2024 European Control Conference (ECC)
June 25-28, 2024. Stockholm, Sweden

Assessment of a Laguerre Polynomial Based Sphere Decoding Algorithm for Direct MPC of Inverters

Demian Struve[1], Andrés Lopez Pulzovan[2], Thomas F. Geyer[3]

***Abstract*— In the last years, model predictive control (MPC) has become a significant competitor to conventional control strategies for power electronic applications. In particular, direct MPC using sphere decoding algorithms (SDAs) as optimizers have gained popularity in this context. Besides such specialized optimization algorithms, it is possible to improve the MPC performance by dimensional reduction of the optimization problem using Laguerre polynomials (LaPs). LaP based MPC has already been proven advantageous for example in robotic and marine applications. This paper presents an approach to link the SDA and LaPs by formulating a Laguerre polynomial based sphere decoding algorithm (LaP-SDA) for the control of inverters on the modulation-level. Following some introductions to the SDA and LaPs, the optimization problem and its particular structure for the SDA is transformed using sets of LaPs. This modification of the SDA results in a new admissible set and an additional optimization problem to find the optimal configuration of the LaPs. Moreover, it changes the shape of the search tree tailoring it wider but more shallow. Finally, the LaP-SDA is verified in a simulation using an example system. There, it is shown that, with a suitable configuration, the results of the SDA and LaP-SDA are identical. However, the simulation does not indicate a performance benefit of the LaP-SDA mainly because of the size of the admissible set. Nevertheless, prospective modifications to improve its performance and further applications of the LaP-SDA are presented.**

***Index Terms*— Model predictive control, sphere decoding, laguerre polynomials, inverters**

I. INTRODUCTION

Direct model predictive control (MPC), that is the control without a modulation stage, has become a relevant control strategy for power electronics in academia [1]. In this domain, various assessments of different direct MPC applications have been published over the last years, for example on the control of cascaded H-bridge inverters [2], two-level inverters for automotive applications [3] or for industrial drives [4].

Beyond its common advantages, that is control of multi-variable non-linear systems and respecting state and input constraints [5], research has shown that MPC of power electronics can surpass conventional control methods from efficiency and current distortion perspectives [3], [4].

In general, the core challenge which emerges with MPC is solving the underlying optimization problem for each step in time. Particularly, this challenge applies for power electronic applications because of the short time constants. Critical for the time required to solve the optimization problem is its dimension. On the one hand, it scales exponentially with the prediction horizon [6] while on the other hand, it is linearly proportional to the system order [7]. In power electronics both of these aspects result in a dimensional gain of the optimization problem. High order systems can quickly emerge for example due to additional components such as filters. Also, long prediction horizons are desirable because of the beneficial effect on distortion and stability [4], [8].

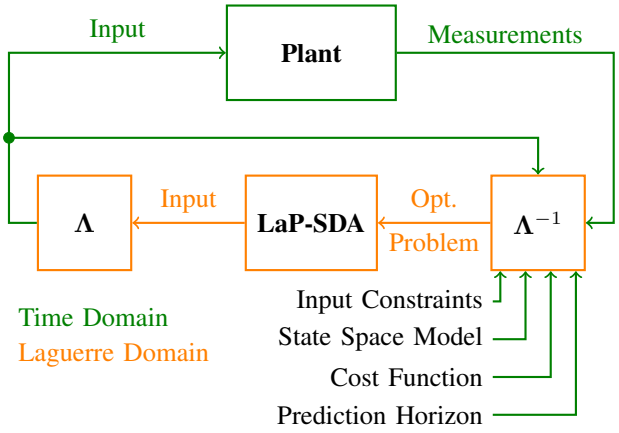


Fig. 1: Essential Idea of Laguerre Polynomial Based Sphere Decoding Algorithm (LaP-SDA) Using a Transformation $\mathbf{\Lambda}$

The aforementioned challenge of solving the optimization problem has been approached from different angles: First, more efficient and specialized solvers such as the miOSQP solver [9] or the sphere decoding algorithm (SDA) [4] have been developed. The latter has evolved as the most prominent method to solve the integer control problem which arises with *direct* MPC of inverters [10]. Secondly, the computational burden of high dimensional optimization problems can be alleviated by transformation in a lower dimension. A common method for this transformation is the incorporation of Laguerre polynomials (LaPs) in the original optimization problem [11]. Successful applications of this approach using LaPs in MPC can be found in [12] and [13] for robotic and marine applications, respectively.

As for now, research on different modifications and extensions to further improve the performance of the SDA have been published [3], [6]. However, to the knowledge of the authors, there is a lack of research on the supplementation of the SDA with LaPs for modulation-level control of inverters. Therefore, this paper introduces a Laguerre polynomial based sphere decoding algorithm (LaP-SDA) to assess the benefit of merging the SDA with the MPC schemes using LaPs.

The authors are with the department of Control of the Propulsion System, DLR Institute of Electrified Aero Engines, 03046 Cottbus, Germany
[1]demian.struve@dlr.de
[2]andres.lopezpulzovan@dlr.de
[3]thomas.geyer@dlr.de

The fundamental idea of the LaP-SDA is displayed in Fig. 1. There, not the original optimization problem (Opt. Problem) in the time domain, but the Laguerre based problem obtained by a transformation $\mathbf{\Lambda}^{-1}$, is solved to compute the system inputs.

II. LAGUERRE POLYNOMIAL BASED COST FUNCTION

A. Regular Optimization Problem

Using the linear time-invariant discrete state space representation of a system with n states, m inputs, and p outputs, the state vector

$$\boldsymbol{x}(k+1) = \boldsymbol{A}\boldsymbol{x}(k) + \boldsymbol{B}\boldsymbol{u}(k)$$

and the output vector

$$\boldsymbol{y}(k+1) = \boldsymbol{C}\boldsymbol{x}(k+1) + \boldsymbol{D}\boldsymbol{u}(k),$$

with $\boldsymbol{A} \in \mathbb{R}^{n\times n}$, $\boldsymbol{B} \in \mathbb{R}^{n\times m}$, $\boldsymbol{C} \in \mathbb{R}^{p\times n}$ and $\boldsymbol{D} = \mathbf{0}_{p\times m}$. At a time step k with the available state information $\boldsymbol{x}(k) \in \mathbb{R}^n$ and a future input sequence $\boldsymbol{U}(k) \in \mathbb{R}^{N_\mathrm{p} m}$, starting from time step k to step $k+N_\mathrm{p}$ with the prediction horizon N_p, the predicted output vector[1]

$$\boldsymbol{Y}(k) = \mathbf{\Gamma}\boldsymbol{x}(k) + \mathbf{\Upsilon}\boldsymbol{U}(k), \tag{1}$$

in which $\boldsymbol{Y}(k) \in \mathbb{R}^{N_\mathrm{p} p}$.

