

Model Predictive Direct Slope Control for Power Converters

Baljit S. Riar, *Member, IEEE*, James Scoltock, *Member, IEEE*, and
Udaya K. Madawala, *Senior Member, IEEE*

Abstract—Model predictive control (MPC) schemes have become popular in the field of power electronics due to their intuitive formulation, flexibility, and ease of implementation. Typically, these schemes have been implemented with the prediction horizon limited to one time-step, and extension of the prediction horizon over multiple time-steps remains an ongoing area of research. In this paper, a variant of the MPC strategy is proposed wherein the slope of the output trajectories is used to emulate long prediction horizons. Each of the outputs, e.g. current, voltage, torque, or flux, is regulated within a set of symmetrical bounds. When switching is necessitated due to collision with a bound, the switching state that yields the set of output trajectories with the minimum slope, relative to the reference trajectory, is applied to the converter. The key benefit of this approach is its ability to achieve low switching frequencies with a minimal level of computational burden. The feasibility of the scheme, which can be adapted easily to different case studies, is demonstrated through simulations of both a Medium-Voltage (MV) induction machine drive and a grid-connected converter. Experimental results, which are presented for a 1.68 kVA prototype grid-connected Neutral-Point-Clamped (NPC) converter, further demonstrate the practical viability of the proposed strategy.

Index Terms—Current control, model predictive control, neutral point clamped converter, slope control.

I. INTRODUCTION

Since being introduced in the 1970s, model predictive control (MPC) has predominantly been used for process control in the petrochemical and electrical power industries [1], and until recently, applications of MPC outside of process control have been uncommon. Recently, however, the development of more powerful processors has enabled the use of MPC in applications with much faster dynamics than those seen in the aforementioned industries; over the last decade, this has come to include power electronics [2]–[5]. Among the advantages of MPC within power electronics are the abilities to handle constraints, multiple inputs and outputs, and non-linearities in a simple and intuitive manner. Additionally, MPC has been shown to offer improved dynamic performance relative to the conventional approaches, e.g. [6].

Several formulations of MPC have emerged within the field of power electronics [5]; among these, the Finite Control Set-MPC (FCS-MPC) strategy has gained the most popularity.

B. S. Riar, is with the Department of Engineering Science, University of Oxford, Oxford, OX1 3PJ, U.K. (email: baljitriar@ieee.org).

J. Scoltock, is with the School of Electrical and Electronic Engineering, The University of Manchester, M13 9PL Manchester, U.K. (email: james.scoltock@manchester.ac.uk).

U. K. Madawala is with the Department of Electrical and Computer Engineering, The University of Auckland, 1052 Auckland, New Zealand (email: u.madawala@auckland.ac.nz).

By directly manipulating the switching state of the converter, FCS-MPC combines the control and modulation problems into a single task. Additionally, FCS-MPC is conceptually straightforward, relatively simple to implement, and is flexible, having been adapted to renewable energy and grid-connected converters [7], [8], machine drives [9]–[11], and standalone inverters [12]. Typically, FCS-MPC is formulated with a prediction horizon, N_p , of one time-step, although recent research indicates that by employing the appropriate optimisation techniques, the prediction horizon can be extended beyond $N_p = 1$ whilst retaining a reasonable level of computational burden [13], [14].

Over the last decade, a related approach has emerged in parallel with FCS-MPC. Model Predictive Direct Control (MPDxC) [2], variants of which include Torque (MPDTC) [15], [16] and Current (MPDCC) [17], also approaches the control and modulation problems as a single task. Unlike FCS-MPC, however, MPDxC regulates the output variables within a set of bounds, and by employing a move-blocking (extrapolation) strategy, wherein the number of transitions within the horizon is limited, very long prediction horizons can be achieved. The number of ‘moves’ within each prediction is controlled by the so-called switching horizon, N_s , which consists of a fixed sequence of switch (S) and extrapolate (E) events, e.g. ‘SE’, ‘SSESE’, and so on. A consequence of this approach is that very low switching frequencies are achievable, making MPDxC a strong candidate for high-power applications [11], [18], [19].

A downside of MPDxC, relative to FCS-MPC, is the high computational burden that it poses; attempts have been made to address this issue, for example [20] proposes a variant of the generalised MPDTC algorithm which uses a branch-and-bound algorithm to solve the optimisation problem. Additionally, MPDxC is more conceptually difficult than FCS-MPC with $N_p = 1$, the algorithm being more complicated to implement. Nonetheless, MPDxC has been shown to offer performance improvements over the standard FCS-MPC strategies in high-power/low switching frequency applications, e.g. [11], and therefore remains a valuable approach.

This paper proposes a modified MPC-based strategy, through which the performance of the MPDxC strategy with $N_s = \text{‘SE’}$ is emulated without the use of extrapolation. This is achieved by adopting the bounds that are used in MPDxC and by minimising the predicted slope of the output variables over a prediction horizon fixed at $N_p = 1$. The resulting strategy, named Model Predictive Direct Slope Control (MPDSC), possesses the key benefit of emulating the performance of

MPDxC with a significantly simpler algorithm, the complexity of which is comparable with the standard formulations of FCS-MPC. As a result, the necessary processing power for MPDSC is substantially reduced relative to MPDxC, with the implementation of MPDSC also being more straightforward.

The performance of the MPDSC scheme is benchmarked against both the MPDTC and MPDCC strategies through simulation, with the respective case studies consisting of a Medium-Voltage (MV) induction machine drive and grid-connected converter. Comparison of the waveforms, as well as a detailed Pareto analysis over a range of operating points, indicate that MPDSC offers a level of performance which is very similar to MPDxC. Additionally, the practical validity of the MPDSC scheme is verified through experimental results. Using a 1.68 kVA prototype grid-connected Neutral-Point-Clamped (NPC) converter, it is shown that the practical performance of MPDSC remains very close to that of MPDCC, while also having an execution time which is 68% lower.

It should be noted that, although a slope-based approach has been proposed previously for torque and flux control of a machine [21], the MPDSC concept differs from this in several ways. In particular, MPDSC imposes only one set of bounds around the output, whereas the existing method uses both inner and outer bounds for selecting switch positions. Moreover, the presentation of MPDSC in this paper is generalised, with the algorithm being applicable to a range of control problems (such as torque and flux, currents, power, etc.). Additionally, [21] provides no detailed simulated or experimental results, and no comparative performance evaluation, all of which are crucial in demonstrating the validity of the strategy and which are provided in this paper.

II. CASE STUDIES AND MODELLING

In this section, two case studies are presented - firstly, a medium-voltage (MV) induction machine drive, and secondly, a MV grid-connected converter. For both studies, a three-phase neutral-point-clamped (NPC) converter is used. A representation of such a setup is shown in Fig. 1, where the inductor, L , corresponds to either the stator leakage inductance or the line filter inductance, and where the voltage, e_r , $r \in \{a, b, c\}$, corresponds to either the stator voltage or the grid voltage. Throughout this paper $t = kT_s$, where $t \in \mathbb{R}$ denotes (continuous) time, $k \in \mathbb{N}_0$ denotes the current (discrete-) time-step, and T_s is the sampling interval. Variables $\xi_{abc} = [\xi_a \ \xi_b \ \xi_c]^\top$ in the three-phase abc reference frame are transformed to $\xi_{\alpha\beta} = [\xi_\alpha \ \xi_\beta]^\top$ in the orthogonal $\alpha\beta$ reference frame through

$$\xi_{\alpha\beta} = \frac{2}{3}P\xi_{abc} \quad (1)$$

where P is the transformation matrix

$$P = \begin{bmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} \end{bmatrix}. \quad (2)$$

Conversely, variables can be converted from the $\alpha\beta$ reference frame to the abc reference frame via

$$\xi_{abc} = P^\top \xi_{\alpha\beta}. \quad (3)$$

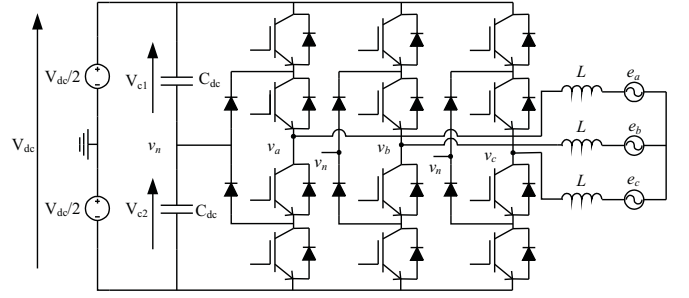


Fig. 1: Schematic of a neutral-point-clamped converter.

It is assumed that the dc-link voltage, V_{dc} , is constant and free of ripple. Additionally, it is assumed that the systems are free of phase imbalance and that non-ideal characteristics, e.g. inductor non-linearity and temperature-dependent effects, can be ignored.

