We are IntechOpen, the world's leading publisher of Open Access books
Built by scientists, for scientists

6,900
Open access books available

185,000
International authors and editors

200M
Downloads

154
Countries delivered to

Our authors are among the

TOP 1%
most cited scientists

12.2%
Contributors from top 500 universities

BOOK CITATION INDEX
CLARIVATE ANALYTICS
INDEXED

WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

MBPC – Theoretical Development for Measurable Disturbances and Practical Example of Air-path in a Diesel Engine

Jose Vicente García-Ortiz
University Jaume I and Vinci Energía
Spain

1. Introduction

Model based predictive controller is a family of control algorithms suitable for some applications. They are useful for most of the real applications. But they are not used very often due to the complexity of them. So this kind of books should be closer to the real application in order to be more used. One of the main objectives, in this chapter, is to clarify the use of the MBPC for real applications.

My experience is based on real application of MBPC. Additionally they are also suitable for small modification. One of the adaptations explained in this chapter is the one related with disturbance measurement. All real systems have disturbance. Many times the disturbance can be measured and identify. The disturbance measurable can be identified but you can not use this variable as a control variable. You can include this information in the optimization process. One the examples explained in this chapter is the management of this information.

In the air management loop in a internal combustion engine use to manage VNT (Variable Nozzle turbine), EGR (Exhaust gas recirculation), SWIRL valve and others, but mass fuel injected is managed by air management loop. But this mass fuel has important influences in the air management process, so it should be considered.

The philosophy considering disturbance measurable has influences on different parts of the control algorithm development. MBPC has 4 main topics: model, control cost, optimization process and receding horizon. In this chapter the development will be explained.

Due to the high quantity of control parameters considered in the MBPC, it use to be rejected for using. In this chapter we will try to explain the influence of the most important parameters in order to show the way for tuning.

2. State of art

Control Engineering Science is a very brand new Science branch compared with other sciences. In fact the first servo system of the history was developed in 1788 by James Watt. But the strong theoretical development was done during 20th century. Additionally the control engineering science has been developed as a auxiliary tool for other real applications. One example is the development was SMC (Sliding Mode Control), which was developed to control artificial satellites.

Probably this is the main reason because, control engineering, is mainly considered as a secondary science. In this chapter MBPC will be explained, mainly the GPC (Generalized Predictive Control).

2.1 Control engineering

As explained in 1788, James Watt developed the first servo system of the history but the real development of this science was during 20th century. The first important mathematical development were done by Nyquist[27], this work studied stability of the system by means of closing the loop. One of the problems when closing the loop is the possibility of instability, that is why Nyquist developed the theory in 1932. During forties some techniques of frequency design were developed, thus control engineering were widely implemented in the Industry. During fifties Evans[14] proposed the root locus for designing algorithms with stability. From sixties the digital hardware were getting importance so digital or disserted algorithms were implemented. With discrete systems more flexible algorithms and structure were used, even thus the first approached were done by discrimination of classical controllers.

2.2 Control algorithms MBPC history

MBPC has its origin in Kalman's work. He developed an observer in which design an optimization cost is used. Additionally the optimal control based on the optimization of a quadratic cost. This control was the LQG. In comparison with Evans[14] theory, the behaviour of the controlled system is not base on system response, but it is based on the minimum cost of a quadratic cost function.

The first MBPC published was in 1976 by Richalet[33]. He formalized the development two years later in the Paper []. This algorithms is known as MPHC (Model Predictive Heuristic Control), which was commercially developed as IDCOM, this commercial software included identification, predictive controller and impulsion model. One year later Cutler and Ramaker[12] developed de DMC (Dynamic Matrix Controller) based also in the impulsion system response. Both algrothm were known as first generation MPC (Model Predictive Controllers), but neither IDCOM nor DMC were based in LQG models or state space models.

During first half of eighties the second generation of algorithms was developed. DMC with constraints were presented by García[16] by using Quadratic Programming QP in the optimization process. Connoisseur from Invensys Corporation is also an algorithm of this family.

The most important advance in MBPC were developed during second half of eighties and nineties were third generation appeared. In 1988 Keyser[20] compared some MBPC adaptative. Shell company worked in the first MBPC based in state space, its name is SMOC. An overview of MBPC was summarized by Qin[31,32], not only theoretical controllers but most of the commercial solutions were analyzed. In 1986 Clarke[8,9,10] presented the GPC or Generalized Predictive Controller. Clarke tried to develop a MBPC unifying all controllers in one algorithm. The way for getting this target is to use a general model, as general as possible, thus he decided to use the CARIMA model, this is a standard ARX plus a derivative white noise. Another proposal was done by Soeterbock[37] in which adaptive controller as general as possible. In 1990 Zafirion [42] exposed the possibility of destabilize the system by including constraints in the optimization process. This fact should be

considered, because a stable system and stable controller can be destabilized by mean of including the constraints.

Galanti[15] proposed, two years later, an interesting solution for constrained by dead zone. But Zheng[43] proposed an intelligent development for solving that problem. It consists on developing soft constraints in place of constraints. The system and controller must be under the constraints, I mean, it is something compulsory. But the soft constraint is not a constraint anymore, this is a recommendation, so id you are out the soft constraint for a while, this is not an alarm, this is a warning. Thus the stability of the system is stronger, this kind of soft constraints are suitable for outout.

In 1997 Rositer and Kouvaritakis[22] proposed the terminal constraint in order to improve the stability when constraints in MBPC. A similar work was presented in Automatica by Havlena [18] where stability of optimum control when constraints were analyzed. That year some people[36] presented studies about tuning in MBPC for SISO systems.

Two years later in 1999 Bordons and Camacho[2] summarized the state of the art in their book. They mainly focus the study in a wide development of GPC as a General proposal of MBPC.

During century XXI MIMO approaches of MBPC were developed. Nuñez[26] proposed Hierarchy control, well accepted by Industry. An additional problem when MIMO MBPC approach is carried out, is instability due to problems in matrix as Pannochia[29] suggested. Poulser and Kouvaritakis[30] exposed some real application in chemical reactor in which MBPC offer its possibilities. Moreover, Kouvaritakis[21] presented different optimization tools in place of quadratic programming.

An special case in the MBPC is the GPC. Due to special characteristics is one of the most developed controller. It was published in 1987 by Clarke, in these papers the linear controller was proposed[7,8] . Thi first idea was to find a controller with weak computational cost to be used in adaptative applications. In fact, Clarke and Gawthrop[5,6] developed the GMV or controller of minimum variance in 1975 and 1979[5,6]. GPC inherited from GVM the prediction model CARIMA. CARIMA is based in a transfer function plus a white noise derivative model. All previous cotroller use to find some problems:

1. Minimum phase zeros process can not be used.
2. Instable process controlled by open loop architecture.
3. Robustness when delays or not expected delays.
4. Robustness when poles zeros were over stimulated.

Not in all contemporaneous controller appeared that problems, but in most of them. CARIMA model include a derivative model which made this model suitable for real applications. GPC contains the main bases the MBPC.:

1. Prediction model: CARIMA model, what have some interesting proprieties.
2. Cost Index: with quadratic error in a time slot or prediction horizon plus a control horizon or steps of control action.
3. Optimization tool: at this point the GPC is using the standard optimization tool, roots in the first derivative and checking the second one is positive.
4. Mobil horizon: it mean applying the first control action and recalculate every step updating the new information.

During eighties the team of Clake were developing GPC. In 1988 Tsang[40] in this team explained the importance of using constraints. Once this point, two problems should be solved: include constraint in the optimization and improving stability in close loop.

In 1992 few paper about robustness in GPC were published. Soeterboek[38] tried to generalize the MBPC in one general book. The importance of T polynomial (noise polynomial poles) were widely explained, specially about robustness of the algorithm. Moreover the correct tuning of lambda parameter (or weight of control actions) in the GPC robustness. That year Kouvaritakis[21] published a paper about GPC Stability and Mosca[25] proposed a control solution about constraints in dead zone. This year Albertos[1] published a new formulation on the model based in Diophantine decomposition.

In 1994 a paper published by Clarke team is presented. Chow[4], the author of this paper, proposed some constraints applied to GPC which are: slew rate constraint, Coulomb friction, histeresys and dead zone by using two optimization algorithms; quadratic programming and Lagrange multipliers. In the University of Seville was presented by Camacho[3] an empirical approach of GPC. It consisted in using standard linear model (normalized) and setting up different model and different GPC every control area, thus a gain Schedule of GPC was proposed.

One of the most important problems available in Industrial application is the disturbance. GPC can carry out this problem due to the CARIMA model. In this field Beg in 1995, published an application of controlling a MIMO system by mean of SISO controllers and feed forward architecture.

In 1996 an approach with heuristic algorithm, genetic algorithms as a optimization tool in GPC scheme. This proposal was developed by CPOH[24], a R+D team in the UPV (Politecnical University of Valencia). In 1997 Corradini[11] proposed as hybrid solution between SMC (Sliding mode control) and GPC, in order to improve the stability.

During this century the papers in this field are focused in real application if MBPC and GPC, in particular, or Industrial applications. The team of Seville University published a high level GPC MIMO controller and PID controllers in low level, the same was done by Kennel[19]. Salcedo[34] from CPOH, published a paper in which CARIMA model is changed by state space CARIMA model. Thus the algorithm is faster, the computational effort is lower and less storage information is needed, but an state observer must be used.

In 2002 Perez de la Parte[13] from Seville, tried to improve the robustness by the same way as Corradini. Sanchis[35] used another solution for improving the robustness, he proposed the use of singular values decomposition SVD.

2.3 Planning

When MBPC is selected to be the controller for a system, some steps must be done. Firstly the system must be studied. Secondly the model system must be obtained. Thirdly the control must be designed. Fourthly the control should be implemented and tuned.

The system must be studied. We have to discover the main variables, inputs, outputs and disturbance. Moreover the main dynamics and behavior of the system should be considered.

Getting the model of the system should be done. The model is one of the key in the MBPC. Getting a good quality model is sometimes the difference between success and failure.

