

# Fast Nonlinear Model Predictive Control using Second Order Volterra Models Based Multi-agent Approach

Bennasr Hichem and M'Sahli Faouzi  
*Institut Supérieur Des Etudes Technologiques de SFAX*  
*Ecole Nationale d'ingénieur de Monastir*  
*Tunisia*

## 1. Introduction

Model predictive control (MPC) refers to a class of computer control algorithms that utilize a process model to predict the future response of a plant. During the past twenty years, a great progress has been made in the industrial MPC field. Today, MPC has become the most widely implemented process control technology. One of the main reasons for its application in the industry is that it can take account of physical and operational constraints. In classical model predictive control (MPC), the control action at each time step is obtained by solving an online optimization problem. If it is possible, MPC algorithms based on linear models should be used because of low computational complexity [Maciejowski J,2002]. Since properties of many technological processes are nonlinear, different nonlinear MPC techniques have been developed [Qin, S. J et al, 2003]. The structure of the nonlinear model and the way it is used on-line affect the accuracy, the computational burden and the reliability of nonlinear MPC. Several different attempts to reduce computational complexity have been released during the last thirty years. The simplest way to reduce on-line computation is to transform the NMPC problem into LMPC. The nonlinear system is transformed into a linear system using a feedback-linearizing law, the input constraints are mapped into constraints on the manipulated input of the transformed system and the obtained constrained linear system is controlled using LMPC [Kurtz M.] et al,1997]. An interesting strategy is presented in [Arahal M.R et al.,1998], when the linear model is used to predict future process behavior and the nonlinear model is used to compute the effect of the past input moves. The most straightforward technique used to implement fuzzy models [Fischer M et al.,1998] is based on a linearization method. The accuracy of the linear model can be improved by relinearizing the model equations several times over a sampling period or by linearizing the model along the computed trajectory [Mollov S.,et al.,2004]. Another approach has been used by a number of researchers such as in [Brooms A et al., 2000], where the NMPC problem is reduced to an LMPC problem at each time step using a successive linearization. The structure of certain nonlinear empirical models allows the NMPC optimization problem to be solved more efficiently than is possible with other forms. Such an approach will be followed in [Abonyi, J et al,2000]. An algorithm for controller reconfiguration for non-linear systems based on a combination of a multiple model

estimator and a generalized predictive controller is presented in [Kanev, S et al., 2000], in which a set of models are constructed. Each corresponding to a different operating condition of the system and an interacting multiple model estimators is utilized to yield a reconstruction of the state of the non-linear system. For unconstrained control based on linear process models and a quadratic cost function, the control sequence can be analytically calculated. When linear constraints are taken into account, the solution can be found using quadratic programming techniques. With the introduction of a nonlinear model into MPC scheme, a nonlinear programming technique (NLP) has to be solved at each sampling time to compute the future manipulated variables in on-line optimization that is generally non-convex which make their implementation difficult for real time control. During the past decade significant theoretical results as well as advances in the implementation strategies of NMPC have been obtained and NMPC has been successfully applied in practice to relatively slow plants, mainly in the process industry. However, the application of such techniques for fast nonlinear systems remains a widely opened problem due to the computation burden associated with solving an open loop optimal control problem. Most of the research has focused on computations carried out by one agent. In [Negenborn R et al., 2004], a survey how a distributed multi-agent MPC setting can reduce the computations of a single MPC agent. Moreover, researchers have investigated feedback linearization model predictive control (FLC-MPC) schemes for their ability to handle constraints on input and output [Soest Van W.R et al., 2005]. These approaches reduce the on-line computation by transforming the NLMPC problem into a LMPC and quadratic programming can be used to handle constraints. When sampling times become so short, the computation times for QP solution can no longer be neglected [Jaochim H et al., 2006]. In [Didier G ,2006], a distributed model predictive control is considered and the proposed strategy allows dramatic reduction of the computational requirement for solving large-scale nonlinear MPC problem due to computation parallelism. However, recent advancements in MPC allow for a faster online solution by shifting some of the computational burden off-line. We can notice that many optimization algorithm solutions for NMPC have been investigated lately; however, an analytical solution in NMPC approach is usually impossible to find. One possible way to address computational complexity is to decentralize the optimization tasks. Attention has been focused on multi-agent model predictive control approach [H.Ben Nasr et al.,2008a,b,c,d,e]. There are multiple agents in multi-agent model predictive control. Each uses a model of its sub-system to determine which action to take. Decentralized agent architecture and decentralized model decomposition are then chosen, in which there are numerous agents that do not have any interaction among one another. A methodology based multiagent has been investigated in the implementation of a given predictive control law for nonlinear systems. Such procedure relies on the decomposition of the overall system into subsystems and a multiple agents each uses a model of its sub-system to determine which action to take.

In this chapter book, new NMPC scheme based MAMPC (Multiagent model predictive control) is implemented to reduce the computational effort. The performance of the proposed controllers is evaluated by applying to single input-single output (SISO) control of non linear system. Moreover, in general, the optimization problem is nonconvex and leads to many difficulties impacting on implementation of MPC. These difficulties are related to feasibility and optimality, computation and stability aspects. In order to avoid solving nonconvex optimization problem, MAMPC (Multiagent model predictive control) optimization procedure, a method for convex NMPC was also developed in this chapter book. Theoretical

analysis and simulation results demonstrate better performance of the MAMPC over a conventional NMPC based on sequential quadratic programming (SQP) in tracking the set point changes as well as stabilizing the operation in the presence of input disturbances. In this work, our main objective has been to illustrate the potential advantage of nonlinear predictive control based multiagent when applied to nonlinear systems. The suggested approach was to identify a new control algorithm that in essence is a bridge between linear and nonlinear control. This resulted in the development of the MAMPC approach. Through simulation-based comparisons, it is shown that a MAMPC control algorithm is capable of delivering significantly improved control performance in comparison to a conventional NMPC, so that the difficulty of minimizing the performance function for nonlinear predictive control is avoided, which is usually carried by the use of NLP solved at each sampling time that generally is non-convex. In this chapter book we describe algorithm that find the solution of a non-convex programming and also demonstrated that global nonlinear requirements can effectively be resolved by considering smaller regimes. The simulation example shows that the multi-agent compares favorably with respect to a numerical optimization routine. Moreover, the MAMPC reduces the online computational burden and hence has the potential to be applied to the system with faster time constants.

