

Real-Time Implicit Model Predictive Control for 3-phase VSI

V. Sabatini, A. Lidozzi, L. Solero
Dept. of Engineering
Roma Tre University, C-PED
Rome, IT
valerio.sabatini@uniroma3.it

A. Formentini, P. Zanchetta
Dept. of Electrical and Electronic Engineering
The University of Nottingham
Nottingham, UK
andrea.formentini@nottingham.ac.uk

S. Bifaretti
Dept. of Industrial Engineering
University of Rome Tor Vergata, C-PED
Rome, IT
bifaretti@ing.uniroma2.it

Abstract – This paper deals with the real time implementation of an Implicit Model Predictive Control strategy applied to a 3-phase Voltage Source Inverter (VSI) in stand-alone applications. Solving the optimization problem online with a sufficiently short sampling time allows to directly control the output voltage while respecting the desired constraints. Due to the resulting complex calculations and the related time-constraints, the implementation has been carried out on an industrial-grade FPGA to take advantage of the true parallel execution capabilities.

Keywords—Model Predictive Control, implicit, FPGA, VSI, LC filter, Active Set, Quadratic Programming.

I. INTRODUCTION

Model Predictive Control (MPC) is a powerful and elegant way to regulate dynamic systems subject to physical constraints. It uses a system model to predict the evolution of system states in the future. The optimal control action is then selected minimizing a cost function. For linear systems with quadratic cost function and linear constraints, the control approach results in solving a Quadratic Programming (QP) at each sample time [1]. With the increase of system states and prediction horizon, however, the problem becomes highly computational demanding to be solved in real time. For this reason, initially, MPC has been mainly applied to chemical and industrial process with low sampling rates. During the last decade, an explicit MPC formulation has been proposed to overcome this limitation. This solution consists in reformulating the MPC problem as a multiparametric program that is solved offline for all possible states, resulting in a lookup table giving the optimal solution as a function of the current state [2]. With this approach, the online computation is drastically reduced at the expense of an increased memory requirement to store the lookup table. The method has been successfully applied to both power converters and drives applications [3]-[4]. In recent years, the increasing computational power of electronic devices, especially Field Programmable Gate Arrays (FPGA), has permitted to extend the use of MPC technology also to higher sampling frequency systems, such as power converters. The intrinsic finite number of possible control actions of power converters has recently suggested the MPC variant, called Finite Control Set MPC (FCS-MPC) [5]. In this strategy the system state is predicted for every possible converter configuration and the one resulting in the minimum cost

function is selected as optimal. The approach has been successfully applied to different power converter topologies, such as shunt active filters [6], matrix converters [7] and drive systems [8]. The method is promising, but the exponential growth of the possible control action imposes the use of a prediction horizon usually smaller than four. Moreover, applying a single voltage vector at each sampling time, as well as implying a variable switching frequency, increases the ripple on the output voltages and currents.

The last step forward to find the optimal control action in power electronics applications, exploiting the full capabilities of MPC, is the fulfillment of the implicit MPC, which entails the implementation at the high sample rates required by such systems. An evidence of its effectiveness has already been provided in [9], where simulation results show the output voltage control of a VSI with LC filter achieved using an efficient modified Active Set technique. However, this method is limited to work only with a single step of prediction and it has not been really implemented on a control board.

This paper presents an implicit MPC applied to a three-phase two-level inverter with LC filter for off-grid applications. The optimization problem is solved online in real time on a FPGA using the Active Set method, and the output of the predictive algorithm is used as the modulating signal for a pulse width modulator. It is well known that controlling the inverter output voltages increases the whole system complexity and order, and the MPC approach is considered particularly suitable to overcome this issue. Simulation and preliminary implementation results are reported to analyze the control scheme and to prove the effectiveness of the method.

II. SYSTEM MODEL

As previously stated, the MPC algorithm has been applied to a two-level grid connected voltage source inverter with an output LC filter, depicted in Figure 1. The converter is fed by a constant DC voltage source and the unknown load in the MPC algorithm has been modeled considering its drained current (i_a), which is kept constant along the prediction horizon.

The plant average model in the dq -reference frame, which is required by the MPC algorithm to predict the future state of the system in order to choose the optimal control action, is characterized by the equations shown in (1).

