Approximate Sphere Decoding Based Model Predictive Control of Cascaded H-Bridge Inverters

Rizwan Amir
School of Electrical Engineering & Computer Science
National University of Sciences and Technology
Islamabad, Pakistan
ramir.msee15seecs@seecs.edu.pk

Ammar Hasan
School of Electrical Engineering & Computer Science
National University of Sciences and Technology
Islamabad, Pakistan
ammar.hasan@seecs.edu.pk

Osman Hasan
School of Electrical Engineering & Computer Science
National University of Sciences and Technology
Islamabad, Pakistan
osman.hasan@seecs.edu.pk

***Abstract*—The sphere decoding algorithm (SDA) offers a computationally efficient technique for direct model predictive control in cascaded H-Bridge (CHB) inverters. However, the variable computational complexity offered by the algorithm for each time step makes its real-time implementation quite challenging due to higher values of prediction horizon. We propose to overcome these limitations by using the partial graph processing approximate computing techniques. The proposed algorithm offers constant computational complexity for each time step by retaining K number of best candidates at each layer of the tree. The value of K is then used to decide the reduction in computational complexity and degree of approximation in the identified solution. The simulation results reveal that the proposed approximate SDA provides a sub-optimal solution with a significant improvement in energy and computational efficiency.**

***Index Terms*—approximate computing, sphere decoding, model predictive control, multi-level converters**

I. Introduction

Energy efficiency has become a major concern for computational platforms ranging from portable devices level to large data processing centers. Approximate computing techniques have provided an answer to the problem of energy efficiency by compromising accuracy to gain computational efficiency. Approximate computing techniques [1], based on modifications in algorithms, compiler, and hardware are being applied in many error-resilient applications [2]–[4]. Some applications include image and signal processing, computer vision and scientific computing [5]. In this paper, we propose to use approximate computing, applied at the algorithmic-level to the control of power electronic circuits. The main motivation for this proposal is to lower the value of computational complexity, leading to a decrease in energy consumption by reduction in the digital activity due to arithmetic computations.

Cascaded H-Bridge (CHB) inverters [6] are used to produce a multi-step voltage waveform with controllable frequency, phase, and amplitude by using an array of semiconductor switches and capacitative voltage sources. They facilitate high voltage operation, low harmonics, and low common mode voltages and thus have become an integral component of many electrical systems, including uninterrupted power supplies (UPS) [7], renewable energy conversion systems [8], and traction drives [9]. Particularly, CHB inverters have become an integral component of hybrid cars, where they are used to drive traction motors from fuel cells or batteries [10].

Direct model predictive control (DMPC) has been successfully implemented as a control technique for multi-level converters [11], including CHB inverters. The algorithm performs online optimization of a cost function over a prediction horizon to determine the control input to be applied to the system. The cost function is formulated to control various parameters, like phase current and load voltage. The algorithm takes advantage of the finite number of states of power converter circuits to compute the cost function and thus identifies the solution with the minimum cost. The algorithm offers many advantages, such as inclusion of system constraints, and elimination of modulation stage by direct application of control signal to semiconductor switches [12]. However, the computational complexity, and hence the power consumption, of direct model predictive controller becomes quite high as the prediction horizon increases [13], [14]. This increased complexity and power dissipation is a growing concern as most of the applications of CHB inverters, like hybrid cars, have very strict constraints on battery power.

Various techniques, such as move-blocking [15], extrapolation [15], and sphere decoding [16], have been proposed to reduce the computational complexity of DMPC. Among these, the sphere decoding based DMPC is quite promising as it reduces the computational complexity by considering only those switching sequences that lie inside a sphere of a particular radius. Primarily, the algorithm searches the optimal switching sequences using the branch and bound technique instead of checking all the paths of the search tree. The algorithm reduces the computational complexity while providing the same optimal solution as the enumeration technique [16]. However, a major drawback of the sphere decoding algorithm (SDA) is its varying computational complexity, which can even prevent real-time termination of the algorithm. In safety-critical applications, this can lead to many problems, like system failure or property damage. In this paper, we propose to overcome these issues by exploiting the error-resilience exhibited by SDA through its quality of processing noisy data in an iterative manner.

