IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received January 2, 2021, accepted January 11, 2021, date of publication January 18, 2021, date of current version January 26, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3052129

Model Predictive Torque Control for Multilevel Inverter fed Induction Machines Using Sorting Networks

KRISTÓF BÁNDY AND PETER STUMPF
Department of Automation and Applied Informatics, Budapest University of Technology and Economics, 1111 Budapest, Hungary
Corresponding author: Peter Stumpf (stumpf@aut.bme.hu)

This research work carried out at the Budapest University of Technology and Economics was supported in part by the National Research, Development and Innovation Office (NKFIH) under Grant FK 124913, and in part by the National Research Development and Innovation Fund based on the charter of bolster issued by the National Research and Innovation Office under the auspices of the Ministry for Innovation and Technology.

ABSTRACT Model Predictive Control is a promising technique for electric drive control, as it enables optimization for multiple parameters and offers reliable operation with non-linear systems. For induction machine drives it can be realized using separate cost functions for the torque and the stator flux. Although this eliminates the problem of calculating any weighting factor, the selection of the final voltage vector requires an additional sorting algorithm. By increasing the number of voltage levels or the prediction horizon, the sorting algorithm becomes more and more time-intensive, which can severely impair the performance of the control algorithm. This paper introduces a novel hybrid sorting algorithm consisting of two sorting networks and a merging step. As a case study, the described control method is applied for an induction machine with high rated frequency fed by a three-level inverter, while also discussing the implementation issues. Experimental results verify the operation of the devised sorting algorithm.

INDEX TERMS Induction motors, predictive control, power electronics, sorting, electric machines.

I. INTRODUCTION

Nowadays, thanks to the dramatic increase in computational power of digital devices, Model Predictive Control (MPC) has become a very promising strategy for the control of electrical drives [1], [2]. The MPC scheme has a different concept of design, contrary to Field Oriented Control (FOC) or Direct Torque Control (DTC), which are widely accepted as standard methods for high-performance electric drives. In MPC the control actions are evaluated via cost function(s), where system constraints can be included. MPC controlled drives have fast dynamic performance and good torque response. Papers [3] and [4] offer a detailed comparison between the performance of MPC and FOC or DTC.

MPC schemes for power electronics converters and drives can be classified according to several aspects. One aspect is, whether it uses a dedicated modulator or not. If a modulator is used, then the MPC algorithm is an indirect controller, which calculates the modulation signals or the duty ratios, that are fed to the modulator generating the switching signals. If no separate modulator is used, the scheme is direct, and the controller outputs the switching signals directly.

In the literature direct MPC with reference tracking, or in other words, Finite Set MPC (FS-MPC), is the most commonly used method, thanks to its well-known advantages. These include simple inclusion of nonlinearities and constraints, multivariable control using a single control loop, intuitive design procedure and a straightforward way to implementation. However, there are some disadvantages of this method, like the variable switching frequency. If proper design guidelines are not followed, it can have an inferior performance compared to conventional methods, like FOC or DTC [1].

Some direct MPC schemes do not only determine the output switching signals/voltage vector, but they calculate their duration as well. It results in a behavior similar to Pulse Width Modulation (PWM) techniques. Paper [5] presents a method for induction machines, where the torque ripple is minimized by calculating the time instant at which the switches of the inverter should change state. An MPC scheme with a modulation algorithm, which calculates the duty cycles of the voltage vectors, to obtain fix switching frequency is

The associate editor coordinating the review of this manuscript and approving it for publication was Zhilei Yao.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

presented in [6] for a three-level inverter. Paper [7] proposes an MPC scheme with a modulation stage for a six-phase induction machine drive with fixed switching frequency.

As for indirect MPC schemes, which use a modulator, there are methods, which apply a programmed PWM, where the switching-instant and the pattern are computed offline based on some optimization criteria [8], [9]. These methods can achieve very low harmonic distortion in the current, but the closed loop implementation poses many challenges. Some other indirect MPC schemes use carrier based PWM techniques, like Space Vector Modulation [10]. Paper [11] combines synchronized space vector PWM schemes with MPC. There a control strategy is developed that does not require the phase compensation of the voltage reference and a smooth transition between different PWM modes is achieved.

MPC schemes for electric drive control can be also further organized into two categories according to their realization. One category is occupied by the model predictive torque control (MPTC), which controls the torque and the stator flux of the machines similarly to DTC. Paper [12] proposed a sensorless MPTC method, where a Model reference adaptive system (MRAS), based on a sliding mode stator voltage model observer, estimates the rotor speed. Another MPTC scheme using a single cost function with weighting factors and duty cycle control to optimize the switching-instant between the nonzero and zero voltage vector is introduced in [13]. While MPC schemes in the other category, called model predictive current control (MPCC), control the real and imaginary component of stator current, which bears a resemblance to FOC. MPCC algorithms are widely applied for Permanent Magnet Synchronous Machines (PMSM) [14].

MPC schemes can be also classified into two categories based on whether a single cost function with weighting factors or multiple cost functions without weighting factors are being used. In the first case, the value of the weighting factors have a great impact on the performance of the drive and their tuning is a nontrivial process, which is generally based on heuristic trial and error procedures. Many papers in the literature deal with the proper tuning of weighting factors. Paper [15] presents an algebraic method to compute weighting factors to minimize the current distortions. A technique to optimize the value of weighting factors in real-time to reduce the torque ripple is introduced in [16]. The clear advantage of methods using cost functions for each control objective is that the previously mentioned problems and difficulties related to the calculation of weighting factors are solved. Furthermore, it can result in an equal compromise of tracking for each control objective at the same time. Paper [17] introduces the concept of an MPC scheme for two-level inverter fed induction machine drive using separate cost functions for the torque and the stator flux. A simplified version of the method is introduced in [18]. In either strategy, selecting the most proper voltage vector requires an additional sorting of the possible switching actions.

The main contribution of the paper is the application of sorting networks to find the most appropriate voltage vector when multiple cost functions are utilized. In the literature most of the papers focus on the calculation of the cost functions and the applied sorting algorithm is not analyzed deeply. In MPC schemes using a single cost function with weighting factors, the implementation to select the voltage vector is straightforward and simple. However, in MPC schemes relying on multiple cost functions, the selection of the final voltage vector requires an additional sorting algorithm. By increasing the number of voltage levels, the sorting algorithm becomes more and more complicated. Paper [17] uses the method quicksort for selecting the output voltage vector in an MPTC scheme using separate cost functions for torque and stator flux for a two-level inverter fed induction machine. A simplified sorting algorithm, called sequential MPC (SMPC), can be found in [18]. In that paper also a two-level inverter fed induction machine is studied. The SMPC algorithm selects the two voltage vectors out of seven with the smallest value of cost function for the torque. Then, the flux cost function is evaluated only for the selected two voltage vectors to determine the final choice for the output voltage vector. Methods of data sorting in hardware using parallel recursive algorithms over a binary tree are introduced in [19].

The paper reviews a few possible solutions for sorting and ranking voltage vectors and benchmark them on a three-level inverter. As it will be demonstrated using a hybrid algorithm of two sorting networks combined with a merging step to find the most appropriate voltage vector can reduce the computational time compared to other widely applied methods for three-level inverters. As a case study, to illustrate the performance of sorting networks, a direct MPTC scheme using separate cost functions for the flux and the torque is implemented for a three-level inverter fed induction machine with high rated frequency.

Nowadays increasing attention has been paid to high speed/high-pole drives with high rated fundamental frequency (from a few hundred up to a thousand Hz). The digital control of these machines poses many challengesdue to the limited ratio between the sampling and switching frequencies [20]. Furthermore, the inductance of high-frequency machines is designed to be small compared to ordinary motors. It can result in a much higher distortion in the current signal. Based on our experience the MPTC scheme can be a feasible solution for controlling machines with high rated frequency. Therefore in the paper the experimental tests were carried using an induction machine with high rated frequency. However, the experiments were performed at lower frequencies as the main focus of the paper is the sorting algorithms.

The following section of the paper will present the mathematical model for the induction machine and the prediction equations. Section III contains the detailed MPTC control strategy. It is followed by an overview about sorting and sorting networks in Section IV. Section V addresses the hardware implementation and experimental results. The conclusions can be found at the end of the paper.

II. MATHEMATICAL BACKGROUND

MPTC method has four main steps: estimation of the non-measurable state variables, prediction for the selected variables, evaluation of the cost function(s), and sorting of the voltage vectors to produce the control signal. For the first step, it is essential to have a proper mathematical model of the drive system. As MPC methods are inherently implemented in the digital domain, the model of the drive in the continuous-time domain has to be discretized.

A. DYNAMIC MODEL OF IM

The operation of a squirrel cage induction machine in a rotating reference frame (RRF), which rotates with an arbitrarily selected ω_R angular speed, can be described by the following two independent differential equations expressing the stator and rotor voltage balance

$$\mathbf{v}_s = R_s\mathbf{i}_s + \frac{d\Psi_s}{dt} + j\omega_R\Psi_s \tag{1}$$

$$\mathbf{v}_r = \mathbf{0} = R_r\mathbf{i}_r + \frac{d\Psi_r}{dt} - j(\omega_R - \omega)\Psi_r \tag{2}$$

and by the stator Ψ_s and rotor Ψ_r flux relations

$$\Psi_s = L_s\mathbf{i}_s + L_m\mathbf{i}_r \tag{3}$$

$$\Psi_r = L_m\mathbf{i}_s + L_r\mathbf{i}_r, \tag{4}$$

where ω is the rotor electrical angular velocity. R_s and R_r are the stator and rotor phase resistance, respectively. The total inductance of the stator and rotor can be given as $L_s = L_m + L_{ls}$ and $L_r = L_m + L_{lr}$, where L_{ls} and L_{lr} denote the leakage inductance of the stator and rotor. The electric torque of the machine can be given as

$$M = \frac{3}{2}P\Psi_s \times \mathbf{i}_s, \tag{5}$$

where P is the number of pole-pairs.