## 2. Statement of the problem

### 2.1 Process model

A broad class of physical systems can be represented using the Volterra model. Particularly, it was shown that a truncated Volterra model could represent any non-linear system, time-invariant with fading memory. This model is thus particularly attractive for non-linear systems modeling and identification purpose. One of the main advantages of the Volterra model is its linearity-in-parameters, i.e. the kernel coefficients. This property allows the extension of some results established for linear model identification to this model. In this work, we consider the control of a class of single-input single output non-linear system described by the following non-linear discrete-time parametric second-order Volterra model (Haber *et al.* 1999a,b):

$$y(k) = y_0 + \sum_{i=1}^{n_y} a_i y(k-i) + \sum_{i=1}^{n_u} b_i u(k-i) + \sum_{i=1}^{n_u} \sum_{j=1}^i b_{ij} u(k-i)u(k-j) + \varepsilon(k) \quad (1)$$

Where  $y_0$  is a bias term,  $y(k)$  is the output,  $u(k)$  is the input,  $a_i, b_i$  and  $b_{ij}$  are the parameters of the parametric Volterra model,  $n_u$  and  $n_y$  are the number of lags on the input and the output, respectively.  $\varepsilon(k)$  Contains all terms up to second-order. One advantage of using the parametric Volterra model is that the one-ahead prediction problem can be formulated as a linear regression, which simplifies the identification of the parameters from input-output data. Therefore, the model given by "Equation (1)" can be written as:

$$y(k) = \theta^T \phi(k) + \varepsilon(k). \quad (2)$$

With:

$$\theta^T = [y_0, a_1, a_2, \dots, a_{n_y}, b_1, b_2, \dots, b_{n_u}, b_{1,1}, \dots, b_{n_u, n_u}] \quad (3)$$

$$\phi^T(k) = [1, y(k-1), \dots, y(k-n_y), u(k-1), \dots, u^2(k-1), \dots, u^2(k-n_u)] \quad (4)$$

Where  $\phi(k)$  and  $\theta$  are the regressor and the parameter vectors, respectively. The model "Equation (3)" is linear in parameters, and its regressors and parameters may be identified from input output information. Moreover, from identification point of view, parametric Volterra models are superior to Volterra series models in the sense that the number of parameters needed to approximate a process are generally much less with parametric Volterra models. This is due to the fact that Volterra models only include previous inputs, while the model (1) includes previous outputs as well as previous inputs.

## 2.2 Optimization criteria

The purpose of the control strategy is to compute future control moves which will minimize some performance function based on the desired output trajectory over a prediction horizon, subject to constraints on input and output signals [D.W. Clarke et al., 1987]. The most common objective cost function, also used here, is:

$$J(N_1, N_2, N_u, \delta) = \sum_{j=N_1}^{N_2} (w(k+j) - \hat{y}(k+j/k))^2 + \sum_{j=1}^{N_u} (\lambda_j \Delta u(k+j-1))^2 \quad (5)$$

Subject to

$$\begin{aligned} \Delta u_{low} \leq \Delta u(j+j-1) \leq \Delta u_{high} \quad & \text{for } 1 \leq j \leq N_u \\ u_{low}(k) \leq u(k+j-1) \leq u_{high}(k) \quad & \text{for } 1 \leq j \leq N_u \end{aligned} \quad (6)$$

Where  $N_1$  is the minimum prediction horizon,  $N_2$  is the maximum prediction horizon,  $\hat{y}(k+j/k)$  is an optimum  $j$ -step ahead prediction of the system output on data up to time  $k$ ,  $w(k+j)$  is a sequence of future set points,  $N_u \leq N_2$  is the control horizon, and  $(\lambda_j)_{j=1 \dots N_u} = (\lambda_1, \dots, \lambda_{N_u})$  are control-weighting factors usually assumed to be equal to each other used to penalize the control increments.  $\Delta u(k+j-1)$ ,  $j \in [1, N_u]$ , is a sequence of future control increments computed by the optimization problem at time  $k$ ;  $\Delta u(k+j-1) = 0$  for  $j > N_u$ . For the constraints  $u_{low}, u_{high}, \Delta u_{low}, \Delta u_{high}$ , are respectively the lower limit, upper limit, lower derivative limit and higher derivative limit of the control input. Using the quadratic prediction equation of the model, the cost function becomes fourth degree equation in the control increments. The objective function never exceeds fourth order, regardless of the value of the prediction horizon. (Haber, 1999a, 1999b)

## 2.3 Nonlinear Predictive Control

Despite of the wide exposure of and the intensive research efforts attracted over the past few decades on Nonlinear model predictive control (NLMPC), this control strategy is still being perceived as an academic concept rather than a practicable control technique. However, nonlinear model predictive control is gaining popularity in the industrial community. The formulations for these controllers vary widely, and almost the only common principle is to retain nonlinearities in the process model [Matthew et al., 2002]. In nonlinear control, a receding horizon approach is typically used, which can be summarized in the following steps:

1. At time  $k$ , solve, on-line, an open-loop optimal control problem over some future interval, taking into account the current and future constraints.
2. Apply the first step in the optimal control sequence.
3. Receding strategy so that at each instant the horizon is displaced towards the future, which involves the application of the first control signal of the sequence calculated at each step.

The process to control is assumed to be represented by a mono-variable second order parametric Volterra model. The model given by (1) can be expressed as:

$$A(q^{-1})y(k) = y_0 + B_1(q^{-1})u(k) + B_2(q_1^{-1}, q_2^{-1})u^2(k) + \frac{\varepsilon(k)}{\Delta(q^{-1})} \quad (7)$$

Where are two polynomials of the backward shifting operator  $q^{-1}$  given by :

$$\begin{aligned} A(q^{-1}) &= 1 + a_1q^{-1} + \dots + a_naq^{-na} \\ B_1(q^{-1}) &= 1 + b_{11}q^{-1} + \dots + b_{1nb}q^{-nb} \end{aligned} \quad (8)$$

$B_2(q_1^{-1}, q_2^{-1})$  represents the quadratic term of the Volterra model, this quantity is defined by:

$$B_2(q_1^{-1}, q_2^{-1})u^2(k) = \sum_{n=0}^{nb} \sum_{m=n}^{nb} b_{2nm}u(k-n)u(k-m) \quad (9)$$

The incremental predictive form of the parametric Volterra model can be expressed as a function of the current and future control increments :

$$\hat{y}(k+j) = v_0^j + v_1^j(q^{-1})\Delta u(k+j) + v_2^j(q_1^{-1}, q_2^{-1})\Delta u^2(k+j) \quad (10)$$

With

$$\begin{aligned} v_0^j &= y_0 + G_j y(k) + \sum_{i=j+1}^{nb+j-1} \left[ \delta_{1i} + \sum_{m=i}^{nb+j-1} \delta_{2im} \Delta u^*(k+j-m) \right] \Delta u^*(k+j-i) \\ v_{1i}^j &= v_{1i} + \sum_{m=j+1}^{nb1+j-1} \delta_{2im} \Delta u^*(k+j-m) \quad i = 1, 2, \dots, j \\ v_{2im}^j &= \delta_{2im} \quad i = 1, 2, \dots, j \quad \text{and} \quad m = 1, 2, \dots, j \end{aligned} \quad (11)$$

The effect of selecting the parameters and the coefficient of the predictive control are not investigated here, for more detail see (Haber et al., 1999a) . Replacing the incremental output by his expression, the cost function (5) can be written as follows:

$$J = (v_0 - w + v_1 \tilde{u} + v_2 \tilde{u}^2)^T (v_0 - w + v_1 \tilde{u} + v_2 \tilde{u}^2) + \lambda \tilde{u}^T \tilde{u} \quad (12)$$

With constraints, the cost function can be minimized numerically by a one-dimensional search algorithm (dynamic algorithm programming). Without constraints the solution leads to a third-degree one-dimensional equation [F.J.Doyle et al.,1995].

