# Enhancing Boost Converter Performance: A Comparison Between Multi-Model Predictive Control and Nonlinear Hammerstein Control

**Hichem Bennasr*, Ali Hmidene**, Faouzi M'Sahli*****

*Higher Institute of Technological Studies of Sfax, Department of Electrical Engineering, Tunisia (hichem2001tn@yahoo.fr)
**Higher Institute of Technological Studies of Sousse, Department of Electrical Engineering, Tunisia (ali.hmidene@gmail.com)
*** 3National Engineering School of Monastir, Department of Electrical Engineering, Tunisia (msahlifaouzi2015@gmail.com)

**Abstract**: In this paper, we present a comparative study of the multi-model approach and the nonlinear approach for predictive control of a Boost converter. The system is first Characterized by a collection of models. The optimal number of models is selected through the Elbow Method, Improved using Particle Swarm Optimization to improve partitioning efficiency. A genetic algorithm is used to refine the resulting models using the adequate validity. In contrast, predictive control combined with the Newton-Raphson method is applied to the Hammerstein model. Both control strategies are implemented on an STM32 microcontroller to enable real-time control of the Boost converter. The efficiency of the developed control strategies is analyzed and compared with respect to tracking precision, robustness, and execution time, demonstrating their suitability for power electronics applications.

*Keywords*: Multimodel, Hammerstein, Predictive Control, Particle Swarm, a genetic algorithm, STM32.

## 1. INTRODUCTION

One of the most prevalent advanced control techniques employed in modern industrial applications (Qin and Badgwell 2003; Kouro et al., 2015; Vidyasagar and Swarup 2023) is Model predictive control. Initially developed for controlling processes with slow dynamics, such as oil refineries and chemical plants, it has gradually become widespread due to its ability to optimize system performance while respecting constraints. Advances in control technology and numerical computation have enabled the application of predictive control to high-reactivity dynamic systems, where speed and precision of responses are critical. In this context, predictive control is increasingly seen as a central approach for controlling high-reactivity dynamic systems. This method relies on anticipating future system dynamics from models, allowing continuous calculation and adjustment of control actions to optimize system performance while respecting physical and operational constraints. Its flexibility and efficiency make it an essential solution today, representing a highly active research area in both academic and industrial sectors such as automotive, robotics, and aerospace (Swief et al., 2019). In these fields, predictive control plays a key role in meeting the growing demands for reactivity and reliability. For fast dynamic systems, which require near-instantaneous adjustments, the predictive control's ability to anticipate disturbances and adapt control actions in real-time is crucial. Compared to traditional control methods, often based on linear models, predictive control allows for managing nonlinearities and constraints, thus offering greater flexibility and efficiency. The importance of this method in modern industrial settings is undeniable, as it helps meet the increasing demands for performance, safety, and robustness in dynamic systems found in various sectors such as aerospace, robotics, automotive, and industrial automation. However, implementing predictive control in fast dynamic contexts presents technical challenges, especially when these systems are controlled by microcontrollers with limited computing capabilities. Microcontrollers, though common and inexpensive, impose processing, storage, and energy consumption constraints, requiring optimized predictive control solutions that are both lightweight and efficient. This paper aims to develop a predictive control algorithm specifically tailored for fast dynamic systems, taking into account the limitations inherent to microcontrollers. The main objective is to explore approaches that address speed and accuracy constraints while remaining compatible with the limited capabilities of microcontrollers used for real-time control. In other context the multimodel predictive control become an attractive strategy which can minimize the computation time required to implement the control law. The multimodel approach has attracted growing interest since Johan's work (Johansen, 1994). This method involves representing the non-linear dynamics of the systems through a set of region-specific models, which can be linear or affine, that describe the system in different regimes of operation. The main advantage of this approach is that it provides a simplified and clear representation of a non-linear system, while facilitating design and design of controllers. One of the most interesting features of the multimodel approach is the possibility approximate a non-linear system using a weighted combination of local models and their weighting is done by validity or membership functions (Delmotte et al., 1996; Li et al., 2004; Roubos et al., 1998), which characterize the "area of action" of each model. From the perspective of multimodel predictive control several studies have been performed (He et al., 2005; Adhemar de et al., 2007; Guo et al., 2003; Özkan et al., 2003). The main advantages of this methods are that they transform a nonlinear and non-convex optimization problem into a convex problem guaranteeing thus the convergence of the solution. Resent research into multimodel predictive

control for fast systems have been presented in previous studies, including those referenced in (Porfírio et al., 2003; Bennasr 2009; Nasr and M'Sahli, 2008; Nasr Hichem and Faouzi, 2008; Bennasr and M'Sahli, n.d.; Bennasr and M'Sahli, 2008; Hichem and Faouzi, 2008; Hmidene et al., 2024). The main challenge in this structure is the optimization of the local number that describe the process and the validity computation for each model which are studied in many works (Hichem and Faouzi, 2021; Bennasr and M'Sahli, 2024; Ltaief et al., 2008; Messaoud et al., 2017; El Kamel et al., 2000; Ben Mohamed et al., 2011; Elfelly et al., 2010; Elfelly et al., 2008; Messaoud et al., 2018). In this work we address to the two challenges by giving new structure of multimodel that in order to minimize the computation load.

In this paper were present two approaches for implementing nonlinear predictive control. The first relies on a multimodel predictive control strategy. A novel design for the model library formulation based on a genetic algorithm with a multiple model validity choice, this approach aims to avoid nonlinear optimization, significantly reducing computational load. The second approach involves implementing nonlinear predictive control with constraints using the Hammerstein model, with simplifying assumptions applied during the prediction phase. For this method, optimization is performed using the Newton-Raphson method.

## 2. PREDICTIVE FORM OF THE MULTIMODEL

The proposed multi-model representation scheme is illustrated in Figure 1.

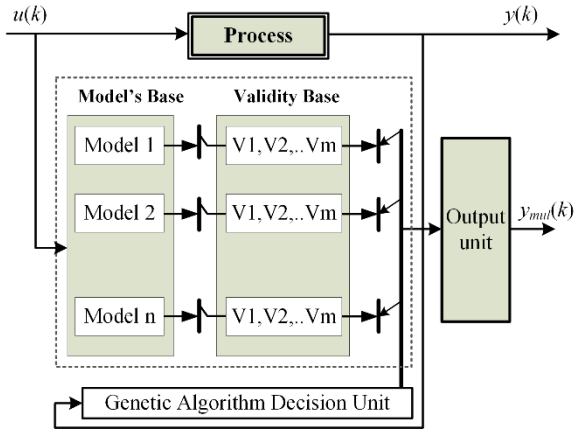


Fig. 1. Multimodel representation.

