A Lyapunov Approach to Nonlinear Programming and Its Application to Nonlinear Model Predictive Control

Kyunghwan Choi and Christoph M. Hackl, *Senior Member*

***Abstract*—This study presents a tuning-parameter-free and matrix-inversion-free method to solve nonlinear programming (NLP). The key idea is to design an update law based on Lyapunov analysis to approach the first-order necessary conditions for optimality. To this aim, first, the Lyapunov function is defined as the summation of the norms of these conditions. Then, the desired steps of optimization variables and Lagrange multipliers, which minimize the Lyapunov function the most, are analytically found, thereby rapidly approaching the necessary conditions. The proposed method does not include tuning parameters or matrix inversion; thus, it can be implemented easily with less iteration and computation time than conventional methods, such as sequential quadratic programming (SQP) and augmented Lagrangian method (ALM). The effectiveness of the proposed method is validated by using it to solve a nonlinear model predictive control (NMPC) problem and comparing it with SQP and ALM.**

***Index Terms*—Lyapunov analysis, necessary conditions for optimality, matrix-inversion-free, nonlinear model predictive control (NMPC), nonlinear programming (NLP), tuning-parameter-free**

I. Introduction

Nonlinear programming (NLP) has been used in various applications as a powerful tool for optimizing the performance of target systems. However, solving NLP problems is still challenging because of the lack of a general analytical solution and the difficulty in designing effective numerical optimization processes.

Typical numerical approaches for NLP are sequential quadratic programming (SQP) and augmented Lagrangian method (ALM). SQP solves an NLP problem effectively when the NLP can be reasonably approximated as QPs and the iteration starts near the optimal solution. However, SQP is computationally expensive, especially for large-scale problems, because it involves the inversion of the Karush-Kuhn-Tucker (KKT) matrix. In addition, SQP does not allow violation of constraints and thus may struggle to find a feasible solution when the starting points are infeasible or the NLP includes many inequality constraints.

By contrast, ALM allows the violation of constraints during the iteration process and thus can efficiently handle infeasible starting points and inequality constraints. However, ALM uses multiple tuning parameters regarding handling the constraints, such as the barrier parameter and the constraint violation tolerance. Finding appropriate values for these parameters can be challenging and may require problem-specific tuning. In addition, ALM typically involves either the inversion of the Hessian or the approximation of the inverted Hessian, which is sometimes computationally demanding.

Most numerical approaches including SQP and ALM involve computationally demanding processes, such as the inversion of the KKT matrix or Hessian and the approximation of the inverted Hessian, and use multiple tuning parameters, which may require problem-specific tuning or a large number of iterations to be optimized. Therefore, this study presents a tuning-parameter-free and matrix-inversion-free numerical optimization method to solve NLP. To this aim, a novel perspective is adopted that interprets the numerical optimization process as a dynamical system so that control principles can be used to design an update law with the two desirable freeness. The key idea is to design an update law based on Lyapunov analysis to approach the first-order necessary conditions for optimality. The Lyapunov function is defined as the summation of the norms of these conditions. Then, the desired steps of optimization variables and Lagrange multipliers, which minimize the Lyapunov function the most, are analytically found, thereby rapidly approaching the necessary conditions.

The proposed method can be implemented easily with less iteration and computation time than conventional methods, such as SQP and ALM, due to its two desirable features; thus, it is particularly effective for nonlinear model predictive control (NMPC), which requires an NLP problem to be solved in real time within a short control period. The proposed method also allows the violation of constraints during the iteration process as ALM does, which is another desirable feature for NMPC with many inequality constraints. The update law of the proposed method can be directly used as the control law for NMPC because each iteration is at least suboptimal and moves toward the local solution rapidly.

To the authors' knowledge, using a Lyapunov approach to NLP has not been investigated except one attempt to prove

Manuscript received Month xx, 2xxx; revised Month xx, xxxx; accepted Month x, xxxx. This work was supported in part by the xxx Department of xxx under Grant (sponsor and financial support acknowledgment goes here).

(Authors' names and affiliation) First A. Author1 and Second B. Author2 are with the xxx Department, University of xxx, City, Zip code, Country, on leave from the National Institute for xxx, City, Zip code, Country (e-mail: author@domain.com).

Third C. Author3 is with the National Institute of xxx, City, Zip code, Country (corresponding author to provide phone: xxx-xxx-xxxx; fax: xxx-xxx-xxxx; e-mail: author@ domain.gov).

convergence of Newton's method via Lyapunov Analysis [1].

II. Preliminaries

This section provides preliminaries for this study: Section II-A defines a general formulation of NLP, Section II-B states necessary conditions for optimality of NLP, and Section II-C describes typical numerical approaches to NLP.

A. NLP

A general formulation of NLP is

$$\min_{x\in\mathbb{R}^n} f(x) \tag{1a}$$
$$\text{subject to}$$
$$c_i(x)=0,\ i\in\mathcal{E}, \tag{1b}$$
$$c_i(x)\geq 0,\ i\in\mathcal{I}, \tag{1c}$$

where f and the functions c_i are all real-value functions on a subset of $\mathbb{R}^n$, and $\mathcal{E}$ and $\mathcal{I}$ are two finite sets of indices. The function f is the objective function, $c_i, i\in\mathcal{E}$ are the equality constraints, $c_i, i\in\mathcal{I}$ are the inequality constraints, and x is the vector of optimization variables. The objective function or some of the constraints are nonlinear for (1) to be NLP. This study assumes the function f and c_i are twice continuously differentiable.

B. Necessary Conditions for Optimality

The Lagrangian for the NLP (1) is defined as

$$\mathcal{L}(x,\lambda)=f(x)-\sum_{i\in\mathcal{E}\cup\mathcal{I}}\lambda_i c_i(x), \tag{2}$$

where λ_i are Lagrange multipliers for the constraints $c_i(x)=0, i\in\mathcal{E}$ or $c_i(x)\geq 0, i\in\mathcal{I}$, and λ is the vector of Lagrange multipliers. The active set $\mathcal{A}(x)$ at any feasible x is the union of the set $\mathcal{E}$ with the indices of the active inequality constraints; that is,

$$\mathcal{A}(x)=\mathcal{E}\cup\{i\in\mathcal{I}\mid c_i(x)=0\}. \tag{3}$$

One constraint qualification for the necessary conditions is defined as follows:

Definition 1. *Given a feasible point x and the active set $\mathcal{A}(x)$, the linear independence constraint qualification (LICQ) holds if the set of active constraint gradients $\{\nabla c_i(x), i\in\mathcal{A}(x)\}$ is linearly independent.*

The first-order necessary conditions for optimality, which provide the foundation for many of the numerical algorithms for NLP, are defined as follows.