For MPC, the cost function J_R is defined to penalize deviations from the reference trajectory $\boldsymbol{Y}^*(k)$ and to penalize input changes both weighted with the positive semidefinite matrix $\boldsymbol{Q}$ and the scalar $\lambda_\mathrm{u} > 0$, respectively. Consequently, at a time step k

$$J_\mathrm{R}(k) = \sum_{i=k}^{i+N_\mathrm{p}-1} \|\boldsymbol{Y}^*(i) - \boldsymbol{Y}(i)\|^2_{\boldsymbol{Q}} + \lambda_\mathrm{u}\|\Delta\boldsymbol{u}(i)\|_2^2$$

holds true, with $\Delta\boldsymbol{u}(i) = \boldsymbol{u}(i) - \boldsymbol{u}(i-1)$, the squared Euclidean norm $\|\boldsymbol{x}\|_2^2$, and the weighted norm $\|\boldsymbol{x}\|^2_{\boldsymbol{Q}} = \boldsymbol{x}^T\boldsymbol{Q}\boldsymbol{x}$. A more compact version of this cost function is obtained using (1), by omitting the terms independent of $\boldsymbol{U}(k)$ and introducing the matrix $\mathbf{\Theta}(k)$. This leads to[2]

$$J_\mathrm{R}(k) = (\boldsymbol{U}(k))^T\boldsymbol{H}\boldsymbol{U}(k) + 2(\mathbf{\Theta}(k))^T\boldsymbol{U}(k). \tag{2}$$

For the SDA, (2) can further be rewritten by completing the squares, computing the unconstrained optimal solution, and introducing the generator matrix $\boldsymbol{V}_\mathrm{R}$ using a Cholesky decomposition of the symmetric and positive definite matrix $\boldsymbol{H}$ such that

$$\boldsymbol{V}_\mathrm{R}^T\boldsymbol{V}_\mathrm{R} = \boldsymbol{H}.$$

This yields

$$\bar{\boldsymbol{U}}_\mathrm{unc}(k) = -\boldsymbol{V}_\mathrm{R}\boldsymbol{H}^{-1}\mathbf{\Theta}(k)$$

and results in a suitable cost function for the SDA. For a three-level-inverter, integer and switch constraints for the elements in $\boldsymbol{U}(k)$ must be added resulting in the optimization problem

$$\boldsymbol{U}_\mathrm{opt}(k) = \arg\min_{\boldsymbol{U}(k)} \|\boldsymbol{V}_\mathrm{R}\boldsymbol{U}(k) - \bar{\boldsymbol{U}}_\mathrm{unc}(k)\|_2^2 \tag{3a}$$

$$\text{subject to } \boldsymbol{U}(k) \in \mathcal{U} = \{-1, 0, 1\}, \tag{3b}$$

$$\|\Delta\boldsymbol{u}(i)\|_\infty \leq 1 \;\forall\; i = k, \ldots, k+N_\mathrm{p}. \tag{3c}$$

Equation (3b) constrains the values in $\boldsymbol{U}(k)$ to the allowed integer switch positions while (3c) prohibits switching from -1 to 1 and vice versa[3].

B. General Laguerre Based MPC

LaPs are a set of orthogonal functions and due to their definition are a suitable and an efficient way to transform general MPC optimization problems into a lower dimension [11], [14]. Namely, for z-transformed LaPs $\mathcal{Z}\{l_j\} = \Gamma_j, j = 1, \ldots, N_\mathrm{L}$ in the discrete frequency domain with network dimension $N_\mathrm{L} \in \mathcal{T}$, pole $a \in \mathcal{B}$, and prediction horizon $N_\mathrm{p} \in \mathbb{N}^+$ according to the sets

$$\mathcal{T} = 1, 2, \ldots, N_\mathrm{p}, \tag{4}$$

$$\mathcal{B} = [0, 1), \tag{5}$$

the recursive computation

$$\Gamma_j(z,a) = \Gamma_{j-1}(z,a)\frac{z^{-1}-a}{1-az^{-1}},\ \Gamma_0(z,a) = \frac{\sqrt{1-a^2}}{1-az^{-1}}$$

holds. With l_j being the inverse z-transform of Γ_j, the time evolving sets of LaPs

$$\boldsymbol{l}(k) = \begin{bmatrix} l_1(k) & l_2(k) & \ldots & l_{N_\mathrm{L}}(k) \end{bmatrix}^T, k \in \mathcal{T} \tag{6}$$

in the discrete time domain are computed as

$$\boldsymbol{l}(k+1) = \boldsymbol{A}_\mathrm{L}\boldsymbol{l}(k)$$

using the auxiliary matrix $\boldsymbol{A}_\mathrm{L}$, whose definition can be found in [11], [15]. For $a = 0$ the set of LaPs in (6) becomes a set of pulses, that is

$$\boldsymbol{l}(k) = \begin{bmatrix} \mathbf{0}_k^T & 1 & 0 & \ldots & 0 \end{bmatrix}^T. \tag{7}$$

This special case is identical to the control strategy of MPC with the input signal

$$\boldsymbol{U}(k) = \begin{bmatrix} \boldsymbol{u}^T(k) & \boldsymbol{u}^T(k+1) & \ldots & \boldsymbol{u}^T(k+N_\mathrm{p}-1) \end{bmatrix}^T, \tag{8}$$

for which in general

[1] For the computation of the following matrices the reader is referred to [4], chapter 5.

[2] see footnote 1

[3] Namely, $\|\Delta\boldsymbol{u}(i)\|_\infty \coloneqq \max|\Delta\boldsymbol{u}(i)| \;\forall\; i = k, \ldots, k+N_\mathrm{p}$

$$\boldsymbol{u}^T(k+i) = \begin{bmatrix}\boldsymbol{\delta}(i) & \boldsymbol{\delta}(i-1) & \dots & \boldsymbol{\delta}(i-N_\mathrm{p}+1)\end{bmatrix}\boldsymbol{U}(k)$$

with $i = 0, \dots, N_\mathrm{p}-1$, m system inputs, and

$$\boldsymbol{\delta}(j) = \begin{cases}\mathbf{1}_m^T, & \text{if } j = 0\\ \mathbf{0}_m^T, & \text{otherwise}\end{cases}$$

holds. Consequently, one can display the input signal using a transformation matrix $\boldsymbol{\Lambda}$ consisting of LaPs resulting in

$$\boldsymbol{U}(k) = \boldsymbol{\Lambda}\boldsymbol{E}(k), \tag{9}$$

with the Laguerre coefficients

$$\boldsymbol{E}(k) = \begin{bmatrix}\boldsymbol{\eta}_1^T(k) & \boldsymbol{\eta}_2^T(k) & \dots & \boldsymbol{\eta}_m^T(k)\end{bmatrix}^T. \tag{10}$$

