Periodica Polytechnica Electrical Engineering and Computer Science, 68(2), pp. 115–123, 2024

Implementation of New Predictive Control Based Volterra Model for Fast Dynamic Systems Using Microcontrollers: Real Time Application

Ali Hmidene[1*], Hichem Bennasr[2], Faouzi M'Sahli[3]

[1] Department of Electrical engineering, Higher Institute of Technological Studies Erriadh City, P. O. B. 35, 4023 Sousse, Tunisia
[2] Department of Electrical engineering, Higher Institute of Technological Studies EL Bustan City, P. O. B 88A, 3099 Sfax, Tunisia
[3] Department of Electrical Engineering, National Engineering School of Tunis, P. O. B. 176, 1002 Tunis, Tunisia
[*] Corresponding author, e-mail: ali.hmidene@gmail.com

Received: 23 July 2023, Accepted: 17 October 2023, Published online: 04 March 2024

Abstract
This paper presents a new contribution in the implementation of Volterra model predictive control for fast dynamics systems. The control approaches considered results on a switch paradigm that combine an online part based on suboptimal solution and an offline part referred to an offline neural network controller. The proposed approach has an advantage in comparison with nonlinear optimization-based control schemes. A real time application on STM32 to control a boost converter is studied and the results show very remarkable performances in time computing.

Keywords
predictive control, volterra model, fast systems, neural network, STM32

1 Introduction

Model Predictive control is an advanced control technique of the model-referenced optimal control type ("Model Based Predictive Control (MBPC)"). The general idea is to determine at each iteration an optimal command sequence taking into account the future behavior of the process predicted by a model. The application of this control is essential when we deal of a chemical and petroleum process that is characterized by "slow" processes. Indeed, predictive control, which is based on the resolution of an online optimization problem, is generally costly in terms of computation time. The first formulations of predictive control are obtained for linear problems. In this case, the analytical expression of the optimal solutions can be established and avoiding online optimization.

Among these formulations, we can mention the generalized predictive control (GPC) [1]. Nevertheless, this linear approach is not completely satisfactory and has certain limitations (highly nonlinear process and subject to significant disturbances, control of systems that regularly change operating point) which motivated the development of nonlinear predictive control ("Nonlinear Model Predictive Control" (NMPC)) [2].

The nonlinear approach of predictive control no longer allows an analytical solution of the optimization problem, which must therefore be solved online. However, the increase in computing time, the new formulations of the problem [3–5] and the improvement of the optimization procedure allowed this command to know a strong development and more particularly for systems with fast dynamics [6–9].

During the two last decades, considerable research has been carried out for the modeling, identification and control of nonlinear systems. Most real dynamical systems can best be represented by nonlinear models, which describe their behavior over a wide operating range. However, a linear model can only approximate the system around a given operating point. With the introduction of a nonlinear model, in the NMPC algorithm, the complexity of the control problem increases significantly [10–12]. The simplest way to reduce the online calculation is to transform the nonlinear problem (NMPC) into a linear problem (LMPC). The extraction of a linear model for a given operating point or the use of successive linearization over the prediction horizon is one of the approaches used to deal with nonlinear systems. The advantage of this technique lies in the fact

Cite this article as: Hmidene, A., Bennasr, H., M'Sahli, F. "Implementation of New Predictive Control Based Volterra Model for Fast Dynamic Systems Using Microcontrollers: Real Time Application", Periodica Polytechnica Electrical Engineering and Computer Science, 68(2), pp. 115–123, 2024. https://doi.org/10.3311/PPee.23033

that a solution from a quadratic formulation can be used to provide the optimal control [13]. Another alternative which it also makes possible to reduce of the computational load is to transform the nonlinear system into a linear system by using a technique of linearization by state feedback. Such approaches are presented in [14, 15]. The major challenge of this technique lies in the formulation of the constraints. The use of empirical models for nonlinear predictive control has been studied by several researchers. Neural networks and fuzzy logic form the framework most "popular" for empirical model development, although techniques based on Hammerstein models, Wiener models, and Volterra models have been introduced. One of the most frequently studied classes of nonlinear models are the separate nonlinearity models (Block Oriented Models) of Hammerstein type or Wiener type. Indeed, this type of presentation can cover the description of a large class of complex processes.

Several methods have been presented in the literature for the identification of these patterns. One can cite the most recent approaches which deal with the transformation into an orthogonal basis and the exploitation of the SVD decomposition to separate the linear parameters from the nonlinear ones [16]. From a command point of view these basic structures are exploited in [17–21]. Solving the control problem is based on nonlinear optimization techniques that are computationally expensive. Although these control strategies are applied on chemical processes, where the optimization time is not constrained by the time constants of the system, their application for systems with fast dynamics remains a difficult problem to solve. The main drawback of these approaches is that the estimation of the static nonlinearity can rarely be obtained with precision. Therefore, modeling and control involving fuzzy logic and artificial neural networks can give acceptable approximations and fast resolver technique [22–30].

From what preceded, we can consider that the nonlinear predictive control remains a strategy based mainly on the optimization of a nonlinear and no convex criterion. However, two major difficulties can be listed:

1. is related to the computation time which remains among the main locks to be solved;
2. the major difficulty concerns the feasibility of the solution.

The feasibility concerns the existence or not of solutions respecting the constraints associated with the optimization problem translating the control problem and finally the inadequacy, sometimes, of real-time control of processes with rapid dynamics. Faced with these inherent difficulties of the optimization problem, the application of this control strategy still remains a problem to be solved for systems with limited computing time, namely systems with fast dynamics [28, 31–34].