### 3. Multi-agent Model Predictive Control

#### 3.1 Control and design

The main idea of the proposed concept model predictive control is to transform the nonlinear optimization procedure used in a standard way into sub-problems, in which the global task can be resolved. The objective of this approach is to regulate the nonlinear system output to the expected values and satisfying the above constraints. This can be done as follows. The global system can first be decomposed on sub-systems independent of one another, for each sub-system an MPC unit sub-system is made constituting the agent controller  $i$ . Based on an analytical solution, which corresponds to the solution of the local receding horizon sub-problems, a logic unit switching tries to find the best sequence of actions sent to the nonlinear system and gives the desired trajectory. Sequences of actions that bring the global system in a desired trajectory are made and avoid any violated constraints on actions. The multi-agent controller consists of synchronizing the output of the true system at every decision step  $k$  with the reference trajectory. In fact, at every decision step the right action is the one that will cause the agent to be the most successful. The parallel controller structure is based on the fact that a neural network can be used to learn from the feedback error controller non linear system. A neural network controller is also made on, in objective to take handle the results of the actions on the global system and monitor the closed-loop system. Figure 1, shows the architecture of the multi-agent controller. In the multi-agent context, the agents are the controllers and the non linear system is the environment.

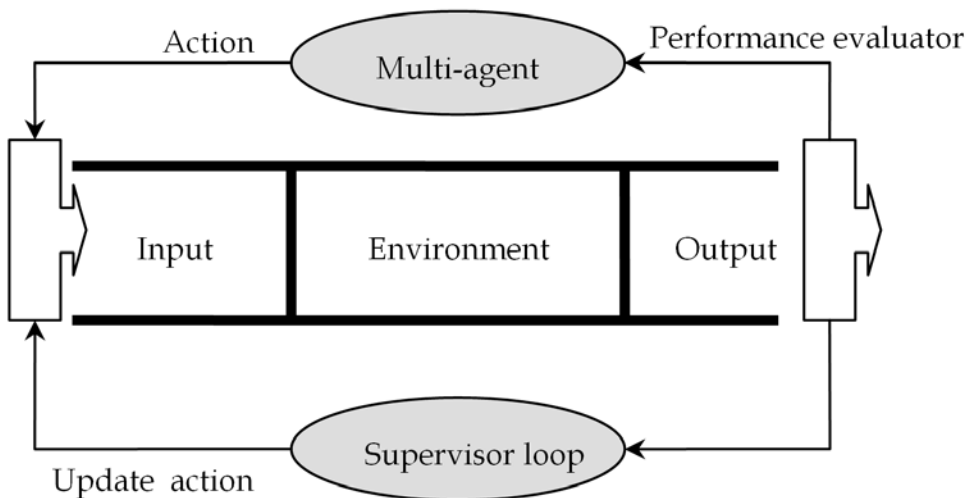


Fig. 1. Architecture of Multi-agent Controller

The basic structure of the control strategy proposed is shown in figure 2. The control problem to solve should be decomposed into supposedly independent subproblems. Each subproblem is solved by designing a controller-agent. The controller-agent is realized by some control algorithm that is operational only under particular operating conditions of the plant being controlled. Moreover, the controller-agent's action consist of the analytical

optimal control sequence elaborated in each sub-system after having learned the trajectory of the control to follow and by minimizing a local cost function. The individual solutions or controller-agents are combined into one overall solution. This implies addressing the global problems by selecting an appropriate coordination mechanism. The conceptual design consists of the following three stages:

**Structuring:** The control problem to solve should be decomposed into supposedly independent subproblems. The global system can first be decomposed on sub-systems independent of one another.

**Solving individual subproblems:** Each subproblem is solved by designing a controller-agent. An MPC unit sub-system is made constituting the controller agent. A supervisor based on performance measure  $J_k$  is used. By means of the output errors  $\epsilon_k$  for each agent's action, the supervisor decides then what action should be applied to the plant during each sampling interval  $k$ . The performance measure is given by:

$$J_k = \epsilon_k - \epsilon_{k-1}e^{-\lambda}, \lambda > 0 \tag{13}$$

Where,  $\epsilon_k$  is the error for the agent I defined by:

$$\epsilon_k = \text{setpoint} - y_a \tag{14}$$

And  $y_a$  is the plant output after agent's action.

**Combining individual solutions** The individual solutions or controller-agents are combined into one overall solution. The parallel controller structure is based on the fact that a neural network can be used to learn from the feedback error controller nonlinear system, to take handle the results of the actions on the global system and monitor the closed-loop system.

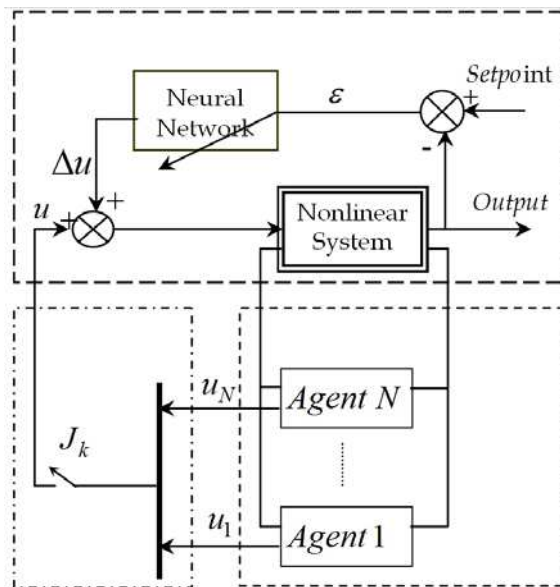


Fig. 2. Architecture of Multi-agent Controller

### 3.2 Control problem decomposition

The extension of MPC for the use of nonlinear process models is one of the most interesting research topics. These algorithms generally lead to the use of computationally intensive nonlinear techniques that make application almost impossible. In order to avoid this problem, the proposed concept algorithm utilizes a linear model extracted from the nonlinear model. A decentralized model and decentralized goals are then considered. A decentralized problem model consists of multiple smaller, independent subsystems in which each subsystem in an overall nonlinear system has its own independent goals and is represented by a discrete model of the form:

$$\begin{cases} x_l(k+1) = A_l x_l(k) + B_l u_l(k) \\ y_l(k) = C_l x_l(k) \end{cases} \tag{15}$$

Where  $x_l \in \mathfrak{R}^{n_x}$  is the local state space;  $y_l \in \mathfrak{R}^{n_y}$  is the measurement output of each subsystem;  $u_l \in \mathfrak{R}^{n_u}$  is the local control input. Therefore the overall nonlinear system can be seen as a collection of smaller subsystems that are completely independent from one another which is referred to as a decentralized model. The variable control of every agent sent to the nonlinear system consists of its agent's optimal input control given by minimizing local standard MPC cost function:

$$J_l = \sum_{N_1}^{N_2} \|y_l(k+j) - \text{Setpoint}(k+j)\|_{Q_l}^2 + \sum_{j=1}^{N_u} \|\Delta u_l(k+j|k)\|_{R_l}^2 \tag{16}$$

Where  $Q_l, R_l$  are suitable weighting matrices.

