**American Journal of Computer Science and Engineering**
2019; 6(2): 16-26
http://www.openscienceonline.com/journal/ajcse

# Predictive Control Based Embedded Processor for Power Converters

Netani Tugia, Joseph Miller, Mansour Assaf*

School of Engineering & Physics, The University of the South Pacific, Suva, Fiji Islands

## Email address

s11076271@student.usp.ac.fj (N. Tugia), s11087362@student.usp.ac.fj (J. Miller), mansour.assaf@usp.ac.fj (M. Assaf)

*Corresponding author

## To cite this article

Netani Tugia, Joseph Miller, Mansour Assaf. Predictive Control Based Embedded Processor for Power Converters. *American Journal of Computer Science and Engineering*. Vol. 6, No. 2, 2019, pp. 16-26.

**Received**: March 19, 2019; **Accepted**: May 4, 2019; **Published**: May 16, 2019

## Abstract

Recently, there has been an increase in the use of model predictive control for power converters. Model predictive control uses the discrete-time model of the system to predict future values of control variables for all possible control actions and computes a cost function related to control objectives. Model predictive control technique can provide fast, dynamic, and reliable response. However, this control method implementation imposes a very high computational burden and causes significant hardware requirements for real-time implementation. In this paper, we propose a Lyapunov-based controller for a two–level voltage source inverter model. The optimized design of the control algorithm is implemented as an embedded processor in Very High-Speed Integrated Circuit Hardware Description Language (VHDL). Afterwards, the controller design synthesized and downloaded onto a Field Programmable Gate Array (FPGA) target board for validation purposes. The design of the proposed predictive control based controller for power converters is implemented on an embedded processor FPGA board. The model predictive control Lyapunov based algorithm needs to converge within a few number of iterations. The effectiveness of the proposed method was studied and performance evaluate in software by running MATLAB/Simulink computer simulations. The proposed control model was implemented and validated as well on FPGA target board. In the proposed controller design technique, all control calculations and the commutation schemes are implemented in VHDL and therefore the need for another digital signal processor is eliminated. The proposed scheme takes full advantages of the parallel computation capability of the FPGA design.

## Keywords

Model Predictive Control, MATLAB, FPGA, PID, VHDL, Flying Capacitor Multi-Level Inverter

## 1. Introduction

Control techniques have evolved over the years in its application for power converters. Classical control techniques for controlling power converters include Proportional-Integral-Derivative Control (PID) controller, Field Oriented Control and Direct Torque Control whereas Fuzzy Logic and Neural Networks have provided a more intelligent control aspect for power converters. Furthermore, control techniques that are predictive in nature have advanced in the form of Hysteresis based control, Deadbeat controllers and Model Predictive Control. This project report revolves around Model Predictive Control and its variation.

These control techniques support in mitigating issues such as high sensitivity to fluctuating parameters and delays in signals. Conventional Model Predictive Control (MPC) methods are simple to implement and operate when handling such problems mentioned prior, however, the calculations executed in the conventional control algorithm generates an increase in time delay. In order to lower the time delay used for calculations, a modified MPC is studied which utilizes a Lyapunov function future reference voltage vector to compare directly in the cost function with the eight possible voltage vectors generated by the voltage source inverter.

The main objectives of this work are to study the proposed controller and its Lyapunov function based MPC for a two – level voltage source inverter model. Implement coding for the studied control algorithms and simulate appropriate signal that comply with result gathered from researched journals. Design and optimize the modified control algorithm as an embedded processor in Very High-Speed Integrated Circuit Hardware Description Language (VHDL) and finally download the synthesized language onto the Field Programmable Gate Array (FPGA) target board. In the following section, we discuss the system overview.

Voltage Source Inverters (VSI) is the connection between the Direct Current (DC) source side with Alternating Current (AC) load side which acts as an inverter when current flows from DC to AC side. Otherwise, the converter acts as a rectifier if current flows vice versa [1]. The inverter is also known as three-phase voltage source pulse width modulation (PWM).

The operation of the circuit is based on the switching state of semiconductor devices which can either be in an ON/OFF state. These semiconductor devices permit flow of current in only one direction and are characterized by the time it takes to transit from OFF to ON state (turn-on time), the time it takes to transit from ON to OFF state (turn-off time), the maximum current and allowable voltage in the ON and OFF state. Due to these characteristics, manufacturers have developed power devices such as Bipolar Junction Transistors (BJT), Metal-Oxide-Semiconductor Field-Effect Transistor (MOSFET) and Insulated-Gate Bipolar Transistor (IGBT) [2].

There are several types of VSIs that are utilized in today's industries. This is portrayed in Figure 1:

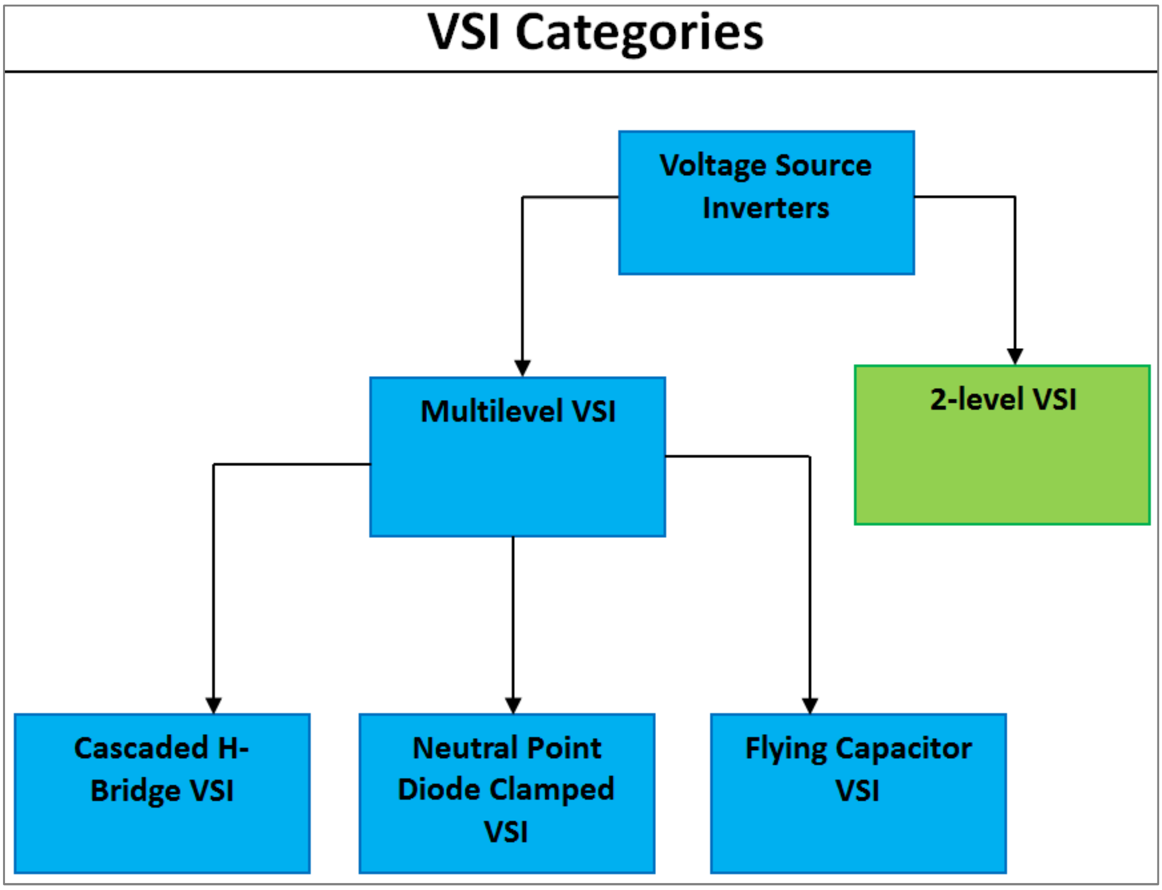


***Figure 1.*** *Class of Voltage Source Inverters.*

The Neutral Point Converter (NPC) design was conducted by Akagi, Takahashi and Nabae in 1981 and considered as a basic three-level diode clamped inverter [3]. Voltage clamping diode is essential for this type of multilevel inverter. The DC side of the inverter is divided by an even number of capacitors that are connected in series with the neutral point located in the midpoint of the line. The Neutral Point Converter-Multilevel Inverter (NPC-MLI) provides solutions to industries that included mining, water marine, oil and chemicals. Its application in conventional high-power AC motor drives such as conveyor, pumps and mills depict the importance of this device [4].