The transformation matrix

$$\boldsymbol{\Lambda} = \begin{bmatrix}\boldsymbol{\Lambda}_0^T & \boldsymbol{\Lambda}_1^T & \dots & \boldsymbol{\Lambda}_{N_\mathrm{p}-1}^T\end{bmatrix}^T \in \mathbb{R}^{N_\mathrm{p}m\times\kappa}, \tag{11}$$

with

$$\kappa = \sum_{j=1}^{m} N_{\mathrm{L},j} \tag{12}$$

is of block structure in which each block

$$\boldsymbol{\Lambda}_i = \begin{bmatrix}\boldsymbol{l}_1(i) & \mathbf{0}_{N_{\mathrm{L},2}}^T & \dots & \mathbf{0}_{N_{\mathrm{L},m}}^T\\ \mathbf{0}_{N_{\mathrm{L},1}}^T & \boldsymbol{l}_2(i) & \dots & \mathbf{0}_{N_{\mathrm{L},m}}^T\\ \vdots & \vdots & & \vdots\\ \mathbf{0}_{N_{\mathrm{L},1}}^T & \mathbf{0}_{N_{\mathrm{L},2}}^T & \dots & \boldsymbol{l}_m(i)\end{bmatrix}. \tag{13}$$

The entries in (13) are given by (6). Thus, (9) assigns each input $\boldsymbol{u}(k+i)$ the transformed input, namely

$$\boldsymbol{u}(k+i) = \boldsymbol{\Lambda}_i\boldsymbol{E}(k).$$

C. Transformation of Regular Optimization Problem

With (9), the cost function (2) is now rewritten, rendering

$$J_\mathrm{L}(k) = (\boldsymbol{\Lambda}\boldsymbol{E}(k))^T\boldsymbol{H}\boldsymbol{\Lambda}\boldsymbol{E}(k) + 2(\boldsymbol{\Theta}(k))^T\boldsymbol{\Lambda}\boldsymbol{E}(k). \tag{14}$$

Further, with the introduction of the matrices

$$\boldsymbol{\Omega} = \boldsymbol{\Lambda}^T\boldsymbol{H}\boldsymbol{\Lambda},$$
$$\boldsymbol{\Psi}(k) = (\boldsymbol{\Theta}(k))^T\boldsymbol{\Lambda},$$

(14) takes the form

$$J_\mathrm{L}(k) = (\boldsymbol{E}(k))^T\boldsymbol{\Omega}\boldsymbol{E}(k) + 2\boldsymbol{\Psi}(k)\boldsymbol{E}(k), \tag{15}$$

with the new optimization variable $\boldsymbol{E}(k)$ defined in (10).

It is important to examine how the optimization problem in (3) alters because of the introduction of LaPs. On the one hand, the dimension of the optimization variable can be lowered from $\mathbb{R}^{N_\mathrm{p}m}$ to $\mathbb{R}^{\kappa}$ because per definition [see (12)) and (4)]

$$\kappa \leq N_\mathrm{p}m.$$

Moreover, this circumstance applies to the dimension of the generator matrix $\boldsymbol{V}$, whose dimension changes from $\mathbb{R}^{N_\mathrm{p}p\times N_\mathrm{p}p}$ to $\mathbb{R}^{\kappa\times\kappa}$. With this in mind, the optimization problem in (3) is transformed to

$$\boldsymbol{E}_\mathrm{opt}(k) = \arg\min_{\boldsymbol{E}(k)} \|\boldsymbol{V}_\mathrm{L}\boldsymbol{E}(k) - \bar{\boldsymbol{E}}_\mathrm{unc}(k)\|_2^2 \tag{16a}$$
$$\text{subject to } \boldsymbol{\eta}_j(k) \in \mathcal{S}_j, j = 1, \dots, m \tag{16b}$$
$$\text{with } \bar{\boldsymbol{E}}_\mathrm{unc}(k) = -\boldsymbol{V}_\mathrm{L}\boldsymbol{\Omega}^{-1}\boldsymbol{\Psi}(k). \tag{16c}$$

Here, the integer as well as switch constraints are summarized in the sets $\mathcal{S}_j$. The calculation of these sets follows in Section III-A. For now, it is crucial to point out that the optimization variable $\boldsymbol{E}(k)$ is not of the time domain and consequently neither is the optimization process in the LaP-SDA.

This transformation of the optimization problem is visualized in Fig. 1. The transformation blocks to and from the Laguerre based domain are labeled with $\boldsymbol{\Lambda}$ and $\boldsymbol{\Lambda}^{-1}$, respectively. In line with the notation of this section, the transformed optimization problem as the input for the LaP-SDA in Fig. 1 is given by (16). The transformation matrix and blocks are defined by (11) and (13).

III. LAGUERRE POLYNOMIAL BASED SDA

A. Computation of the Admissible Set

The main change introduced by the LaPs in Section II-C are the change of the admissible set of the three-level-inverter from $\mathcal{U}$ to the sets $\mathcal{S}_j$. The computation of the admissible set $\mathcal{S}_j$ now requires the relation between the original input $\boldsymbol{U}(k)$ and alternate input $\boldsymbol{E}(k)$, namely the matrix of LaPs $\boldsymbol{\Lambda}$ as described in (11). Both the number of columns κ of $\boldsymbol{\Lambda}$ and the poles $a_j, j = 1, \dots, m$ can arbitrarily be chosen with respect to $\mathcal{T}$, see (4), and $\mathcal{B}$ defined in (5). To find their optimal values with respect to (3), an auxiliary optimization problem with cost function $\tilde{J}_{\mathrm{L},j}$, for each input j,

$$\boldsymbol{Z}_{j,\mathrm{opt}} = \arg\min_{N_{\mathrm{L},j}, a_j} \sum_{\boldsymbol{u}\in\mathcal{V}} \|\boldsymbol{u} - \boldsymbol{\Lambda}_j(N_\mathrm{L}, a)\boldsymbol{\eta}(\boldsymbol{u})\|_2, \tag{17a}$$
$$\text{subject to } N_{\mathrm{L},j} \in \mathcal{T}, \tag{17b}$$
$$a_j \in \mathcal{B}, \tag{17c}$$
$$\boldsymbol{u} \in \mathcal{V} \tag{17d}$$
$$\text{with } \boldsymbol{\eta}(\boldsymbol{u}) \in \mathcal{S}_j = \{(\boldsymbol{\Lambda}_j^T\boldsymbol{\Lambda}_j)^{-1}\boldsymbol{\Lambda}_j^T\boldsymbol{u}\}, \tag{17e}$$

