Switching Frequency Limitation with Finite Control Set Model Predictive Control via Slack Variables

Luca M. Hartmann, Orcun Karaca, Tinus Dorfling, Tobias Geyer

***Abstract*—Past work proposed an extension to finite control set model predictive control to simultaneously track both a current reference and a switching frequency reference. Such an objective can jeopardize the current tracking performance, and this can potentially be alleviated by instead limiting the switching frequency. To this end, we propose to limit the switching frequency in finite control set model predictive control. The switching frequency is captured with an infinite impulse response filter and bounded by an inequality constraint; its corresponding slack variable is penalized in the cost function. To solve the problem efficiently, a sphere decoder with a computational speed-up is presented.**

***Index Terms*—Power electronics, model predictive control, integer optimization, power conversion.**

I. Introduction

Given the developments in mathematical optimization techniques and their applications on embedded systems, model predictive control (MPC) has established itself as a promising control methodology in power conversion systems, where the system time constants are well below 1ms [1]–[5]. Several different variants of MPC have been developed for converter control. Among those, finite control set MPC (FCS-MPC) has gained a lot of attention for its advantages, such as its intuitive design, simple implementation, and high dynamic performance.

FCS-MPC achieves regulation of the states along their references by directly manipulating the switch positions of a power electronic converter. Since the switch positions are discrete, the resulting optimization problem is an integer program. Its first variants utilized a horizon of one step and an exhaustive enumeration to solve the underlying integer program [6], [7]. Whenever a linear prediction model is available, an efficient branch-and-bound algorithm called the sphere decoder can be adopted [8]–[10]. This enables the use of long horizons within short sampling intervals, which can bring performance benefits to different MPC variants, e.g., increasing the closed-loop stability margin [11]–[13].

One of the main shortcomings of FCS-MPC is the tuning of its variable switching frequency, which is strongly correlated with the switching losses of the semiconductors [14], [15]. Most commonly, the control effort is penalized with the aim to reduce the average switching frequency. The tuning of this penalty to attain a particular average switching frequency is known to be sensitive to the other control parameters and the operating point, which greatly complicates the tuning task. Without any switching penalty, on the other hand, the switching frequency would be limited only by the choice of the sampling interval [16] (see [17, Ch. 4] for an empirical study), rendering it similar to a deadbeat-type controller. To avoid the tuning process altogether, alternative MPC methods have been proposed to set the switching frequency directly by fixing the number of possible switching transitions in a given time interval, see [18]–[22] and the references therein. However, these modified controllers differ significantly from the original FCS-MPC method, and thus they are considered out of scope for this paper.

To control the switching frequency of FCS-MPC, the authors in [23] have shown that the penalty term can be replaced with a more meaningful term: a switching frequency tracking term. A second-order infinite impulse response filter can be utilized to capture the predicted switching frequency throughout the horizon. Even more importantly, this modification does not imply any major changes to the FCS-MPC formulation.

The current tracking performance generally improves with an increase in the switching frequency. Yet, this general trend does not necessarily hold for all cases, for example, there exist certain time instant, in particular during transients, where a momentary decrease in switching frequency could help improve current tracking. In such cases, when using the formulation proposed in [23], the two tracking objectives contradict each other. However, a switching frequency lower than a certain limit is in fact not a concern, since it is used as a surrogate to limit the switching losses or even the device temperatures. To this end, the main contribution of this paper is to formulate an FCS-MPC method that limits the switching frequency. This will be achieved by penalizing the slack variable corresponding to a constraint that upper bounds the switching frequency.

Solving FCS-MPC with hard or soft state constraints is in fact rather straightforward in the case of exhaustive enumeration. However, this is largely an unsolved problem when using the sphere decoder. As mentioned before, the sphere decoder is only applicable to optimization problems with linear prediction models. The prediction model for the slack variable of the switching frequency constraint has a nonlinear input-output relation. This nonlinearity is due to a max operator in its output function and an absolute value operator in its state update. One of the first attempts to include constraints (in particular, current constraints) in the sphere decoder has been made in [24], [25]. The authors have proposed an algorithm to map the state constraints to the input constraints, if the state itself has

Corresponding author: O. Karaca. L. M. Hartmann is with University of California San Diego, USA. email: lhartmann@ucsd.edu. O. Karaca and T. Dorfling are with ABB Corporate Research, Switzerland. emails: orcun.karaca@ch.abb.com and martinusdorfling@protonmail.com. T. Geyer is with ABB System Drives, Switzerland. email: t.geyer@ieee.org. The authors thank Prof. F. Dörfler of Automatic Control Laboratory for this thesis work opportunity and his feedback.

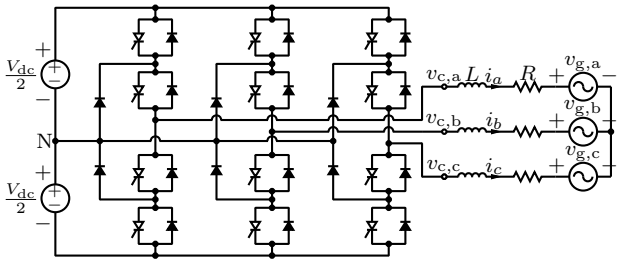


Fig. 1: Grid-connected 3L-NPC converter.

a linear prediction model. A more recent work by [26] took an entirely different perspective by working with linear Gaussian models and constraints expressed in terms of Gaussian priors. However, this approach comes with a loss in optimality. Instead, the approach in this paper will augment the input vector with the slack variables to then formulate the corresponding sphere decoding algorithm, resembling the approach in [27] that included the neutral-point dynamics in FCS-MPC. Moreover, we will provide a computational speed-up to this sphere decoder by exploiting a provable lower bound that incorporates the prediction model of the slack variable we are interested in.

Our contributions are as follows. We propose an FCS-MPC method that limits the switching frequency by penalizing slack variable of a constraint on the switching frequency. We formulate a sphere decoder that utilizes slack variables. Finally, a significant computational speed-up is achieved by exploiting a lower bound on the future cost to be incurred from the slack variables corresponding to the switching frequency constraints.

II. Preliminaries

Consider a 3-level neutral-point-clampled (3L-NPC) converter connected to the grid, as in Figure 1.[1] The half dc-link voltages $\frac{V_{\rm dc}}{2}$ are realized by ideal voltage sources, and the neutral-point-potential is fixed to zero; its balancing is out of scope, we kindly refer to [27], [28]. The grid is modeled with an RL-load in series with an infinite bus voltage. We assume all grid parameters and states are known. The grid voltage is measured or estimated, and its future values are predicted by assuming it is an ideal voltage source.

Here, the converter voltages are denoted by $\boldsymbol{v}_{\rm c} = [v_{\rm c,a}\ v_{\rm c,b}\ v_{\rm c,c}]^\top \in \{-\frac{V_{\rm dc}}{2},\ 0,\ \frac{V_{\rm dc}}{2}\}^3$. The vectors $\boldsymbol{i}_{\rm abc} = [i_{\rm a}\ i_{\rm b}\ i_{\rm c}]^\top$ and $\boldsymbol{v}_{\rm g} = [v_{\rm g,a}\ v_{\rm g,b}\ v_{\rm g,c}]^\top$ represent the current and the grid voltage, respectively. Three-phase variables in the abc-coordinates are transformed into $\boldsymbol{\xi}_{\alpha\beta}$ in the stationary $\alpha\beta$ reference frame by $\boldsymbol{\xi}_{\alpha\beta} = \boldsymbol{K}\boldsymbol{\xi}_{\rm abc}$. The inverse operation is denoted by $\boldsymbol{\xi}_{\rm abc} = \boldsymbol{K}^{-1}\boldsymbol{\xi}_{\alpha\beta}$. The matrices $\boldsymbol{K}$ and $\boldsymbol{K}^{-1}$ are defined by the Clarke transformation and its pseudo-inverse, $\boldsymbol{K} = \frac{2}{3}\begin{bmatrix}1 & -\frac{1}{2} & -\frac{1}{2}\\ 0 & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2}\end{bmatrix}$, and $\boldsymbol{K}^{-1} = \frac{3}{2}\boldsymbol{K}^\top$. The subscript $\alpha\beta$ is dropped from $\boldsymbol{\xi}_{\alpha\beta}(k)$, unless stated otherwise.

A. Physical system model

The continuous-time current dynamics for Figure 1 are:

$$\frac{d\boldsymbol{i}_{\alpha\beta}(t)}{dt} = -\frac{R}{L}\boldsymbol{I}_2\boldsymbol{i}_{\alpha\beta}(t) + \frac{1}{L}\boldsymbol{K}\boldsymbol{v}_{\rm c,abc}(t) - \frac{1}{L}\boldsymbol{K}\boldsymbol{v}_{\rm g}(t),$$

[1]The control methods under consideration are applicable to any system governed by linearized dynamics (e.g., induction machines as in [23], with or without output filtering stages [25]).

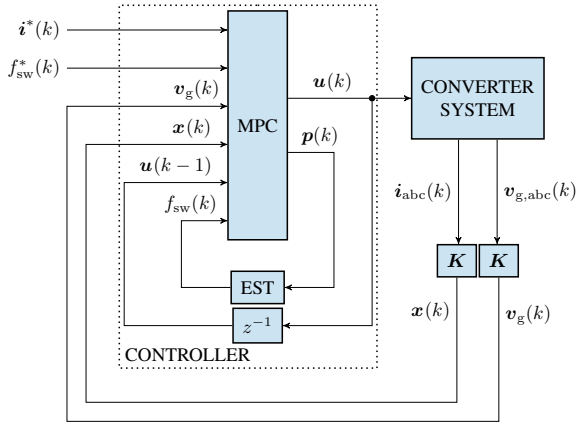


Fig. 2: Block diagram of FCS-MPC with switching frequency estimation (EST).

with the 2×2 identity matrix $\boldsymbol{I}_2$, the converter voltage $\boldsymbol{v}_{\rm c,abc}(t) = \frac{V_{\rm dc}}{2}\boldsymbol{u}(t)$, and the switch position $\boldsymbol{u}(t) = [u_a(t)\ u_b(t)\ u_c(t)]^\top \in \{-1, 0, 1\}^3$. Using exact Euler discretization, the discrete-time model becomes

$$\begin{aligned}\boldsymbol{x}(k+1) &= \boldsymbol{A}\boldsymbol{x}(k) + \boldsymbol{B}\boldsymbol{u}(k) + \boldsymbol{D}\boldsymbol{v}_{\rm g}(k),\\ \boldsymbol{y}(k) &= \boldsymbol{C}\boldsymbol{x}(k),\end{aligned} \tag{1}$$

where $\boldsymbol{x} = \boldsymbol{i}_{\alpha\beta}$ is the state, $\boldsymbol{y}$ the output, $T_{\rm s}$ the control sampling interval, and $k \in \mathbb{N}$ the discrete-time index.