Flying Capacitor Multilevel Inverters (FC-MLI) like the NPC-MLI has an even number of capacitors on the DC source side. In FC-MLI, clamping capacitors are used instead of clamping diodes in NPC-MLI which gives rise to the number of switching patterns available since reverse voltage is not blocked by capacitors [5].

A single full-bridge or H-bridge inverter is considered when obtaining a three-level waveform. In this case, a DC source is provided to each of the three inverters. This type of multilevel inverter is suitable for battery-based applications, reactive power control and interface with renewable energy sources [6].

A basic two-Level Voltage Source Inverters (2L-VSI) scheme consisting of six semiconductor switches was proposed in [6]. The switching components compromise of IGBT's and MOSFET's which are suitable for this type of inverter. Its simple structure and capacity to keep the system stable are ideal in uninterruptable power supply (UPS) applications which are favored in the industry.

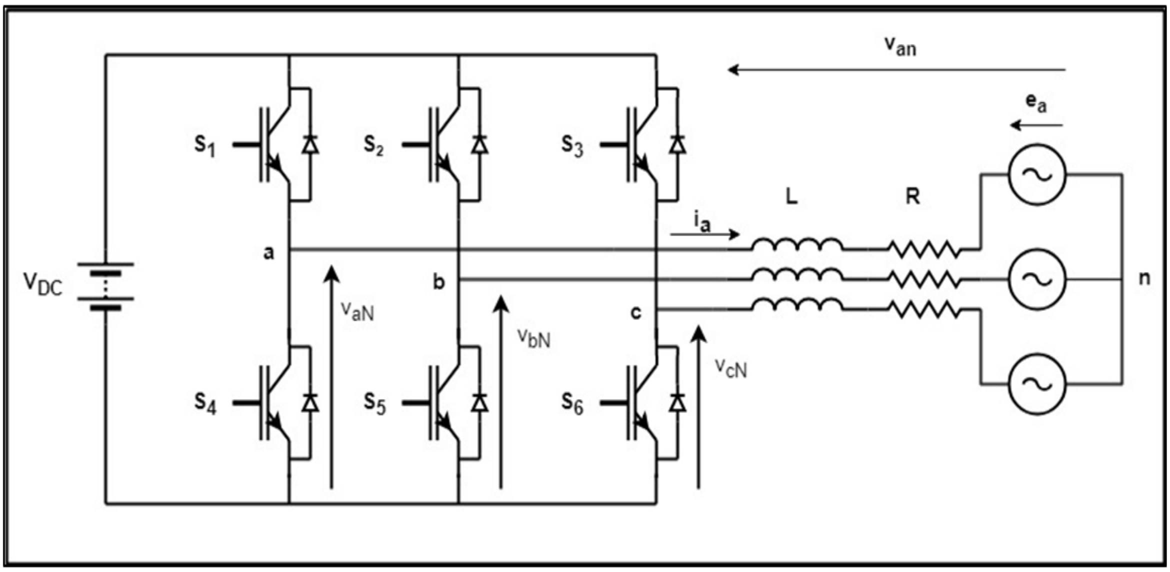


***Figure 2.*** *Two-Level Voltage Source Inverter.*

There have been several techniques that have developed over the years to control power converters. These include classical control techniques such as Hysteresis Control which can be subdivided into direct torque control (DTC), direct power control (DPC) and current control. Another classical control technique is the Linear Control method which includes field-oriented control (FOC) and voltage-oriented control (VOC). Advance control techniques for power converters have been developed over the years and these methods include intelligent controllers such as fuzzy and artificial neural network control, sliding mode technique and predictive controls [7].

Predictive control can be further subdivided into several control techniques. These categories are Deadbeat Control, Hysteresis based Predictive Control, Trajectory based Predictive Control and Model Predictive Control (MPC) [8]. We will focus on The Model Predictive Control in this paper.

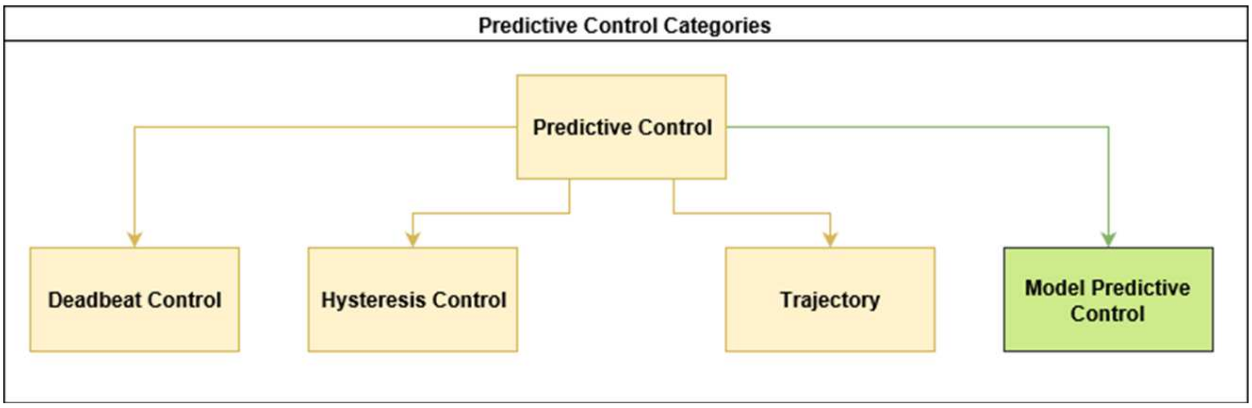


***Figure 3.*** *Predictive Control Categories.*

In recent years, the implementation of MPC to control power converters has been beneficial due to its' predictive control, inherent feedback and constraints inclusion. The control action in MPC is performed by minimizing the cost function that defines the systems behavior. Comparison between the predicted output of the system and reference is achieved through the cost function. The predicted system output is calculated from the model of the system. At every sampling instant, the MPC algorithm is repetitive in a receding horizon fashion [9, 10].

The conventional MPC technique is basically executed through current-oriented control method. The reference value is tracked by the current by exploiting the discrete behavior of the converter. Each of the eight possible switching states is applied in the calculations of the future current of the converter and the state used to fire the power switches is selected through the minimization of the cost function. A major drawback of the conventional MPC is predicting the future current due procedures' execution time. To improve the execution time, Lyapunov function is interfaced with MPC which minimizes the execution time by selecting the optimum voltage vector from the converter in which tracks the calculated future reference voltage vector. In this case, Lyapunov function with MPC is based on voltage-oriented control [11, 12].