Particularly, we propose to reduce the computational complexity of SDA based DMPC by the using the principles of approximate computing [17]. In particular, we propose to use the partial graph processing algorithm [18] to compromise the accuracy of the solution to gain computational and energy efficiency by performing a breadth-first search while traversing K search paths on the tree. The traversal of a limited number of search paths and absence of backtracking leads to a constant and low number of tree nodes in each time step. The degree of approximation in the solution and thus the reduction in computational complexity can be controlled by setting the value of K, i.e., a higher value of K leads to a better solution and vice versa. The performance of the proposed approximate SDA is judged using the metrics of total harmonic distortion (THD), number of nodes visited at each tree level and time and resources required to find the solution. These metrics are evaluated using MATLAB based simulations and the resources are acquired through an field-programmable gate array (FPGA) implementation of the proposed algorithm.

The paper is organized as follows: The CHB inverter model is explained in Section II. The DMPC problem is formulated for the CHB inverter in Section III, while the proposed approximate SDA is discussed in Section IV. The simulation results and comparison of performance indices, i.e., total harmonic distortion (THD), number of nodes visited, and FPGA resource utilization is provided in Section V. Finally, Section VI concludes the paper.

II. CHB Inverter Model with Passive Load

The schematic of three-phase five-level H-bridge inverter is shown in Fig. 1, where each HB cell is fed with an isolated DC voltage source V_{dc}. The inverter supplies power to the passive RL load. The state-space model in abc-framework for continuous time can be expressed as [19]:

$$\dot{x}(t) = Ax(t) + Bu(t), \tag{1a}$$

$$y(t) = Cx(t), \tag{1b}$$

where

$$A = \begin{bmatrix} -R/L & 0 \\ 0 & -R/L \end{bmatrix}, \quad B = \frac{V_{dc}}{3L} \begin{bmatrix} 2 & -1 & -1 \\ -1 & 2 & -1 \end{bmatrix},$$

$$\boldsymbol{C} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

The state vector is defined as $x(t) \triangleq [i_a(t)\ i_b(t)]^T$, $u(t) \in \mathbb{V}^3$ represents the control input vector, and $y(t)$ represents the output vector. The output contains the load currents in phases a and b that need to be tracked, also $i_c(t) = -(i_a(t) + i_b(t))$. The above continuous-time equations can be discretized by the forward Euler approximation to obtain

$$x_{k+1} = Ex_k + Fu_k, \tag{2a}$$

$$y_k = Gx_k, \tag{2b}$$

where $E = I + AT_s$, I is the identity matrix, $F = BT_s$, and $G = C$.

The control input is expressed as:

$$u_k = \begin{bmatrix} v_{la}(k) \\ v_{lb}(k) \\ v_{lc}(k) \end{bmatrix}, \tag{3}$$

where

$$v_{ly} \in \mathbb{V} \triangleq \{-\eta, -\eta + 1, ..., \eta - 1, \eta\}. \tag{4}$$

v_{ly} represents phase voltage levels for $y \in \{a, b, c\}$ and η represents the number of HB cells per phase. More details on u_k can be found in [19].