with $\boldsymbol{Z}_{j,\mathrm{opt}} = \begin{bmatrix}N_{\mathrm{L},j,\mathrm{opt}} & a_{j,\mathrm{opt}}\end{bmatrix}^T$ and $\boldsymbol{\Lambda}_j = \boldsymbol{\Lambda}_j(N_\mathrm{L}, a)$, $\ker\boldsymbol{\Lambda}_j = \{0\}$ is defined. The cost function in (17a) penalizes the summed Euclidean distance of the difference between the true inputs $\boldsymbol{u}$ and the transformed inputs $\boldsymbol{\Lambda}_j(N_\mathrm{L}, a)\boldsymbol{\eta}(\boldsymbol{u})$. The admissible set $\mathcal{V}$ in (17d) comprises all possible switching sequences for $\boldsymbol{u} \in \mathbb{R}^{N_\mathrm{p}}$ which respect the integer (3b) and switch constraints (3c) *except* for $i = k$. Equation (17e) highlights that S_j is the transformed set of $\mathcal{V}$ using the Moore-Penrose inverse because $\boldsymbol{u} = \boldsymbol{\Lambda}_j\boldsymbol{\eta}$ is usually

Prediction Horizon N_p	$\lvert\mathcal{S}\rvert_\mathrm{unc}$	$\lvert\mathcal{S}\rvert_\mathrm{con}$	$\lvert\mathcal{S}\rvert_\mathrm{con}/\lvert\mathcal{S}\rvert_\mathrm{unc}$
1	3	3	1
3	27	17	0.63
5	234	99	0.41
7	2187	577	0.26

TABLE I: Size of Admissible Sets for Different Prediction Horizons for one Input

overdetermined except for $N_{\mathrm{L},j} = N_\mathrm{p}$. As a result, it can be derived that the size of the sets S_j of the LaP-SDA is identical to the number of variations $|\mathcal{V}|$ of $\boldsymbol{u}$.

Here, the solving of (17) and the analysis of the results is done for one input and thus the index j is dropped. The findings can be applied directly to multiple inputs. For one input, Tab. I compares the size of $\mathcal{S}$ of the constrained (index "con") and unconstrained (index "unc") case.

The third column shows that the constraints reduce the size of the set approximately exponentially. This is of importance because the set size has a strong effect on the performance of the LaP-SDA as will be seen in Section IV.

To solve (17) the continuous set $\mathcal{B}$ in (5) is discretized with a step size s_a such that

$$a_{j+1} = a_j + s_\mathrm{a},\ j = 0, \dots, \lfloor(\max(\mathcal{B}) - \min(\mathcal{B}))/s_\mathrm{a}\rfloor.$$

Performing an exhaustive search reveals the optimal set of network parameters $\boldsymbol{Z}_\mathrm{opt} = \begin{bmatrix} N_\mathrm{p} & 0 \end{bmatrix}^T$. This is in line with (7) because with these parameters $\boldsymbol{\Lambda}$ transforms $\boldsymbol{E}(k)$ to the aforementioned set of pulses and the difference in (17a) is minimal. Fig. 2 shows that every configuration $\boldsymbol{Z}$ other than $\boldsymbol{Z}_\mathrm{opt}$, leads to higher costs $\tilde{J}_\mathrm{L}$ and is consequently suboptimal. However, though the costs are minimal, with $\boldsymbol{Z}_\mathrm{opt} = \begin{bmatrix} N_\mathrm{p} & 0 \end{bmatrix}^T$ no dimensional reduction is achieved.

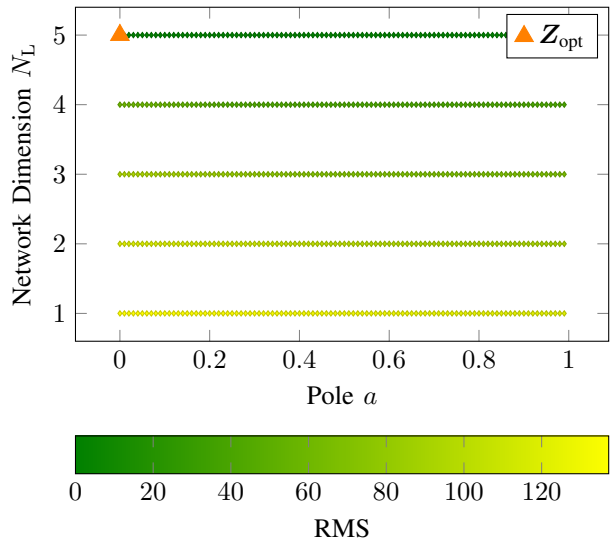


Fig. 2: Cost $\tilde{J}_\mathrm{L}$ for Different Network Parameters for $N_\mathrm{p} = 5$

More relevant are the cases with $N_\mathrm{L} < N_\mathrm{p}$, which result in an optimization problem of lower dimension than the original one. For these, $a = 0$ and the maximum allowed value for N_L are the optimal network configurations. Such configurations are essentially MPC schemes which distinguish between *control* and *prediction* horizon, see for example [11].

Furthermore, Fig. 2 shows that if the constraint $a > 0$ and $N_\mathrm{L} < N_\mathrm{p}$ is introduced, the optimizer shifts the pole a towards one. However, as a consequence, the values in $\boldsymbol{U}_\mathrm{opt}(k)$ are not integers which must be considered. These cases as well as individual network configurations $\boldsymbol{Z}_j, j = 1, \dots, m$ for each input are not investigated. Consequently, for simplicity the index j for $\mathcal{S}$, $\boldsymbol{\Lambda}$, and $\boldsymbol{Z}$ is dropped for the remainder of this paper.

B. Algorithmic Modifications

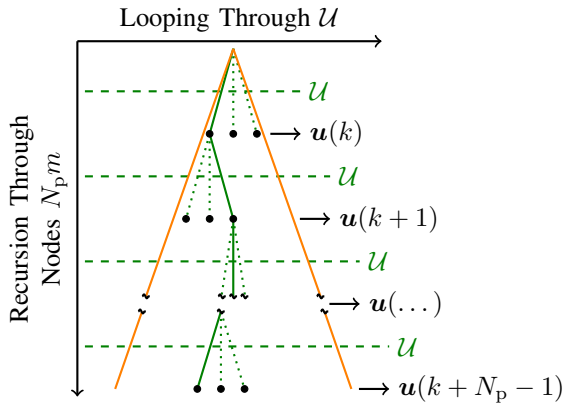
Comparing the original (3) and transformed (16) optimization problems reveals two main changes for the implementation of the LaP-SDA. On the one hand, the admissible set changes from $\mathcal{U}$ to $\mathcal{S}$ for which $|\mathcal{U}| \leq |\mathcal{S}|$ holds true, see Tab. I. On the other hand, the optimization variables are now the Laguerre coefficients $\boldsymbol{\eta}_j(k)$ in $\boldsymbol{E}(k)$, which in combination with $\boldsymbol{\Lambda}$ generate a control trajectory $\boldsymbol{U}(k)$ for the prediction horizon N_p. Consequently, the search for an optimal solution $\boldsymbol{U}_\mathrm{opt}(k)$ by successively adding switch positions in $\boldsymbol{U}(k)$, until a full input sequence has been found, must be modified.