One of the advantages of the state-space representation is that it simplifies the prediction; the prediction for this model is given:

$$\hat{y}_l(k+i|k) = C_l (A_l^i \hat{x}_l(k|k) + \sum_{j=1}^i A_l^{i-j} B_l u_l(k+j|k)) \tag{17}$$

For local suitable matrix  $\Psi_l, \Gamma_l, \Theta_l$  and  $\Lambda_l$ , we can rewrite the local predictive model output for future time instants as:

$$Y_l(k) = \Psi_l x_l(k) + \Gamma_l u_l(k-1) + \Theta_l \Delta u_l(k) \tag{18}$$

Where

$$\Psi = \begin{bmatrix} C_l A_l \\ \vdots \\ C_l A_l^{N_u} \\ \vdots \\ C_l A_l^{N_u+1} \\ \vdots \\ C_l A_l^{N_2} \end{bmatrix} \quad \Theta_l = \begin{bmatrix} C_l B_l \\ \vdots \\ C_l \sum_{j=0}^{N_u-1} A_l^j B_l \\ \vdots \\ C_l \sum_{j=0}^{N_2-1} A_l^j B_l \end{bmatrix} \tag{19}$$



$$\Gamma_l = \begin{bmatrix} C_l B_l \cdots & \cdots & \cdots 0 \\ C_l (A_l B_l + B_l) & \cdots & \cdots 0 \\ C_l \sum_{j=0}^{N_u-1} A_l^j B_l & \cdots & C_l B_l \\ \cdots & \cdots & \cdots \\ C_l \sum_{j=0}^{N_2-1} A_l^j B_l & \cdots & C_l \sum_{j=0}^{N_2-N_u} A_l^j B_l \end{bmatrix}$$

The cost function (16) can be rewritten as:

$$J_l = \varepsilon_l(k)^T Q_l \varepsilon_l(k) - \Delta u_l(k)^T G_l + \Delta u_l(k)^T H_l \Delta u_l(k) \tag{20}$$

Where:

$$\begin{aligned} \varepsilon_l(k) &= \text{Setpoint}(k) - \Psi_l x_l(k) - \Gamma_l u_l(k-1) - \Lambda_l \\ G_l &= 2\Theta_l^T Q_l \varepsilon_l(k) \\ H_l &= \Theta_l^T Q_l \Theta_l + R_l \end{aligned} \tag{21}$$

Therefore the control law that minimizes the local cost function (16) is given by:

$$\Delta u_l(k) = \frac{1}{2} H_l^{-1} G_l \tag{22}$$

In order to take into account constraints on the manipulated variables, a transformation method for each action is made. The control action based on (22) is transformed into new action with the following transformation [R. Fletcher, 1997].

$$\left\{ \begin{aligned} u_l(k) &= f_{moy} + f_{amp} \tanh\left(\frac{u_l(k) - f_{moy}}{f_{amp}}\right) \\ f_{moy} &= \frac{f_{\max} - f_{\min}}{2} \\ f_{amp} &= \frac{f_{\max} + f_{\min}}{2} \\ f_{\max} &= \min(u_{l\max}, u_l(k-1) + \Delta u_{l\max}) \\ f_{\min} &= \max(u_{l\min}, u_l(k-1) + \Delta u_{l\min}) \end{aligned} \right. \tag{23}$$

The optimum control law (22) for each agent does not guarantee the global optimum. Accordingly to that, nonlinear system requires coordination among the control agent's action. The required coordination is done by a logic switch added to supervisory loop based neural networks which compute the global optimum control subject to constraints.

### 3.3 The supervisor loop

A neural network is used with the proper control architecture by changing the results of switched input  $u_i$  of each agent's action through a stable online NN weights which can guarantee the tracking performance of the overall closed-loop nonlinear system. Moreover, the neural network should reduce the deleterious effect of constraints attached with the different actions [Wenzhi, G et al., 2006]. In this work the neural network is represented by feed-forward single-input single output. The neural network tries to optimize the control action  $\Delta u$ .

$$\Delta u = f_{NN} \left( \sum_{i=1}^{n_u} b_i u(k-i) + \sum_{i=1}^{n_u} \sum_{j=1}^i b_{ij} u(k-i) u(k-j) \right) \quad (24)$$

The method of Levenberg Marquardt was designed for the optimization due to its properties of fast convergence and robustness. The main incentive of the choice of the algorithm of Levenberg Marquardt rests on the fast guarantee of the convergence toward a minimum.

### 4. Simulation results

The chosen example used in aim to valid the theory exposed above is given [B.Laroche et al.,2000]. A continuous state space representation of this example is as follow:

$$\begin{cases} \dot{x}_1 = x_3 - x_2 u \\ \dot{x}_2 = -x_2 + u \\ \dot{x}_3 = x_2 - x_1 + 2x_2(u - x_2) \end{cases} \quad (25)$$

The system model is implanted in the Matlab-simulink environment of which the goal is to get the input/output vector for the identification phase. Matlab® discrete these equations by the 4<sup>th</sup> order Runge-Kutta method. The vector characterizing the Volterra model that linking the output  $x_3$  with the input  $u$  is given by:

$$A = [1 \quad -1.9897 \quad .9997]^T, B_1 = [-0.0318 \quad -0.0096] \\ B_2 = \begin{pmatrix} 0.0396 & 0.0656 & 0 \\ 0 & 0.0388 & 0 \\ 0 & 0 & 0 \end{pmatrix} \quad (26)$$

Moreover, the Chiu procedure is developed to divide the nonlinear system into independent subsystem [Chiu S.L, 1994]. The modeling of the dynamic system, led to the localization of two centers with respective values  $c_1 = 0.0483, c_2 = 0.5480$ . The classification parameters adopted for the algorithm are as follows :  $r_a = .6; r_b = 1.25r_a; \varepsilon_1 = .5; \varepsilon_2 = .1$ . The procedure of identification and modeling has been applied to the whole measures input/output come out of the global system, driving to the different following subsystem models:

$$\begin{aligned} A_1 &= \begin{pmatrix} 0.552 & 1 \\ 0.2155 & 0 \end{pmatrix}, B_1 = \begin{bmatrix} 0.0496 \\ 0.1419 \end{bmatrix} \\ A_2 &= \begin{pmatrix} 0.7962 & 1 \\ 0.0481 & 0 \end{pmatrix}, B_2 = \begin{bmatrix} 0.0239 \\ -0.0088 \end{bmatrix} \end{aligned} \quad (27)$$

The result of modelisation is reported in figure 3. These results showed the application Chiu algorithm for the classification which has a better quality of local approximation of the system.

