INTEGRATION OF INVERTER CONSTRAINTS IN GEOMETRICAL QUANTIFICATION OF THE OPTIMAL SOLUTION TO A MPC CONTROLLER

J. Raath * and H. du Toit Mouton† and T. Geyer‡

* *Department of Electrical, Electronic and Computer Engineering, Central University of Technology, Bloemfontein, South Africa, E-mail: jraath@cut.ac.za*

† *Department of Electrical and Electronic Engineering, University of Stellenbosch, Cape Town, South Africa, E-mail: dtmouton@sun.ac.za*

‡ *ABB Corporate Research, ABB Switzerland Ltd, Power Electronic Systems, Switzerland, E-mail: t.geyer@ieee.org*

Abstract: This paper considers a model predictive controller with reference tracking that manipulates the integer switch positions of a power converter. It can be shown that the optimal switch position can be computed without solving an optimization problem. Specifically, in a new coordinate system, the optimization problem can be solved offline, leading to a polyhedral partition of the solution space. The optimal switch position can then be found using a binary search tree. This concept is exemplified for a three-level single-phase converter with an RL load.

Key words: Model Predictive Control, Vector Quantization, Delaunay triangulation, Voronoi diagram, Binary Search Tree.

1. INTRODUCTION

Model predictive control (MPC), also referred to as receding horizon control is a type of predictive control which in general uses a model of the system to predict the future behavior of the controlled parameters. The predictions are then used to obtain the optimal control decision by following a specific optimization criterion. Traditional MPC demanded a great amount of online computation, since an optimization problem (often a constrained quadratic program) is solved at each sampling instant. This has limited the use of these controllers to processes with relatively slow dynamics but because of advances in the fields of mathematical optimization and computational power of the controller hardware it became possible to consider MPC in power electronic systems with short sampling intervals [1]. MPC using larger horizons also has the potential to give significant performance benefits, but requires more computations at each sampling instant to solve the associated optimization problem [2, 3]. The online computational burden of MPC can be lessened by obtaining an solution to the MPC problem offline by means of multi-parametric quadratic programming (mpQP) [4, 5]. The offline solution is a state-feedback control over a polyhedral partition of the state-space. The control law can be stored in a look-up table, avoiding the need for solving an optimization problem online [6].

The main purpose of our research is to reduce the online computational burden so as to practically implement MPC for a multilevel inverter. To achieve this goal it is necessary in the offline solution to construct a binary search tree with minimum depth, which can only be achieved if the partitioned state-space is of lowest complexity. This paper presents an algorithm for reducing the complexity of the partitioned state-space by utilizing the Delaunay triangulation. The paper is organized as follows: Section 2 introduces the model of a multilevel inverter with *RL* load. The mathematical background to the MPC problem is laid out in section 3. In section 4 the partitioned state-space is geometrically presented. The approach to complexity reduction along with the proposed algorithm are presented in section 5. Section 6 concludes the paper.

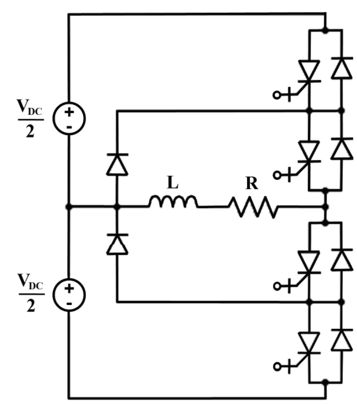

Figure 1: Single-phase Neutral Point Clamped inverter.

2. MODELING

We consider a single-phase multilevel inverter as one leg of a Neutral Point Clamped (NPC) inverter with neutral point assumed to be constant. The topology is shown in figure 1. The inverter leg can deliver three voltage levels of $-0.5V_{DC}$, $0V_{DC}$ and $+0.5V_{DC}$ across the load. These output levels can be represented by the integer values $u \in \{-1, 0, +1\}$ that define the state of the switch positions in the inverter leg. The voltage applied to the *RL* load can thus be stated as $v(t) = 0.5V_{DC} \cdot u(t)$. A possible destructive situation that can occur in this inverter topology is switching directly from state $+1$ to state -1 and vice versa. It is called shoot-through and can lead to high currents in the inverter leg. This transition is undesirable. For detailed operation and switching sequence of a NPC inverter refer to [7].