B. DISCRETIZATION

By using the forward Euler approximation for the derivatives for a sampling time denoted by T_s, the following equation for the prediction of stator flux vector in the stationary reference frame (SRF) ($\omega_R = 0$) can be obtained from (1)

$$\Psi_s^p(kT_s + T_s) = \hat{\Psi}_s(kT_s) + \mathbf{v}_s(kT_s)T_s - R_s\mathbf{i}_s(kT_s), \tag{6}$$

where superscript p denotes predictions and $\hat{}$ represents estimations, as flux values are not measured directly due to complexities.

The stator current vector in the Stationary Reference Frame (SRF) can be predicted as

$$\mathbf{i}_s^p(kT_s + T_s) = \left(1 - \frac{R_eT_s}{\sigma L_s}\right)\mathbf{i}_s(kT_s) + \frac{T_s}{\sigma L_s}\mathbf{v}_s(kT_s) + -j\frac{L_m}{\sigma L_sL_r}T_s\omega\hat{\Psi}_r(kT_s) + \frac{L_mR_r}{\sigma L_r^2L_s}T_s\hat{\Psi}_r(kT_s), \tag{7}$$

where $R_e = R_s + \frac{L_m^2R_r}{L_r^2}$ and $\sigma = 1 - \frac{L_m^2}{L_rL_s}$. For the prediction of the stator current vector the estimated value of the rotor flux vector is necessary, which can be calculated as

$$\hat{\Psi}_r(kT_s) = \frac{L_r}{L_m}\left(\hat{\Psi}_s(kT_s)) - \sigma L_s\mathbf{i}_s(kT_s)\right). \tag{8}$$

Finally, the electromagnetic torque can be predicted as

$$\begin{aligned} M_e^p(k+1) &= \frac{3}{2}P\Psi_s^p(kT_s + T_s) \times \mathbf{i}_s^p(kT_s + T_s) \\ &= \frac{3}{2}P(\Psi_{s\alpha}^p(kT_s + T_s)i_{s\beta}^p(kT_s + T_s) \\ &\quad - \Psi_{s\beta}^p(kT_s + T_s)i_{s\alpha}^p(kT_s + T_s)), \end{aligned} \tag{9}$$

where α and β denote the real and imaginary components of the vectors in the SRF.

The applied discretization technique can have a great effect on performance. Therefore, the selection of the approximation method plays a crucial role. Paper [21] demonstrates that, by discretizing the equations using an improved Taylor method results in better performances compared to the Euler method. The discretization of induction machine equations by using the Tustin method is introduced in [22].

C. STATOR FLUX PREDICTION FROM ROTOR FLUX

In the traditional MPTC scheme the stator flux is predicted/calculated by using (6) and the rotor flux is obtained from the stator flux as (8). In the paper, it is suggested to use the so-called current model, where the rotor flux is calculated based on the stator current. Then the stator flux is calculated from the rotor flux. A similar method is applied in [18] using the backward Euler method. This method has the advantage over the previous one, that it applies a closed-loop integrator. It requires the mechanical angle for coordinate transformation, which can be obtained by integration.

By selecting the angular speed of the RRF to be the mechanical angular speed $\omega_R = \omega$, (1)-(4) can be simplified to

$$\frac{d\Psi_r}{dt} = \frac{R_rL_m}{L_r}\mathbf{i}_s - \frac{R_r}{L_r}\Psi_r \tag{10}$$

As the value of the rotor flux is crucial to have an accurate and stable response, in the current paper the so-called trapezoidal (Tustin) integral approximation is utilized. The discrete version of the rotor flux estimator using Tustin approximation is derived by one of the author in [22] and [23] as

$$\hat{\Psi}_r(kT_s) = K_1\hat{\Psi}_r((kT_s - T_s)) + K_2(\mathbf{i}_s(kT_s) + \mathbf{i}_s(kT_s - T_s)) \tag{11}$$

where

$$K_1 = \frac{1 - \frac{R_rT_s}{2L_r}}{1 + \frac{R_rT_s}{2L_r}} \quad \text{and } K_2 = \frac{\frac{R_rL_mT_s}{2L_r}}{1 + \frac{R_rT_s}{2L_r}} \tag{12}$$

After estimating $\hat{\Psi}_r(kT_s)$ in the RRF, it is transformed back to the SRF.

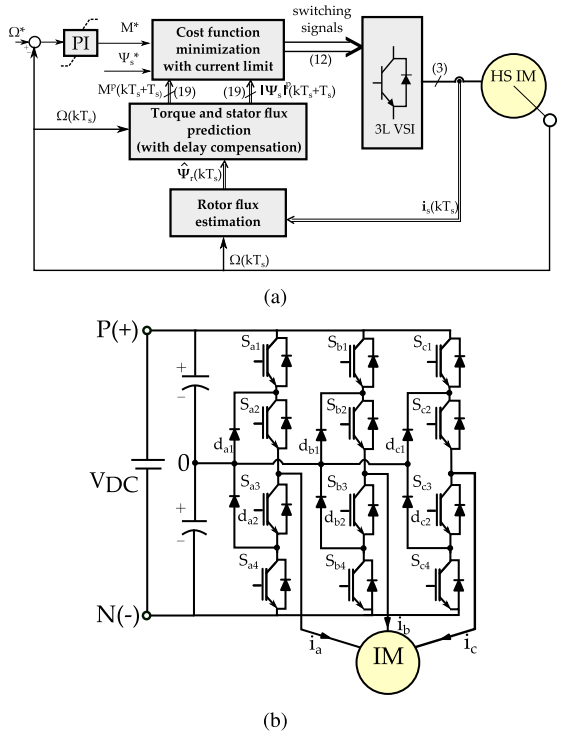


FIGURE 1. Control diagram of direct MTPC scheme (a) and three-level NPC topology (b).

By using (8) the stator flux can be predicted for all the possible voltage vectors as

$$\hat{\Psi}_s^p(kT_s + T_s) = \frac{L_m}{L_r}\hat{\Psi}_r(kT_s) + \sigma L_s \mathbf{i}_s(kT_s) \tag{13}$$

Furthermore, the value of $\hat{\Psi}_r(kT_s)$ calculated from (11) can be used for the stator current prediction as well.

The devised MPTC scheme relies heavily on the predictor equations of the stator current (7), rotor flux (11) and stator flux (13). Therefore, we provide these equations with the exact numerical values of the actual system in the Appendix. The detailed system parameters can be found later in Table 3 and $T_s = 50\mu s$.

III. MODEL PREDICTIVE TORQUE CONTROL

The block diagram of the studied direct MPTC scheme using a three-level inverter is shown in Fig.1, where an external PI speed controller generates the M^* reference value of the electric torque. In the paper, experimental results using an NPC type three-level inverter will be presented. The topology can be seen in Fig.1(b). As it can be seen, compared to a two-level converter NPC-type inverter has two extra semiconductors per phase and two clamping diodes. These clamping diodes allow the connection of the phase output to the converter neutral point, which enables the three-level characteristic of the topology.

A. SCHEME WITH WEIGHTING FACTOR

The traditional scheme predicts both the electromagnetic torque and the stator flux for all possible voltage vectors by using (9) and (13). In the case of a three-level inverter, the number of different possible voltage vectors is 19 for one time-step forward. One method to select the most appropriate voltage vector is to minimize a cost function, which can be expressed as a linear combination of torque and stator flux errors:

$$g = |M^* - M^p(kT_s + T_s)| + k_\Psi |\Psi_s^* - |\Psi_s^p|(kT_s + T_s)| \tag{14}$$

where k_Ψ is the so-called weighting factor.

As it was mentioned previously, the value of the weighting factor has a great impact on the performance of the drive which this paper aims to circumvent by utilizing separate cost functions.

B. SCHEME WITHOUT WEIGHTING FACTOR

In the current paper, the torque and flux errors are evaluated separately by using two cost functions [17], [18], [23]

$$g_1 = |M^* - M^p(kT_s + T_s)|^2 \tag{15}$$
$$g_2 = |\Psi_s^* - |\Psi_s^p|(kT_s + T_s)|^2 \tag{16}$$

The cost functions g_1 and g_2 are evaluated for each possible voltage vector. Then the voltage vectors are sorted and ranked based on the value of error: voltage vectors with a lower error are assigned a lower ranking. r_1 and r_2 denote the ranking values assigned to g_1 and g_2, respectively. The ranking value expresses a relative quality of the voltage vector compared to the other possible voltage vectors. Finally, the voltage vector with the minimum average value of its rankings is selected as

$$\mathbf{v}_{opt}(kT_s) = \arg \min_{\mathbf{v}_1 \ldots \mathbf{v}_{19}} \frac{r_1 + r_2}{2}. \tag{17}$$

It results in an equal compromise of tracking for torque and flux at the same time.