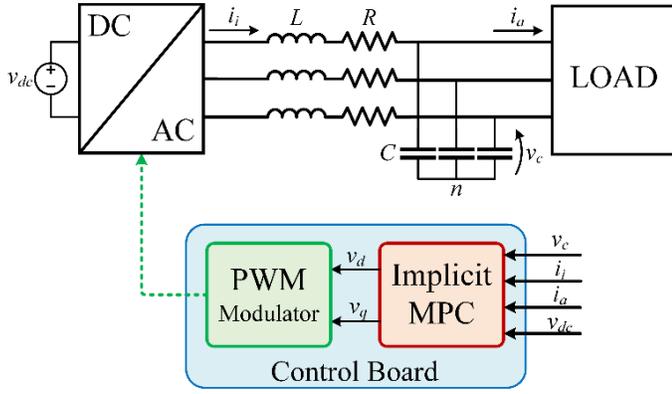


Figure 1. Scheme of the considered system configuration.

$$\begin{cases}
 \frac{di_{id}}{dt} = -\frac{R}{L} \cdot i_{id} + \omega_{el} \cdot i_{iq} - \frac{v_{cd}}{L} + \frac{v_{id}}{L} \\
 \frac{di_{iq}}{dt} = -\frac{R}{L} \cdot i_{iq} - \omega_{el} \cdot i_{id} - \frac{v_{cq}}{L} + \frac{v_{iq}}{L} \\
 \frac{dv_{cd}}{dt} = \frac{i_{id}}{C} + \omega_{el} \cdot v_{cq} - \frac{i_{ad}}{C} \\
 \frac{dv_{cq}}{dt} = \frac{i_{iq}}{C} - \omega_{el} \cdot v_{cd} - \frac{i_{aq}}{C} \\
 \frac{di_{ad}}{dt} = 0 \\
 \frac{di_{aq}}{dt} = 0
 \end{cases} \quad (1)$$

Where i_i is the inductor current, v_c is the output capacitor voltage, v_i is the inverter voltage and ω_{el} is the AC angular frequency. Furthermore, R , L , C are the output LC filter resistance, inductance and capacitance, respectively.

The corresponding state space representation is shown in (2), where the system state has been augmented to include the load current components. In this application, it is supposed that at each sampling instant it is possible to measure the whole state of the system (x).

$$A = \begin{bmatrix}
 -\frac{R}{L} & -\frac{1}{L} & \omega & 0 & 0 & 0 \\
 \frac{1}{C} & 0 & 0 & \omega & -\frac{1}{C} & 0 \\
 -\omega & 0 & -\frac{R}{L} & -\frac{1}{L} & 0 & 0 \\
 0 & -\omega & \frac{1}{C} & 0 & 0 & -\frac{1}{C} \\
 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0
 \end{bmatrix} \quad (2)$$

$$B = \begin{bmatrix}
 \frac{1}{L} & 0 \\
 L & 0 \\
 0 & 0 \\
 0 & \frac{1}{L} \\
 0 & 0 \\
 0 & 0 \\
 0 & 0 \\
 0 & 0
 \end{bmatrix} \quad C = \begin{bmatrix}
 1 & 0 & 0 & 0 & 0 & 0 \\
 0 & 1 & 0 & 0 & 0 & 0 \\
 0 & 0 & 1 & 0 & 0 & 0 \\
 0 & 0 & 0 & 1 & 0 & 0
 \end{bmatrix}$$

$$\begin{cases}
 \frac{dx}{dt} = Ax + Bu \\
 y = Cx
 \end{cases} \quad y = [i_{id} \quad v_{cd} \quad i_{iq} \quad v_{cq}]^T$$

$$x = [i_{id} \quad v_{cd} \quad i_{iq} \quad v_{cq} \quad i_{ad} \quad i_{aq}]^T \quad u = [v_{id} \quad v_{iq}]^T$$

The continuous-time system model has been discretized using the well-known Zero-Order Hold (ZOH) method, leading to a state space representation with A_d , B_d and C_d matrices, which are provided to the MPC algorithm.