In [18], a non-linear multi-parametric predictive control model is considered for micro grid supplied by renewable energy sources and with flywheel energy storage system is proposed. Control algorithm was designed for implementation on an Field-Programmable gate array (FPGA) board, for its the great advantages of reducing the overall system cost, size, and real-time operation.

A low-cost model predictive control (MPC) system is proposed in [19]. The system uses a Raspberry Pi 3 board to implement the MPC in real time. Experimental results show that despite the technical limitations and the high computational cost that this controller presents high performance and good response in the presence of noise. In the next section, we present the system design methodology.

## 2. Algorithm Design and Methodology

The Lyapunov function associated with the Model Predictive Control (MPC) algorithm incorporates multiple variables and parameters. The proposed MPC algorithm associated with the Two–Level Voltage Source Inverter model was analyzed and evaluated [8]. The algorithm is coded in MATLAB, while Simulink is used to implement and study the Load and Inverter blocks [11].

To validate the proposed control algorithm, the design is carried out in hardware, downloaded and executed on an Embedded Processor board. The MPC algorithm split into two separate Datapath and Controller sections including various general and special registers, arithmetic/logic units and control signals [14].

The creation of the system datapath and controller provided a clear template of how the hardware of the embedded processor has to be programmed. Knowing the layout of registers and various control signals as well as data lines allowed for somewhat simpler coding, especially in coding with VHDL. Straight coding in VHDL of certain components is done but this proved more tedious than building the system using the model blocks of Xilinx's System Generator. The Xilinx System Generator allows the construction of the system within Matlab/Simulink environment using the Xilinx Blocks libraries. One of the main advantages of using the System Generator is its Gateway In and Gateway Out blocks that converts data from floating point and fixed point data types, and vice versa. Thus, using the Xilinx System Generator, the control algorithm is implemented. This is done in sections, for example, reading the input phase currents and converting it into the reference current vector ($\mathrm{I_{ref}(k)}$), and tests and then ultimately combining all components together to form a complete system. The design is then computer simulated, synthesized and finally downloaded onto the Spartan 6 FPGA board for final testing. The results and performance of the system design is then analyzed.

The analysis of the applied Lyapunov Law to the modified Model Predictive Control was attained in [11] and it can be noted that the working principle of the studied Two – Level Voltage Source Inverter was gathered from [8] to obtain the voltage space vectors for the studied voltage source inverter shown below in table 1.

***Table 1.*** *Two-level VSI Voltage Space Vectors.*

| Switching States | | | Voltage Space Vectors |
|---|---|---|---|
| **Sa** | **Sb** | **Sc** | $\vec{v}_{inv}$ |
| 0 | 0 | 0 | $V_0 = 0$ |
| 1 | 0 | 0 | $V_1 = \frac{2}{3}V_{dc}$ |
| 1 | 1 | 0 | $V_2 = \frac{1}{3}V_{dc} + j\frac{\sqrt{3}}{3}V_{dc}$ |
| 0 | 1 | 0 | $V_3 = -\frac{1}{3}V_{dc} + j\frac{\sqrt{3}}{3}V_{dc}$ |
| 0 | 1 | 1 | $V_4 = -\frac{2}{3}V_{dc}$ |
| 0 | 0 | 1 | $V_5 = -\frac{1}{3}V_{dc} - j\frac{\sqrt{3}}{3}V_{dc}$ |
| 1 | 0 | 1 | $V_6 = \frac{1}{3}V_{dc} - j\frac{\sqrt{3}}{3}V_{dc}$ |
| 1 | 1 | 1 | $V_7 = 0$ |

The modified Model Predictive Control technique implements the voltage vector directly from the fixed set. Therefore, the future reference voltage vector $\left(\vec{\mathrm{v}}(\mathrm{k}+1)\right)$ is equal to the sum of the generated voltage vector $(\vec{\mathrm{v}}_{\mathrm{inv}}(\mathrm{k}+1))$ from the voltage source inverter model and the inevitable quantization error vector $\left(\delta(\mathrm{k}+1)\right)$.

$$\vec{v}(k+1) = \vec{v}_{inv}(k+1) + \delta(k+1) \qquad (1)$$

The quantization error vector satisfies $\left|\left|\delta(\mathrm{k}+1)\right|\right| \leq \varphi$ where the constant $\varphi > 0$. It is noted that the future voltage vector is bounded in the fixed set as shown in the above in table 1.

Analysis of the inverters control point of view is made to determine the modification of the conventional Model Predictive Control to the studied Lyapunov based MPC. Hence, the future current error vector can be assessed as such:

$$\vec{\mathrm{i}}_{\mathrm{error_{inv}}}(\mathrm{k}+1) = \vec{\mathrm{i}}_{\mathrm{s_{inv}}}(\mathrm{k}+1) - \vec{\mathrm{i}}_{\mathrm{ref_{inv}}}(\mathrm{k}+1) = \frac{1}{\mathrm{RT_s+L}}\left[\mathrm{L}\vec{\mathrm{i}}_{\mathrm{s_{inv}}}(\mathrm{k}) + \mathrm{T_s}\left(\vec{\mathrm{v}}_{\mathrm{inv}}(\mathrm{k}+1) - \vec{\mathrm{v}}_{\mathrm{s}}(\mathrm{k}+1)\right)\right] - \vec{\mathrm{i}}_{\mathrm{ref_{inv}}}(\mathrm{k}+1) \qquad (2)$$

The Lyapunov method is directly applied for the current tracking error to asymptotically converge to zero. The discrete Lyapunov function L (k) is evaluated as:

$$\mathrm{L(k)} = \frac{1}{2}\left(\vec{\mathrm{i}}_{\mathrm{error}}(\mathrm{k})\right)^{\mathrm{T}}\left(\vec{\mathrm{i}}_{\mathrm{error}}(\mathrm{k})\right) \qquad (3)$$

Implementing equations (2) and (3) will give the rate of change of the Lyapunov Function as:

$$\Delta L(k) = L\left(\vec{i}_{error_{inv}}(k+1)\right) - L\left(\vec{i}_{error_{inv}}(k)\right) = \frac{1}{2}\left(\begin{matrix} \frac{1}{RT_s+L} * \left[L\vec{i}_{s_{inv}}(k) + T_s\left(\vec{v}_{inv}(k+1) + \delta(k+1) - \vec{v}_s(k+1)\right)\right] \\ -\vec{i}_{ref_{inv}}(k+1) \end{matrix}\right)^T * \left(\frac{1}{RT_s+L} * \left[L\vec{i}_{s_{inv}}(k) + T_s\left(\vec{v}_{inv}(k+1) + \delta(k+1) - \vec{v}_s(k+1)\right)\right] - \vec{i}_{ref_{inv}}(k+1)\right) - \frac{1}{2}\left(\vec{i}_{error}(k)\right)^T\left(\vec{i}_{error}(k)\right) \quad (4)$$

The derivative of the Lyapunov Function always needs to be negative to create an effective control algorithm that converges the tracking error to zero. Thus, the discrete Lyapunov future reference voltage vector at the next sampling instant which satisfies that the derivative of the Lyapunov Function is negative is shown below:

$$\vec{v}(k+1) = -\frac{L}{T_s}\vec{i}_s(k) + \vec{v}_s(k+1) + \frac{RT_s+L}{T_s}\vec{i}_{ref}(k+1) \quad (5)$$

Figure 4 shown below presents the modified Model Predictive Control (MPC) process.