By using the presented ranking approach, multiple voltage vectors may have the same averaged ranking [17]. To solve this issue, priorities can be assigned for each objective but only for the condition of multiple optimal voltage vectors. In this paper, for multiple optimal voltage vectors, the vector which minimizes the torque error is selected.

C. CONSIDERATION OF CALCULATION TIME

One well-known disadvantage of real-time digital implementation of MPC techniques is the required high processing capability due to the the extensive number of calculations. It results in a delay between the measurements and the actuation, which can deteriorate the performance of the drive if it is not considered and not compensated. The principle of delay compensation is that the control variables should be predicted for the future instant $T_s k + 2T_s$. This delay compensation strategy is well documented in [24].

D. REDUNDANT VOLTAGE VECTORS

For a three-level inverter (see Fig.1(b)) altogether $3^3 = 27$ switching states can be generated, which produce 19 different

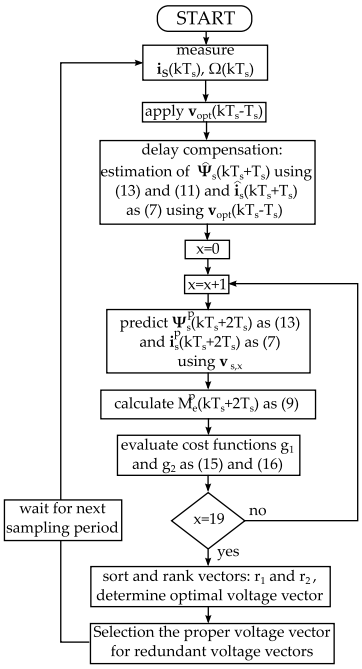


FIGURE 2. Flow diagram of the MPTC technique.

voltage vectors. It should be noted that some switching states are redundant, generating the same voltage vector. Redundant voltage vectors are the zero vectors and the so-called small voltage vectors having an amplitude of $\frac{1}{3}V_{DC}$, where V_{DC} is the total DC link voltage.

After the cost functions are evaluated for each of the voltage vectors, the sorting and ranking algorithm selects the optimal voltage vector. If a redundant voltage vector has been selected, there are two possibilities. One possibility is to choose the vector that requires less switching. This method can be applied always for zero voltage vectors. It can be also used for small voltage vectors for three-level Cascaded H-bridge (CHB) inverters or by Neutral Point Clamped (NPC) Inverter or T-type inverters having a fixed neutral point potential. The other option is using the redundant small vectors to balance the capacitor voltages. If the neutral point is not fixed, for example, NPC or T-type inverters, this other possibility should be used. This strategy is well documented in [25], [26].

E. FLOW DIAGRAM OF MPTC SCHEME

Figure 2 presents the flow diagram of the studied MPTC technique by giving the exact equations as well.

IV. SORTING AND RANKING ALGORITHM

The MPC scheme proposed in this paper utilizes two cost functions. The objective functions g_1 and g_2 are evaluated for each possible voltage vector. Then, the voltage vectors are sorted and ranked based on the value of error. Practical experiences indicate that the sorting (and ranking) algorithm can take a much longer time than the calculation of predictions and the evaluation of the cost functions. For a three-level inverter, there are $n = 19$ different voltage vectors that need to be ranked in accordance with their cost function values. Naturally, the method presented hereby can be applied in essence for smaller and greater n values as well.

It should be noted, that the values of the cost functions do not need to be sorted, because the goal is to sort the indexes that point to the given voltage vector. This is favorable to computation, as most likely float type variables are used for the cost function calculations, but the indexes are integer type variables.

A. QUICKSORT

The most straightforward sorting algorithm that can be utilized for smaller datasets is the so-called quicksort. It is a divide-and-conquer algorithm, which selects a pivot element and partitions the other elements into two sections depending on whether they are smaller or not than the pivot element. Then the partitions are sorted recursively. The operation principle of the algorithm is demonstrated in Fig.3(a) using an array with $n = 7$ elements. Here the pivot is the last element. To this day quicksort algorithm is being analyzed and further optimized and researches mainly focus on the selection of the pivot element(s). The best-case scenario for the algorithm is when the selected pivot elements are always the actual median value, but the algorithm has an $O(n^2)$ worst-case time complexity and it can be carried out only after all the predictions are complete. A pseudocode for the implementation of quicksort can be found in [27]. Paper [17] applies the quicksort method to select the voltage vector using MPC scheme for two-level inverter fed induction machine drive using separate const function for the torque and the stator flux.

B. INSERTION SORT

A higher execution speed can be realized by using insertion sort instead of quicksort. Insertion sort can be implemented into the loop that calculates the cost function values and it does not contain recursion. At each iteration step, the corresponding new cost function value is compared to the largest value in the sorted list. If its value is larger, then it keeps the element in place and moves to the next element in the sorted list. If its value is smaller, it finds its proper position within the sorted list. After each of the larger values are shifted to make space to insert this value into this correct position. As it was mentioned previously, it is enough to sort the indexes instead of the actual cost function values. The operation principle of the algorithm is demonstrated in Fig.3(b) using an array with $n = 7$ elements. One the figure the array is already specified. The insertion sort has an order of $O(n^2)$ time complexity like quicksort but has the advantage of being able to run simultaneously with the calculation of the cost functions.

A pseudocode for the implementation of insertion sort can be found in [28].

The best-case scenario for the insertion sort is an already sorted array, in which case only $n - 1$ comparisons and no swaps are made. However, the worst-case scenario is a

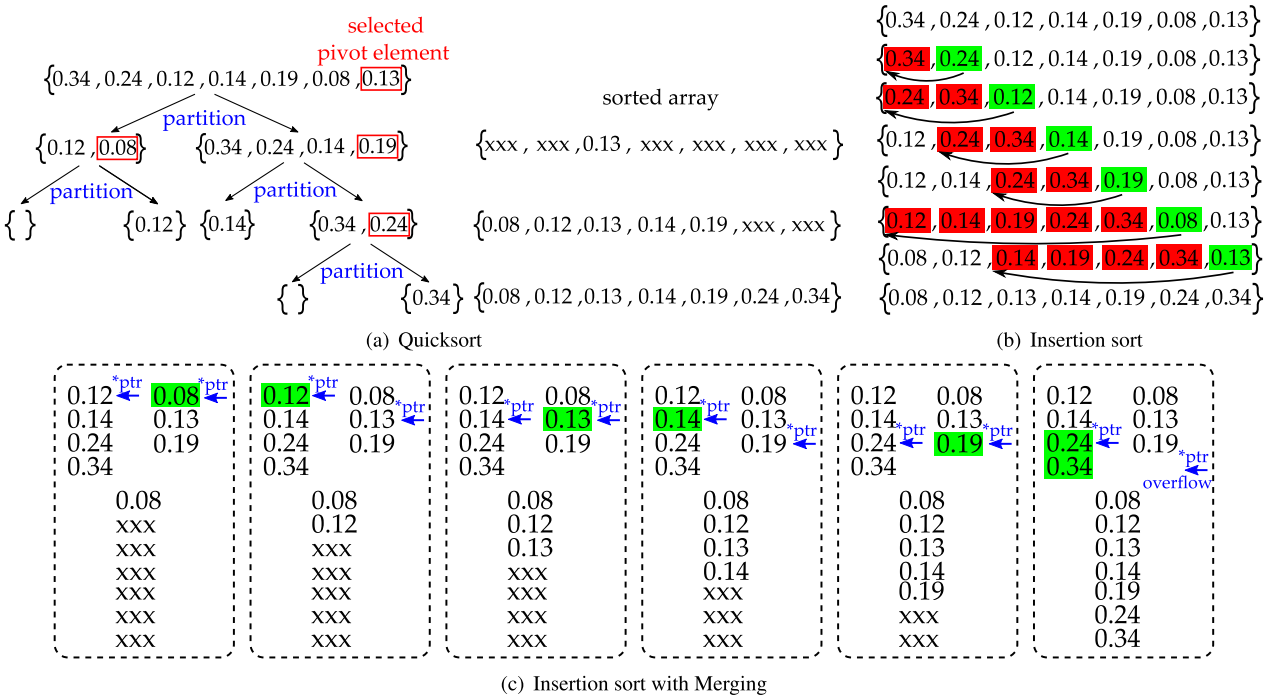


FIGURE 3. Sorting algorithms: Quicksort (a), Insertion sort (b) and Insertion sort with merging (c).

reversely sorted array where insertion sort needs to make $n(n-1)/2$ comparisons and the same number of swaps. If only insertion sort is used, then the worst-case scenario in our case would result in 171 comparisons and swaps. This shows that the execution time for the insertion sorting of $n = 19$ values would vary heavily and in the worst-case could take up to four times the execution time needed to actually obtain those values. The insertion sort algorithm can be a favorable solution for two-level inverters, where $n = 7$ resulting in a much lower number of comparisons and swaps.

It should be noted that a single insertion sort method requires a second loop, which gives the rank values to the voltage vectors.

Although, insertion sort has an $O(n^2)$ time complexity, for smaller datasets it is still a favorable solution, like for example for two-level inverters, where $n = 7$. Furthermore, the insertion sort method has the best performance compared to other algorithms having the same time complexity, like selection sort or bubble sort for smaller datasets. Paper [29] presents a pseudocode for implementation of insertion sort for an FPGA for three-level NPC inverter, where $n = 27$ as the cost function are evaluated for all the possible voltage vectors, even for the redundant ones.