Theorem 1 (First-Order Necessary Conditions)**.** *Suppose that x^* is a local solution of (1) and that the LICQ holds at x^*. Then there is a Lagrange multiplier vector λ^*, with components $\lambda_i^*, i\in\mathcal{E}\cup\mathcal{I}$, such that the following conditions are satisfied at (x^*,λ^*):*

$$\nabla_x\mathcal{L}(x^*,\lambda^*)=0, \tag{4a}$$
$$c_i(x^*)=0,\ \text{for all } i\in\mathcal{E}, \tag{4b}$$
$$c_i(x^*)\geq 0,\ \text{for all } i\in\mathcal{I}, \tag{4c}$$
$$\lambda_i^*\geq 0,\ \text{for all } i\in\mathcal{I}, \tag{4d}$$
$$\lambda_i^* c_i(x^*)=0,\ \text{for all } i\in\mathcal{E}\cup\mathcal{I}. \tag{4e}$$

The conditions (4) are known as the KKT conditions. The condition (4e) implies that the Lagrange multipliers corresponding to inactive inequality constraints are zero; thus, the first condition (4a) can be rewritten by omitting the terms for indices $i\notin\mathcal{A}(x^*)$ as follows:

$$0=\nabla_x\mathcal{L}(x^*,\lambda^*)=\nabla f(x^*)-\sum_{i\in\mathcal{A}(x^*)}\lambda_i^*\nabla c_i(x^*). \tag{5}$$

C. Typical Numerical Approaches to NLP

Two typical numerical approaches, SQP and ALM, are described below.

1) SQP: SQP is an iterative method for NLP, solving a sequence of optimization subproblems, each of which is an approximation of the NLP as a quadratic programming (QP). The subproblem is defined as follows:

$$\min_{p_x\in\mathbb{R}^n} f(x^k)+\nabla f(x^k)^T p_x+\frac{1}{2}p_x^T W(x^k,\lambda^k)p_x \tag{6a}$$
$$\text{subject to}$$
$$\nabla c_i(x^k)^T p_x+c_i(x^k)=0,\ i\in\mathcal{E}, \tag{6b}$$
$$\nabla c_i(x^k)^T p_x+c_i(x^k)\geq 0,\ i\in\mathcal{I}, \tag{6c}$$

where $p_x=x^{k+1}-x^k$ and W denotes the Hessian of the Lagrangian: $W(x,\lambda)=\nabla_{xx}^2\mathcal{L}(x,\lambda)$. Because the Hessian may not be easy to compute due to its second derivative and may not always be positive definite on the constraint null space, alternate choices for W, such as full quasi-newton approximations and reduced-hessian approximations, can be used instead.

The solution of (6) can be obtained by solving the following KKT matrix:

$$\begin{bmatrix} W(x^k,\lambda^k) & -C_{\mathcal{A}}(x^k)^T \\ C_{\mathcal{A}}(x^k) & 0\end{bmatrix}\begin{bmatrix} p_x \\ p_\lambda\end{bmatrix}=\begin{bmatrix} -\nabla_x L(x^k,\lambda^k) \\ -c_{\mathcal{A}}(x^k)\end{bmatrix}, \tag{7}$$

where $p_\lambda=\lambda^{k+1}-\lambda^k$, matrix $C_{\mathcal{A}}(x)\in\mathbb{R}^{m\times n}$ defined by $C_{\mathcal{A}(x)}^T:=[\nabla c_i(x)]_{i\in\mathcal{A}(x)}$, and vector $c_{\mathcal{A}}(x)\in\mathbb{R}^{m\times 1}$ is defined by $c_{\mathcal{A}}(x):=[c_i(x)]_{i\in\mathcal{A}(x)}$, where m is the number of active constraints. This iteration is well-defined when the non-singularity of the KKT matrix holds, which is a consequence of the following conditions:

- The LICQ holds.
- The Hessian is positive definite on the constraint null space.

SQP solves the NLP effectively when each subproblem approximates the NLP reasonably. However, solving the KKT matrix (7) involves the inversion of matrix, which is computationally expensive. In addition, SQP does not allow violation of constraints and thus may struggle to find a feasible solution when the starting points are infeasible or the NLP includes many inequality constraints.

2) ALM: ALM is an iterative method for NLP, which combines aspects of both Lagrange multipliers and penalty methods to handle constraints by defining the augmented Lagrangian function as follows:

$$\begin{aligned}\mathcal{L}_A(x,\lambda^k;\mu^k) := f(x) - \sum_{i\in\mathcal{E}}\lambda_i^k c_i(x) + \frac{1}{2\mu}\sum_{i\in\mathcal{E}} c_i^2(x)\\ + \sum_{i\in\mathcal{I}}\psi(c_i(x),\lambda_i^k;\mu^k),\end{aligned}\tag{8}$$

where λ^k is the current estimate of λ^* with components $\lambda_i^k, i \in \mathcal{E}\cup\mathcal{I}$, μ_k is the barrier parameter, and

$$\psi(t,\sigma;\mu) := \begin{cases} -\sigma t + t^2/(2\mu) & \text{if } t-\mu\sigma \le 0,\\ -\mu\sigma^2/2 & \text{otherwise.}\end{cases}\tag{9}$$

Note that the augmented Lagrangian function $\mathcal{L}_A$ differs from the standard Lagrangian (2) for (1) by the presence of the squared terms.

The vector of optimization variables, x, is updated to minimize the augmented Lagrangian function $\mathcal{L}_A$ for given λ^k and μ^k as

$$\min_x \mathcal{L}_A(x,\lambda^k;\mu^k).\tag{10}$$

Methods of unconstrained optimization, such as Newton and quasi-Newton methods, can be used in this process. Then, the estimates of the Lagrange multipliers, λ_i^k, are updated based on the extent of constraint violation as follows:

$$\lambda_i^{k+1} = \lambda_i^k - c_i(x^k)/\mu^k, \quad \text{for all } i\in\mathcal{E}, \tag{11a}$$
$$\lambda_i^{k+1} = \max(\lambda_i^k - c_i(x^k)/\mu^k, 0), \quad \text{for all } i\in\mathcal{I}. \tag{11b}$$

The barrier parameter μ^k is adjusted during the iterations to balance convergence and numerical stability.