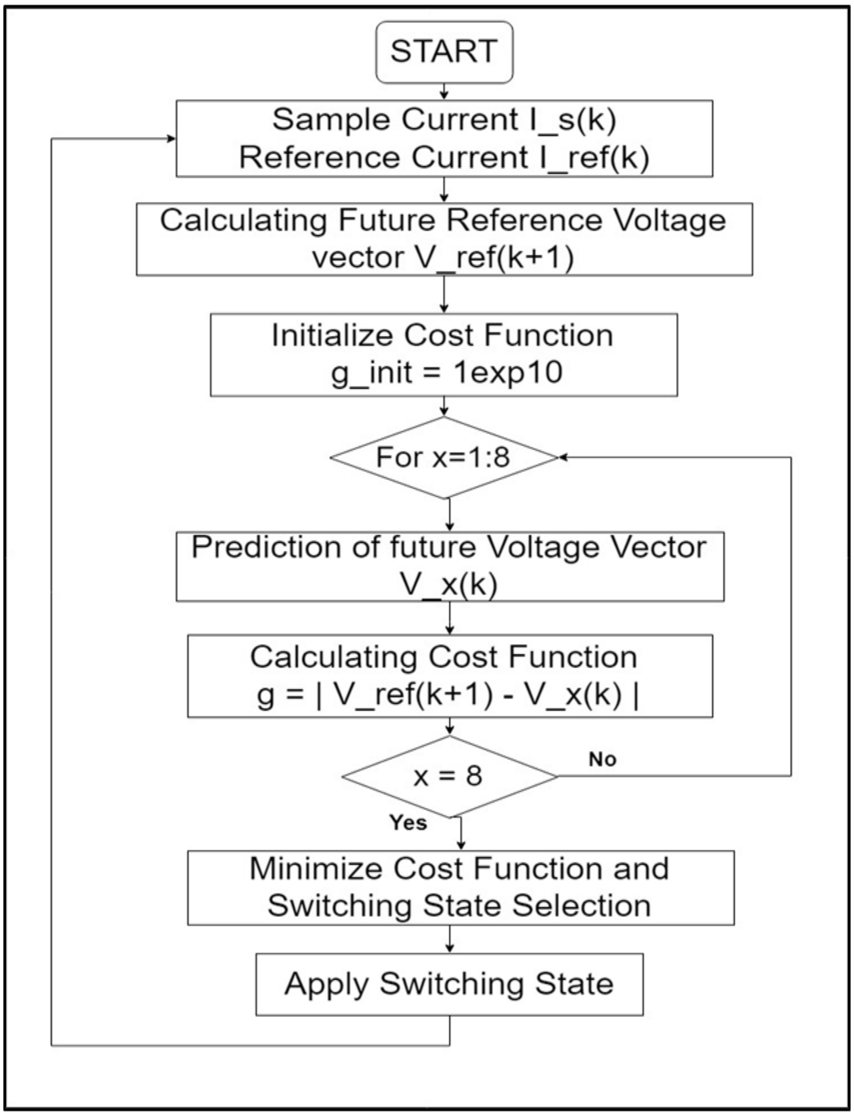


***Figure 4.*** *Modified Model Predictive Control algorithm.*

We present the system implementation and prototyping in the next section.

## 3. System Implementation and Results

The modified control algorithm is implemented in MATLAB/Simulink using an existing Model Predictive Control for a Two–Level Voltage Source Inverter [8]. The modified to Lyapunov based controller calculates the Future Reference Voltage vector. Computer simulations are carried out with modification made to the control algorithm within function blocks. Figure 5 presents the Simulink model with modified control.

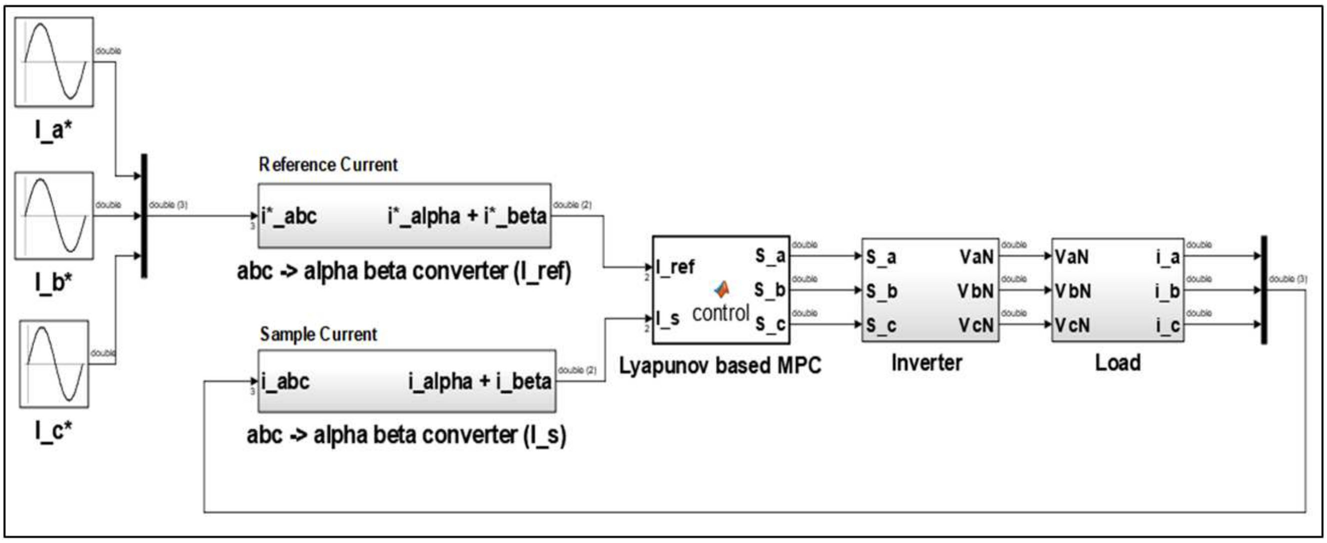


***Figure 5.*** *Simulink model with modified controller function.*

The signal generator blocks represented by phases (I_a*, I_b*, I_c*) generated the reference current that were fed into the converter block to obtain the alpha and beta signals to be used as I_ref in the modified control function block. The conventional MPC block from the studied text was modified to the Lyapunov based MPC control block shown above. This was done to verify whether the currents generated by the Load block would track the reference current signals as performed by the conventional MPC algorithm.

In Figure 6, it can be noted that the load currents that are fed into the alpha and beta converter blocks depicted in Figure 5 generate the sampled alpha and beta signal respectively which track the reference alpha and reference beta current signals. The Load Response Current (i_alpha, i_beta) and Reference Current (i*_alpha, i*_beta) with step in reference amplitude.