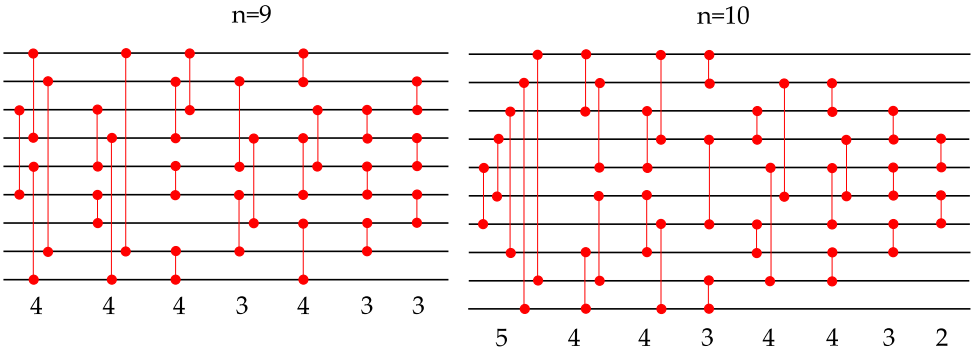
C. INSERTION SORT WITH MERGING

The performance of the insertion sort can be improved by reducing the great variance of the comparisons for example by merging. For example, the voltage vectors required to be sorted can be split into two subgroups, which are sorted independently. For example, for $n = 19$, after sorting the first 10 indexes the remaining 9 values were sorted with another insertion sort. Then, the two separately sorted arrays were sorted with a merging process. The merging step would create two pointers to the first elements of the first element of each sorted arrays, and then compare the elements at those pointers. The smaller valued item is placed into a new array and its array's pointer is shifted. This is being done until one of the pointers would overflow at which point all the items in the other array are placed into the new array in their respective order. The operation principle of the algorithm is demonstrated in Fig.3(c) using an array with $n = 7$ elements, which was split into two groups having 4 and 3 elements, respectively.

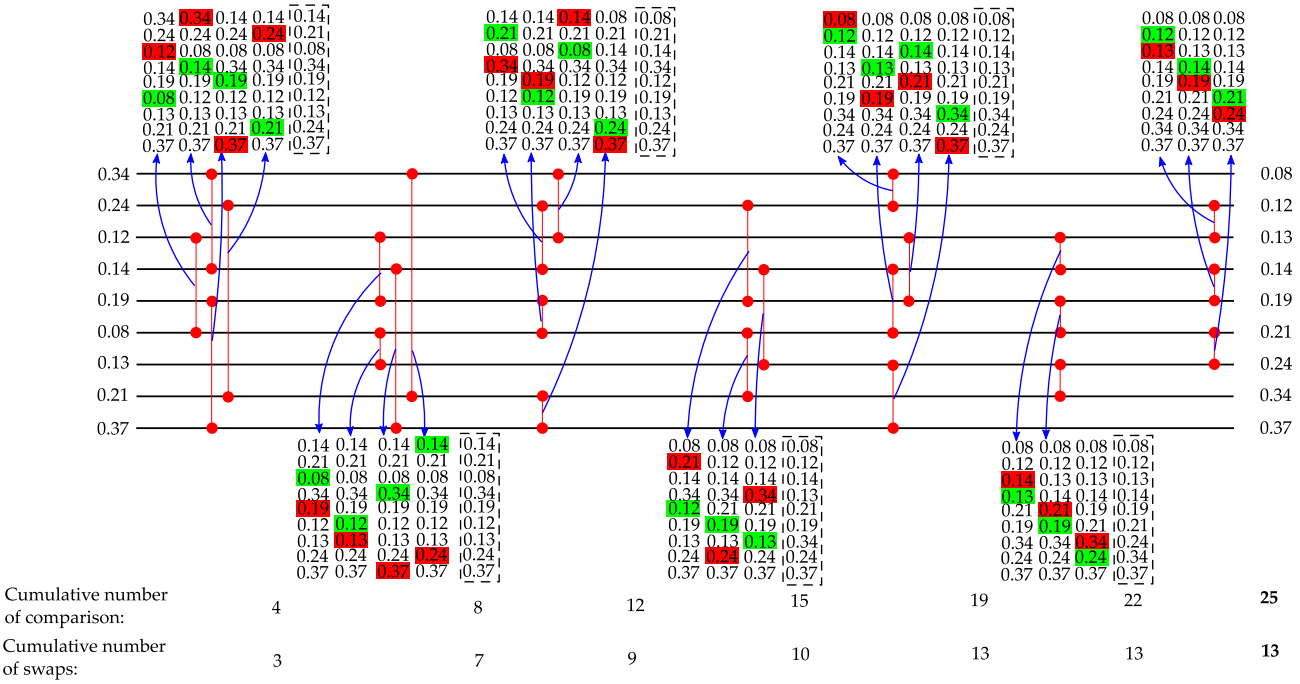
If the lengths of the arrays that need to be sorted are denoted with k and m ($k + m = n$), then the best-case would need the minimum of k and m number of comparisons compared to the $k + m - 1$ of the worst-case. Although merging needs a new array in which it places the sorted indexes, it can be also used to directly assign rank values to the vectors in our case.

This hybrid solution of insertion and merge sorts would yield for $n = 19$ altogether $(10-1)+(9-1)+9 = 26$ comparisons in the best-case while $10(10-1)/2+9(9-1)/2+10+9-1 = 99$ comparisons are required at the worst-case. As it can be noticed, this method effectively halves the required execution time compared to the case, when only the insertion sort is being used.

The length of insertion sorted arrays can be further reduced, but it requires more merge steps. This would further reduce the number of comparisons but would introduce more loops that could not be incorporated with each other and each merging would need a new array to store the results.



(a) Sorting network comparator diagrams for 9 and 10 inputs



(b) Example for using of sorting network with $n = 9$ elements

FIGURE 4. **Sorting networks.**

For $n = 19$ case, four subgroups (having three with 5 elements and one with 4 elements) can be created. After sorting the four subgroups with insertion, three merge sort steps are required to obtain the final result. Although fewer comparisons are required, due to the fact, that, the merging needs to check that if any of the pointers are overflowed we observed an increase in the computational time.

D. SORTING NETWORK

The required comparisons of the hybrid solution still varied greatly because of the insertion sorts. Therefore, another alternative was examined to replace the insertion sorts with sorting networks. Sorting networks are state-of-the-art solutions developed for parallel processing of the input array. Sorting networks require a fixed amount of comparisons to sort the input array of values. For $n = 19$ having two subgroups containing 9 and 10 elements, the number of comparisons required is 25 and 29, respectively [30]. Furthermore, these networks are proven to be optimal [31]. The sorting networks for $n = 9$ and $n = 10$ are shown in Fig.4(a). The horizontal lines represent the array elements and the vertical lines mark the comparisons between these elements. The operation principle using a sorting network is demonstrated in Fig.4(b) using an array with 9 elements.

Using a merge sort step to sort the outputs of the sorting networks would yield a total of 63 comparisons at the best-case, while 72 at the worst-case scenarios, respectively. The implementation of the sorting network is also more favorable to the insertion sort with only requiring the indexes of the comparators, therefore no inner loop is required.

Based on our experience, using only one sorting network having $n = 19$ elements has a longer computational time than creating two subgroups. Based on the literature, so far no optimal solution developed $n = 19$ case and the current best solution needs 86 comparisons being made [32]. Currently, the lower bound for the comparisons required for this task is

TABLE 1. Total execution time comparison.

Algorithm	execution time [pu]
Quicksort (19 elements)	1.6
Insertion sort (19 elements)	1.2
Insertion sort using two subgroups (9 and 10 element) and merging	1.11
Insertion sort using four subgroups (5,5,5 and 4 elements) and three merge sorting	1.17
Network sorting (19 elements)	1.73
Network sorting on two subgroups (9 and 10 elements) and merging	1

TABLE 2. Comparison in the number of swaps.

Parameter	Insertion sort using two subgroups (9 and 10 elements) and merging	Network sorting on two subgroups (9 and 10 element) and merging
Average number of swaps	40.09	18.06
Standard deviation of swaps	7.38	3.77
Minimum number of swaps	3	1
Maximum number of swaps	75	39
Maximum number of possible swaps	81	54

estimated to be 70. A second loop is also needed in this case to rank the sorted index list of the voltage vectors.

The solution presented in this paper only focuses on programs developed on one thread, therefore the solution would perform even better if parallelization is an option. The numbers below sorting networks in Fig.4(a) presents the number of comparisons that can be done in parallel. It can result in a drastic improvement in the execution time using for example an FPGA or multicore digital controller.

E. BENCHMARK OF SORTING METHODS

The aforementioned algorithms were implemented for demonstration purposes using Visual Studio 2019 and C language. Each of the algorithms were tested by the same randomly generated 100 million input arrays with $n = 19$ elements. The execution times are recorded to make a comparison between the techniques.

The total computational time for 100 million arrays is presented in Table 1 in per unit. The base value selected to be the computational time of the network sorting method having two subgroups. The actual computational time for this method was 47.17 sec using a desktop computer with an AMD Ryzen 9 3900X processor.