This separate update of optimization variables and Lagrange multipliers can lead to more efficient and scalable solutions. Particularly, this approach allows the violation of constraints during the iteration process and thus can efficiently handle infeasible starting points and inequality constraints. However, ALM uses multiple tuning parameters regarding handling the constraints, such as the barrier parameter and others used for practical implementation. Finding appropriate values for these parameters can be challenging and may require problem-specific tuning. In addition, ALM typically involves either the inversion of the Hessian or the approximation of the inverted Hessian, which is sometimes computationally demanding.

III. Lyapunov Approach to NLP

This section presents a Lyapunov approach for solving NLP: Section III-A presents an update law devised based on Lyapunov analysis with proof of convergence, Section III-B provides an algorithm for implementing the proposed method, and Section III-C describes properties of the proposed method, compared with SQP and ALM.

A. Update law

Define the vector of Lagrange multipliers of active constraints as $\lambda_\mathcal{A} := [\lambda_i]_{i\in\mathcal{A}(x)} \in \mathbb{R}^{m\times 1}$. The desired steps of x and $\lambda_\mathcal{A}$ are computed by

$$\Delta x = \alpha p_x, \tag{12a}$$
$$\Delta\lambda_\mathcal{A} = \alpha p_\lambda, \tag{12b}$$

with the step length α and the search directions p_x and p_λ:

$$p_x = -\left(W^T g + C_\mathcal{A}^T c_\mathcal{A}\right)/2, \tag{13a}$$
$$p_\lambda = -(C_\mathcal{A}^T)^+ W C_\mathcal{A}^T c_\mathcal{A} + C_\mathcal{A} g/2, \tag{13b}$$
$$p_g = W p_x - C_\mathcal{A}^T p_\lambda, \tag{13c}$$
$$p_{c_\mathcal{A}} = C_\mathcal{A} p_x, \tag{13d}$$
$$\alpha = -\frac{p_g^T g + p_{c_\mathcal{A}}^T c_\mathcal{A}}{p_g^T p_g + p_{c_\mathcal{A}}^T p_{c_\mathcal{A}}}, \tag{13e}$$

where g denotes $\nabla_x L(x,\lambda)$ and the inputs of functions $W(x,\lambda)$, $g(x,\lambda)$, $C_\mathcal{A}(x)$, and $c_\mathcal{A}(x)$ are omitted here for simplicity.

Theorem 2 states the convergence of the update law (12) as follows.

Theorem 2. *Suppose the iteration to find the local solution (x^*,λ^*) starts sufficiently close to the solution. Then, the update law (12) guarantees the convergence to the solution with some α.*

Proof. Define the Lyapunov function as

$$V := \frac{1}{2}g^T g + \frac{1}{2}c_\mathcal{A}^T c_\mathcal{A}. \tag{14}$$

The perturbation in the Lyapunov function under perturbations in x and $\lambda_\mathcal{A}$ is derived as

$$\begin{aligned}\Delta V = \frac{1}{2}(g+\Delta g)^T(g+\Delta g)\\ + \frac{1}{2}(c_\mathcal{A}+\Delta c_\mathcal{A})^T(c_\mathcal{A}+\Delta c_\mathcal{A}) - V\end{aligned}\tag{15a}$$
$$= g^T\Delta g + c_\mathcal{A}^T\Delta c_\mathcal{A} + \frac{1}{2}\Delta g^T\Delta g + \frac{1}{2}\Delta c_\mathcal{A}^T\Delta c_\mathcal{A} \tag{15b}$$

The perturbation ΔV is approximated as

$$\Delta V \approx \Delta V_0 := g^T\Delta g + c_\mathcal{A}^T\Delta c_\mathcal{A} \tag{16}$$

to derive the search directions in a simpler form, where

$$\Delta g = W\Delta x - C_\mathcal{A}^T\Delta\lambda_\mathcal{A} := \alpha p_g, \tag{17a}$$
$$\Delta c_\mathcal{A} = C_\mathcal{A}\Delta x := \alpha p_{c_\mathcal{A}}, \tag{17b}$$

Putting (17) into (16) yields

$$\begin{aligned}\Delta V_0 = -\alpha g^T\left(\frac{WW^T + C_\mathcal{A}^T C_\mathcal{A}}{2}\right)g - \alpha c_\mathcal{A}^T\left(\frac{C_\mathcal{A}C_\mathcal{A}^T}{2}\right)c_\mathcal{A}\\ + \alpha g^T\left(C_\mathcal{A}^T(C_\mathcal{A}^T)^+ - I\right)WC_\mathcal{A}^T c_\mathcal{A}\end{aligned}\tag{18a}$$
$$< -\alpha g^T\frac{C_\mathcal{A}^T C_\mathcal{A}}{2}g \tag{18b}$$
$$\le 0. \tag{18c}$$

Consequently, $\Delta V_0 < 0$, which proves a descent property of p_x and p_λ. In the derivation of (18b) from (18a), the following inequality was applied:

$$\begin{aligned}\left\|g^T\left(C_\mathcal{A}^T(C_\mathcal{A}^T)^+ - I\right)WC_\mathcal{A}^T c_\mathcal{A}\right\| < \left\|g^T W\right\|\left\|C_\mathcal{A}^T c_\mathcal{A}\right\|\\ \le g^T\left(\frac{WW^T}{2}\right)g + c_\mathcal{A}^T\left(\frac{C_\mathcal{A}C_\mathcal{A}^T}{2}\right)c_\mathcal{A}\end{aligned}\tag{19}$$

The last inequality (18c) holds because $C_\mathcal{A}^T C_\mathcal{A}$ is positive semidefinite.

Equation (15b) is rewritten as follows:

$$\Delta V = \frac{v_2}{2}\alpha^2 + v_1\alpha \tag{20}$$

where

$$v_1 = g^T p_g + c_{\mathcal{A}}^T p_{c_{\mathcal{A}}}, \tag{21a}$$
$$v_2 = p_g^T p_g + p_{c_{\mathcal{A}}}^T p_{c_{\mathcal{A}}}. \tag{21b}$$

Because v_1 is negative by (18c) and v_2 is intrinsically positive, ΔV is negative for a step length in the range of

$$0 < \alpha < -\frac{2v_1}{v_2}. \tag{22}$$

Therefore, the update law (12) makes the Lyapunov function V decrease to zero, consequently approaching the necessary conditions for optimality. Therefore, the update law (12) guarantees the convergence to the solution if the iteration starts sufficiently close to the solution. □

Lemma 1 states properties of the step length α.