The article is organized in the following way. Section 2 describes the new MPC strategy adopted. The software and hardware implementation of the MPC algorithm using the STM32 microcontroller is detailed in Section 3 as well as the experimental results are presented based on a Boost converter showing very satisfactory results. Finally, Section 4 concludes this paper.

2 New concepts for nonlinear predictive control based on the parametric Volterra model

This approach focuses on nonlinear discrete single-variable systems defined in the form of a nonlinear input-output type equation given by:

$$y(k) = f\left(y(k-i), u(k-j)\right), \forall i, j > 0, \tag{1}$$

where $y(k) \in$ is the output of the system, $u(k) \in$ is the command variable and f is an application of $\Re \times \Re$ in $\Re$.

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 [35]. The structure of the parametric Volterra model can be designed of a second-order discrete models given by:

$$y(k) = y_0 + \sum\nolimits_{i=1}^{n_y} a_i y(k-i) + \sum\nolimits_{i=1}^{n_u} b_i u(k-i) + \sum\nolimits_{i=1}^{n_y} \sum\nolimits_{j=1}^{i} b_{ij} u(k-i) u(k-j) + \varepsilon(k), \tag{2}$$

where $y(k)$ and $u(k)$ are respectively the output and input $u(k)$ with the parameters Volterra model a_i, b_i, and 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 and y_0 is a bias term. The use of a the parametric Volterra model has one advantage 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 Eq. (2) can be written as:

$$y(k) = \theta^T \varphi(k) + \varepsilon(k), \tag{3}$$

with:

$$\theta^T = \left[y_0, a_1, a_2, \ldots a_{n_y}, b_1, b_2, \ldots b_{n_u}, b_{1,1}, \ldots b_{n_u, n_u} \right], \tag{4}$$

$$\varphi^T(k) = \begin{bmatrix} 1, y(k-1), \ldots y(k-n_y), \\ u(k-1), \ldots u^2(k-1), \ldots u^2(k-n_u) \end{bmatrix}, \tag{5}$$

where $\varphi(k)$ and θ are the regressor and the parameter vectors, respectively. The model Eq. (3) is linear in parameters, and its regressors and parameters may be identified from input output information. The objective of the control law is to compute at each sampling time k, the control input that minimizes a cost function given by:

$$\min_{\Delta u(k|k),\cdots\Delta u(k+H_u-1|k)} J\left(H_w, H_p, H_u, R\right),$$

$$J = \sum_{j=H_w}^{H_p} y(k+j) - y_{\mathrm{Ref}}(k+j)_Q^2 + \sum_{j=0}^{H_u-1} \Delta u(k+j|k)_R^2 . \quad (6)$$

Subject to the following constraints:

$$\begin{cases} u_{\min} \le u \le u_{\max} \\ \Delta u_{\min} \le \Delta u \le \Delta u_{\max} \end{cases}, \quad (7)$$

where H_u, H_p are the minimum and the maximum predictions, H_u is the control horizon $u_{\min}$, $u_{\max}$, $\Delta u_{\min}$, $\Delta u_{\max}$ are respectively the lower limit, upper limit, lower rate of change limit and higher rate of change limit of the input control. $\boldsymbol{Q}$ is the diagonal matrix of dimension ($\boldsymbol{n_y} \times \boldsymbol{n_y}$), $\boldsymbol{R}$ is ($\boldsymbol{n_u} \times \boldsymbol{n_u}$) diagonal.

$y \in \Re^{n_y}$, $\Delta u \in \Re^{n_u}$. 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.

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. From a predictive control point of view, several works have been developed, concerning the exploitation of the parametric Volterra type nonlinear model. The formulation of the control problem based on a parametric Volterra type model has been proposed in [35, 36]. The introduction of constraints on the input, on its increment and on the output, in the predictive control strategy based on the parametric Volterra model has been developed in [37, 38]. The presentation of a general nonlinear predictive control strategy, based on effective optimization techniques, has been developed in [39].

One possible way to address computational complexity is studied in this paper by using at first the suboptimal solution combined in the second way of an offline neural network controller. The bloc diagram of the considered MPC strategies is presented in Fig. 1.

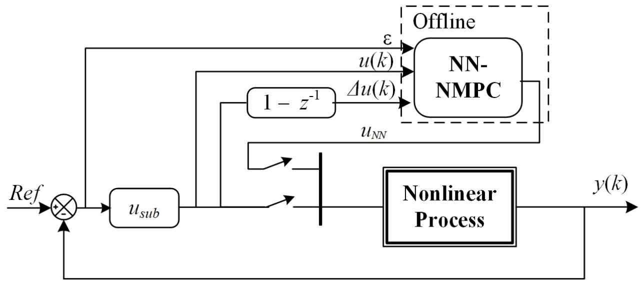


Fig. 1 Bloc diagram of MPC strategies

2.1 The suboptimal solution

Considering the polynomial form of the Volterra model, the control law subject to constraints are based on this form. So, the model given by Eq. (2) can be put in the polynomial form as:

$$A\left(q^{-1}\right) y(k) = y_0 + B_1\left(q^{-1}\right) u(k) + B_2\left(q_1^{-1}, q_2^{-1}\right) u^2(k) + \varepsilon(k), \quad (8)$$

with:

$$\begin{cases} A\left(q^{-1}\right) = 1 - a_1 q^{-1} - \ldots - a_{n_A} q^{-n_A}, \\ B\left(q^{-1}\right) = b_{10} + b_{11} q^{-1} + \ldots + b_{n_{B1}} q^{-n_{B1}} \\ B_2\left(q_1^{-1}, q_2^{-1}\right) u^2(k) \\ = \sum_{i=0}^{n_{B1}} \sum_{j=i}^{n_{B2}} b_{2ij} u(k-i) u(k-j) \end{cases} . \quad (9)$$