B. Switching frequency estimation

The switching frequency at time step k, $f_{\rm sw}(k)$, is defined as the number of on- or off-transitions averaged over the number of semiconductor devices and divided by the sampling interval. As discussed in [23], a discrete-time second-order infinite impulse response (IIR) filter can be utilized to estimate the device switching frequency:

$$\begin{aligned}\boldsymbol{x}_{\rm sw}(k+1) &= \underbrace{\begin{bmatrix}a_1 & 1-a_1\\ 0 & a_2\end{bmatrix}^\top}_{\boldsymbol{A}_{\rm sw}}\boldsymbol{x}_{\rm sw}(k) + \underbrace{\frac{1-a_2}{12T_s}\begin{bmatrix}1 & 0\\ 1 & 0\\ 1 & 0\end{bmatrix}^\top}_{\boldsymbol{B}_{\rm sw}}\boldsymbol{p}(k),\\ y_{\rm sw}(k) &= f_{\rm sw}(k) = [0\ 1]\boldsymbol{x}_{\rm sw}(k) = \boldsymbol{C}_{\rm sw}\boldsymbol{x}_{\rm sw}(k).\end{aligned} \tag{2}$$

Here, $a_1, a_2 \in [0, 1]$ are the tuning parameters, and $\boldsymbol{p}(k) = |\Delta\boldsymbol{u}(k)|$ is the element-wise absolute value of the three-phase switching transition $\Delta\boldsymbol{u}(k) = \boldsymbol{u}(k) - \boldsymbol{u}(k-1)$. The filtering window increases and the bandwidth reduces as a_1 and a_2 are set closer to 1, see [29].

C. FCS-MPC for $f_{\rm sw}$ tracking

The switching frequency tracking FCS-MPC (FT-MPC) of [23] is an MPC algorithm with a prediction horizon of length $N_{\rm p} \in \mathbb{N}$, following the block diagram representation in Figure 2. The FT-MPC differs from the conventional FCS-MPC in [9] by tracking both the current reference and the switching frequency reference.

Define an augmented system with state $\boldsymbol{x}_{\rm T}(k) = [\boldsymbol{x}(k)^\top\ \boldsymbol{x}_{\rm sw}(k)^\top]^\top$, input $\boldsymbol{u}_{\rm T}(k) = [\boldsymbol{u}(k)^\top\ \boldsymbol{p}(k)^\top]^\top$, and output $\boldsymbol{y}_{\rm T}(k) = [\boldsymbol{y}(k)^\top\ y_{\rm sw}(k)]^\top$. The resulting dynamics are

$\boldsymbol{x}_{\mathrm{T}}(k+1) = \boldsymbol{A}_{\mathrm{T}}\boldsymbol{x}_{\mathrm{T}}(k)+\boldsymbol{B}_{\mathrm{T}}\boldsymbol{u}_{\mathrm{T}}(k)+\boldsymbol{D}_{\mathrm{T}}\boldsymbol{v}_g(k)$, $\boldsymbol{y}_{\mathrm{T}}(k+1) = \boldsymbol{C}_{\mathrm{T}}\boldsymbol{x}_{\mathrm{T}}(k+1)$, where $\boldsymbol{A}_{\mathrm{T}}$, $\boldsymbol{B}_{\mathrm{T}}$, $\boldsymbol{C}_{\mathrm{T}}$, and $\boldsymbol{D}_{\mathrm{T}}$ are defined according to (1) and (2). In the remainder, for any vector $\boldsymbol{\xi}(k)$, we also let $\boldsymbol{\Xi}(k) = [\boldsymbol{\xi}(k)^\top\ \boldsymbol{\xi}(k+1)^\top\ \dots\ \boldsymbol{\xi}(k+N_{\mathrm{p}}-1)^\top]^\top$ denote its full-horizon vector for the horizon N_{p}.

FT-MPC solves the following optimization problem at each time step k:

$$\begin{aligned}\min\ & \sum_{\ell=k}^{k+N_{\mathrm{p}}-1} ||\boldsymbol{y}^*_{\mathrm{T}}(\ell+1)-\boldsymbol{y}_{\mathrm{T}}(\ell+1)||^2_{\boldsymbol{Q}_{\mathrm{T}}}+\lambda_{\mathrm{u}}||\Delta\boldsymbol{u}(\ell)||_2^2\\ \text{s.t. }& \boldsymbol{x}_{\mathrm{T}}(\ell+1) = \boldsymbol{A}_{\mathrm{T}}\boldsymbol{x}_{\mathrm{T}}(\ell)+\boldsymbol{B}_{\mathrm{T}}\boldsymbol{u}_{\mathrm{T}}(\ell)+\boldsymbol{D}_{\mathrm{T}}\boldsymbol{v}_g(\ell), \forall \ell,\\ & \boldsymbol{y}_{\mathrm{T}}(\ell+1) = \boldsymbol{C}_{\mathrm{T}}\boldsymbol{x}_{\mathrm{T}}(\ell+1), \forall \ell,\\ & \boldsymbol{u}(\ell) \in \{-1,\ 0,\ 1\}^3, \forall \ell,\end{aligned} \tag{3}$$

where $||\boldsymbol{\Xi}||^2_{\boldsymbol{Q}} = \boldsymbol{\Xi}^\top\boldsymbol{Q}\boldsymbol{\Xi}$ refers to the Mahalanobis norm for a given $\boldsymbol{Q} \succ 0$, $\boldsymbol{\xi}^*$ denotes the reference of a vector $\boldsymbol{\xi}$, and $\ell \in \mathbb{N}$ is used to iterate over the prediction horizon. We have $\boldsymbol{Q}_{\mathrm{T}} = \mathrm{diag}([1\ 1\ \lambda_{\mathrm{sw}}])$, which uses the relative switching frequency penalty $\lambda_{\mathrm{sw}} > 0$ as a trade off between current and switching frequency tracking. The switching penalty $\lambda_{\mathrm{u}} > 0$ is set to be very small to not interfere with the switching frequency tracking. It is used to guarantee the positive-definiteness of the Hessian, shown explicitly in Appendix A. After solving (3) for its optimal solution $\boldsymbol{U}_{\mathrm{T}}^{\mathrm{opt}}(k) = [\boldsymbol{u}_{\mathrm{T}}(k)^\top\ \boldsymbol{u}_{\mathrm{T}}(k+1)^\top\ \dots\ \boldsymbol{u}_{\mathrm{T}}(k+N_{\mathrm{p}}-1)^\top]^\top$, the FT-MPC applies then the first input $\boldsymbol{u}_{\mathrm{T}}^{\mathrm{opt}}(k)$.

To solve the FT-MPC in real-time, a sphere decoder similar to [8], [27] can be deployed. The sphere decoding algorithm is a branch-and-bound technique applicable to integer least squares (ILS) problems[2] and is far more efficient than an exhaustive enumeration. The ILS reformulation of the FT-MPC and the algorithm are relegated to Appendix A. Note that these were previously not derived or provided in [23].

III. FCS-MPC for f_{sw} Limiting via Slack Variables

This section introduces switching frequency limiting FCS-MPC (FL-MPC). We introduce a switching frequency constraint with the slack variable

$$s(\ell) = \max\{f_{\mathrm{sw}}(\ell) - f^*_{\mathrm{sw}}, 0\}, \tag{4}$$

where f^*_{sw} is the upperbound on the switching frequency, and penalize the slack variable in the cost function. We then develop a sphere decoder for solving the underlying problem and discuss a significant computational speed-up by utilizing lower bounds on this new cost. The FL-MPC block diagram is identical to that of FT-MPC depicted in Figure 2.

A. Problem formulation

Similar to the $\boldsymbol{p}(k)$ variable, the slack variable $s(k)$ also has a nonlinear output relation to the switch positions, which can be inferred from the definition in (4). The nonlinearity originates from both the $\max$ operator and also the absolute value operator that is used to obtain the switching transition variable $\boldsymbol{p}(k)$ appearing in (2). Thus, it would later not be possible to obtain an ILS reformulation unless the slack variable is included in the input vector.

[2]The ILS problem is known to be NP-hard [30].

To this end, we augment the input vector as $\boldsymbol{u}_{\mathrm{S}}(k) = [\boldsymbol{u}(k)^\top\ s(k+1)]^\top$. Given $\boldsymbol{x}_{\mathrm{S}}(k) = \boldsymbol{x}(k)$ and $\boldsymbol{y}_{\mathrm{S}}(k) = \boldsymbol{y}(k)$, the new dynamics are $\boldsymbol{x}_{\mathrm{S}}(k+1) = \boldsymbol{A}_{\mathrm{S}}\boldsymbol{x}_{\mathrm{S}}(k) + \boldsymbol{B}_{\mathrm{S}}\boldsymbol{u}_{\mathrm{S}}(k) + \boldsymbol{D}_{\mathrm{S}}\boldsymbol{v}_{\mathrm{g}}(k)$ and $\boldsymbol{y}_{\mathrm{S}}(k) = \boldsymbol{C}_{\mathrm{S}}\boldsymbol{x}_{\mathrm{S}}(k)$. Here, $\boldsymbol{A}_{\mathrm{S}}$, $\boldsymbol{B}_{\mathrm{S}}$, $\boldsymbol{C}_{\mathrm{S}}$, and $\boldsymbol{D}_{\mathrm{S}}$ can be defined simply by the physical system matrices in (1).