Lemma 1. *The step length α computed by (13e) is positive and minimizes the perturbation in the Lyapunov function the most.*

Proof. Because v_1 is negative and v_2 is positive, α computed by $-v_1/v_2$ is positive and minimizes the quadratic function (20) the most. □

B. Algorithm

The optimization variables and Lagrange multipliers are updated using the computation result of (12) as follows:

$$x^{k+1} \leftarrow x^k + \Delta x^k, \tag{23a}$$
$$\lambda_{\mathcal{A}}^{k+1} \leftarrow \lambda_{\mathcal{A}}^k + \Delta\lambda_{\mathcal{A}}^k. \tag{23b}$$

Even though the Lyapunov function defined in (14) does not explicitly include three necessary conditions of (4c), (4d), and (4e) for the inequality constraints, these three conditions can be easily satisfied by adding the following actions at the end of each iteration:

- Limit the Lagrange multipliers for the inequality constraints greater than or equal to 0.
- Remove the inequality constraints with $\lambda_i = 0$ from the active set $\mathcal{A}(x)$.
- Add the inequality constraints with $c_i(x) < 0$ to the active set $\mathcal{A}(x)$.

The proposed method can be implemented by Algorithm 1.

Algorithm 1: Proposed method for NLP

Compute a feasible starting point (x^0, λ^0);
Set the initial active set $\mathcal{A}(x^0)$;
for $k = 0, 1, 2, \ldots$ **do**
  Compute Δx^k and $\Delta\lambda_{\mathcal{A}}^k$ using (12);
  $x^{k+1} \leftarrow x^k + \Delta x^k$;
  $\lambda_{\mathcal{A}}^{k+1} \leftarrow \lambda_{\mathcal{A}}^k + \Delta\lambda_{\mathcal{A}}^k$;
  $\lambda_i^{k+1} \leftarrow \max(\lambda_i^{k+1}, 0), i \in \mathcal{A}(x^k) \cap \mathcal{E}$;
  $\lambda_i^{k+1} \leftarrow \lambda_i^k, i \notin \mathcal{A}(x^k)$;
  $\mathcal{A}(x^{k+1}) \leftarrow \mathcal{A}(x^k) \backslash \{j\}$, for all $j \in \mathcal{A}(x^k)$ with $\lambda_j^{k+1} = 0$;
  $\mathcal{A}(x^{k+1}) \leftarrow \mathcal{A}(x^k) \cup \{j\}$, for all $j \notin \mathcal{A}(x^k)$ with $c_j(x^{k+1}) < 0$;

TABLE I
PROPERTY COMPARISON OF PROPOSED METHOD, SQP, AND ALM

	SQP	ALM	Proposed Method
Inversion of matrix	Yes (e.g., KKT)	Yes (e.g., Hessian)	No
Tuning parameters	Not necessary	Necessary (μ and others)	No
Allowance of constraint violation	No	Yes	Yes
Positive definiteness of (approximated) Hessian	Required	Required	Not necessary
Update of x and λ	Simultaneous	Seperate	Simultaneous

C. Property Comparison of Proposed Method, SQP, and ALM

Equations (12) and (13) show that the proposed method is matrix-inversion-free and tuning-parameter-free. There are two more to note. First, the proposed method does not require the positive definiteness of the Hessian matrix because matrix WW^T, which is at least positive semidefinite regardless of the positive definiteness of Hessian W, determines the negative definiteness of the perturbation in the Lyapunov function. Second, the update law for $\lambda_{\mathcal{A}}$, (12b), is similar to that of ALM, (11), in that both include information on constraint violations (i.e., c_i) but differs in that (12b) also includes information on $g(= \nabla_x L(x, \lambda))$. The inclusion of g allows the appropriate update of $\lambda_{\mathcal{A}}$ without using tuning parameters as in ALM. Actually, the term $C_{\mathcal{A}}g/2$ in (13b) yields the term in right-hand side of inequality (18b), thereby endowing the proposed method with a descent property.

The properties of the proposed method are summarized in Table II as compared with SQP and ALM. Note that this comparison only considers the case for local convergence.

IV. APPLICATION OF THE PROPOSED METHOD TO NMPC

This section presents an application of the proposed method to NMPC: Section IV-A explains two different formulations of a general NMPC problem, Section IV-B defines a target problem of NMPC in this study, and Section IV-C provides validation results of the proposed method for the target NMPC problem.

A. Problem formulation of NMPC

1) Typical formulation: A general NMPC formulation is

$$\min_{u_t,\cdots,u_{t+N-1}} \|e_{t+N}\|_F^2 + \sum_{n=0}^{N-1} \|e_{t+n}\|_Q^2 + q\left(x_{t+n}, u_{t+n}\right) \tag{24a}$$

subject to

$$x_{t+n+1} = f\left(x_{t+n}, u_{t+n}\right),\ y_{t+n} = g(x_{t+n}, u_{t+n}), \tag{24b}$$

$$h(x_{t+n}, u_{t+n}) \geq 0,\ n = 1, \ldots, N \tag{24c}$$

$$x_t = x(t), \tag{24d}$$

where subscripts $t, t+1, \cdots$ denote discrete-time instants; N is the prediction horizon; x, u, and y denote the state, control input, and output, respectively; $e := r - y$ represents the error to the reference signal r; F and Q are symmetric positive definite matrices; $\|z\|_S^2$ denotes $z^T S z$; q represents the control effort function whose typical choice is $\|u\|_R^2$ with a matrix R; and f, g, and h denote functions of the state, output, and inequality constraints, respectively. This problem corresponds to NLP, so the proposed method can be applied.

This problem formulation can describe optimal tracking control of various applications. However, this formulation requires an appropriate tuning of matrices F and Q, and function q to guarantee convergence of the error to zero because of the trade-off between the tracking error terms and control effort term in the objective function. In this regard, reformulation of this problem is presented below.