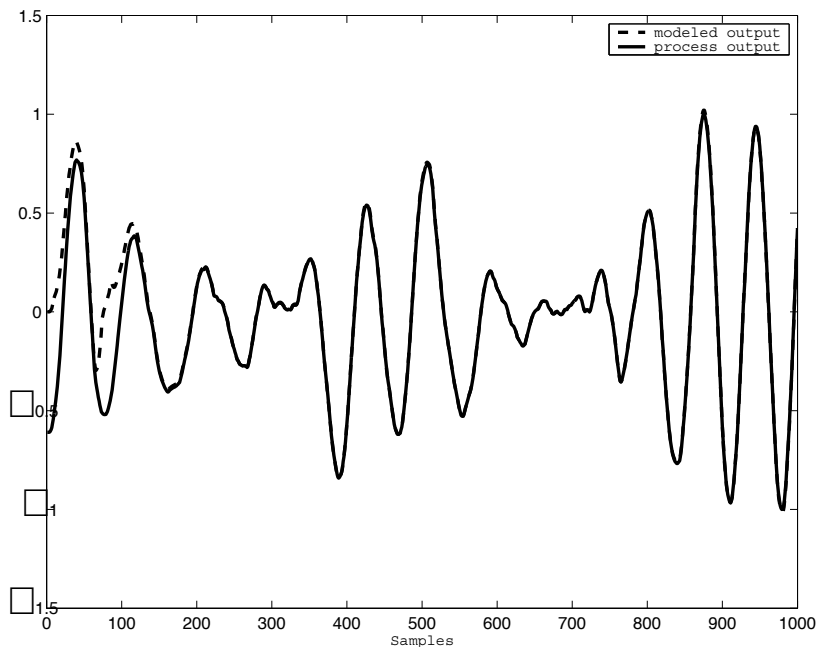


Fig. 3. Validation of the obtained model

#### 4.1 Set point tracking

The proposed concept as seen in section 3 is used, to control the nonlinear system. The tuning parameters of the multi-agent consists of the parameters values of each agent given by:  $N_1 = 1$  ;  $N_2 = 5$  ;  $N_u = 1$  ;  $R_1 = R_2 = 4$  ;. Assuming for the sake of simplicity but without loss of generality, the prediction and control horizons are the same for each agent. The tuning parameters for the NMPC are:  $N_1 = 1$  ;  $N_2 = 5$  ;  $N_u = 1$  ;  $\delta = .001$  . The gradient of the control  $\Delta u_{\min}$  and  $\Delta u_{\max}$  are taken, respectively, equal to  $-0.2$  and  $0.1$  and the control is limited between 0 and 1. In this application example, the neural network was a feedforward network and it consisted of three hidden layer nodes with tangent sigmoid transfer functions and one output layer node with linear transfer function .In this section, we present a comparative study between the proposed method and the NMPC procedure. The results shown in Fig. 4 and Fig. 5 are obtained in the constrained case

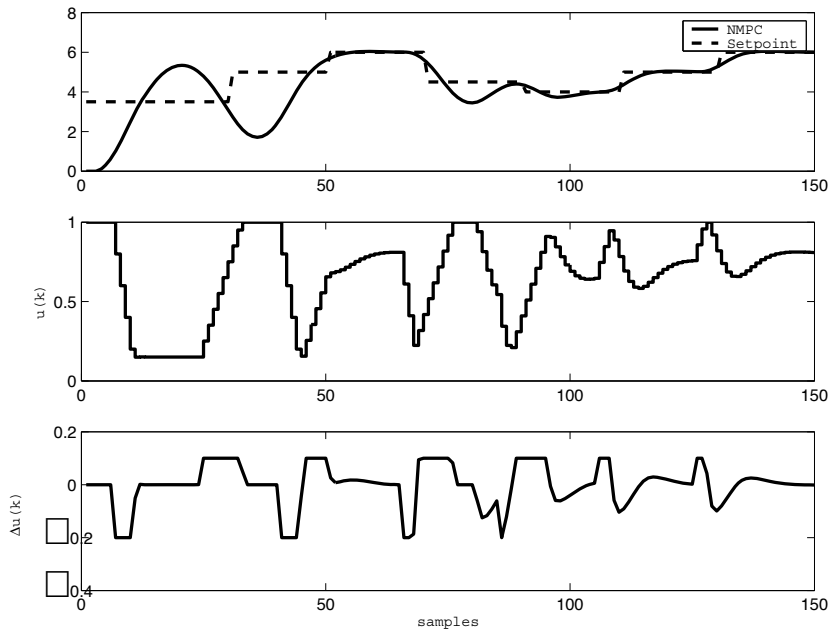


Fig. 4. Evolution of the set point, the output and the control (NMPC): constrained case

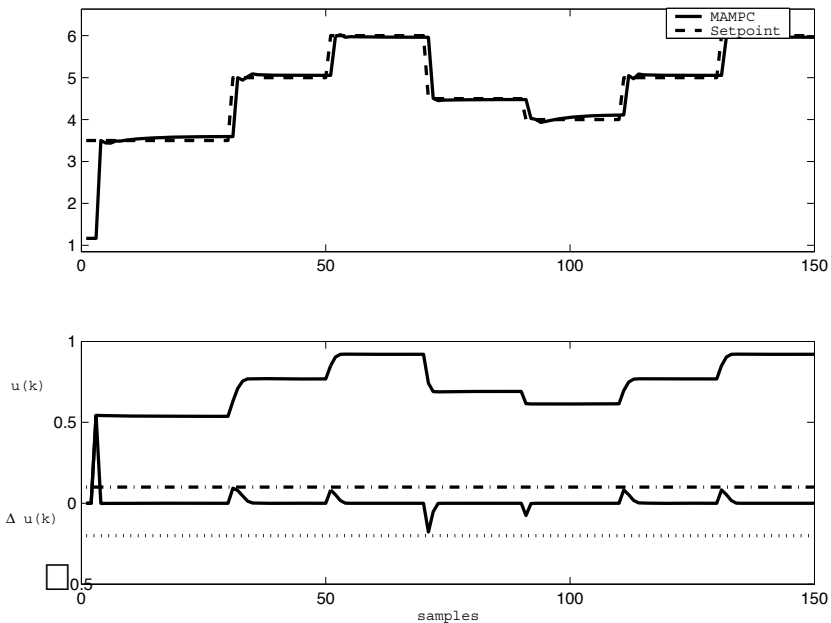


Fig. 5. Evolution of the set point, the output and the control (MAMPC): constrained case

It is clear from this figures that the new strategy of control leads to satisfactory results with respect to set-point changes. Indeed, the tracking error is reduced and with a smooth control action. It is shown that NMPC also gives consistently a good performance for the range examined. The two controllers are remarkably similar, which indicates that the MAMPC controller is close to optimal for this control problem. Moreover, the new controller meets all the required performance specifications within given input constraints and the results show a significant improvement in the system performance compared with the results obtained when only nonlinear programming model is used and the multi-agent compares favorably with respect to a numerical optimization routine as shown in Figure 6, the final control law to the nonlinear system obeying the specified constraints and with the proposed concept the constrained input and rate of change inputs cannot violate the specified range premise.

