MODEL BASED PREDICTIVE CONTROL FOR LINEAR SYSTEMS

Robin DE KEYSER
Ghent University, Belgium

Keywords: MBPC, MPC, NMPC, linear, nonlinear, model, process, process model, disturbance, disturbance model, prediction, predictive control, cost function, constraint, multivariable, interactive, SISO, MIMO, prediction horizon, control horizon, coincidence horizon, reference trajectory, base response, optimizing response, semiconductor, wafer, temperature control, singular value decomposition, principal components analysis

Contents

Summary

Model Based Predictive Control (MBPC) is a control methodology which uses *on-line* (=in the *control computer*) a process model for calculating predictions of the future plant output and for optimizing future control actions. In fact MBPC is not a single specific control strategy but rather a family of control methods which have been developed with certain ideas in common. In this chapter these ideas are introduced and developed step by step, specifically for linear systems and using the principles of the EPSAC methodology as a typical member of the MBPC family. The MBPC method is described here in detail both for Single Input Single Output (SISO) and for Multiple Input Multiple Output (MIMO) systems and attention is paid to the practically important concept of constrained control.

Predictive control is nowadays quite popular and successful in real-life industrial applications.

The actual computer implementation, for which the algorithms are presented in detail in this contribution, turns out to be extremely simple although computationally more demanding than the widely used PID controller. However, with the current computer power available, computational complexity is not the real restriction. The performance of MBPC is however critically dependent on the quality of the dynamic process model that is available. The first step in a predictive control design is then the modeling and identification of the process. In this contribution it is assumed that a good process model is available.

1. Introduction

Model Based Predictive Control (MBPC) is a control methodology developed around certain common key principles. Two of these principles are:

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

The various algorithms, members of the large MBPC-family, differ mainly in:

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

The MBPC-strategy is simple to understand and makes good practical sense: 1) use the process model to predict the evolution of the process output as a function of future (intended) control actions and 2) minimize (over these control actions) a specified cost index; this cost includes the errors between desired and predicted process outputs, and possibly also the required control effort.

This type of control is of an *open generic nature*: the *basic principles* of predictive control became now well-known and well-understood, and are open to many useful future contributions from the academic as well as from the industrial world. The methodology has also been widely accepted in both worlds, contrary to many other control strategies. Many applications of predictive control are nowadays successfully in use in all fields of industrial activity. The good performance of these applications is well appreciated and has triggered an increasing interest in the methodology during the last decennium, although the pioneering work started already 25 years ago.

Industrial developments in Europe (MAC) and in the USA (DMC) resulted around 1980 in the first commercial control packages using explicitly a process model to predict and to control the process variables. Simultaneously, although independently, some European *academic* research groups, with a strong history in adaptive prediction and control, started the development of controllers based on multistep predictors (EPSAC; EHAC; MUSMAR; GPC).

These names are only stated to be indicative and represent a few of the (earlier) MBPC algorithms. Since the beginning of the 1990s, a real boom in the number of industrial MBPC applications has been reported (first in the USA and Japan, later also in Europe), in parallel with a dramatic increase of the academic research activity on MBPC-related topics.

A number of benefits are at the basis of this growing success story:

- MBPC concepts can be mastered in a short time by technical staff members with a limited control background: the principles are *intuitive* and the design parameters are *performance oriented*;
- MBPC can handle *nonlinear multivariable interactive* control problems in a straightforward way;
- MBPC is currently the most natural approach to *constrained* control, a topic which is of utmost importance in everyday industrial practice (actuator constraints, safety constraints, quality constraints, …);
- MBPC has inherent *dead-time compensation* of time-delay processes and inherent *feedforward compensation* of measured disturbances; it is able to control processes with *unusual dynamic behavior* (such as non-minimum phase, highly oscillatory or unstable processes);
- MBPC is an *open methodology*, based upon certain *key principles*, but allowing for future extensions, contributions, alternatives and improvements from many different sources;

Besides these convincing benefits, which gradually attracted more members into the MBPC club, the industrial dissemination of MBPC technology was also made possible by the fact that certain MBPC-*enabling technologies* have now reached a certain state of maturity:

- *modeling and identification*: the techniques are now diverse and powerful, able to work in an adverse environment, even with poor signal/noise ratios;
- *digital computers*: they are fast, reliable and affordable, able to crank *on-line* complex algorithms such as constrained MBPC.