This structure consists of two blocks: one dedicated to the construction of the sub-model library and the other focused on validity calculation. Each model is associated with three types of validity: a simple validity, a reinforced validity, and a combined validity that integrates both. The choice of the combined validity arises from research findings by (Hichem and Faouzi, 2018), which indicates that, with a certain degree of optimization, it is possible to effectively combine the two types of validity. However, the integration of all validities for each model can become computationally intensive during online calculations. Therefore, we have opted to select one of the three validities for each model to ensure adequate modeling results. This selection process is allocated to a genetic algorithm procedure, which optimizes a specific cost function, determining for each sub-model which validity is most relevant. The objective is to ensure that the sum of these sub-models, weighted by their chosen validities, accurately approximates the process.

The expressions for these normalized validities are given by the following equations:

$$\begin{cases} v_1(k) = {1 - r_i^{norm}(k)}/{N_m - 1} \ , \ i = 1, \cdots, N_m \\ v_2(k) = v_1(k) \prod_{j=1, j \neq i}^{N_m} \left(1 - v_1(k)\right), \ i = 1, \cdots, N_m \\ v_{3(k)} = v_1(k) + v_2(k) \end{cases} \quad (1)$$

Where,

$v_1(k), v_2(k)\ and\ v_3(k)$ respectively denote simple validity, reinforced validity, and combined validity.

$r_i^{norm}$ is the normalized form of the residue expressed by:

$$r_i^{norm}(k) = {1 - r_i(k)}/{\sum_{i=1}^{N_m} r_i(k)} \ , \ i = 1, \cdots, N_m \quad (2)$$

### *2.1 K-means Elbow Method for Optimal Model Selection through Practical swarm optimization*

Determining the optimal number of sub-models is a significant challenge. We present an innovative approach that integrates the K-means Elbow Method with Particle Swarm Optimization. This technique intends to enhance the accuracy of the sub-model selection process. K-means is an unsupervised learning technique that automatically groups similar data points into *k* clusters by minimizing intra-cluster variance, where each cluster is represented by a centroid. K-means aims to minimize the inertia, which quantifies the total squared Euclidean distance. The K-means algorithm consists of several key steps to identify the optimal clusters in a dataset (Kodinariya and Makwana, 2013; Yuan and Yang, 2019). The algorithm begins with the initialization phase, where initial centroids are selected. Next, Each point is grouped with the centroid closest to it, determined by the distance calculated using the following equation:

$$argmin \left\| x_i - c_j \right\|^2 \quad (3)$$

Where: $x_i$ is a data point, $c_j$ is the centroid of the $j-th$ cluster.

After this point assignment, Centroid positions are updated by averaging the data points within each cluster, as illustrated in the following equation:

$$c_j = \frac{1}{C_j} \sum_{x_i \in c_j} x_i \quad (4)$$

Where $c_j$ represents the set of points assigned to cluster $j$, and $C_j$ denotes the number of points within that cluster.

This iterative process continues until convergence. The measure of compactness of the clusters is determined by inertia, which is the sum of squared distances between each data point and its corresponding centroid. In mathematical terms, inertia for a data point in a specific cluster is calcula1ted by :

$$SE(k) = \sum_{i=1}^{n}\sum_{j=1}^{k} \left\| x_i^{(j)} - c_j \right\|^2 \quad (5)$$

Where $x_i^{(j)}$ is the data point $i$ in cluster $j$ and $c_j$ is the centroid of cluster $j$.

The Elbow Method is used. This involves searching the inertia values for different values of the number of clusters to identify the "elbow" point where the rate of decrease in inertia slows down (Bholowalia and Kumar, 2014; Humaira and Rasyidah, 2020). The value at this elbow point is typically considered the optimal number of clusters for the dataset.

In the second part a Particle Swarm Optimization (PSO) is then used to fine-tune the choice of k by minimizing an objective function. The goal is to improve the choice of the number of clusters by applying particle optimization in the search space. The objective is to minimize the following function:

$$f(k) = \sum_{i=1}^{n}\sum_{j=1}^{k} \left\| x_j^{(j)} - c_j \right\|^2 \quad (6)$$

PSO works by adjusting the positions of the particles which represent different candidate solutions of different values of k to find the optimal value of k that minimizes the inertia. So, the PSO updates the positions of particles based on their velocities and personal and global experiences. This is done according to the following equations:

$$\begin{aligned} x_i^{(t+1)} &= x_i^{(t)} + v_i^{(t+1)} \\ v_i^{(t+1)} &= w v_i^{(t)} + c_1 r_1 (p_i^{(t)} - x_i^{(t)}) + c_2 r_2 (g^{(t)} - x_i^{(t)}) \end{aligned} \quad (7)$$

Withe :

$w$ is the inertia weight, $c_1$ and $c_2$ are acceleration coefficients, $r_1$ and$r_2$ are random values between 0 and 1, $p_i^{(t)}$ is the personal best position of particle and $g(t)$ is the global best position.

The novel approach adopted allows for a more precise by combining the simplicity of the Elbow Method and the optimization power of PSO in order to confirm the optimal cluster number. So, the procedure consists of using K-means to calculate inertia for different values of k and calculate the inertia to identify the "elbow". Then, applying PSO to optimize the choice of k by refining the search for the optimal number of clusters. PSO operates by adjusting particle positions in the search space of k while using inertia as the objective function to minimize.

*2.2 Cluster Modeling*

After determining the number of models, the data from each cluster is used to define a model for each one. This model takes the form :

$$y_{local}(k) = c_0 + \sum_{i=1}^{i_{lag}} a_i y(k-i) + \sum_{j=1}^{j_{lag}} b_j u(k-j) \quad (8)$$

Where: $c_0$ is a constant, $a_i$ and $b_j$are the model coefficients $i_{lag}$and $j_{lag}$ are the time lags for the local output and input respectively. The identification of the appropriate time lags for building the accurate dynamic local models that describe each cluster is focus in this study on using a cross-validation to determine the optimal time lags for each local dynamic model where the output variable $y_{local}$ depends on both its past values and the past values of an input variable $u_{local}$. In fact, the data is split into k subsets. At each iteration, a different subset of the data is reserved for validation, while the remaining portion is used for training. The model is trained on the training set, and its prediction error is evaluated using the validation set.

This process is repeated for all subsets, and the overall cross-validation error is computed as the mean square error given by :

$$MSE_{localmodel} = \frac{1}{N_{val}} \sum_{k=1}^{N_{val}} (y_{localpredictedvalue}(k) - y_{validation}(k))^2 \quad (9)$$

Where $y_{loaclpredictedvalue}(k)$ and $y_{validation}(k)$ are the predicted local value and the true values in the validation set. The optimal values of $i_{lag}$ and $j_{lag}$ are selected as the ones that minimize the average cross-validation error.