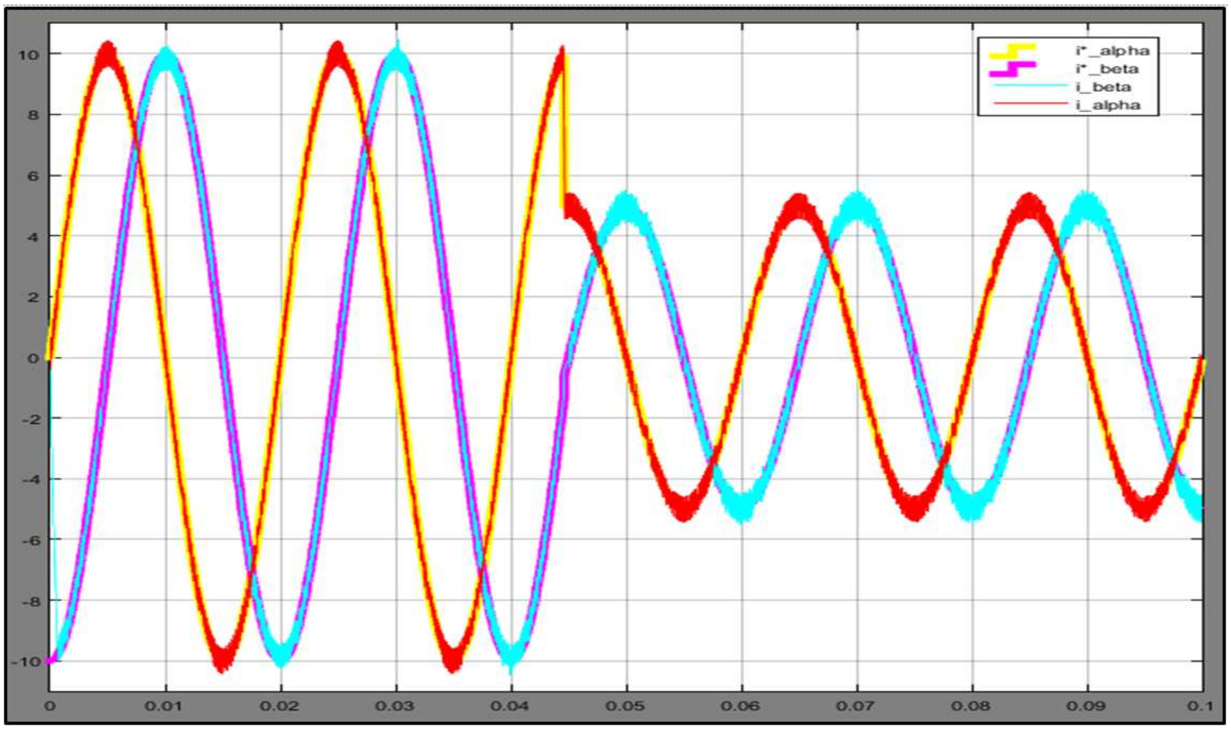


***Figure 6.*** *Load Response Vs. Reference current.*

Figure 7 shows the fast-dynamic response of the i_alpha current signal when an applied step from 10–5 A is executed onto the reference currents. It is noticeable how the Load Response Current (i_alpha, i_beta) is tracking the Reference Current (i*_alpha, i*_beta).

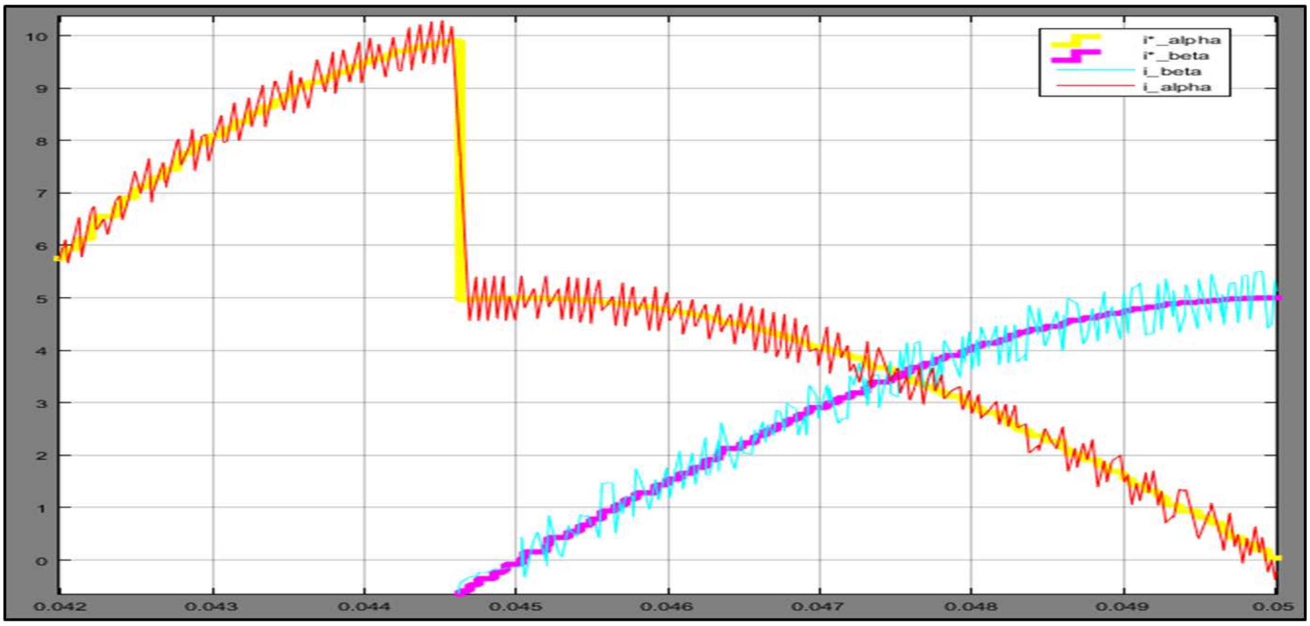


***Figure 7.*** *Load Response tracking Reference current.*

To validate the proposed modified MPC algorithm, the design was implemented as an embedded processor board. The algorithm design is divided into system datapath and a controller part. The datapath highlights the number of needed registers, the arithmetic and logic units as well as any other units such as shifters that may be needed and how all these components interact with one another in the forms of data transfer between registers over data, address and control buses. Figure 8 presents the Controller and datapath of the MPC embedded processor.

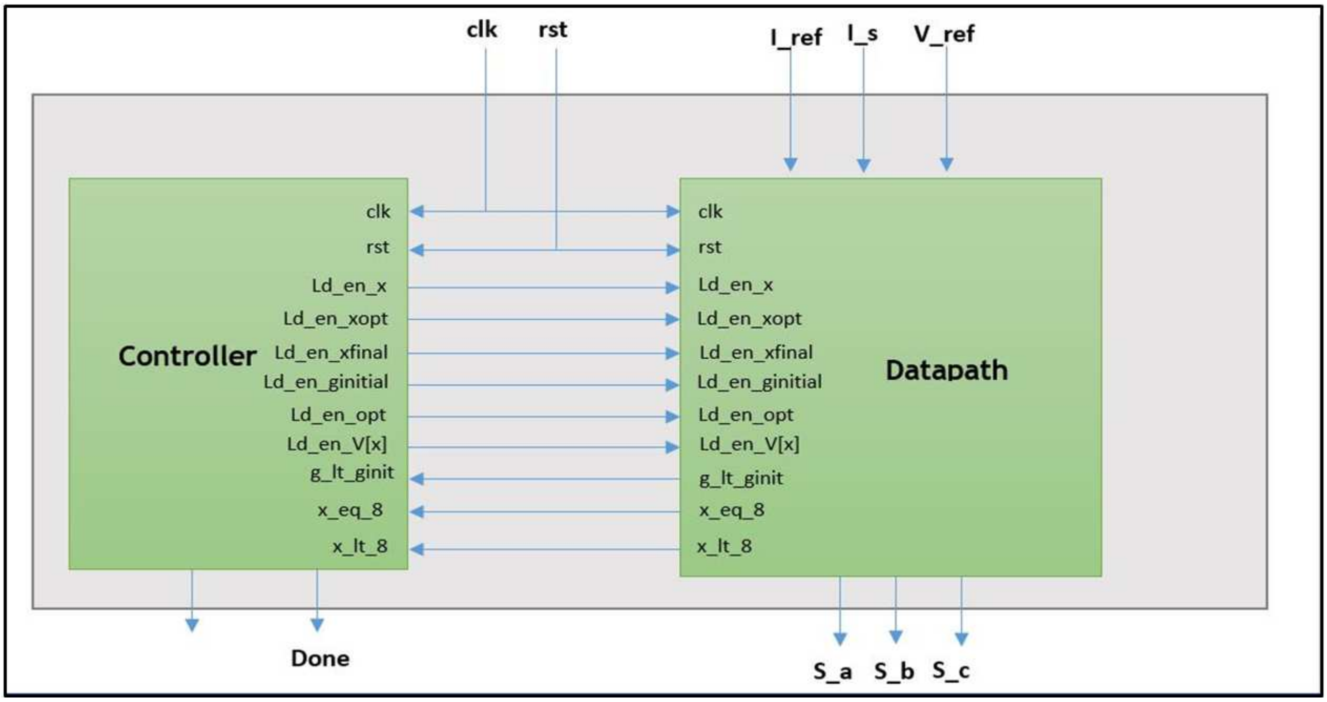


***Figure 8.*** *MPC processor design.*

A top-level datapath and controller diagram is shown above with an accompanying table describing various control and status signals as shown in table 2.