Indeed, to be fair w.r.t. the popular and relatively easy PID controller (the ‘bread and butter’ of the instrumentation engineer in industry), it should be mentioned that MBPC *does need a process model* (and model *identification* still requires much effort and expertise) and that the *computational requirements* of MBPC are of an order of magnitude larger than those of PID.

In a sound engineering approach, PID and MBPC should be considered as *complementary* rather than as competitive technologies: PID deals with the bulk of control loops, mainly situated at the lower level of the control pyramid (regulatory/stabilizing/DDC controls); MBPC tackles the fewer but economically more important loops, situated mainly at the optimization level. This is the current trend in the *process* and *production* industries.

Besides this, MBPC might also have a role to play at the *product* level: more and more of our sophisticated *mechatronic* devices require a high-performing control system to function adequately; such control system is an *integrated* part of the product and is *vital* for its operation.

2. The MBPC Principle

The following notation will be used throughout this text:

- t: denotes the discrete-time index (t = 0, 1, 2, …)
- $u(t)$: denotes the process input (manipulated variable MV);
- $y(t)$: denotes the process output (controlled variable CV);
- $w(t)$: denotes the setpoint;
- $u(t+k|t)$: denotes future values of the input, *postulated* at time t;
- $y(t+k|t)$: denotes predicted values of the output based on:
  - ⇒ measurements available at time t: $\{y(t), y(t-1), \cdots, u(t-1), u(t-2), \cdots\}$
  - ⇒ future *postulated* values of the input: $\{u(t\,|\,t), u(t+1\,|\,t), \cdots\}$.

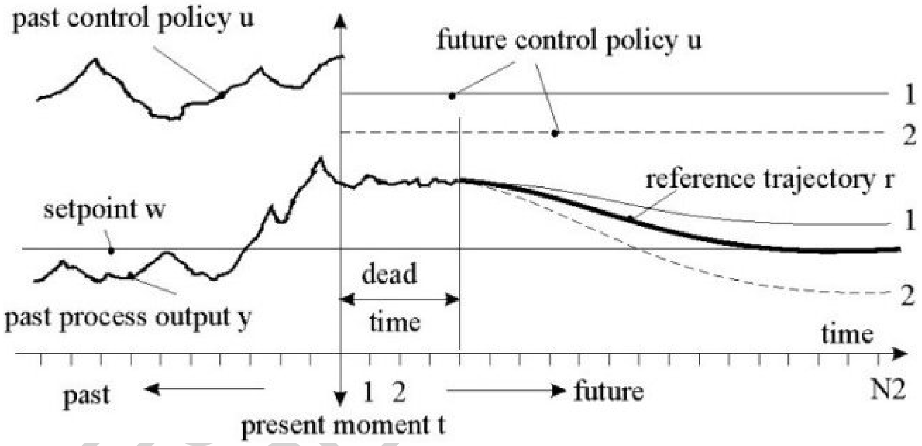

Figure 1: The MBPC Principle

Referring to Figure 1, the MBPC principle is characterized by the following strategy:

- at each ‘current’ moment t, the process output $y(t+k)$ is predicted over a time horizon k=1..N_2. The predicted values are indicated by $y(t+k|t)$ and the value N_2 is called the *prediction horizon*. The prediction is done by means of a model of the process; it is assumed that this model is available. Moreover in this text only linear models will be dealt with (see *Model Based Predictive Control*; see *Model Based Predictive Control for Nonlinear Systems*). The forecast depends on the past inputs and outputs, but also on the future control scenario $\{u(t+k\,|\,t),\;\; k = 0 \ldots N_2 - 1\}$ (i.e. the control actions that we *intend* to apply from the present moment t on);
- a *reference trajectory* $\{r(t+k\,|\,t),\;\; k = 1 \ldots N_2\}$, starting at $r(t|t)=y(t)$ and evolving towards the setpoint w, is defined over the prediction horizon, describing how we want to guide the process output from its current value $y(t)$ to

its setpoint *w*; in case the process has a time-delay (dead-time), it is reasonable to start the reference trajectory after the time-delay (ref. Figure 1);