*2.3 Genetic selection procedure*

The mere availability of a set of local sub-models to represent a nonlinear system is not sufficient, as determining the validity of each model is also a critical aspect. Therefore, we aim to identify, among the set of candidate models, the combination that most accurately represents the dynamics of the nonlinear system. To this end, every individual model is assigned a specific domain of validity, and a competition is initiated through a genetic algorithm, which selects the most appropriate local model whose combination leads to a more accurate mathematical representation of its dynamics. A genetic algorithm (GA) is used in this study to perform the optimization for the selection of local models representing the process. Each individual in the GA population is encoded as a binary string of fixed length, where each bit corresponds to the inclusion equal to 1 or exclusion equal to 0 of a specific candidate model. The initial population, composed of a set of randomly generated individuals, evolves over successive generations. The quality of each individual is evaluated using a cost function defined as the RMSE between the true and predicted outputs of the system reconstructed by the sum of the selected models. Each individual in the population is encoded as a binary vector $x = [x^1, x^2, \cdots, x^m]$ where $x_i = 1$ indicates that the $i^{th}$ local model is selected, and $x_i = 0$ otherwise. The population is initialized with $P$ such vectors randomly generated. For each individual, the predicted output is computed as the sum of the selected models. A binary-coded GA is used to iteratively evolve model combinations that best approximate the output of the system. The selection frequency of each model across multiple runs is used as an indicator of its contribution to optimal solutions. The selected model is done by obeying:

$$RMSE(x) = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(y - \sum_{k=1}^{m} x_k M_{i,k}\right)^2} \quad (10)$$

Where $y$ is the real output, $m$ the number of model, $x_k$ is a binary number, $M \in \Re^{n\times m}$ and $M_{I,k}$ is the predicted by model $j$ at data point $i$.

This fitness function guides the evolutionary process. During each generation, individuals with lower cost values are more likely to be selected for reproduction. Pairs of selected individuals undergo crossover, where segments of their binary strings are exchanged to create new offspring, promoting the exploration of new model combinations. Random mutations are applied to preserve population diversity and avoid premature convergence by flipping bits with a low probability. By iteratively applying selection, crossover, and mutation, the GA converges to an optimal solution or near-optimal set of models that collectively minimize the prediction error, thereby identifying the most relevant components for approximating the nonlinear system dynamics. Then, the probability of selection of individual xi is proportional to its fitness:

$$fitness(x) = \frac{1}{1 + RMSE(x)} \tag{11}$$

In this genetic algorithm, selection is performed using MATLAB's default stochastic uniform method.

$$P(x_i) = \frac{fitness(x_i)}{\sum_{j=1}^{m} fitness(x_j)} \tag{12}$$

In this implementation, a single-point crossover operator is used. In fact, during reproduction, a random crossover point is selected along the binary chromosome, and the offspring inherits genes from the first parent before that point and from the second parent after it. The mutation operator used is bit-flip mutation, suitable for binary chromosomes. Each bit in the individual has a small probability of being flipped which introduces genetic diversity into the population. To assess model robustness and importance, the GA is executed at each run produces an optimal selection $x^{(r)}$ .The cumulative inclusion frequency which is converted to percentage of each model is calculated. This percentage reflects how often each model was part of near-optimal local model serving as an importance metric. Termination of the algorithm occurs upon reaching the predefined maximum number of generations..

*2.4 Multimodel control algorithm*

Each model's base structure is specified in Equation (15), is used to search an explicite solution given by the minimization of :

$$J = \sum_{n_e=1}^{H_p} \left(r(k+n_e) - \hat{y}(k+n_e|k)\right)^2 + \lambda \sum_{j=1}^{H_c} \Delta u^2(k+j-1) \tag{13}$$

The prediction of a local output is given by successive iteration is given by:

$$y_i(k+j) = C_i A^J x_i(k) + \sum_{m=0}^{j-1} C_i A_i^{-1-m}(B_i u_i(k+m) + a_i) + c_i \tag{14}$$

Where $A_i, C_i, B_i, a_i \; and \; c_i$ are given by the corresponding discret representation of the local model.

$$\begin{cases} x(k+1) = A_i x_i(k) + B_i u(k) + a_i \\ y_i(k) = C_i x_i(k) + c_i \end{cases} \tag{15}$$

Taking into account equation given by (15), the local predicted output id given by:

$$Y_i(k) = \Psi_i x_i(k) + \Phi_i U_i(k) = \Theta_i \tag{16}$$

Where,

$$\Psi_i = \begin{bmatrix} C_i B_i & 0 & \cdots & 0 \\ C_i A_i B_i & C_i B_i & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ C_i A_i{}^{H_p-1} B_i C_i A_i{}^{H_p-2} B_i \cdots C_i \sum_{m=H_u}^{H_p-1} A_i{}^{H_p-m-1} B_i \end{bmatrix}$$

$$\Phi_i = \begin{bmatrix} C_i A_i \\ C_i A_i^2 \\ \vdots \\ C_i A_i{}^{Hp} \end{bmatrix}, \quad \Theta_i = \begin{bmatrix} c_i + Ca_i \\ c_i + CAa_i \\ \vdots \\ c_i + \sum_{m=0}^{H_p-1} C_i A_i^m a_i \end{bmatrix} \tag{17}$$

And the optimal solution is defined by :

$$U_i(k) = -(\Psi_i^T \Psi_i + 1)^{-1} \Psi_i^T (\Phi_i x_i + \Theta_i - r_i) \tag{18}$$

## 3. PREDICTIVE FORM OF THE HAMMERSTEIN MODEL

The Hammerstein model (Narendra and Gallman, 1966), is widely applicable for modeling a variety of nonlinear systems commonly found in industrial . Alternative approaches to the design of nonlinear predictive control of a hammerstein model algorithms have been explored (Fruzzetti et al., 1997; Xu et al., 2023; Huo et al., 2008). However, most existing methods minimize performance functions using nonlinear programming techniques to compute future manipulated variables through online optimization. This often makes the real-time implementation of such algorithms quite challenging. In this paper, we develop a suboptimal solution aimed at simplifying the control algorithm, which is otherwise computation ally intensive.