III. PROPOSED CONTROL STRATEGY

The algorithm proposed in this paper aims to control the output voltage of a three-phase Voltage Source Inverter, managing the right amount of power which is required by the loads. Such a system usually works at constant voltage reference and it must be able to fast compensate all the load variations. The described control strategy is based on the implicit MPC technique.

The first task performed by the algorithm consists in the evaluation of the effects due to the control action which was just applied in order to compensate the computational delay. In fact, in the proposed strategy there is a delay equal to a sampling period (T_{sp}) between the sampling and the actuation of the control action. The change of the system state during this period must be taken into account and this can be simply done by using the discrete model to evaluate the state at the k -instant starting from the measured values.

The plant state is then predicted in the future for a certain number of steps, which corresponds to the prediction horizon (H_p). During this calculation, it is assumed that the control action can be changed for a number of steps equal to the control horizon (H_u), which is obviously lower than or equal to the prediction horizon. As a result, the possibility of selecting different prediction and control horizons provides two degrees of freedom to influence the complete control dynamic and performance. In fact, it can be noticed that choosing H_p greater than H_u penalizes the changes in the control action, providing a stabilizing effect to the system.

A quadratic cost function (J), shown in (3), is then minimized with the aim of finding the optimal control action. The first summation considers the difference between the output (y) and its reference (r) weighted with \tilde{Q} while the second summation penalizes input changes ($\Delta u(k) = u(k) - u(k-1)$) weighted with \tilde{R} .

$$J(k) = \sum_{i=1}^{H_p} \|y(k+i) - r(k+i)\|_{\tilde{Q}(i)}^2 + \sum_{i=0}^{H_u-1} \|\Delta u(k+i)\|_{\tilde{R}(i)}^2 = \|Y(k) - T(k)\|_{\tilde{Q}}^2 + \|\Delta U(k)\|_{\tilde{R}}^2 \quad (3)$$

$$Q(i) = \begin{bmatrix} W_{i_d}(i) & 0 & 0 & 0 \\ 0 & W_{v_{cd}}(i) & 0 & 0 \\ 0 & 0 & W_{i_{iq}}(i) & 0 \\ 0 & 0 & 0 & W_{v_{cq}}(i) \end{bmatrix} \quad R(i) = \begin{bmatrix} W_{\Delta v_{id}}(i) & 0 \\ 0 & W_{\Delta v_{iq}}(i) \end{bmatrix}$$

$$\tilde{Q} = \begin{bmatrix} Q(1) & 0 & \dots & 0 \\ 0 & Q(2) & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & Q(H_p) \end{bmatrix} \quad \tilde{R} = \begin{bmatrix} R(1) & 0 & \dots & 0 \\ 0 & R(2) & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & R(H_u) \end{bmatrix} \quad (5)$$

$$Y(k) = [y(k+1) \dots y(k+H_p)]^T \quad T(k) = [r(k+1) \dots r(k+H_p)]^T \quad \Delta U(k) = [\Delta u(k) \dots \Delta u(k+H_u-1)]^T$$

As can be seen in the expression of $Q(i)$, each quadratic error is weighted with a coefficient (W_{i_d} , $W_{v_{cd}}$, $W_{i_{iq}}$, $W_{v_{cq}}$), which can be selected to change the behavior of the algorithm by increasing or decreasing the priority of the control of a quantity over another. In this particular application, being the voltages across the capacitors the goal of the control algorithm, the weights associated to the dq -axes currents have been set to zero. Analogously, in order not to penalize the dynamic performance of the control, the weights associated with the change of the control actions ($W_{\Delta v_{id}}$, $W_{\Delta v_{iq}}$) have been set to zero.

constraints required to define the desired maximum and minimum values of $U(k)$ and $Y(k)$.

The constraints on $U(k)$ are written in the form of equation (6).

$$F = [F_1 \dots F_{H_u}]$$

$$\Phi \begin{bmatrix} U(k) \\ 1 \end{bmatrix} \leq \bar{0}_{4 \times 1} \quad \Phi = [F \quad \tilde{f}]$$

$$\tilde{f} = \begin{bmatrix} f_1 \\ \vdots \\ f_{H_u} \end{bmatrix} \quad F_k = \begin{bmatrix} \bar{0}_{(k-1) \times 2} \\ I_{2 \times 2} \\ -I_{2 \times 2} \\ \bar{0}_{(H_u-k) \times 2} \end{bmatrix} \quad (6)$$

$$f_k = [-v_{id}^{MAX} \quad -v_{iq}^{MAX} \quad v_{id}^{MIN} \quad v_{iq}^{MIN}]^T$$

Where I and 0 represent the identity matrix and the zero matrix, respectively. The corresponding number of rows and columns are specified by their subscripts. The vector f_k contains the maximum and minimum values of v_{id} and v_{iq} that can be chosen by the algorithm as future control actions.