2) Reformulation: The problem (24) is reformulated by regarding the tracking error terms in the objective function as an equality constraint as follows:

$$\min_{u_t,\cdots,u_{t+N-1}} \sum_{n=0}^{N-1} q\left(x_{t+n}, u_{t+n}\right) \tag{25a}$$

subject to

$$x_{t+n+1} = f\left(x_{t+n}, u_{t+n}\right),\ y_{t+n} = g(x_{t+n}, u_{t+n}), \tag{25b}$$

$$h(x_{t+n}, u_{t+n}) \geq 0,\ n = 1, \ldots, N \tag{25c}$$

$$x_t = x(t), \tag{25d}$$

$$e_{t+N} = 0. \tag{25e}$$

There is no guarantee to satisfy the equality constraint (25e) if the reference signal r cannot be attained within the prediction horizon no matter what control effort, or if a numerical optimization method for this problem does not allow violation of the equality constraint. However, if a numerical optimization method allows the violation of constraints like the proposed method, this problem can be solved by such a method even if the reference signal cannot be attained. Even so, it is better if the reference signal changes smoothly so that the equality constraint can be satisfied at all time instants.

This problem separates the tracking error and control effort terms so that the trade-off between these two terms is resolved. Unlike SQP cannot handle this problem because it does not allow violation of constraints, ALM can handle this. However, using the proposed method is advantageous than using ALM according to the analysis presented in Section III-C.

TABLE II
PARAMETERS OF THE PMSM

Parameter	Value
R_s	25 mΩ
L_d	0.45 mH
L_q	0.66 mH
λ_{pm}	0.0563 Wb
P	8
V_{max}	56.5 V
T_s	0.1 ms

B. Target problem

A simple NMPC problem presented in [2] is considered the target problem to validate the proposed method. This problem defines the optimal torque control of permanent magnet synchronous machines (PMSMs) and can be expressed as both (24) and (25) with $N = 1, F = Q = w^{-1} I_2$, where $w > 0$ is the weighting factor, and

$$q(x_t, u_t) = x_{t+1}^T x_{t+1}, \tag{26a}$$

$$\begin{aligned} f\left(x_t, u_t\right) = &\begin{bmatrix} 1 - T_s R_s L_d^{-1} & T_s w_r L_q L_d^{-1} \\ -T_s w_r L_d L_q^{-1} & 1 - T_s R_s L_q^{-1} \end{bmatrix} x_t \\ &+ \begin{bmatrix} T_s L_d^{-1} & 0 \\ 0 & T_s L_q^{-1} \end{bmatrix} u_t + \begin{bmatrix} 0 \\ -T_s w_r \lambda_{pm} L_q^{-1} \end{bmatrix}, \end{aligned} \tag{26b}$$

$$\begin{aligned} g(x_t, u_t) = &\begin{bmatrix} 0 & 1.5 P \lambda_{pm} \end{bmatrix} x_t \\ &+ x_t^T \begin{bmatrix} 0 & 0.75 P (L_d - L_q) \\ 0.75 P (L_d - L_q) & 0 \end{bmatrix} x_t, \end{aligned} \tag{26c}$$

$$h(x_t, u_t) = V_{\max}^2 - u_t^T u_t, \tag{26d}$$

where x and u denote the stator current and voltage vectors, respectively; R_s denotes the stator resistance; L_d and L_q denote the d- and q-axis inductances, respectively; w_r is the electrical rotor speed; λ_{pm} denotes the magnetic flux linakge due to the permanent magnets; P is the number of pole pairs; $V_{\max}$ denotes the voltage limit; and y is the output torque. The parameters are selected as listed in Table II.

C. Validation

To validate the proposed method, the target problem presented in Section IV-B was solved by SQP, ALM, and the proposed method, respectively, in MATLAB 2022a. The reference signal r was given 30 Nm. Two different values for w_r, 840 rad/s and 1090 rad/s, were used to examine the cases when the inequality constraint ($h(x, u) \geq 0$) was inactive and active, respectively.

ALM and the proposed method solved the problem in the formulation of (25), while SQP solved the problem in the formulation of (24) because SQP did not guarantee to handle the equality constraint (25e). SQP was implemented by the 'fmincon' function available in MATLAB, which is a well-known NLP solver, with the option of 'sqp'. Three different values were used for w: 10^{-4}, 10^{-2}, and 10^0, to examine the trade-off between the tracking error terms and control effort term in the objective function. ALM was implemented by solving (10) using Newton's method with an iteration termination condition of $|\nabla_x \mathcal{L}_A| \leq 10^{-4}$. Three different values were used for μ: 10^{-2}, 10^0, and 10^2, to examine the sensitivity of solutions to this tuning parameter. The proposed method was implemented by Algorithm 1. Two different cases were examined when the for-loop in Algorithm 1 was repeated once and twice, respectively, for each time instant t.

The NMPC results obtained using SQP with $w = 10^{-4}, 10^{-2}$, and 10^0 are shown in Figs. 1, 2, and 3, respectively. x_1 and x_2 on the horizontal and vertical axes represent the first and second elements of the state x, respectively. The MTPA line, denoted by dashed lines in the figures, represents the points satisfying

$$\min_x q(x, u) \tag{27a}$$

$$\text{subject to}$$

$$e = r - y = 0, \tag{27b}$$

meaning the target points of the state x when the inequality constraint $h(x, u) \geq 0$ is inactive. With $w = 10^{-4}$, the state trajectory, x_t, satisfied the tracking condition $e_{t+N} = 0$ well but quite deviated from the control-effort-minimizing points (i.e., MTPA line) due to the large portion of the tracking error terms (see Fig. 1). With $w = 10^0$, the state trajectory aligned with the MTPA line when the inequality constraint was inactive (see Fig. 3(a)) or moved close to the MTPA line even when the constraint was active (see Fig. 3(b)). However, the state trajectory hardly satisfied the tracking condition due to the large portion of the control effort term. The balance between the tracking error terms and control effort term was made with $w = 10^{-2}$ (see Fig. 2).

The NMPC results obtained using ALM with $\mu = 10^{-2}, 10^0$, and 10^2 are shown in Figs. 4, 5, and 6, respectively. With $\mu = 10^{-2}$, the state trajectory, x_t, was unstable until reaching the final point when the inequality constraint was inactive (see Fig. 4(a)). This was because the barrier parameter μ was too small so the Lagrange multiplier for the inequality constraint (25c) was easily updated and considered active, which had to be inactive. With $w = 10^2$, by contrast, the state trajectory hardly satisfied the tracking condition (see Fig. 6) because the Lagrange multiplier for the equality constraint (25e) was updated slowly due to the large value of μ. The Lagrange multipliers for the constraints were updated appropriately with $\mu = 10^0$, and thus a desirable state trajectory was obtained (see Fig. 5).