After that, control architecture and design should be done. Finally tuning the algorithm and checking parameters as robustness or others must be done.

3. Model based predictive controller

In this section the control algorithms will be developed. MBPC are based in 4 different beams. One of the most important is the system model. Other important tool is the optimization process.

Many time the system has a, more or less, linear behaviour. At this time a linear model uses to be a good approach. Linea model has the main advantages of the small computational cost. Specially when constraint are not considered. The optimization can be carried out off-line. Thus the control action is a combination of multiplications and additions. So the computational effort into a microcontroller is very small.

Sometimes the system has a nonlinear behaviour. Then linear approach can not be carried out by its selves. A non-linear model with constraints use to cost the highest computational effort, but with a high accurate control behaviour.

Sometimes linear approaches in gaps use to be an intelligent solution. It consists on using different linear models around linear system behaviour, and changing the model when the system is out of its gap of influence of that model. Thus an offline optimization can be carried out and the computational effort, during the control action, is very low. It looks like gain schedule but, in place of PID parameters, GPC controllers are used.

Thus a standard GPC controller is not going to be developed because it can be obtained by simplifying the GPC with disturbance measurement.

The system use to be influenced by some parameters. Those parameters many times could be considered as measurable disturbance. A measurable disturbance is an input of the system that is not controlled by the control algorithm. For instance, in a Diesel engine, the air management loop control VNT (variable nozzle turbine) and the EGR system. Mass fuel injected is controlled by the user. But the mass fuel injected in the engine affects quite a lot in the air management behaviour. Thus this important input has to be considered as measurable disturbance.

These kinds of inputs need a special management in the algorithm. So this chapter shows one way to consider them into the algorithm, for past disturbance and also for future disturbances. Two different algorithms will be developed with this approach, the one based in the linear Markov parameters and the GPC (Generalized Predictive Controller) developed by Clarke.

3.1 GPCDM, algorithm development

We now briefly describe the control algorithm which is key issues of this paper. The control algorithm was based on the model shown in the Figure 1 in which there is the CARIMA model plus an input p which is a measurable disturbance. To simplify the development one input and one output are considered.
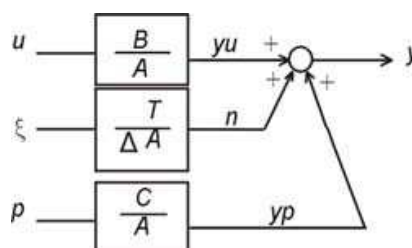

Fig. 1. CARIMA model proposed.

$$A \cdot y(z) = B \cdot u(k) + C \cdot p(k) + \frac{T}{\Delta} \cdot \xi(k)$$

$$y(k+i) = y_u(k+i) + y_P(k+i) + n(k+i) \tag{1}$$

Equation (1) is the discrete one of the model shown in figure X. The yu is the output behaviour depending on the direct input. yp is output behaviour from measurable disturbance and n is the behaviour depending on few things like non-linearity, others inputs etc.

$$n(k+i \mid k) = \frac{T}{\Delta \cdot A} \cdot \xi(k+i) \tag{2}$$

n(k+i|k) is the n prediction of (k+i) with the information in k sample time. Diophantine decomposition is applied in order to get the system response due to pass inputs and future inputs in different terms. Moreover the mean of ξ is zero, then:

$$n(k+i \mid k) = \xi(k+i) \cdot \frac{T(z^{-1})}{\Delta \cdot A} = E_i \cdot \xi(k+i) + \frac{F_i}{\Delta \cdot A} \cdot \xi(k) \tag{3}$$

and n(k+i|k) is at the same time:

$$n(k+i \mid k) = \frac{F_i}{T}[y(k) - y_u(k) - y_p(k)] \tag{4}$$

We do the same with y as a result we have:

$$y(k+i \mid k) = \Delta u(k+i-1) \cdot \frac{B \cdot E_i}{T} + \Delta p(k+i-1) \cdot \frac{C \cdot E_i}{T} + \frac{F_i}{T} \cdot y(k) \tag{5}$$

$$\begin{aligned} G_i &= B \cdot E_i \\ GP_i &= C \cdot E_i \end{aligned} \tag{6}$$

We can apply Diophantine decomposition to equation (6) and finally we can get:

$$\begin{aligned} \frac{G_i}{T} &= G_i^| + \frac{\Gamma_i \cdot z^{-1}}{T} \\ \frac{GP_i}{T} &= GP_i^| + \frac{\Gamma P_i \cdot z^{-1}}{T} \end{aligned} \tag{7}$$

With that we can develop the equation (5) for future prediction and sum up all terms in an equation

$$y = G \cdot u + GP \cdot p + \Gamma \cdot u^f + \Gamma P \cdot p^f + F \cdot y^f \tag{8}.$$

Where: y is the predicted system output vector; model matrices are represented by G, Γ, ΓP, GP, and F (G and GP represent the future response; and Γ, ΓP and F represent the past system response or natural response); Δ is the incremental operator $(1-z^{-1})$; u is the vector of future control action increment; p is the disturbance increment vector; the past action control

filtered by $1/T$ is u^f ; the past disturbance filtered by $1/T$ is p^f ; the past system output filtered by $1/T$ is y^f.

Equation (8) is the system model in a matrix form. More over we need the control cost which depend on error cost and control effort, represented in (9) as a matrix equation. Where y_{sp} is a vector of future set points, λ is a diagonal matrix of control effort factor and α is the diagonal matrix of error cost.

$$J = (y - y_{sp}) \cdot \alpha \cdot (y - y_{sp}) + u^T \cdot \lambda \cdot u \tag{9}$$

We put the model in the control cost, then using an optimization tool, like (10).

$$\frac{\partial J}{\partial \Delta u} = 0 \tag{10}$$

Eventually the result is given in next equation (11)

$$\bar{u} = (G^T \cdot \alpha \cdot G + \lambda)^{-1} \cdot G^T \cdot \alpha \cdot (y_{sp} - \Gamma \cdot u^f - \Gamma P \cdot p^f - GP \cdot p - F \cdot y^f) \tag{11}$$

It is also possible to calculate the first value of equation (11) and using it like a combination of discrete time transfer functions. More details can be studied in Garcia-Ortiz[17] where the algorithm is widely explained and Camacho & Bordons[3].

There are other ways to reach the same result. In terms of model in figure 1, may be considers ad a MISO system (Multiple input single output). Getting ecuations from MIMO CARIMA developed models we can find:

$$y = G \cdot u + \Gamma \cdot u^f + F \cdot y^f \tag{12}$$

Where MISO system is:

$$y = (G_{11} \quad G_{12}) \cdot \begin{pmatrix} \Delta u_1 \\ \Delta u_2 \end{pmatrix} + (\Gamma_{11} \quad \Gamma_{12}) \cdot \begin{pmatrix} u_1^f \\ u_2^f \end{pmatrix} + F_1 \cdot y^f \tag{13}$$

Thus G_{11} is the G in equation (8) and G_{12} is GP in equation (8). So is very easy to see that both ways get the same result.

3.2 DMCDM, algorithms development

A similar approach can be developed for DMC controller. The differences are that prediction model is based in Markov coefficients.
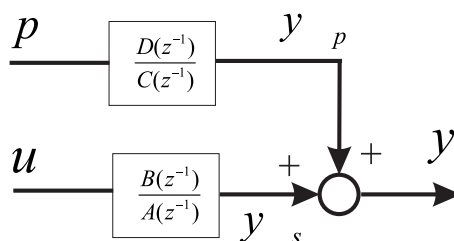

Fig. 2. Box Jenkins model. Model based for impulse response with measurable disturbances.

$$y(k) = \sum_{i=1}^{\infty} g_i \cdot u_{k-i} \tag{14}$$

Where g0 = 0 due to the delay in the discrete systems and gN = 0. From gN to gN+M=0 whatever M. from this point all systems controlled by DMC should be type 0. All systems must be stable and integer term is not allowed. Most of the industrial processes have these characteristic, which is why this controller have been developed widely in the industrial applications.

$$y(k) = \sum_{i=1}^{\infty} s_i \cdot \Delta u_{k-i} \tag{15}$$

Thus there are a equivalence between impulsion coefficients and step ones.

$$s_i = \sum_{j=1}^{i} g_i \tag{16}$$

So during the development both models will be used, depending the interest every time. Additionally some information will be defined:

P: prediction horizon. Number of time steps will be considered in the cost index and the optimization process.

M: Control Horizon. How many control actions are allowed in the optimizer to reach the target. From M to the future no movements in the control action are considered.

N: number of coefficients with information in the impulse response. From N to the future g coefficient is zero.

All that information will be used in the control cost(16).

$$J = \sum_{i=1}^{P} \psi_i^2 \cdot [y(k+i \mid k) - y_{SP}(k+i)]^2 + \sum_{i=0}^{M-1} R_{i+1}^2 \cdot (\Delta u(k+i))^2 \tag{16}$$

Where y(k+i|k) is the value of the prediction for the step k+i, calculated in the instant k with the information available in that time k. ysp(k+i) is the set point in the step k+i (when future set point is not available, this variable is considered as the set point in k. $\Psi2$ is the weight of error cost and R2,i+1 is the weight of the control effort. $\Delta u(k+i)$ is the increment of the control action in time=k+i.

From equation (15) we obtain:

$$y(k) = \sum_{i=0}^{\infty} s_i \cdot \Delta u(k-i) = \sum_{i=0}^{N} s_i \cdot \Delta u(k-i) + s_N \cdot \Delta u(k-N-1) \tag{17}$$

Thus we are ready to get the prediction model in the DMC. The main target is getting the future response of the system from one step time and the behaviour if the system when future control actions will be applied. The model will contain the system natural response, the one due to the past control actions applied in the system (this one is not changeable, because is produced) and the forced response, the one due to future control action (this one can be different because will be decided by the controller).