The constraints on $Y(k)$ are expressed by (7).

$$\Psi = \begin{bmatrix} C_d A_d \\ \vdots \\ C_d A_d^{H_p} \end{bmatrix} \quad \Upsilon = \begin{bmatrix} C_d B_d \\ \vdots \\ \sum_{i=0}^{H_p-1} C_d A_d^i B_d \end{bmatrix}$$

$$\Theta = \begin{bmatrix} C_d B_d & \dots & 0 \\ \vdots & \ddots & \vdots \\ \sum_{i=0}^{H_p-1} C_d A_d^i B_d & \dots & \sum_{i=0}^{H_p-H_u} C_d A_d^i B_d \end{bmatrix} \quad (4)$$

$$H = 2\Theta' \tilde{Q} \Theta + \tilde{R} \quad G(k) = -2\Theta' \tilde{Q} E(k)$$

$$Z(k) = \Psi x(k) + \Upsilon u(k-1) \quad E(k) = T(k) - Z(k)$$

$$\min_{\Delta U} \left\{ \frac{1}{2} \Delta U' H \Delta U + G' \Delta U : \alpha \Delta U \leq \beta \right\}$$

$$G = [G_1 \dots G_{H_p}]$$

$$\Xi \begin{bmatrix} Y(k) \\ 1 \end{bmatrix} \leq \bar{0}_{8 \times 1} \quad \Xi = [G \quad \tilde{g}]$$

$$\tilde{g} = \begin{bmatrix} g_1 \\ \vdots \\ g_{H_p} \end{bmatrix} \quad G_k = \begin{bmatrix} \bar{0}_{(k-1) \times 8} \\ I_{4 \times 4} \\ -I_{4 \times 4} \\ \bar{0}_{(H_p-k) \times 8} \end{bmatrix} \quad (7)$$

$$g_k = \begin{bmatrix} -y^{MAX} \\ y^{MIN} \end{bmatrix} \quad y^{MAX} = \begin{bmatrix} i_{id}^{MAX} \\ v_{cd}^{MAX} \\ i_{iq}^{MAX} \\ v_{cq}^{MAX} \end{bmatrix} \quad y^{MIN} = \begin{bmatrix} i_{id}^{MIN} \\ v_{cd}^{MIN} \\ i_{iq}^{MIN} \\ v_{cq}^{MIN} \end{bmatrix}$$

The minimization of the cost function can be written as shown in equation (4), which corresponds to a quadratic programming (QP) problem. α and β are the matrices associated to the linear

The vector g_k contains the maximum and minimum values of

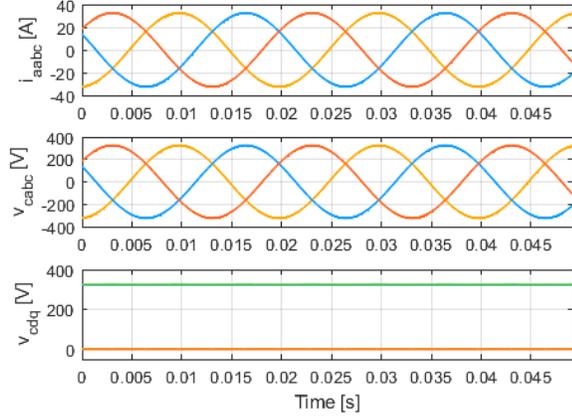


Figure 2. Load currents and output voltages in steady state condition with a resistive load wye connected.