3. MODEL PREDICTIVE CONTROL

Model predictive control is a method in which the control action is determined by solving a finite horizon open-loop optimal control problem at each sampling instant, using the current state of the system as initial state, searching for an optimal control sequence over the set horizon and then applying the first control in this sequence to the system. With reference tracking the general aim is to control the inverter switches in such a manner so as to generate an output current i in the *RL* load that tracks a reference current i_r as close as possible. The closer the output tracks the reference, the lower the harmonic distortion will be. With the switching frequency being inversely proportional to harmonic distortion in the output current and directly proportional to internal switching losses of the inverter, a trade-off between current distortions and switching losses arises.

An MPC controller operates in the discrete time domain with the sampling interval T_s. During every sample period the load current is sampled and from its value all the possible future load currents are determined which may arise from applying the different switching states also called control options, $u \in \{-1, 0, +1\}$ to the inverter. The effect of the possible voltages, $v(t) = 0.5V_{DC} \cdot u(t)$ applied to the mathematical model of the load results in a number of possible load currents equal to the number of control options. These predicted currents are subjected to a cost function as one of the control objectives that define the control system. The switching state u that result in a predicted current of minimal cost is selected as the optimal control input for application to the inverter. The cost function for our application includes the two contradictory objectives, optimal reference-current tracking and minimal switching cost.

3.1 Load model

In order for the MPC controller to predict the possible currents in the load, a mathematical model for the system needs to be derived. The *RL* load equation in the continuous time domain,

$$v(t) = Ri(t) + L\frac{di}{dt}$$

can be rewritten as,

$$\frac{di(t)}{dt} = -\frac{Ri(t)}{L} + \frac{0.5V_{DC}(t) \cdot u(t)}{L} \tag{1}$$

With the controller operating at discrete time instants $t = kT_s$ and $k \in \mathbb{N}$ the load model can be redefined in the discrete-time domain with $u(k)$ as input vector and $i(k)$ as state vector. The predicted load current at the discrete time step $(k+1)$ originating from the present output current value $i(k)$ for an applied control option $u(k)$, can be stated as,

$$i(k+1) = Ai(k) + Bu(k) \tag{2}$$

with,

$$A = e^{-T_s/\tau}, \qquad B = \frac{V_{DC}}{2R}(1 - e^{-T_s/\tau}),$$
$$\tau = \frac{L}{R}.$$

3.2 Cost function

To find the optimal control input to the inverter, all the predicted load currents $i(k+1)$ and the respective switching states $u(k) \in \{-1, 0, +1\}$ are subjected to a cost function (J),

$$J = \|i_r(k+1) - i(k+1)\|_2^2 + \lambda_u \|\Delta_u(k)\|_2^2 \tag{3}$$

with,

$$\|\Delta_u(k)\|_2^2 = \|u(k) - u(k-1)\|_2^2 \tag{4}$$

This quadratic cost function J, consist of two terms. The first one determine the tracking error of the predicted load current $i(k+1)$ with respect to the reference current $i_r(k+1)$, and the second term determines the switching cost from the previous switching state $u(k-1)$ to $u(k)$. A tuning factor $\lambda_u > 0$ adjusts the balance between tracking error cost and switching cost. To avoid shoot through, the switching constraint $\|\Delta_u(k)\|_2^2 \leq 1$ must be adhered to. The switching state that satisfies this constraint and results in minimum cost is deemed the optimal control option $u_{opt}(k)$ for application to the inverter.