The NMPC results obtained using the proposed method of repeating the for-loop once and twice are shown in Figs. 7 and 8, respectively. Figure 7 shows that the proposed method provided an almost direct move to the target points even with one repetition of the for-loop. Just one more repetition of the for-loop guaranteed a very accurate move, as shown by Fig. 8. This result is significant in that conventional methods like SQP and ALM provided satisfactory results only with appropriate values of tuning parameters, whereas the proposed method guaranteed a desirable result just by repeating the simple algorithm once or twice, not relying on tuning parameters.

Computation times of representative results of each method are shown in Fig. 9. The representative results were selected as Figs. 2(b), 5(b), and 7(b) for SQP, ALM, and the proposed method, respectively. The computation times of SQP were approximately one hundred times greater than those of ALM and the proposed method, which is probably due to the inversion of the KKT matrix for SQP and multiple iterations to elaborate solutions. The average computation time of ALM was approximately twice greater than that of the proposed method. This is because ALM used multiple iterations to guarantee the convergence of Newton's method. This result verifies that the proposed method, which is matrix-inversion-free and does not require large numbers of iterations, is computationally efficient.

V. Conclusion

This study presented a novel numerical optimization method to solve NLP. This method was designed based on a Lyapunov approach to reach the necessary conditions for optimality. The advantage of using the Lyapunov approach was the update law was derived in a tuning-parameter-free and matrix-inversion-free manner; thus, the proposed method could be implemented easily with less iteration and computation time than conventional methods, such as SQP and ALM. The effectiveness of the proposed method was validated by using it to solve an NLP problem, which was an NMPC problem for optimal torque control of PMSMs, and comparing it with SQP and ALM.

The following issues need to be handled in a future study:

- **Validation for large-scale problems:** This study used a small-scale NLP problem, whose number of optimization variables was two. The effectiveness of the proposed method is expected to be further highlighted for large-scale problems. A possible target problem would be an NMPC problem with a longer prediction horizon like the one presented in [3].
- **Proof of the convergence rate:** No discussion was made on the convergence rate of the proposed method. The convergence rate of the proposed method needs to be investigated in that other methods such as SQP guarantee quadratic convergence near local solutions.
- **Considerations on global convergence:** Ensuring global convergence is a very well-known but challenging issue in numerical optimization. Nonetheless, it is expected to guarantee global and fast convergence for an NMPC problem if a well-approximated problem is solved and the approximated solution is provided as the initial guess for the original problem. For instance, define the optimization variable as $u_t = \hat{u}_t + \Delta u_t$, where $\hat{u}_t$ is the solution of an approximated problem and Δu_t is the deviation between the true and approximated solutions. The approximated solution $\hat{u}_t$ could be obtained by solving the following problem:

$$\min_{u_t} \hat{q}\left(x_t, u_t\right) \tag{28a}$$

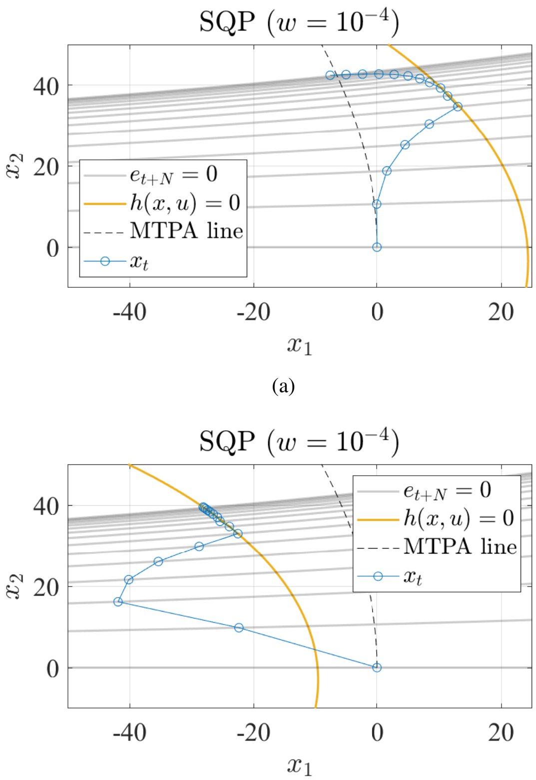
$$\text{subject to}$$

$$x_{t+1} = \hat{f}\left(x_t, u_t\right), \;\; y_t = \hat{g}(x_t, u_t), \tag{28b}$$

$$x_t = x(t), \tag{28c}$$

$$e_{t+1} = 0, \tag{28d}$$

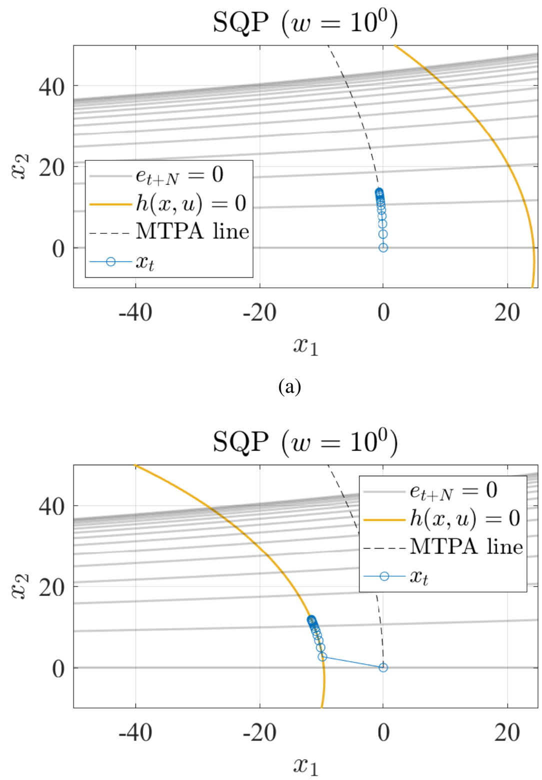
where the prediction horizon N is set to 1, $\hat{q}$ is a quadratic approximation of function q, and, $\hat{f}$ and $\hat{g}$ are linear approximations of function f and g, respectively. This is a QP with equality constraints, whose solution is easily obtained by solving a KKT matrix. Then, the original problem (25) is rewritten with the new optimization variable Δu_t and solved by the proposed method.