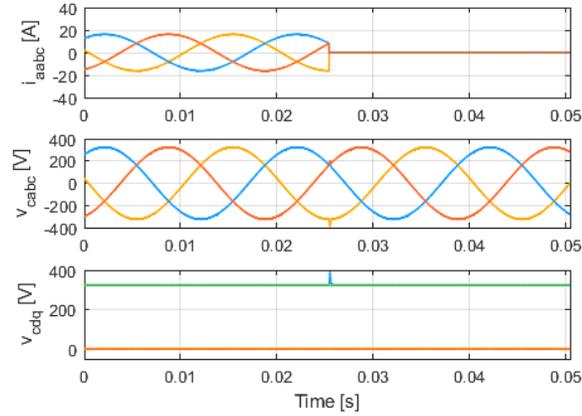


Figure 4. Load currents and output voltages during a transient from load to no load operation.

i_{id} , v_{cd} , i_{iq} and v_{cq} .

In order to obtain α and β , the equations related to the constraints on $U(k)$ and $Y(k)$ must be modified by expressing the relations as functions of $\Delta U(k)$ (8). Π_1 contains only the first two columns of Π .

$$\Pi = \begin{bmatrix} \sum_{i=1}^{H_u} F_i & \sum_{i=2}^{H_u} F_i & \dots & F_{H_u} \end{bmatrix}^{-T} \quad (8)$$

$$\alpha = \begin{bmatrix} \Pi \\ G\Theta \end{bmatrix} \quad \beta = \begin{bmatrix} -\Pi_1 U(k-1) - \tilde{f} \\ -GZ - \tilde{g} \end{bmatrix}$$

Given the small number of variables, the technique which seems to perform better for the solution of the inequality constrained QP problem of equation (4) on a FPGA device is the Active Set method [10]. The implementation has been carried out using the LU decomposition to efficiently solve the linear systems associated with the equality constrained QP problems which must be solved to obtain the final solution [11]. It is important to note that, at the end of each algorithm execution, only the first control action $\Delta u(k)$ is actually applied, while the remaining $(H_u - 1)$ are deleted and they will be entirely recalculated at the following iteration. The obtained $\Delta u(k)$ is then added to $u(k-1)$ and the resulting $u(k)$ is sent to a PWM

modulator to control the converter switches, ensuring a constant

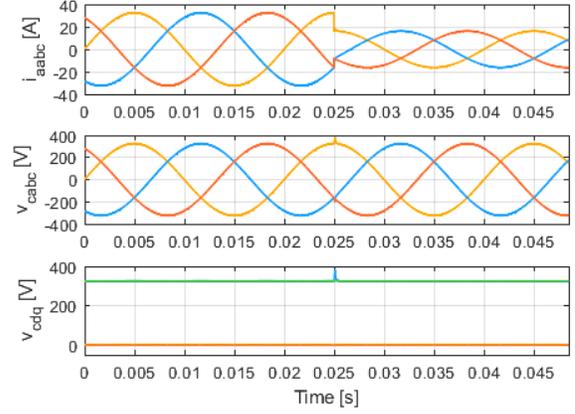


Figure 3. Load currents and output voltages during a load step change with a resistive load wye connected.

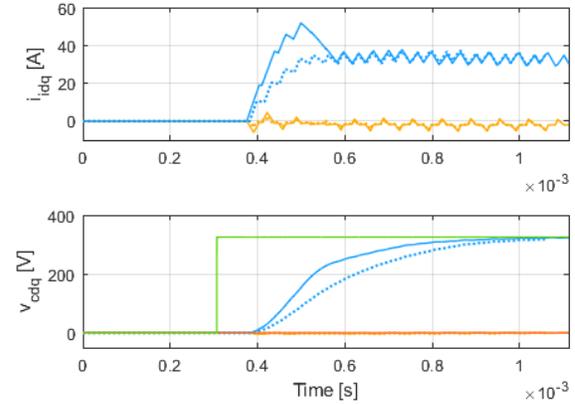


Figure 5. Inductor currents and output voltages during a voltage reference step with (dotted line) and without (solid line) current limitation. switching frequency.