The core actions of the SDA are the loop through the admissible switch positions $\mathcal{U}$ and the continuation with the next item in $\boldsymbol{U}(k)$ if the respective costs are less than the one of the current incumbent. The relation between these two algorithmic actions and the shape of the search tree spanned by the optimization problem is displayed in Fig. 3a for a multiple input system. The height of the search tree is set by the number of elements in $\boldsymbol{U}(k)$, namely $N_\mathrm{p}m$. The elements of $\boldsymbol{U}(k)$, see (8), are marked by the dots. The branches connecting the dots represent the elements of the admissible set $\mathcal{U}$ of the SDA.

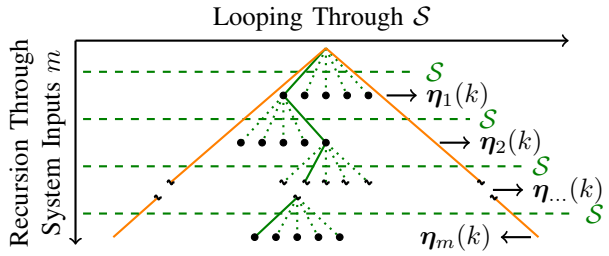
Thus, from an algorithmic perspective, for which the reader is referred to [4], looping through the admissible set $\mathcal{U}$ is a lateral movement through the search tree. In contrast, a vertical movement in the search tree corresponds to a recursive call in the SDA.

Comparing the SDA and the LaP-SDA at the bottom of Fig. 3 shows that the shape of the SDA is narrower and deeper compared to that of the LaP-SDA.

This difference in shape roots back to the difference in the admissible sets and the domain of the optimization variables. As mentioned before, the admissible set $\mathcal{S}$ does not comprise the switch positions, but the transformed variations of $\boldsymbol{u} \in \mathcal{V}$, which results in a larger set. Consequently, the number of branches increases and the result is a wider search tree for the LaP-SDA. This means that instead of switch positions, the LaP-SDA tests the variations with the corresponding Laguerre coefficients $\boldsymbol{\eta}(k)$ of each input to find the optimal solution $\boldsymbol{E}(k)$. In contrast, the search tree is shorter because the recursion is only done over the inputs. Applying the



(a) SDA



(b) LaP-SDA

Fig. 3: Search Trees with Algorithmic Actions

concept of the SDA to the Laguerre based LaP-SDA is justified by the structure of $\boldsymbol{\Lambda}$ and its blocks in (11), which again have a block-diagonal structure [see (13)] in which each block corresponds to only one input.

The pseudo code for the offline computation prior to LaP-SDA is shown in Alg. 1.

It is important to notice that the computation of the variations in $\mathcal{V}$ as well as solving the optimization problem (17) to find the optimal Laguerre network configuration $\boldsymbol{Z}_\mathrm{opt}$ are done offline.

The online component of the LaP-SDA is displayed in Alg. 2. Its fundamental structure is identical to the SDA. In the LaP-SDA, r_1 and r_2 shape the range for $\mathcal{S}$, which respects the switch constraints [see (3c)] between two consecutive time steps $k-1$ and k. This results in a potentially further reduced admissible set compared to the values in Tab. I. The index "sub" indicates that not all items of that vector or matrix are used or set. Omitting the particular indices is done for better readability.

C. Theoretical Performance

As a consequence of the aforementioned change of the admissible set and the number of recursions, common metrics to assess the performance of the SDA and LaP-SDA must be defined. Geyer [4] defined the number of explored nodes N_N as a key index for the performance evaluation. The minimum number of explored nodes, i.e. the lower bound (LB),

$$N_{\mathrm{N,R,LB}}(m, N_\mathrm{p}) = N_\mathrm{p} m \tag{18}$$

Algorithm 1 Offline Computations LaP-SDA

function $\mathcal{V}$ = VARIATIONS (N_p)
for $j = \begin{bmatrix}1 & 2 & \dots & m\end{bmatrix}$ **do**
  for $a \in \mathcal{B}$ **do**
    for $N_\mathrm{L} \in \mathcal{T}$ **do**
      function $\boldsymbol{\Lambda}$ = LAMBDA (a, N_L) ▷ (11), (13)
      function $[\mathcal{S}, \tilde{J}_\mathrm{L}]$ = ADMSET ($\mathcal{V}, \boldsymbol{\Lambda}$)
        $\boldsymbol{\eta}(\boldsymbol{u}) = (\boldsymbol{\Lambda}^T\boldsymbol{\Lambda})^{-1}\boldsymbol{\Lambda}^T\boldsymbol{u}$
        $\tilde{J}_\mathrm{L} = \sum_{\boldsymbol{u}\in\mathcal{V}} \|\boldsymbol{u} - \boldsymbol{\Lambda}(N_\mathrm{L}, a)\boldsymbol{\eta}(\boldsymbol{u})\|_2$
      end
      if $\tilde{J}_\mathrm{L} < J_\mathrm{opt}$ **then**
        $a_\mathrm{opt} = a$
        $N_{\mathrm{L,opt}} = N_\mathrm{L}$
        $J_\mathrm{opt} = \tilde{J}_\mathrm{L}$
      end if
    end for
  end for
end for
function $[\boldsymbol{V}_\mathrm{L}, \boldsymbol{\Omega}]$ = AUXILIARYMATRICES (...)

for the SDA and

$$N_{\mathrm{N,L,LB}}(m) = m \tag{19}$$

for the LaP-SDA to find an optimal solution.