We formulate the FL-MPC's optimization problem as

$$\begin{aligned}\min\ & \sum_{\ell=k}^{k+N_{\mathrm{p}}-1} ||\boldsymbol{y}^*_{\mathrm{S}}(\ell+1) - \boldsymbol{y}_{\mathrm{S}}(\ell+1)||^2_{\boldsymbol{Q}_{\mathrm{S}}}\\ & \qquad + \lambda_{\mathrm{sw}}||s(\ell+1)||_2^2 + \lambda_{\mathrm{u}}||\Delta\boldsymbol{u}(\ell)||_2^2\\ \text{s.t. }& \boldsymbol{x}_{\mathrm{S}}(\ell+1) = \boldsymbol{A}_{\mathrm{S}}\boldsymbol{x}_{\mathrm{S}}(\ell) + \boldsymbol{B}_{\mathrm{S}}\boldsymbol{u}_{\mathrm{S}}(\ell) + \boldsymbol{D}_{\mathrm{S}}\boldsymbol{v}_g(\ell), \forall \ell,\\ & \boldsymbol{y}_{\mathrm{S}}(\ell+1) = \boldsymbol{C}_{\mathrm{S}}\boldsymbol{x}_{\mathrm{S}}(\ell+1), \forall \ell,\\ & \boldsymbol{x}_{\mathrm{sw}}(\ell+1) = \boldsymbol{A}_{\mathrm{sw}}\boldsymbol{x}_{\mathrm{sw}}(\ell) + \boldsymbol{B}_{\mathrm{sw}}\boldsymbol{p}(\ell), \forall \ell,\\ & s(\ell+1) = \max\left\{\boldsymbol{C}_{\mathrm{sw}}\boldsymbol{x}_{\mathrm{sw}}(\ell+1) - f^*_{\mathrm{sw}},\ 0\right\}, \forall \ell,\\ & \boldsymbol{u}(\ell) \in \{-1,\ 0,\ 1\}^3, \forall \ell,\end{aligned} \tag{5}$$

where $\boldsymbol{Q}_{\mathrm{S}} = \boldsymbol{I}_2$ and $\boldsymbol{I}_n \in \mathbb{R}^{n\times n}$ denotes the identity matrix. Observe that the switching frequency penalty now relates to the penalization of the slack variable.

We reformulate (5) as an ILS problem. The full-horizon vector $\boldsymbol{Y}_{\mathrm{S}}(k+1)$ is a function of $\boldsymbol{U}_{\mathrm{S}}(k)$, $\boldsymbol{x}_{\mathrm{S}}(k)$, and $\boldsymbol{v}_{\mathrm{g}}(k)$, i.e., $\boldsymbol{Y}_{\mathrm{S}}(k+1) = \boldsymbol{\Gamma}_{\mathrm{S}}\boldsymbol{x}_{\mathrm{S}}(k) + \boldsymbol{\Upsilon}_{\mathrm{S}}\boldsymbol{U}_{\mathrm{S}}(k) + \boldsymbol{\Psi}_{\mathrm{S}}\boldsymbol{v}_{\mathrm{g}}(k)$.

Other full-horizon variables $\boldsymbol{S}(k+1)$ and $\Delta\boldsymbol{U}_{\mathrm{S}}(k)$ are defined as $\Delta\boldsymbol{U}_{\mathrm{S}}(k) = \boldsymbol{\Pi}_{\mathrm{S}}\boldsymbol{U}_{\mathrm{S}}(k) - \boldsymbol{E}_{\mathrm{S}}\boldsymbol{u}_{\mathrm{S}}(k-1)$, and $\boldsymbol{S}(k+1) = \boldsymbol{L}_{\mathrm{S}}\boldsymbol{U}_{\mathrm{S}}(k)$, where $\boldsymbol{\Gamma}_{\mathrm{S}}$, $\boldsymbol{\Upsilon}_{\mathrm{S}}$, $\boldsymbol{\Psi}_{\mathrm{S}}$, $\boldsymbol{\Pi}_{\mathrm{S}}$, $\boldsymbol{E}_{\mathrm{S}}$, $\boldsymbol{L}_{\mathrm{S}}$, and $\bar{\boldsymbol{Q}}_{\mathrm{S}}$ are defined in Appendix B following the procedure originally described in [8].

The objective in (5) is equivalent to

$$\begin{aligned}J(\boldsymbol{U}_{\mathrm{S}}(k)) = &||\boldsymbol{Y}^*_{\mathrm{S}}(k+1) - \boldsymbol{Y}_{\mathrm{S}}(k+1)||^2_{\bar{\boldsymbol{Q}}_{\mathrm{S}}}\\ &+ \lambda_{\mathrm{sw}}||\boldsymbol{S}(k+1)|| + \lambda_{\mathrm{u}}||\Delta\boldsymbol{U}_{\mathrm{S}}(k)||,\end{aligned}$$

and can be regrouped as

$$J(\boldsymbol{U}_{\mathrm{S}}(k)) = \boldsymbol{U}_{\mathrm{S}}(k)^\top\boldsymbol{H}_{\mathrm{S}}\boldsymbol{U}_{\mathrm{S}}(k) + 2\boldsymbol{\Theta}_{\mathrm{S}}(k)\boldsymbol{U}_{\mathrm{S}}(k) + \theta_{\mathrm{S}}(k),$$

where $\boldsymbol{H}_{\mathrm{S}} = \boldsymbol{\Upsilon}_{\mathrm{S}}^\top\bar{\boldsymbol{Q}}_{\mathrm{S}}\boldsymbol{\Upsilon}_{\mathrm{S}} + \lambda_{\mathrm{sw}}\boldsymbol{L}_{\mathrm{S}}^\top\boldsymbol{L}_{\mathrm{S}} + \lambda_{\mathrm{u}}\boldsymbol{\Pi}_{\mathrm{S}}^\top\boldsymbol{\Pi}_{\mathrm{S}}$ is called the Hessian and $\boldsymbol{\Theta}_{\mathrm{S}}(k) = ((\boldsymbol{\Gamma}_{\mathrm{S}}\boldsymbol{x}_{\mathrm{S}}(k) - \boldsymbol{Y}^*_{\mathrm{S}}(k+1) + \boldsymbol{\Psi}_{\mathrm{S}}\boldsymbol{v}_{\mathrm{g}}(k))^\top\bar{\boldsymbol{Q}}_{\mathrm{S}}\boldsymbol{\Upsilon}_{\mathrm{S}} - \lambda_{\mathrm{u}}(\boldsymbol{E}_{\mathrm{S}}\boldsymbol{u}_{\mathrm{S}}(k-1))^\top\boldsymbol{\Pi}_{\mathrm{S}})^\top)^\top$ is the linear part of the cost. The cost term $\theta_{\mathrm{S}}(k)$ is independent of the decision variable $\boldsymbol{U}_{\mathrm{S}}(k)$ and, therefore, gets discarded, and instead gets replaced by another constant term to complete the squares.

The matrices $\boldsymbol{\Upsilon}_{\mathrm{S}}$, $\boldsymbol{\Pi}_{\mathrm{S}}$, and $\boldsymbol{L}_{\mathrm{S}}$ are constructed so that $\lambda_{\mathrm{sw}}\boldsymbol{L}_{\mathrm{S}}^\top\boldsymbol{L}_{\mathrm{S}}+\lambda_{\mathrm{u}}\boldsymbol{\Pi}_{\mathrm{S}}^\top\boldsymbol{\Pi}_{\mathrm{S}} \succ 0$ and $\boldsymbol{\Upsilon}_{\mathrm{S}}^\top\bar{\boldsymbol{Q}}_{\mathrm{S}}\boldsymbol{\Upsilon}_{\mathrm{S}} \succeq 0$. Consequently, it holds for the Hessian that $\boldsymbol{H}_{\mathrm{S}} \succ 0$, and there exists a generator matrix $\boldsymbol{V}_{\mathrm{S}}$ such that $\boldsymbol{V}_{\mathrm{S}}^\top\boldsymbol{V}_{\mathrm{S}} = \boldsymbol{H}_{\mathrm{S}}$. The matrix $\boldsymbol{V}_{\mathrm{S}}$ is lower triangular, and every fourth row has only the slack variable penalty as a diagonal entry.

With the definitions above, (5) is equivalent to the following ILS problem:

$$\begin{aligned}\min_{\boldsymbol{U}_{\mathrm{S}}(k)}\ & ||\hat{\boldsymbol{U}}_{\mathrm{S}}(k) - \boldsymbol{V}_{\mathrm{S}}\boldsymbol{U}_{\mathrm{S}}(k)||_2^2\\ \text{s.t. }& \boldsymbol{u}(\ell) \in \{-1,\ 0,\ 1\}^3, \forall \ell,\\ & s(\ell) = \tilde{h}_\ell(\boldsymbol{U}_{\mathrm{past}}(\ell)), \forall \ell > k.\end{aligned} \tag{6}$$

where $\hat{\boldsymbol{U}}_{\mathrm{S}}(k) = -\boldsymbol{V}_{\mathrm{S}}\boldsymbol{H}_{\mathrm{S}}^{-1}\boldsymbol{\Theta}_{\mathrm{S}}(k)$ is the unconstrained solution to (5), i.e., the solution of (5) without integer and slack constraints. To simplify the presentation, the slack

variables' functional dependence on the inputs is captured by the functions $\tilde{h}_\ell : \mathbb{R}^{3(\ell-k)} \mapsto \mathbb{R}_{\geq 0}$, resulting in the constraint $s(\ell) = \tilde{h}_\ell(\boldsymbol{U}_{\mathrm{past}}(\ell)), \forall \ell > k$, where $\boldsymbol{U}_{\mathrm{past}}(\ell) = [\boldsymbol{u}(k)^\top \dots \boldsymbol{u}(\ell-1)^\top]^\top, \forall \ell > k$ denotes the past inputs.[3] These functions depend on the initial states, e.g., $\boldsymbol{x}_{\mathrm{sw}}(k)$, and can be evaluated iteratively via the constraints in (5) for different ℓ. For the slack variables $s(\ell)$, corresponding entries of the unconstrained solution $\hat{\boldsymbol{U}}_{\mathrm{S}}$ are 0, since the constraint $s(\ell) = \tilde{h}_\ell(\boldsymbol{U}_{\mathrm{past}}(\ell))$ is not imposed on the unconstrained solution.

B. Sphere decoder

The objective in (6) relates to the squared distance between the unconstrained solution $\hat{\boldsymbol{U}}_{\mathrm{S}}(k)$ and the optimization variable $\boldsymbol{U}_{\mathrm{S}}(k)$ in the space of the lattice defined by $\boldsymbol{V}_{\mathrm{S}}$. Given a feasible initial solution $\boldsymbol{U}_{\mathrm{S,ini}}(k)$ with the objective value d^2_{ini}, the sphere decoder systematically searches for a better feasible solution within a sphere with the initial squared radius of $\rho^2 = d^2_{\mathrm{ini}}$. The sphere decoder shrinks the squared radius ρ^2 whenever it finds a better solution $\boldsymbol{U}_{\mathrm{S}}(k)$. Through this process, suboptimal solutions are discarded until only the optimal solution $\boldsymbol{U}^{\mathrm{opt}}_{\mathrm{S}}(k)$ is left inside the sphere.