***Table 2.*** *Control and status signals.*

| NAME | SIGNAL TYPE | DESCRIPTION |
|---|---|---|
| ld_en_x | Control | Control's data flow into the X register<br>Determined by the x_lt_8 signal |
| ld_en_xopt | Control | Loads optimum X into the x_OPT register<br>Enabled after every comparison between cost function and initial cost function.<br>Depends on g_lt_ginit signal |
| ld_en_xfinal | Control | Loads final optimum X value into the xfinal register<br>Determined by the x_eq_8 signal |
| ld_en_ginitial | Control | Loads g_init value into g_inital register<br>Also depends on g_lt_ginit signal. |
| ld_en_gopt | Control | Loads g_opt value into g_opt register<br>Also depends on g_lt_ginit signal. |
| Ld_en_V[x] | Control | Control's data flow into the V[x] register<br>Determined by the x_lt_8 signal |
| g_lt_ginit | Status | Controls Ld_en_V[x], ld_en_ginitial and ld_en_xopt signals |
| x_eq_8 | Status | Controls ld_en_xopt signals |
| x_lt_8 | Status | Controls Ld_en_x and ld_en_xfinal signals |

The state diagram of the controller is shown below in figure 9.

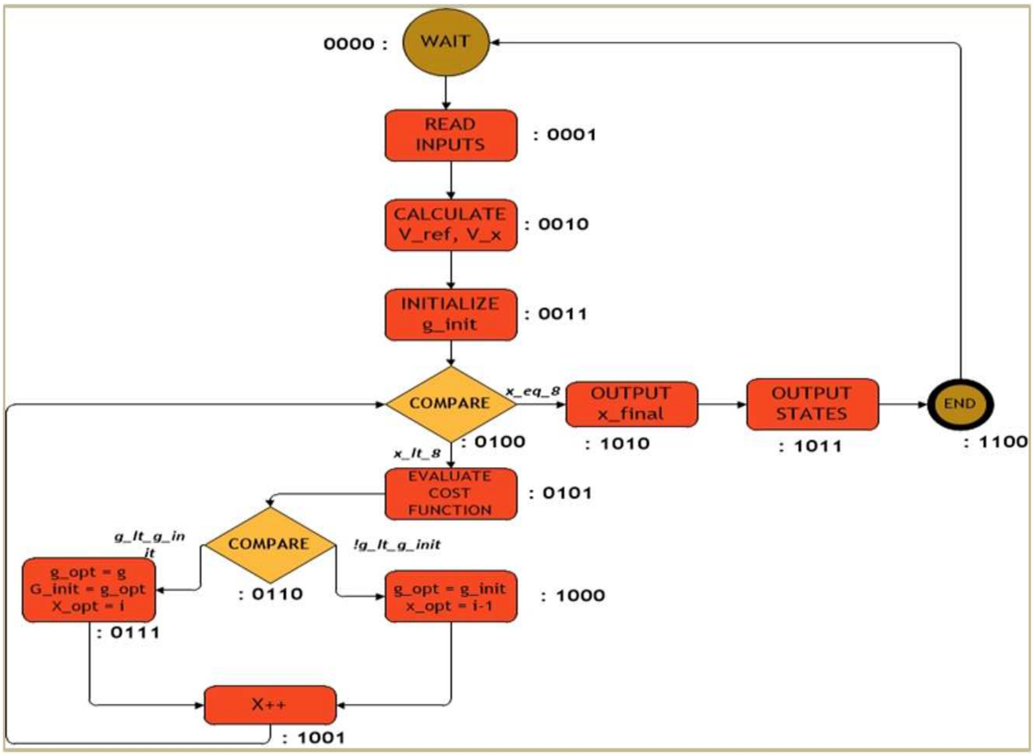


***Figure 9.*** *System controller.*

The initial state is the wait (idle) state 0, the input variables are read into registers in state 1. These variables include the reference current ($I_{ref}(k)$), the sampled current ($I_s(k)$) and the sampled voltage ($V_s(k)$). It must be mentioned that these are all vector values and as such have real and imaginary values. Since, FPGAs do not recognize this distinction, the real and imaginary components of these and other complex values within the system have been stored in separate, corresponding registers. For example, for the sampled current ($I_s(k)$), it is stored in two separate registers as *I_s_real* and *I_s_imaginary*. For the remainder of this discussion, these complex values will be referred to as the variable or register name only, without mention of its real and imaginary components. Any operation performed on a variable or register implicitly involves both of its complex components. Next, in state 2, the future reference voltage vector ($V_{ref}(k+1)$) is calculated using equation (5). This is seen in the datapath where the content of the input variable registers are passed through various adders, subtractors, multipliers and dividers and output into the register for future reference voltage vector.

Then, after the cost function is initialized in state 3, the algorithm's *for-loop*, is implemented with the use of a comparator unit. State 5 involves the evaluation of the cost function *(g)* wherein the future reference voltage vector is subtracted from the optimal voltage vector of that for-loop iteration *(i)*. The absolute value of the difference of the voltages is taken and compared to the initialised cost function value in state 6. Depending on the result of the comparison, either state 7 or 8 will be executed along with the corresponding actions.

Next, in state 9 the value of x is incremented, and the loop is reverted to state 4. State 10 involves outputting the optimum x value and state 11 involves outputting the corresponding switching state with sate 12 being the finish state.

Then after, the content of these input variables registers are passed through various ALUs so as to implement equation (5). The result is then stored in a register for future reference voltage vector (($V_{ref}(k+1)$)).

Figures 10 and 11 present the system datapath sections.