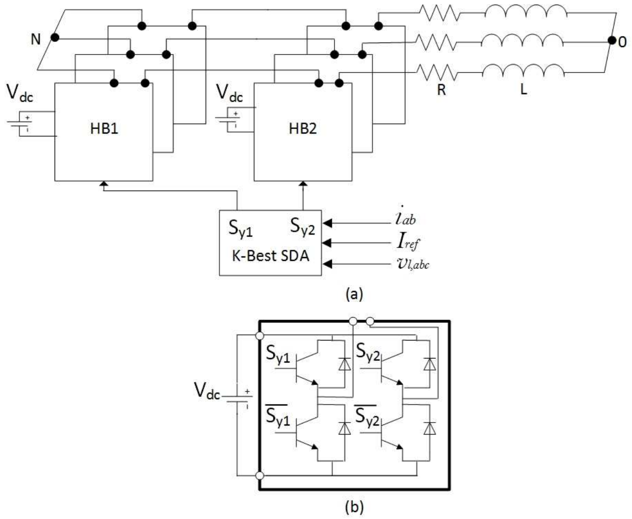


Fig. 1. Schematic of a three phase five-level CHB converter: (a) block diagram of the approximate computing algorithm; (b) single H-Bridge converter.

III. Model Predictive Control

With the state-space model (2), the optimization problem to be solved in direct model predictive control [19] can be stated as follows:

$$\boldsymbol{U}_{opt,k} = \underset{\boldsymbol{U}_k}{\text{argmin}} \sum_{l=k}^{k+N-1} \|i_{e,l+1}\|_2^2 + \lambda_u \|\Delta u_l\|_2^2, \tag{5a}$$

$$\text{subj. to} \quad x_{k+1} = Ex_k + Fu_k, \tag{5b}$$

$$u_k = [v_{la}(k)\ v_{lb}(k))\ v_{lc}(k)]^T, \tag{5c}$$

$$v_{ly} \in \{-\eta, -\eta + 1, ..., \eta - 1, \eta\}, \tag{5d}$$

where $i_e \triangleq i_{ab,ref} - i_{ab}$ represents the current error, $\Delta u_l \triangleq u_{l,ref} - u_l$ represents the error in control input, λ_u is the weighting factor, and $\boldsymbol{U}_k \triangleq \{u_k, u_{k+1}, \dots, u_{k+N-1}\}$ is the sequence of control inputs. The reference current is given by $i_{ab,ref} = [I_{ref}sin(wt) \; I_{ref}sin(wt - 2\pi/3)]^T$ where I_{ref} represents the amplitude of reference current. Also, the controller tracks the control input references to minimize common mode voltage [19]. The desired voltage contained in reference control input is given by

$$v_{ly,ref} = \frac{1}{V_{dc}}(I_{ref}(X_L cos(\omega t + \phi_y) + Rsin(\omega t + \phi_y)). \tag{6}$$

ϕ_y represents the phase angle associated with each phase. A larger value of weighting factor λ_u helps achieve more symmetrical inverter voltages.

The application of SDA requires the above problem to be formulated as an integer least squares problem:

$$\boldsymbol{U}_{opt,k} = \underset{\boldsymbol{U}_k}{\text{argmin}} \quad \|H\boldsymbol{U}_k - \boldsymbol{U}_{unc,k}\|_2^2 \tag{7a}$$

$$\text{subj. to} \quad u_k = [v_{la}(k) \; v_{lb}(k)) \; v_{lc}(k)]^T, \tag{7b}$$

$$v_{ly} \in \{-\eta, -\eta+1, ..., \eta-1, \eta\}, \tag{7c}$$

where $H \in \mathbb{R}^{3N \times 3N}$ denotes a lower triangular matrix (details in [16]), $\boldsymbol{U}_{unc,k} \in \mathbb{R}^{3N}$ denotes the unconstrained solution to the optimal control problem, and $\boldsymbol{U}_k \triangleq [u_k^T \; u_{k+1}^T \dots u_{k+N-1}^T]^T$. Exploiting the triangular structure of matrix H, the partial Euclidean distance (PED) at ith level of tree is calculated as

$$d_i = (H(i, 1:i)U(1:i) - U_{unc}(i))^2 + d_{i-1}. \tag{8}$$

IV. Approximate Sphere Decoding Algorithm

The proposed approximate sphere decoding algorithm (ASDA) uses breadth-first search that traverses the search tree in the forward direction only. Unlike the traditional sphere decoding, no backtracking is involved and the proposed ASDA can work in a parallel, pipelined fashion if required.