The exploration procedure of the sphere decoder for a better solution relies on the structure of the squared distance d^2. Let $\boldsymbol{\Xi}_i$ be the i^{th} row vector of a matrix $\boldsymbol{\Xi}$, and φ_i be the i^{th} element of a vector $\boldsymbol{\varphi}$, then

$$d^2 = \sum_{i=1}^{N_{\mathrm{p}}} \left|\left| \hat{U}_{\mathrm{S},i} - \boldsymbol{V}_{\mathrm{S},i} \boldsymbol{U}_{\mathrm{S}}(k) \right|\right|_2^2 .$$

Instead of directly calculating d^2 for the full candidate solution vector, we can iteratively compute an incremental squared distance term d_l^2, where $l \in \{1, \dots, 4N_{\mathrm{p}}\}$ and $d^2_{4N_{\mathrm{p}}} = d^2$. For this, we exploit the lower triangular structure of $\boldsymbol{V}_{\mathrm{S}}$, i.e.,

$$d_l^2 = \sum_{i=1}^{l} \left(\hat{U}_{\mathrm{S},i} - \sum_{j=1}^{i} V_{\mathrm{S},i,j} U_{\mathrm{S},j}(k) \right)^2 , \tag{7}$$

where $\Xi_{i,j}$ denotes the element in the i^{th} row and j^{th} column of a matrix $\boldsymbol{\Xi}$. The incremental squared distance d_l^2 allows the sphere decoder to avoid evaluating the objective function at once for a fully-determined candidate solution. Instead, we can determine the entries of the candidate vector one by one, starting at time step k. We can then increment the squared distance accordingly. This way, a large set of candidates can be discarded whenever the incremental distance already exceeds the sphere's current radius, $d_l^2 > \rho^2$, before all the entries are determined.

With this in mind, the sphere decoder can be formulated as described in Algorithm 1. When starting the recursive algorithm, we let $\boldsymbol{U}^{\mathrm{opt}}_{\mathrm{S}} = \boldsymbol{U}^{\mathrm{ini}}_{\mathrm{S}}(k)$, $\hat{\boldsymbol{U}}_{\mathrm{S}} = \hat{\boldsymbol{U}}_{\mathrm{S}}(k)$, $d_l^2 = 0$, $l = 1$, $\ell = k$, $\boldsymbol{u}^{\mathrm{prev}}_{\mathrm{S}} = \boldsymbol{u}_{\mathrm{S}}(k-1)$, and $\rho^2 = \rho^2_{\mathrm{ini}}(k)$. The initial solution $\boldsymbol{U}^{\mathrm{ini}}_{\mathrm{S}}(k)$ can be obtained with an educated guess (i.e., a shifted version of the previous optimal solution $\boldsymbol{U}^{\mathrm{opt}}_{\mathrm{S}}(k-1)$), the Babai estimate (e.g., see [10]), or the best of both. Additionally, we need to pre-compute the corresponding slack variables in $\boldsymbol{U}^{\mathrm{ini}}_{\mathrm{S}}(k)$.

[3]This change in notation highlights that the sphere decoder in the next subsection will in fact be applicable to slack variables of constraints with any nonlinear relationship to the past inputs $\boldsymbol{U}_{\mathrm{past}}(\ell)$.

Algorithm 1 Modified sphere decoder for FL-MPC

1: Function: FLSphDec
2: Input: $\boldsymbol{U}_{\mathrm{S}}$, $\boldsymbol{U}^{\mathrm{opt}}_{\mathrm{S}}$, $\hat{\boldsymbol{U}}_{\mathrm{S}}$, ρ^2, d^2_{l-1}, $\underline{\mathrm{d}}^2_\ell$, l, ℓ, $\boldsymbol{u}^{\mathrm{prev}}$, $\boldsymbol{x}_{\mathrm{sw}}(\ell)$
3: Output: $\boldsymbol{U}^{\mathrm{opt}}_{\mathrm{S}}$, ρ^2
4: **if** $\mod(l,4) \neq 0$ **then**
5: $\quad \mathcal{U} \leftarrow \{-1, 0, 1\}$
6: **else**
7: $\quad \boldsymbol{u}^{\mathrm{prev}} \leftarrow \boldsymbol{u}(\ell)$
8: $\quad \ell \leftarrow \ell + 1$
9: $\quad$ Compute $\boldsymbol{x}_{\mathrm{sw}}(\ell)$, $s(\ell)$ and $\underline{\mathrm{d}}^2_\ell$, and let $\mathcal{U} \leftarrow s(\ell)$
10: **end if**
11: **for all** $u \in \mathcal{U}$ **do**
12: $\quad U_{\mathrm{S},l} \leftarrow u$
13: $\quad d_l^2 \leftarrow d^2_{l-1} + \left(\hat{U}_{\mathrm{S},l} - \sum_{j=1}^{l} V_{\mathrm{S},l,j} U_{\mathrm{S},j}(k)\right)^2$
14: $\quad$ **if** $d_l^2 + \underline{\mathrm{d}}^2_\ell < \rho^2$ **then**
15: $\quad\quad$ **if** $l < 4N_{\mathrm{p}}$ **then**
16: $\quad\quad\quad$ New recursion: $[\boldsymbol{U}^{\mathrm{opt}}_{\mathrm{S}}, \rho^2] \leftarrow$ FLSphDec($\boldsymbol{U}_{\mathrm{S}}$, $\boldsymbol{U}^{\mathrm{opt}}_{\mathrm{S}}$, $\hat{\boldsymbol{U}}_{\mathrm{S}}$, ρ^2, d_l^2, $\underline{\mathrm{d}}^2_\ell$, $l+1$, ℓ, $\boldsymbol{u}^{\mathrm{prev}}$, $\boldsymbol{x}_{\mathrm{sw}}(\ell)$)
17: $\quad\quad$ **else**
18: $\quad\quad\quad \boldsymbol{U}^{\mathrm{opt}}_{\mathrm{S}} \leftarrow \boldsymbol{U}_{\mathrm{S}}$ and $\rho^2 \leftarrow d_l^2$
19: $\quad\quad$ **end if**
20: $\quad$ **end if**
21: **end for**

A tree of depth $4N_{\mathrm{p}}$, and levels denoted by l, representing all possible solutions of $\boldsymbol{U}_{\mathrm{S}}(k)$, is traversed. Every first three nodes, i.e., $\mod(l,4) \neq 0$, decide upon the switch position $\boldsymbol{u}(\ell)$, whereas every fourth node computes the corresponding slack variable $s(\ell+1)$. This can be done by evaluating functions $\tilde{h}_\ell$. For (6), this is done with the definition of the slack variable in (4) after iteratively computing $\boldsymbol{x}_{\mathrm{sw}}(\ell+1)$ via (2). Note that the complexity introduced by the fourth node is low, since it is a simple computation without requiring any exploration. Note also that the slack variable prediction models are not explicitly included when deriving $\boldsymbol{V}_{\mathrm{S}}$, but remain hidden in the computation in Line 9. A simple lower bound for the slack variable cost to be incurred in the future steps within the prediction horizon is given by $\underline{\mathrm{d}}^2_\ell = 0$. In the next subsection, to cut branches as early as possible, we will provide a tighter lower bound $\underline{\mathrm{d}}^2_{\mathrm{bound},\ell}$. This bound will be derived based on the prediction model of the slack variable for the switching frequency constraint. The algorithm can be used with either of $\underline{\mathrm{d}}^2_\ell = 0$ or $\underline{\mathrm{d}}^2_\ell = \underline{\mathrm{d}}^2_{\mathrm{bound},\ell}$.

After picking the value for the input, the incremental cost (7) at level l is compared with the sphere's squared radius ρ^2, i.e., the current best solution. The tree is further traversed if $d_l^2 + \underline{\mathrm{d}}^2_l < \rho^2$ still holds. Otherwise, the node with its sub-tree is pruned.

This procedure is repeated until either all branch values of a node are cut or a solution with $d^2_{4N_{\mathrm{p}}} < \rho^2$ is found at a leaf node. If $d^2_{4N_{\mathrm{p}}} < \rho^2$, then the sphere radius is reduced to $\rho^2 = d^2_{4N_{\mathrm{p}}}$ and $\boldsymbol{U}^{\mathrm{opt}}_{\mathrm{S}}$ is updated. The algorithm starts backtracking. If a new solution with cost $(d_i')^2 < \rho^2$ is found, the algorithm traverses the new branch again. The optimal solution

$\boldsymbol{U}_{\mathrm{S}}^{\mathrm{opt}}(k) = \boldsymbol{U}_{\mathrm{S}}^{\mathrm{opt}}$ is found once the last branch has been cut and the node at $l = 1$ is reached.

C. Lower bound for computational speed-up

Consider $\ell < k + N_{\mathrm{p}}$, since otherwise there is no future cost to be incurred by the slack variable. Let

$$J_{\mathrm{S}}(\ell) = \lambda_{\mathrm{sw}} \sum_{n=1}^{k+N_{\mathrm{p}}-\ell} ||\max(\boldsymbol{C}_{\mathrm{sw}}\boldsymbol{x}_{\mathrm{sw}}(\ell+n) - f_{\mathrm{sw}}^{*}, 0)||_2^2, \quad (8)$$

denote the cost incurred from the slack variable (strictly) after time step ℓ if generic inputs $\boldsymbol{u}(\ell-1+n)$, $n \in \{1, \ldots, k+N_{\mathrm{p}}-\ell\}$, are used for the rest of the horizon. We show that there exists a lower bound $\underline{\mathrm{d}}^2_{\mathrm{bound},\ell} \leq J_{\mathrm{S}}(\ell)$, which is often nontrivial, i.e., $\underline{\mathrm{d}}^2_{\mathrm{bound},\ell} > 0$, and can simply be computed from the already determined state variable $\boldsymbol{x}_{\mathrm{sw}}(\ell)$ in step ℓ. Intuitively, this will originate from the case where no more switching occurs in the remainder of the prediction horizon of the algorithm.