As it can be seen using a sorting network with an array with 19 elements has far the longest computational time. Even the quicksort algorithm has a faster response. However, using the method of sorting network having two subgroups with 10 and 9 elements has the shortest computational time and it is around 20% faster than the widely used insertion sort algorithm for $n = 19$ elements. As it was discussed previously, better performance can be obtained by using subgroups for insertion sorting. However, for $n = 19$, it is not worth to create four subgroups as it results in longer computational time than using only two subgroups.

As the insertion sort and network sorting algorithms having two subgroups have the best performance, they were further analyzed. Table 2 presents a comparison between the number of swaps for the two algorithms.

An important observation is that even during the execution of 100 million randomly generated input arrays, the sorting networks, while always comparing $25 + 29 = 54$ values, the maximum number of swaps was only 39 and the average number of swaps is only around 18. For the insertion sort algorithm, the maximum number of swaps is much closer to the worst-case scenario $(10(10-1)/2 + 9(9-1)/2 = 81$ and the average number of swaps is more than twice as using sorting network algorithm. Furthermore, as it can be seen, the standard deviation is also smaller for the sorting network method, which results in practically constant execution time in the long run.

F. REMARKS

So far, the case of sorting and ranking of all possible voltage vectors, which is $n = 19$ for a three-level inverter, has been investigated. A practical question arises whether the method could not be simplified by sorting and ranking fewer voltage vectors similar as in [18] for a two-level inverter and in [33] for a three-level inverter.

To examine this for our case, a complete and realistic simulation model of the presented MPTC algorithm with the sorting networks is developed using Matlab/Simulink environment by using the parameters of an induction machine with high rated frequency (see next section, Table 3). A large number of simulation tests were carried out at different reference speeds, at different DC link voltages, at different load conditions, and at different sampling frequencies. More than one million voltage vector selections have been evaluated to

find the r_1 and r_2 ranks of the finally selected voltage vector. Figure 5(a) presents a three-dimensional histogram, which shows the relative distribution of r_1 torque and r_2 flux rank values of $\mathbf{v}_{opt}$ (see (17)). For a better visibility, the essential part of the figure is cut out in Fig.5(b).

From the figure, it can be deduced that it may be sufficient to take into account for example only the first 6 voltage vectors and sort only them. For example, it can be realized by a selection routine, which selects and ranks the best 6 voltage vectors according to the value of the torque cost function g_1. It is followed by a sorting algorithm based on the flux cost function. This would probably result in a good performance. However, it should be noted this algorithm cannot be parallelized and the histogram can be different for other parameter values. Furthermore, the best 6 voltage vectors based on the torque cost function may not be the best six choices in terms of the flux cost function. For example, it can be seen in Fig. 5 that in almost 5% of the cases a voltage vector is selected where $r_1 = 3$ and $r_2 = 3$. It means that in that case, the voltage vector with the minimum cost function value for g_1 ($r_1 = 1$) has at most only the seventh best value based on the flux cost function g_2 ($r_2 \geq 7$). The optimal voltage vector according to (17) can be selected if all the possible voltage vectors are taken into account. As it was discussed for this purpose the sorting networks presented in the paper can be a feasible solution.

Furthermore, if the number of voltage levels is increased or the prediction horizon is greater, even if not all possible voltage vectors need to be considered, numerous voltage vectors may need to be sorted for a proper operation. In these cases, the presented sorting networks based method can be also useful to reduce the computational time.

V. RESULTS

A. HARDWARE IMPLEMENTATION

The algorithm of the direct MPTC with the given equations as well as the sorting algorithm using two sorting networks with merging is implemented using C language on a low-cost DSP TMS320F28379D running at 200 MHz clock frequency. This DSP has two cores (CPU1 and CPU2) and each core is equipped with an independent 32-bit floating-point math accelerator called CLA (Control Law Accelerator).

CPU1, as the main core, handles the peripherals (like Timers, ADC, PWM and QEP) and is responsible for diagnostics and communication tasks. Furthermore, it calls the MPTC algorithm with fixed sampling frequency $f_s = 1/T_s$ from an interrupt routine. After reading the current signals and the measured mechanical speed and activating the switching signals of the voltage vector calculated in the previous sampling interval, CPU1 calls a function on CLA1. This function is responsible to calculate the reference signals (M^* and Ψ_s^*) and the prediction of control variables for delay compensation. The results are stored in a shared memory of CPU1 and CPU2. After the delay compensation CPU1 calls another function on CLA1 which predicts the stator current vector and electric torque and then evaluates the g_1 torque

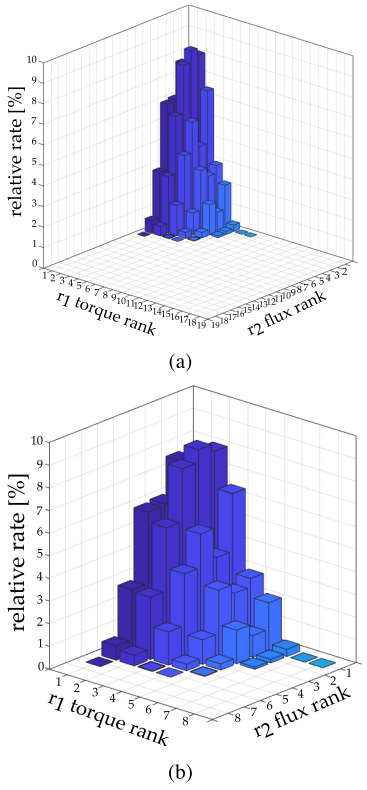


FIGURE 5. Relative distribution of r_1 torque and r_2 flux rank values of $\mathbf{v}_{opt}$.

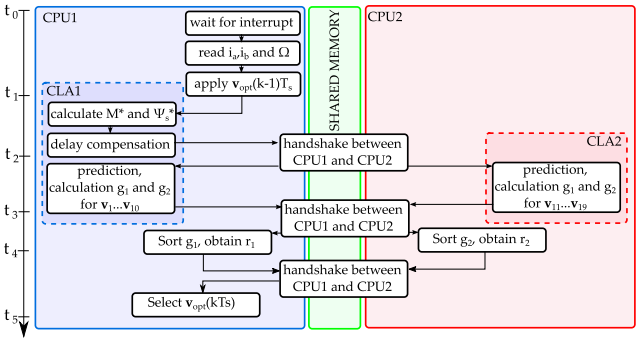
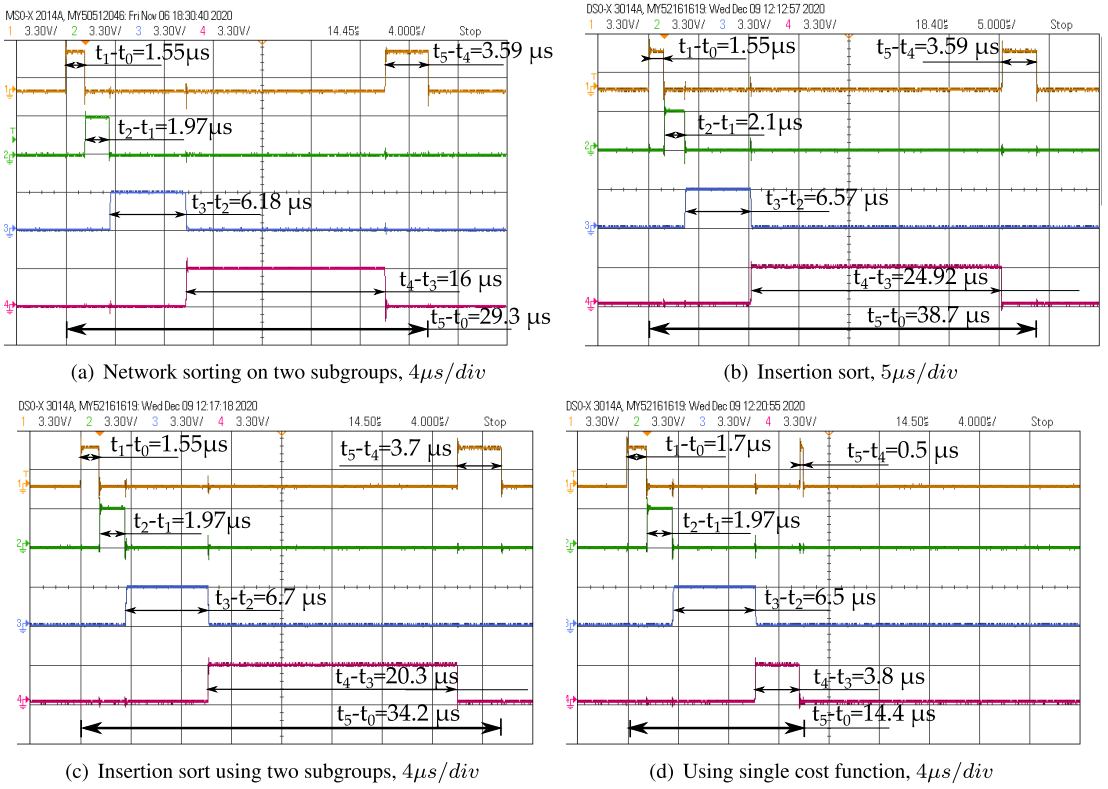


FIGURE 6. Flowchart of the implemented algorithm on a dual core DSP using CLA.

and g_2 flux cost functions for 10 possible voltage vectors. The same is carried out on a function in CLA2, called by CPU2, for the remaining 9 possible voltage vectors. After both CLA-s have finished computing the predictions, the CPU-s share the results in a fashion that CPU1 will have all torque, while CPU2 will have all flux predictions. This is being done via shared memory. The next step is to sort the voltage vectors using sorting networks on the CPU-s. As discussed before, the sorting algorithms will also assign the rank values for the voltage vectors. CPU2 then sends the flux rank values to CPU1 via their shared memory. As a final step CPU1 selects the voltage vector of minimum rank value according to (17). This voltage vector will be applied at the beginning of the next sampling interval. Figure 6 presents the simplified flowchart of the implemented algorithm for the better understanding.