The algorithm, given in Algorithm 1, initiates by calculating PED of nodes at the first level of search tree and keeping K number of nodes corresponding to smallest PED. The search path moves in the direction of the path of these K nodes' by extending these nodes into child nodes. PED is calculated for these child nodes, and again K number of nodes with the smallest PED are kept and extended into child nodes. This way the process is repeated until we get to the final layer of the tree. In this layer, the switching sequence corresponding to the smallest PED among the K switching sequences is selected as an optimal sequence. The algorithm is invoked by

$$\boldsymbol{S}_{opt,k} = \text{ASDA}(\boldsymbol{E}_B, \boldsymbol{0}, 0, \boldsymbol{U}_{unc,k}).$$

$\boldsymbol{E}_B$ represents the matrix consisting of K number of column vectors of switching sequences, all column vectors are initialized as empty vectors. $\boldsymbol{0}$ represents a row vector consisting of K number of values of PED of respective switching sequences initialized with the value zero. The algorithm proceeds by filling up the individual column vectors contained in $\boldsymbol{S}_B$ with possible control input values and assigning the corresponding PED to the individual variables contained in $\boldsymbol{d}_B^2$.

TABLE I
System Parameters

Variable	Description	Value
V_{dc}	dc voltage supply per HB (V)	180
I_{ref}	reference current amplitude (A)	5
f	fundamental frequency of load current (Hz)	50
R	load resistor (Ω)	47
L	load inductor (mH)	15
T_s	sampling time (μs)	20
N	horizon length	3
λ_u	weighting factor	10^{-6}

$$\boldsymbol{S}_B = [\boldsymbol{S}_{1,k}, \boldsymbol{S}_{2,k}, \boldsymbol{S}_{3,k}, \dots, \boldsymbol{S}_{K,k}].$$

$$\boldsymbol{d}_B^2 = [\boldsymbol{d}_{1,k}^2, \boldsymbol{d}_{2,k}^2, \boldsymbol{d}_{3,k}^2, \dots, \boldsymbol{d}_{K,k}^2].$$

At the final stage, the switching sequence corresponding to minimum PED denoted by $\boldsymbol{S}_{min,k}$ is assigned to $\boldsymbol{S}_{opt,k}$ as the optimal solution at time step k.