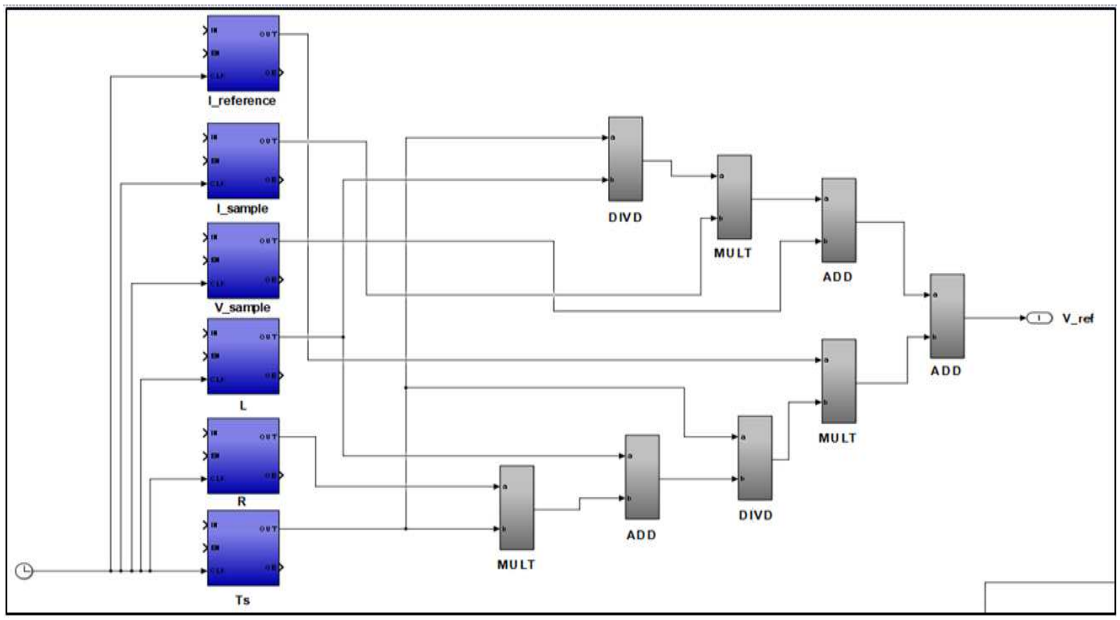


***Figure 10.*** *Datapath.*

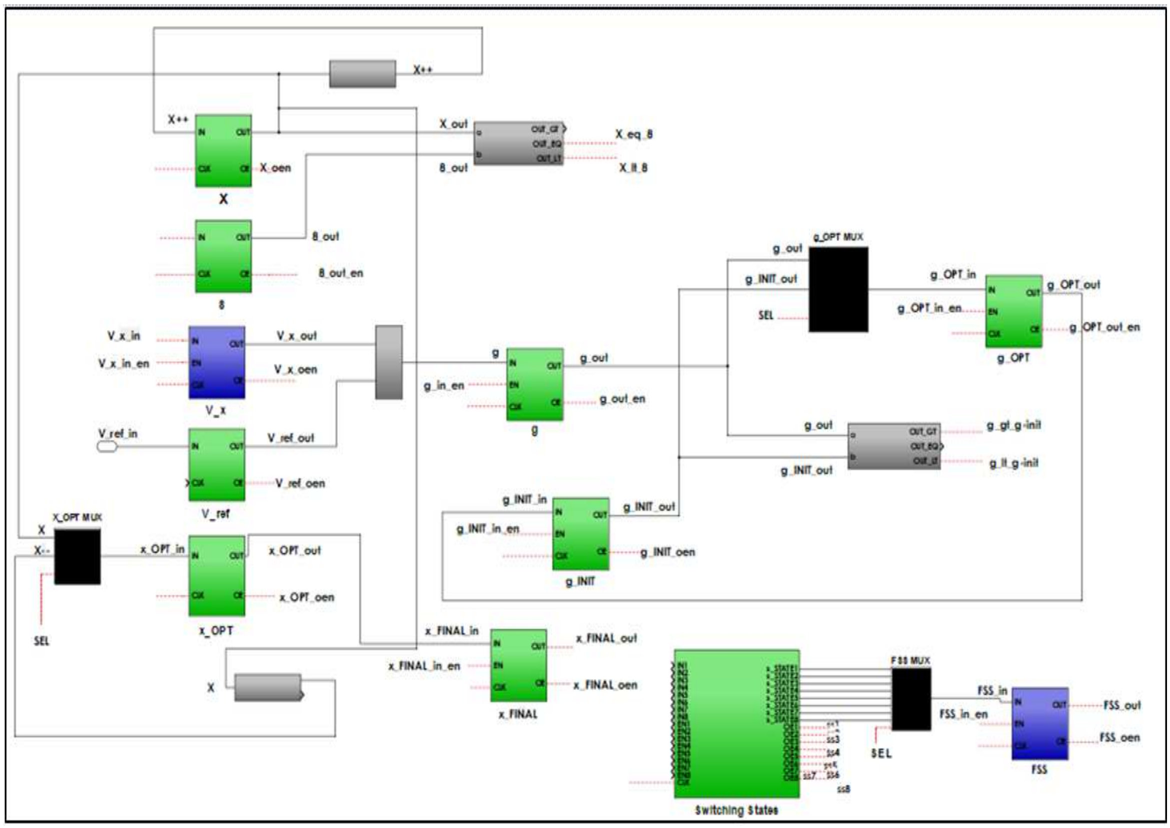


***Figure 11.*** *Datapath (Con't).*

The MPC controller design is implemented in Xilinx's System Generator. Shown in Figure 12 is the entire VSI and

MPC controller system. Its main subsystems are the Clarke's Transformation block labelled here as alpha_beta blocks, the Lyapunov MPC block and the inverter and load blocks. The blocks between the *Gateway In* and *Gateway Out* are Xilinx Blockesets whilst those outside are from the Simulink environment.

The three phase currents generated by the signal generator blocks on the far left pass through the *Gateway In* blocks and then undergo Clarke's Transformation so as to give the alpha and beta components of the reference current ($I_{ref}(k)$). These are then sent into the Lyapunov MPC block which contains the controller algorithm. The output of this block is the switching states which are outputted through *Gateway Out* blocks and into the inverter model which is acted on by a load modelled in the Load Block. Lastly the load currents are sampled and passed through another Clarke's transformation block to give the alpha and beta components of the sampled current ($I_s(k)$). The overall system architecture is shown in figure 12.

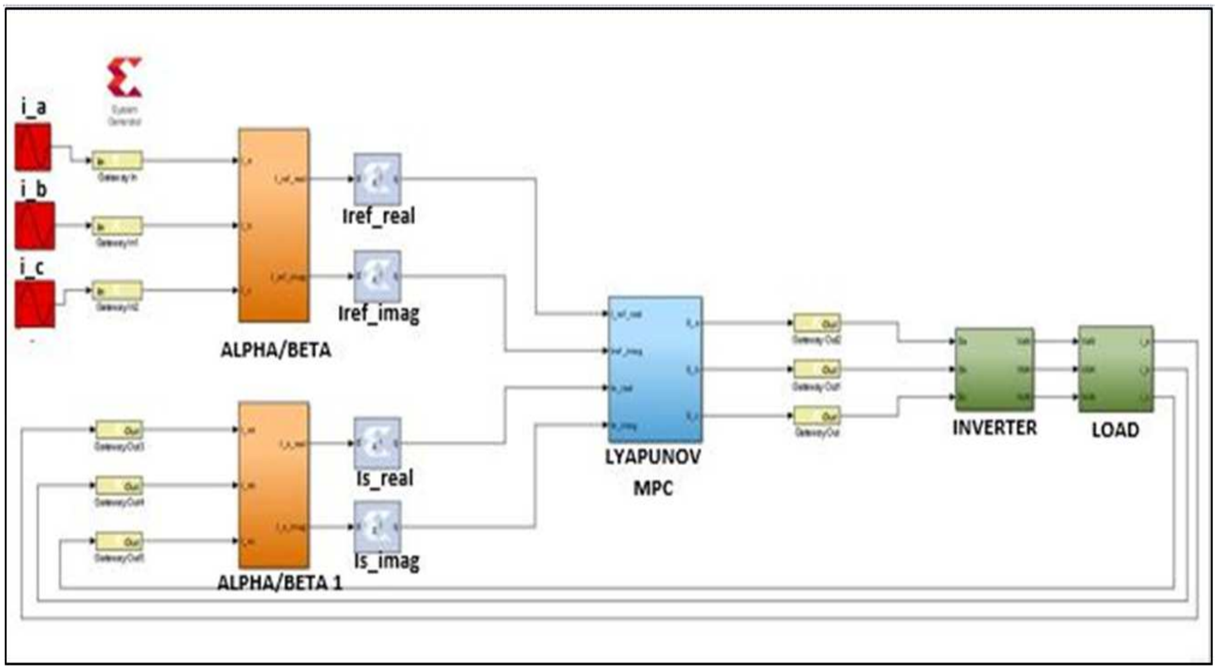


***Figure 12.*** *The overall system.*

System modules are put together to create a complete operational system. The system was simulated in software and validated on Xilinx Spartan FPGA board with fixed-point arithmetic and 200MHz clock speed as a digital control platform. The control objectives and Xilinx FPGA used resources are shown in table 3.