The output given by Eq. (8), can be written as:

$$\hat{y}(k+j|k) = y_0^* + G\left(q^{-1}\right) y(k) + \beta_1\left(q^{-1}\right) u(k+j) + \beta_2\left(q_1^{-1}, q_2^{-1}\right) u^2(k+j), \quad (10)$$

with:

$$\begin{cases} y_0^* = E(1) y_0 \\ \beta_1\left(q^{-1}\right) = E\left(q^{-1}\right) B_1\left(q^{-1}\right) \\ \beta_2\left(q_1^{-1}, q_2^{-1}\right) = E\left(q^{-1}\right) B_2\left(q^{-1}, q_2^{-1}\right) \end{cases}, \quad (11)$$

where $F(q^{-1})$ and $G(q^{-1})$ are two polynomials unique solutions of the following Diophantine equation:

$$1 = E_j\left(q^{-1}\right) A\left(q^{-1}\right) + q^{-j} F_j\left(q^{-1}\right), \quad (12)$$

with:

$$\begin{cases} E_j\left(q^{-1}\right) = 1 + e_1^{(j)} q^{-1} + \ldots + e_{n_{E_j}}^{(j)} q^{-n_{E_j}} \\ F_j\left(q^{-1}\right) = 1 + f_1^{(j)} q^{-1} + \ldots + f_{j-1}^{(j)} q^{-j+1} \\ \deg\left[E_j\left(q^{-1}\right)\right] = n_E = j - 1 \\ \deg\left[F_j\left(q^{-1}\right)\right] = n_F = n_A - 1 \end{cases} . \quad (13)$$

Knowing that the present and future control signals can be written according to the present and future control increments and the old control $u(k-1)$ according to the following relation:

$$u(k+i) = u(k-1) + \sum_{j=0}^{i} \Delta u(k+j);\ i = 0,1,\ldots,H_c. \quad (14)$$

In addition we define:

$$\begin{cases} \Delta u^*(k) = u(k) - u(k-1)\, si k \geq 0 \\ \Delta u^*(k) = u(k)\, si k \leq 0 \end{cases}. \quad (15)$$

Considering equations, Eqs. (13) to Eq. (15), the optimal predictor of the system output can be expressed only as a function of the present and future increments of the command as follows [40]:

$$\begin{aligned} y(k+j) &= v_0^{(j)} + v_1^{(j)}(q^{-1})\Delta u(k+j) \\ &+ v_2^{(j)}(q_1^{-1}, q_2^{-1})\Delta u^2(k+j), \end{aligned} \quad (16)$$

with:

$$\begin{aligned} v_0^{(j)} &= y_0^* + F_j y(k) \\ &+ \sum_{i=j+1}^{n_{B1}+j-1} \left[\delta_{1i} + \sum_{m=i}^{n_{B1}+j-1} \delta_{2im} \Delta u^* \begin{pmatrix} k+j \\ -m \end{pmatrix} \right] \Delta u^* \begin{pmatrix} k+j \\ -i \end{pmatrix}, \end{aligned} \quad (17)$$

$$v_{1i}^{(j)} = \delta_{1i} + \sum_{m=j+1}^{n_{B1}+j-1} \delta_{2im} \Delta u^*(k+j-m), \quad (18)$$

$$v_{2im}^{(j)} = \delta_{2im},\ i = 1,2,\ldots,j \text{ and } m = 1,2,\ldots,j, \quad (19)$$

$\deg\left[v_1^j\right] = H_p;\ \deg\left[v_2^j(q_1^{-1}, q_2^{-1})\right] = (H_p, H_p)$, δ_1 is a polynomial of order $(n_{B_1} + j)$ and δ_2 is a polynomial matrix of dimension $(n_{B_1} + j,\ n_{B_1} + j)$ whose coefficients are given by:

$$\begin{cases} \delta_{1i} = \sum_{v=0}^{i} \beta_{1v};\ i = 0,1,\ldots,n_{B_1} + j - 1 \\ \delta_{2ii} = \sum_{v=0}^{i} \sum_{\mu=v}^{i} \beta_{2v\mu};\ i = 0,1,\ldots,n_{B_1} + j - 1, \\ \delta_{2im} = \sum_{v=0}^{i} \left[\sum_{\mu=v}^{i} \beta_{2v\mu} + \sum_{\mu=v}^{m} \beta_{2v\mu} \right], \\ i = 0,1,\ldots,n_{B_1} + j - 1,\ m = i+1,\ldots,n_{B_2} + j - 1. \end{cases} \quad (20)$$

The cost function Eq. (6) can be put in the following form: by taking into account the equation of the predictor in incremental form:

$$J = \left(v_0^* + v_1\tilde{u} + v_2\tilde{u}^2\right)^T \left(v_0^* + v_1\tilde{u} + v_2\tilde{u}^2\right) + \lambda_u \tilde{u}^T \tilde{u}, \quad (21)$$

with:

$v_0^* = v_0 - y_{ref}$, $\Delta u = \tilde{u}$ and $\Delta = 1 - q^{-1}$.