(a) Network sorting on two subgroups, $4\mu s/div$

(b) Insertion sort, $5\mu s/div$

(c) Insertion sort using two subgroups, $4\mu s/div$

(d) Using single cost function, $4\mu s/div$

FIGURE 7. Measured execution times.

The CLA significantly improves the floating-point calculations required for the estimation and prediction phases because of its inherent design. This also means that the CPU will have an advantage in sorting since it is about integer operations. However, this is only true if the sorting operation can be implemented in the RAM, therefore bypassing the complex operations of FLASH memories. An effort was made to further parallelize the previously described algorithm with the prediction and sorting phases shared by the CPU-s and CLA-s, but it was highly inferior for this many voltage vectors.

B. EXECUTION TIME

The execution time of the MPTC algorithm with sorting networks was measured by a digital oscilloscope by setting and clearing IO pins of the DSP. The measured waveform, which presents the duration of the different subtasks can be seen in Fig.7(a). To interpret the time intervals denoted by t_x $(x = 0, 1..5)$ please check the timeline on the left side of Fig.6. The total execution time of the algorithm is measured to be 29.3 μs, which translates to a 34.1 kHz operation speed.

For comparison, the insertion sort using 19 elements as well as the insertion sort using two subgroups (9 and 10 elements) with merging were also implemented on the DSP. Their measured execution times can be seen in Fig.7(b) and Fig.7(c), respectively. It can be seen that, the execution time of insertion sort is around 50% longer, while insertion sort using two subgroups requires around 25% more time compared to the proposed sorting networks based hybrid solution. It means that the proposed sorting network scheme fared even better on an embedded processor compared to what we obtained using a desktop computer (see Table 1). It should be noted that the execution time of the two insertion sort type algorithms can vary heavily, while the proposed sorting network based algorithm is stable with regards to execution time.

Even if the proposed sorting algorithm has the shortest execution time, it still takes up the half of the sampling period (see Fig.7(a)). Although this is the price of using two cost functions, it can only be eliminated by using only one cost function, but then the selection of the proper weighting factors can be an issue. To demonstrate this, the execution time of the MPTC scheme using a single cost function is measured. The measured execution time can be seen in Fig.7(d). After calculating g_1 and g_2 cost functions separately for all the possible voltage vectors using the two CPUs of the DSP, a resulting cost function is obtained as $g = g_1 + k_\Psi g_2$. During this calculation, the voltage vector with the smallest g value can be directly determined. It drastically reduces the computation time (see $t_5 - t_3$ duration in Fig.7(d)).

A similar conclusion can be found in [17], which introduces the concept of MPTC having two separate cost functions for torque and flux control. In [17] the method of quicksort is used for a two-level inverter to select the final voltage vector. As it was demonstrated, the proposed sorting-network based algorithm is much faster than quicksort for a three-level inverter if all the voltage vectors are taken into account.

FIGURE 8. Laboratory test bench (left) and three-level NPC inverter (right).

TABLE 3. Rated data and parameters of the Z62-M260.23 S5 induction machine at rated frequency.

P_n	Rated power (S1)	0.67 kW
n_n	Rated rated speed	60 krpm
f_{1n}	Rated frequency	1000 Hz
$V_{LL,rms}$	Rated voltage	140 V
$I_{ph,rms}$	Rated current	4 A
M_n	Rated torque	0.11 Nm
R_s	Stator resist.	0.82 Ω
R_r	Rotor resist.	0.6 Ω
L_{ls}	Stator leak. induct.	0.48 mH
L_{lr}	Rotor leak. induc.	0.34 mH
L_m	Magnetizin ind.	21 mH
B	Viscous friction c.	0.0141 mNms

Later on, experimental results will be presented only for MPTC using two separate cost functions with the proposed sorting algorithm. There will be no experimental results for the MPTC algorithm using single cost function with weighting factors. The paper focuses on sorting algorithms, not on the comparison of MTPC algorithms using single or multiple cost functions.

C. TESTBENCH

The implemented MPTC algorithm with sorting networks was tested on a drive system consisting of a custom-made three-level NPC inverter and a high-speed induction machine. The inverter was built by using STGF19NC60KD IGBTs and MUR1640CT diodes. The machine was loaded with the help of a high-speed PMSM, with the speed measured by a digital differential magneto resistor. A photo of the drive system and the NPC inverter can be seen in Fig.8. The main parameters of the machine can be found in Table 3. The total DC link voltage of the NPC inverter was $V_{DC} = 60$ V and for simplicity, the potential of the neutral point was fixed. Due to the smaller DC bus voltage, the measurements were carried out at a lower rotational speed than the rated one. It was done as the focal point of the paper is sorting algorithms. The sampling frequency is selected to be 20 kHz ($T_s = 50\mu s$). The parameters of the PI controller (see Fig.1) is calculated by using the symmetrical optimum method. In the DSP a discrete parallel PI controller with a trapezoidal integration method is realized. The parameters are $K_p = 0.004$ and $K_I = 1/T_i = 0.01$.

D. EXPERIMENTAL TESTS

In the paper, a novel hybrid algorithm of two sorting networks combined with a merging step is introduced. In this subsection some steady-state and transient responses of the MPTC scheme using two cost functions are presented, where the proposed sorting algorithm is used to select the optimal voltage vector.

Figure 9 shows the measured steady-state behavior of the drive at no-load and at rated load when the reference speed is $n^* = 6$ krpm ($\Omega^* = 628$ rad/s, $f_1 \approx 119$ Hz) and $n^* = 12$ krpm ($\Omega^* = 1256$ rad/s, $f_1 \approx 219$ Hz). The stator flux reference is set in both cases to $\Psi_s^* = 0.018$ Vs. The variables in the figures are the real part of the stator current vector $\mathbf{i}_s$, the real part of the stator flux Ψ_s and the real part of the stator voltage vector $\mathbf{v}_s$. As it can be seen the stator flux tracks its reference value. All variables show the typical waveforms delivered by a three-level inverter. Similar steady-state results can be found in [26], [33] or in [34]. It can be concluded that the proposed sorting algorithm works properly in the MPTC scheme.

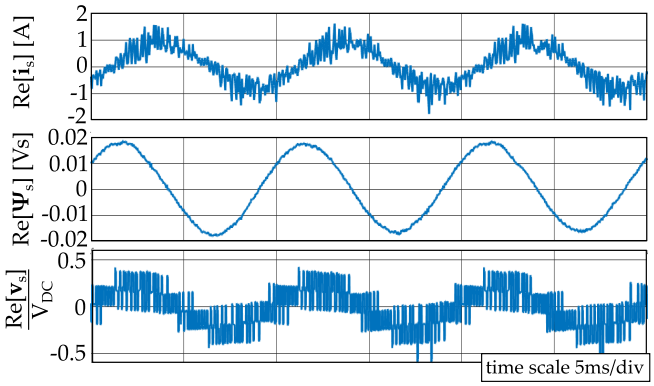
During the measurement, the Total Harmonic Distortion (THD) of the stator current was calculated based on [29], with 20 cycles up to a maximum 10 kHz with the help of the *power_fftscope* function of Matlab. The THD values are given in the caption of the figures. The switching frequency was estimated by counting the number of ON transitions over a given time interval and dividing the sum by the interval's length. The average device switching frequency, given as f_{sw} in the caption of Fig.9, is obtained by dividing the computed fraction by the number of transistors [29].

It can be concluded, that the obtained THD value is greater than the typical values in the literature at similar operating conditions. For example in paper [26], where an NPC inverter fed induction machine using MPTC with a single cost function is studied, the measured current THD values at loaded condition are between 3.5-4.5% at similar switching ($1.2 - 1.7$ kHz) and sampling frequencies. Similar THD values like in [26] are presented for an NPC inverter fed induction machine in [34], where the THD of the current at rated condition is around 3.5% when the average switching frequency is 1.6 kHz.

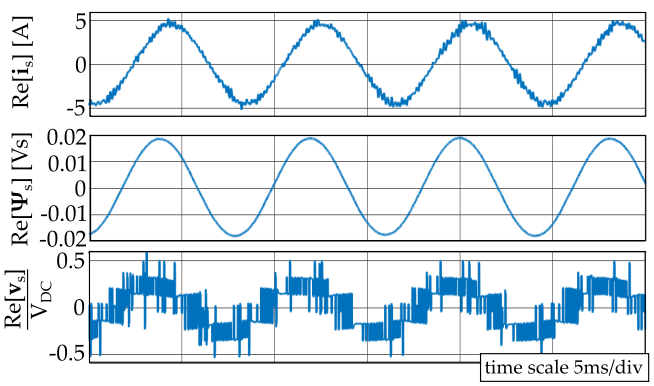
The larger value of THD in our case can be caused by the special parameters of the induction machine with high rated frequency, like the extremely low value of the transient stator inductance $\sigma L_s = 0.81$ mH. It can be the source for much higher harmonic current components at the same harmonic voltage components.