- the *control vector* $\{u(t+k\,|\,t),\ k=0...N_2-1\}$ is calculated in order to minimize a specified *cost function*, depending on the predicted control errors$\{[r(t+k\,|\,t)-y(t+k\,|\,t)],\ k=1...N_2\}$; also, in most methods there is some *structuring* of the future control law $\{u(t+k\,|\,t),\ k=0...N_2-1\}$ and there might also be *constraints* on the process variables; these are all *very important concepts* of MBPC which will be explained later on;
- the 1st element *u*(*t/t*) of the optimal control vector $\{u(t+k\,|\,t),\ k=0...N_2-1\}$ is *actually* applied to the real process. All other elements of the calculated control vector can be forgotten, because at the next sampling instant all time-sequences are shifted, a new output measurement $y(t+1)$ is obtained and the whole procedure is repeated; this leads to a new control input *u*(*t*+1|*t*+1), which is generally different from the previously calculated *u*(*t*+1|*t*); this approach is called the *'receding horizon'* principle.

In above strategy, some important elements characterizing MBPC can be recognized:

- prediction by means of a *process model;*
- specification of a *reference trajectory*;
- *structuring* of the future (postulated) control law;
- definition of a *cost function* (and *constraints*);
- calculation of the *optimizing* control scenario.

The MBPC strategy can be visualized in the block scheme of Figure 2:

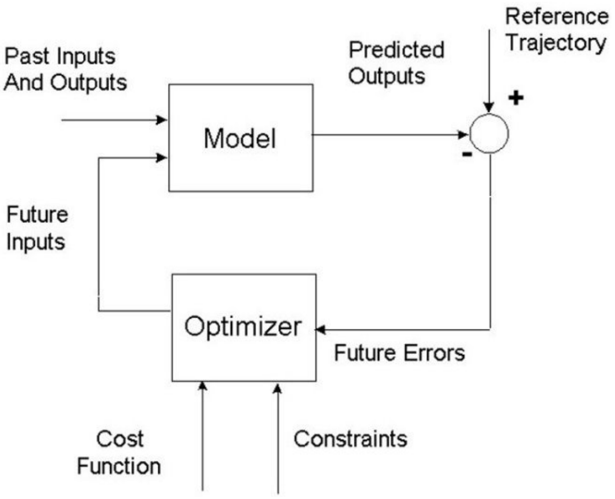

Figure 2: MBPC Strategy as a Block Scheme

3. SISO MBPC

3.1. The Process Model

The process is modeled as

$$y(t) = x(t) + n(t) \tag{1}$$

which is illustrated in Figure 3, with:

- $y(t)$: (measured) process output
- $u(t)$: process input
- $x(t)$: model output
- $n(t)$: process/model disturbance

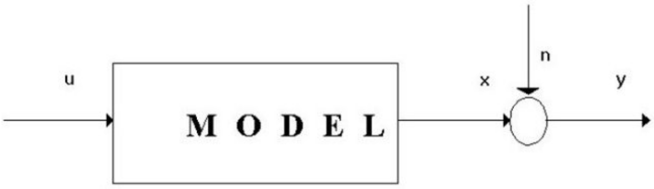

Figure 3: Process Model

In this section of the chapter, the control strategy is presented for a SISO-process (**S**ingle **I**nput **S**ingle **O**utput), i.e. only 1 measured output and 1 control input. In Section 5, the extension to a MIMO-process (**M**ultiple **I**nput **M**ultiple **O**utput) will be presented.

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. It includes effects of process disturbances, of other (unmodeled) process inputs, of measurement noise, of model errors, etc. The net effect of all these unknown disturbances has a stochastic character, generally with *non-zero average* value. It can be modeled by a colored noise process:

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

with: $$\begin{cases} e(t): \text{white noise (uncorrelated noise with zero mean value)} \\ C(q^{-1}) = 1 + c_1 q^{-1} + ... + c_{n_c} q^{-n_c} \\ D(q^{-1}) = 1 + d_1 q^{-1} + ... + d_{n_d} q^{-n_d} \end{cases}$$

where q^{-1} is the backward shift operator, i.e. $q^{-n}s(t) = s(t-n)$, $s(t)$ being a time-dependent signal and t denoting the discrete-time index (t=0,1,2,...). The filter $C(q^{-1})/D(q^{-1})$ is the *disturbance model*.

It is common practice in the MBPC approach to consider this disturbance model as a *design* filter. A smart choice of this filter leads to:

- elimination of non-zero average disturbances (if $D(q^{-1})$ contains the factor Δ=1-q^{-1});
- suppressing of disturbances in a specific frequency range;
- increasing of robustness against modeling errors.