(a)

(b)

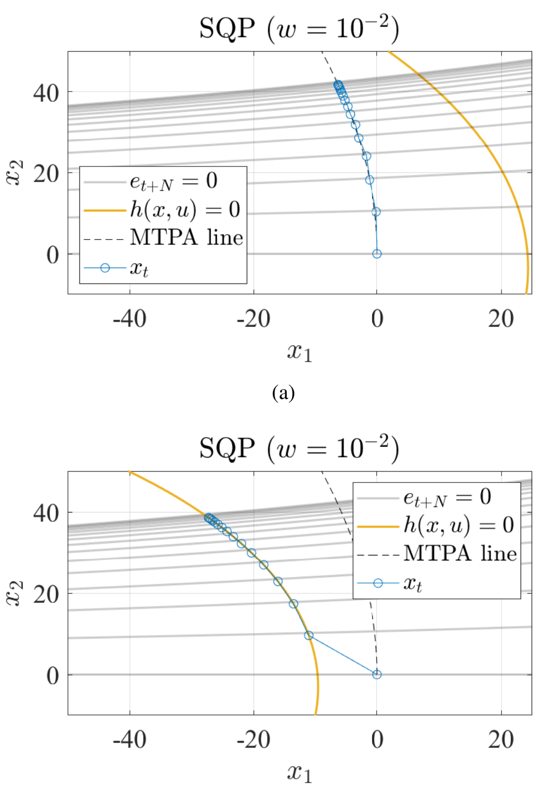
Fig. 1. NMPC results obtained using SQP with $w = 10^{-4}$ for the conditions of (a) $w_r = 840$ rad/s and (b) $w_r = 1090$ rad/s.



(a)

(b)

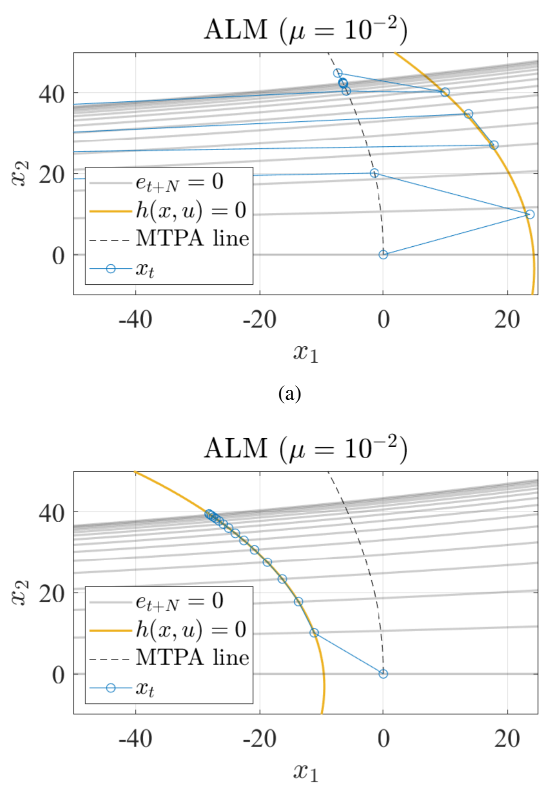
Fig. 3. NMPC results obtained using SQP with $w = 10^{0}$ for the conditions of (a) $w_r = 840$ rad/s and (b) $w_r = 1090$ rad/s.



(a)

(b)

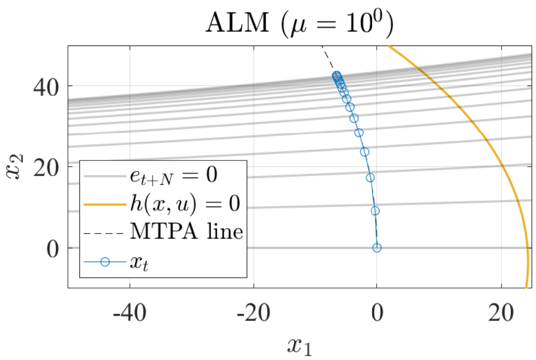
Fig. 2. NMPC results obtained using SQP with $w = 10^{-2}$ for the conditions of (a) $w_r = 840$ rad/s and (b) $w_r = 1090$ rad/s.



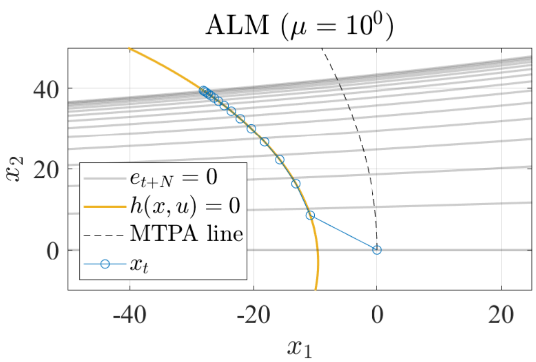
(a)

(b)

Fig. 4. NMPC results obtained using ALM with $\mu = 10^{-2}$ for the conditions of (a) $w_r = 840$ rad/s and (b) $w_r = 1090$ rad/s.

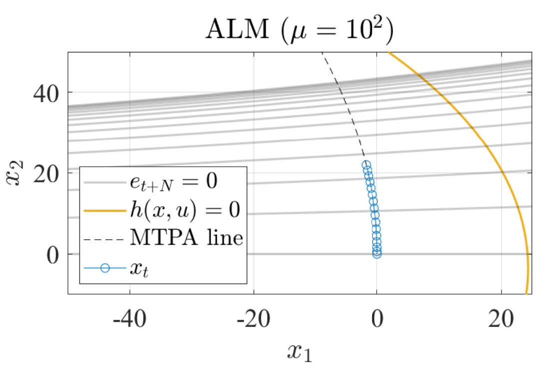


(a)

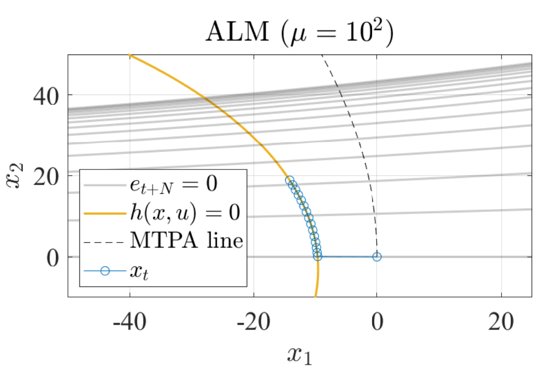


(b)

Fig. 5. NMPC results obtained using ALM with $\mu = 10^0$ for the conditions of (a) $w_r = 840$ rad/s and (b) $w_r = 1090$ rad/s.