Once the bases are fixed we will start with the model development, thus we will get the system response in a generic step k+t, from the information available in k.

$$
\begin{aligned}
+y(k+1) &= g_1 \cdot u(k) + g_2 \cdot u(k) + \cdots + g_N \cdot u(k-(N-1)) \\
-y(k) &= g_1 \cdot u(k-1) + g_2 \cdot u(k-2) + \cdots + g_N \cdot u(k-N) \\
\hline
y(k+1 \mid k) &= g_1 \cdot \Delta u(k) + g_2 \cdot \Delta u(k-1) + \cdots + g_N \cdot \Delta u(k-(N-1)) + y(k)
\end{aligned}
\tag{18}
$$

In the second iteration we can obtain:

$$
\begin{aligned}
y(k+2 \mid k) &= g_1 \cdot u(k) + g_2 \cdot u(k) + \cdots + g_N \cdot u(k-(N-2)) \\
y(k+1) &= g_1 \cdot u(k) + g_2 \cdot u(k-1) + \cdots + g_N \cdot u(k-(N-1)) \\
y(k+2 \mid k) &= g_1 \cdot \Delta u(k+1) + g_2 \cdot \Delta u(k) + \cdots + g_N \cdot \Delta u(k-(N-2)) + y(k+1) \\
y(k+2 \mid k) &= g_1 \cdot \Delta u(k+1) + (g_1+g_2) \cdot \Delta u(k) + (g_2+g_3) \cdot \Delta u(k-1) + \cdots \\
&\quad \cdots + (g_{N-1}+g_N) \cdot \Delta u(k-(N-2)) + g_N \cdot \Delta u(k-(N-1)) + y(k)
\end{aligned}
\tag{19}
$$

By deducting generic element:

$$
\begin{aligned}
y(k+t \mid k) &= g_1 \cdot u(k+t-1) + g_2 \cdot u(k+t-2) + \cdots + g_N \cdot u(k+t-N) \\
y(k+t-1 \mid k) &= g_1 \cdot u(k+t-2) + g_2 \cdot u(k+t-3) + \cdots + g_N \cdot u(k+t-N-1) \\
y(k+t \mid k) &= g_1 \cdot \Delta u(k+t-1) + g_2 \cdot \Delta u(k+t-2) + \cdots \\
&\quad \cdots + g_N \cdot \Delta u(k+t-N) + y(k+t-1)
\end{aligned}
\tag{20}
$$

Getting all the information and summarizing the data in a matrix ecuation:

$$
\begin{bmatrix}
y(k+1) \\
y(k+2) \\
\vdots \\
y(k+M) \\
\vdots \\
y(k+P)
\end{bmatrix}
= S_f \cdot
\begin{bmatrix}
\Delta u(k) \\
\Delta u(k+1) \\
\vdots \\
\Delta u(k+M-1)
\end{bmatrix}
+ S_p \cdot
\begin{bmatrix}
\Delta u(k-1) \\
\Delta u(k+2) \\
\vdots \\
\Delta u(k-(N-1))
\end{bmatrix}
+ y(k) \cdot L_p
\tag{21}
$$

Where $L_P = \begin{pmatrix} 1 & \cdots & 1 \end{pmatrix}^T_{Px1}$ and:

$$
S_f =
\begin{pmatrix}
s_{\min(1,N)} & 0 & \cdots & 0 \\
s_{\min(2,N)} & s_{\min(1,N)} & \cdots & 0 \\
\vdots & \ddots & \ddots & 0 \\
s_{\min(M,N)} & s_{\min(M-1,N)} & \cdots & s_{\min(1,N)} \\
\vdots & \vdots & \ddots & \vdots \\
s_{\min(P,N)} & s_{\min(P-1,N)} & \cdots & s_{\min(N+P-1,N)}
\end{pmatrix}_{Px(N-1)}
\qquad \text{Matrix Sf}
$$

$$
S_P =
\begin{pmatrix}
s_{\min(2,N)} - s_1 & \cdots & s_{\min(N,N)} - s_{N-1} \\
s_{\min(3,N)} - s_1 & \cdots & s_{\min(N+1,N)} - s_{N-1} \\
\vdots & \ddots & \vdots \\
s_{\min(P+1,N)} - s_1 & \cdots & s_{\min(N+P-1,N)} - s_{N-1}
\end{pmatrix}_{Px(N-1)}
\qquad \text{Matrix Sp}
$$

At this time we are able to get future response from the data available in the step k, but we have to include system information every step time so: d(k) = yreal(k) - y(k), this is the difference between real system and prediction.

Finally yf is an array of output system predictions. Δuf is an array of future increment control actions. Δup is an array of past control actions. Sf is a matrix with dynamic future system response or forced response matrix and Sp is a matrix with past dynamic system response or natural response matrix.

$$y_f = S_f \cdot \Delta u_f + S_P \cdot \Delta u_P + y_{real}(k) \cdot L_P \tag{22}$$

Applying this prediction model to the control cost:

$$J = (y_f - y_{sp})^T \cdot \psi^T \cdot \psi \cdot (y_f - y_{sp}) + \Delta u_f^T \cdot R^T \cdot R \cdot \Delta u_f \tag{23}$$

$$J = ((S_f \cdot \Delta u_f + S_P \cdot \Delta u_P + y_{real}(k) \cdot L_P) - y_{sp}) \cdot \psi^T \cdot \psi \cdot$$
$$((S_f \cdot \Delta u_f + S_P \cdot \Delta u_P + y_{real}(k) \cdot L_P) - y_{sp}) + \Delta u_f^T \cdot R^T \cdot R \cdot \Delta u_f \tag{24}$$

The optimization tool can be used at this point. An analytical optimization is applied by getting dJ/dΔuf and second derivative is positive defined.

$$\frac{\partial J}{\partial \Delta u_f} = 2 \cdot S_f^T \cdot \psi^T \cdot \psi \cdot (S_f \cdot \Delta u_f + S_P \cdot \Delta u_P + y_{real}(k) \cdot L_P - y_{sp}) + R^T \cdot R \cdot \Delta u_f = 0 \tag{25}$$

$$\frac{\partial J}{\partial \Delta u_f} = 2 \cdot S_f^T \cdot \psi^T \cdot \psi \cdot S_f \tag{26}$$

As a result Δuf optimal is:

$$\Delta u_f = (S_f^T \cdot \psi^T \cdot \psi \cdot S_f + R^T \cdot R)^{-1} \cdot S_f^T \cdot \psi^T \cdot \psi \cdot (y_{sp} - S_P \cdot \Delta u_P - y_{real}(k) \cdot L_P) \tag{27}$$

And simplifying, H Matrix is defined as follow:

$$H = (S_f^T \cdot \psi^T \cdot \psi \cdot S_f + R^T \cdot R)^{-1} \cdot S_f^T \cdot \psi^T \cdot \psi \qquad (\textbf{\textit{H}} \text{ matrix})$$

Once the standard DMC is developed and considering figure 2:

$$y_{fs} = S_{fs} \cdot \Delta u_f + S_{Ps} \cdot \Delta u_P + y_{sreal}(k) \cdot L_P$$
$$y_{fp} = S_{fp} \cdot \Delta p_f + S_{Pp} \cdot \Delta p_P + y_{preal}(k) \cdot L_P \tag{28}$$

Where:

y_{fs} is the array of y_s predictions

y_{fp} is the array of predictions in disturbance output component.

S_{fs} and S_{fp} are matrixes of future system response and disturbance response respectively.

S_{ps} and S_{pp} are matrixes of past system response and disturbance response respectively.

ysreal is the system component of the output

y_{preal} is the disturbance component of the output

Δu_f is the future increment in control action.

Δp_f is the future increment in disturbance.
Δu_p is the past increment in control action.
Δp_f is the past increment in disturbance.
Thus system predictions are:

$$y_f = S_{fs} \cdot \Delta u_f + S_{Ps} \cdot \Delta u_P + S_{fp} \cdot \Delta p_f + S_{Pp} \cdot \Delta p_P + (y_{preal}(k) + y_{sreal}(k)) \cdot L_P \qquad (29)$$

Using equation (23) and (29); and using the optimization tool, the result is:

$$\Delta u_f = (S_{fs}^T \cdot \psi^T \cdot \psi \cdot S_{fs} + R^T \cdot R)^{-1} \cdot S_{fs}^T \cdot \psi^T \cdot \psi \cdot$$
$$(y_{sp} - S_{Ps} \cdot \Delta u_P - S_{Pp} \cdot \Delta p_P - y_{real}(k) \cdot L_P) \qquad (30)$$

Thus H is the same as (H Matrix) but, intead of S_f is S_{fs}.
Every step time the controller is applying the first control action and update the data for getting new optimal control every step time. Thus only the first line in the matrix equation should be calculated. Additionally this matrix equation, first line, can be presented as a Z transform function transfer. This is the application of the fourth basis of MBPC, receding horizon policy. It means that every step time a new control action will be calculated by means of updating data in the algorithm. Thus the control algorithm must be presented as:
In which h(z) is the first row of the H Matrix. The vector h(z) has different coefficients which will be multiplied by future set point, if the information is available. If the future set point is not available, a constant is calculated by adding all the h(z) vector coefficients.
D(z1) polynomial in which the coefficients are the result by multiplying h(z)·Sp. This array is in z-1, because is using past information. All in all the algorithms structure is shown in figure 5.3.
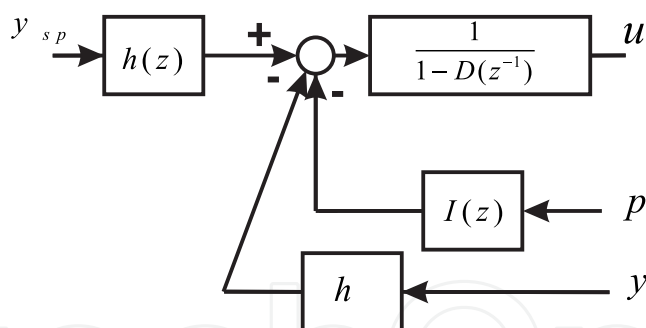

Fig. 3. DMCDM architecture with receding horizon.

When I(z-1) is the convolution of two vectors: one array is product of h(z)·Spp, the second array is Δ operator. Spp is new matrix calculated as Sp but using Markov coefficients of disturbance of the system. More information is available at Thesis of Garcia[16].