The optimization cost function becomes:

$$h_0 + h_1\Delta u(k) + h_2\Delta u^2(k) + h_3\Delta u^3(k) = 0. \quad (22)$$

Under the assumption that all the increments of future orders are equal:

$$\Delta u(k) = \Delta u(k+1) = \ldots = \Delta u(k + H_c - 1), \quad (23)$$

with:

$$\begin{cases} h_0 = \left(v_0^j - y_{ref}(k+j)\right)\left[\sum_{i=1}^{j} v_{1i}^j\right], \\ h_1 = 2\left(v_0^j - y_{ref}(k+j)\right)\left[\sum_{i=1}^{j}\sum_{m=i}^{j} v_{2im}^j\right] \\ \quad + \left[\sum_{i=1}^{j} v_{1i}^j\right]^2 + \sum_{i=1}^{j} \lambda_i, \\ h_2 = 3\left[\sum_{i=1}^{j} v_{1i}^j\right]\left[\sum_{i=1}^{j}\sum_{m=i}^{j} v_{2im}^j\right], \\ h_3 = 2\left[\sum_{i=1}^{j}\sum_{m=i}^{j} v_{2im}^j\right]^2. \end{cases} \quad (24)$$

However, the sub-optimal u_{sub} solution is chosen according to the following procedure [35, 36]:

1. If all the roots are real, we will choose the root for which the cost function is minimal.
2. The real root is chosen in the case where the other two roots are complex conjugate.

In the presence of constraints, it is obvious that it is no longer possible to find an analytical solution to the control problem. The predictive control law should be computed iteratively using a one-dimensional search algorithm [39].

In order to satisfy the constraints on the command as well as on its increment, model. However, the new predictive control algorithm approach can be improved in order to reduce the computation time, in particular by proceeding to first limit the optimal solution by the values $u_{\min}$ and $u_{\max}$ permitted. Secondly, the developed control u_{sub} is delimited by two bounds which take into account their maximum and minimum deviations. These bounds are given by:

$$\begin{cases} \lim_{\max} = \min\left(u_{\max} - u(k-1), \Delta u_{\max}\right) \\ \lim_{\min} = \max\left(u_{\min} - u(k-1), \Delta u_{\min}\right) \end{cases}. \quad (25)$$

In the case where the control process violates the different constraints, a test is carried out based on the tracking error $\varepsilon(k)$, the control action $u(k)$ and its variations $\Delta u(k)$. So, in this case a trained neural network was done offline is used which provide the process the adequate control action. We thus replace the non-convex nonlinear optimization by a test carried out on the violation of the constraints as well as the tracking error. Our contribution lies here which makes it possible to reduce the time calculation of the nonlinear optimization.

2.2 The offline neural network controller

For the off-line part a neural predictive control is done with a nonlinear optimization algorithm, and a nonlinear neural model is employed for prediction. At every moment k, the future values of the control signals $u_{NN}(k)$, are determined as a solution to a nonlinear optimization problem [40–44]. The development of the control law is based on the minimization of a cost function given by:

$$J\left(w,u_{NN}\right)=\sum_{j=H_w}^{H_p}\begin{bmatrix} y_{set\ \text{point}}\left(k+j\right) \\ -y_{RN}\left(k+j\right) \end{bmatrix}^2 + \sum_{j=1}^{H_u}\Delta u\left(k+j-1\right)^2 . \tag{26}$$

The Newton-Raphson optimization method is used and J is iteratively minimized to determine the best control vector [26].

$$u_{NN}^{n}\left(k\right)=\begin{bmatrix} u_{NN}^{n}\left(k\right),u_{NN}^{n}\left(k+1\right),\ldots, \\ u_{NN}^{n}\left(k+N_u+1\right) \end{bmatrix}. \tag{27}$$

The adaptation of the control vector is calculated by Eq. (28):

$$u_{NN}^{n+1}\left(k\right)=u_{NN}^{n}\left(k\right)-\left(\left.\frac{\partial^2 J\left(k\right)}{\partial u_{NN}\left(k\right)^2}\right|_{u_{NN}\left(k\right)=u_{NN}^{(n)}\left(k\right)}\right) \times \left.\frac{\partial J\left(k\right)}{\partial u_{NN}\left(k\right)}\right|_{u_{NN}\left(k\right)=u_{NN}^{(n)}\left(k\right)}, \tag{28}$$

with $\partial^2 J\left(k\right)/\partial u_{NN}\left(k\right)^2$ and $\partial J\left(k\right)/\partial u_{NN}\left(k\right)$ are the Jacobian and Hessian matrices and n is the number of iterations.

To avoid calculating the inverse of the Hessian matrix, Eq. (10) is rewritten as a system of linear equations $Ax = B$.

$$A=\frac{\partial^2 J\left(k\right)}{\partial u_{NN}\left(k\right)^2};\ B=\frac{\partial J\left(k\right)}{\partial u_{NN}\left(k\right)}, \tag{29}$$

$$x=u_{NN}\left(k+1\right)-u_{NN}\left(k\right). \tag{30}$$

3 Experimental results

The process used is a Boost converter that is designed for a new implementation of model predictive control. The bloc diagram is shown in Fig. 2.

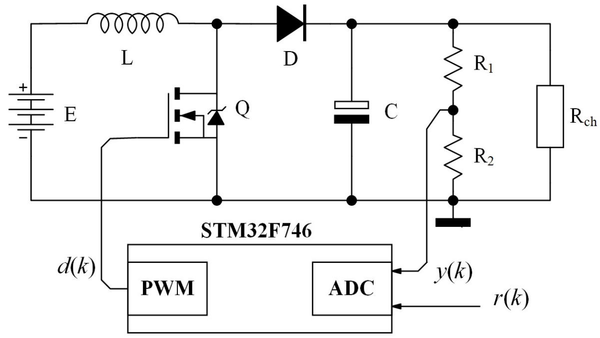


Fig. 2 Boost Converter bloc diagram

This signal controls the switching of the power transistor via the MOSFET driver. The output of the boost is connected to the input of the analog to digital converter (ADC) of the microcontroller. The system to be identified having the following characteristics:

- Supply voltage: $V_{DC} = 12$ V,
- Output voltage: $V_{out} = 24$ V,
- Hash frequency: 80 KHz,
- Load resistance: $Rch = 24\ \Omega$,
- Duty cycle: $D_0 = 0.5$.