Neglecting the switch constraints, in the worst case, i.e. upper bound (UB), for the SDA

$$N_{\mathrm{N,R,UB}}(m, N_\mathrm{p}) = \sum_{i=0}^{N_\mathrm{p}m-1} |\mathcal{U}|^i. \tag{20}$$

Analogous, for the worst case of the LaP-SDA

$$N_{\mathrm{N,L,UB}}(m, N_\mathrm{p}) = \sum_{i=0}^{m-1} |\mathcal{S}(N_\mathrm{p})|^i, \tag{21}$$

Algorithm 2 Online Computations LaP-SDA

function $[\boldsymbol{E}_\mathrm{opt}, \rho^2]$ = LAP-SDA (r_1, r_2, $\mathcal{S}$, ρ^2, $\bar{\boldsymbol{E}}_\mathrm{unc}$, $\boldsymbol{V}_\mathrm{L}$, j, m)
  for $k = r_1$ **to** r_2 **do**
    $\boldsymbol{E}_\mathrm{sub,cand} = \mathcal{S}(k)$
    $\rho^2_\mathrm{cand} = \|\boldsymbol{V}_{\mathrm{L,sub}}\boldsymbol{E}_\mathrm{sub,\ cand} - \bar{\boldsymbol{E}}_\mathrm{unc}\|_2^2$
    if $\rho^2_\mathrm{cand} < \rho^2$ **then**
      if $j < m$ **then**
        function $[\boldsymbol{E}_\mathrm{opt}, \rho^2]$ = LAP-SDA (r_1, r_2, $\mathcal{S}$, ρ^2, $\bar{\boldsymbol{E}}_\mathrm{unc}$, $\boldsymbol{V}_\mathrm{L}$, $j+1$, m)
      else
        $\boldsymbol{E}_\mathrm{sub,opt} = \boldsymbol{E}_\mathrm{sub,\ cand}$
        $\rho^2 = \rho^2_\mathrm{cand}$
      end if
    end if
  end for
end

in which the upper bound of the sum m and the base changed compared to (20). It can be seen that both, (20) and (21), depend on the number of inputs m and the prediction horizon N_p, but in different ways. For the SDA, N_p and m affect only the upper bound of the sum and thus the maximum exponent i. Conversely, for the LaP-SDA in (21) the prediction horizon scales the size of the admissible set $\mathcal{S}$, which results in a base that is greater than or equal to the base in the SDA (20). This circumstance corresponds to the shape of the search trees in Fig. 3 directly; the larger the base, the wider the search tree. Furthermore, the search tree's depth is directly linked to the upper bound of the sums and thus the search tree for the SDA is much deeper than the one of the LaP-SDA as already discussed in Section III-B.

The analysis of (20) and (21) for different prediction horizons with $m = 3$ is displayed in Fig. 4a. It can be seen that the lower bound of explored nodes for the LaP-SDA equals m. This means that in the ideal case of finding the optimal solution directly, the LaP-SDA outperforms the SDA for $N_\mathrm{p} > 1$. The same applies to the worst case scenario in which the LaP-SDA must try all possible solutions in which the LaP-SDA explores at least one order of magnitude fewer nodes than the SDA.

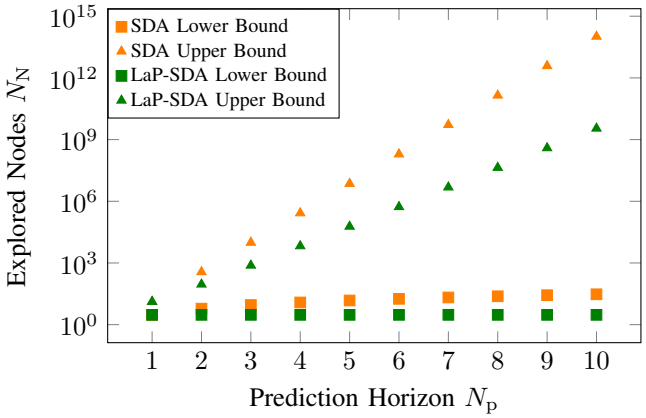
For the number of explored branches, the lower bound of explored branches

$$N_{\mathrm{B,R,LB}}(N_\mathrm{p}, m) = N_\mathrm{p} m |\mathcal{U}| \tag{22}$$

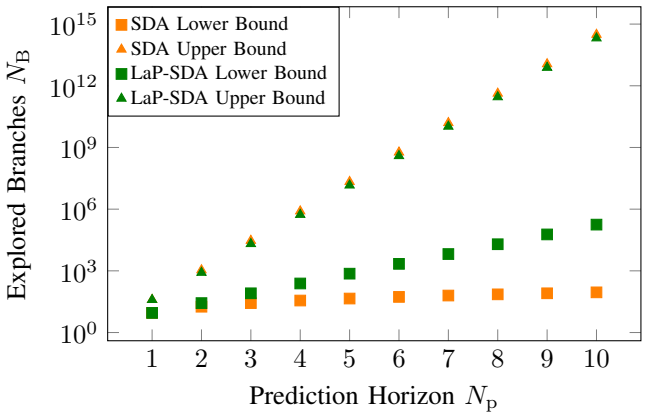
and

$$N_{\mathrm{B,L,LB}}(m) = m|\mathcal{S}| \tag{23}$$

for the SDA and the LaP-SDA. The upper bound of branches is calculated with (20) and (21) as well, but with the sums lower bounds $i = 1$ and upper bounds $N_\mathrm{p} m$ and m, respectively. The respective values are plotted in Fig. 4b and show that the lower bound of explored branches of the LaP-SDA is significantly higher than the one of the SDA. This is a consequence of the larger base, i.e. set, of the LaP-SDA. However, the upper bound for the approaches do not differ significantly.



(a) Explored Nodes



(b) Explored Branches

Fig. 4: Theoretical Lower and Upper Bounds for the Performance of SDA and LaP-SDA

IV. SIMULATION RESULTS

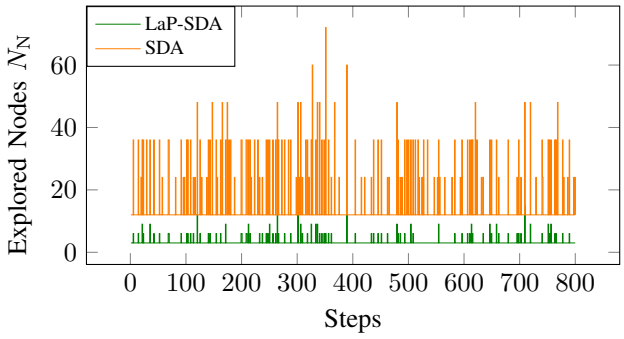
A. Simulation Setup

Since it was found that the minimum number of explored nodes for the LaP-SDA indicates a performance benefit although it must explore significantly more branches, simulations were performed in order to evaluate the significance of these differences and their impact on the actual performance. The system setup comprises an induction machine linked to a three-level inverter, which was implemented and subsequently simulated. It is assumed that the full state information is available and is directly fed back to the MPC within the sampling time. There is no observer or any simulation of the rotor. The state space matrices and machine parameters are taken from [4] with $\boldsymbol{x} = \begin{bmatrix} i_\alpha & i_\beta & \psi_\alpha & \psi_\beta \end{bmatrix}^T$, $\boldsymbol{u} = \begin{bmatrix} u_1 & u_2 & u_3 \end{bmatrix}^T$ and $\boldsymbol{y} = \begin{bmatrix} i_\alpha & i_\beta \end{bmatrix}^T$. The elements in $\boldsymbol{u}$ correspond to the switching state of each phase, consequently $u_i \in \mathcal{U} = \{-1, 0, 1\}, i = 1, 2, 3$. The Laguerre polynomials configuration is $a = 0$ and $N_\mathrm{L} = N_\mathrm{p}$, which does not result in a dimensional reduction of the optimization problem but yields the optimal solution found by the SDA. The weight $\lambda_\mathrm{u} = 6 \times 10^{-3}$ and $\boldsymbol{Q}$ is in this case set equal to the identity matrix. To introduce parameter uncertainties of the plant, each item in the state space matrices for the MPC is disturbed slightly by a normally drawn factor $k_\mathrm{d} \sim \mathcal{N}(1, 1 \times 10^{-4})$.