4. Practical set up and experimental behavior

This chapter is presenting the author's experience for tuning the algorithm. Also the steps recommended for carry out a control solution. MBPC is a family on controller very suitable for controlling systems in different conditions. They can consider constraints, control cost, future information, model behavior etc... Getting the model by linearization should be considered. There are many identification algorithms, using the right ones and using the best parameters can bring you success or failure.

4.1 System identification

In control engineering ther are two important concepts that are opposite: Robustness and Performance. Usually as much Robust is a controller as less Performance it offers you. So as much Performance we demand as much easy to reach instable zones. There is one way to get both characteristic at the same time, Model accuracy. If we have a better model or a more accurate model, we will be able to demand more performance and more Robustness. So the system model is one of the most important topics in control engineering.

The most famous PID control tuning techniques are Ziegler-Nochols rules or Cohen-Coon method. Methods have a main characteristic, you can tune a PID controller with Robustness, but unfortunately performance is not good enough. The main reason for is that you have not a system model. So the quality model is very poor, thus Robustness is more important than performance. When Performance is important, model quality must be improved. As good is the model as good will be the control.

By other hand, the easiest way to get a model is by mean of physical equations. If we know the physics of the system, by using physical equation we can obtain a very good quality model. When physical equations are used model reach an important complexity, sometimes the computational effort done by the microcontroller or commuter is very high. Some models need mode than two day for calculating few second of real physical behavior.

An important approach use to be linear model. The computational cost of linear model, use to be very low (few micro seconds every step time). Additionally much real system use to have a linear behavior, so linear approach, sometimes, is the most intelligent solution. When system has a non-linear behavior other solutions can be approached. Linearization is a good solution for that.

When physical equations are hard to compute, linearization of them bring us a model easy to carry out and fast to be calculated. Another solution is linear model by identification techniques. When identification is our choice this kind of model works very well in the identified point and around of it, but we have problems when the operating point is too far from the identification point. When this problem appears a new linear model can be identified. We can do this topic as many times as we need. Thus we apply a technique similar to gain schedule but with models.

The main advantages of liner models are:

It is a simpler, so you have an easier solution for solving Differential equations.

The system behavior can be observed.

Any kind of order can be used. We should fit the order to the one system expected.

As disadvantages we can find:

The solution is exact, only in the operating pint and good enough close to it, but it is not good as far as we are.

Experimental data is needed and sometimes particular experiments are not possible to carry out.

No physical sense in poles and zeros identified can be found.

Math of the transfer functions, state space model and other can be studied in Ogata continuous , Ogata Discrete or Zhu [43] books. Additionally some identification algorithms were tested and studied widely in bibliography.

Lineal model available in bibliography:

Markov coefficients or impulse response model, there is an equivalent model, the step response.

ARX: autoregressive with exogenous variable. ARMAX: Autoregressive moving average with exogenous variable. OE: Output error. BJ: box Jenkins. CARIMA or ARIMAX:

autoregressive integral moving average with exogenous variable. State space models deterministic, advanced and stochastic ones etc. all models are descried in Garcia [17].
Finally the solution proposed is linear models by sections. When linear model is valid, then it shall be used, when linear model is not valid a new linear model must be placed.

4.1.1 Experimental data for identification

Many times data available is not a suitable data for identification. That is why system is working and particular experiments are not allowed. Thus a measurement of system behavior should be done and transient data must be used in the identification. Steady data does not include important dynamic information, so transient contains the system dynamics.

When particular experiments can be done, they must be well chosen. An easy experiment use to be a step applied to the system. The step should contain some characteristics: powerful enough, long enough, the response must be in the linear zone. Theoretically white noise is the best signal. But some authors, as Luo[22] or Zhu[43], studied physical inputs and pseudo aleatori input is the best choice.

Some times a simple step or impulse can be applied and Strejc, Csypkin methods can be applied. Simple model is obtained and some time it is good enough. Frequency identification methods are also available, based in bode diagram or graphical techniques.

Identification algorithms are proposed in this chapter. Experimental data must be pre processed and model structure should be well chosen. Parametrical and state space identification can be carried out.

When parametrical identification is proposed the following steps should be respected:
1. Experimental identification design and data acquisition.
2. Data pre processed: filtering, remove data tends, data normalization etc.
3. definition of model structure.
4. Identify the model by using an appropriate identification algorithm.
5. Studying model proprieties (zeros, poles, gain, …) and model validation.
6. If the model is the expected one, this task is finished, else come back to step 1, 2 or 3, depending on problems detected.

4.1.2 Identification algorithms

The identification algorithms proposed and tested in this work is the PEM (Prediction error method). Other where tested N4SID (Numerical Algorithm for Sub Space State System Identification) widely explained in [108,141,150]. Bud PEM is preferred by this author with very well results. PEM is an identification algorithm of state space model, so this model should be converted in a equivalent transfer function.

4.1.3 System identification

A Diesel engine is identified. Some experimental data is shown. At this studied case air path or air management system is going to be controlled. If suitable linear models cannot be found, algorithms based on the Hamertein and Wiener models [12] can be used or non linear models. The disadvantages of this latter group are: high computational cost and the difficulty of programming them in a commercial ECU (electronic control unite).

The system air-path in a Diesel engine consists on controlling VNT (variable nozzle turbine) in order to keep boost pressure in the set point. The engine controlled is a Diesel engine of heavy duty applications, see figure 4. This is a truck engine, where EGR system is not

included but VNT is available. Air-path is influenced by mass fuel injected, as much fuel injected into cylinder, more power available in the turbine and more boost pressure. Fuel injected is available information in the ECU.
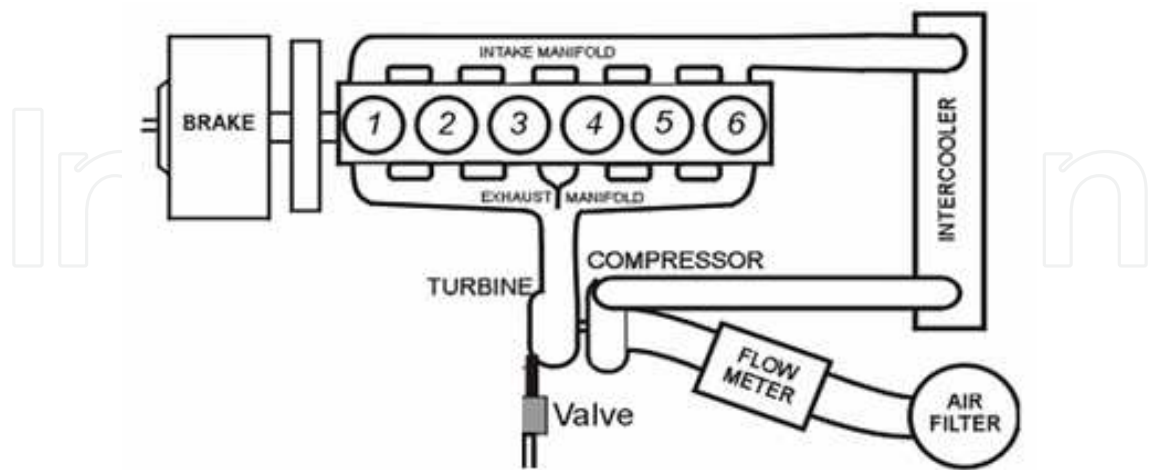

Fig. 4. System controlled.

Unfortunately the fuel injected is selected by user's needs. Thus fuel injected can not be controlled by control algorithms and it is determined as a measurable disturbance. Experimental identification done is in figure 5.
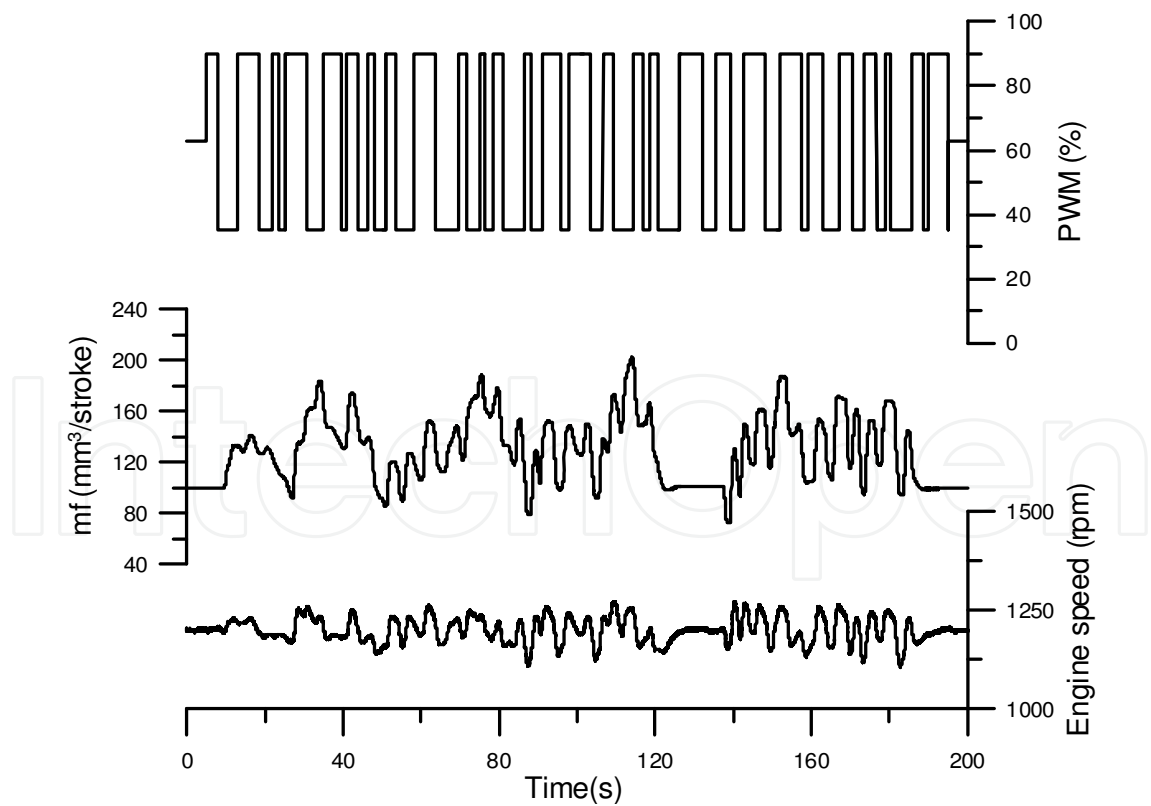

Fig. 5. Identification test performed at 1200 rpm. PWM signal is the control action on the VNT, and mf is the injected fuel. Important variations in the engine speed are due to the dynamometer response.