All the implemented algorithms have been executed on the STM32F746 DISC0 microcontroller platform, known for its cost-effectiveness. This platform features the STM32F746, an Arm Cortex-M7 32-bit RISC Core based digital signal controller (DSC). Developed by STMicroelectronics, the STM32F746 combines advanced hardware features with a rich set of peripherals and a robust ecosystem for efficient development. Operating at speeds of up to 216MHz, this processor supports single-cycle DSP and SIMD instructions, making it well-suited for applications that require rapid and efficient data processing.

One of the standout features of the STM32F746 is its integrated hardware floating-point unit (FPU), which empowers the microcontroller to execute complex mathematical operations involving real numbers with high precision and speed. This capability makes the STM32F746 an ideal choice for applications involving signal processing, control systems, and scientific computations.

The microcontroller boasts an array of embedded memories that play a crucial role in its efficiency. With up to 1 Mbyte of Flash memory, it can store program code and data, enabling flexible and reliable application deployment. The STM32F746 also incorporates 320 Kbytes of SRAM, including a dedicated 64 Kbytes of Data TCM RAM for critical real-time data, as well as 16 Kbytes of instruction TCM RAM for time-sensitive routines.

Furthermore, the STM32F746 is equipped with a comprehensive set of peripherals. It includes multiple 12-bit Analog-to-Digital Converters (ADCs) for accurate analog signal acquisition, along with Digital-to-Analog Converters (DACs) for generating analog outputs. The microcontroller supports a generous assortment of general-purpose timers, some of which even offer Pulse-Width Modulation (PWM) capabilities. Its communication interfaces encompass USART, SPI, I2C, I2S, CAN, USB, and Ethernet, facilitating seamless connectivity with a variety of external devices and networks.

For real-time control, the Keil MDK-ARM toolchain is employed. This software development platform, operating in a window-based environment, combines a powerful editor with a project manager and make facility tool. It serves as an integrated solution for developing embedded applications, providing a C/C++ compiler, macro assembler, linker/locator, and an AXF file generator. The program itself is coded in the C language.

In configuring the analog-to-digital converter ADC1, the scan mode with DMA is utilized. Additionally, a timer generates a PWM signal operating at an 80 kHz frequency. This signal serves to control a power transistor with an impressive 0.1% resolution on the duty cycle.

When it comes to programming, the floating-point hardware unit (FPU) is harnessed for operations involving real variables. Meanwhile, the DSP_Lib library comes into play for matrix operations. The entire program, consisting of two parts, an online segment is embedded within the microcontroller. The offline portion is calculated through Matlab ® [45].

The Experimental boost converter lab with different components is presented in Fig. 3.

To estimate the Volterra parameters a collection of experimental data is generated based on an amplitude-modulated pseudo-random binary signal (APRBS) used as excitation signal. A Recursive Least Squares' method (RLS) is retained for the parametric identification of the Volterra model. The results of empirical data and the modeling of the Volterra model are illustrated in Fig. 4.

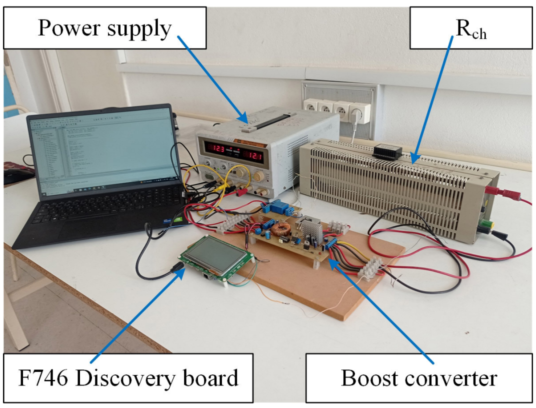
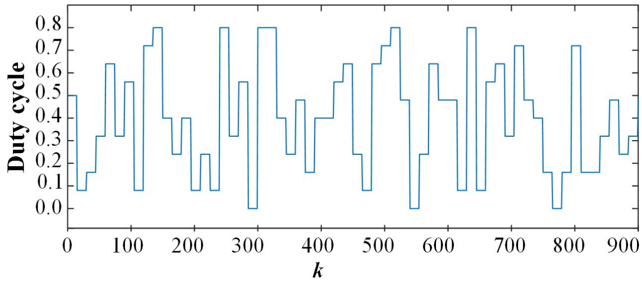


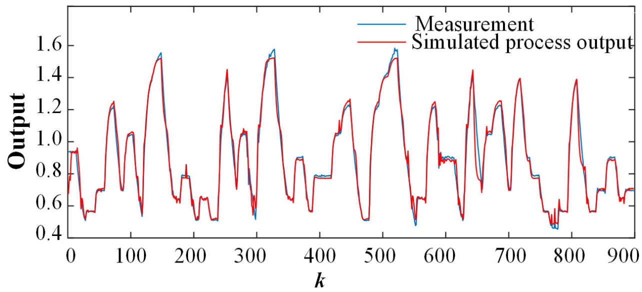
Fig. 3 Experimental control Boost converter lab

Having used the structure command given in Fig. 1, the obtained results are shown in Fig. 5. A feedforward neural network is used with five neurons Andan hyperbolic tangent function in the hidden layer are used. It respectively shows the trajectory to follow and the output of the system as well as the evolution of the control law. We Shows that the output of the system quickly converges to the reference.