A. NPC Converter Model

The switching state of the NPC converter is denoted as

$$\mathbf{u}_{abc} = [u_a \ u_b \ u_c]^\top \in \{-1, 0, 1\}^3. \quad (4)$$

The total DC-link voltage is the sum of the voltages across the upper and lower capacitors, i.e. $V_{dc} = V_{C1} + V_{C2}$. The neutral-point voltage is defined relative to the mid-point of the DC-link and is given by $v_n = (V_{C2} - V_{C1})/2$. The continuous-time dynamics of the neutral-point voltage, v_n , can be related to switching state, \mathbf{u}_{abc} , and converter currents, \mathbf{i}_{abc} , as

$$\frac{dv_n}{dt} = \frac{1}{2C_{dc}} |\mathbf{u}_{abc}|^\top \mathbf{i}_{abc} \quad (5)$$

where $|\mathbf{u}_{abc}| = [|u_a| \ |u_b| \ |u_c|]^\top$. The output voltage of each phase of the converter, v_r , $r \in \{a, b, c\}$, has three levels, $V_{dc}/2$, 0, and $-V_{dc}/2$, which correspond to three switching states of 1, 0, and -1 , respectively. The 27 switching states of the NPC converter are transformed into 19 space-vectors of the form $\mathbf{u}_{\alpha\beta} = [u_\alpha \ u_\beta]^\top$ via (1), with the output voltage vectors of the NPC converter being given by

$$\mathbf{v}_{\alpha\beta} = \frac{V_{dc}}{2} \mathbf{u}_{\alpha\beta}. \quad (6)$$

where $\mathbf{v}_{\alpha\beta} = [v_\alpha \ v_\beta]^\top$. For the converter under consideration, all switching transitions are allowed except for those which involve switching between the upper and lower rails. For example, a transition from the switching state $\mathbf{u}_{abc} = [1 \ 0 \ 0]^\top$ to $[-1 \ 0 \ 0]^\top$ is not allowed.

B. Drive Model

The IM drive is modeled with the mechanical load assumed to be constant. The $\alpha\beta$ -frame stator (converter) current and rotor flux vectors are denoted as $\mathbf{i}_{s,\alpha\beta} = [i_{s\alpha} \ i_{s\beta}]^\top$ and $\Psi_{r,\alpha\beta} = [\Psi_{r\alpha} \ \Psi_{r\beta}]^\top$, respectively, and the state vector \mathbf{x}_D is defined as

$$\mathbf{x}_D = [i_{s,\alpha\beta}^\top \ \Psi_{r,\alpha\beta}^\top]^\top = [i_{s\alpha} \ i_{s\beta} \ \Psi_{r\alpha} \ \Psi_{r\beta}]^\top. \quad (7)$$

The continuous-time state equation can then be written as [22]

$$\frac{d\mathbf{x}_D}{dt} = \mathbf{A}_D \mathbf{x}_D + \mathbf{B}_D \mathbf{v}_{\alpha\beta} \quad (8)$$

where the matrices \mathbf{A}_D and \mathbf{B}_D are given in Appendix A. The electromagnetic torque, T_e , is given by

$$T_e = \frac{3}{2} k_r (i_{s\beta} \Psi_{r\alpha} - i_{s\alpha} \Psi_{r\beta}) \quad (9)$$

where the rotor coupling factor, k_r , is given in Appendix A. Finally, the relationship between rotor speed and torque is given by

$$\frac{d\omega_r}{dt} = \frac{1}{J} (T_e - T_l). \quad (10)$$

Equations (7) – (10) provide a complete description of the dynamic behaviour of the IM when non-idealities such as magnetic saturation, the skin effect and variations in the rotor resistance are ignored.

C. Grid Model

The continuous-time dynamics of the grid currents and grid voltages are now modelled. The $\alpha\beta$ -frame grid (converter) current and grid voltage vectors are denoted as $\mathbf{i}_{\alpha\beta} = [i_\alpha \ i_\beta]^\top$ and $\mathbf{v}_{g,\alpha\beta} = [v_{g,\alpha} \ v_{g,\beta}]^\top$, respectively, and the state vector \mathbf{x}_G is defined as

$$\mathbf{x}_G = [\mathbf{i}_{\alpha\beta}^\top \ \mathbf{v}_{g,\alpha\beta}^\top]^\top = [i_\alpha \ i_\beta \ v_{g,\alpha} \ v_{g,\beta}]^\top. \quad (11)$$

The continuous-time state equation can then be written as

$$\frac{d\mathbf{x}_G}{dt} = \mathbf{A}_G \mathbf{x}_G + \mathbf{B}_G \mathbf{v}_{\alpha\beta} \quad (12)$$

where definitions of the system matrices \mathbf{A}_G and \mathbf{B}_G are given in Appendix A. Finally, the real and reactive power delivered to the grid, p and q , are given by

$$p = \frac{3}{2} (v_{g\alpha} i_\alpha + v_{g\beta} i_\beta), \quad q = \frac{3}{2} (v_{g\beta} i_\alpha - v_{g\alpha} i_\beta). \quad (13)$$

It should be noted that the model that has been presented assumes that the reference direction of the currents is from converter to grid.

D. Model Discretisation

Here, the converter, drive and grid models are discretised. In this section, for the sake of brevity, $\mathbf{x} \in \{\mathbf{x}_D, \mathbf{x}_G\}$, $\mathbf{A} \in \{\mathbf{A}_D, \mathbf{A}_G\}$, and $\mathbf{B} \in \{\mathbf{B}_D, \mathbf{B}_G\}$.

Recalling that the stator and grid currents are both identical to the converter currents, the non-linear dynamics of the neutral-point voltage are expressed by

$$v_n(k+1) = \frac{T_s}{2C_{dc}} |\mathbf{u}_{abc}|^\top [\mathbf{P}^\top \mathbf{0}_{3 \times 2}] \mathbf{x}(k) + v_n(k) \quad (14)$$

where $\mathbf{0}_{3 \times 2}$ is the 3×2 zero matrix. From the continuous-time state equations from (7) and (12), the linear discrete-time state equations of the form

$$\mathbf{x}(k+1) = \mathbf{F} \mathbf{x}(k) + \mathbf{G} \mathbf{v}_{\alpha\beta}(k) \quad (15)$$

can be derived for both the drive and grid, where the matrices \mathbf{F} and \mathbf{G} are given by

$$\mathbf{F} = e^{\mathbf{A}T_s}, \quad \mathbf{G} = \mathbf{A}^{-1}(\mathbf{F} - \mathbf{I}_4)\mathbf{B} \quad (16)$$

where \mathbf{I}_4 is the 4×4 identity matrix.

For the drive, the output vector $\mathbf{y}_D(k)$ is composed of the electromagnetic torque, stator flux magnitude, $|\Psi_s(k)|$, and neutral-point potential, being defined as

$$\mathbf{y}_D(k) = [T_e(k) \ |\Psi_s(k)| \ v_n(k)]^\top = g_D(\mathbf{x}_D(k), v_n(k)) \quad (17)$$

and for the grid is composed of the $\alpha\beta$ -frame grid currents and the neutral-point potential, i.e.

$$\mathbf{y}_G(k) = [i_\alpha(k) \ i_\beta(k) \ v_n(k)]^\top = g_G(\mathbf{x}_G(k), v_n(k)). \quad (18)$$

Concrete forms of $g_D(\mathbf{x}_D(k), v_n(k))$ and $g_G(\mathbf{x}_G(k), v_n(k))$ are straightforward to derive based on the preceding modelling, and can be found in, e.g., [17], [23] and [24], respectively.

III. MODEL PREDICTIVE DIRECT SLOPE CONTROL

A. Background

MPDTC and MPDCC have been demonstrated as effective approaches for the control of high-power converters in several works, with significant reductions in switching frequency over classical (PWM-based) and FCS-MPC strategies having been observed [11]. However, MPDxC also poses a significant computational burden, which can be attributed to one key feature of the framework- namely, the use of extension steps, which require the use of iterative calculations or numerical approximation techniques to predict the state and output trajectories over a long prediction horizon.

In this section, Model Predictive Direct Slope Control (MPDSC) is proposed as an alternative to MPDxC. Under MPDSC, constraints in the form of symmetrical bounds are imposed around each of the output references, a feature which is inherited directly from MPDxC. Instead of using an extension step, however, MPDSC *emulates* the performance of MPDxC by instead minimising the slope of the output variables over a single time-step. As a result, the computational requirements of MPDSC are lower than those of MPDxC and are more similar to those of FCS-MPC-based approaches.