Proposition 1. *Given $\ell < k + N_{\mathrm{p}}$, define*

$$\underline{d}^2_{bound,\ell} = \lambda_{\mathrm{sw}} \sum_{n=1}^{k+N_{\mathrm{p}}-\ell} ||\max(\boldsymbol{C}_{\mathrm{sw}}\boldsymbol{A}_{\mathrm{sw}}^{n}\boldsymbol{x}_{\mathrm{sw}}(\ell) - f_{\mathrm{sw}}^{*}, 0)||_2^2.$$

We have that $\underline{d}^2_{bound,\ell} \leq J_{\mathrm{S}}(\ell)$.

Proof: Observe that we obtain the switching state $\boldsymbol{x}_{\mathrm{sw}}(\ell+n)$ in a sequential manner, i.e.,

$$\boldsymbol{x}_{\mathrm{sw}}(\ell+n) = \boldsymbol{A}_{\mathrm{sw}}^{n}\boldsymbol{x}_{\mathrm{sw}}(\ell) + \sum_{j=0}^{n-1} \boldsymbol{A}_{\mathrm{sw}}^{j}\boldsymbol{B}_{\mathrm{sw}}|\Delta\boldsymbol{u}(\ell+j)|.$$

We have that

$$\boldsymbol{A}_{\mathrm{sw}}^{n}\boldsymbol{x}_{\mathrm{sw}}(\ell) \leq \boldsymbol{A}_{\mathrm{sw}}^{n}\boldsymbol{x}_{\mathrm{sw}}(\ell) + \sum_{j=0}^{n-1} \boldsymbol{A}_{\mathrm{sw}}^{j}\boldsymbol{B}_{\mathrm{sw}}|\Delta\boldsymbol{u}(\ell+j)|,$$

due to non-negativity of the term on the right. Moreover, $\max((\cdot) - f_{\mathrm{sw}}^{*}, 0)$ is a non-negative and a non-decreasing operator.

Invoking these two observations, the bound $\underline{\mathrm{d}}^2_{\mathrm{bound},\ell}$ below obtained by assuming no future switchings can be shown to provide a lower bound to (8), that is, the slack variable cost to be incurred in the future steps for any generic future inputs:

$$\begin{aligned}\underline{\mathrm{d}}^2_{\mathrm{bound},\ell} &= \lambda_{\mathrm{sw}} \sum_{n=1}^{k+N_{\mathrm{p}}-\ell} ||\max(\boldsymbol{C}_{\mathrm{sw}}\boldsymbol{A}_{\mathrm{sw}}^{n}\boldsymbol{x}_{\mathrm{sw}}(\ell) - f_{\mathrm{sw}}^{*}, 0)||_2^2 \\ &\leq \lambda_{\mathrm{sw}} \sum_{n=1}^{k+N_{\mathrm{p}}-\ell} ||\max(\boldsymbol{C}_{\mathrm{sw}}\boldsymbol{x}_{\mathrm{sw}}(\ell+n) - f_{\mathrm{sw}}^{*}, 0)||_2^2 \\ &= J_{\mathrm{S}}(\ell).\end{aligned}$$

This concludes the proof. ∎

IV. Case Studies

We benchmark the FL-MPC against FT-MPC. We showcase the benefits of FL-MPC in common test cases such as steady-state current tracking and power ramp-ups. We also demonstrate the controllers during changes in the switching frequency reference to showcase the impact of the different objective functions.

TABLE I: Simulation Parameters

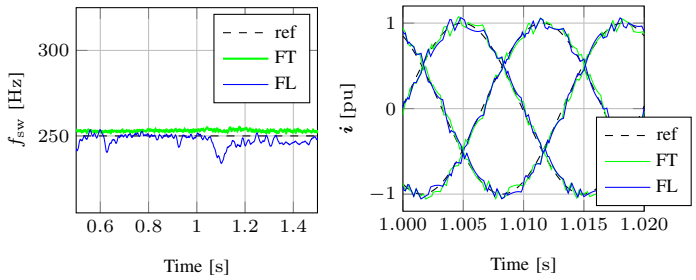
Physical System		Controllers		Simulation	
L	0.266 pu	N_{p}	5	T_{sim}	0.5 µs
R	0.015 pu	a_1	0.99	a_1^{visual}	0.995
V_{g}	1 pu	a_2	0.99	a_2^{visual}	0.995
V_{dc}	1.9 pu	λ_{u}	13×10^{-3}		
f_1	50 Hz	λ_{sw}	60		
		T_{c}	100 µs		

TABLE II: Simulation results; Current TDD and average switching frequency $\bar{f}_{\mathrm{sw}}$ for the steady-state simulation.

	FT-MPC	FL-MPC	Improvement [%]
Current TDD [%]	4.95	4.70	5.1
Avg. sw. frequency [Hz]	253	248	-

TABLE III: Simulation results; Errors $e_{I,\mathrm{FT}}$ and $e_{I,\mathrm{FL}}$ for the power ramp and the switching frequency step-up.

	$e_{I,\mathrm{FT}}$ [%]	$e_{I,\mathrm{FL}}$ [%]	Improvement [%]
Power ramp-up	6.8	6.1	9.8
f_{sw} step	8.8	3.9	56



(a) Switching frequencies (b) Currents over a fundamental

Fig. 3: Steady-state operation.

For the system presented in Section II, the parameters are shown in Table I. Here, T_{sim} refers to the simulation step size, a_1^{visual} and a_2^{visual} are used for plotting purposes. The controller's current tracking performance is evaluated with the current total demand distortion (TDD). The current TDD is well-suited for steady-state measurements, however, it is less applicable for evaluating the current tracking performance over short intervals during transients. Hence, we define the 2-norm of the current tracking error:

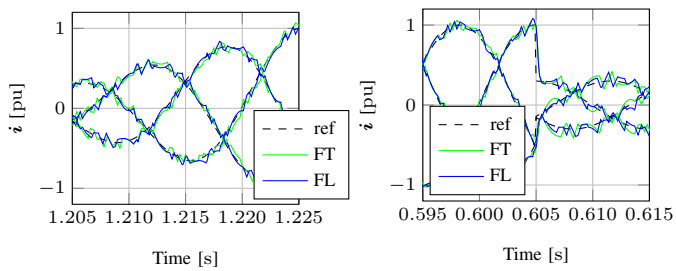
$$e_I^2 = \frac{1}{\sqrt{2}T} \sum_{t=t_0}^{t_0+T} ||\boldsymbol{x}^{*}(t) - \boldsymbol{x}(t)||_2^2.$$

The average switching frequency $\bar{f}_{\mathrm{sw}}$ is defined as

$$\bar{f}_{\mathrm{sw}} = \frac{1}{T} \sum_{t=t_0}^{t_0+T} f_{\mathrm{sw}}(t).$$

The two controllers are tuned with exactly the same parameters λ_{sw} and λ_{u} to ensure a fair comparison that focuses solely on the effect of a change from a switching frequency tracking objective to a switching frequency limiting objective.

We first perform a steady-state simulation for 1.5 s (i.e., 75 fundamental periods). An outer controller generates a



(a) Currents during power ramp. (b) Currents during power step.

Fig. 4: Current transients.

current reference, injecting to the grid $P = 1\,\text{pu}$, $Q = 0\,\text{pu}$. We initialize the measurement at $t_0 = 0.5\,\text{s}$ and $T = 1\,\text{s}$. Considering typical medium voltage applications, we choose a reference of $f^*_{\text{sw}} = 250\,\text{Hz}$. In the steady-state, we measure an average switching frequency of $\bar{f}_{\text{sw,FT}} = 253\,\text{Hz}$ for FT-MPC and $\bar{f}_{\text{sw,FL}} = 248\,\text{Hz}$ for FL-MPC. The difference in average switching frequencies is due to the different objectives and it is visualized in Figure 3a. The current TDD measurement is provided in Table II and shows a 5.1% improvement for the FL-MPC over the FT-MPC at a lower average switching frequency. To justify choosing the same parameters λ_{sw} and λ_{u} for both controllers, we varied these two parameters for FT-MPC and verified that it cannot achieve a TDD similar to FL-MPC while being close to $f^*_{\text{sw}} = 250\,\text{Hz}$. For example, for $\lambda_{\text{u}} = 8 \times 10^{-3}$ and $\lambda_{\text{sw}} = 45$, we obtained TDD$= 4.90\%$ and $\bar{f}_{\text{sw,FT}} = 255\,\text{Hz}$, whereas for $\lambda_{\text{u}} = 17 \times 10^{-3}$ and $\lambda_{\text{sw}} = 75$, we obtained TDD$= 5.02\%$ and $\bar{f}_{\text{sw,FT}} = 252\,\text{Hz}$. Finally, Figure 3b shows the currents over a fundamental during steady-state.

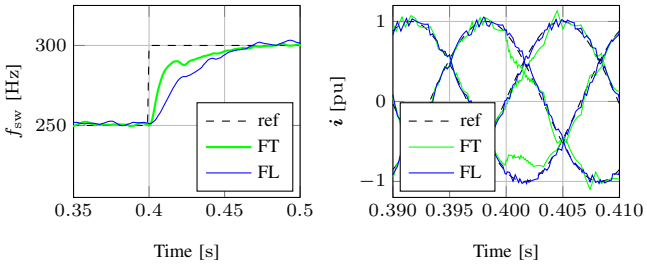
Similar improvements can also be observed when applying an active power ramp. Let $t_0 = 1.205\,\text{s}$, and consider one fundamental period $T_1 = \frac{1}{f_1}$ length. The ramp is initialized at exactly $t_0 = 1.205\,\text{s}$ with $P = 0.3\,\text{pu}$ and ends after one fundamental period at $P = 1\,\text{pu}$, while $Q = 0\,\text{pu}$. The simulation results with the improved current tracking performance of the FL-MPC over FT-MPC are shown in Figure 4a and Table III. Even though it is not shown explicitly with a plot in Figure 4, note that FL-MPC lowers the switching frequency during a ramp, which might explain the performance improvement shown in Table III. Such a difference in switching frequency has not been observed during the power step. These different scenarios show that FL-MPC is at least as good, and if not better, in tracking a current reference when compared to FT-MPC. This is not surprising, considering that FT-MPC might unnecessarily prioritize switching frequency tracking.