A second-order process is developed by :

$$A(q^{-1})y(k) = B(q^{-1})v(k-1) \tag{19}$$

With,

$$\begin{aligned} v(k) &= c_0 + c_1 u(k) + c_2 u^2(k) \\ (q^{-1}) &= 1 + a_1 q^{-1} + a_2 q^{-2} \\ B(q^{-1}) &= b_0 + b_1 q^{-1} \end{aligned} \tag{20}$$

By expanding equation (19), we obtain:

$$y(k) = c_p - a_1 y(k-1) - a_2 y(k-2) + b_{11} u(k-1) + b_{12} u(k-2) + b_{21} u^2(k-1) + b_{22} u^2(k-2) \tag{21}$$

With,

$$c_p = c_0(b_b + b_1)$$
$$B_1(q^{-1}) = c_1 B(q^{-1}) = b_{11} + b_{12}q^{-1}$$
$$B_2(q^{-1}) = c_2 B(q^{-1}) = b_{21} + b_{22}q^{-1}$$

The incremental form of the predictive model depends on both the current control increment and the future increments, represented by $\Delta u(k+i)$, $i = 0, \cdots, H_c - 1$. The current control increment as well as the future increments is expressed as follows:

$$u(k+i) = u(k-1) + \sum_{j=0}^{i} \Delta u(k+j) \tag{22}$$

To simplify the implementation of nonlinear optimization , we adopt a unit control horizon ($H_c = 1$).

The predicted output $\hat{y}(k+i)$ over a prediction horizon $H_p$ is calculated recursively as

$$\hat{y}(k+i) = c_p - a_1 y(k) - a_2 y(k-1) + b_{11}u(k) + b_{12}u(k-1) + b_{21}u^2(k) + b_{22}u^2(k-1) \tag{23}$$

By replacing $u(k+i) = u(k-1) + \Delta u(k)$ and using matrix notation we obtain :

$$\underbrace{\begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ a_1 & 1 & 0 & \cdots & 0 \\ a_2 & a_1 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \cdots & a_2 & a_1 & 1 \end{bmatrix}}_{\boldsymbol{C_A}} \underbrace{\begin{bmatrix} \hat{y}(k+1) \\ \hat{y}(k+2) \\ \vdots \\ \hat{y}(k+H_p) \end{bmatrix}}_{\boldsymbol{Y}} =$$
$$\underbrace{\begin{bmatrix} c_p \\ \vdots \\ c_p \end{bmatrix}}_{\boldsymbol{C_0}} + \underbrace{\begin{bmatrix} -a_1 & -a_2 \\ -a_2 & 0 \\ 0 & 0 \\ \vdots & \vdots \\ 0 & 0 \end{bmatrix}}_{\boldsymbol{H_A}} \underbrace{\begin{bmatrix} y(k) \\ y(k-1) \end{bmatrix}}_{\boldsymbol{Y_{past}}}$$
$$+ \underbrace{\begin{bmatrix} b_{11} + b_{12} \\ \vdots \\ b_{11} + b_{12} \end{bmatrix}}_{\boldsymbol{C_{b_1}}} u(k-1) + \underbrace{\begin{bmatrix} b_{21} + b_{22} \\ \vdots \\ b_{21} + b_{22} \end{bmatrix}}_{\boldsymbol{C_{b_2}}} u^2(k-1) \tag{24}$$
$$+ \underbrace{\begin{bmatrix} b_{11} + 2b_{21}u(k-1) \\ b_{11} + b_{12} + 2(b_{21} + b_{22})u(k-1) \\ \vdots \\ b_{11} + b_{12} + 2(b_{21} + b_{22})u(k-1) \end{bmatrix}}_{\boldsymbol{H_{b_{21}}}} \Delta u(k)$$
$$+ \underbrace{\begin{bmatrix} b_{21} \\ b_{21} + b_{22} \\ \vdots \\ b_{21} + b_{22} \end{bmatrix}}_{\boldsymbol{H_{b_{22}}}} \Delta u^2(k)$$

From equation (24), the vector of predicted outputs is written as :

$$\boldsymbol{Y} = \boldsymbol{C_{p0}} + \boldsymbol{G Y_{past}} + \boldsymbol{\beta_1} u(k-1) + \boldsymbol{\beta_2} u^2(k-1) + \boldsymbol{\gamma_1} \Delta u(k) + \boldsymbol{\gamma_2} \Delta u^2(k) \tag{25}$$

With,

$\boldsymbol{C_{p0}} = \boldsymbol{C_A^{-1} C_0}$, $\boldsymbol{G} = \boldsymbol{C_A^{-1} H_A}$, $\boldsymbol{\beta_1} = \boldsymbol{C_A^{-1} C_{b_1}}$, $\boldsymbol{\beta_2} = \boldsymbol{C_A^{-1} C_{b_2}}$, $\boldsymbol{\gamma_1} = \boldsymbol{C_A^{-1} H_{b_{21}}}$ and $\boldsymbol{\gamma_2} = \boldsymbol{C_A^{-1} H_{b_{22}}}$

As a result, the prediction equation at step j is expressed as follows:

$$\hat{y}(k+j|k) = y_{libre}(k+j|k) + \gamma_{1j}\Delta u(k) + \gamma_{2j}\Delta u^2(k) \tag{26}$$

With,

$$y_{libre}(k+j|k) = C_{poj} + G_j Y_{past} + \beta_{1j} u(k-1) + \beta_{2j} u^2(k-1)$$

The predicted outputs from equation (26) do not take into account the last measurement $y(k)$. As a result, the model's accumulated inaccuracies, as well as unmeasured disturbances, may lead to inaccurate predictions. However, integrating the last measurement into the prediction calculation helps improve its accuracy. This approach, generally referred to as output feedback (Swief et al., 2019), involves adding a correction bias $d(k)$ to the prediction. In practice, this bias is often defined as the difference between the last measurement $y(k)$ and the corresponding predicted value $\hat{y}(k|k-1)$, with the assumption that it remains constant for the various prediction outputs.

$$d(k) = d(k+j) = y(k) - \hat{y}(k|k-1) \tag{27}$$

This correction bias can be integrated into the open-loop output equation as :

$$y_{libre}(k+j|k) = C_{poj} + G_j Y_{past} + \beta_{1j} u(k-1) + \beta_{2j} u^2(k-1) + d(k) \tag{28}$$

Equation (13), which Can be described by a fourth-degree such as:

$$J = a_4 \Delta u^4(k) + a_3 \Delta u^3(k) + a_2 \Delta u^2(k) + a_1 \Delta u(k) + a_0 \tag{29}$$