We can, moreover, notice the respect of the constraints. Comparisons are made between the proposed approach and the nonlinear control approach (NMPC). We notice that:



(a)



(b)

Fig. 4 Identification boost converter, (a) The APRBS signal; (b) The measurement (red line) and simulated process output (blue line)

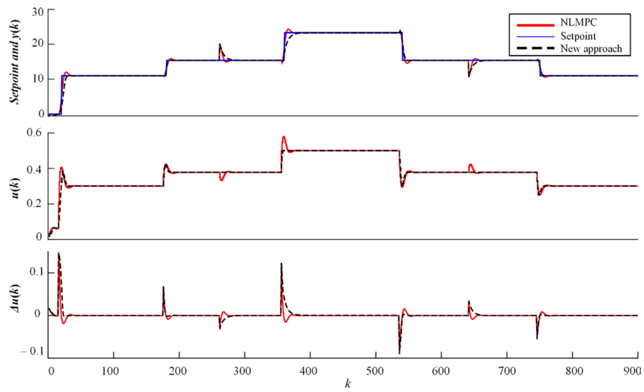


Fig. 5 Evolution of the output/setpoint and command signals: Proposed approach and NMPC

- The resolution of the optimization problem in the proposed approach is convex.
- The best dynamics are seen when evaluating the outputs regarding the nonlinear control approach, while the proposed approach shows similar dynamics to that obtained by a classical multi-model control.
- To verify the ability of this proposed strategy to rejecting disturbances, an output disturbance is introduced at the instants $k = 250$ and $k = 700$. The disturbance is eliminated showing a satisfactory ability to reject disturbances. So, it is shown that the control concept used is capable of delivering significantly improved control performance.

In Fig. 6(a) we have plotted the required computation time at each step to get the sequence command for both the new approach and the NMPC case. No iteration exceeds 1 ms for the proposed approach. It should be remembered that the sampling period is 1ms. The optimization task is clearly the most computationally demanding, which shows the infeasibility of real-time application for time-limited systems. In Fig. 6(b) we show the sampling time where we use the solution provided by the neural network predictive control implemented in a lookup table. During 900 iterations we used only 18.7% of the cases where we used the solution given the neural network, which shows that the computational load is reduced by 80%.

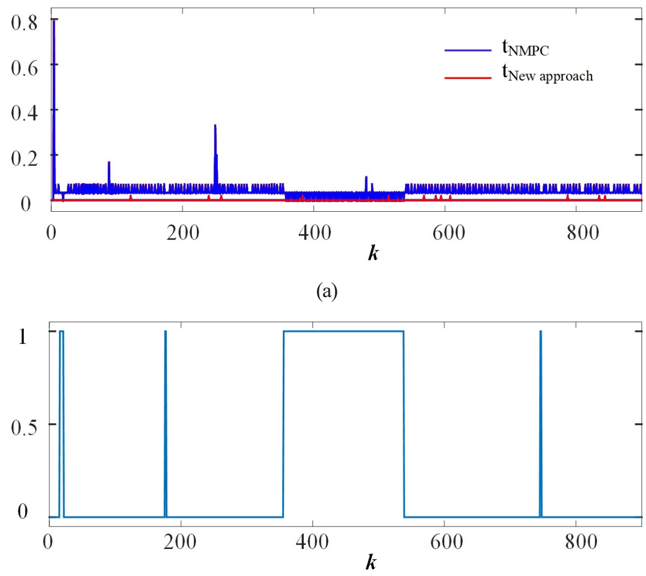


Fig. 6 Time computation; (a) Evaluation of the required implementation time; (b) Instants of the use of the neural network controller

4 Conclusions

The novel approach presents very remarkable performances in time computing and avoids nonlinear optimization procedure. The proposed concept compares favorably with respect to a numerical optimization routine when applied to Boost converter system. Moreover, the novel algorithm reduces the online computational burden and hence has the potential to be applied to the system with faster time constants. Computation in the new design procedure is simpler and faster than the nonlinear optimization and it offer very good control performance. Due to this approach the on-line optimization procedure at each simple time is avoided, so for that it can useful with very small sampling time.

References

[1] Clarke, D. W., Mohtadi, C., Tuffs, P. S. "Generalized predictive control –Part I. The basic algorithm", Automatica, 23(2), pp. 137–148, 1987.
https://doi.org/10.1016/0005-1098(87)90087-2

[2] Qin, S. J., Badgwell, T. A. "A survey of industrial model predictive control technology", Control Engineering Practice, 11(7), pp. 733–764, 2003.
https://doi.org/10.1016/S0967-0661(02)00186-7

[3] Lourenço, J. M., Lemos, J. M. "Predictive multiple model adaptive control of plants with fast changing dynamics", In: Proceedings of the 1st IFAC Workshop on Nonlinear Model Predictive Control for Fast Systems, Grenoble, France, 2006, pp. 165–170.

[4] Merabet,A., Ouhrouche, M., R.T. Bui, R. T.,Ezzaidi, H. "Nonlinear PID predictive control of induction motor drives", In: Proceedings of the 1st IFAC Workshop on Nonlinear Model Predictive Control for Fast Systems, Grenoble, France, 2006, pp. 53–58.

[5] Gruber, J. K., Guzmán, J. L., Rodríguez, F., Bordons, C., Berenguel, M., Sánchez, J. A. "Nonlinear MPC based on a Volterra series model for greenhouse temperature control using natural ventilation", Control Engineering Practice, 19(4), pp. 354–366, 2011.
https://doi.org/10.1016/j.conengprac.2010.12.004

[6] Heath, W. P. "Multipliers for quadtatic programming with box constraints", In: Proceedings of the 1st IFAC Workshop on Nonlinear Model Predictive Control for Fast Systems, Grenoble, France, 2006, pp. 23–28.