The VNT input applied can be seen in figure 6, where pseudo-random PWM (Pulse Width Modulation PWM) is applied in the VNT. Aleatori fuel is injected in the engine. With these inputs the boost pressure response is shown in figure below:
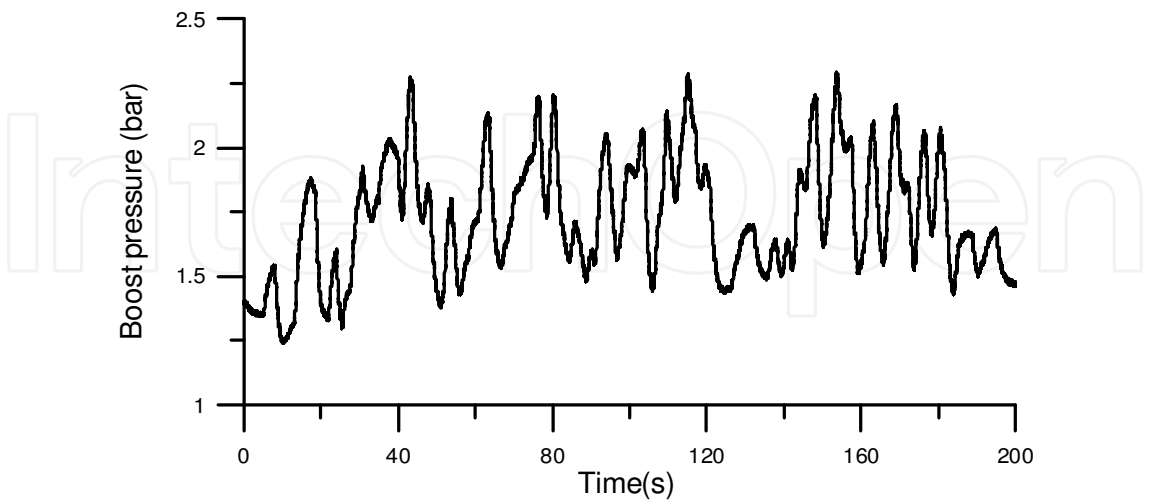
Fig. 6. Boost pressure response.

Apparently these data have not any kind of relation, but continuing with identification process we can obtain a system model. As shown the model poles are two. Physical systems don't use to have more than three poles, when model is bigger than this we could find some problems, identification algorithm tried to identify signal noise, and it has identified fast poles.

$$A = \begin{pmatrix} 0,983 & -0,022 & 0,008 \\ 0,016 & 0,886 & 0,115 \\ -0,002 & -0,027 & 0,907 \end{pmatrix}; B = \begin{pmatrix} -1,7.10^{-5} & -5,1.10^{-5} \\ 5.8.10^{-5} & -5.10^{-4} \\ -2,1.10^{-4} & 5.10^{-4} \end{pmatrix}$$

$$C = \begin{pmatrix} 8,89 & -1,18 & -1,2 \end{pmatrix}; D = 0$$

Checking the results, we could realize that the system model is good enough, see figure 7.
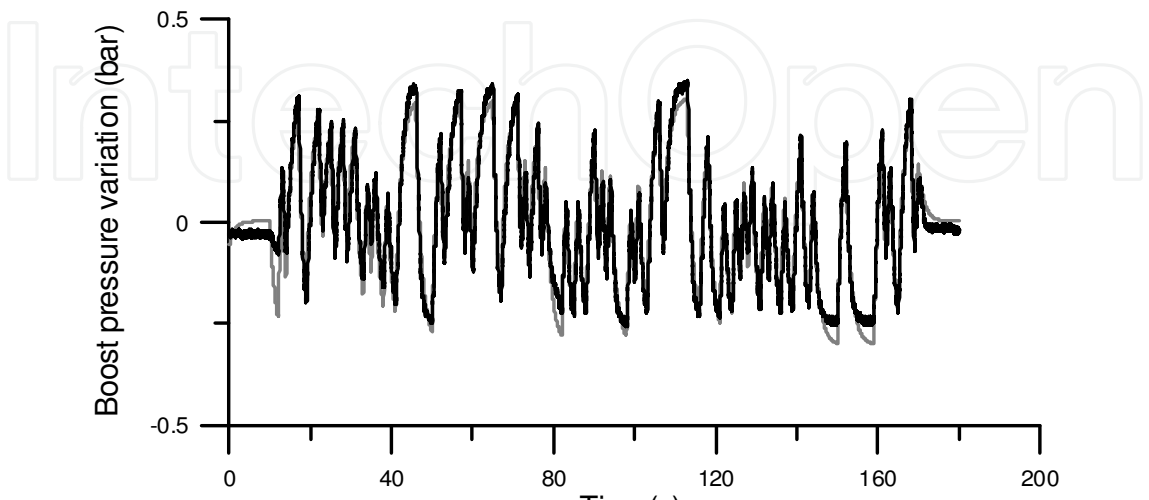
Fig. 7. Measured evolution (black) and predicted evolution (grey) of the boost pressure for the validation test at 1200 rpm.

4.2 Control system examples

Once the model is identified the controller is calculated by using control algorithms developed in section 3, it is time to control the system and testing with experimental data. A comparison is going to be shown in the following figures. The test proposed is a transient from steady conditions to full load. This is a typical situation in which a Diesel engine is requested full load to make advancement in a road. The engine is turning at 1200 rpm and no torque is need, suddenly all torque available is needed, and then engine, electronics and thermodynamics must change to the new conditions. Four different algorithms are controlling the system. The idea is testing those algorithms in order to check the engine behavior if the controller of air path is one algorithm or other one.

The algorithms tested were: PID controller with feed forward of fuel behavior, PID with Fuzzy login in parameters plus feed forward in fuel injected, GPCDM developed before and DMCDM shown in section 3.

PID controller was tuned for getting the best feasible performance. It was tuned from Ziegler Nichols parameter to updating PID until the performance was good enough. Feed forward was programmed to use measurable disturbance behavior available.

PID Fuzzy: this controller was programmed as one improvement of standard PID. The proposed algorithm is a gain schedule depending on error. Feed forward is also used here.

DMCDM: a model predictive controller based on impulse response with measurable disturbance. Control horizon and prediction horizon are chosen as long time. When performance is required long control horizon and prediction horizon must be chosen. Fro tuning the algorithm one weight is fixed to 1 and the other is changed.

GPCDM: this is an other MBPC, so many ideas explained for DMCDM are applicable here. Long time control horizon and long time prediction horizon are used for reaching good performance. The real control cost is not available, so we keep one weight fixed and we fit the second weight for tuning the system. GPC and GPCDM have a very special parameter, which is not in other kinds of MBPC. T polynomial is a parameter, which theoretically is a polynomial with zeros of the colored derivative noise from CARIMA model. In fact, it is a parameter for tuning the GPC. When noise is present in the system, this parameter must be tuned else you could put it as 1. there are many works explaining the behavior of the parameter Clarke [26]. Moreover the robustness of the system is widely improved by tuning the parameter [28,132]. Typical values of T polynomial are $T = 1-0.7*z^{-1}$; or T = convolution of ($[1-0.7.z^{-1}],[1-0.7.z^{-1}]$), sometimes 0,7 are placed by 0,8 or 0,6. This is an empirical rule that can help the designer to choose the best option.

The structures are programmed as follow:

PID feed forward:

The boost pressure set point is depending on engine speed and mass fuel requested by used. Additionally the fuel injected is the measurable system disturbance. So the fuel injected is the feed forward control action. Error between boost pressure set point and measured boot pressure is the input to the PID controller. Control action calculated is the addition between feed forward and PID in figure 8. Finally control action applied is the calculated one processed by one anti-windup. Experimental result will be shown in following figures.

The controller proposed and programmed in the figure 9 is similar architecture to the PID proposed but some differences. The PID parameters are scheduled by a Fuzzy logic technique. Other sub-systems are the same, PID, feed forward and anti-windup. The behavior is similar to the PID but parameters can be tuned more accurate than a standard PID controller. The main difficult in this controller is the tuning process, because the