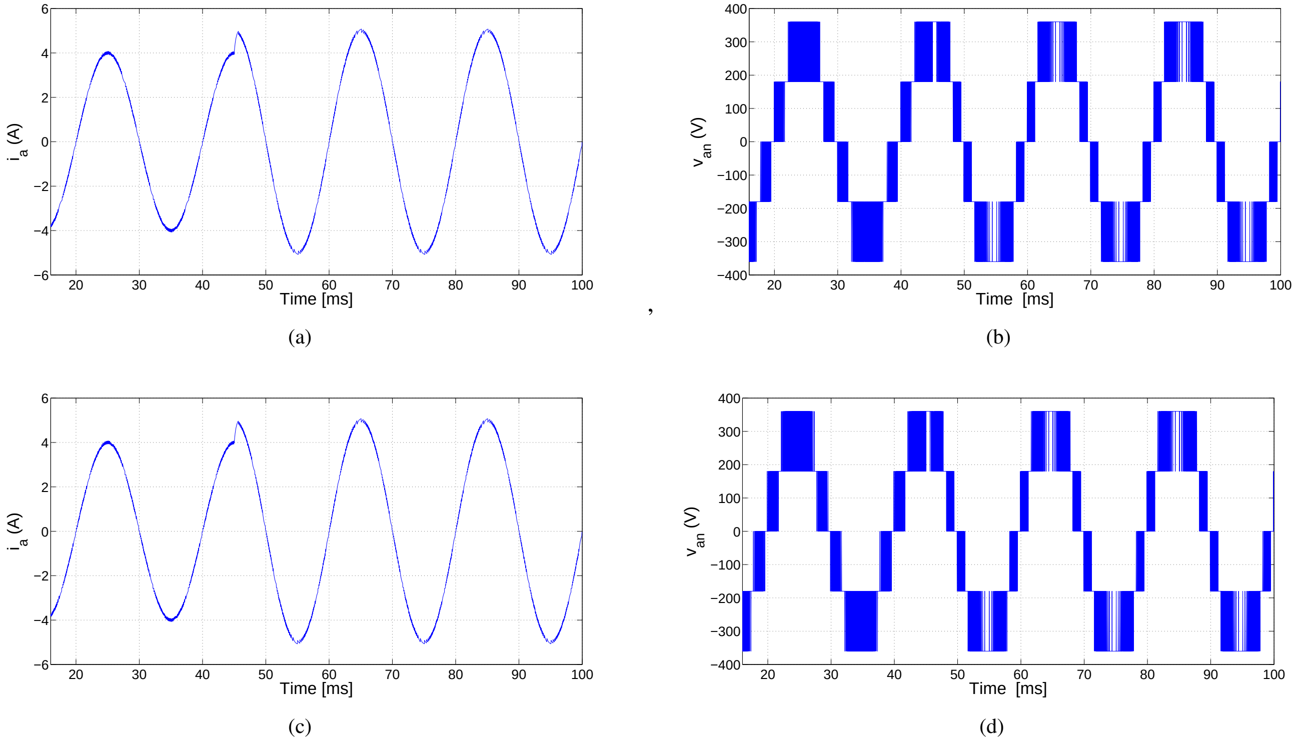
PED is calculated for every single node visited by the algorithm, and thus a higher number of nodes visited translates into a higher number of operations performed in finding the solution. For the proposed algorithm, the number of operations to be performed in finding the solution are constant and can be predicted correctly for each time step even before the algorithm is invoked. In the case of SDA, the computational complexity is variable for each time step, i.e., the number of nodes visited varies for each time step, and thus the number of operations to be performed in each time step cannot be predicted [20].

Algorithm 1 Approximate Sphere Decoding Algorithm

```
function $\boldsymbol{S}_{opt,k} = \text{ASDA}(\boldsymbol{S}_B, \boldsymbol{d}_B^2, i, \boldsymbol{U}_{unc,k})$
    At level $i$ calculate PED $\boldsymbol{d}_k^2$ for each node expanded
    from former level
    Sort the nodes according to their PED and choose K
    nodes with smallest PED $\boldsymbol{d}_{1,k}^2, \boldsymbol{d}_{2,k}^2, ...\boldsymbol{d}_{K,k}^2$
    Store the switching sequence of the chosen K nodes
    in $\boldsymbol{S}_B$ and PED in $\boldsymbol{d}_B^2$
    if $i < N - 1$ then
        ASDA($\boldsymbol{S}_B, \boldsymbol{d}_B^2, i + 1, \boldsymbol{U}_{unc,k}$)
    else
        Find the switching sequence $\boldsymbol{S}_{min}$ corresponding
        to smallest PED among K switching sequences
        $\boldsymbol{S}_{opt,k} = \boldsymbol{S}_{min,k}$
    end if
end function
```

V. Simulation Results

The simulation results for a single phase of three-phase two-cell ($\eta = 2$) CHB inverter are provided in this section for parameters shown in Table I. Fig. 2 shows the phase a load current i_a and inverter voltage v_{an} when SDA and ASDA



(a)

(b)

(c)

(d)

Fig. 2. The simulation results for step change in reference current amplitude using ASDA and SDA (a) load current in phase a, i_a, using ASDA (b) inverter phase a voltage v_{an} using ASDA (c) load current in phase a, i_a, using SDA (d) inverter phase a voltage v_{an} using SDA.

with $K = 1$ is employed. A step change in reference current amplitude is enforced at $45\,ms$ mark from $I_{ref} = 4A$ to $I_{ref} = 5A$. ASDA successfully follows the new load current reference with a settling time of $t_s \approx 0.6ms$. Whereas, in the case of SDA, optimal solution is found at each time step and thus a lower value of current THD is obtained.