#### 4.2 Effect of load disruptions and noise

In order to test the effect of load disruptions, we have added to the system output a constant equals to 0.02 from iteration 100 to iteration 125 and from iteration 200 to iteration 225. And in the case of noise, we have added to the output of the process an uncertain pseudo-noise of maximal amplitude equal to 0.025. Figs. 6 and 7 present the evolutions of the set point, the outputs obtained, respectively, with the presence of load disruption and noise. Fig. 6 shows the evolutions of the set point, the outputs signals obtained with both NMPC and MAMPC control strategy. It is clear from this figure that the presence of load disruptions, from iteration 70 to iteration 90 and from iteration 120 to iteration 140, does not lead to a correct pursuit. Thus, the presence of load disruptions has more effect on NMPC control than the MAMPC strategy. Fig. 8 shows the evolutions of the set point, the outputs obtained with NMPC and MAMPC strategy. According to the obtained results, we notice that the MAMPC controller is capable to deliver a less fluctuate output than that obtained with NMPC approach.

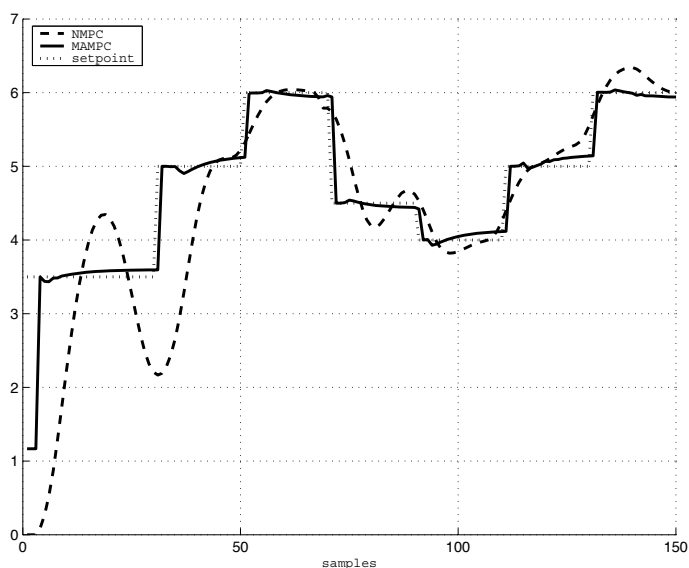


Fig. 6. Evolution of the set point, the output NMPC and MAMPC control in the case of load disruptions.

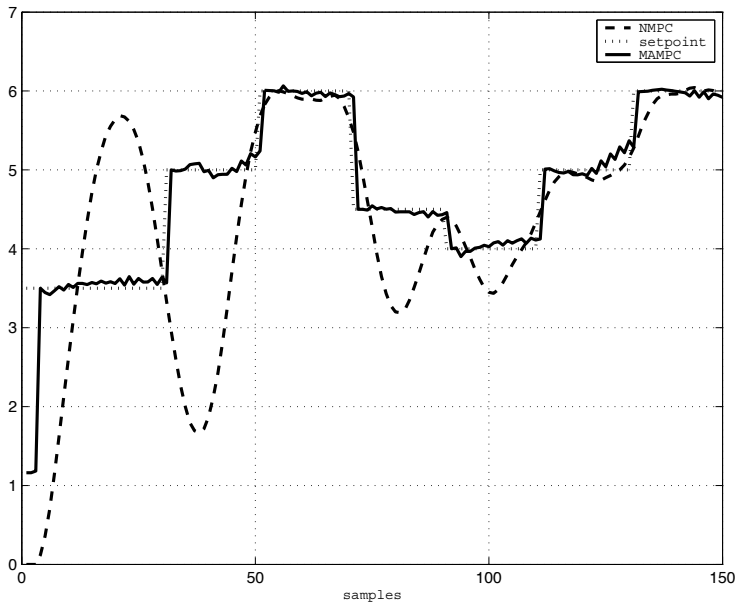


Fig. 7. Evolution of the set point, the outputs NMPC and MAMPC control in the case of the effect of the noise.

### 4.3 Convex optimization approach

In order to avoid solving nonconvex optimization problem, MAMPC optimization procedure, a method for convex NMPC was also developed in this chapter book. The performance of the proposed controllers is evaluated by applying to the same process and the attention has been focused on multi-agent model predictive control approach as a possible way to resolve non-convex optimization tasks. We have shown in Figure 8, a new constraint where the control is limited between 0 and .5. The nonlinear programming algorithm (NLP) cannot find a solution for the optimization problem. So because of the use of a nonlinear model, the NMPC calculation usually involves a non-convex nonlinear program, for which the numerical solution is very challenging. Therefore, finding a global optimum can be a difficult and computationally very demanding task, if possible at all. In other words, non-convexity makes the solution of the NLP uncertain. The proposed approach describe algorithm that find the solution of a non-convex programming.

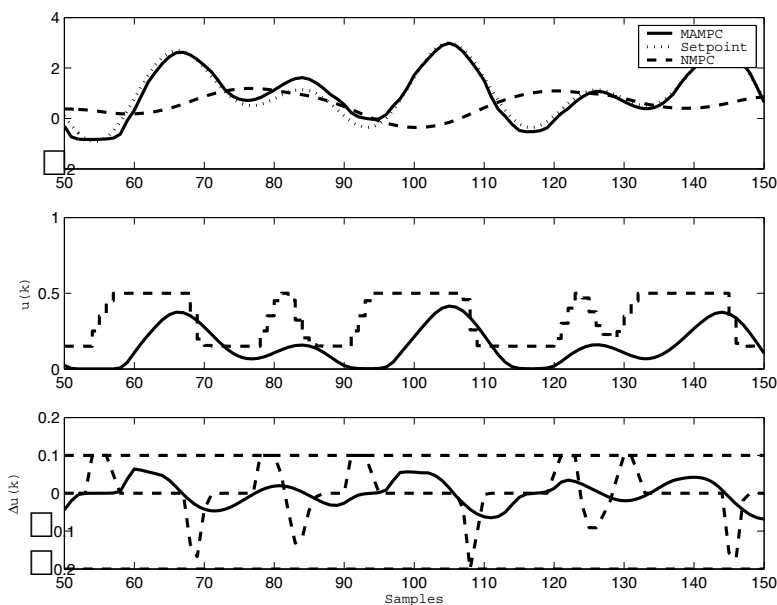


Fig. 8. Evolution of the set point, the outputs NMPC and MAMPC control: Restriction applicability of the NMPC

#### 4.4 Computational time study

The load computational time constitutes a gate in the scheme of the predictive control indeed (Leonidas et al, 2005). The performances of the computational load established with the proposed concept are compared to a nonlinear programming. In Figure 9 the time required to compute the control input at each time step  $k$  for the two approaches is plotted. We also reported in Table I, the mean and the maximum value of the implementation time required for the control law for the two cases. In Figure 10, the CPU time required to compute the control input at each time step  $k$  for the two approaches is plotted. It is very easy to see, from figure 9, 10 and table 1, that the NLMPC controller is too CPU time consuming and the computation for optimization in the new design procedure is simpler, faster and has good response curve and control performance because it uses a simple analytical solution to the minimization of the performance objective. On average, the NMPC method was about ten times slower than the novel approach and the control input in the MAMPC procedure require a twenty time smaller in the operating action.