With,

$$a_0 = \sum_{j=1}^{H_p} (y_{libre}(k+j|k)) - r(k+j|k))^2$$
$$a_1 = 2\sum_{j=1}^{H_p} (y_{libre}(k+j|k)) - r(k+j|k)) \cdot \gamma_{1j}$$
$$a_2 = \sum_{j=1}^{H_p} 2(y_{libre}(k+j|k)) - r(k+j|k)) \cdot \gamma_{2j} + \gamma_{1j}^2 + \lambda \tag{30}$$
$$a_3 = 2\sum_{j=1}^{H_p} \gamma_{1j} \cdot \gamma_{2j}$$
$$a_4 = \sum_{j=1}^{H_p} \gamma_{2j}^2$$

In the absence of constraints, minimizing the cost function may lead to a suboptimal solution. This solution is determined by :

$$\frac{\partial J}{\partial \Delta u(k)} = 2\sum_{j=1}^{H_p}\left(\gamma_{1j} + 2\gamma_{2j}\Delta u(k)\right)\cdot\left(y_{libre}(k+j|k) + \gamma_{1j}\Delta u(k) + \gamma_{2j}\Delta u^2(k) - r(k+j|k)\right) + 2\lambda\Delta u(k) = 0 \quad (31)$$

The search then is a solution of a third-order equation.

$$k_3\Delta u^3(k) + k_2\Delta u^2(k) + k_1\Delta u(k) + k_0 = 0 \quad (32)$$

With,

$$\begin{aligned} k_0 &= \sum_{j=1}^{H_p}\left(y_{libre}(k+j|k) - r(k+j|k)\right) + \gamma_{1j} \\ k_1 &= \sum_{j=1}^{H_p}\left(2\left(y_{libre}(k+j|k) - r(k+j|k)\right)\cdot\gamma_{2j} + \gamma_{1j}^2\right) + \lambda \\ k_2 &= \sum_{j=1}^{H_p}\gamma_{1j}\cdot\gamma_{2j} \\ k_3 &= 2\sum_{j=1}^{H_p}\gamma_{2j}^2 \end{aligned} \quad (33)$$

The Newton-Raphson method is an iterative technique widely employed for solving optimization problems and nonlinear equations due to its rapid convergence properties (Dang and Aschemann, 2021; Chen et al., 2023). In the context of optimization, this method relies on a local quadratic approximation of the objective function $J$ . The Newton-Raphson algorithm, incorporating input constraints, updates $\Delta u(k)$ at each iteration according to the relation given by Equation (34).

$$\Delta u^{n+1}(k) = \Delta u^n(k) - \left(\frac{\partial^2 J}{\partial^2 \Delta u(k)}\Big|_{\Delta u(k)=\Delta u^n(k)}\right)^{-1}\cdot\left(\frac{\partial J}{\partial \Delta u(k)}\Big|_{\Delta u(k)=\Delta u^n(k)}\right)$$

$$\begin{cases}\Delta u_{min}^{(k)} = \max(u_{\min} - u(k-1), \Delta u_{lb}) \\ \Delta u_{max}^{(k)} = \min(u_{max} - u(k-1), \Delta u_{ub})\end{cases} \quad (34)$$

$$\Delta u^{n+1}(k) = \begin{cases}\Delta u_{min}^{(k)} & if \quad \Delta u^{n+1}(k) < \Delta u_{min}^{(k)} \\ \Delta u^{n+1}(k) if \Delta u_{min}^{(k)} \le \Delta u^{n+1}(k) \le \Delta u_{max}^{(k)} \\ \Delta u_{max}^{(k)} & if \quad \Delta u^{n+1}(k) < \Delta u_{max}^{(k)}\end{cases}$$

With $\frac{\partial J}{\partial \Delta u(k)}$ and $\frac{\partial^2 J}{\partial^2 \Delta u(k)}$ representing the Jacobian and Hessian matrices, respectively, and n denoting the iteration number.

## 4. EXPERIMENT RESULTS

The above study is developed in order to control a Boost converter. The bloc scheme of the process is given by Figure 2 and the system to be identified has the following characteristics :

- Supply voltage : Vg=12 V
- Output voltage : Vo=24 V
- Switching frequency : 80 kHz
- Load resistance : R=50 Ω
- Inductance: L=430 µH
- Capacitor value : C=440 µF
- Capacitor series resistance: Rc=80 mΩ

We will begin by studying our approach for modeling a converter for which experiments have been conducted in the laboratory to acquire the full input-output dataset.

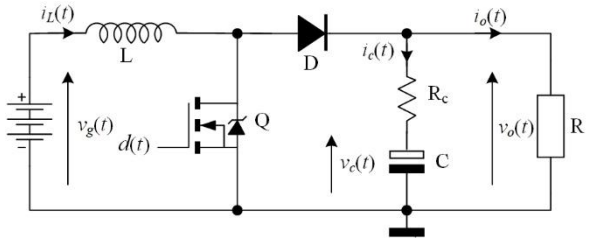


Fig. 2. Bloc diagram Boost Converter.

Black-box modeling is an experimental method that relies on data to determine the transfer function of the system. The identification test involves superimposing a low-amplitude pseudo-random sequence onto the system's input (Forsyth and Mollov, 1998). This sequence, which simulates white noise with a zero mean, preserves the system's operating point. The experimental data obtained during this test are presented in Figure 3.

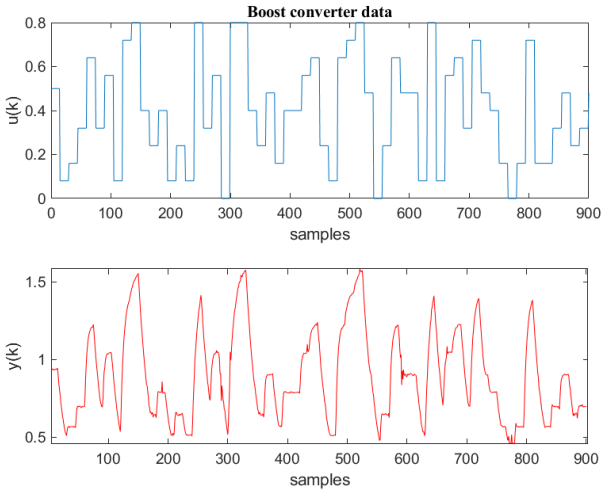


Fig. 3. Experimental data collection.

Figure 4 presents the simulation results obtained through the application of the Elbow method combined with Particle Swarm Optimization (PSO) for cluster number optimization, highlighting the convergence of the number of clusters to the value 3. This indicates that the system can be described by 3 sub-models.