parameters are more difficult to set up. The experience of the designer must be higher than the standard PID. For tuning the controller many experiments must be done and widely analyzed for choosing the best option.
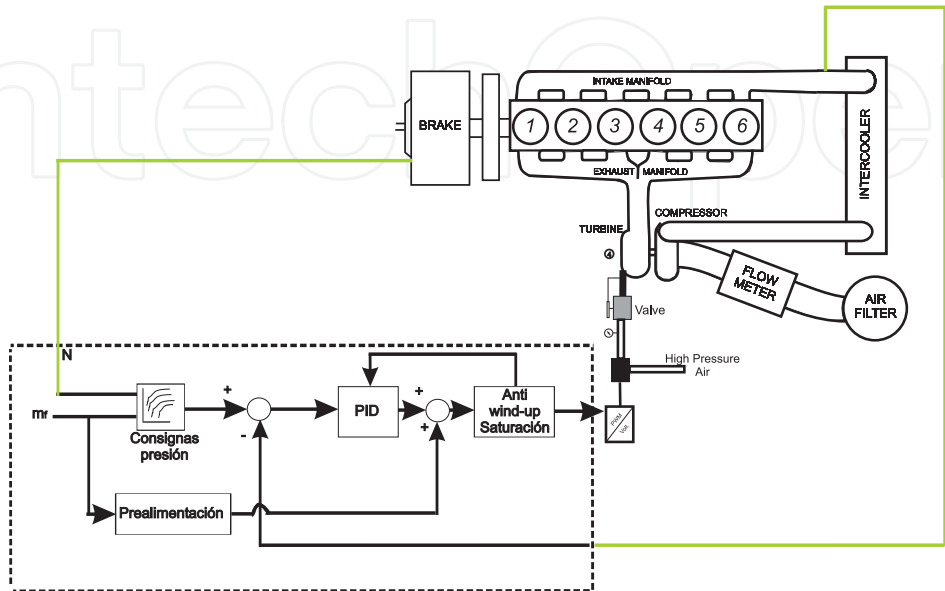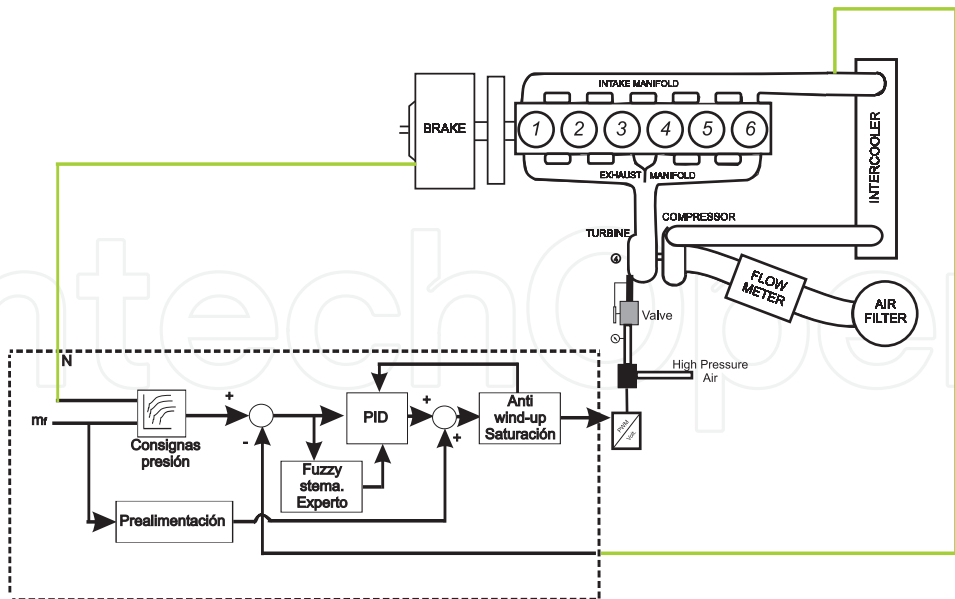

Fig. 8. PID Architecture programmed.

Fig. 9. PID scheduled by fuzzy controller and feed forward.

GPCDM architecture is the one proposed in the figure 10. Set point is processed by h(z). when future set point is not available, future ones are the actual one. Thus h(z) became as a constant term. Additionally measurable disturbance can be processed as past information
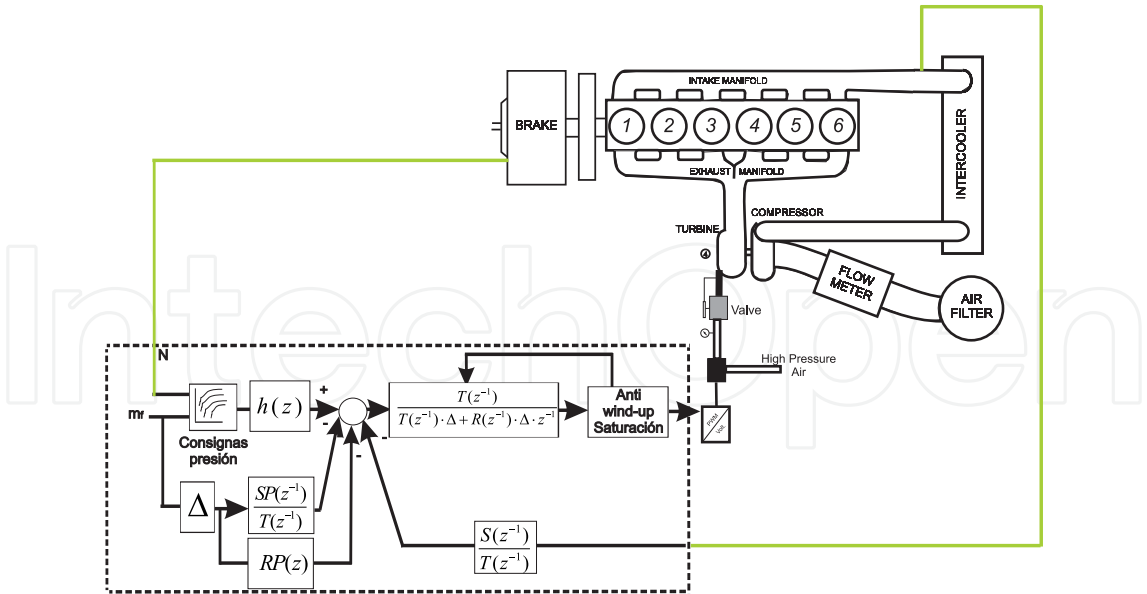
Fig. 10. GPCDM programmed.

and future information, see equation (30). In this experiment only actual and past fuel injected can be processed, but future disturbances can be considered. Anti-windup is also programmed, when control action calculated is higher than maximum, the algorithm is frozen. Thus the integral component of the algorithm can not affect to the algorithms performance.
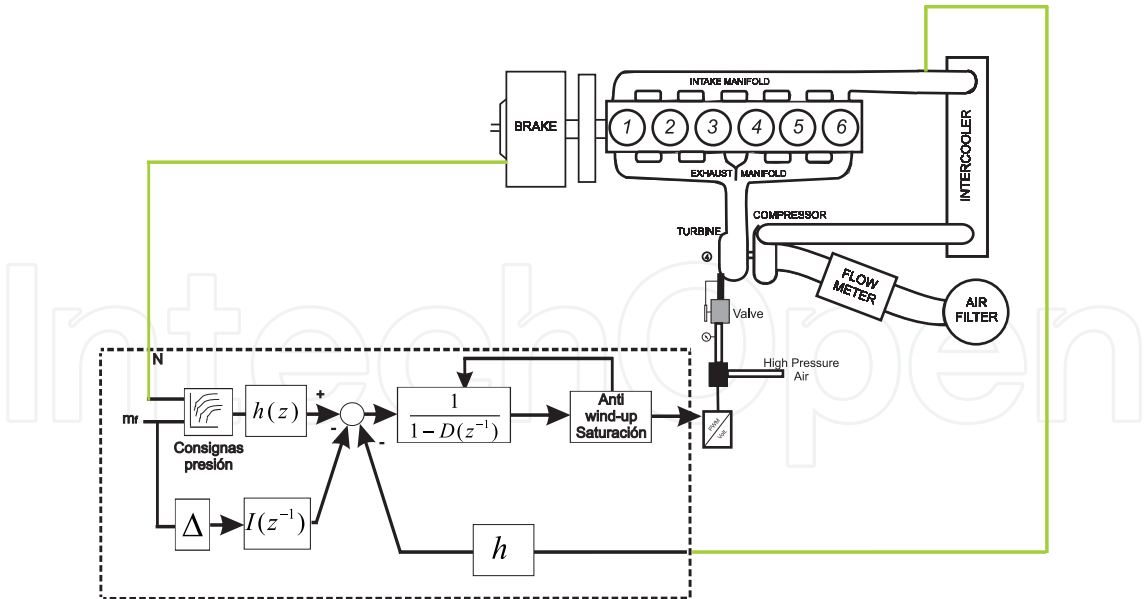

Fig. 11. DMCDM programmed.

DMCDM programmed is shown in figure 11. Measurable disturbances are considered and included in the controller. Anti-windup is programmed, a similar system to the one used in the GPCDM. When maximum or minimum control action is reached, the algorithm is frozen until control action is going to the opposite way.

5. Experimental results

As explained before full load transient test were done. The engine was in steady conditions or idle conditions at 1200 rpm. When test starts, full load is required from the engine. Thus maximum quantity of fuel is injected in order to burn it and getting the power. Additionally air mass is needed for burning the fuel, which is why VNT must be controller to give the boost pressure necessary for the new conditions. More boost pressure that necessary produces an increment in exhaust pressure and some loses of torque and power and more pollution in exhaust gases. If we have less boost pressure than set point then we will not have enough air flow for burning the fuel in the right conditions.

Many experiments were done, but only the best performance is shown in this chapter. In figure 12, boost pressure can be observed, figure 13 contains exhaust pressure behavior, and the figure 14 shows engine torque.
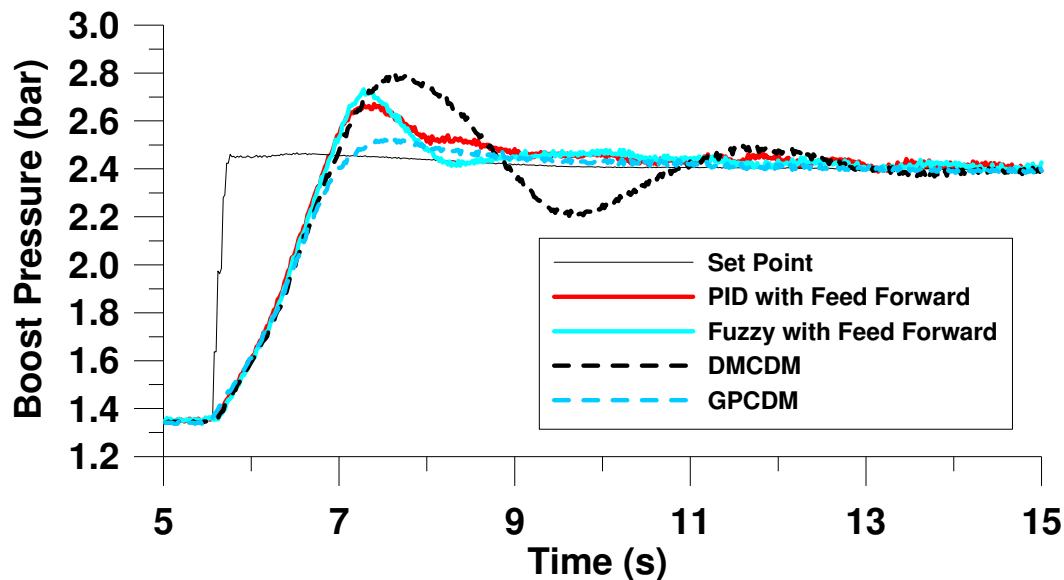