A popular (but sub-optimal) 'default' structure for the disturbance model is:

$$\frac{C(q^{-1})}{D(q^{-1})}=\frac{1}{1-q^{-1}}\,.$$

The model output $x(t)$ represents the effect of the process input $u(t)$ on the process output $y(t)$. This is again a non-measurable signal, as only the combined effect of (process input + disturbances) is measurable.

The relationship between $u(t)$ and $x(t)$ is a *dynamic* relationship, i.e. the current output $x(t)$ does not depend on the current input $u(t)$ but on the previous inputs $\{u(t-1),u(t-2),\cdots\}$ and on the previous outputs $\{x(t-1),x(t-2),\cdots\}$. Moreover this relationship can be described either by a *linear* or by a *nonlinear* dynamic model (e.g. a neural network). Most real-life processes are nonlinear, but fortunately many of them can be locally linearized.

In general, the effect $u(t)\Rightarrow x(t)$ can thus be represented by the *generic dynamic model:*

$$x(t)=\mathrm{f}\left[x(t-1),x(t-2),\cdots,u(t-1),u(t-2),\cdots\right] \qquad (3)$$

where f[...] is supposed to represent a *known* (possibly nonlinear) function. If the process model in a specific application is unknown, it has to be identified as the first step of the MBPC design (see *Modeling, Identification and Parameter Estimation*). In the case of a time-varying process, the process model has to be estimated in real-time (see *Adaptive Predictive Control*). The EPSAC-MBPC approach described in this text can deal with linear as well as with nonlinear systems. However essentially the *linear* version will be presented here, with a short extension to the nonlinear case. A more detailed treatment of nonlinear MBPC is not the topic of this chapter (see *Model Based Predictive Control for Nonlinear Systems*).

This means that the function f[…] is supposed from now on (except in Section 4.3) to be linear in {$x(t$-1), $x(t$-2), …, $u(t$-1), $u(t$-2), …}. Popular *linear* models are the transfer function and the linear state space model (see *General Models of Dynamic Systems*). As an example, consider the transfer function model:

$$x(t)=\frac{B(q^{-1})}{A(q^{-1})}u(t)$$

or

$$x(t)=-a_1x(t-1)-a_2x(t-2)-....-a_{n_a}x(t-n_a)+b_1u(t-1)+b_2u(t-2)+... \qquad (4)$$

Using the system model (4) and the disturbance model (2) in the generic process model (1) leads - for the special case of $D(q^{-1})=(1-q^{-1})A(q^{-1})$ - to the popular CARIMA-model:

$$A(q^{-1})y(t) = B(q^{-1})u(t) + \frac{C(q^{-1})}{1-q^{-1}}e(t) \qquad (5)$$

3.2. The EPSAC Approach to MBPC

3.2.1. The Multistep Predictor

As explained in Section 2, a fundamental step in the MBPC methodology consists of the prediction of the process output $\{y(t+k\,|\,t),\ \ k=1\ldots N_2\}$ based on:

- measurements available at time *t*: $\{y(t), y(t-1),\cdots,u(t-1),u(t-2),\cdots\}$
- future (*postulated*) values of the input: $\{u(t\,|\,t),u(t+1\,|\,t),\cdots\}$.

In the GPC approach to MBPC, the CARIMA model is used and a *k-step-ahead* prediction problem is solved using *Diophantine* equations (see *Adaptive Predictive Control*). In the EPSAC approach, the generic model (1)(2)(3) introduced in Section 3.1 is used, and a *multistep* prediction problem is solved using *filtering* techniques.

Note: If the CARIMA model (5) is used as a special case of the generic model (1)(2)(3), it can be proven mathematically that both approaches - although fundamentally different from the implementation point of view - lead to exactly the same results.

Shifting the process model (1) in time gives:

$$y(t+k\,|\,t) = x(t+k\,|\,t) + n(t+k\,|\,t) \qquad (6)$$

Prediction of *x*(*t*+*k*|*t*)

This can be done by recursion of the system model (3). There are 2 possible implementation configurations, illustrated for a 3rd order model in Figure 4 and in Figure 5:

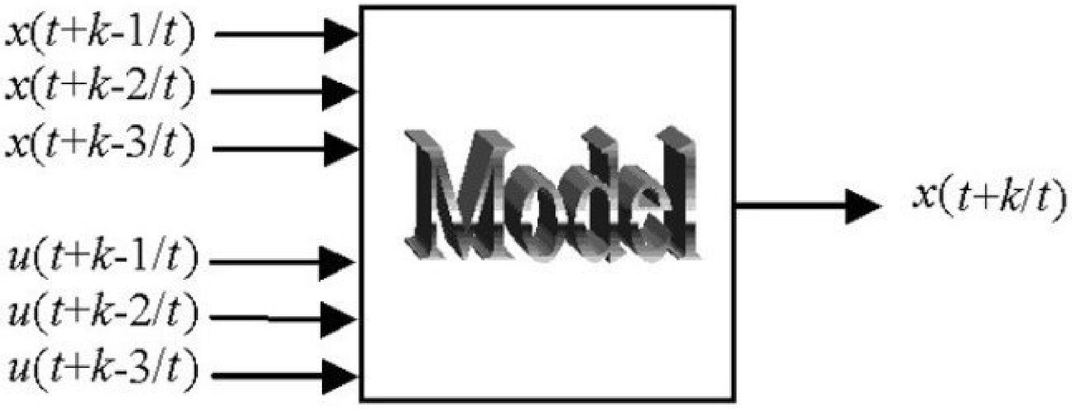

Figure 4: Parallel Model

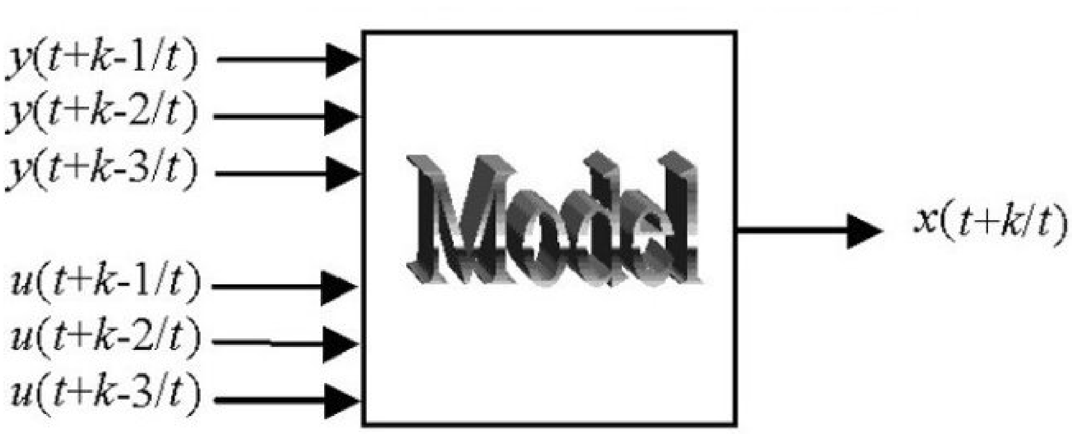

Figure 5: Series-Parallel Model

The *parallel* model (also called *independent* model) can only be used for stable processes. The *series-parallel* model (also called *realigned* model) can also be used for unstable processes. The difference between both implementation structures is illustrated in Figure 6.

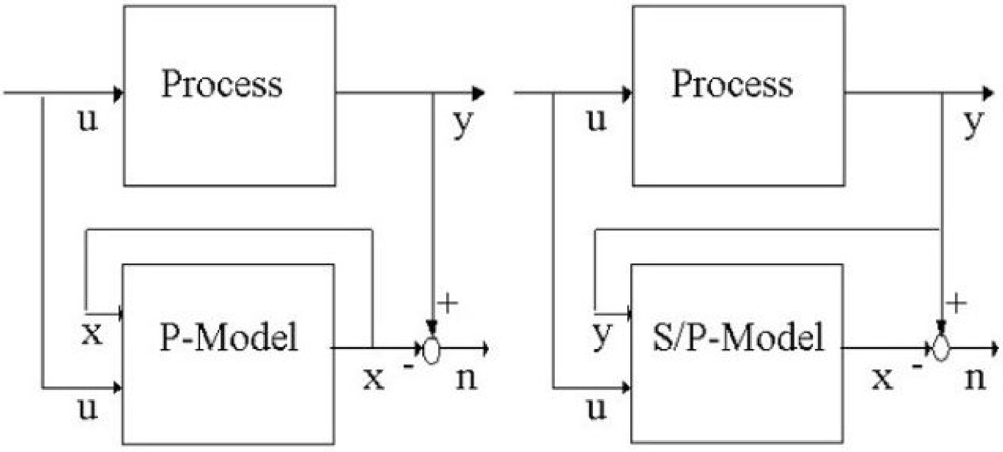

Figure 6: P and S/P Model Implementations

Notice that it can easily be shown mathematically that, for *linear* systems, both implementation forms are equivalent if the disturbance model $C(q^{-1})/D(q^{-1})$ is chosen appropriately. For the sequel we assume a P-model, but the following strategy is completely similar for a S/P-model.