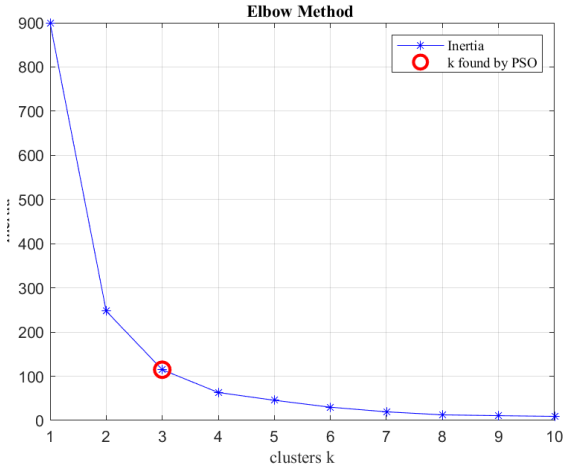


Fig. 4. Cluster Number Optimization: K-Means Elbow Method with PSO-Based Validation.

Furthermore, the data collection for each cluster underwent the modeling method for the different clusters, allowing us to observe the variations of the different lags of each sub-model in relation to the error Figure 5. These results led to sub-models provided by the equations given by :

$$\begin{aligned} Model_1: y(k) &= 0.0150 + 0.45y(k-3) \\ &\quad +0.25u(k-2) \\ Model_2: y(k) &= 0.0541 + 0.3255y(k-2) \\ &\quad +0.2598u(k-2) \\ Model_3: y(k) &= 0.0180 + 0.3500y(k-3) \\ &\quad +0.3200u(k-2) \end{aligned} \tag{35}$$

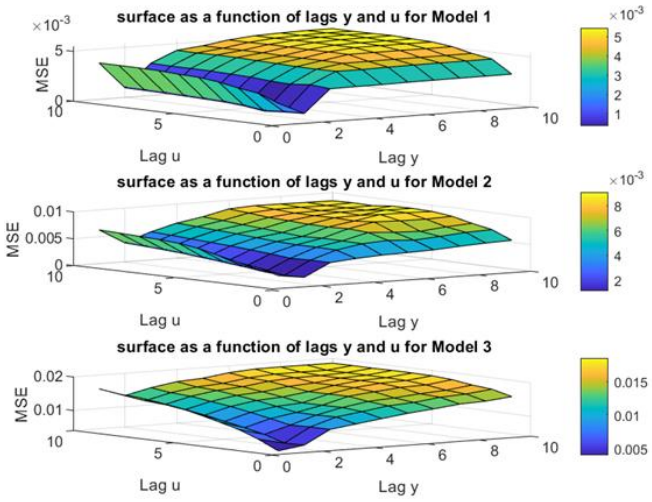


Fig. 5. Analysis of Lag Variations per Sub-Model Relative to Error.

In order to enforce constraints on the control input and its incremental variation, both control strategies make use of upper and lower bounds. These limits are defined as follows:

$$\begin{aligned} \lim_{\max} &= \min(u_{\max} - u(k-1), \Delta u_{\max}) \\ \lim_{\min} &= \max(u_{\min} - u(k-1), \Delta u_{\min}) \end{aligned} \tag{36}$$

According to the Genetic Algorithm (GA) employed, it was found that only two models significantly contribute to the modeling of the boost converter, each associated with different validity levels Figure 6. A weighting phase is introduced to determine the appropriate validity values corresponding to Models 2 and 3. In Figure 7, the results of the modeling process are presented, incorporating only Models 2 and 3 along with their normalized validities. Additionally, we illustrate the evolution of the modeling error between the multimodel output and the proposed model, comparing to the traditional approaches: (i) using a simple validity applied to all three models, (ii) using reinforced validity to all three models. This approach demonstrates that the number of models required can be minimized, and that the selection of appropriate validity for each model plays a critical role in improving overall modeling accuracy.

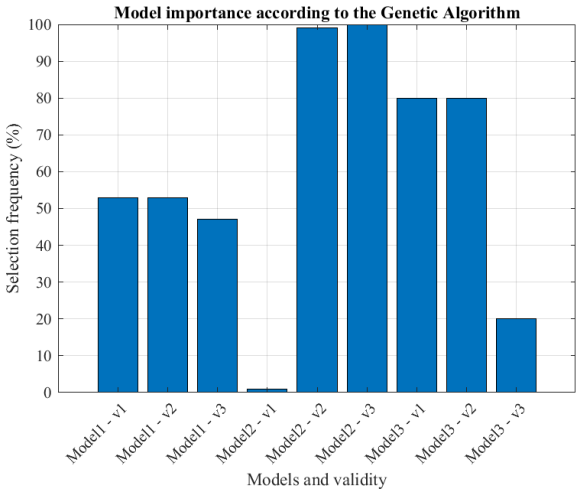


Fig. 6. Model importance selection.

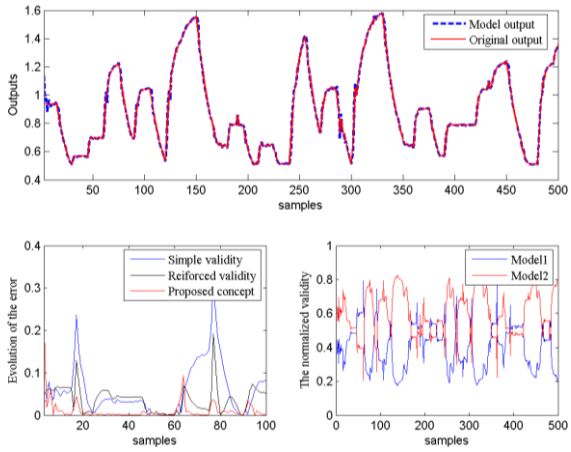


Fig. 7. Model Output and Validation: Comparative Analysis of Absolute Error Evolution Versus Conventional Methods.

In the case of Hammerstein model, it is important to recall that, within the context of the studied system, the control input $u(k)$, representing the cyclic ratio, is constrained to the interval. $0 \leq u(k) \leq 0.8$ . This condition is essential to ensure the feasibility of the solutions and the physical consistency of the system. A network of curves shown in Figure 8 reveals that the global solution, corresponding to the minimum of the cost function, lies within the restricted interval between -0.5 and 0.5. This observation allows for effectively narrowing the search for the optimal solution to this interval, there by reducing computational complexity.

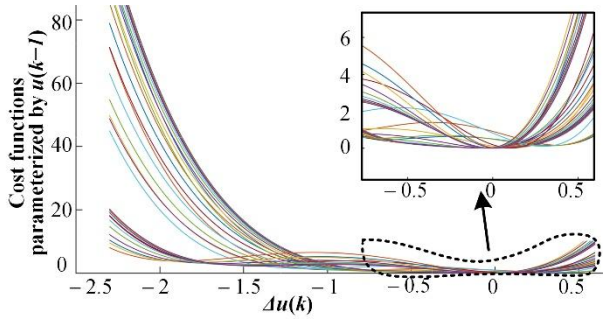


Fig. 8. J as a function of Δu(k) for different values of u(k−1).