The comparison of performance index of current THD for different values of K is shown in Table II. It is observed that for a prediction horizon of three, a value of 1.30% is obtained for SDA based MPC while a value of 1.32% is obtained for the proposed ASDA with K=1. A higher value of current THD for ASDA is mainly because of the approximations performed by the algorithm. However, both values are quite close and a higher value is acceptable in exchange for constant and lower computational complexity. With an increase in the value of K, the difference in value of current THD for both algorithms decreases due to a decrease in the degree of approximation. It can also be noted from Table II that a higher value of prediction horizon results in a decrease in value of the current THD for both algorithms.

TABLE II
Current total harmonic distortion (THD) for SDA and ASDA for different values of prediction horizon (N).

N	% THD for SDA	% THD for ASDA (K=1)	% THD for ASDA (K=2)	% THD for ASDA (K=3)
3	1.30	1.32	1.32	1.31
6	1.29	1.32	1.31	1.30

TABLE III
Number of nodes expanded at each level by SDA and ASDA for a prediction horizon of six (N=6).

Tree level	Maximum number of nodes expanded at the level by SDA	Average number of nodes expanded at the level by SDA	Average and maximum number of nodes expanded at each level by ASDA (K=1)	Average and maximum number of nodes expanded at each level by ASDA (K=2)
1	5	3.1987	1	2
2	22	9.3371	1	2
3	76	39.2141	1	2
4	120	44.7676	1	2
5	168	71.6282	1	2
6	165	77.4398	1	2

Table III shows the comparison for number of nodes expanded at each level by both algorithms for a high value of prediction horizon, i.e., N=6. A large number of nodes expanded at each level results in a large number of search paths being traversed by an algorithm, leading to an increase in computational complexity. For a small value of K, the

TABLE IV
FPGA RESOURCE UTILIZATION BY SPHERE DECODING AND APPROXIMATE SPHERE DECODING ALGORITHMS.

Device Utilization	Total	Used by SDA	% usage for SDA	Used by ASDA	% usage for ASDA
Total Slices	13300	13190	99.2	7029	52.8
Slice Registers	106400	43263	40.7	17408	16.4
Slice LUTs	53200	47916	90.1	22283	41.9
Block RAMs	140	4	2.9	4	2.9
DSP48s	220	7	3.2	13	5.9

TABLE V
COMPARISON OF AVERAGE TIME FOR FINDING THE CONTROL SEQUENCE IN EACH TIME STEP FOR DIFFERENT VALUES OF PREDICTION HORIZON (N).

N	Average time for SDA [ms]	Average time for ASDA (K=1) [ms]	Average time for ASDA (K=2) [ms]
3	3.6	0.58	1.30
4	6.8	0.68	1.70
5	14.7	0.84	2.0
6	24.9	1.40	2.3

difference between the number of nodes expanded by both algorithms indicates a similar difference in computational complexity of both algorithms. ASDA keeps the computational complexity constant by expanding K number of nodes at each tree level. The degree of reduction in computational complexity offered by the ASDA is readily acceptable, in most cases, for the small degree of approximation, shown in Table II, offered by the algorithm. The value of K is selected depending on the values of performance indeces required by the application.

Table IV shows the FPGA resources allocated to both algorithms when implemented on the NI sb-RIO 9607 controller. For the case of proposed ASDA with K=1, it can be observed that the algorithm uses almost 50% of total slices of FPGA as compared to SDA. This difference of FPGA resources also indicates that the proposed approximate version of SDA is also energy-efficient compared to the traditional SDA. With the use of less FPGA resources, less digital activity would be performed, thus leading to greater energy efficiency.