3.3 Extended horizons

Solving the problem stated above results in a control action after evaluating the predicted currents at *one* discrete time-step into the future, a horizon (N) of one. This horizon can be extended by determining the predicted currents over a finite number of time-steps into the future. The cost of the possible switching sequences are determined and result in an improved optimal control decision while still adhering to the the switching constraint. Equation (3) can be written as a finite horizon quadratic cost function,

$$J = \sum_{l=k}^{k+N-1} \|i_r(l+1) - i(l+1)\|_2^2 + \lambda_u \|\Delta_u(l)\|_2^2 \tag{5}$$

J is now a function of the *switching sequence* $U(k) =$

$[u(k), u(k+1), ...u(k+N-1)]$ over the prediction horizon N. The number of possible switching sequences that needs to be evaluated thus increases exponentially with an extension of the prediction horizon. With $U(k)$ the optimization problem for N-horizon can be formulated as,

$$U_{opt}(k) = \arg\min_{U(k)} J \tag{6}$$

subject to

$$i(l+1) = Ai(l) + Bu(l) \tag{7}$$

$$u(l) \in \{-1, 0, +1\}$$
$$\|\Delta_u(l)\|_2^2 \leq 1$$
$$\forall l = k, ..., k+N-1$$

3.4 Vectorization of cost function

In order to obtain a solution for the optimization problem (6) it is useful to vectorize the cost function (5). Iterating (7) for all l and rewriting in matrix notation gives,

$$I(k) = \Gamma i(k) + \Upsilon U(k) \tag{8}$$

with,

$$I(k) = \begin{bmatrix} i(k+1) \\ i(k+2) \\ i(k+3) \\ \vdots \\ i(k+N) \end{bmatrix}, \Gamma = \begin{bmatrix} A \\ A^2 \\ A^3 \\ \vdots \\ A^N \end{bmatrix},$$

and

$$\Upsilon = \begin{bmatrix} B & 0 & \cdots & 0 \\ AB & B & \cdots & 0 \\ A^2B & AB & B & 0 \\ \vdots & \vdots & \vdots & \vdots \\ A^{N-1}B & A^{N-2}B & \cdots & B \end{bmatrix}.$$

$I(k)$ then represents the predicted output currents over the finite horizon from time step $k+1$ to $k+N$. Substituting (8) into the cost function (5) results in,

$$J = \|\Gamma i(k) + \Upsilon U(k) - I_R(k)\|_2^2 + \lambda_u \|SU(k) - Eu(k-1)\|_2^2 \tag{9}$$

I_R is the reference current values over the prediction horizon. S and E are introduced to extract the switching cost over the extended horizon.

$$S = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ -1 & 1 & \cdots & 0 \\ 0 & -1 & 1 & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix} E = \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

Minimization of (9) and omitting all constraints, results in the well known expression for the unconstrained optimum [2] or unconstrained minimizer [6].

$$U_{unc}(k) = -Q^{-1}\Theta(k) \tag{10}$$

with,

$$Q = \Upsilon^T \Upsilon + \lambda_u S^T S$$
$$\Theta(k) = ((\Gamma x(k) - I_R(k))^T \Upsilon - \lambda_u (Eu(k-1))^T S)^T$$

and resulting cost function,

$$J = [U(k) - U_{unc}(k)]^T Q [U(k) - U_{unc}(k)] + const(k) \tag{11}$$

The Cholesky decomposition of Q produces a lower triangular matrix H so that $H^T H = Q$. Substituting in (11) gives,

$$J = \|HU(k) - HU_{unc}(k)\|_2^2 + const(k) \tag{12}$$

The constant term in (12) is independent from $U(k)$ and can be excluded with the resulting optimization problem,

$$U_{opt}(k) = \arg\min_{U(k)} \|HU(k) - HU_{unc}(k)\|_2^2 \tag{13}$$

subject to the switching constraint,

$$\|\Delta_u(l)\|_2^2 \leq 1, \forall l = k, ..., k+Np-1$$

3.5 Nearest Neighbor Quantization

In summary, the optimization problem (13) states that the optimum switching sequence for the finite horizon N, can be found as the minimum Euclidean distance from the optimal unconstrained vector $HU_{unc}(k)$ to any of the switching-sequence vectors $HU(k)$ in the coordinate system set by the H-transformation matrix with $H \in \mathbb{R}$. This translates into the nearest neighbor search of the multi-dimensional vector $HU_{unc}(k)$ to the finite set of output vectors $HU(k)$ in N-dimensional Euclidean space.