[7] Jaochim, H., Bock, H. G.,Diehl, M. "An online active set strategy for fast parametric quadratic programming in mpc applications", In: Proceeding of the 1st IFAC Workshop on Nonlinear Model Predictive Control for Fast Systems, Grenoble, France, 2006, pp. 13–22.

[8] Yang, J. "Numerical Methods for Model Predictive Control", Master's thesis, Technical University of Denmark, 2008.

[9] Zavala, V. M., Laird, C. D., Biegler, L. T. "Fast solvers and rigorous models can both be accommodated in NMPC", In: Proceedings of the 1st IFAC Workshop on Nonlinear Model Predictive Control for Fast Systems, Grenoble, France, 2006, pp. 97–109.

[10] Henson, M. A. "Nonlinear model predictive control: current status and future directions", Computers &Chemical Engineering, 23(2), pp. 187–202, 1998.
https://doi.org/10.1016/S0098-1354(98)00260-9

[11] Zeynal, H., Zakaria, Z., Kor, Yaghoobi, A. "An Efficient Linearized Volterra Series for Nonlinear Model Predictive Control in Dynamic Systems", In: 2020 IEEE International Conference on Power and Energy (PECon), Penang, Malaysia, 2020, pp. 415–419. ISBN 978-1-7281-7069-5
https://doi.org/10.1109/PECon48942.2020.9314390

[12] Liu, Z., Xie, L., Bemporad, A.,Lu, S. "Fast Linear Parameter Varying Model Predictive Control of Buck DC-DC Converters Based on FPGA", IEEE Access, 6, pp. 52434–52446, 2018.
https://doi.org/10.1109/ACCESS.2018.2869043

[13] Brooms, A. C., Kouvaritakis, B. "Successive constrained optimization and interpolation on non-linear model based predictive control", International Journal of Control, 73(4), pp. 312–316, 2000.
https://doi.org/10.1080/002071700219669

[14] Kurtz, M. J., Henson, M. A. "Input-output linearizing control of constrained nonlinear process", Journal of Process Control, 7(1), pp. 3–17, 1997.
https://doi.org/10.1016/S0959-1524(96)00006-6

[15] van Soest, W. R., Chu, Q. P., Mulder, J. A. "Combined Feedback Linearization and Constrained Model Predictive Control for Entry Flight", Journal of Guidance, Control, and Dynamics, 29(2), pp. 427–434, 2006.
https://doi.org/10.2514/1.14511

[16] Gomez, J. C., Baeyens, E. "Identification of Nonlinear Systems using Orthonormal Bases", In: Proceedings of the IASTED International Conference on Intelligent Systems and Control, ISC 2001, Tampa, Florida, USA, 2001, pp. 126–131. ISBN 0-88986-315-6.

[17] Abdennour, R. B., Ksouri, M., M'Sahli, F. "Nonlinear Model-Based Predictive Control Using a Generalized Hammerstein Model and its Application to a Semi-Batch Reactor", International Journal of Advanced Manufacturing Technology, 20(11), pp. 844–852, 2002.
https://doi.org/10.1007/s001700200225

[18] Jeong, B.-G., Yoo, K.-Y., Rhee, H.-K. "Nonlinear Model Predictive Control Using a Wiener Model of a Continuous Methyl Methacrylate Polymerization Reactor", Industrial & Engineering Chemisrty Research, 40(25), pp. 5968–5977, 2001.
https://doi.org/10.1021/ie990887b

[19] Gomez, J. C., Jutan, A., Baeyens, E. "Wiener model identification and predictive control of a pH neutralisation process", IEE Proceedings - Control Theory and Applications, 151(3), pp. 329–338, 2004.
https://doi.org/10.1049/ip-cta:20040438

[20] Fruzzetti, K. P., Palazoğlu, A., Mcdonald, K. A. "Nonlinear model predictive control using Hammerstein models", Journal of Process Control, 7(1), pp. 31–41, 1997.
https://doi.org/10.1016/S0959-1524(97)80001-B

[21] Norguay, J.S., Palazoğlu, A., Romagnoli, J. A. "Model predictive control based on Wiener models", Chemical Engineering Science, 53(1), pp. 75–84, 1998.
https://doi.org/10.1016/S0009-2509(97)00195-4

[22] Huang, Y. L., Lou, H. H., Gong, J. P., Edgar, T. F. "Fuzzy model predictive control", IEEE Transactions on Fuzzy Systems, 8(6), pp. 665–678, 2000.
https://doi.org/10.1109/91.890326

[23] Babuška, R. Verbruggen, H. "Neuro-fuzzy methods for nonlinear system identification", Annual Reviews in Control, 27(1), pp. 73–85, 2003.
https://doi.org/10.1016/S1367-5788(03)00009-9

[24] Lucia, S., Navarro, D., Lucía, Ó. Zometa, P., Findeisen, R. "Optimized FPGA Implementation of Model Predictive Control for Embedded Systems Using High-Level Synthesis Tool", IEEE Transactions on Industrial Informatics, 14(1), pp. 137–145, 2018.
https://doi.org/10.1109/TII.2017.2719940

[25] Ren, Y. M., Alhajeri, M. S., Luo, J., Chen, S., Abdullah, F., Wu, Z., Christofides, P. D. "A tutorial review of neural network modeling approaches for model predictive control", Computers & Chemical Engineering, 165, 107956, 2022.
https://doi.org/10.1016/j.compchemeng.2022.107956

[26] Stogiannos, M., Alexandridis, A., Sarimveis, H. "Model predictive control for systems with fast dynamics using inverse neural models", ISA Transactions, 72, pp. 161–177, 2018.
https://doi.org/10.1016/j.isatra.2017.09.016