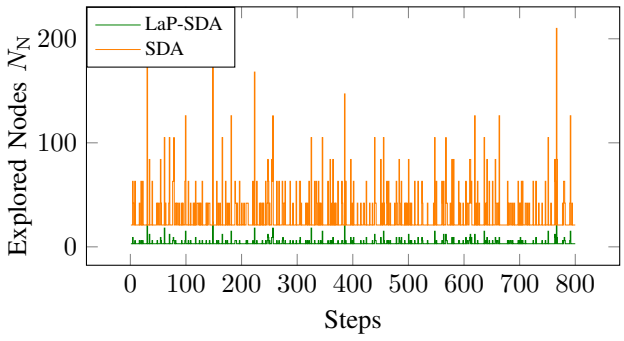
Moreover, the sampling time of the plant and controller $s = 25 \times 10^{-6}$ s. The simulation is done for $N_\mathrm{p} = 1, \ldots, 7$ with a 50 Hz reference signal of amplitude $\hat{a} = 0.7$ in the per unit system. Moreover, it is ensured that the system has reached steady state before measuring the number of explored nodes and branches for eight fundamentals.

B. Performance Evaluation

Due to the Laguerre polynomials configuration, the results for the optimal trajectories found by the SDA and LaP-SDA are identical. For other configurations, e.g. with distinct sampling times for controller and plant, the LaP-SDA yields different results for $\boldsymbol{U}_\mathrm{opt}$. However, this discrepancy between

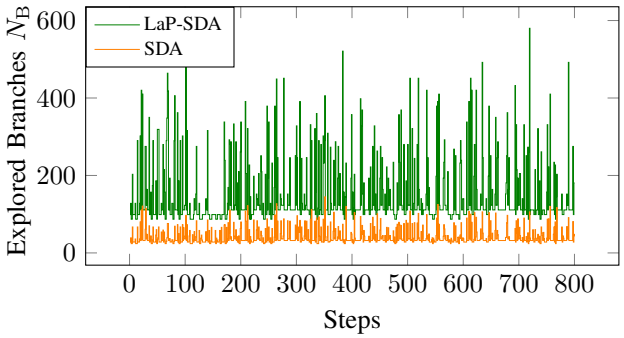


(a) $N_\mathrm{p} = 4$

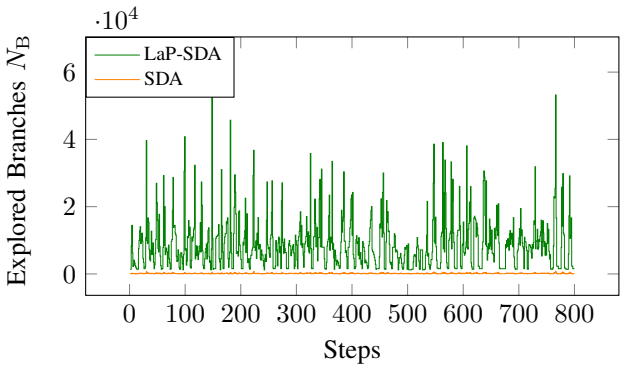


(b) $N_\mathrm{p} = 7$

Fig. 5: Explored Nodes N_N for One Fundamental Period



(a) $N_\mathrm{p} = 4$



(b) $N_\mathrm{p} = 7$

Fig. 6: Explored Branches N_B for One Fundamental Period

the SDA and LaP-SDA has only been noticed by the authors to occur in a handful of configurations. As a consequence, the resulting currents are not focused on, but the explored nodes and branches. The explored nodes for one fundamental period and prediction horizons $N_\mathrm{p} = 4$ and $N_\mathrm{p} = 7$ are displayed in Fig. 5. It can be seen that for both prediction horizons the number of explored nodes of the LaP-SDA and SDA differ significantly, with the mean value obtained with the LaP-SDA being smaller by factors $k_{N_\mathrm{N},N_\mathrm{p}=4} = 5$ and $k_{N_\mathrm{N},N_\mathrm{p}=7} = 8$ with respect to the means of the SDA. Moreover, for both SDA and LaP-SDA, the lower bounds are clearly visible. In particular, the direct scaling of the lower bound $N_\mathrm{N,R,LB}$ with the prediction horizon compared the lower bound $N_\mathrm{N,L,LB}$ can be observed, Fig. 5b.

However, examining the number of explored branches in Fig. 6 reveals a significantly worse performance of the LaP-SDA for both prediction horizons. It is important to notice that the means of the LaP-SDA and SDA differ by factors $k_{N_\mathrm{B},N_\mathrm{p}=4} = 4$ and $k_{N_\mathrm{B},N_\mathrm{p}=7} = 64$ in favor of the SDA.

Moreover, granulating these findings for all prediction horizons, presented in Fig. 7, reveals that this observation holds for all investigated prediction horizons. There, the explored nodes and branches for all eight fundamentals are evaluated.

This shows, that the LaP-SDA can not maintain its advantage resulting from fewer explored nodes. The inferior performance of the LaP-SDA for the number of explored branches shows that the LaP-SDA cannot utilize the sphere decoding approach as well as the SDA. The general performance benefit of the SDA compared to, for example, an exhaustive search, roots back to the successive supplementation of the candidate solution $\boldsymbol{U}_\mathrm{cand}(k)$, while not considering suboptimal candidates and the associated solutions early in the optimization process. Due to the more shallow search tree of the LaP-SDA, ignoring suboptimal candidates is not possible as often as in the SDA. This is not compensated for by the shorter search tree and consequently fewer items in the optimization variable $\boldsymbol{E}(k)$ and fewer explored nodes. Though, the LaP-SDA still remains close to its lower bound of the explored branches. Moreover, the LaP-SDA does not benefit from a well suited initial candidate and cost as much as the SDA since it still must try a significant number of branches nevertheless. These issues origin from the generally higher lower bound of the LaP-SDA for the explored branches which grows exponentially with the prediction horizon, see (23) and Tab. I.