Next, though it is not realistic, to showcase an obvious case of current tracking improvement, we apply a step in the switching frequency reference. The simulation results are depicted in Figure 5. Here, the FT-MPC realizes a faster step than FL-MPC in terms of switching frequency. We initialize the measurement at $t_0 = 0.4\,\text{s}$ and measure one fundamental period. Table II shows a significantly improved performance for FL-MPC because FT-MPC jeopardizes current tracking to increase the switching frequency as fast as possible.

To evaluate the benefits of the computational speed-up, we simulated the system while measuring the total time for sphere decoding.[4] The results are shown in Table IV. Here, we can see a clear improvement in the FL-MPC computation times; long computation instances are eliminated thanks to the lower bound in Proposition 1. These reduced times are comparable to those of FT-MPC. To better understand the comparison of the two controllers, we provide also the average number of converter switching state nodes traversed in Table IV. Since the nodes for $\boldsymbol{p}(k)$ and $s(k)$ are computations based on the previous nodes, this quantity represents the level of exploration done by the sphere-decoders. We observe that FL-MPC traverses fewer nodes than FT-MPC. However, the computation times are similar due to the slightly longer computations required for the slack variable. A final comparison of the characteristics of the two controllers is provided in Table V.

TABLE IV: Total simulation time (Sim time), maximum solving time (Max time), 70^{th} percentile, 95^{th} percentile, and the average number of converter switching state nodes traversed (# nodes) for FT-MPC, and FL-MPC with and without computational speed-up.

	FT-MPC	FL-MPC no spd.-up	FL-MPC w/ spd.-up
Sim time [s]	30	78	22
Max time [ms]	9.6	232	7.7
70^{th} per. [ms]	0.60	0.87	0.55
95^{th} per. [ms]	1.40	13.50	1.40
# nodes	101.9	323.5	58.9



(a) Switching frequencies (b) Currents

Fig. 5: Switching frequency step-up.

TABLE V: Comparison of the characteristics of controllers.

	FT-MPC	FL-MPC
Cost function	Current tracking f_{sw} tracking	Current tracking f_{sw} limiting
Weighting factors	λ_{sw}: f_{sw} tracking weight λ_{u}: switching penalty	λ_{sw}: f_{sw} constraint slack weight λ_{u}: switching penalty
ILS optimizer dimension	$\dim(\boldsymbol{U}_{\text{T}}(k)) = 6N_{\text{p}}$	$\dim(\boldsymbol{U}_{\text{S}}(k)) = 4N_{\text{p}}$
Computational time	Low	Similar to FT-MPC via speed-up (Prop. 1)
Transient current tracking performance	Good	Improved

V. Conclusion

We introduced FL-MPC, replacing the switching frequency tracking objective of FT-MPC with switching frequency limiting. FL-MPC has been shown to perform better than FT-MPC

[4]All problems are solved via MATLAB on a computer equipped with 16 GB RAM and a 1.8 GHz quad-core Intel i7 processor.

in multiple scenarios. Additionally, a computational speed-up was proposed for the sphere decoder of FL-MPC. Future work could implement FL-MPC on an embedded FPGA.

REFERENCES

[1] P. Karamanakos, E. Liegmann, T. Geyer, and R. Kennel, "Model predictive control of power electronic systems: Methods, results, and challenges," *IEEE Open J. of Ind. Appl.*, vol. 1, pp. 95–114, 2020.

[2] J. Rodriguez, C. Garcia, A. Mora, F. Flores-Bahamonde, P. Acuna, M. Novak, Y. Zhang, L. Tarisciotti, S. A. Davari, Z. Zhang *et al.*, "Latest advances of model predictive control in electrical drives—Part I: Basic concepts and advanced strategies," *IEEE Trans. on Power Elec.*, vol. 37, no. 4, pp. 3927–3942, 2021.

[3] J. Rodriguez, C. Garcia, A. Mora, S. A. Davari, J. Rodas, D. F. Valencia, M. Elmorshedy, F. Wang, K. Zuo, L. Tarisciotti *et al.*, "Latest advances of model predictive control in electrical drives—Part II: Applications and benchmarking with classical control methods," *IEEE Trans. on Power Elec.*, vol. 37, no. 5, pp. 5047–5061, 2021.

[4] I. Harbi, J. Rodriguez, E. Liegmann, H. Makhamreh, M. L. Heldwein, M. Novak, M. Rossi, M. Abdelrahem, M. Trabelsi, M. Ahmed *et al.*, "Model predictive control of multilevel inverters: Challenges, recent advances, and trends," *IEEE Trans. on Power Elec.*, 2023.

[5] E. Zafra, S. Vazquez, T. Geyer, R. P. Aguilera, and L. G. Franquelo, "Long prediction horizon FCS-MPC for power converters and drives," *IEEE Open J. of the Ind. Elec. Society*, 2023.

[6] S. Muller, U. Ammann, and S. Rees, "New modulation strategy for a matrix converter with a very small mains filter," in *IEEE PESC*, vol. 3, 2003, pp. 1275–1280.

[7] J. Rodrıguez, J. Pontt, C. Silva, M. Salgado, S. Rees, U. Ammann, P. Lezana, R. Huerta, and P. Cortés, "Predictive control of three-phase inverter," *Elec. letters*, vol. 40, no. 9, 2004.

[8] T. Geyer and D. E. Quevedo, "Multistep finite control set model predictive control for power electronics," *IEEE Trans. on Pow. Elec.*, vol. 29, no. 12, pp. 6836–6846, 2014.

[9] P. Karamanakos, T. Geyer, T. Mouton, and R. Kennel, "Computationally efficient sphere decoding for long-horizon direct model predictive control," in *ECCE*, 2016, pp. 1–8.

[10] T. Dorfling, H. du Toit Mouton, T. Geyer, and P. Karamanakos, "Long-horizon finite-control-set model predictive control with nonrecursive sphere decoding on an FPGA," *IEEE Trans. on Pow. Elec.*, vol. 35, no. 7, pp. 7520–7531, 2020.

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

[12] T. Geyer and D. E. Quevedo, "Performance of multistep finite control set model predictive control for power electronics," *IEEE Trans. on Power Elec.*, vol. 30, no. 3, pp. 1633–1644, 2014.

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

[14] A. Rajapakse, A. Gole, and P. Wilson, "Electromagnetic transients simulation models for accurate representation of switching losses and thermal performance in power electronic systems," *IEEE Trans. on Pow. Del.*, vol. 20, no. 1, pp. 319–327, 2005.

[15] O. Al-Naseem, R. Erickson, and P. Carlin, "Prediction of switching loss variations by averaged switch modeling," in *APEC*, vol. 1, 2000, pp. 242–248 vol.1.

[16] P. Karamanakos and T. Geyer, "Guidelines for the design of finite control set model predictive controllers," *IEEE Trans. on Power Elec.*, vol. 35, no. 7, pp. 7434–7450, 2019.

[17] J. Rodriguez and P. Cortes, *Predictive control of power converters and electrical drives*. John Wiley & Sons, 2012.

[18] L. Tarisciotti, P. Zanchetta, A. Watson, S. Bifaretti, and J. C. Clare, "Modulated model predictive control for a seven-level cascaded h-bridge back-to-back converter," *IEEE Trans. on Ind. Elec.*, vol. 61, no. 10, pp. 5375–5383, 2014.

[19] S. Vazquez, A. Marquez, R. Aguilera, D. Quevedo, J. I. Leon, and L. G. Franquelo, "Predictive optimal switching sequence direct power control for grid-connected power converters," *IEEE Trans. on Ind. Elec.*, vol. 62, no. 4, pp. 2010–2020, 2014.

[20] P. Karamanakos, R. Mattila, and T. Geyer, "Fixed switching frequency direct model predictive control based on output current gradients," in *IECON*. IEEE, 2018, pp. 2329–2334.

[21] Q. Yang, P. Karamanakos, W. Tian, X. Gao, X. Li, T. Geyer, and R. Kennel, "Computationally efficient fixed switching frequency direct model predictive control," *IEEE Trans. on Pow. Elec.*, vol. 37, no. 3, pp. 2761–2777, 2022.

[22] P. Karamanakos, M. Nahalparvari, and T. Geyer, "Fixed switching frequency direct model predictive control with continuous and discontinuous modulation for grid-tied converters with LCL filters," *IEEE Trans. on Contr. Sys. Tech.*, vol. 29, no. 4, pp. 1503–1518, 2021.

[23] B. Stellato, T. Geyer, and P. J. Goulart, "High-speed finite control set model predictive control for power electronics," *IEEE Trans. on Pow. Elec.*, vol. 32, no. 5, pp. 4007–4020, 2017.

[24] P. Karamanakos, T. Geyer, and R. Kennel, "Constrained long-horizon direct model predictive control for power electronics," in *ECCE*. IEEE, 2016, pp. 1–8.

[25] M. Rossi, P. Karamanakos, and F. Castelli-Dezza, "Constrained long-horizon direct model predictive control for grid-connected converters with LCL filters," in *EPE ECCE Europe*. IEEE, 2022, pp. P–1.

[26] R. Keusch, H.-A. Loeliger, and T. Geyer, "Long-horizon direct model predictive control for power converters with state constraints," *IEEE Trans. on Contr. Sys. Tech.*, 2023.

[27] E. Liegmann, P. Karamanakos, T. Geyer, T. Mouton, and R. Kennel, "Long-horizon direct model predictive control with active balancing of the neutral point potential," in *IEEE PRECEDE*, 2017, pp. 89–94.

[28] N. Jin, D. Dai, H. Xie, J. Wu, and L. Guo, "Virtual vector-based FCS-MPC for npc three-level grid-tied inverter without weighting factor of neutral-point voltage balancing," *IEEE Access*, vol. 10, pp. 72 806–72 814, 2022.

[29] A. V. Oppenheim, A. S. Willsky, S. H. Nawab, and J.-J. Ding, *Signals and systems*. Prentice hall Upper Saddle River, NJ, 1997, vol. 2.

[30] P. van Emde Boas, "Another np-complete problem and the complexity of computing short vectors in a lattice," *Tecnical Report, Department of Mathmatics, University of Amsterdam*, 1981.