The performance of the Newton-Raphson method was evaluated through experimental results presented in Figure 9. These results highlight the effectiveness of this method in solving real-time optimization problems, even in systems with fast dynamics. Due to its quadratic convergence and ability to incorporate input constraints, the method enables rapid convergence to an optimal solution while respecting the limits imposed by the system. While the multimodel approaches avoid the nonlinear optimization. A comparison of the two algorithms studied, namely the Hammerstein model and the proposed multimodel approach, reveals contrasting performance in terms of trajectory tracking, robustness, control and increment command The experimental results are shown in Figure 9. This response is taken for $H_c = 1$, $H_p = 7$, $0 \leq u(k) = 0.8$ and $-02 \leq \Delta u(k) \leq 0.2$. The robustness to reject perturbations is done in the two instants $k = 150$ and $k = 350$ with the corresponding value $\pm 12V$.

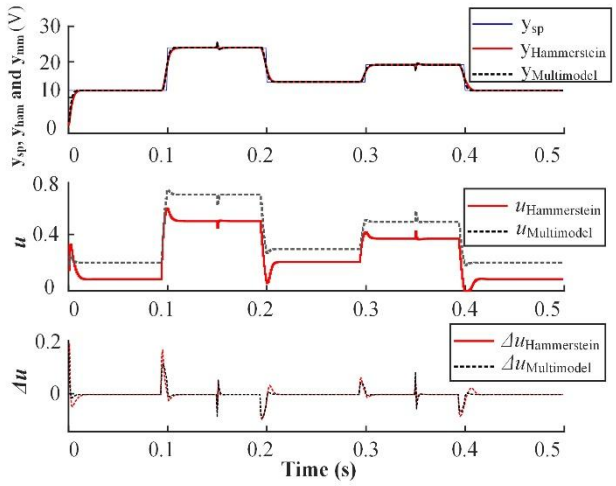


Fig. 9. Comparison study performances.

In term of execution time, the algorithms show varied performance, which is detailed in Table 1. Multimodel approach method is distinguished by its speed, making this approach particularly suitable for application requiring reduced computation times. The analytic solution, while slower, offer simplified implementation without explicit consideration of constraints. Owing to its quadratic convergence properties and inherent ability to incorporate input constraints the Newton Raphson optimization procedure allows for rapid achievement of an optimal solution while respecting the limits imposed by the system.

**Table 1.** Execution time comparison.

| Analytical solution | Newton Raphson method | Multimodel approach |
|---|---|---|
| 376µs | 262µs | 5.136µs |

## 6. CONCLUSIONS

The findings of this study validate the practical applicability and performance benefits of employing both multiple model and nonlinear model for real-time predictive control of a Boost converter. The multi-model approach, enhanced through K-means clustering, PSO, and genetic algorithms, offers adaptability and robustness by leveraging model diversity. Meanwhile, the nonlinear strategy based on the Hammerstein model provides accurate tracking through efficient numerical computation. Experimental implementation on an STM32 microcontroller confirms that both methods are capable of meeting the performance requirements of power electronics applications, with each exhibiting distinct advantages in terms of modeling flexibility and control precision.

## REFERENCES

Adhemar de, B.F., Maitelli, A.L. and de Oliveira Cavalcanti, A.L. (2007). Generalized predictive control based in multivariable bilinear multi-model. *IFAC Proceedings Volumes*, 40(5), pp.89–94.

Ben Mohamed, R., Ben Nasr, H. and M'Sahli, F. (2011). A multimodel approach for a nonlinear system based on neural network validity. *International Journal of Intelligent Computing and Cybernetics*, 4(3), pp.331–352.

Bennasr, H. (2009). *H. Bennasr, Commande prédictive des systèmes nonlinéaires à dynamique rapide* [unpublished]. Sfax: ENIS, Université SFAX.

Bennasr, H. and M'Sahli, F. (2024). AN APPROACH FOR MULTIMODEL REPRESENTATION OF PROCESS: A CASE STUDY OF BOOST CONVERTER. *Journal of Systems Engineering and Electronics*, 34(7), pp.437–445.

Bennasr, H. and M'Sahli, F. (2008). Une nouvelle stratégie de commande prédictive des systèmes non linéaires à dynamique rapides. In: *Conférence Internationale Francophone d'Automatique*. Conférence Internationale Francophone d'Automatique. Bucarest, Roumanie, pp.3–5.

Bennasr, H. and M'Sahli, F. Une supervision d'action pour la commande prédictive multiagent. In: *International Workshop on Systems Engineering Design & Applications*. International Workshop on Systems Engineering Design & Applications. SENDA 2008.

Bholowalia, P. and Kumar, A. (2014). EBK-Means: A Clustering Technique based on Elbow Method and K-Means in WSN. *International Journal of Computer Applications*, 105(9), pp.17–24.

Chen, Q., Dong, Z., Qin, J. and Qu, X. (2023). Model Predictive Control Based on Newton-Raphson Iterative Method of Triple-Active-Bridge Converter. In: *2023 IEEE 2nd International Power Electronics and Application Symposium (PEAS)*. 2023 IEEE 2nd

International Power Electronics and Application Symposium (PEAS). Guangzhou, China, pp.988–992.

Dang, N.D. and Aschemann, H. (2021). Application Study of Newton-Raphson-Type Nonlinear MPC for Hydrostatic Transmissions Under Disturbance and System Uncertainty. *SYSTEM THEORY, CONTROL AND COMPUTING JOURNAL*, 1(1), pp.21–29.

Delmotte, F., Dubois, L. and Borne, P. (1996). A general scheme for multi-model controller using trust. *Mathematics and Computers in Simulation*, 41(1), pp.173–186.

El Kamel, A., Ksouri-Lahmari, M. and Borne, P. (2000). Contribution to multimodel analysis and control. *Studies in Informatics and Control*, 9(1), pp.29–38.

Elfelly, N., Dieulot, J.-Y., Benrejeb, M. and Borne, P. (2010). A new approach for multimodel identification of complex systems based on both neural and fuzzy clustering algorithms. *Engineering Applications of Artificial Intelligence*, 23(7), pp.1064–1071.

Elfelly, N., Dieulot, J.-Y. and Borne, P. (2008). A neural approach of multimodel representation of complex processes. *International Journal of Computers Communications & Control*, 3(2), pp.149–160.

Forsyth, A.J. and Mollov, S.V. (1998). Modelling and control of DC-DC converters. *Power Engineering Journal*, 12(5), pp.229–236.