B. Overview of MPDxC

Before explaining the MPDSC concept, MPDxC is first discussed for the case with $N_s = \text{'SE'}$. The first objective of the conventional MPDxC scheme is to regulate the output variables by keeping them within a set of bounds defined about each reference. The second objective is to minimise the switching losses of the converter, which is achieved indirectly, by minimising the number of switching commutations over time¹. As mentioned in the introduction, MPDxC achieves these objectives through the use of a switching horizon, denoted N_s , which contains an ordered sequence of events within each prediction - 'S' for switch, which gives the controller the freedom to change the input (switching state), and 'E' for extend, which restricts the controller to holding the input constant. During 'E' events, the output trajectories are extended until a bound is reached. Each 'E' event can

¹Alternatively, the switching losses can be minimised directly, by using a model of the semiconductor devices - see, e.g., [23].

thus occupy a different number of time-steps, depending on the exact input that is being applied, which gives rise to a prediction horizon, N_p , of variable length. It should be noted that in order to be considered, an input must be a *candidate*. A candidate input is one for which each of the corresponding outputs is predicted to be within its bounds, or moving closer to its reference, at time-step $k+1$. The basic algorithm for the case with $N_s = \text{'SE'}$ is provided as follows, for a three-phase NPC converter:

- 1) Measure the states $\mathbf{x}(k)$ and neutral-point potential $v_n(k)$ and determine the output vector $\mathbf{y}(k)$. With $\mathbf{u}_{abc}(k) = \mathbf{u}_{abc}(k-1)$, predict $\mathbf{x}(k+1)$, $v_n(k+1)$ and $\mathbf{y}(k+1)$ using the model of the system.
- 2) If each component of the output $\mathbf{y}(k+1)$ is within its bounds, or moving closer to its reference, then set $\mathbf{u}_{abc}(k) = \mathbf{u}_{abc}(k-1)$ and skip the remaining Steps. Else, proceed to Step 3.
- 3) Determine the set of inputs that can be applied at time-step k without violating the converter's switching restrictions, as discussed for the NPC converter in Section II. Denote the inputs as $\mathbf{u}_{abc}^j(k)$, $j \in \mathcal{J}$, where \mathcal{J} is an index set.
- 4) For each input $\mathbf{u}_{abc}^j(k)$, predict $\mathbf{x}^j(k+1)$, $v_n^j(k+1)$ and $\mathbf{y}^j(k+1)$. This Step corresponds to the 'S' within the switching horizon. Disregard the inputs that are not candidates. Denote the candidate inputs as $\mathbf{u}_{abc}^j(k)$, $j \in \mathcal{J}_c \subseteq \mathcal{J}$, where \mathcal{J}_c contains the indices of all candidate inputs.
- 5) For each candidate input, extend the trajectories of each output forward in time until the first output variable reaches its bounds. This Step corresponds to the 'E' within the switching horizon, and gives the prediction horizon, N_p^j , corresponding to each candidate input.
- 6) For each candidate input, compute the cost, c^j . For minimisation of switching frequency, the cost is given by [18]

$$c^j = \frac{\|\Delta \mathbf{u}_{abc}^j(k)\|_1}{N_p^j}. \quad (19)$$

where $\Delta \mathbf{u}_{abc}^j(k) = \mathbf{u}_{abc}^j(k) - \mathbf{u}_{abc}(k-1)$.

- 7) Determine the index of the candidate input with the lowest cost

$$j = \arg \min_{j \in \mathcal{J}_c} c^j. \quad (20)$$

and apply the corresponding input to the converter, i.e. $\mathbf{u}_{abc}(k) \triangleq \mathbf{u}_{abc}^j(k)$.

At time-step $k+1$, the algorithm is repeated with an updated measurement of the system states, thus closing the loop and enforcing a receding-horizon style control policy. The key features of the MPDxC strategy are the extension step, summarised in Step 5 of the algorithm, and the formulation of the cost function, as per (19), and at this point both of these are discussed in further detail, as they are important points in the development of the MPDSC strategy.

The extension step can be implemented using a number of different techniques - including iterative extension using the system model, linear interpolation or extrapolation, and quadratic interpolation or extrapolation [25]. The simplest of

these strategies is linear extrapolation, having been used in several previous implementations of MPDxC. Using linear extrapolation, the number of steps taken for the h -th component of the output vector to reach its bounds is given by

$$N_h^j = \frac{y_h^*(k) \pm \delta_h - y_h(k)}{y_h^j(k+1) - y_h(k)} \quad (21)$$

where δ_h is half the total bound width for reference y_h^* , and where the ' \pm ' sign in (21) is replaced with a '+' if $y_h^j(k+1) > y_h(k)$, and is replaced with a '-' otherwise. For an output vector with h components, the overall prediction horizon is given by

$$N_p^j = \min\{N_1^j, N_2^j, \dots, N_h^j\} \quad (22)$$

The cost function, given in (19), is minimised by the controller, as per Step 7 of the algorithm. It is obvious that the cost function encapsulates the second control objective, which is minimisation of the switching frequency. The numerator 'penalises' inputs which require a significant number of transitions at time-step k , whilst the denominator 'rewards' inputs which regulate the outputs within their bounds over a large number of time-steps, i.e. which defer the next switching event for a relatively long time. Treating the numerator and denominator as being independent, we can say that $c^j \propto \|\Delta \mathbf{u}_{abc}^j(k)\|_1$ and $c^j \propto 1/N_p^j$.

Further details on the formulation of the MPDxC problem, and more generalised versions of the algorithm, can be found in [17]. Nonetheless, the preceding summary contains the key details that are necessary in presenting the MPDSC concept clearly, as follows.

C. Overview and Details of MPDSC

As was stated in the introduction, the MPDSC strategy emulates the performance of MPDxC with $N_s = \text{'SE'}$ by minimising the predicted *slope* of the output variables over a prediction horizon fixed at $N_p = 1$. To illustrate why this is a valid approach, referring back to the cost function for MPDxC, where the cost for a particular input is proportional to the inverse of the prediction horizon, $c^j \propto 1/N_p^j$. First considering a system with only one output component, y , and the equations relating to the prediction horizon, (21) and (22), it follows that

$$N_p^j = \frac{y^*(k) \pm \delta - y(k)}{y^j(k+1) - y(k)} \quad (23)$$

from which it can then be seen that

$$c^j \propto \frac{y^j(k+1) - y(k)}{y^*(k) \pm \delta - y(k)}. \quad (24)$$

For the single-output case, each of the variables in the denominator of (24) is fixed at each time-step k , meaning that the denominator has no impact on the *relative* cost of each input. Since the cost is always positive, (24) can then be further simplified to

$$c^j \propto |y^j(k+1) - y(k)| \quad (25)$$

i.e., the cost is proportional to the *slope* of the output variable over a prediction horizon of one time-step. This is illustrated in Fig. 2 for a hypothetical system with the same single output,

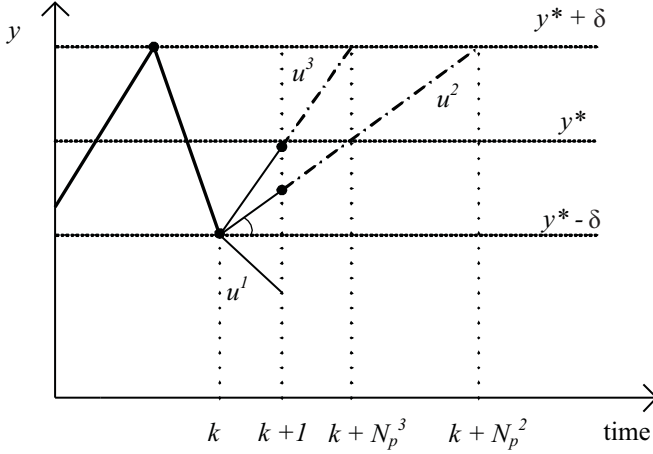


Fig. 2: Example of output predictions for MPDSC.

y , and three discrete inputs, denoted $u^1 - u^3$. At time-step k , a new input needs to be chosen in order to avoid violation of the lower output bound, $y^* - \delta$. Of the three inputs, it is clear that the input u^2 achieves the lowest slope, and is thus the optimal input with respect to slope minimisation. Note also that, when extrapolated forwards, u^2 achieves the longest prediction horizon. Finally, notice that for the situation illustrated in Fig. 2, the input u^1 is not a candidate, as it is unable to keep the output within its bounds at the next time-step.

D. Algorithm

This section presents the generalised MPDSC algorithm. In order to facilitate the explanation, some additional notation is defined. For a system with h outputs, the output error vector, ϵ , is defined at time-step k as²

$$\epsilon(k) = [\epsilon_1(k) \dots \epsilon_h(k)]^T = \mathbf{y}^*(k) - \mathbf{y}(k) \quad (26)$$

The normalised output error vector, $\bar{\epsilon}$, is defined as

$$\bar{\epsilon} = [\bar{\epsilon}_1 \dots \bar{\epsilon}_h]^T = \text{diag}\left(\frac{1}{\delta_1}, \frac{1}{\delta_2}, \dots, \frac{1}{\delta_h}\right) \cdot \epsilon, \quad (27)$$

where δ_h is half of the bound width set around the h^{th} output. The slope of the normalised output error vector between time-steps k and $k+1$, $\Delta\bar{\epsilon}(k)$, is given by

$$\Delta\bar{\epsilon} = [\Delta\bar{\epsilon}_1 \dots \Delta\bar{\epsilon}_h]^T = \bar{\epsilon}(k+1) - \bar{\epsilon}(k). \quad (28)$$

Finally, the set of inputs that can be applied at each time-step, without violating the converter's switching constraints, is denoted as $\mathcal{U}(k)$, and each individual input is denoted as $\mathbf{u}_{abc}^j(k) \in \mathcal{U}(k)$, where $j \in \mathcal{J}$ and \mathcal{J} is an index set. The MPDSC algorithm can be stated as follows.