A general technique for solving the optimization problem (13) is the exhaustive search method. This method enumerates all possibilities and verifies if the switching constraint is satisfied. For example, consider an MPC current controller with reference tracking and horizon $N = 2$ for the single-phase three-level NPC inverter with an RL load as in figure 1. Steady state conditions with the following parameters are assumed. A sampling interval of $T_S = 25\mu$s, load- resistance of $R = 2\Omega$, and inductance $L = 2m$H. The rated rms output voltage of the inverter is $V_{AC} = 3.3k$V with an input dc-link voltage of $V_{DC} = 5.2k$V. Base quantities are used to establish a per unit system and the current reference is assumed to be 0.8pu amplitude at 50Hz. Applying a tuning factor of $\lambda_u = 0.02$ and the above

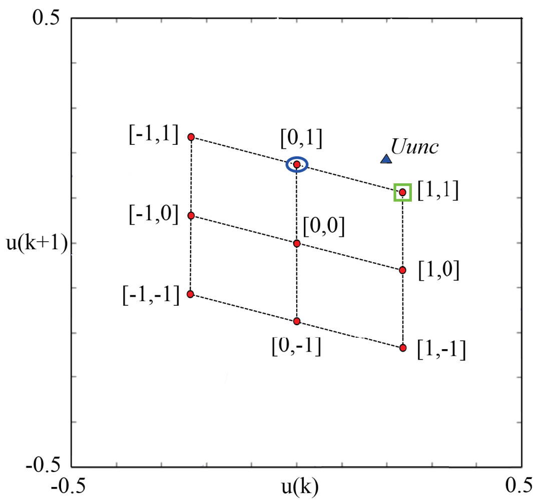

Figure 2: Transformed H-coordinate system

stated parameters generates an H-transformation matrix of,

$$H = \begin{bmatrix} 0.2286 & 0 \\ -0.0679 & 0.1711 \end{bmatrix}.$$

A horizon $N = 2$ results in a transformed H-coordinate system in the two-dimensional Euclidean space. Figure 2 shows the transformed H-coordinate system with possible switching sequences $U(k)$ indicated as dots. For explanation sake consider the unconstrained optimum $U_{unc}(k)$ indicated by the triangle. Assuming a previous switching state of $u(k-1) = -1$ and investigating the spatial arrangement of the vectors it is evident that the switching sequence $U(k) = [+1, +1]$ (enclosed by the rectangle) has the smallest Euclidean norm $\|HU(k) - HU_{unc}(k)\|_2^2$ and seems to be the the optimum solution but the first control action in this switching sequence is $u(k) = +1$. This value does not satisfy the switching constraint $\|u(k) - u(k-1)\|_2^2 \leq 1$ since $u(k-1) = -1$. Further investigation leads to the second nearest neighbor $U(k) = [0, +1]$ (enclosed by the ellipse) with $u(k) = 0$ which do satisfy the switching constraint. This sequence is then considered the optimal solution with $u(k) = 0 = u_{opt}(k)$ which is applied as control input to the inverter. This process of finding the optimal control input to the inverter repeats at every sampling interval, generating an inverter output voltage for application to the RL load which results in subsequent current flow. Figure 3 shows the simulated result of one cycle of the sinusoidal reference current i_r, switched inverter output voltage v and tracking load current i displayed in per unit values.

4. VECTOR QUANTIZATION

Although the exhaustive search technique suggested above is simple to implement, its computational cost is proportional to the number of possible solutions which increases exponentially as the horizon is extended.

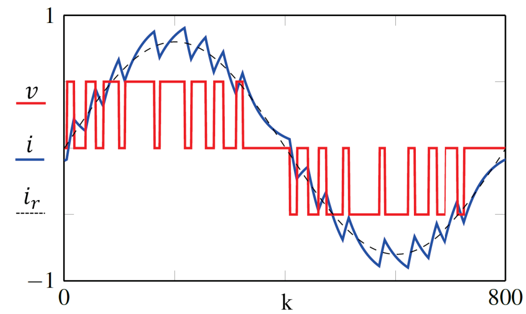

Figure 3: Inverter output voltage and reference tracking load current.