[27] Benrabah, M., Kara, K., AitSahed, O., Hadjili, M. L. "Constrained Nonlinear Predictive Control Using Neural Networks and Teaching–Learning-Based Optimization", Journal of Control, Automation and Electrical Systems, 32(5), pp. 1228–1243, 2021.
https://doi.org/10.1007/s40313-021-00755-4

[28] Hou, G., Gong, L., Huang, C., Zhang, J. "Fuzzy modeling and fast model predictive control of gas turbine system", Energy, 200, 117465, 2020.
https://doi.org/10.1016/j.energy.2020.117465

[29] Chaber, P. "Fast Nonlinear Model Predictive Control Algorithm with Neural Approximation for Embedded Systems: Preliminary Results", In: Proceedings of KKA 2020—The 20th Polish Control Conference, Łódź, Poland, 2020, pp. 1067–1078. ISBN 978-3-030-50936-1
https://doi.org/10.1007/978-3-030-50936-1_89

[30] Chaber, P., Ławryńczuk, M. "Fast analytical model predictive controllers and their implementation for STM32 ARM microcontroller", IEEE Transaction on Industrial Informatics, 15(8), pp. 4580–4590, 2019.
https://doi.org/10.1109/TII.2019.2893122

[31] Zarzycki, K., Ławryńczuk, M. "Fast Nonlinear Model Predictive Control Using LSTM Networks: A Model Linearisation Approach", In: 2022 30th Mediterranean Conference on Control and Automation (MED), Vouliagmeni, Greece, 2022, pp. 1–6. ISBN 978-1-6654-0673-4
https://doi.org/10.1109/MED54222.2022.9837211

[32] Doncevic, D. T., Schweidtmann, A. M., Vaupel, Y., Schäfer, P., Caspari, A., Mitsos, A. "Deterministic Global Nonlinear Model Predictive Control with Neural Networks Embedded", IFAC-PapersOnLine, 53(2), pp. 5273–5278, 2020. https://doi.org/10.1016/j.ifacol.2020.12.1207

[33] Wojtulewicz, A., Ławryńczuk, M. "Implementation of Multiple-Input Multiple-Output Dynamic Matrix Control Algorithm for Fast Processes Using Field Programmable Gate Array", IFAC-PapersOnLine, 51(6), pp. 324–329, 2018. https://doi.org/10.1016/j.ifacol.2018.07.174

[34] Wojtulewicz, A., Ławryńczuk, M. "Computationally Efficient Implementation of Dynamic Matrix Control Algorithm for Very Fast Processes Using Programmable Logic Controller", In: 2018 23rd International Conference on Methods & Models in Automation & Robotics (MMAR), Miedzyzdroje, Poland, 2018, pp. 579–584. ISBN 978-1-5386-4325-9 https://doi.org/10.1109/MMAR.2018.8486132

[35] Haber, R., Bars, R., Lengyel, O. "Three extended horizon adaptive nonlinear predictive control schemes on the parametric volterra model", In: 1999 European Control Conference (ECC), Karlshruhe, Germany, 1999, pp. 1790–1795. ISBN 978-3-9524173-5-5 https://doi.org/10.23919/ECC.1999.7099575

[36] Haber, R., Bars, R., Lengyel, O. "Sub-Optimal Nonlinear Predictive and Adaptive Control based on the Parametric Volterra Model", International Journal of Applied Mathematics and Computer Science, 9(1), pp. 161–173, 1999.

[37] M'Sahli, M., Abdennour, R. B., Ksouri, M. "Identification and predictive control of non-linear process using a parametric Volterra Model", International Journal of Computational Engineering Science, 2(4), pp. 633–651, 2001. https://doi.org/10.1142/S1465876301000490

[38] Ali, H., Hichem, B., Faouzi, M. "New strategy for fast model based predictive control using Volterra model: Real time application", In: 2022 IEEE 21st International Conference on Sciences and Techniques of Automatic Control and Computer Engineering (STA), Sousse, Tunisia, 2022, pp. 12–17. ISBN 978-1-6654-8261-5 https://doi.org/10.1109/STA56120.2022.10018987

[39] Nejib, B. N. "Contributions to the analysis and synthesis of predictive control of nonlinear systems, PhD Thesis, National Engineering School of Sfax (ENIS), 2006.

[40] Ozgur, Y. "A comparative study on optimization methods for the constrained nonlinear programming problems", Mathematical Problems in Engineering, 2005, 181483, 2005. https://doi.org/10.1155/MPE.2005.165

[41] Ławrynczuk, M. "A Family of Model Predictive Control Algorithms With Artificial Neural Networks", International Journal of Applied Mathematics and Computer Science, 17(2), pp. 217–232, 2007. https://doi.org/10.2478/v10006-007-0020-5

[42] Tatjewski, P., Ławrynczuk, M. "Soft computing in model-based predictive control", International Journal of Applied Mathematics and Computer Science, 16(1), pp. 101–120, 2006.

[43] Yu, D. L., Gomm, J. B. "Implementation of neural network predictive control to a multivariable chemical reactor", Control Engineering Practice, 11(11), pp. 1315–1323, 2003. https://doi.org/10.1016/S0967-0661(02)00258-7

[44] Zeng, G. M., Qin, X. S., Le, H., Huang, G. H., Liu, H. L., Lin, Y. P. "A neural network predictive control system for paper mill wastewater treatment", Engineering Applications of Artificial Intelligence, 16(2), pp. 121–129, 2003. https://doi.org/10.1016/S0952-1976(03)00058-7

[45] MathWorks "Matlab 7.0.1.24704 (R14)", [computer program] Available at: https://mathworks.com [Accessed: 22 July 2023]