- 1) Make the appropriate measurements of the system and determine the output vector $\mathbf{y}(k)$. With $\mathbf{u}_{abc}(k) = \mathbf{u}_{abc}(k-1)$, predict $\mathbf{y}(k+1)$ using the model of the system. Compute $\epsilon(k)$ and $\epsilon(k+1)$.

- 2) If each of the component of the output error vector $\epsilon(k+1)$ is within its bounds, or moving closer to zero (i.e. the output is moving closer to its reference), then set $\mathbf{u}_{abc}(k) = \mathbf{u}_{abc}(k-1)$ and skip the remaining Steps. Else, proceed to Step 3.
- 3) Determine the set inputs that can be applied at time-step k , $\mathcal{U}(k)$, without violating the converter's switching restrictions.
- 4) Using the model of the system, predict $\mathbf{y}^j(k+1)$, and $\epsilon^j(k+1)$, for each input $\mathbf{u}_{abc}^j(k) \in \mathcal{U}(k)$.
- 5) Compute the cost, c^j , for each input. If each of the component of the output error vector $\epsilon(k+1)$ is within its bounds, or moving closer to zero, the input is said to be a candidate, and the cost is given by

$$c^j = \|\Delta\bar{\epsilon}^j\|_2^2 + \lambda \|\Delta\mathbf{u}_{abc}^j(k)\|_1. \quad (29)$$

Conversely, if the input is a non-candidate, the cost is given by

$$c^j = \|\bar{\epsilon}^j(k+1)\|_\infty + \gamma \quad (30)$$

where γ is an arbitrarily large positive constant.

- 6) Determine the index of the input with the lowest cost, j , using the same equation as in (20), and apply the corresponding input to the converter, i.e. $\mathbf{u}_{abc}^j(k) \triangleq \mathbf{u}_{abc}^j(k)$.

At time-step $k+1$, the algorithm is repeated with an updated measurement of the system states, thus closing the loop and enforcing a receding-horizon style control policy.

It can be seen that the basic structure of the MPDSC algorithm is close to that of MPDxC. The main differences are the lack of an extrapolation step, and the resulting formulation of the cost function, (29). It is important to observe that, for the multi-output case, (29) does not account for the *sign* of the output error slopes. If each of the outputs are at one of their bounds, this is not significant, due to the definition of a candidate input. In general, however, one or more of the outputs may be well-inside its bounds when another output is at its bound. When this is the case, the definition of a candidate input means that the signs of the slopes are not considered for the outputs that are inside their bounds. Although this is a sub-optimal policy when compared to MPDxC, where the sign and magnitude of each output slope is always (implicitly) considered, this heuristic simplification eliminates the need for division operations, which are very costly in the context of Digital Signal Processor- (DSP-) based implementations. At the same time, the slope-based cost function provides a good approximation of the MPDxC control objective - to achieve output regulation with the minimum switching effort, and it is shown in the coming sections that with the specific cost function that has been presented, MPDSC offers a very similar level of performance to MPDxC. In the case of an MPDxC, the division operation can also be replaced with a much simpler shift operation by limiting the number of switching transitions to either one or two. This, however, limits the number of possible switch positions being considered at each time step.

The formulation of the cost for non-candidate inputs, which is based on the infinity-norm of the output error vector, is similar to that of MPDxC, e.g. [16]. One should note, however, that the proposed algorithm does not explicitly determine when

²In the preceding discussions, it was assumed that the output references, y^* , are constant. In general, however, this is not true, e.g. for current control. By formulating the algorithm in terms of output error, rather than the nominal error, it is applicable to scenarios with both constant and sinusoidal references.

a ‘deadlock’ has occurred (the situation where no candidate inputs exist), as in MPDxC [16], [26]. Instead, deadlocks are implicitly handled via the formulation of (30), whereby the addition of a very large positive constant to the cost of non-candidate inputs ensures that they will only be considered when no candidate input exists. This has the benefit of further simplifying the implementation of MPDSC.

Although MPDSC is based on and emulates the control objective of MPDxC, the use of a prediction horizon that is strictly limited to one time-step is an obvious similarity to most formulations of FCS-MPC, e.g. [5]. Additionally, the second term of the cost function (29), which imposes a penalty on switching transitions, is similar to that of many FCS-MPC formulations. A consequence of this is an additional degree of freedom in tuning MPDSC, relative to MPDxC, as the widths of the output bounds, as well as the weighting factor in (29), need to be selected such that the converter operates in the desired manner.

It should finally be noted that the MPDSC concept is applicable to a wide range of scenarios. As long as the model of the system is tailored correctly, MPDSC can be applied to two-level and multi-level converters, where internal voltages need to be treated as components of the output vector, and to different applications, including machine drives and grid-connected converters.

IV. SIMULATIONS

In this section, MATLAB simulation results are provided to validate the performance of the MPDSC strategy. In order to demonstrate that the concept is adaptable to different scenarios, results of both the IM drive and grid-connected converter case studies that were outlined in Section II are presented. The parameters and ratings for both case studies are provided in Appendix B. In order to benchmark the MPDSC strategy, its performance is compared against that of the conventional MPDTC and MPDCC, for the drive and grid case studies, respectively. The MPDTC strategy is implemented according to that outlined in [23], whilst the MPDCC strategy is implemented according to that outlined in [24]. For all strategies, the sampling time is $T_s = 100 \mu s$. It should be noted that the simulation results ignore the non-ideal behaviour, such as measurement delay, controller delay, and model mismatch.

A. Induction Machine Drive: Torque and Flux Control

For the IM drive, both MPDSC and MPDTC are formulated to regulate the electromagnetic torque, stator flux magnitude, and neutral-point potential, as per (17). For both strategies, bounds denoted δ_{T_e} , $\delta_{|\Psi_s|}$, and δ_{v_n} , are defined about the references of $T_e^* = 1$ p.u., $|\Psi_s|^* = 1$ p.u., and $v_n^* \triangleq 0$ p.u., respectively. The rotor speed is $\omega_r = 1$ p.u. Figs. 3 and 4 show the regulation of the electromagnetic torque and stator flux magnitude over 20 ms (one fundamental period), with $\delta_{T_e} = 0.064$ p.u., $\delta_{|\Psi_s|} = 0.02$ p.u., and $\delta_{v_n} = 0.03$ p.u. For MPDSC, the weighting factor, λ , in the cost function is 1. It is clear from Fig. 3 that the regulation of the electromagnetic torque is similar for both MPDSC and MPDTC. Likewise, from Fig. 4, it can be seen that the regulation of the stator

flux magnitude is also similar. Fig. 5 shows that the stator currents are similar for both MPDSC and MPDTC, with the two strategies achieving Total Demand Distortion (TDD) values of 5.28% and 5.19%, respectively. Additionally, the switching frequencies are very similar, at 398 Hz and 388 Hz, respectively.

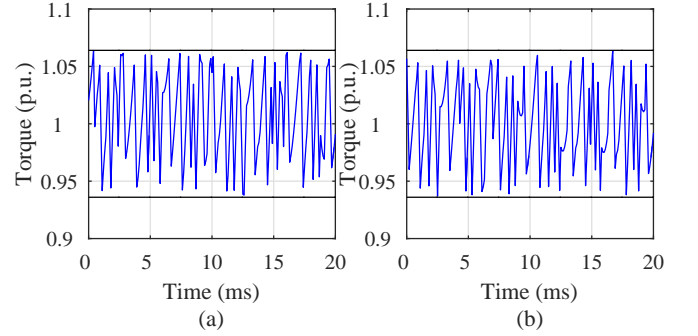


Fig. 3: Electromagnetic torque for (a) MPDSC and (b) MPDTC.

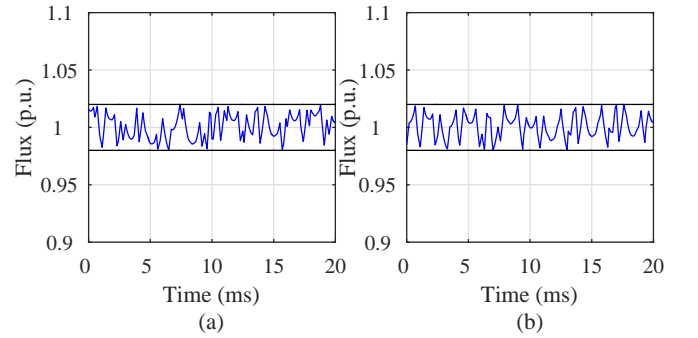


Fig. 4: Stator flux magnitude for (a) MPDSC and (b) MPDTC.

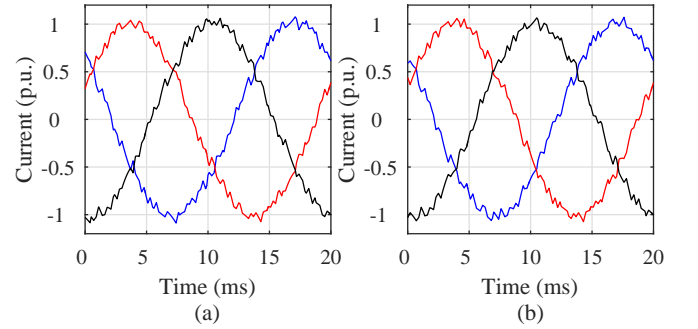


Fig. 5: Stator currents for (a) MPDSC and (b) MPDTC.