It should be noted, that it is possible to modify the MPTC algorithm by including other constraints to improve for example the harmonic performance of the drive [1], [26]. However, in this paper, preference was given to the sorting algorithms and the basic MPTC concept using two cost functions is not changed.

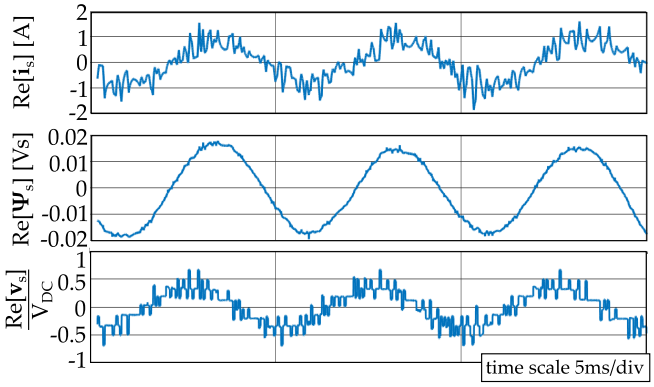
The switching and conduction losses of the transistors and the diodes were estimated using the measurement results based on [35]–[37] and the datasheets of the switching devices. The switching losses (denoted as P_{sw}) is given in the captions in Figure 9. The η efficiency of the total drive system is also determined as $\eta = \frac{P_{out}}{P_{out}+P_{loss}}$, where P_{loss} denotes



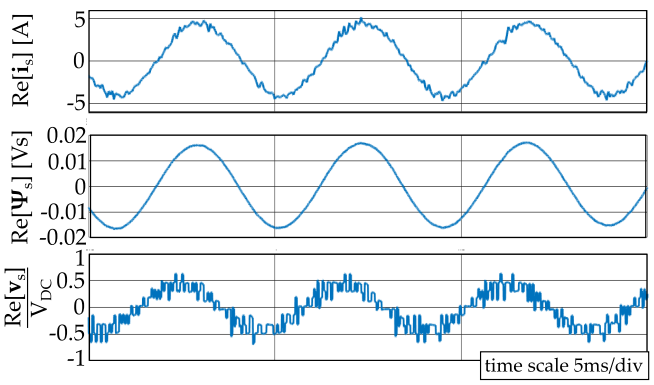
(a) $\Omega^* = 628$ rad/s, No-load, $I_{THD} = 44.1\%$, $P_{sw} = 1.66$ W, $f_{sw} = 1.9\,kHz$



(b) $\Omega^* = 628$ rad/s ($n^* = 6000rpm$), $M_{load} = 0.11$ Nm, $I_{THD} = 7.1\%$,$P_{sw} = 6.43$ W, $\eta = 51\%$, $f_{sw} = 1.23\,kHz$



(c) $\Omega^* = 1256$ rad/s, No-load, $I_{THD} = 45,1\%$, $P_{sw} = 1.56$ W, $f_{sw} = 1.2\,kHz$



(d) $\Omega^* = 1256$ rad/s, $M_{load} = 0.11$ Nm, $I_{THD} = 6.5\%$, $\eta = 62.3\%$, $P_{sw} = 7.4$ W, $f_{sw} = 1.6$ kHz

FIGURE 9. Experimental results for steady-state at $\Omega^* = 628$ rad/s ($n^* = 6000$ *rpm*) and at $\Omega^* = 1256$ rad/s ($n^* = 12000$ *rpm*), $f_s = 20$ kHz.

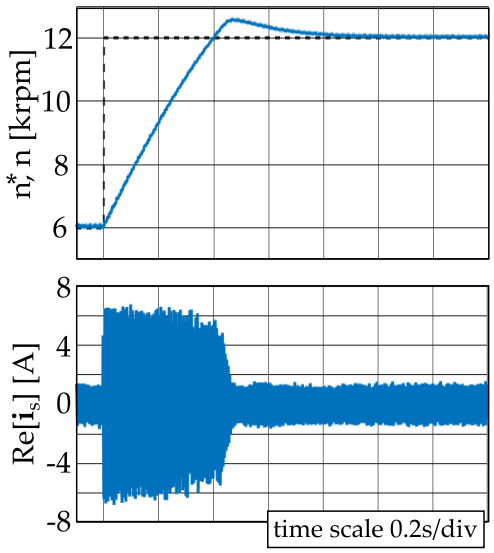


FIGURE 10. Experimental results for transients.

the cumulative losses of both the inverter and the induction machine. The respective value of η for the loaded condition can be found also in the captions. It should be noted that as the machine operates at lower speeds than its rated value, and the slip is higher than its rated value, the stator and rotor copper losses therefore significantly reduce the total efficiency.

Figure 10 shows the drive's measured dynamic response for a sudden change in the reference speed when n^* is changed from 6 krpm to 12 krpm. The variables recorded are the mechanical speed n and the real part of the stator current vector $\mathbf{i}_s$. The amplitude of the stator flux is kept constant. It can be observed that the stator current has an immediate increase in its amplitude, generating a fast change in the torque. The mechanical speed reaches the reference value after a short overshoot. The digital oscilloscope, which was used for recording the transients, did undersample the current signal. This is the source of the observable small fluctuations in the waveform of the current. The parameters of the PI controller could be further adjusted so that the system would have a more favorable transient response, but this is not the primary goal of the paper.

VI. CONCLUSION

The paper focuses on sorting and ranking algorithms, which are necessary for MPTC schemes relying on multiple cost functions to select the final voltage vector. The paper reviews a few possible solutions for three-level inverters, where the number of possible voltage vector is $n = 19$, and a novel hybrid algorithm of two sorting networks combined with a merging step is introduced. The techniques are implemented and a detailed benchmark is carried out. As it is shown the

presented hybrid method can reduce the computational time compared to other widely applied methods for three-level inverters. The method presented can be applied in essence for smaller and greater n values as well.

As a case study the MPTC algorithm, using separate cost functions for the torque and the stator flux, is realized using a three-level NPC inverter fed induction machine with high rated frequency. The stator flux is predicted using the estimated rotor flux, which is calculated from the current model using Tustin approximation. The algorithm can be implemented on a digital device by the recursive equations and the flow chart presented in the paper.

The MPTC algorithm with the hybrid sorting and ranking algorithm was implemented on a low-cost dual-core DSP. The way of implementation is summarized in the paper. Experimental results verify the operation of the MPTC scheme with the proposed sorting algorithm.

APPENDIX
PREDICTOR EQUATIONS WITH NUMERICAL VALUES

The stator current (7), rotor flux (11) and stator flux (13) equations with the exact numerical values of the actual system are given below. The detailed system parameters can be found in Table 3 and $T_s = 50\mu s$.

$$\begin{aligned}\mathbf{i}_s^p(kT_s + T_s) = {} & 0.9140\mathbf{i}_s(kT_s) + 0.0614\mathbf{v}_s(kT_s) + \\ & - j0.0604\omega\hat{\Psi}_r(kT_s) + 1.6983\hat{\Psi}_r(kT_s)\end{aligned} \tag{A.1}$$

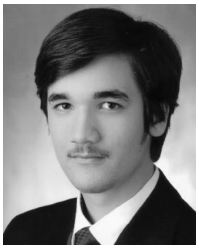
$$\begin{aligned}\hat{\Psi}_r(kT_s) = {} & 0.9986\hat{\Psi}_r((kT_s - T_s)) \\ & + 1.4751 \cdot 10^{-5}(\mathbf{i}_s(kT_s) + \mathbf{i}_s(kT_s - T_s))\end{aligned} \tag{A.2}$$

$$\hat{\Psi}_s^p(kT_s + T_s) = 0.9841\hat{\Psi}_r(kT_s) + 8.1458 \cdot 10^{-4}\mathbf{i}_s(kT_s) \tag{A.3}$$