At *each* sampling instant t, the recursion is started with k=0 and $x(t|t)$ is computed using the model input vector $\left[x(t-1)\,x(t-2)\,x(t-3)\cdots u(t-1)\,u(t-2)\,u(t-3)\cdots\right]$, which contains values from the past, thus known at time t (available in the computer database). Notice that $x(t)\equiv x(t\,|\,t)$ and that this value has to be saved in the database for further use at the next sampling instants. Then for k=1, the previously computed $x(t|t)$ is used at the model input to compute $x(t+1|t)$, etc.

Prediction of $n(t+k|t)$**:**

At time *t*, *x*(*t*) can be calculated as described before. Using the measured process output *y*(*t*), it is then possible to compute the current value of the disturbance *n*(*t*) with the generic process model (1): $n(t) = y(t) - x(t)$. Notice that the previous values *n*(*t*-1), *n*(*t*-2), ... are also available in the database, as they have been computed at the previous time instants.

Compute the filtered disturbance signal

$$n_{\mathrm{f}}(t) = \frac{D(q^{-1})}{C(q^{-1})} n(t) \text{ or}$$

$$n_{\mathrm{f}}(t) = -c_1 n_{\mathrm{f}}(t-1) - c_2 n_{\mathrm{f}}(t-2) - \dots + n(t) + d_1 n(t-1) + d_2 n(t-2) + \dots \qquad (7)$$

and store the result $n_{\mathrm{f}}(t)$ in the computer database, where also the previous values $n_{\mathrm{f}}(t-1)$, $n_{\mathrm{f}}(t-2)$, … are available.

As the disturbance model (Section 3.1) is given by:

$$n(t) = \frac{C(q^{-1})}{D(q^{-1})} e(t) \qquad (8)$$

the conclusion is that the signal $n_{\mathrm{f}}(t)$ thus calculated will be a white noise with zero average value: $n_{\mathrm{f}}(t)=e(t)$. Because white noise is by definition uncorrelated, its best prediction is the mean value, thus leading to:

$$n_{\mathrm{f}}(t+k \mid t) \equiv 0, \quad k = 1 \dots N_2 \qquad (9)$$

So the best prediction of the disturbance is obtained from

$$n(t+k \mid t) = \frac{C(q^{-1})}{D(q^{-1})} n_{\mathrm{f}}(t+k \mid t) \qquad (10)$$

which can be calculated using the corresponding difference equation:

$$n(t+k \mid t) = -d_1 n(t+k-1 \mid t) - d_2 n(t+k-2 \mid t) - \ \dots \ + n_{\mathrm{f}}(t+k \mid t) + c_1 n_{\mathrm{f}}(t+k-1 \mid t) + \ \dots \qquad (11)$$

The recursion goes from *k*=1...N_2. For *k*=1, the signal values in the right hand side *n*(*t*|*t*), *n*(*t*-1|*t*), ..., $n_{\mathrm{f}}(t|t)$, $n_{\mathrm{f}}(t-1|t)$, ... are known in the database, while $n_{\mathrm{f}}(t+1|t)=0$ according to (9). The computed value *n*(*t*+1|*t*) is then used in the right hand side, together with $n_{\mathrm{f}}(t+2|t)=0$, in order to compute *n*(*t*+2|*t*), etc.

As mentioned earlier, it can be shown that this procedure, based on filtering techniques, leads to exactly the same solution as the alternative procedure using Diophantine equations (see *Adaptive Predictive Control*).

-
-
-

Bibliography

Since the development of the first ideas of predictive control around 1980, a vast amount of papers has been published on this topic, broadly accessible in the control engineering literature. For this reason, the following list has been intentionally restricted to a few papers that present the original MBPC-ideas (for historical reasons), to some papers that are directly referenced to in the text and to some important recent books.

Camacho E. and C. Bordons (1999). *Model Predictive Control*. Springer-Verlag London.
Clarke D., C. Mohtadi and P. Tuffs (1987). *Generalized Predictive Control*. Automatica **23**, 137-160 [the GPC approach to MBPC].