***Table 3.*** *Control objectives and FPGA resources.*

| Objectives | Combinational functions | Dedicated registers | Memory usage | Execution time |
|---|---|---|---|---|
| Load current | 17.40% | 7.20% | 19.00% (bits) | 1.71 (μsec) |
| Load current + Switching frequency | 20.90% | 7.85% | 18.75% (bits) | 1.95 (μsec) |

The use of a high clock speed (200 MHz) processor contributes to the reduction of execution time. We have measured the execution times of some calculations in the proposed Lyapunov-based controller implementation as shown in table 4.

***Table 4.*** *Calculations.*

| Task | Execution time |
|---|---|
| Initialization | 0.03 (μsec) |
| Source current prediction | 0.24 (μsec) |
| Source voltage prediction | 0.24 (μsec) |
| Load current prediction | 0.32 (μsec) |
| Evaluate cost function | 0.60 (μsec) |
| Optimization | 0.52 (μsec) |
| Decision | 0.32 (μsec) |
| Total | 2.27 (μsec) |

According to the above results, cost function evaluation and optimization take longer time compared to other calculation tasks. The total execution time for the implementation of the proposed control algorithm is 2.27μsec.

## 4. Conclusion

A modified MPC algorithm design and implementation is proposed in this paper. A detailed analysis of the Lyapunov function future reference voltage vector is presented. System design was carried out by using some Simulink models and by updating the control algorithm on the conventional MPC MATLAB function block which generated load currents to track the reference currents. The fast-dynamic response of the load current was noted when an applied change in the reference currents amplitude was performed. Algorithm design was validated in hardware on an embedded processor FPGA board. According to experimental results, model predictive control works well under both steady-state and transient conditions. The FPGA-based implementation provides good dynamic response as well. This work could be extended by applying the modified control algorithm to control a two-level VSI in real time.

## References


[1] A. Gebreel, "Simulation and Implementation of Two-Level and Three-Level Inverters by Matlab and RT-Lab," Ohio, 2011.

[2] M. Cirrincione, M. Pucci and G. Vitale, "Pulse Width Modulation of Voltage Source Inverters," in Power Converters and AC Electrical Drives with Linear Neural Networks, Taylor and Francis Group, 2012, p. 17.

[3] S. Tahir, J. Wang, M. H. Baloch and G. S. Kaloi, "Digital Control Techniques Based on Voltage Source Inverters in Renewable Energy Applications: A Review," Electronics, vol. 7, no. 2, pp. 3-8, 2018.

[4] A. Nabae, I. Takahashi and H. Akagi, "A new neutral-point-clamped PWM inverter," IEEE Transactions on Industry Applications, Vols. 1A-17, no. 5, pp. 518-523, 1981.

[5] L. Franquelo, J. Rodriquez, J. Leon, S. Kouro, R. Portillo and M. Prats, "The age of multilevel converters arrives," IEEE Industrial Electronics Magazine, vol. 2, no. 2, pp. 28-39, 2008.

[6] J.-S. Lai and F. Z. Peng, "Multilevel converters-A new breed of power converters," IEEE Transactions on Industry Applications, vol. 32, no. 3, pp. 509-517, 1996.

[7] S. Kouro, P. Cortez, R. Vargas, U. Ammann and J. Rodriques, "Model Predictive Control - A Simple and Powerful Method to Control Power Converters," IEEE Transactions on Industrial Applications, vol. 56, no. 6, p. 1826, 2009.

[8] J. Rodriquez and P. Cortes, "Chapter 4 Predictive Control of a Three-Phase Inverter," in Predictive Control of Power Converters and Electrical Drives, John Wiley and sons, 2012, pp. 43-63.

[9] I. Syed and K. Raahemifar, "MPC of a Single Phase Inverter," International Journal of Energy and Power Engineering, vol. 8, no. 11, pp. 1730-1735, 2014.

[10] S. Vazquez, J. Rodriquez, M. Rivera, L. Franquelo and M. Norambuena, "Model Predictive Control for Power Converters and Drives: Advances and Trends," IEEE Transactions on Industrial Electronics, vol. 64, no. 2, pp. 935-947, 2017.

[11] P. Akter, S. Mekhilef, N. Tan and H. Akagi, "Modified Model Predictive Control of a Bidirectional AC-DC Converter Based on Lyapunov Function for Energy Storage Systems," IEEE Transactions on Industrial Electronics, vol. 63, no. 2, pp. 704-715, 2016.

[12] A. Dadu, T. Soon, S. Mekhilef and M. Nakaoka, "Lyapunov Law Based Model Predictive Control Scheme for Grid Connected Three Phase Three Level Neutral Point Clamped Inverter," in IEEE 3rd International Future Energy Electronics Conference and ECCE Asia, 2017.

[13] E. Kerrigan, G. Constantinides, A. Suardi and B. Khuisanov, "Computer Architectures to Close the Loop in Realtime Optimization," in 2015 54th IEEE Conference on Decision and Control (CDC), 2015.

[14] S. Lucia, D. Navarro, O. Lucıa, P. Zometa and R. Findeisen, "Optimized FPGA Implementation of Model Predictive Control for Embedded Systems Using High-Level Synthesis Tool," IEEE Transactions on Industrial Informatics, vol. 14, no. 1, pp. 137-145, 2018.

[15] E. Monmasson, L. Idkhajine, M. N. Cirstea, I. Bahri, A. Tisan and M. W. Naouar, "FPGAs in Industrial Control Applications," IEEE Transactions on Industrial Informatics, vol. 7, no. 2, pp. 224-243, 2011.

[16] O. Gulbudak and E. Santi, "FPGA-Based Model Predictive Controller for Direct Matrix Converter," IEEE Transactions on Industrial Electronics, vol. 63, no. 7, pp. 4560-4570, 2016.

[17] P. Sanchez, O. Machado, E. Pena, F. Rodriguez and F. Meca, "FPGA-Based Implementation of a Predictive Current Controller for Power Converters," IEEE Transaction on Industrial Informatics, vol. 9, no. 3, pp. 1312-1321, 2013.

[18] H. K. Mudaliar, D. M. Kumar, S. Rokocakau, K. Ram, M. Cirrincione, and M. Assaf, "FPGA-based Multi-parametric Non-linear Model Predictive Control for a Micro-Grid in Small Islands", International Conference on Electrical Machines and Systems (ICEMS 2018), Jeju, Korea, October 2018.

[19] A. Dadu, T. Soon, S. Mekhilef and M. Nakaoka, "Real-Time Implementation of Model Predictive Control in a Low-Cost Embedded," in the 9th International Multi-Conference on Complexity, Informatics and Cybernetics (IMCIC 2018), Orlando, USA, October 2018.