Additionally, the average time required by proposed approximate algorithm for finding the control sequence is far less than the time required for SDA based MPC. Table V shows the average time required to find the solution to the optimal control problem. SDA takes a longer time to find the optimal solution while its approximate version finds sub-optimal solutions in a much shorter time. The difference in solutions found by ASDA and SDA is small as indicated by current THD in Table II. Such a small difference in solution found is usually acceptable.

An increase in the number of H-Bridge cells leads to an increase in number of candidate solutions (5d), thus increasing the number of nodes to be explored. In our case, with $\eta = 2$ and $N = 1$ we have 155 nodes in our tree, and thus an increase in the value of η would increase this number for the same value of N. The use of the proposed approximate SDA to traverse the limited number of search paths on such large search trees could prove to be more profound. This proposed algorithm can also be quite beneficial for the cases when the microcontroller capability is limited or energy efficiency needs to be prioritized over the optimality of the solution.

VI. CONCLUSION

The approximation offered by the proposed ASDA improves energy and computational efficiency by reducing the computation time for DMPC of CHB inverter at the cost of optimality of the solution. Additionally, it offers the benefit of constant computational complexity over SDA. The value of K decides the trade-off between approximation and computational complexity. In case of CHB inverter, the solution provided by the proposed ASDA gives a slightly higher current THD, while significantly reducing the FPGA resources and time required to find the solution. In cases where number of real-time operations are bound by an upper limit or energy efficiency needs to be prioritized, the ASDA offers a solution to keep the computational complexity constant and low while providing a solution that is very close to the optimal one. The proposed technique can be applied to other configurations of inverter circuits, like neutral-point clamped [16] and flying capacitor inverters [21] that allow application of SDA based DMPC. Moreover, this paper provides a very interesting application of approximate computing and, given our promising results, other well-known approximate computing techniques can also be explored in the context of DMPC of CHB inverters.

REFERENCES

[1] T. Moreau, J. S. Miguel, M. Wyse, J. Bornholt, A. Alaghi, L. Ceze, N. E. Jerger, and A. Sampson, "A taxonomy of general purpose approximate computing techniques," *IEEE Embedded Syst. Lett.*, vol. 10, no. 1, pp. 2–5, 2018.

[2] S. Sidiroglou-Douskos, S. Misailovic, H. Hoffmann, and M. Rinard, "Managing performance vs. accuracy trade-offs with loop perforation," in *Proc. of the 19th ACM SIGSOFT Symp. and the 13th Eur. Conf. on Found. of Softw. Eng.*, ser. ESEC/FSE '11, 2011, pp. 124–134.

[3] C. Rubio-Gonzlez, C. Nguyen, H. D. Nguyen, J. Demmel, W. Kahan, K. Sen, D. H. Bailey, C. Iancu, and D. Hough, "Precimonious: Tuning assistant for floating-point precision," in *2013 Int. Conf. for High Performance Comput., Netw., Storage and Anal. (SC)*, Nov 2013, pp. 1–12.

[4] A. Cilardo, "A new speculative addition architecture suitable for two's complement operations," in *2009 Design, Automation Test in Europe Conference Exhibition*, April 2009, pp. 664–669.

[5] Y. K. Chen, J. Chhugani, P. Dubey, C. J. Hughes, D. Kim, S. Kumar, V. W. Lee, A. D. Nguyen, and M. Smelyanskiy, "Convergence of recognition, mining, and synthesis workloads and its implications," *Proc. IEEE*, vol. 96, no. 5, pp. 790–807, May 2008.

[6] J. Rodriguez, L. G. Franquelo, S. Kouro, J. I. Leon, R. C. Portillo, M. a. M. Prats, and M. A. Perez, "Multilevel converters: An enabling technology for high-power applications," *Proc. IEEE*, vol. 97, no. 11, pp. 1786–1817, Nov 2009.