In order to further demonstrate that the performance of MPDSC is very close to that of MPDTC, a Pareto analysis has been performed. For both strategies, simulations were run with the values of δ_{T_e} and $\delta_{|\Psi_s|}$ concurrently varied on the interval $[0.02, 0.10]$ p.u. Fig. 6 shows the set of data points that emerge for MPDSC at $\omega_r = 1.0$ p.u., where the key performance metrics of stator current TDD and average device switching frequency are shown on the y - and x -axes, respectively. The Pareto front, which corresponds to the lower envelope of the data points, can be thought of as showing the best possible value of stator current TDD as the average device switching frequency varies.

Fig. 7(a) shows the Pareto fronts for MPDSC and MPDTC that emerge when $\omega_r = 0.6$ p.u. It is clear that there is almost

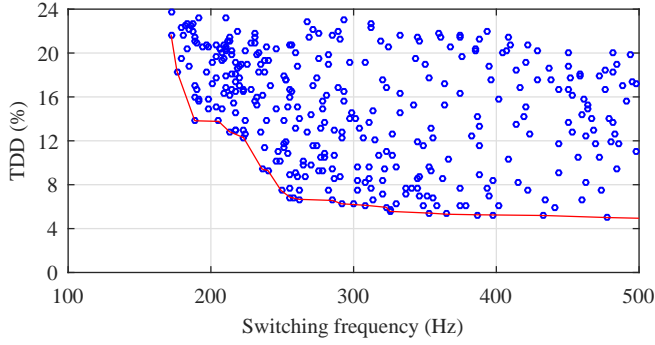


Fig. 6: Data points and Pareto front for MPDSC.

no difference between the curves, indicating that both MPDSC and MPDTC achieve an almost identical level of performance. Fig. 7(b) shows the same with $w_r = 1.0$ p.u. Again, the two curves are very similar in shape and value, with the only significant difference occurring between 180 and 220 Hz, where MPDTC achieved a marginally lower level of TDD. However, this is a relatively narrow band of frequencies, and at the other points the performance is again nearly identical. This validates the applicability of MPDSC to the control of torque and flux in an IM, with the performance being very similar to MPDTC at both $w_r = 0.6$ p.u. and 1.0 p.u.

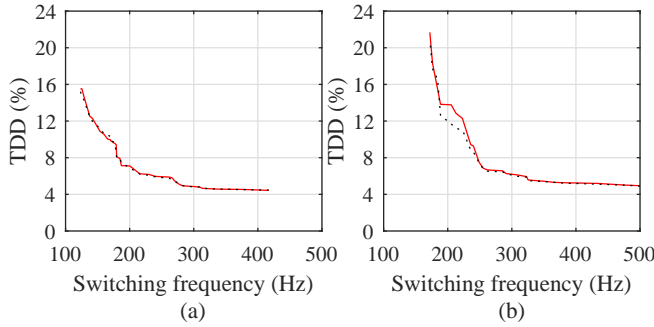


Fig. 7: Pareto fronts for MPDSC (red, solid) and MPDTC (black, dashed) at (a) $w_r = 0.6$ p.u. and (b) $w_r = 1.0$ p.u.

B. Grid-Connected Converter: Current Control

For the grid-connected converter, MPDSC and MPDCC are formulated to regulate the $\alpha\beta$ -frame grid currents, and neutral-point potential, as per (18). For both strategies, bounds denoted $\delta_\alpha = \delta_\beta \triangleq \delta_i$, and δ_{v_n} , are defined about the references, i_α^* , i_β^* , and $v_n^* \triangleq 0$ p.u., respectively. The instantaneous values of i_α^* and i_β^* are set according to the real and reactive power requirements. Figs. 8 and 9 show the regulation of the $\alpha\beta$ -frame grid currents, and the real and reactive power, over 20 ms (one fundamental period), with $\delta_i = 0.072$ p.u., and $\delta_{v_n} = 0.03$ p.u. For MPDSC, the weighting factor in the cost function is again $\lambda = 1$.

Fig. 8(a) shows that MPDSC regulates the output currents within their bounds, and comparison with Fig. 8(b) shows that the waveforms are very similar to those of MPDCC. It follows that the real and reactive power are properly regulated by MPDSC, as shown Fig. 9(a), again achieving similar waveforms to MPDCC, which are shown in Fig. 9(b). The similarity of the waveforms leads to almost identical levels of harmonic content in the currents, with MPDSC and MPDCC

TDD values of 5.15% and 5.16%, respectively. Again, the switching frequencies are very similar, at 367 and 370 Hz, respectively.

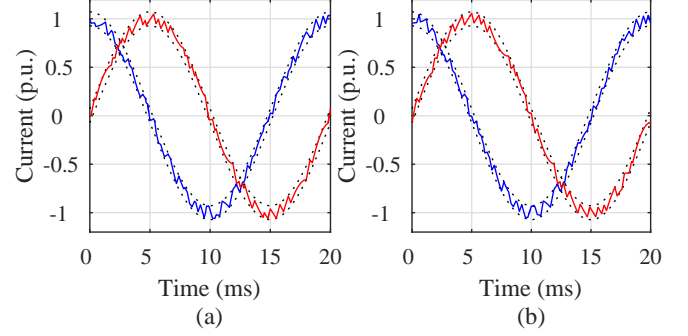


Fig. 8: Grid currents in the $\alpha\beta$ -frame for (a) MPDSC and (b) MPDCC.

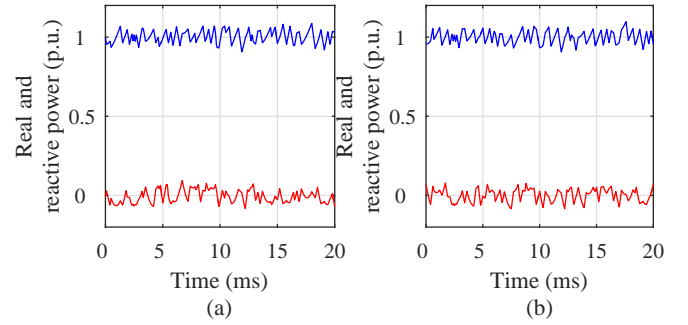


Fig. 9: Real and reactive power delivery for (a) MPDSC and (b) MPDCC.

As with the case of torque control, MPDSC and MPDCC are further compared through a Pareto analysis, with the value of δ_i varied on the interval $[0.06, 0.20]$ p.u.. As shown in Fig. 10, the Pareto fronts are similar, with very minor deviation between the two curves being visible. This validates the applicability of MPDSC to the control of currents for a grid-connected converter, with the best-case performance being near-identical for MPDSC and MPDCC.

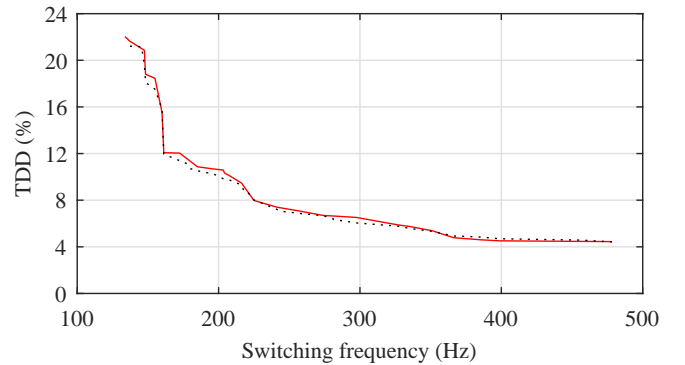


Fig. 10: Pareto fronts for MPDSC (red, solid) and MPDCC (black, dashed).

C. Tuning

For the results shown in Figs. 3 - 10, the tuning parameter λ was fixed at 1. However, it is important to comment on the effect that different values of λ , which penalises switching transitions within the cost function, can have on the switching frequency of the converter. From the formulation of the cost

function, as shown in (29), it can be seen that for very low values of λ the cost function only minimises the slope of the outputs, with the number of switching transitions, $\|\Delta \mathbf{u}_{abc}^j(k)\|_1$, having little influence on the cost. Conversely, for very high values of λ , the number of switching transitions ‘dominates’ the cost function, with the slope only being significant in separating switching states which require the same number of transitions from the previous switching state. It is important to note that the value of λ does not have much impact on the current TDD, which is almost entirely determined by the width of the output bounds.