Therefore, it is only used for short horizon solutions [8]. Various solution algorithms for (13) have been developed but only a recent initiative by [2] incorporated the switching constraint $\|u(k) - u(k-1)\|_2^2 \leq 1$ into an very effective adaptation of the Sphere decoding algorithm. In contrast, this research is aimed at exploring the H-space and attempting to compute a geometrical solution in the format of a binary search tree (BST) to solve the MPC problem.

4.1 Voronoi partitioning

A Voronoi diagram of a set of points also called sites or seeds, is the partition of $\mathbb{R}$ Euclidean space into convex polyhedra of points nearest to each of the sites [9]. Each of these polytopes is called a Voronoi- or Dirichlet cell. The nearest neighbor search or quantification of the vector HU_{unc} can be done by partitioning the N-dimensional H-space ($H \in \mathbb{R}$) into a finite subset of Voronoi cells from the HU-sites and then determine in which of these cells HU_{unc} resides. In our case, the Voronoi diagram for 3^N number of HU-sites can be defined as the following set of polytopes,

$$V_i = \left\{x : \|x - HU_i\| < \left\|x - HU_j\right\|\right\} \tag{14}$$

for

$$i = 1, 2, , 3^N, \forall j \neq i$$

The vector HU_{unc} thus resides in a Voronoi cell V_i corresponding to a site HU_i if,

$$\|HU_{unc} - HU_i\| < \left\|HU_{unc} - HU_j\right\| \tag{15}$$

for

$$i = 1, 2..3^N, \forall j \neq i$$

Figure 4 graphically illustrates the partitioned H-coordinate space into nearest-neighbor Voronoi cells for the respective HU-sites. The most immediate way of determining in which polyhedral region HU_{unc} resides is to do a sequential search through all the regions but this is computationally expensive and not viable for application in higher dimensions [10]. By example, to determine the location of HU_{unc} as stated in (15), requires

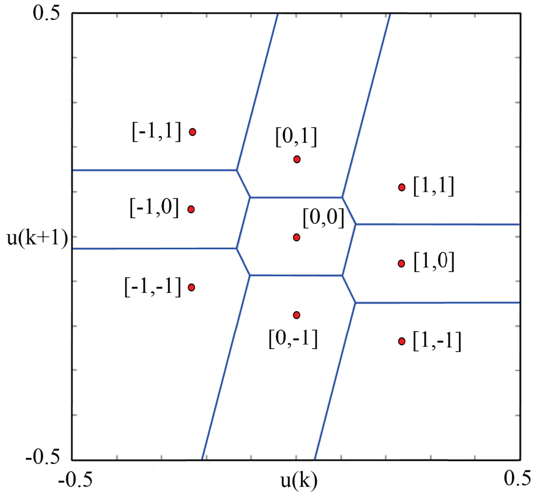

Figure 4: Voronoi partitions for HU-sites in the transformed H-coordinate space.

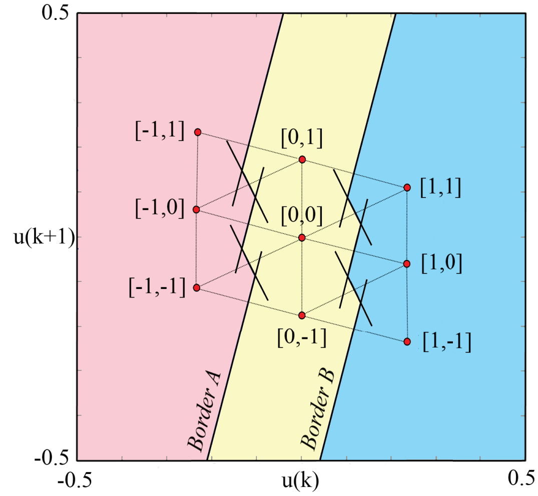

Figure 5: Unified Voronoi partitions in the transformed H-coordinate space.

the linear investigation ($A^T x - b$) of 16 hyperplanes that defines the 9 polyhedral regions (figure 4), $N = 3$ results in 27 polyhedra with 98 hyperplanes and $N = 4$ results in 81 polyhedra with 544 hyperplanes. Thus finding the region wherein HU_{unc} resides will result optimal control sequence $U(k)$ and hence $u(k)$, the optimal control action for application to the inverter.

