBPC also gives the possibility to specify a *disturbance model*, next to the *system model*. An intelligent design of this disturbance model allows to supply to the controller useful information regarding the type of disturbance. In doing so, the controller will be very effective in suppressing this disturbance.

A smart choice of this disturbance filter can then result in:

- elimination of steady-state errors;
- but also in suppressing specific disturbances;
- and increasing robustness against modeling errors.

## REFERENCES

- [1] Camacho, E.F. and C. Bordons, *Model Predictive Control*, Springer-Verlag, London, 1999.
- [2] De Keyser, R.M.C., P. Van de Velde and F. Dumortier, A comparative study of self-adaptive long-range predictive control methods, *Automatica*, 24, no. 2, pp.149-163, 1988.
- [3] De Keyser, R.M.C., Basic Principles of Model Based Predictive Control, *ECC 91 European Control Conference, Grenoble*, pp.1753-1758, 1991.
- [4] Garcia, C.E., D.M. Prett and M. Morari, Model Predictive Control: Theory and Practice – a Survey, *Automatica*, 25, no. 3, pp.1753-1758, 1989.
- [5] Qin, S.J. and T.A. Badgwell, An overview of industrial model predictive control technology, in *Chemical Process Control-AIChE Symposium Series (J. Kantor, C. Garcia and B. Carnahan, Eds.)*, AIChE, New York, pp.232-256, 1997.
- [6] Rawlings, J.B., Tutorial Overview of Model Predictive Control, *IEEE Control Systems Magazine*, 20, No. 3, pp.38-52, 2000.
- [7] Richalet, J.A., A. Rault, J.L. Testud and J. Papon, Model predictive heuristic control: applications to an industrial process, *Automatica*, 14, pp.413-428, 1978.
- [8] Cutler, C.R. and B.L. Ramaker, Dynamic matrix control – A computer control algorithm, *Proc. Joint Automatic Control Conference*, San Francisco, 1980.
- [9] De Keyser, R.M.C. and A.R. Van Cauwenberghe, Extended prediction self-adaptive control, *IFAC Symp. on Identification and System Parameter Estimation*, York, pp.1255-1260, 1985.
- [10] Clarke, D.W., C. Mohtadi and P.S. Tuffs, Generalized predictive control: I – the basic algorithm and II – Extensions and interpretations, *Automatica*, 23, pp.137-160, 1987.

[11] De Keyser, R.M.C., A Gentle Introduction to Model Based Predictive Control, *European Union EC-ALFA-PADI2 Int. Conf. on Control Engineering and Signal Processing*, Piura, 1998.

[12] De Keyser, R.M.C., Model Based Predictive Control for Linear Systems, Invited Paper *UNESCO Encyclopaedia of Life Support Systems* (to appear 2003).

[13] Åström, K.J. and T. Hägglund, *PID Controllers: Theory, Design and Tuning*, Instrument Society of America, Research Triangle Park, 1995.