[7] F. Tourkhani, P. Viarouge, and T. A. Meynard, "Optimal design and experimental results of a multilevel inverter for an ups application," in *Proc. of 2nd Int. Conf. on Power Electron. and Drive Syst.*, vol. 1, May 1997, pp. 340–343.

[8] F. Filho, Y. Cao, and L. M. Tolbert, “11-level cascaded h-bridge grid-tied inverter interface with solar panels,” in *25th Annu. IEEE Appl. Power Electron. Conf. and Expo. (APEC)*, Feb 2010, pp. 968–972.

[9] S. Bernet, “Recent developments of high power converters for industry and traction applications,” *IEEE Trans. Power Electron.*, vol. 15, no. 6, pp. 1102–1117, Nov 2000.

[10] Z. Du, L. M. Tolbert, J. N. Chiasson, B. Ozpineci, H. Li, and A. Q. Huang, “Hybrid cascaded h-bridges multilevel motor drive control for electric vehicles,” in *37th IEEE Power Electron. Specialists Conf.*, June 2006, pp. 1–6.

[11] S. Vazquez, J. I. Leon, L. G. Franquelo, J. Rodriguez, H. A. Young, A. Marquez, and P. Zanchetta, “Model predictive control: A review of its applications in power electronics,” *IEEE Ind. Electron. Mag.*, vol. 8, no. 1, pp. 16–31, March 2014.

[12] C. Bordons and C. Montero, “Basic principles of MPC for power converters: Bridging the gap between theory and practice,” *IEEE Ind. Electron. Mag.*, vol. 9, no. 3, pp. 31–43, Sept 2015.

[13] D. E. Quevedo, G. C. Goodwin, and J. A. De Doná, “Finite constraint set receding horizon quadratic control,” *Int. J. Robust Nonlin. Control*, vol. 14, no. 4, pp. 355–377, Jan 2004.

[14] J. Rodriguez, M. P. Kazmierkowski, J. R. Espinoza, P. Zanchetta, H. Abu-Rub, H. A. Young, and C. A. Rojas, “State of the art of finite control set model predictive control in power electronics,” *IEEE Trans. Ind. Informat.*, vol. 9, no. 2, pp. 1003–1016, May 2013.

[15] P. Karamanakos, T. Geyer, N. Oikonomou, F. D. Kieferndorf, and S. Manias, “Direct model predictive control: A review of strategies that achieve long prediction intervals for power electronics,” *IEEE Ind. Electron. Mag.*, vol. 8, no. 1, pp. 32–43, March 2014.

[16] T. Geyer and D. E. Quevedo, “Multistep finite control set model predictive control for power electronics,” *IEEE Trans. Power Electron.*, vol. 29, no. 12, pp. 6836–6846, Dec 2014.

[17] J. Han and M. Orshansky, “Approximate computing: An emerging paradigm for energy-efficient design,” in *2013 IEEE Eur. Test Symp. (ETS)*, May 2013, pp. 1–6.

[18] S. Singh and R. Nasre, “Scalable and performant graph processing on GPUs using approximate computing,” *IEEE Trans. Multi-Scale Comput. Syst.*, pp. 1–1, 2018.

[19] R. Baidya, R. P. Aguilera, P. Acua, S. Vazquez, and H. d. T. Mouton, “Multistep model predictive control for cascaded H-Bridge inverters: Formulation and analysis,” *IEEE Trans. Power Electron.*, vol. 33, no. 1, pp. 876–886, Jan 2018.

[20] B. Hassibi and H. Vikalo, “On the sphere-decoding algorithm I. expected complexity,” *IEEE Trans. Signal Process.*, vol. 53, no. 8, pp. 2806–2818, 2005.

[21] Z. Zhang, J. Schenek, and R. Kennel, “Direct model predictive control of five-level dual flying capacitor active neutral point clamped converters with modified sphere decoding,” in *IEEE Transp. Electrific. Conf. and Expo. (ITEC)*, June 2017, pp. 350–355.