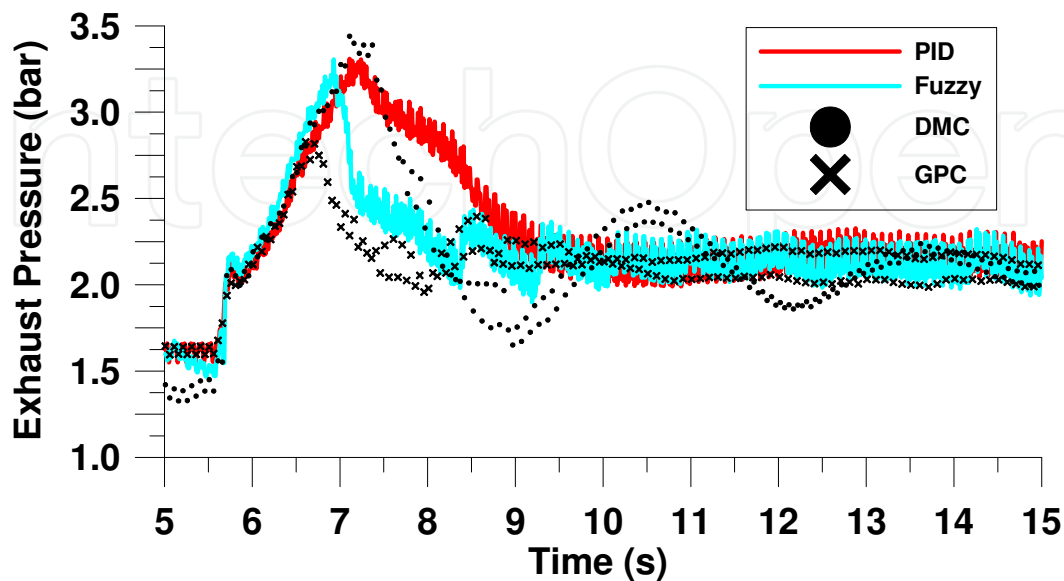Fig. 12. Boost pressure evolution in transient conditions.

Fig. 13. Exhaust pressure evolution in transient conditions.

In figure 12, boost pressure behavior is presented. Thin line is the set point requested from idle condition to full load one. This is the best boost pressure conditions considering, efficiency, torque, power, pollution etc. so the system must be in set point as soon as possible. Continuous lines are PID controller, solid red line is the PID with feed forward and solid blue line is the PID Fuzy logic controller. GPCDM is the dashed blue line and DMCDM is the dashed black line.

When transient test starts, the first part of the test are the same in all controllers. This happens because VNT is in bounds, completely close and no different performance can be appreciated. When VNT must be opened controller performance is different. GPCDM is opening the VNT before than others, that is why over oscillation is lower than others. The predictive model CARIMA helps the GPCDM to advance the system behavior, thus GPCDM can predict the VNT open to be before and avoiding the over oscillation. PID must be tuned with low influences of derivative term, the reason is the stability. Derivative term can produce instable conditions in the system. This controller realizes later about over oscillation and start opening the VNT later. The worst performance is done by DMCDM. Maybe the system model is very single and it is more sensible to non-linearity. Fuzzy controller improves the PID behavior because it is an improvement of PID.

Figure 12. contains the exhaust boost pressure. The steady conditions are reached in similar time to boost pressure. The over-pressure in exhaust produces over-speed in turbine or torque loses, and more pollutants than permitted.
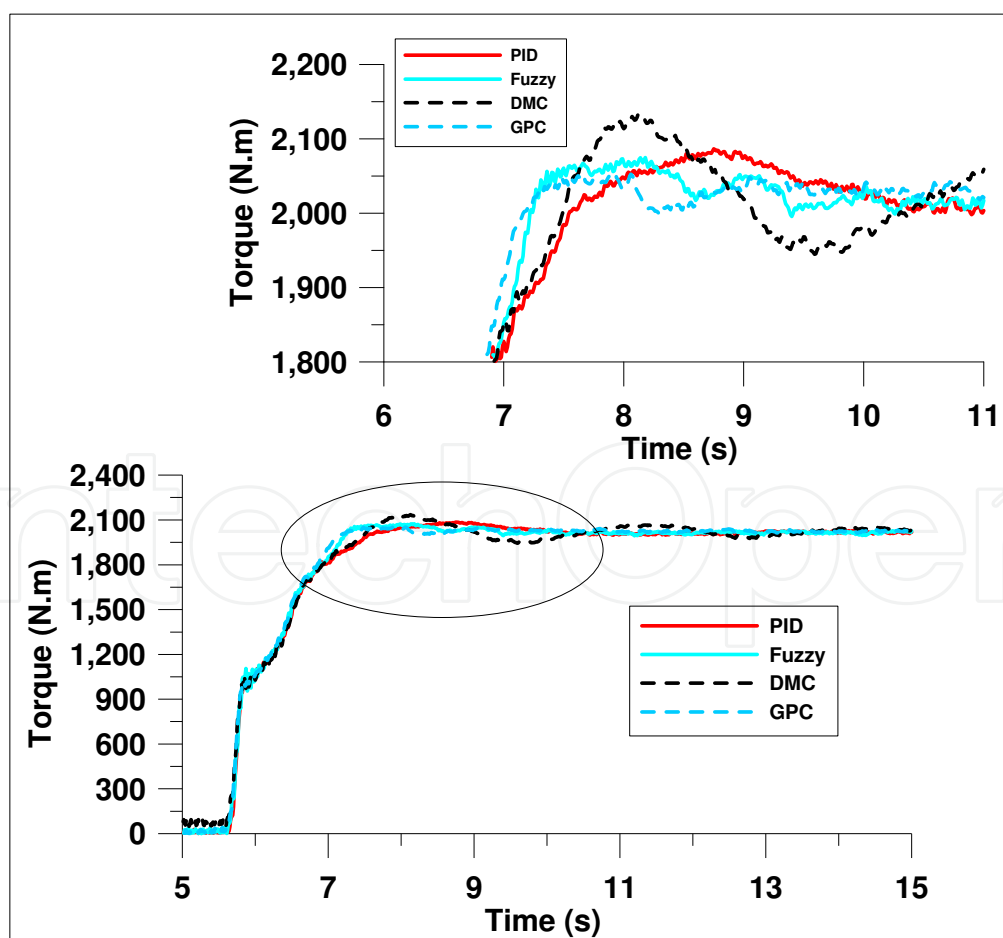

Fig. 14. Torque evolution and air-path influence during transient.

Figure 14. shows torque performance. Both GPCDM and Fuzzy reach steady conditions at the same time. So the torque is not very influenced by boost pressure. But if other controllers are analyzed engine torque is influenced and oscillations are very uncomfortable.

Finally a summary of system response are shown in table 1 and table 2. Table 1 summarizes the system response from the point of view of engine analysis. Table 2 analyzes the system from the point of view of control parameters.

Experimental results					
	Max Exhaust pressure (bar)	**Overshoot Exhaust pressure from steady conditions**	**max (Exhaust Pressure) - max(Boost Pressure)**	**Max(Torque)**	**Torque overshoot**
PID+FF	3.223	53.48	0.482	2031	85
PID+FF+Fuzzy	3.443	63.48	0.702	2100	86
DMCDM	3.220	53.33	0.428	2101	133
GPCDM	2.655	26.43	0.129	2055	36

Table 1. Experimental results of system behavior.

Experimental results				
	$\int_{5.5}^{22.5} e(t)^2 \cdot dt$	$\int_{5.5}^{22.5} \Delta u^2 \cdot dt$	J	**Boost pressure overshoot**
PID+FF	29.11	454.38	51.83	34.10
PID+FF+Fuzzy	29.13	2911.15	174.68	33.10
DMCDM	36.96	18.80	37.90	39.20
GPCDM	27.62	37.40	29.49	12.60

Table 2. Experimental results, control parameters.

The most important variable from the point of view of control is the control Cost or J. This variable summarized the control behavior from the point of view of control cost, considering control effort and control error.

5.2 Discussion

The first conclusion that can be drawn from analyzing the results is that when fuel mass is considered in the algorithm, behavior is always improved. This fact had been observed during the identification process and is logical, since the more information the algorithm has about the system, the better control it can apply. This is due to the air management process being more influenced by fuel amount than VGT control actions. The fuel amount gives energy, this energy goes to the turbine and the turbocharger increases its speed, thus the air management system changes. The delivered fuel quantity therefore has a very strong influence on the air management system. If this variable is included in control algorithms, better control will be obtained.

The analysis of the behavior of the Fuzzy controller seems to show quite a good response but at the cost of a high control effort. One of the disadvantages of this type of controller is

the high overshoot of the exhaust pressure. The control effort is one order of magnitude higher than the baseline controller. However, the greatest disadvantage is their complicated tuning process due to having a large number of independent parameters.

Another interesting aspect is that not all the MPCs work better than the existing controllers. For example, the DMC provided rather doubtful results. This was a surprise and could be explained as follows: the prediction model of the DMC is considered to be highly sensitive to nonlinearities in the system, since the system itself is highly nonlinear (in spite of having linear zone models). This algorithm finds it difficult to predict engine behavior and therefore does not take the right decisions. However, the predictive control GPC algorithm is based on the CARIMA model and is more robust to system nonlinearities. In fact, one of the most important parameters for the correct GPC tuning was the polynomial T(z).