APPENDIX A
SPHERE DECODER FOR FT-MPC

Besides the conventional FCS-MPC of [8], the FT-MPC optimization problem is also amenable to an ILS reformulation. We write the full-horizon vector $\boldsymbol{Y}_{\mathrm{T}}(k+1)$ as a function of $\boldsymbol{U}_{\mathrm{T}}(k)$, $\boldsymbol{x}_{\mathrm{T}}(k)$, $\boldsymbol{v}_{\mathrm{g}}(k)$, that is,

$$\boldsymbol{Y}_{\mathrm{T}}(k+1) = \boldsymbol{\Gamma}_{\mathrm{T}}\boldsymbol{x}_{\mathrm{T}}(k) + \boldsymbol{\Upsilon}_{\mathrm{T}}\boldsymbol{U}_{\mathrm{T}}(k) + \boldsymbol{\Phi}_{\mathrm{T}}\boldsymbol{v}_{\mathrm{g}}(k).$$

We also define the vector $\boldsymbol{P}(k) = [\Delta\boldsymbol{u}^{\top}(k)\,\boldsymbol{p}^{\top}(k)\ldots\Delta\boldsymbol{u}^{\top}(k+N_{\mathrm{p}}-1)\,\boldsymbol{p}^{\top}(k+N_{\mathrm{p}}-1)]^{\top}$, which can be computed as $\boldsymbol{P}(k) = \boldsymbol{\Pi}_{\mathrm{T}}\boldsymbol{U}_{\mathrm{T}}(k) - \boldsymbol{E}_{\mathrm{T}}\boldsymbol{u}_{\mathrm{T}}(k-1)$. We remark that this definition slightly deviates from the original notation we have for the full-horizon vectors. The matrices $\boldsymbol{E}_{\mathrm{T}}$, $\boldsymbol{\Pi}_{\mathrm{T}}$, $\boldsymbol{\Gamma}_{\mathrm{T}}$, $\boldsymbol{\Upsilon}_{\mathrm{T}}$, $\boldsymbol{\Phi}_{\mathrm{T}}$, $\bar{\boldsymbol{Q}}_{\mathrm{T}}$ are defined in Appendix B.

We use the full-horizon vectors to write the objective (3) as

$$J(\boldsymbol{U}_{\mathrm{T}}(k))=||\boldsymbol{Y}^{*}_{\mathrm{T}}(k+1)-\boldsymbol{Y}_{\mathrm{T}}(k+1)||^2_{\bar{\boldsymbol{Q}}_{\mathrm{T}}}+\frac{\lambda_{\mathrm{u}}}{2}||\boldsymbol{P}(k+1)||^2_2.$$

As a remark, $||\boldsymbol{P}(k)||^2_2 = 2||\Delta\boldsymbol{U}(k)||^2_2$ because $||\Delta\boldsymbol{u}(\ell)||^2_2 = ||\boldsymbol{p}(\ell)||^2_2$. A similar trick was utilized also in [27]. By using the definitions of $\boldsymbol{Y}_{\mathrm{T}}(k+1)$ and $\boldsymbol{P}(k)$, we obtain the quadratic form

$$J(\boldsymbol{U}_{\mathrm{T}}(k)) = \boldsymbol{U}_{\mathrm{T}}(k)^{\top}\boldsymbol{H}_{\mathrm{T}}\boldsymbol{U}_{\mathrm{T}}(k) + 2\boldsymbol{\Theta}_{\mathrm{T}}(k)\boldsymbol{U}_{\mathrm{T}}(k) + \theta_{\mathrm{T}}(k),$$

where the Hessian is defined as $\boldsymbol{H}_{\mathrm{T}} = \boldsymbol{\Upsilon}_{\mathrm{T}}^{\top}\bar{\boldsymbol{Q}}_{\mathrm{T}}\boldsymbol{\Upsilon}_{\mathrm{T}} + \frac{\lambda_{\mathrm{u}}}{2}\boldsymbol{\Pi}_{\mathrm{T}}^{\top}\boldsymbol{\Pi}_{\mathrm{T}}$ and the linear cost term is $\boldsymbol{\Theta}_{\mathrm{T}}(k) = (\boldsymbol{\Gamma}_{\mathrm{T}}\boldsymbol{x}_{\mathrm{T}}(k) - \boldsymbol{Y}^{*}_{\mathrm{T}}(k+1) + \boldsymbol{\Psi}_{\mathrm{T}}\boldsymbol{v}_{\mathrm{g}}(k))^{\top}\bar{\boldsymbol{Q}}_{\mathrm{T}}\boldsymbol{\Upsilon}_{\mathrm{T}} - \frac{\lambda_{\mathrm{u}}}{2}(\boldsymbol{E}_{\mathrm{T}}\boldsymbol{u}_{\mathrm{T}}(k-1))^{\top}\boldsymbol{\Pi}_{\mathrm{T}})^{\top}$. The cost term $\theta_{\mathrm{T}}(k)$ is independent of the decision variable $\boldsymbol{U}_{\mathrm{T}}(k)$ and, therefore, gets discarded. Instead, we replace it with another constant term to complete the squares.

Algorithm 2 Modified sphere decoder for FT-MPC

Function: FTSphDec
Input: $\boldsymbol{U}_\mathrm{T}$, $\boldsymbol{U}_\mathrm{T}^\mathrm{opt}$, $\hat{\boldsymbol{U}}_\mathrm{T}$, ρ^2, d_{l-1}^2, l, ℓ, k, $\boldsymbol{u}^\mathrm{prev}$
Output: $\boldsymbol{U}_\mathrm{T}^\mathrm{opt}$, ρ^2
$j = l - 6(\ell - k)$
$\mathcal{U} \leftarrow \begin{cases} \{-1, 0, 1\}, \text{ if } j \in \{1, 2, 3\} \\ \{|u_{j-3}(\ell) - u_{j-3}^\mathrm{prev}|\}, \text{ otherwise} \end{cases}$
if $j = 6$ **then** $\boldsymbol{u}^\mathrm{prev} \leftarrow \boldsymbol{u}(\ell)$, $\ell \leftarrow \ell + 1$
end if
for all $u \in \mathcal{U}$ **do**
  $U_{\mathrm{T},l} \leftarrow u$, $d_l^2 \leftarrow d_{l-1}^2 + \left(\hat{U}_{\mathrm{T},l} - \sum_{j=1}^{l} V_{\mathrm{T},l,j} U_{\mathrm{T},j}(k)\right)^2$
  if $d_l^2 < \rho^2$ **then**
    if $l < 6N_\mathrm{p}$ **then**
      New recursion: $[\boldsymbol{U}_\mathrm{T}^\mathrm{opt}, \rho^2] \leftarrow$ FTSphDec($\boldsymbol{U}_\mathrm{T}$, $\boldsymbol{U}_\mathrm{T}^\mathrm{opt}$, $\hat{\boldsymbol{U}}_\mathrm{T}$, ρ^2, d_l^2, $l+1$, ℓ, k, $\boldsymbol{u}_\mathrm{T}^\mathrm{prev}$)
    else $\boldsymbol{U}_\mathrm{T}^\mathrm{opt} \leftarrow \boldsymbol{U}_\mathrm{T}$ and $\rho^2 \leftarrow d_l^2$
    end if
  end if
end for

The ILS reformulation of the FT-MPC problem is

$$\begin{aligned} \min_{\boldsymbol{U}_\mathrm{T}(k)} \; & ||\hat{\boldsymbol{U}}_\mathrm{T}(k) - \boldsymbol{V}_\mathrm{T}\boldsymbol{U}_\mathrm{T}(k)||_2^2 \\ \text{s.t. } & \boldsymbol{u}(\ell) \in \{-1,\ 0,\ 1\}^3, \forall \ell, \\ & \boldsymbol{p}(\ell) = |\Delta \boldsymbol{u}(\ell)|, \forall \ell, \end{aligned} \tag{9}$$

where the unconstrained solution $\hat{\boldsymbol{U}}_\mathrm{T}(k) = -\boldsymbol{V}_\mathrm{T}\boldsymbol{H}^{-1}\boldsymbol{\Theta}(k)$ is defined by the generator matrix $\boldsymbol{V}_\mathrm{T}$ from the Cholesky decomposition of $\boldsymbol{H}_\mathrm{T} = \boldsymbol{V}_\mathrm{T}^\top \boldsymbol{V}_\mathrm{T}$. For the absolute switching transitions $\boldsymbol{p}(\ell)$, corresponding entries of the unconstrained solution $\hat{\boldsymbol{U}}_\mathrm{T}(k)$ are uncorrelated to the $\boldsymbol{u}(\ell)$ entries, since the constraint $\boldsymbol{p}(\ell) = |\Delta \boldsymbol{u}(\ell)|$ is not imposed.

To solve the ILS problem presented in (9), a modified version of the sphere decoder similar to that from [9], [27] can be used. It is provided in Algorithm 2 for the sake of completeness. The algorithm iterates over inputs, starting from $\ell = k$, exploiting the lower triangular structure of the generator matrix $\boldsymbol{V}_\mathrm{T}$. Here, $\boldsymbol{V}_{\mathrm{T},l}$ refers to the l^th row of the matrix $\boldsymbol{V}_\mathrm{T}$, and similarly, $\hat{U}_{\mathrm{T},l}$ is the l^th entry of the vector $\hat{\boldsymbol{U}}_\mathrm{T}$. While maintaining the conventional algorithm structure, the input set $\mathcal{U}$ can simply be modified for the iterations involving the variable $\boldsymbol{p}(k)$. The sphere decoder would first iterate over the input $u_j(\ell)$ for some $j \in \{a, b, c\}$ at a certain time step ℓ. The absolute switching transition variables $p_j(\ell)$ would then simply be computed based on the switch positions $u_j(\ell)$ and $u_j(\ell - 1)$.

The algorithm is initialized with $\boldsymbol{U}_\mathrm{T} = \boldsymbol{U}_\mathrm{T}^\mathrm{ini}(k)$, $\hat{\boldsymbol{U}}_\mathrm{T} = \hat{\boldsymbol{U}}_\mathrm{T}(k)$, $\rho^2 = \rho_\mathrm{ini}^2(k)$, $d_{l-1}^2 = 0$, $l = 1$, $\ell = k$, $\boldsymbol{u}^\mathrm{prev} = \boldsymbol{u}(k-1)$. The initial solution $\boldsymbol{U}_\mathrm{T}^\mathrm{ini}(k)$ can be computed with the educated guess (i.e., a shifted version of the previous optimal solution $\boldsymbol{U}_\mathrm{T}^\mathrm{opt}(k-1)$), Babai estimate (a rounded but feasible/valid version of the unconstrained solution $\hat{\boldsymbol{U}}_\mathrm{T}(k)$), or both [10]. Recall that the dependence of $\boldsymbol{p}(\ell)$ on $\Delta \boldsymbol{u}(\ell)$ has to be imposed to obtain a valid/feasible initial solution. The initial sphere radius $\rho_\mathrm{ini}^2(k)$ corresponds to the cost of the initial solution $\boldsymbol{U}_\mathrm{T}^\mathrm{ini}(k)$.