IV. SIMULATION RESULTS

Preliminary results have been achieved by an accurate simulation model, where the effects of the real implementation, such as the computational delay, have been included. The parameters which have been used are shown in Table I. The effectiveness and the performances of the proposed strategy are shown in the following figures. Figure 2 illustrates the behavior of the control algorithm in steady state condition with a resistive load equal to 10Ω on each phase, wye connected. Figure 3 shows the effectiveness of the proposed strategy during a load step from 10Ω to 20Ω on each phase, wye connected. Figure 4 displays a significant load step variation from 10Ω on each phase to a no-load condition. Finally, Figure 5 shows the efficacy of the current limitation and its effects on the dynamic behavior of the control action during a voltage reference step.

Table I. System parameters

Parameter	Value
DC-bus voltage v_{dc}	750 V
LC Filter Capacitance C	20 μ F
LC Filter Inductance L	800 μ H
Resistance R	50 m Ω
Sampling time $T_{sp} = 1/f_{sw}$	66.67 μ s
Prediction horizon H_p	2
Control horizon H_u	1
$W_{vd} = W_{vq}$	100

V. EXPERIMENTAL SETUP AND PRELIMINARY IMPLEMENTATION

The control board used to achieve the experimental results, where the proposed control algorithm is implemented, is shown in

Figure 6 (PED-Board®). PED-Board, which is based on the National Instruments System-on-Module sbRIO-9651, has been designed with dedicated peripherals specifically for power electronics and drives applications. NI sbRIO-9651 has a dual-core ARM processor and an Artix7 FPGA. FPGA manages the on-board ADCs, resolver, PWM unit and scheduler, CAN-bus communication and DACs interface. The Real-Time target (i.e. the ARM micro-processor) manages the whole external communication structure forwarding and sending/receiving the corresponding commands and data to/from the FPGA.

The algorithm implementation on the industrial grade FPGA target takes the benefits from using the well-known LabVIEW graphical development environment. It is time effective and exhibits a high efficiency in generating the low-level code. FPGA space occupancy, reported in Table II, corresponds to a preliminary implementation of the control algorithm with 2 step-ahead prediction and 32-bit single precision floating-point arithmetic.

Figure 7 illustrates the MOSFET inverter based on SiC devices used to carry out the experimental verification of the proposed control algorithm.



Figure 6. PED-Board control unit.

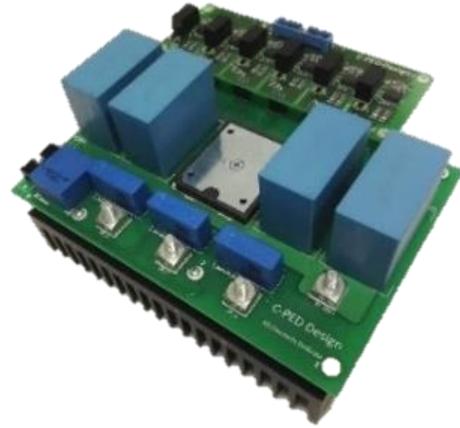


Figure 7. SiC MOSFET inverter used in the experimental campaign.

A scheme of the implemented algorithm is shown in Figure 8, where the tasks entirely executed by the FPGA are highlighted.

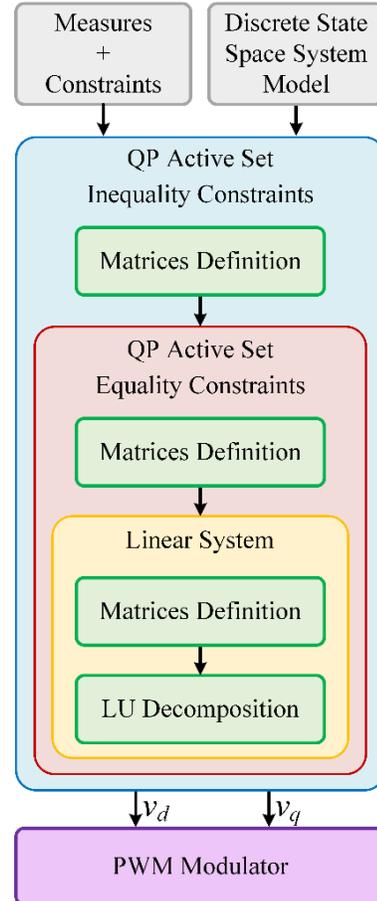


Figure 8. Scheme of the algorithm implemented on the FPGA target.