REFERENCES

[1] P. Karamanakos and T. Geyer, "Guidelines for the design of finite control set model predictive controllers," *IEEE Trans. Power Electron.*, vol. 35, no. 7, pp. 7434–7450, Jul. 2020.
[2] S. Vazquez, J. Rodriguez, M. Rivera, L. G. Franquelo, and M. Norambuena, "Model predictive control for power converters and drives: Advances and trends," *IEEE Trans. Ind. Electron.*, vol. 64, no. 2, pp. 935–947, Feb. 2017.
[3] F. Wang, Z. Zhang, X. Mei, J. Rodríguez, and R. Kennel, "Advanced control strategies of induction machine: Field oriented control, direct torque control and model predictive control," *Energies*, vol. 11, no. 1, p. 120, Jan. 2018.
[4] P. Karlovsky and J. Lettl, "Induction motor drive direct torque control and predictive torque control comparison based on switching pattern analysis," *Energies*, vol. 11, no. 7, p. 1793, Jul. 2018.
[5] P. Karamanakos, P. Stolze, R. M. Kennel, S. Manias, and H. D. T. Mouton, "Variable switching point predictive torque control of induction machines," *IEEE J. Emerg. Sel. Topics Power Electron.*, vol. 2, no. 2, pp. 285–295, Jun. 2014.
[6] F. Donoso, A. Mora, R. Cardenas, A. Angulo, D. Saez, and M. Rivera, "Finite-set model-predictive control strategies for a 3L-NPC inverter operating with fixed switching frequency," *IEEE Trans. Ind. Electron.*, vol. 65, no. 5, pp. 3954–3965, May 2018.
[7] O. Gonzalez, M. Ayala, J. Doval-Gandoy, J. Rodas, R. Gregor, and M. Rivera, "Predictive-fixed switching current control strategy applied to six-phase induction machine," *Energies*, vol. 12, no. 12, p. 2294, Jun. 2019.
[8] N. Oikonomou, C. Gutscher, P. Karamanakos, F. D. Kieferndorf, and T. Geyer, "Model predictive pulse pattern control for the five-level active neutral-point-clamped inverter," *IEEE Trans. Ind. Appl.*, vol. 49, no. 6, pp. 2583–2592, Nov. 2013.
[9] M. Vasiladiotis, A. Christe, and T. Geyer, "Model predictive pulse pattern control for modular multilevel converters," *IEEE Trans. Ind. Electron.*, vol. 66, no. 3, pp. 2423–2431, Mar. 2019.
[10] R. O. Ramirez, J. R. Espinoza, F. Villarroel, E. Maurelia, and M. E. Reyes, "A novel hybrid finite control set model predictive control scheme with reduced switching," *IEEE Trans. Ind. Electron.*, vol. 61, no. 11, pp. 5912–5920, Nov. 2014.
[11] H. Yang, P. Huang, Y. Zhang, and J. Zhu, "Model predictive flux control based on synchronous pulse-width modulation," in *Proc. IEEE Energy Convers. Congr. Expo. (ECCE)*, Oct. 2020, pp. 2701–2707.
[12] F. Wang, S. A. Davari, Z. Chen, Z. Zhang, D. A. Khaburi, J. Rodríguez, and R. Kennel, "Finite control set model predictive torque control of induction machine with a robust adaptive observer," *IEEE Trans. Ind. Electron.*, vol. 64, no. 4, pp. 2631–2641, Apr. 2017.
[13] Y. Zhang and H. Yang, "Model-predictive flux control of induction motor drives with switching instant optimization," *IEEE Trans. Energy Convers.*, vol. 30, no. 3, pp. 1113–1122, Sep. 2015.
[14] X. Zhang, L. Zhang, and Y. Zhang, "Model predictive current control for PMSM drives with parameter robustness improvement," *IEEE Trans. Power Electron.*, vol. 34, no. 2, pp. 1645–1657, Feb. 2019.
[15] T. Geyer, "Algebraic tuning guidelines for model predictive torque and flux control," *IEEE Trans. Ind. Appl.*, vol. 54, no. 5, pp. 4464–4475, Sep. 2018.
[16] S. A. Davari, D. A. Khaburi, and R. Kennel, "An improved FCS–MPC algorithm for an induction motor with an imposed optimized weighting factor," *IEEE Trans. Power Electron.*, vol. 27, no. 3, pp. 1540–1551, Mar. 2012.
[17] C. A. Rojas, J. Rodriguez, F. Villarroel, J. R. Espinoza, C. A. Silva, and M. Trincado, "Predictive torque and flux control without weighting factors," *IEEE Trans. Ind. Electron.*, vol. 60, no. 2, pp. 681–690, Feb. 2013.
[18] M. Norambuena, J. Rodriguez, Z. Zhang, F. Wang, C. Garcia, and R. Kennel, "A very simple strategy for high-quality performance of AC machines using model predictive control," *IEEE Trans. Power Electron.*, vol. 34, no. 1, pp. 794–800, Jan. 2019.
[19] D. Mihhailov, V. Sklyarov, I. Skliarova, and A. Sudnitson, "Hardware implementation of recursive sorting algorithms," in *Proc. Int. Conf. Electron. Devices, Syst. Appl. (ICEDSA)*, Apr. 2011, pp. 33–38.
[20] S.-C. Yang, Y.-L. Hsu, P.-H. Chou, J.-Y. Chen, and G.-R. Chen, "Digital implementation issues on high speed permanent magnet machine FOC drive under insufficient sample frequency," *IEEE Access*, vol. 7, pp. 61484–61493, 2019.
[21] Q.-L. Meng, J. Li, H. Li, and Y.-Z. Yan, "Model predictive control of induction motors based on improved discretizing method under low switching frequency," in *Proc. IECON-43rd Annu. Conf. IEEE Ind. Electron. Soc.*, Oct. 2017, pp. 5144–5149.
[22] P. Stumpf and A. L. Varadi, "Investigation of estimator algorithms for high speed drive systems," in *Proc. 18th Int. Conf. Mechatronics*, Dec. 2018, pp. 1–8.
[23] P. Stumpf and I. Bara, "Model predictive torque control with synchronized sampling frequency for high frequency induction machine drives," in *Proc. IEEE 29th Int. Symp. Ind. Electron. (ISIE)*, Jun. 2020, pp. 332–338.
[24] P. Cortes, J. Rodriguez, C. Silva, and A. Flores, "Delay compensation in model predictive current control of a three-phase inverter," *IEEE Trans. Ind. Electron.*, vol. 59, no. 2, pp. 1323–1325, Feb. 2012.
[25] J. Rodriguez and P. Cortes, *Predictive Control of Power Converters and Electrical Drives*. Hoboken, NJ, USA: Wiley, 2012.
[26] M. Habibullah, D. D.-C. Lu, D. Xiao, I. Osman, and M. F. Rahman, "Selected prediction vectors based FS-PTC for 3L-NPC inverter fed motor drives," *IEEE Trans. Ind. Appl.*, vol. 53, no. 4, pp. 3588–3597, Jul. 2017.
[27] *Quicksort*. Accessed: Sep. 4, 2020. [Online]. Available: https://www.geeksforgeeks.org/quick-sort/
[28] *Insertion Sort*. Accessed: Jul. 25, 2020. [Online]. Available: https://www.geeksforgeeks.org/insertion-sort/
[29] B. Stellato, T. Geyer, and P. J. Goulart, "High-speed finite control set model predictive control for power electronics," *IEEE Trans. Power Electron.*, vol. 32, no. 5, pp. 4007–4020, May 2017.

[30] D. E. Knuth, *The Art of Computer Programming, Volume 3: (2nd Ed.) Sorting and Searching*. Reading, MA, USA: Addison-Wesley Longman Publishing Co., 1998.
[31] M. Codish, L. Cruz-Filipe, M. Frank, and P. Schneider-Kamp, "Sorting nine inputs requires twenty-five comparisons," *J. Comput. Syst. Sci.*, vol. 82, no. 3, pp. 551–563, May 2016.
[32] V. K. Valsalam and R. Miikkulainen, "Using symmetry and evolutionary search to minimize sorting network," *J. Mach. Learn. Res.*, vol. 14, no. 69, pp. 303–331, 2013.
[33] Y. Yang, H. Wen, M. Fan, M. Xie, and R. Chen, "Fast finite-switching-state model predictive control method without weighting factors for T-type three-level three-phase inverters," *IEEE Trans. Ind. Informat.*, vol. 15, no. 3, pp. 1298–1310, Mar. 2019.
[34] D. Xiao, K. S. Alam, I. Osman, M. P. Akter, S. M. S. I. Shakib, and M. F. Rahman, "Low complexity model predictive flux control for three-level neutral-point clamped inverter-fed induction motor drives without weighting factor," *IEEE Trans. Ind. Appl.*, vol. 56, no. 6, pp. 6496–6506, Nov. 2020.
[35] F. Blaabjerg, J. K. Pedersen, S. Sigurjonsson, and A. Elkjaer, "An extended model of power losses in hard-switched IGBT-inverters," in *Proc. IAS Conf. Rec. IEEE Ind. Appl. Conf. 31st IAS Annu. Meeting*, vol. 3, Dec. 1996, pp. 1454–1463.
[36] Y. Zhu, M. Xiao, X. Su, G. Yang, K. Lu, and Z. Wu, "Modeling of conduction and switching losses for IGBT and FWD based on SVPWM in automobile electric drives," *Appl. Sci.*, vol. 10, no. 13, p. 4539, Jun. 2020.
[37] *Loss Calculation in a Three-Phase 3-Level Inverter*. Accessed: Dec. 10, 2020. [Online]. Available: https://www.mathworks.com/help/physmod/sps/ug/loss-calculation-in-a-three-phase-3-level-inverter.html



KRISTÓF BÁNDY was born in Budapest, Hungary, in 1995. He received the B.S. degree in mechatronics engineering from the Budapest University of Technology and Economics, in 2019, where he is currently pursuing the M.S. degree in mechatronics engineering.

Since 2020, he has been a Research Assistant with the Department of Automation and Applied Informatics, Faculty of Electrical Engineering and Informatics, Budapest University of Technology and Economics. His current research interests focus on model predictive control of electric drives.

PETER STUMPF was born in Budapest, Hungary, in 1985. He received the M.S. degree in mechanical engineering and the Ph.D. degree in mechanical engineering from the Budapest University of Technology and Economics, in 2009 and 2014, respectively.

In 2013 and 2019, he was a Senior Lecturer with the Department of Automation and Applied Informatics, Faculty of Electrical Engineering and Informatics, Budapest University of Technology and Economics. Since 2019, he has been an Associate Professor. His current research interests focus on high speed electrical drives, power electronics, and variable structure nonlinear systems.