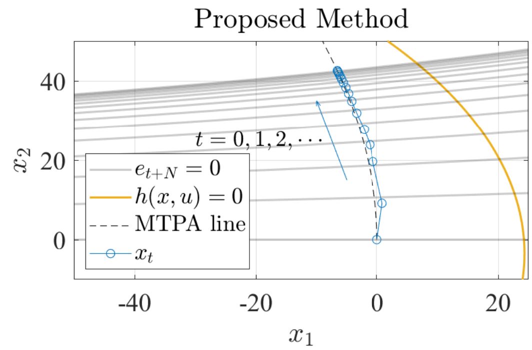


(a)

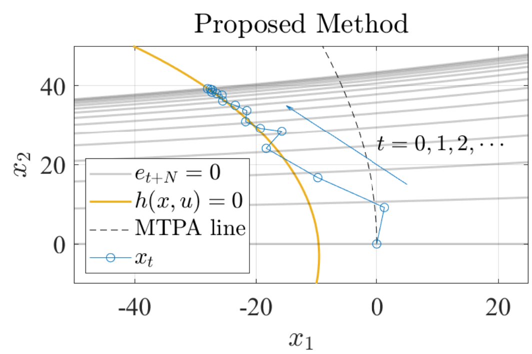


(b)

Fig. 6. NMPC results obtained using ALM with $\mu = 10^2$ for the conditions of (a) $w_r = 840$ rad/s and (b) $w_r = 1090$ rad/s.

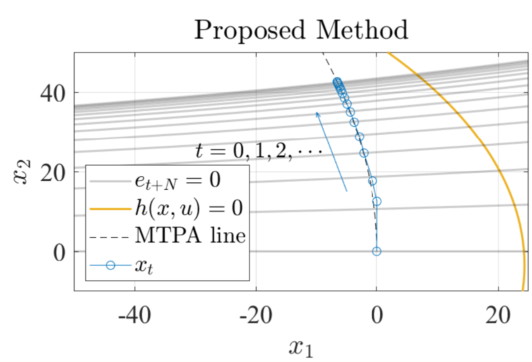


(a)

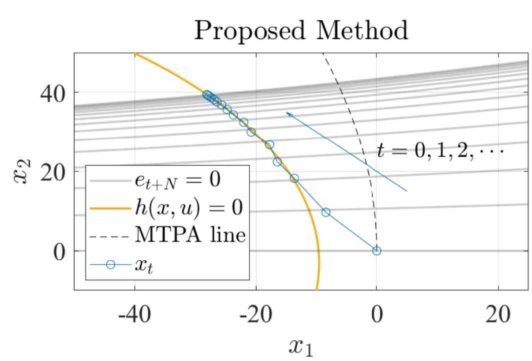


(b)

Fig. 7. NMPC results obtained using the proposed method of repeating the for-loop once for each time instant t, for the conditions of (a) $w_r = 840$ rad/s and (b) $w_r = 1090$ rad/s.

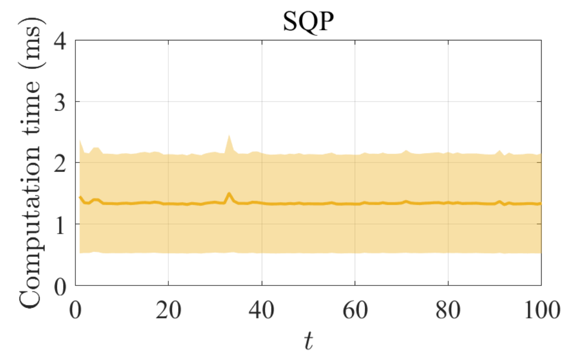


(a)

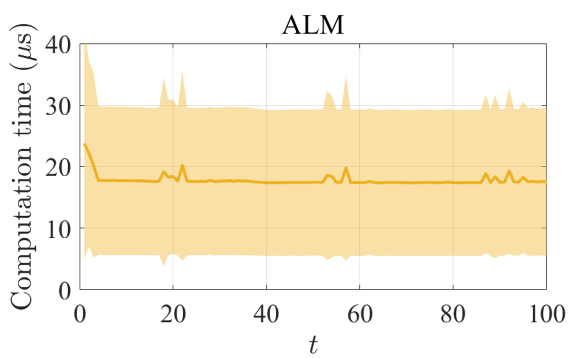


(b)

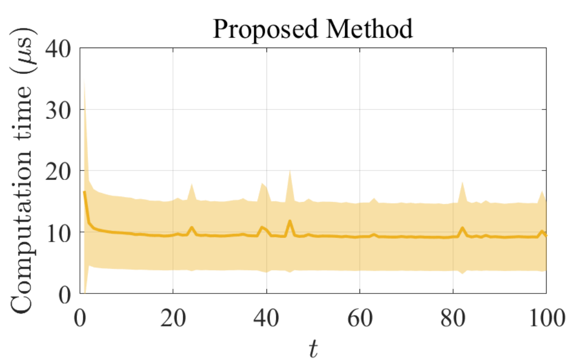
Fig. 8. NMPC results obtained using the proposed method of repeating the for-loop twice for each time instant t, for the conditions of (a) $w_r = 840$ rad/s and (b) $w_r = 1090$ rad/s.



(a)



(b)



(c)

Fig. 9. Computation time at each time instant t of (a) the SQP's result presented in Fig. 2(b), (b) the ALM's result presented in Fig. 5(b), and (c) the proposed method's result presented in Fig. 7(b). The solid lines and shaded areas represent the mean values and standard deviations, respectively.

References

[1] J. E. Hurtado and R. D. Robinett III, "Convergence of newton's method via lyapunov analysis," *Journal of guidance, control, and dynamics*, vol. 28, no. 2, pp. 363–364, 2005.

[2] K. Choi and K.-B. Park, "Model predictive torque control of synchronous machines without a current or stator flux reference generator," in *2023 IEEE 32nd International Symposium on Industrial Electronics (ISIE)*, pp. 1–6. IEEE, 2023.

[3] I. De Martin, F. Tinazzi, M. Zigliotto, and C. Hackl, "System parameter-free continuous control-set predictive current control of synchronous motors," in *IECON 2022–48th Annual Conference of the IEEE Industrial Electronics Society*, pp. 1–6. IEEE, 2022.