5. COMPLEXITY REDUCTION

From the example above it is evident that the complexity rapidly increases with an extension of the horizon resulting in higher dimensionality of the H-space. The standard approach to complexity reduction is to unify adjacent polyhedral partitions with similar control laws $u(k)$. From figure 4 it can be noticed that some of the $HU(k)$-sites have the same first term value and thus represent the same control law. Unification of the 9 polyhedral regions into subsets, each representing one of the control laws $u \in \{-1, 0, +1\}$ reduces the number of hyperplanes to be investigated in the point location of HU_{unc} from 16 to 10 for the horizon $N = 2$ case. Horizon $N = 3$ reduce from 98 to 50 and $N = 4$ from 544 to 250. Figure 5 shows the three sets of unified polyhedral regions representing the different control laws for the $N = 2$ case. It can be observed that the control regions are separated by two decision borders, made up of five hyperplanes each. Investigation of border A will result in a decision between control laws -1 and 0 and border B will distinguish between laws 0 and $+1$. The investigation of the border hyperplanes in solving the point location problem can further be optimized by constructing a binary search as proposed by [10]. Further discussion on binary search trees are omitted since the major concern of this paper is obtaining the border hyperplanes to a partitioned state-space of minimal complexity.

The unified polyhedral regions are made up of convex polyhedral sets but the combined regions them self are non-convex in nature which complicate the process of defining the border hyperplanes. This paper proposes an algorithm with a more direct approach in finding *only* the hyperplanes defining the decision-borders. Other than following the traditional approach of determining Voronoi regions, applying complexity reduction and extracting common facets, we utilize the Delaunay triangulation and unique spatial arrangement of the $U(k)$ sites to extract only the border defining hyperplanes.

The Delaunay triangulation have various structural properties, see [11]. One in specific, being that the Voronoi diagram is the dual graph of the Delaunay triangulation and vice versa [9]. This duality translates into a Delaunay edge (line-segment connecting two sites) being orthogonal to, and bisected by the Voronoi plane shared by the respective sites. It can be observed in figure 5 where the Delaunay triangulation edges are shown in dotted lines. The principle is used in many algorithms for obtaining the Voronoi diagram from its dual. We apply the same principle but only determine the exact border defining hyperplanes, hence eliminating unnecessary computations. We achieve complexity reduction by removing certain edges from the Delaunay triangulation before calculating the respective Voronoi planes. Delaunay edges that connect sites with the same control law values $u(k)$ are removed since their dual (Voronoi plane) would be of no significance in solving the point location problem. Various algorithms exist for determining the Delaunay triangulation of which the *Bowyer-Watson* algorithm is a good option since it is effective in any number of dimensions. Our proposed procedure for the complexity reduction and extraction of the border defining hyperplanes is described in Algorithm 1.

The border hyperplanes obtained from algorithm 1 have been used in the generation of a binary search tree adapted from [10]. The binary search tree was then implemented

Algorithm 1 Border selection algorithm

Step 1 Find the Delaunay triangulation of $HU(k)$-sites.
Step 2 For all Delaunay edges(line segments),
Index edges connecting sites with non-similar control laws $u(k)$, realizing border-spanning edges.
Step 3 For each border-spanning edge,
Assign the site with $u(k) \neq 0$ as the normal vector to a hyperplane,
Assign the mid-point of the edge as a point in the hyperplane,
Define the hyperplane in point-normal format.
Step 4 Index the hyperplanes in border defining sets (Border A and B).