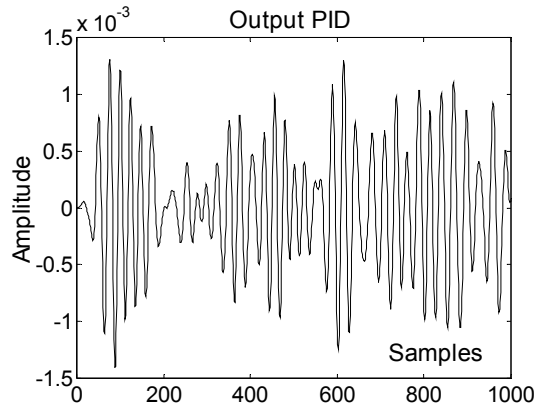


Fig. 4. Process output  $y(t)$  for PID

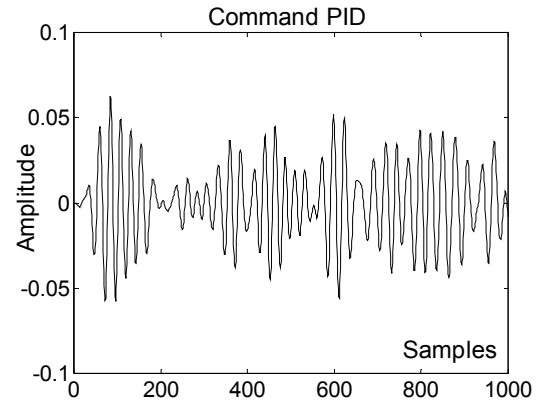


Fig. 7. Controller output  $u(t)$  for PID

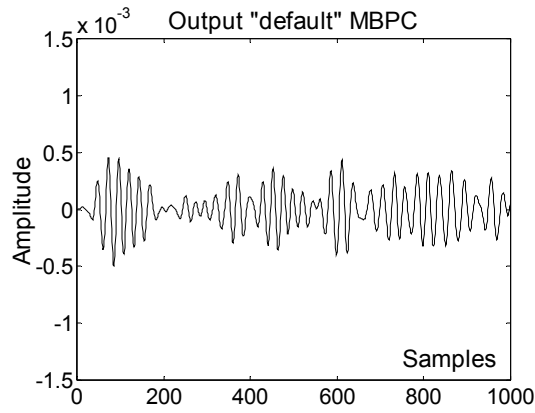


Fig. 5. Process output  $y(t)$  for *default* MBPC

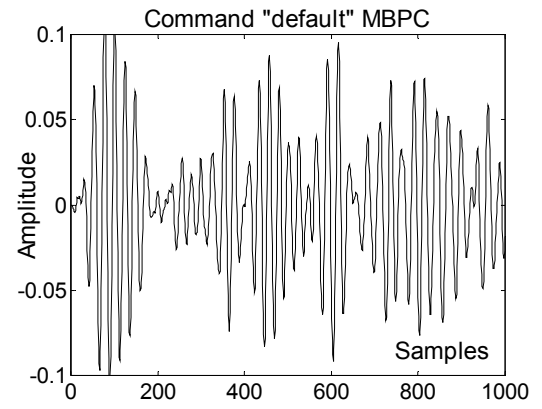


Fig. 8. Controller output  $u(t)$  for *default* MBPC

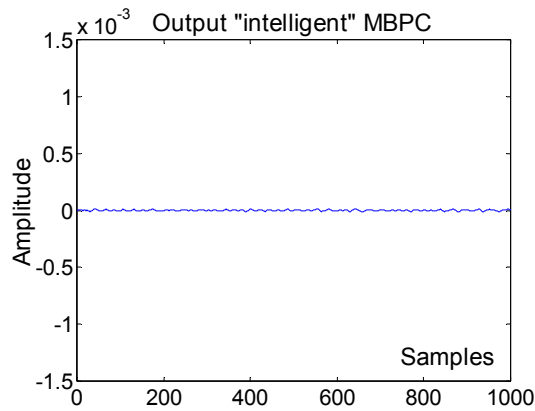


Fig. 6. Process output  $y(t)$  for *intelligent* MBPC

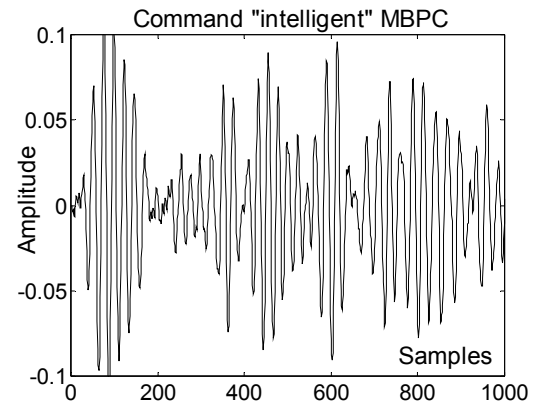


Fig. 9. Controller output  $u(t)$  for *intelligent* MBPC