As previously described, after the measurement of the required quantities and the definition of the discrete system model, the optimization problem is formulated in the form of a QP problem with inequality constraints, which is solved using the Active Set method. This technique breaks up the initial problem with

inequality constraints into simpler problems characterized by equality constraints. Each equality constrained problem, after an appropriate matrices definition, implies the resolution of a linear system. This task is efficiently achieved by decomposing the matrices using the LU technique.

It is important to note that LabVIEW FPGA does not support the use of two-dimensional matrices, which have been stored using one-dimensional vectors. In order to reduce the computational burden needed to index the required matrices elements, the vectors have been stored concatenating the columns of the initial matrices. Furthermore, the Active Set method natively uses variable-sized matrices, and this is not possible on the FPGA (i.e. FPGA does not support variable length arrays). For this reason, the QP solver has been designed to support the maximum possible size of the matrices involved in the calculations, and the actual number of elements is a parameter which can be different at each iteration.

Table II - FPGA utilization

Xilinx Artix7 FPGA device	Percent
Slice registers	95.2
Slice LUTs	91.4
Block RAMs	39.3
DSP48s	80.8

CONCLUSIONS

In this paper, a powerful and effective strategy based on a real time implementation of the implicit MPC for the control of a 3-phase VSI with output LC filter for stand-alone applications has been presented. Simulations results show the effectiveness of the proposed algorithm to regulate the output capacitor voltages while respecting the imposed constraints on filter currents during both steady state condition and load changes. Preliminary implementation results on an industrial grade FPGA also demonstrate the feasibility of such demanding real time implementation. Future steps involve certainly the co-simulation of the realized FPGA code with a time-synchronized platform for the power electronics part such as NI-Multisim. After that, hardware-in-the-loop verification can be accomplished before starting the experimental campaign.

REFERENCES

[1] Maciejowski, J. M. (2002). Predictive control: with constraints. Pearson education.

[2] Bemporad, A., Borrelli, F., Morari, M. (2002). Model predictive control based on linear programming~ the explicit solution. IEEE Transactions on Automatic Control, 47(12), 1974-1985.

[3] Mariétoz, S., Morari, M. (2009). Explicit model-predictive control of a PWM inverter with an LCL filter. IEEE Transactions on Industrial Electronics, 56(2), 389-399.

[4] Bolognani, S., Bolognani, S., Peretti, L., Zigliotto, M. (2009). Design and implementation of model predictive control for electrical motor drives. IEEE Transactions on industrial electronics, 56(6), 1925-1936.

[5] Rodriguez, J., Kazmierkowski, M. P., Espinoza, J. R., Zanchetta, P., Abu-Rub, H., Young, H. A., Rojas, C. A. (2013). State of the art of finite control set model predictive control in power electronics. IEEE Transactions on Industrial Informatics, 9(2), 1003-1016.

[6] Rabbeni, R., Tarisciotti, L., Gaeta, A., Formentini, A., Zanchetta, P., Pucci, M., ... Rivera, M. (2015, September). Finite states modulated model predictive control for active power filtering systems. In Energy Conversion Congress and Exposition (ECCE), 2015 IEEE (pp. 1556-1562). IEEE.

[7] Vargas, R., Ammann, U., Rodriguez, J. (2009). Predictive approach to increase efficiency and reduce switching losses on matrix converters. IEEE Transactions on Power Electronics, 24(4), 894-902.

[8] Preindl, M., Bolognani, S. (2013). Model predictive direct torque control with finite control set for PMSM drive systems, Part 1: Maximum torque per ampere operation. IEEE Transactions on Industrial Informatics, 9(4), 1912-1921.

[9] M. Nauman and A. Hasan, "Efficient Implicit Model-Predictive Control of a Three-Phase Inverter with an Output LC Filter," in IEEE Transactions on Power Electronics, vol. 31, no. 9, pp. 6075-6078, Sept. 2016.

[10] M. S. K. Lau, S. P. Yue, K. V. Ling and J. M. Maciejowski, "A comparison of interior point and active set methods for FPGA implementation of model predictive control," 2009 European Control Conference (ECC), Budapest, 2009, pp. 156-161.

[11] Nocedal, J., Wright, S. (2006). Numerical optimization. Springer Science & Business Media.