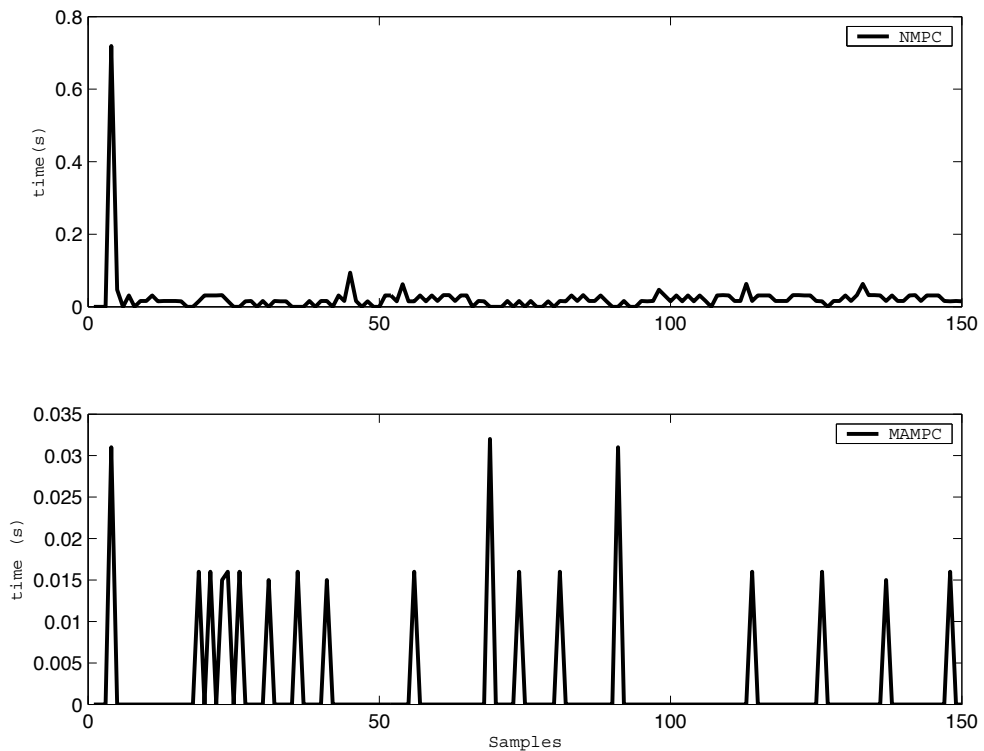


Fig. 9. Computational time requirement



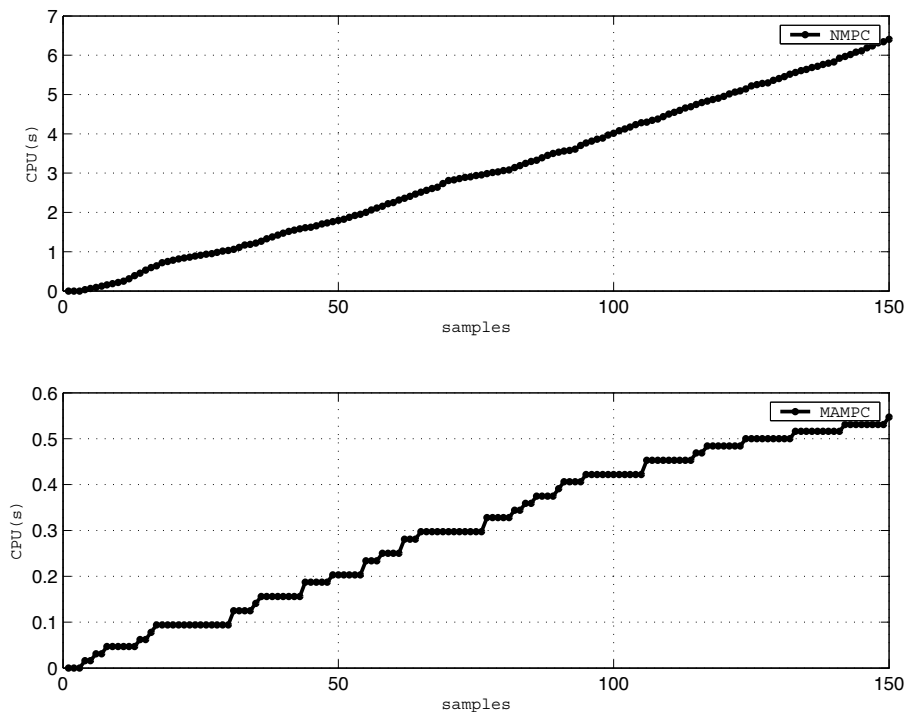


Fig. 10. CPU time comparison

	Mean	Max
NMPC	0.0224	0.7190
MAMPC	9.6875e-004	0.032

Table 1. Comparison of operating time

#### 4.5 Controller performance comparison

Through simulation-based comparisons, it is shown that a MAMPC control system is capable of delivering significantly improved control performance in comparison to a conventional NMPC, so that the difficulty of minimizing the performance function for nonlinear predictive control is avoided, which is usually carried by the use of NLP solved at each sampling time that generally is non-convex. Moreover, the nonlinear controls based on MAMPC approach provide excellent performance, both in terms of disturbance rejection, noise suppression and set point tracking. The NMPC controller is also good for disturbance rejection and noise suppression, but the set point tracking is not succeeded. In order to make a comparison of the novel concept to the NLMPC controller, the performance of the controller was measured by the following performance indices in unconstrained and constrained cases given by [Abonyi J,2003]:

$$\begin{aligned}
 SSE &= \sum_{k=1}^N (\text{Setpoint} - y(k))^2. \\
 SSU &= \sum_{k=1}^N (u(k) - u(k-1))^2. \\
 SS\Delta U &= \sum_{k=1}^N \Delta u^2(k).
 \end{aligned} \tag{28}$$

Where SSE denotes the sum of the square error, SSU the sum of the square of the control signal, SSΔU the sum of the square of the change of the control signal and N is the number of samples. The values are summarized in Table1, shows that the MAMPC achieving control performance improves more with the use of the NLMPC controller. Moreover the MAMPC produces the best tracking performance and the smallest energy consumption.

	Constrained case		
	SSE	SSU e-011	SSΔU e-013
NMPC	0.0135	5.5415	16754
MAMPC	0.0042	1.2464	3.8622

Table 2. Control performance comparison

## 5. Conclusions

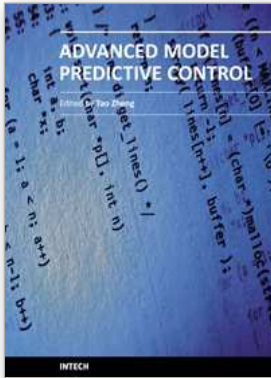
One of the main drawbacks of NMPC schemes is the enormous computational effort these controllers require. On the other hand, linear MPC methods can be implemented solving just Quadratic Programming (QP) or Linear Programming problems (LP). The main focus of this chapter is to develop a new control algorithm that in essence is a bridge between linear and nonlinear control. This resulted in the development of the MAMPC (Multiagent model predictive control) approach. The new NMPC scheme based MAMPC is implemented to reduce the computational effort. The control performance of MAMPC algorithm is

evaluated by illustrative comparison with general NMPC. All the results prove that MAMPC approach is a fairly promising algorithm by delivering significantly improved control. The performance of the proposed controllers is evaluated by applying to single input-single output control of non linear system. Theoretical analysis and simulation results demonstrate better performance of the MAMPC over a conventional NMPC based on sequential quadratic programming in tracking the setpoint changes as well as stabilizing the operation in the presence of input disturbances.