V. CONCLUSION

The results of the previous section showed that the LaP-SDA does not prove beneficial compared to the SDA. The theoretical benefit of the LaP-SDA was refuted by the simulation results. Consequently, the approach presented in the current paper for the LaP-SDA with this system, controller, and Laguerre configuration, is not competitive compared to the SDA. The main challenge introduced with LaP-SDA

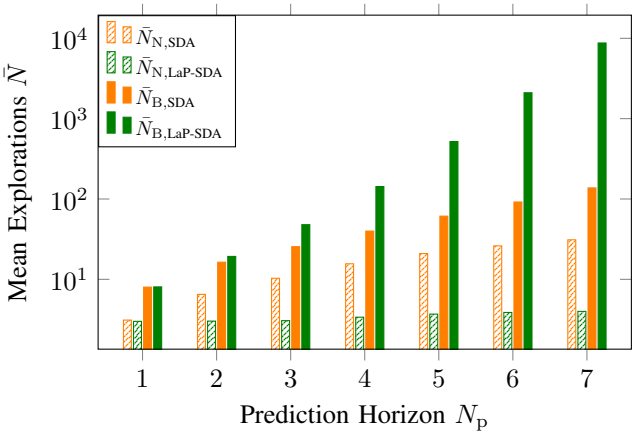


Fig. 7: Mean of Explored Nodes $\bar{N}_{\mathrm{N}}$ and Branches $\bar{N}_{\mathrm{B}}$ over Eight Fundamentals for Different Prediction Horizons N_{p}

is to tackle the size of the admissible set $\mathcal{S}$ by reducing the explored branches. However, it has to be be considered that, with the setup used here, the LaPs were purposely not configured to achieve any dimensional reduction.

Consequently, further prospective research continuing the current work can be identified. First, it must investigated under which configurations the LaP-SDA does not yield the optimal solution found by an exhaustive search or the SDA. Since this occurs only in certain configurations, reasons for this could for example be numerical errors. Further, the research could be applied to four different areas: 1) controlled systems; 2) control parameters; 3) algorithmic modifications; and 4) Laguerre network configuration. From a system perspective, it is of interest to investigate more sophisticated load scenarios, covering for example transient behavior, stronger signal disturbances or model uncertainties. These aspects could result in a stronger difference between the optimal solution of the previous and current input sequence from which the LaP-SDA could benefit compared to the SDA, since the latter profits from a well suited initial guess. Moreover, the LaP-SDA performance for systems with inputs $m > 3$ should be examined, because this leads to a relatively deeper search tree for the SDA than for the LaP-SDA, see Fig. 3.

Regarding the control parameters, for modified weights $\lambda_{\mathrm{u}}, \boldsymbol{Q}$ allowing for potentially more switching actions, the previously mentioned disadvantage of the LaP-SDA could be alleviated since this usually introduces more explorations to find the optimal solution.

Algorithmic modifications could include investigations on reducing the size of the admissible set by ignoring elements leading to suboptimal solutions in the optimization process of the LaP-SDA. This aspect covers modifications of the search to resemble the SDA and profiting of ignoring branches.

Lastly, other Laguerre configurations should be examined. Namely, this could cover $a = 0$ and the Laguerre network dimension less than the prediction horizon, $N_{\mathrm{L}} < N_{\mathrm{p}}$. It could be investigated how a dimensional reduction of the optimization problem using the Laguerre network parameters influences the performance of the LaP-SDA. Furthermore, extending this idea to the network poles unequal zero as well would also require strategies to compute integer values.

ACKNOWLEDGMENT

For his contributions and support the authors would like to deeply thank Dr. Sascha D. Wolff. His insights laid the foundation for the work hereby presented.

REFERENCES

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

[2] R. Baidya, R. P. Aguilera, P. Acuna, S. Vazquez, and H. T. Du Mouton, "Multistep model predictive control for cascaded h-bridge inverters: Formulation and analysis," *IEEE Transactions on Power Electronics*, vol. 33, no. 1, pp. 876–886, 2018.

[3] A. Andersson and T. Thiringer, "Assessment of an improved finite control set model predictive current controller for automotive propulsion applications," *IEEE Transactions on Industrial Electronics*, vol. 67, no. 1, pp. 91–100, 2020.

[4] T. Geyer, *Model predictive control of high power converters and industrial drives*, 1st ed. Chichester, West Sussex, United Kingdom: John Wiley & Sons Incorporated, 2017.

[5] S. Kouro, P. Cortes, R. Vargas, U. Ammann, and J. Rodriguez, "Model predictive control—a simple and powerful method to control power converters," *IEEE Transactions on Industrial Electronics*, vol. 56, no. 6, pp. 1826–1838, 2009.

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

[7] D. E. Quevedo, G. C. Goodwin, and J. A. de Doná, "Finite constraint set receding horizon quadratic control," *International Journal of Robust and Nonlinear Control*, vol. 14, no. 4, pp. 355–377, 2004.

[8] T. Geyer and D. E. Quevedo, "Multistep direct model predictive control for power electronics — part 2: Analysis," in *2013 IEEE Energy Conversion Congress and Exposition*, 2013, pp. 1162–1169.

[9] B. Stellato, V. V. Naik, A. Bemporad, P. Goulart, and S. Boyd, "Embedded mixed-integer quadratic optimization using the osqp solver," in *2018 European Control Conference (ECC)*, 2018, pp. 1536–1541.

[10] 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 Journal of the Industrial Electronics Society*, vol. 4, pp. 159–175, 2023.

[11] L. Wang, *Model predictive control system design and implementation using MATLAB*, ser. Advances in industrial control. London: Springer, 2009.

[12] V. K. Manoj and J. Jacob, "On the design of mpc using laguerre functions for formation tracking control of multiple wheeled mobile robots," in *2022 3rd International Conference for Emerging Technology (INCET)*, 2022, pp. 1–6.

[13] W. Zhang, Q. Wang, W. Wu, and P. Han, "Mpc for 3-d trajectory tracking of uuv with constraints using laguerre functions," in *OCEANS 2022, Hampton Roads*, 2022, pp. 1–6.

[14] G. Mandyam and N. Ahmed, "The discrete laguerre transform: derivation and applications," *IEEE Transactions on Signal Processing*, vol. 44, no. 12, pp. 2925–2931, 1996.

[15] Y. Feng, L. Wang, and W. Luo, "Laguerre functions based nonlinear model predictive control using multi-model approach," in *34th Annual Conference of IEEE Industrial Electronics, 2008*. IEEE, 2008, pp. 247–252.