Fig. 11 shows the switching frequency against λ for the grid-connected converter case study with different values of δ_i . Fig. 11 shows how the switching frequency changes with $\delta_i = 0.05$ p.u., from which it is clear that the trend can be split into two portions. As λ increases from 0 - 0.75, there is a downward trend in switching frequency, from around 740 to 630 Hz. Then, for $\lambda \geq 1$, the switching frequency remains level at slightly over 630 Hz, indicating that increasing the value of λ beyond 1 has no effect on the choice of optimal switching state at each time-step. Fig. 11 also shows how the switching frequency changes with $\delta_i = 0.10$ p.u.. The overall trend is similar to the previous example, with the switching frequency initially decreasing, before remaining level at 240 Hz for values of $\lambda \geq 0.5$. Finally, Fig. 11 shows how the switching frequency changes with $\delta_i = 0.15$ p.u.. The same overall trend is seen again, with the switching frequency initially decreasing before remaining level at 195 Hz for values of $\lambda \geq 0.75$. From $\lambda = 0.25$ - 0.5, the switching frequency is in fact slightly lower than for the higher values of λ , but the difference is only a couple of Hz and is therefore of minimal significance.

Broadly speaking, the results indicate that the best (or near-best) performance is achieved with higher values of λ , and for the different bound widths considered in Fig. 11, this performance is guaranteed for $\lambda \geq 1$. This justifies the choice of $\lambda = 1$ that was used throughout the simulation results, and which is carried into the experimental results.

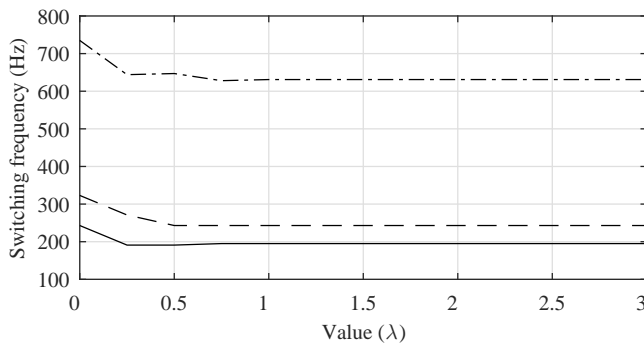


Fig. 11: Effect of the tuning parameter λ with different values of δ_i - 0.05 p.u. (dashed-dotted line), 0.10 p.u. (dashed line), 0.15 p.u. (solid line).

V. EXPERIMENTAL RESULTS

In order to verify the viability of the proposed MPDSC scheme, experimental results have been obtained for both MPDSC and MPDCC using a 1.68 kVA, 230 V laboratory-scale version of the MV grid-connected converter that was used in Section IV-B. The ratings and parameters of the

system can be found in Table II in Appendix B. It should be noted that the experimental system preserves the p.u. values of the MV case study, allowing direct comparison between simulated and experimental results. Both the MPDSC and MPDCC algorithms were implemented on a TMS320F28335 DSP, with the on-board Analog-to-Digital Converter (ADC) being used to measure all of the currents and voltages. Dead-time and gating signal generation were handled by an Altera DE2 FPGA. A delay compensation strategy, as outlined in [27], was included to alleviate the issues associated with the run-time of the algorithms. As in simulation, a sampling interval of $T_s = 100 \mu s$ was used.

Because the MPDSC algorithm does not extrapolate the output trajectories, as with MPDCC, the number of computations required to execute the algorithm is significantly lower than that of the MPDCC. For the selected control platform, the execution times of the MPDSC and MPDCC algorithms were 24.5 μs and 78.0 μs , respectively. Relative to MPDCC, the proposed MPDSC scheme was thus able to achieve a reduction in execution time of 68%. For both algorithms, a fixed sampling interval $T_s = 100 \mu s$ was used, enabling the performance of the strategies to be compared directly.

A. MPDSC Waveforms

1) *Steady-State*: The regulation of the output currents and neutral-point voltage are shown over one fundamental period in Fig. 12(a) and Fig. 12(b), where bounds of $\delta_i = 0.06$ p.u. and $\delta_{v_n} = 0.03$ p.u. were used. The resulting TDD, when averaged over 1 s, was measured as 5.15 %. It should be noted that this is very similar to the 5.16% that was achieved in simulation with $\delta_i = 0.072$ p.u.; the narrower bounds that are used experimentally are necessary to account for measurement noise and sensor delay, which cause the outputs to sometimes deviate from their nominal bounds.

The switching state of the converter is shown in Fig. 12(c) over one fundamental period. It can be seen that the switching pattern exhibits the irregular distribution of pulses that is typically seen in both MPDxC and FCS-MPC. When averaged over 1 s, the average device switching frequency was measured as 440 Hz. The grid current spectrum is shown in Fig. 12(d). Due to the nature of the switching pattern, the spectral content is spread over a wide range of frequencies, with the only notable peak being at the 11th harmonic.

Fig. 12(e) and Fig. 12(f) show that the real and reactive powers are properly regulated to their desired values of $p = 1$ p.u. and $q = 0$ p.u. The controller implicitly regulates the instantaneous values of these quantities to within 0.1 p.u. of their desired values.

2) *Transient*: The transient response of MPDSC is demonstrated in Figs. 13 and 14. Both output bound widths, $\delta_i = 0.06$ p.u. and $\delta_{v_n} = 0.03$ p.u., were left the same as in steady-state evaluation. At $t = 20$ ms, the current references were changed from their nominal values to 0 p.u. Fig. 13(a) shows that MPDSC achieves a fast transient response, taking around 2 ms to track the new references. Fig. 13(b) shows that the abc -frame currents, which are regulated indirectly, respond accordingly. At $t = 40$ ms, the current references are shifted

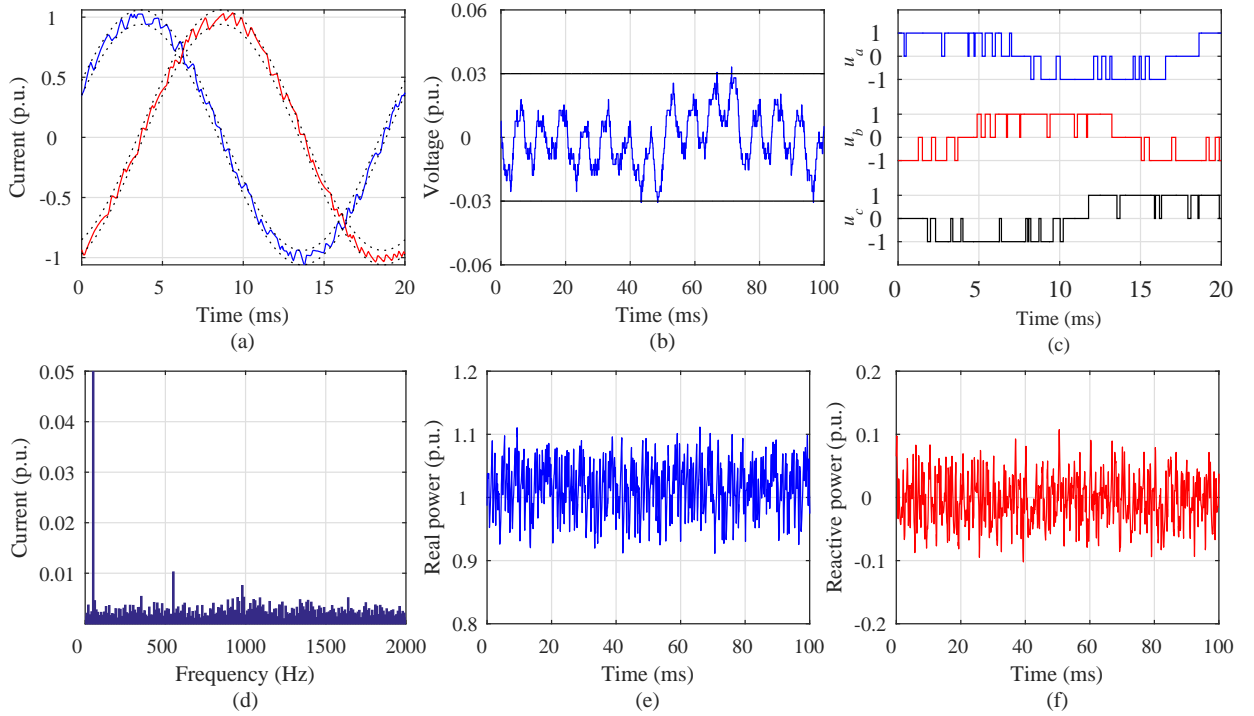


Fig. 12: Experimental waveforms of the (a) grid currents in the $\alpha\beta$ -frame, (b) neutral-point voltage, (c) switch inputs, (d) grid current spectrum, (e) real power, and (f) reactive power during steady-state conditions.