## 6. References

- Maciejowski J (2002), *Predictive Control with constraints*, Prentice Hall, London.
- Qin, S. J and Badgwell T.A A survey of industrial model predictive control technology, *Control Engineering Practice* 11 733-764.(2003)
- Kurtz M.J and.Henson M.A Input-output linearizing control of constrained nonlinear process, *Journal of Process Control*, 7(1) 3-17(1997).
- Arahal M.R and Berenguel M and Camacho E.F , Neural identification applied to predictive control of a solar plant, *Control Engineering Practice* 6 334- 344,(1998)
- Fischer M and.Nelles O , Predictive control based on local linear fuzzy models, *International Journal of system Science*, 29 679-697(1998).
- Mollov S., Babuska R., Abonyi J., and Henk B.. Effective Optimization for Fuzzy Model Predictive Control . *IEEE Transactions on fuzzy systems*, Vol. 12, No. 5,661-675 (2004)
- Brooms A and Kouvaritakis B, Successive constrained optimization and interpolation on non-Linear model based predictive control, *International journal of control*, 68 (3) 599- 623(2000)
- Abonyi, J and Babuska.R and Abotto M and Szeifert F and Nagy L. Identification and control of nonlinear systems using fuzzy Hammerstein models, *Ind.Eng.Chem.Res*, 39 4302-4314.(2000)
- Kanev, S and Vergaegen, M. "Controller e-configuration for non-linear systems", *Control Engineering Practice*, Vol. 8 No. 11,(2000) pp. 1223-35.
- Negenborn R.R, De Schutter B. and Hellendroom J Multi-agent model predictive control A survey, Technical report 04-010, Delf Center for systems and control.(2004)
- Soest Van W.R. and Q.P. Chu Q.P, and Mulder J.A Combined Feedback Linearization and Constrained Model Predictive Control for Entry Flight *Journal of Guidance, Control, and Dynamics*, 29(2) 427- 434 (2005)
- Jaochim H, Bock H.G and Diehl M An online active set strategy for fast parametric quadratic programming in mpc applications, *Proceeding IFAC Workshop on Nonlinear Model Predictive Control for fast Systems NMPC- FS'06, France*,pp 13-22.(2006),
- Didier G., Distributed model predictive control via decomposition- coordination techniques and the use of an augmented lagrangian. *IFAC. Workshop on NMPC, France*, 111-116(,2006)

- H.Ben Nasr and F.M'Sahli, A Computational Time Requirement Comparison Between Two Approaches In MBPC, *International Journal of soft computing*, 3 (2), pp:147-154 (2008a)
- H.Ben Nasr and F.M'Sahli, Une supervision d'action pour la commande prédictive multi-agent, 1<sup>st</sup> International Workshop on Systems Engineering Design & Applications,,SENDA (2008b).
- H.Ben Nasr and F.M'Sahli, Une nouvelle stratégie de commande prédictive des systèmes non linéaires à dynamique rapides. Conférence Internationale Francophone d'Automatique Bucarest, Roumanie, 3-5 septembre CIFA(2008c)
- H.Ben Nasr, M'Sahli F, A Multi-agent Approach To TS-Fuzzy Modeling and Predictive Control of Constrained Nonlinear System", *International Journal of Intelligent Computing and Cybernetics*. Vol,3,pp:398-424 (2008d)
- H.Ben Nasr, M'Sahli F, A Multi-agent predictive control approach based on fuzzy supervisory loop for fast dynamic systems. The fifth International Multi-Conference on Systems, Signal and Devices, SSD (2008e).
- D.W. Clarke, C.Mohtadi and P.S.Tuffs, Generalized predictive control. I: The basic algorithm II: Extensions and interpretations, *Automatica*, 23, pp. 137-160, 1987.
- Haber R and Bars R.and Engel L.(1999a) Three Extended Horizon Adaptive Nonlinear predictive Control Schemes on the Parametric Volterra Model, ECC, Proceedings of the European Control Conference, Karlsruhe, Germany
- Haber R and Bars R. and Engel L(1999b) Sub-optimal Nonlinear Predictive and Adaptive Control based on the Parametric Volterra Model, *Applied Mathematics and Computer Science* 70 161-173
- Matthew J. Tenny\_and James B. Rawlings TWMCC (Texas-Wisconsin Modeling and Control Consortium) Technical report number 2002-04
- F.J.Doyle, B.A.Ogunnike and R.K.Pearson, "Non-linear model-based control using seconds order Volterra models". *Automatica*.1995,Vol5, N31.pp.697-714.
- B.Laroche, M. Philippe and P. Rochon. "Motion planning for the heat equation", *international journal of robotic and nonlinear control*. pp. 629-643. 2000.
- R. Fletcher. *Practical Methods of Optimization*. JohnWiley and Sons, 1987.
- Wenzhi, G. and Rastko, R. , "Neural network control of a class of nonlinear systems with actuator saturation", *IEEE Trans. On Neural Networks*, 17 (1) (2006) 147-56.
- Chiu S.L., "Fuzzy Model identification based on cluster estimation", *Journal of Intelligent Fuzzy Systems* 2 (1994) 267-278.
- Leonidas, G.B. and Mayuresh, V.K. (2005), "Real-time implementation of model predictive control", *American Control Conference*, Portland, OR, pp. 4166-71.
- Abonyi J. (2003) *Fuzzy Model identification for control*, Birkhauser Boston.



## Advanced Model Predictive Control

Edited by Dr. Tao ZHENG

ISBN 978-953-307-298-2

Hard cover, 418 pages

**Publisher** InTech

**Published online** 24, June, 2011

**Published in print edition** June, 2011

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:

Bennasr Hichem and M'Sahli Faouzi (2011). Fast Nonlinear Model Predictive Control using Second Order Volterra Models Based Multi-agent Approach, 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/fast-nonlinear-model-predictive-control-using-second-order-volterra-models-based-multi-agent-approac>

# INTECH

open science | open minds

### InTech Europe

University Campus STeP Ri  
Slavka Krautzeka 83/A  
51000 Rijeka, Croatia  
Phone: +385 (51) 770 447  
Fax: +385 (51) 686 166  
[www.intechopen.com](http://www.intechopen.com)

### InTech China

Unit 405, Office Block, Hotel Equatorial Shanghai  
No.65, Yan An Road (West), Shanghai, 200040, China  
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元  
Phone: +86-21-62489820  
Fax: +86-21-62489821

© 2011 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the [Creative Commons Attribution-NonCommercial-ShareAlike-3.0 License](#), which permits use, distribution and reproduction for non-commercial purposes, provided the original is properly cited and derivative works building on this content are distributed under the same license.