Clarke D. and C. Mohtadi (1989). *Properties of Generalized Predictive Control.* Automatica **25**, 859-875 [robustness issues]

Clarke D. and R. Scattolini (1991). *Constrained Receding-Horizon Predictive Control.* Proc. IEE-Pt.D **138**, 347-354 [end-point equality constraints for stability].

Clarke D. (1994). *Advances in Model-Based Predictive Control.* Oxford University Press NY.

Cutler C. and B. Ramaker (1980). *Dynamic Matrix Control – A Computer Control Algorithm*. American Control Conference San Francisco [the DMC approach to MBPC].

De Keyser R. and A. Van Cauwenberghe (1981). *A Self-Tuning Multistep Predictor Application*. Automatica **17**, 167-174 [initial EPSAC ideas].

De Keyser R. and A. Van Cauwenberghe (1985). *Extended Prediction Self-Adaptive Control*. In: IFAC Symposium on Identification, York, 1317-1322 [the EPSAC approach to MBPC].

De Keyser R., P. Van de Velde, F. Dumortier (1988). *A Comparative Study of Self-Adaptive Long-Range Predictive Control Methods*. Automatica **24**, 149-163 [an experimental comparison of some popular MBPC strategies].

De Keyser R. (1991). *Basic Principles of Model Based Predictive Control*. European Control Conference Grenoble, 1753-1758 [a survey of the MBPC methodology].

Greco C., G. Menga, E. Mosca and G. Zappa (1984). *Performance Improvement of Self-Tuning Controllers by Multistep Horizons*. Automatica **20**, 681-700 [the MUSMAR approach to MBPC].

Kouvaritakis B. and J. Rossiter (1992). *Stable Generalized Predictive Control.* Proc. IEE-Pt.D **139**, 349-362 [stability issues].

Maciejowski J. (2002). *Predictive Control with Constraints.* Prentice Hall London.

Martin Sanchez J. and J. Rodellar (1996). *Adaptive Predictive Control*. Prentice Hall London.

Morari M. and J. Lee (1999). *Model Predictive Control: Past, Present and Future*. Computers and Chemical Engineering **23**, 667-682 [survey paper].

Mosca E. and J. Zhang (1992). *Stable Redesign of Predictive Control.* Automatica **28**, 1229-1233 [stability via terminal constraints].

Mosca E. (1995). *Optimal, Predictive and Adaptive Control*. Prentice Hall London.

Rawlings J. and K. Muske (1993). *The Stability of Constrained Receding Horizon Control*. IEEE-AC **38**, 1512-1516 [stability issues].

Richalet J., A. Rault, J. Testud and J. Papon (1978). *Model Predictive Heuristic Control - Application to Industrial Processes*. Automatica **14**, 413-428 [the MAC approach to MBPC].

Richalet J. (1993). *Pratique de la Commande Prédictive.* Editions Hermès Paris.

Scokaert P. (1997). *Infinite Horizon Generalized Predictive Control*. Int. J. of Control **66**, 161-175 [stability issues].

Soeterboek A. (1992). *Predictive Control – A Unified Approach.* Prentice Hall New York.

Ydstie B. (1984). *Extended Horizon Adaptive Control*. In: IFAC World Congress, Budapest [the EHAC approach to MBPC].

Yoon T. and D. Clarke (1995). *A Reformulation of Receding-Horizon Predictive Control.* Int. J. Syst. Sci. **26**, 1383-1400 [stability and robustness issues].

Biographical Sketch

Robin De Keyser graduated in electrical-mechanical engineering and obtained a PhD-degree in control engineering from Ghent University, Belgium. He is currently full professor of control engineering at this university, teaching and guiding research in many fields of advanced control.

This includes process modeling and simulation, identification, autotuning and adaptive control, predictive control, neuro-fuzzy and self-learning control, with many applications and pilot implementations in industrial projects (particularly in chemical, steel, marine, mechatronic and semiconductor industries) as well as in non-industrial fields (agro-engineering, biomedical engineering).

Being one of the pioneers who produced the original concepts of predictive control about 20 years ago, he developed and continuously improved the EPSAC predictive control strategy and is mentioned as (co-) inventor in several patents related to this method.

He is (co-)author of about 200 papers and gives many international lectures and courses on above-mentioned topics, is involved in industrial consulting and in several professional organizations in Belgium and abroad.