APPENDIX B
MATRICES FOR FT-MPC AND FL-MPC

First consider FT-MPC. Let $\boldsymbol{\Pi}_\mathrm{T} \in \mathbb{R}^{6N_\mathrm{p} \times 6N_\mathrm{p}}$,

$$\boldsymbol{\Pi}_\mathrm{T} = \begin{bmatrix} \boldsymbol{I}_3 & & & & & \\ \boldsymbol{0}_3 & \boldsymbol{I}_3 & & & & \\ -\boldsymbol{I}_3 & \boldsymbol{0}_3 & \boldsymbol{I}_3 & & & \\ & \ddots & \ddots & \ddots & & \\ & & -\boldsymbol{I}_3 & \boldsymbol{0}_3 & \boldsymbol{I}_3 & \\ & & & \boldsymbol{0}_3 & \boldsymbol{0}_3 & \boldsymbol{I}_3 \end{bmatrix},$$

where $\boldsymbol{I}_n \in \mathbb{R}^{n \times n}$ denotes the identity matrix. Further, we define $\boldsymbol{0}_n \in \mathbb{R}^{n \times n}$ and $\boldsymbol{0}_{n \times m} \in \mathbb{R}^{n \times m}$ as zero matrices. Notice that $\boldsymbol{\Pi}_\mathrm{T}$ is full-rank. With this definition we obtain $\boldsymbol{P}(k) = [\Delta \boldsymbol{u}^\top(k)\ \boldsymbol{p}^\top(k) \dots \Delta \boldsymbol{u}^\top(k + N_\mathrm{p} - 1)\ \boldsymbol{p}^\top(k + N_\mathrm{p} - 1)]^\top$ from $\boldsymbol{P}(k) = \boldsymbol{\Pi}_\mathrm{T}\boldsymbol{U}_\mathrm{T}(k) - \boldsymbol{E}_\mathrm{T}\boldsymbol{u}_\mathrm{T}(k-1)$, given that

$$\boldsymbol{E}_\mathrm{T} = \begin{bmatrix} \boldsymbol{I}_3 & \boldsymbol{0}_3 & \cdots & \boldsymbol{0}_3 \\ \boldsymbol{0}_3 & \boldsymbol{0}_3 & \cdots & \boldsymbol{0}_3 \end{bmatrix}^\top \in \mathbb{R}^{6N_\mathrm{p} \times 6}.$$

For the FL-MPC, we define $\boldsymbol{\Pi}_\mathrm{S} \in \mathbb{R}^{4N_\mathrm{p} \times 4N_\mathrm{p}}$ and $\boldsymbol{L}_\mathrm{S} \in \mathbb{R}^{N_\mathrm{p} \times 4N_\mathrm{p}}$. The matrix $\boldsymbol{L}_\mathrm{S}$ extracts the slack variable with a 1 on every fourth diagonal entry, i.e., $\boldsymbol{L}_\mathrm{S} = \mathrm{diag}(\begin{bmatrix}\boldsymbol{0}_{1\times 3}, 1, \boldsymbol{0}_{1\times 3}, 1, \dots\end{bmatrix})$. The matrix $\boldsymbol{\Pi}_\mathrm{S}$ is defined as

$$\boldsymbol{\Pi}_\mathrm{S} = \begin{bmatrix} \boldsymbol{I}_3 & & & & & \\ \boldsymbol{0}_{1\times 3} & 0 & & & & \\ -\boldsymbol{I}_3 & \boldsymbol{0}_{3\times 1} & \boldsymbol{I}_3 & & & \\ & \ddots & \ddots & \ddots & & \\ & & -\boldsymbol{I}_3 & \boldsymbol{0}_{3\times 1} & \boldsymbol{I}_3 & \\ & & & 0 & \boldsymbol{0}_{1\times 3} & 0 \end{bmatrix},$$

so $\Delta \boldsymbol{U}_\mathrm{S}(k) = \boldsymbol{\Pi}_\mathrm{S}(k)\boldsymbol{U}_\mathrm{S}(k) - \boldsymbol{E}_\mathrm{S}\boldsymbol{u}_\mathrm{S}(k-1)$ holds with

$$\boldsymbol{E}_\mathrm{S} = \begin{bmatrix} \boldsymbol{I}_3 & \boldsymbol{0}_{3\times 1} & \boldsymbol{0}_{3\times 4(N_\mathrm{p}-1)} \\ \boldsymbol{0}_{1\times 3} & 0 & \boldsymbol{0}_{1\times 4(N_\mathrm{p}-1)} \end{bmatrix}^\top \in \mathbb{R}^{4N_\mathrm{p} \times 4}.$$

Notice that $\lambda_\mathrm{u}\boldsymbol{\Pi}_\mathrm{S}^\top\boldsymbol{\Pi}_\mathrm{S} + \lambda_\mathrm{sw}\boldsymbol{L}_\mathrm{S}^2\boldsymbol{L}_\mathrm{S} \succ 0$ for all $\lambda_\mathrm{u}, \lambda_\mathrm{sw} > 0$.

Let $\bar{\boldsymbol{Q}}_\mathrm{T} = \boldsymbol{I}_{N_\mathrm{p}} \otimes \boldsymbol{Q}_\mathrm{T} \in \mathbb{R}^{3N_\mathrm{p} \times 3N_\mathrm{p}}$ be the full-horizon output weight of FT-MPC, where $\otimes$ denotes the Kronecker product. The output weight of FL-MPC can similarly be defined as $\bar{\boldsymbol{Q}}_\mathrm{S} = \boldsymbol{I}_{N_\mathrm{p}} \otimes \boldsymbol{Q}_\mathrm{S} = \boldsymbol{I}_{2N_\mathrm{p}}$.

For both FT-MPC and FL-MPC, the dynamics are given by the matrices $\boldsymbol{A}_{(\cdot)} \in \mathbb{R}^{n \times n}$, $\boldsymbol{B}_{(\cdot)} \in \mathbb{R}^{n \times m}$, $\boldsymbol{D}_{(\cdot)} \in \mathbb{R}^{n \times 3}$, and $\boldsymbol{C}_{(\cdot)} \in \mathbb{R}^{q \times n}$, where $(\cdot) \in \{\mathrm{T}, \mathrm{S}\}$, $q = 3$, $n = 4$, $m = 6$ for FT-MPC, and $q = 2$, $n = 2$, $m = 4$ for FL-MPC. The horizon is denoted as

$$\begin{aligned} \boldsymbol{Y}_{(\cdot)}(k+1) &= [\boldsymbol{y}_{(\cdot)}(k+1)^\top\ \dots\ \boldsymbol{y}_{(\cdot)}(k+N_\mathrm{p})^\top]^\top, \\ \boldsymbol{Y}_{(\cdot)}(k+1) &= \boldsymbol{\Gamma}_{(\cdot)}\boldsymbol{x}_{(\cdot)}(k) + \boldsymbol{\Upsilon}_{(\cdot)}\boldsymbol{U}_{(\cdot)}(k) + \boldsymbol{\Psi}_{(\cdot)}\boldsymbol{V}_\mathrm{g}(k), \end{aligned}$$

where the following definitions hold:

$$\boldsymbol{\Gamma}_{(\cdot)} = \begin{bmatrix} \boldsymbol{C}_{(\cdot)}\boldsymbol{A}_{(\cdot)} \\ \vdots \\ \boldsymbol{C}_{(\cdot)}\boldsymbol{A}_{(\cdot)}^{N_\mathrm{p}} \end{bmatrix},$$

$$\boldsymbol{\Upsilon}_{(\cdot)} = \begin{bmatrix} \boldsymbol{C}_{(\cdot)}\boldsymbol{B}_{(\cdot)} & \boldsymbol{0}_{q\times m} & \cdots & \boldsymbol{0}_{q\times m} \\ \boldsymbol{C}_{(\cdot)}\boldsymbol{A}_{(\cdot)}\boldsymbol{B}_{(\cdot)} & \boldsymbol{C}_{(\cdot)}\boldsymbol{B}_{(\cdot)} & & \vdots \\ \vdots & & \ddots & \boldsymbol{0}_{q\times m} \\ \boldsymbol{C}_{(\cdot)}\boldsymbol{A}_{(\cdot)}^{N_\mathrm{p}-1}\boldsymbol{B}_{(\cdot)} & & \cdots & \boldsymbol{C}_{(\cdot)}\boldsymbol{B}_{(\cdot)} \end{bmatrix},$$

$$\boldsymbol{\Psi}_{(\cdot)} = \begin{bmatrix} \boldsymbol{C}_{(\cdot)}\boldsymbol{D}_{(\cdot)} & \boldsymbol{0}_{q\times 3} & \cdots & \boldsymbol{0}_{q\times 3} \\ \boldsymbol{C}_{(\cdot)}\boldsymbol{A}_{(\cdot)}\boldsymbol{D}_{(\cdot)} & \boldsymbol{C}_{(\cdot)}\boldsymbol{D}_{(\cdot)} & & \vdots \\ \vdots & & \ddots & \boldsymbol{0}_{q\times 3} \\ \boldsymbol{C}_{(\cdot)}\boldsymbol{A}_{(\cdot)}^{N_\mathrm{p}-1}\boldsymbol{D}_{(\cdot)} & & \cdots & \boldsymbol{C}_{(\cdot)}\boldsymbol{D}_{(\cdot)} \end{bmatrix}.$$

The matrices are of the sizes $\boldsymbol{\Gamma}_{(\cdot)} \in \mathbb{R}^{qN_\mathrm{p} \times n}$, $\boldsymbol{\Upsilon}_{(\cdot)} \in \mathbb{R}^{qN_\mathrm{p} \times mN_\mathrm{p}}$, and $\boldsymbol{\Psi}_{(\cdot)} \in \mathbb{R}^{qN_\mathrm{p} \times 3N_\mathrm{p}}$.