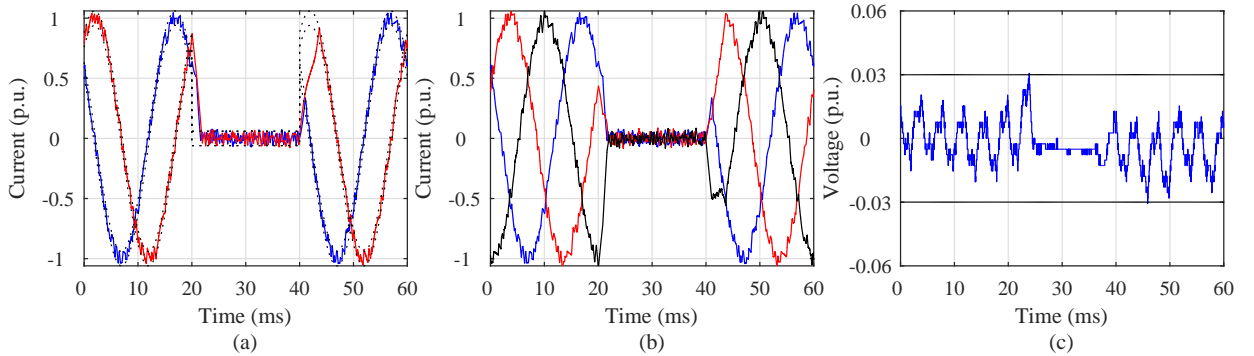


Fig. 13: Experimental waveforms of the (a) grid currents in the $\alpha\beta$ -frame, (b) grid currents in the abc -frame, and (c) neutral-point voltage during transient conditions.

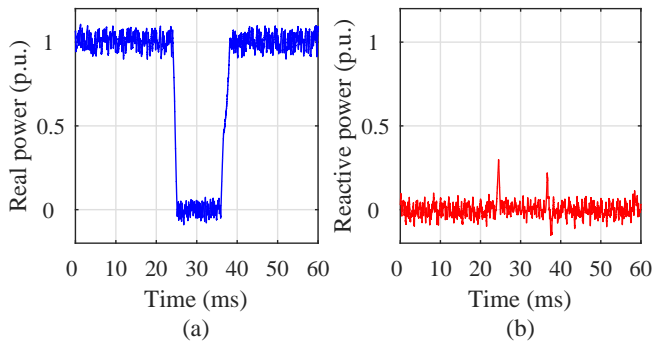


Fig. 14: Experimental waveforms of the (a) real power, and (b) reactive power during transient conditions.

back to their nominal values, with the controller exhibiting a similar performance. Fig. 13(c) demonstrates that the neutral-point voltage is maintained within its bounds during both transients. Fig. 14(a) shows that the real power responds in

accordance with the currents, with the response being free of overshoot or ringing. As shown in Fig. 14(b), minor spikes in the reactive power are exhibited during the transients, but these are quickly corrected.

B. Steady-State Performance Comparison

As a final demonstration of the efficacy of the proposed MPDSC strategy, the steady-state performance of MPDSC is compared against that of MPDCC. Fig. 15 compares the switching frequency and grid current TDD achieved by the two schemes, with the bound width δ_i varied on the interval $[0.06, 0.12]$ p.u. Fig. 15(a) shows the switching frequencies. It can be seen that the switching frequency of MPDSC is marginally higher than that of MPDCC over the range of points considered, with both exhibiting a similar trend. The discrepancy in switching frequency can be attributed to the presence of noise and delay which were not considered in simulation. A possible solution for reducing the susceptibility

of MPDSC to noise is the re-formulation of the cost function in terms of the Infinity-norm rather than 2-norm; such a min-max approach could provide an even closer approximation of MPDCC, which effectively uses a min-max approach through the formulation of (21) and (22). However, it is also worth mentioning that the much lower execution time that is achievable with MPDSC means that it has the potential to be implemented with substantially lower sampling intervals than MPDCC, the likely effect of this being an improvement in the relative performance of MPDSC, even without modification to the cost function. Fig. 15(b) shows that the grid current TDD achieved by MPDSC and MPDCC is very similar, with both exhibiting a linear trend as the bound width is varied. This is usually observed for MPDCC as long as the bounds are well utilised [19], [28], and further demonstrates that the performance of MPDSC approximates that of MPDCC.

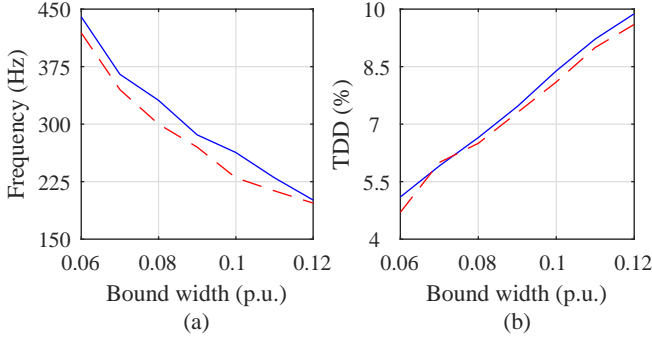


Fig. 15: (a) Average switching frequency and (b) TDD of the grid current against δ_i . MPDSC and MPDCC are shown as solid and dashed lines, respectively.

VI. CONCLUSIONS

This paper has presented a Model Predictive Direct Slope Control (MPDSC) strategy as an alternative to the established MPDxC control strategy. By controlling the predicted slope of the output variables, MPDSC is able to emulate the performance of long-horizon MPDxC strategy with a prediction horizon of only one time-step. The key benefit of the proposed approach is the ability to approximate the high level of performance of MPDxC with a significantly lower computational burden. The MPDSC algorithm has been presented, and its applicability to both a MV induction machine drive and grid-connected converter has been demonstrated through simulation. The results demonstrate that the performance approximates that of MPDxC. The performance of MPDSC was further demonstrated experimentally, with its steady-state performance compared against that of MPDCC using a down-scaled grid-connected NPC converter. It was shown that MPDSC is able to achieve a similar level of experimental performance to MPDCC, at the same time having an execution time that was 68% lower for the chosen control hardware platform.

APPENDIX A

The matrices of the state and input vectors for the drive model from Section II are given as follows

$$A_D = \begin{bmatrix} -\frac{1}{\tau'_\sigma} & 0 & \frac{k_r}{r_\sigma \tau_r \tau'_\sigma} & \frac{k_r \omega_r}{r_\sigma \tau_r \tau'_\sigma} \\ 0 & -\frac{1}{\tau'_\sigma} & -\frac{k_r \omega_r}{r_\sigma \tau_r \tau'_\sigma} & \frac{k_r}{r_\sigma \tau_r \tau'_\sigma} \\ \frac{x_m}{\tau_r} & 0 & -\frac{1}{\tau_r} & -\omega_r \\ 0 & \frac{x_m}{\tau_r} & \omega_r & -\frac{1}{\tau_r} \end{bmatrix} \quad (31)$$

$$B_D = \begin{bmatrix} \frac{1}{r_\sigma \tau_\sigma'} & 0 & 0 & 0 \\ 0 & \frac{1}{r_\sigma \tau_\sigma'} & 0 & 0 \end{bmatrix}^T \quad (32)$$

where the parameters are the angular velocity of the rotor, ω_r , the resistances of the stator and rotor, r_s and r_r , respectively, the reactances of the stator and rotor, x_{ls} and x_{lr} , respectively, and the mutual reactance, x_m . The deduced parameters are the rotor coupling factor $k_r = x_m/x_r$, total leakage factor $\sigma = 1 - x_m^2/x_s x_r$, leakage reactance $x_\sigma = \sigma x_s$, where $x_s = x_{ls} + x_m$ and $x_r = x_{lr} + x_m$, and equivalent resistance $r_\sigma = r_s + k_r^2 r_r$. The deduced time constants include the transient stator time constant $\tau'_\sigma = x_\sigma/r_\sigma$, and the rotor time constant $\tau_r = x_r/r_r$.

The matrices of the state and input vectors for the grid model from Section II are given as follows

$$A_G = \begin{bmatrix} -\frac{R_g}{L_g} & 0 & -\frac{1}{L_g} & 0 \\ 0 & -\frac{R_g}{L_g} & 0 & -\frac{1}{L_g} \\ 0 & 0 & 0 & -\omega \\ 0 & 0 & \omega & 0 \end{bmatrix} \quad (33)$$

and

$$B_G = \begin{bmatrix} \frac{1}{L_g} & 0 & 0 & 0 \\ 0 & \frac{1}{L_g} & 0 & 0 \end{bmatrix}^T \quad (34)$$

where the parameters are the grid inductance, L_g , the grid resistance, R_g , and the angular frequency of the grid, $\omega = 2\pi f$, where f is the frequency of the grid.

APPENDIX B

The parameters for the IM drive and grid-connected converter case studies are given in Tables I and II, respectively. For the IM drive, the p.u. values are taken from base values of $V_B = \sqrt{2/3}V_{rat} = 2694$ V, $I_B = \sqrt{2}I_{rat} = 504$ A, and $f_B = f_{rat} = 50$ Hz. For the grid connected converter, the p.u. values are taken from base values of $V_B = \sqrt{2/3}V_{rat} = 2445$ V, $I_B = \sqrt{2}I_{rat} = 1053$ A and $f_B = f_{rat} = 50$ Hz.

TABLE I: Ratings (left) and p.u. parameters (right) of the IM drive used in simulations.

Ratings and parameters			
DC-link voltage	5.2 kV	DC-link capacitance	11.77 p.u.
Stator voltage (V_{rat})	3.3 kV	Stator resistance	0.011 p.u.
Stator current (I_{rat})	356 A	Rotor resistance	0.009 p.u.
Real power	1.59 MW	Stator reactance	0.149 p.u.
Apparent power	2.04 MVA	Rotor reactance	0.110 p.u.
Rated frequency (f_{rat})	50 Hz	Mutual reactance	2.349 p.u.
Rotational speed	596 rpm		

TABLE II: Ratings and parameters of the MV and LV grid-connected converter used in simulations and for experimental validation, respectively.