Fruzzetti, K.P., Palazoğlu, A. and McDonald, K.A. (1997). Nolinear model predictive control using Hammerstein models. *Journal of Process Control*, 7(1), pp.31–41.

Guo, Q.-G., Wang, D.-F., Han, P. and Lin, B.-H. (2003). Multi-model GPC for steam temperature system of circulating fluidized bed boiler. In: *Proceedings of the 2003 International Conference on Machine Learning and Cybernetics*. the 2003 International Conference on Machine Learning and Cybernetics (IEEE Cat. No.03EX693),. Xi'an, China, pp.906–911.

He, M., Cai, W.-J. and Li, S.-Y. (2005). Multiple fuzzy model-based temperature predictive control for HVAC systems. *Information Sciences*, 169(1), pp.155–174.

Hichem, B. and Faouzi, M. (2021). An Optimization Procedure of Model's Base Construction in Multimodel Representation of Complex Nonlinear Systems. In: *Engineering Problems-Uncertainties, Constraints and Optimization Techniques*. IntechOpen.

Hichem, B. and Faouzi, M. (2018). Multimodel Representation of Complex Nonlinear Systems: A Multifaceted Approach for Real-Time Application. *Mathematical Problems in Engineering*, 2018(1), p.1829396.

Hichem, B.N. and Faouzi, M. (2008). A multi-agent predictive control approach based on fuzzy supervisory loop for fast dynamic systems. In: *2008 5th International Multi-Conference on Systems, Signals and Devices*. 2008 5th International Multi-Conference on Systems, Signals and Devices. Amman, Jordan, pp.1–6.

Hmidene, A., Bennasr, H. and M'Sahli, F. (2024). On the use of an offline PID and multimodel predictive control approach for dc-dc boost converter. *Journal of Systems Engineering and Electronics*, 34(7), pp.251–257.

Humaira, H. and Rasyidah, R. (2020). Determining The Appropiate Cluster Number Using Elbow Method for K-Means Algorithm. In: Proceedings of the 2nd Workshop on Multidisciplinary and Applications (WMA) 2018, 24-25 January 2018, Padang, Indonesia. EAI.

Huo, H.-B., Zhu, X.-J., Hu, W.-Q., Tu, H.-Y., Li, J. and Yang, J. (2008). Nonlinear model predictive control of SOFC based on a Hammerstein model. *Journal of Power Sources*, 185(1), pp.338–344.

Johansen, T.A. (1994). *Operating Regime Based Process Modeling and Identification* [unpublished]. Norway: University of Trondheim.

Kodinariya, T.M. and Makwana, P.R. (2013). Review on determining number of Cluster in K-Means Clustering. *International Journal*, 1(6), pp.90–95.

Kouro, S., Perez, M.A., Rodriguez, J., Llor, A.M. and Young, H.A. (2015). Model predictive control: MPC's role in the evolution of power electronics. *IEEE industrial electronics magazine*, 9(4), pp.8–21.

Li, N., Li, S.-Y. and Xi, Y.-G. (2004). Multi-model predictive control based on the Takagi–Sugeno fuzzy models: a case study. *Information Sciences*, 165(3), pp.247–263.

Ltaief, M., Abderrahim, K., Abdennour, R.B. and Ksouri, M. (2008). Contributions to the multimodel approach: Systematic determination of a models' base and validities estimation. *International Journal of Automation and Systems Engineering*, 2(3), pp.213–220.

Messaoud, A.B., Aoun, S.T.B. and Ksouri, M.L. (2017). A New Strategy of Validities' Computation for Multimodel Approach: Experimental Validation. *International Journal of Advanced Computer Science and Applications (ijacsa)*, 8(7), pp.233–241.

Messaoud, A.B., Talmoudi, S. and Ksouri-Lahmari, M. (2018). Multimodel approach for modelling of nonlinear systems: validities' optimal computation. *COMPEL - The international journal for computation and mathematics in electrical and electronic engineering*, 37(1), pp.153–175.

Narendra, K. and Gallman, P. (1966). An iterative method for the identification of nonlinear systems using a Hammerstein model. *IEEE Transactions on Automatic Control*, 11(3), pp.546–550.

Nasr, H.B. and M'Sahli, F. (2008). A computational time requirement comparison between two approaches in MBPC. *International Journal of soft computing*, 3(2), pp.147–154.

Nasr Hichem, B. and Faouzi, M. (2008). A multi-agent approach to TS-fuzzy modeling and predictive control of constrained nonlinear system. *International Journal of Intelligent Computing and Cybernetics*, 1(3), pp.398–424.

Özkan, L., Kothare, M.V. and Georgakis, C. (2003). Control of a solution copolymerization reactor using multi-model predictive control. *Chemical Engineering Science*, 58(7), pp.1207–1221.

Porfírio, C.R., Almeida Neto, E. and Odloak, D. (2003). Multi-model predictive control of an industrial C3/C4

splitter. *Control Engineering Practice*, 11(7), pp.765–779.

Qin, S.J. and Badgwell, T.A. (2003). A survey of industrial model predictive control technology. *Control engineering practice*, 11(7), pp.733–764.

Roubos, J.A., Babuska, R., Bruijn, P.M. and Verbruggen, H.B. (1998). Predictive control by local linearization of a Takagi-Sugeno fuzzy model. In: *1998 IEEE International Conference on Fuzzy Systems Proceedings. IEEE World Congress on Computational Intelligence (Cat. No.98CH36228)*. 1998 IEEE International Conference on Fuzzy Systems. IEEE World Congress on Computational Intelligence (Cat. No.98CH36228). Anchorage, AK, USA, pp.37–42.

Swief, A., El-Zawawi, A. and El-Habrouk, M. (2019). A Survey of Model Predictive Control Development in Automotive Industries. In: *2019 International Conference on Applied Automation and Industrial Diagnostics (ICAAID)*. 2019 International Conference on Applied Automation and Industrial Diagnostics (ICAAID). pp.1–7.

Vidyasagar, P. and Swarup, K.S. (2023). *Introduction to Model Predictive Control in Design and Development of Model Predictive Primary Control of Micro Grids: Simulation Examples in MATLAB*. Springer Singapore.

Xu, Y., Jia, L., Peng, D. and Yang, W. (2023). Iterative Neuro-Fuzzy Hammerstein Model Based Model Predictive Control for Wind Turbines. *IEEE Transactions on Industry Applications*, 59(5), pp.6501–6512.

Yuan, C. and Yang, H. (2019). Research on K-Value Selection Method of K-Means Clustering Algorithm. *J*, 2(2), pp.226–235.