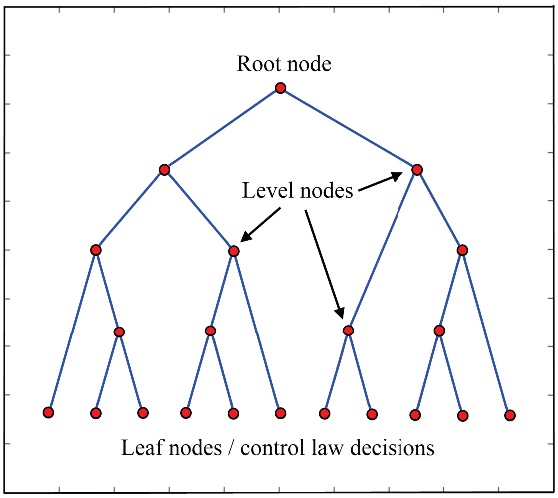

Figure 6: Binary Search Tree structure utilizing 10 border hyperplanes

in a MPC controller for the mentioned inverter. For the horizon $N = 2$ case with 10 border defining hyperplanes a tree structure of 4 levels and 20 nodes as shown in figure 6 was obtained. This translated into a maximum of 4 linear $(A^T x - b)$ calculations that is required during online operation to traverse the tree in finding a solution to the point location problem and hence the control action $u_{opt}(k)$. Simulations utilizing algorithm 1 up to horizon $N = 4$ has produced identical inverter outputs (voltage and current waveforms) as found from simulations using the benchmark exhaustive search method.

6. CONCLUSION

We have presented an algorithm for extracting the border hyperplanes defining the control regions in the partitioned state-space of an MPC controller for a single phase multi-level NPC inverter. The algorithm is simple, efficient and extend into higher dimensions. The simulation results to date are encouraging in terms of system output parameters. Currently more work is being done on reducing the complexity of the partitioned state-space to an even lower level. Ensuring a minimum hyperplane count equates to an optimal binary search tree which will be necessary if we are to be successful in implementing the MPC controller in a practical real-time application.

REFERENCES

[1] J. Rodriguez, M. P. Kazmierkowski, J. R. Espinoza, P. Zanchetta, H. Abu-Rub, H. Young, et al., "State of the art of finite control set model predictive control in power electronics", *Industrial Informatics, IEEE Transactions on*, vol. 9, pp. 1003-1016, 2013.

[2] T. Geyer and D. E. Quevedo, "Multistep finite control set model predictive control for power electronics", *Power Electronics, IEEE Transactions on*, vol. 29, pp. 6836-6846, 2014.

[3] D. Quevedo, R. Aguilera, and T. Geyer, "Predictive Control in Power Electronics and Drives: Basic Concepts, Theory, and Methods", *in Advanced and Intelligent Control in Power Electronics and Drives*, Vol. 531, T. Orowska-Kowalska, F. Blaabjerg, and J. Rodrguez, Eds., ed: Springer International Publishing, pp. 181-226, 2014.

[4] Bemporad, A., Morari, M., Dua, V., and Pistikopoulos, E. N., "The explicit linear quadratic regulator for constrained systems", *Automatica*, Vol. 38, No. 1, pp. 320, 2002.

[5] P. Tondel, T.A. Johansen, and A. Bemporad, "An algorithm for multi-parametric quadratic programming and explicit MPC solutions", *Automatica*, Vol. 39, No. 3, pp. 489497, 2003.

[6] D. E. Quevedo, G. C. Goodwin, and J. A. De Dona, "Finite constraint set receding horizon quadratic control," *International journal of robust and nonlinear control*, vol. 14, pp. 355-377, 2004.

[7] A. Nabae, I. Takahashi, and H. Akagi, *A new neutral-point-clamped PWM inverter*, Industry Applications, IEEE Transactions on, pp. 518-523, 1981.

[8] Y. Wang and S. Boyd, "Fast model predictive control using online optimization", *Control Systems Technology, IEEE Transactions on*, vol. 18, pp. 267-278, 2010.

[9] G. M. Voronoi, "Nouvelles applications des parametres continus a la theorie des formes quadratiques. deuxieme Memoire: Recherches sur les parallelloedres primitifs", *J. Reine Angew. Math.*, 134:198-287, 1908.

[10] P. Tondel, T. A. Johansen, and A. Bemporad, "Evaluation of piecewise affine control via binary search tree", *Automatica*, Vol. 39, pp. 945-950, 2003.

[11] B. Delaunay, "Sur la sphere vide. A la memoire de Georges Voronoi", *Izv. Akad. Nauk SSSR, Otdelenie Matematicheskih i Estestvennyh Nauk*, 7:793-800, 1934.