The best all-round algorithm was the GPCDM. The most significant difference between algorithms can be seen in the exhaust pressure in Figures 8 and 9. This parameter indicates the effort made by the engine to control the air path and high pressure has a negative influence on engine torque.

The air management system, and consequently various engine parameters, was strongly influenced by the controller used. One of the most susceptible to this influence is PMEP, which also affects engine torque, turbocharger pressure, etc. If an overshoot occurs it will produce exhaust pressure peak and pumping losses. This fact produced less efficiency and torque during transient conditions.

6. Conclusions

Few linear models were obtained from the air management system of the engine and they are quiet good approximations of the nonlinear plant. These models can be used for linear controller design with low computational effort. GPCDM, DMCDM, PID and Fuzzy PID are tested in a standard test bench. In comparison to standard gain scheduled PID, setpoint tracking could be improved. The additional degrees of freedom of GPC can be used for tuning the robustness of control.

7. Acknowledgements

8. References

[1] Albertos P. y Ortega R. "On generalized predictive control two alternative formulations". Automatica, Vol. 25 No.5, pp. 753–755, 1989.
[2] Bordons C. y Camacho E.F. "A generalized predictive controller for a wide class of industrial process". IEEE Transactions on control system technology, Vol. 6 No. 3, May 1998.

[3] Camacho E.F., Berenguel M. y Rubio F.R. "Application of a Gain Scheduling Generalized Predictive Controller to a Solar Power Plant". Control Engineering Practice, Vol. 2(2), pp. 227–238, 1994.

[4] Chow C. y Clarke D.W. "Actuator nonlinearities in predictive control advances in model-based predictive control". Ed. Oxford university press, p´ag. 245, 1994.

[5] Clarke D.W. y Gawthrop P.J. "Self-tuning controller". Proc. IEE, Vol. 122, pp. 929–934, 1975.

[6] Clarke D.W. y Gawthrop P.J. "Self-tuning control". Proc. IEE, Vol. 126, pp. 633–640, 1979.

[7] Clarke D.W. yMothadi C. "Properties of generalized predictive control". Automatica, Vol. 25(6), pp. 859–875, 1989.

[8] Clarke D.W., Mothadi C. y Tuffs P.S. "Generalized predictive control -Part I. The basic algirithm". Automatica, Vol. 23 No.2, pp. 137–148, 1987.

[9] Clarke D.W., Mothadi C. y Tuffs P.S. "Generalized predictive control -Part II. Extension and interpretation". Automatica, Vol. 23 No.2, pp. 149–160, 1987.

[10] Clarke D.W. y Zhang L. "Long-range predictive control using weightingsecuence models". IEE Proceedings, Vol. 134 D No.3, 1987.

[11] Corradini M.L. y Orlando G. "A VSC algorithm based on generalized predictive control". Automatica, Vol. 33 ,5, pp. 927–932, 1997.

[12] Cutler C.R. y B.L. Ramaker. "Dynamic matrix control–a computer control algorithm". Proceedings of the Joint Automatic Control Conference, 1980.

[13] de la parte M. Perez, Camacho O. y Camacho E.F. "Developement of a GPC-based sliding mode controller". ISA Transaction, Vol. 41 No.1, pp. 19–30, 2002.

[14] Evans W.R. "Control system synthesis by root locus method". AIEE Transaction, Part II, Vol. 69, pp. 66–69, 1950.

[15] Galanti M., Magrini S., Mosca E. y Spicci V. "Receding horizon predictive control applied to a high performance servo-controlled pedestal". Proc. of ESPRIT (III workshop on industrial application of MBPC, Cambridge, 1992.Soeterboek R. Predictive control a unified approach. Prentice Hall int. UK Cambridge, 1992.

[16] Garcia C.E. y Morshedi A.M. "Quadratic programming solution of dynamic matrix control (QDMC)". Chemical engineering communication, Vol. 46, pp. 73–87, 1986.

[17] García-Ortiz, J.V. "Aportación a la mejora del control de la renovación de la carga en motores Diesel Turboalimentados mediante distintos algoritmos de control". Thesis in Universidad Politécnica de Valencia. CMT Motores térmicos. 9 Julio 2004.

[18] Havlena V. y F.J. Kraus. "Receding horizon MIMO LQ controller design with guaranteed stability". Automatica, Vol. 33 No.8, pp. 1567–1570, 1997.

[19] Kennel R. "Generalized predictive control GPC-ready for use in drive application?". 32nd IEEE electronics specialist conference PELS Vancouver June, 2001.

[20] Keyser R.M.C. De, de Velde G.A. Van y Dumortier F.A.G. "A comparative study of self-adaptative long-range predictive control methods". Automatica, Vol. 24(2), pp. 149–163, 1988.

[21] Kouvaritakis B., Cannon M. y Rossiter J.A. "Who needs QP for linear MPC anyway?". Automatica, Vol. 38, pp. 879–884, 2002.

[22] Kouvaritakis B., Rossiter J.A. y Chang A.O.T. "Stable gerenalized predictive control: an algorithm with guaranteed stability". Proc. IEE, Part D,, Vol. 139(4), pp. 349–362, 1992.

[23] Luo X. Analysis of automatic control system for deposition of CIGS films. Tesis Doctoral, Colorado School of Mines, Colorado Illinois, USA, 2002.

[24] Martinez M., Senent J. y Blasco X. "A comparative study of classic vs genetic algorithms optimization applied in GPC controller". IFAC world congress, San Francisco USA, 1996.

[25] Mosca E. y Zhang J. "Stable redesign of predictive control". Automatica, Vol. 28, pp. 1229–1233, 1992.

[26] Nuñez A., Cueil J.R. y Bordons C. "Modelado y control de una almazara de extraccion de aceite de oliva". XXII Jornadas de Autimatica, 2001.

[27] Nyquist H. "Regeneration theory". Bell system tech. Journal, Vol. 11, 1932.

[28] Overschee P. Van y Moor B. De. "N4SID: Subspace algorithm for the identification of combined deterministic-stochastic systems". Automatica, Vol. 30, pp. 75–93, 1994.

[29] Pannocchia G. y Rawlings J.B. "Robustness of MPC and Disturbance Models for Multivariable Ill-conditioned Processes". Technical report TWMCC No. 2001-02, 2001.

[30] Poulser N.K., Kuovaritakis B. y Cannon M. "Constrained predictive control and its application to a coupled-tanks apparatus". Int. Journal of control, Vol. 74 No. 6, pp. 552–564, 2001.

[31] Qin S.J. y Badgwell T.A. "An overview of industrial model predictive control technology". AIche symposium series, Vol. 316(93), pp. 232–256, 1996.

[32] Qin S.J. y Badgwell T.A. "A survey of industrial model predictive control technology". Control engineering practice, Vol. 11, pp. 733–764, 2003.

[33] Richalet J., Rault A., Testud J.L. y Papon J. "Model predictive heuristic control: Applications to industrial processes". Automatica, Vol. 14, pp. 413–428, 1978.

[34] Salcedo J.V, Martinez M., Sanchis J. y Blasco X. "Design of GPC's in state space". Automatika, Vol. 42(3-4), pp. 159–167, 2001.

[35] Sanchis J., Martinez M., Ramos C. y Salcedo J.V. "Principal component GPC with terminal equality constraint". In 15th IFAC World Congress , Barcelona (Spain), July 2002.

[36] Shridhar R. y Cooper D.J. "A tuning strategy for unconstrained SISO model predictive control". Ind. Eng. Chem Res., Vol. 36, pp. 729–746, 1997.

[37] Soeterboek A.R.M., Verbrugger H.B., der Bosch P.P.J Van y Butler H. "Adaptative predictive control-A unified approach". Proc. 6th Yale workshop on applied of adapt. system theory new Haver USA, 1990.

[38] Soeterboek A.R.M., Verbrugger H.B., der Bosch P.P.J Van y Butler H. "On the unification of predictive control algorithm". Proc. 29th IEEE conference on decision and control, Honolulu USA, 1990.

[39] Soeterboek R. Predictive control a unified approach. Prentice Hall int. UK Cambridge, 1992.

[40] Tsang B.A. y Clarke D.W. "Generalized predictive control with input constraints". IEE Proceedings, Vol. 135, 1988.

[41] Verhaegen M. "Identification of the deterministic part of MIMO state space models". Automatica, Vol. 30, pp. 61–74, 1994.

[42] Zafirion E. "Robust model predictive control of process with hard constraints". Comp. Chen. Eng., Vol. 14, pp. 359–371, 1990.

[43] Zheng A. y Morari M. "Stability of model predictive control with soft restraints". Proceedings 33rd CDC Florida, pp. 1018–1023, 1994.

[44] Zhu Y. y Backx T. Identification of multivariable industrial process, for simulation, diagnosis and control. Springer-verlag, ISBN 3-540-19835-0, 1993.

Advanced Model Predictive Control

Edited by Dr. Tao ZHENG

Model Predictive Control (MPC) refers to a class of control algorithms in which a dynamic process model is used to predict and optimize process performance. From lower request of modeling accuracy and robustness to complicated process plants, MPC has been widely accepted in many practical fields. As the guide for researchers and engineers all over the world concerned with the latest developments of MPC, the purpose of "Advanced Model Predictive Control" is to show the readers the recent achievements in this area. The first part of this exciting book will help you comprehend the frontiers in theoretical research of MPC, such as Fast MPC, Nonlinear MPC, Distributed MPC, Multi-Dimensional MPC and Fuzzy-Neural MPC. In the second part, several excellent applications of MPC in modern industry are proposed and efficient commercial software for MPC is introduced. Because of its special industrial origin, we believe that MPC will remain energetic in the future.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Jose Vicente Garcia-Ortiz (2011). MBPC – Theoretical Development for Measurable Disturbances and Practical Example of Air-path in a Diesel Engine, Advanced Model Predictive Control, Dr. Tao ZHENG (Ed.), ISBN: 978-953-307-298-2, InTech, Available from: http://www.intechopen.com/books/advanced-model-predictive-control/mbpc-theoretical-development-for-measurable-disturbances-and-practical-example-of-air-path-in-a-dies

INTECH

open science | open minds