Ratings and parameters			
Quantity	MV value	LV value	p.u. value
DC-link voltage	5 kV	400 V	2.041
Grid voltage (V_{rat})	3 kV	240 V	1.225
Grid current (I_{rat})	1.29 kA	4.04 A	0.707
Grid apparent power	6.72 MVA	1.68 kVA	1.000
Grid frequency (f_{rat})	50 Hz	50 Hz	1.000
DC-link capacitance	10 mF	390 mF	4.200
Grid Inductance	1.13 mH	29 mH	0.266
Grid Resistance	20 m Ω	0.5 Ω	0.015

REFERENCES

- [1] S. J. Qin and T. A. Badgwell, "A survey of industrial model predictive control technology," *Contr. Eng. Practice*, vol. 11, no. 7, pp. 733–764, Jul. 2003.
- [2] T. Geyer, "Low complexity model predictive control in power electronics and power systems," Ph.D. dissertation, Autom. Control Lab. ETH Zurich, 2005.
- [3] J. Rodríguez, J. Pontt, C. A. Silva, P. Correa, P. Lezana, P. Cortés, and U. Ammann, "Predictive current control of a voltage source inverter," *IEEE Trans. Ind. Electron.*, vol. 54, no. 1, pp. 495–503, Feb. 2007.
- [4] P. Cortés, M. P. Kazmierkowski, R. M. Kennel, D. E. Quevedo, and J. Rodríguez, "Predictive control in power electronics and drives," *IEEE Trans. Ind. Electron.*, vol. 55, no. 12, pp. 4312–4324, Dec. 2008.
- [5] J. Rodríguez, M. Kazmierkowski, J. Espinoza, P. Zanchetta, H. Abu-Rub, H. Young, and C. Rojas, "State of the Art of Finite Control Set Model Predictive Control in Power Electronics," *IEEE Transactions on Industrial Informatics*, vol. 9, no. 2, pp. 1003–1016, May 2013.
- [6] A. Linder and R. Kennel, "Model predictive control for electrical drives," in *Proc. IEEE Power Electron. Spec. Conf.*, Recife, Brasil, 2005, pp. 1793–1799.
- [7] D. Quevedo, R. Aguilera, M. Perez, P. Cortes, and R. Lizana, "Model predictive control of an AFE rectifier with dynamic references," *IEEE Trans. Power Electron.*, vol. 27, no. 7, pp. 3128–3136, Jul. 2012.
- [8] V. Yaramasu, B. Wu, and J. Chen, "Model-Predictive Control of Grid-Tied Four-Level Diode-Clamped Inverters for High-Power Wind Energy Conversion Systems," *IEEE Trans. Power Electron.*, vol. 29, no. 6, pp. 2861–2873, Jun. 2014.
- [9] H. Miranda, P. Cortes, J. Yuz, and J. Rodriguez, "Predictive torque control of induction machines based on state-space models," *IEEE Trans. Ind. Electron.*, vol. 56, no. 6, pp. 1916–1924, Jun. 2009.
- [10] M. Preindl and S. Bolognani, "Model predictive direct speed control with finite control set of PMSM drive systems," *IEEE Trans. Power Electron.*, vol. 28, no. 2, pp. 1007–1015, Feb. 2013.
- [11] T. Geyer, "A comparison of control and modulation schemes for medium-voltage drives: Emerging predictive control concepts versus PWM-based schemes," *IEEE Trans. on Ind. Appl.*, vol. 47, no. 3, pp. 1380–1389, May. 2011.
- [12] P. Cortes, G. Ortiz, J. Yuz, J. Rodriguez, S. Vazquez, and L. Franquelo, "Model predictive control of an inverter with output LC filter for UPS applications," *IEEE Trans. Ind. Electron.*, vol. 56, no. 6, pp. 1875–1883, Jun. 2009.
- [13] T. Geyer and D. Quevedo, "Multistep finite control set model predictive control for power electronics," *IEEE Trans. Power Electron.*, vol. 29, no. 12, pp. 6836–6846, Dec. 2014.
- [14] —, "Performance of multistep finite control set model predictive control for power electronics," *IEEE Trans. Power Electron.*, vol. 30, no. 3, pp. 1633–1644, Mar. 2015.
- [15] T. Geyer, G. Papafotiou, and M. Morari, "Model Predictive Direct Torque Control – Part I: Concept, Algorithm and Analysis," *IEEE Trans. Ind. Electron.*, vol. 56, no. 6, pp. 1894–1905, Jun. 2009.
- [16] G. Papafotiou, J. Kley, K. G. Papadopoulos, P. Bohren, and M. Morari, "Model Predictive Direct Torque Control – Part II: Implementation and Experimental Evaluation," *IEEE Trans. Ind. Electron.*, vol. 56, no. 6, pp. 1906–1915, Jun. 2009.
- [17] T. Geyer, "Model Predictive Direct Current Control: Formulation of the Stator Current Bounds and the Concept of the Switching Horizon," *IEEE Ind. Appl. Mag.*, vol. 18, no. 2, pp. 47–59, Mar. 2012.
- [18] J. Scoltock, T. Geyer, and U. Madawala, "Model predictive direct power control for grid-connected neutral-point-clamped converters," *IEEE Trans. Ind. Electron.*, vol. 62, no. 9, pp. 5319 – 5328, Sept. 2015.
- [19] B. Riar, T. Geyer, and U. Madawala, "Model Predictive Direct Current Control of Modular Multilevel Converters: Modelling, Analysis and Experimental Evaluation," *IEEE Transactions on Power Electronics*, vol. 30, no. 1, pp. 431–439, Jan. 2015.
- [20] T. Geyer, "Computationally efficient model predictive direct torque control," *IEEE Trans. Power Electron.*, vol. 26, no. 10, pp. 2804–2816, Oct. 2011.
- [21] R. Fotouhi, "Steuerung eines wechselrichters für eine elektrische maschine," DE, 2014.
- [22] P. C. Krause, O. Wasynczuk, and S. D. Sudhoff, *Analysis of Electric Machinery and Drive Systems*, 2nd ed. Intersci. Publ. John Wiley & Sons Inc., 2002.
- [23] T. Geyer, "Generalized model predictive direct torque control: Long prediction horizons and minimization of switching losses," in *Proc. IEEE Conf. Decision Control*, Shanghai, China, Dec. 2009, pp. 6799–6804.
- [24] J. Scoltock, T. Geyer, and U. Madawala, "Model predictive direct current control for a grid-connected converter: LCL-filter versus L-filter," in *Industrial Technology (ICIT), 2013 IEEE International Conference on*, Cape Town, South Africa, Feb. 2013, pp. 576–581.
- [25] Y. Zeinaly, T. Geyer, and B. Egardt, "Trajectory extension methods for model predictive direct torque control," in *Proc. IEEE Appl. Power Electron. Conf. Expo.*, Fort Worth, TX, USA, 2011, pp. 1667–1674.
- [26] T. Burtscher and T. Geyer, "Deadlock avoidance in model predictive direct torque control," *IEEE Trans. Ind. Appl.*, vol. 49, no. 5, pp. 2126 – 2135, 2013.
- [27] 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.
- [28] B. S. Riar, "Design and control of modular multilevel converters," Ph.D. dissertation, The University of Auckland, New Zealand, 2015.



Baljit S. Riar (S'09 - M'15) received B.E. (Hons) and Ph.D. in electrical and electronic engineering both from The University of Auckland, New Zealand, in 2010 and 2015, respectively. He is currently a postdoctoral researcher with the Energy and Power Group, Department of Engineering Science, University of Oxford, Oxford, U.K. His research interests include power electronics, model predictive control, and inductive power transfer.



James Scoltock (S'09 - M'15) received the B.E. (Hons) and Ph.D. degrees, both in electrical and electronic engineering, from The University of Auckland, New Zealand, in 2009 and 2015, respectively. He is currently a Research Associate with the School of Electrical and Electronic Engineering, The University of Manchester, United Kingdom. His research interests include power electronics, electrical drives and model predictive control.



Udaya K. Madawala (M'93 - SM'06) received the B.Sc.(Hons.) degree in electrical engineering from the University of Moratuwa, Sri Lanka, in 1987 and the Ph.D. degree in power electronics from The University of Auckland, New Zealand, in 1993.

After working in industry, he joined The University of Auckland in 1997, where he is currently a Full Professor in the Department of Electrical and Computer Engineering. His research interests are in the fields of power electronics, inductive power transfer, and renewable energy, and he is a consultant

to industry.

Dr. Madawala is an IEEE volunteer and serves as an Associate Editor for the IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS and the IEEE TRANSACTIONS ON POWER ELECTRONICS. He is a member of the Power Electronics Technical Committee of the IEEE Industrial Electronics Society and the Sustainable Energy Systems Committee of the IEEE Power Electronics Society.