# Support Vector Machine Informed Explicit Nonlinear Model Predictive Control Using Low-Discrepancy Sequences

Ankush Chakrabarty[1], Vu Dinh[2], Martin J. Corless[3], Ann E. Rundell[4], Stanisław H. Żak[1], Gregery T. Buzzard[2]

***Abstract*—In this paper, an explicit nonlinear model predictive controller (ENMPC) for the stabilization of nonlinear systems is investigated. The proposed ENMPC is constructed using tensored polynomial basis functions and samples drawn from low-discrepancy sequences. Solutions of a finite-horizon optimal control problem at the sampled nodes are used (i) to learn an inner and outer approximation of the feasible region of the ENMPC using support vector machines, and (ii) to construct the ENMPC control surface on the computed feasible region using regression or sparse-grid interpolation, depending on the shape of the feasible region. The attractiveness of the proposed control scheme lies in its tractability to higher-dimensional systems with feasibility and stability guarantees, significantly small on-line computation times, and ease of implementation.**

***Index Terms*—Model predictive control, nonlinear systems, supervised learning, function approximation, low-discrepancy sampling.**

## I. Introduction

Nonlinear Model Predictive Control (NMPC) has been widely applied in numerous industrial applications due to its ability to handle constraints and its inherent robustness properties [1]. There are, however, certain drawbacks associated with the NMPC. These include: (i) complexity of implementation on low-memory devices [2], (ii) increased computational burden due to iterative computation of on-line optimal control actions, (especially in the design of nonlinear controllers), (iii) reduced computational efficiency when applied to higher-dimensional models, and, (iv) difficulty in guaranteeing closed-loop properties of the control scheme [3], [4]. To address the above issues, the Explicit Model Predictive Control (EMPC) method was proposed. The advantage of the EMPC is that it replaces the finite horizon optimal control problem at each iteration with pre-computed *explicit* control laws based on current states of the system, thereby increasing the computational efficacy [4].

For linear systems, EMPC laws are computed off-line using methods such as those reported in [3], [5]–[14]. An extension to nonlinear EMPC, however, is not straightforward except for certain special classes of nonlinear systems. For example, it is shown in [15] that optimal control laws and stability guarantees can be analytically derived for unconstrained input-affine systems. Recent advances have also been made in convex multi-parametric nonlinear programming for nonlinear EMPC control laws where stability guarantees are provided on the sub-optimal controller by adjusting approximation error tolerances [16]–[18]. Other reports of satisfactory nonlinear EMPC performance with approximated state-feedback NMPC control law include neural network formulations of the EMPC for locally Lipschitz systems [19], [20]. Although these controllers perform well on benchmark problems, the above algorithms require the partition of admissible regions into boxes or hypercubes. Hence, current EMPC formulations for nonlinear systems incur large computational costs for systems with higher dimensionality. Thus, EMPC is commonly used to control smaller ($n < 5$) systems [21]. To overcome the storage limitations associated with hypercubes, we propose a scalable, sampling-based nonlinear EMPC on domains of arbitrary shape.

Application of learning-based methods in NMPC can be found in the recent literature. For example, neural networks are employed to approximate the predictive control surface [22]–[26] and the feasible region is maximized using support vector machines (SVMs) in [27]. In this paper, we also choose the SVM classifier to approximate the feasible region for two main reasons. First, the SVM is a supervised learning algorithm that works efficiently in the presence of sparsely distributed data [28], [29]. Second, the SVM is relatively fast and works well in higher dimensional spaces owing to the 'kernel-trick', when compared to existing artificial neural networks [30]. A major difference of our work from the method proposed in [27] is that we propose a deterministic learning method for estimating the feasible region. That is, the training samples are derived from a low-discrepancy sequence, as investigated in [31]; not randomly extracted from a probability distribution. The rationale behind the selection of such sampling patterns is (i) to ensure that the samples are distibuted uniformly over the admissible space and, (ii) to reduce the number of samples required for solving the classification problem in the SVM framework.

In this paper, we propose an easily implementable sampling-based Explicit Nonlinear Model Predictive Controller (ENMPC) for nonlinear systems with guaranteed feasibility and stability. First, we sample data points on the state-space region of interest using using deterministic sampling. At each sam-

[1] A. Chakrabarty (chakraa@purdue.edu) and S. H. Żak (zak@purdue.edu) are affiliated with the School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN, USA.

[2] G.T. Buzzard (buzzard@purdue.edu) and V. Dinh (vdinh@purdue.edu) are affiliated with the Department of Mathematics, Purdue University, West Lafayette, IN.

[3] M.J. Corless (corless@purdue.edu) is affiliated with the School of Aeronautics and Astronautics, Purdue University, West Lafayette, IN.

[4] A.E. Rundell (rundell@purdue.edu) is affiliated with the Weldon School of Biomedical Engineering, Purdue University, West Lafayette, IN.

pled point, we solve a terminal constraint-based finite horizon optimal control problem and store feasibility information and optimal control actions. Next, inner and outer approximations of feasible region boundaries are computed using SVM bi-classifiers. Although results for SVM bi-classifiers exist for random sampling schemes [32], we develop new convergence results for kernel-based SVM bi-classifiers informed by deterministic low-discrepancy sampling sequences. Finally, the ENMPC control surface is constructed within the feasible region using the stored optimal control actions at each sample point. Feasibility and stability guarantees are provided for the ENMPC. We demonstrate the performance of the proposed controller on a 2-dimensional and 8-dimensional simulated example. This paper extends some results obtained in [33].

The rest of the paper is organized as follows. In Section II we present our notation and in Section III, we present a class of nonlinear models for which we develop our proposed ENMPC and discuss briefly the finite horizon optimal control problem. In Section IV, we briefly review the support vector bi-classifier scheme and explain how it is utilized to estimate an inner and outer approximation of the feasible region. The guaranteed feasibility and stability of the proposed ENMPC is discussed in V. We illustrate the effectiveness of the proposed control scheme on two nonlinear systems in Section VI. Our method is first tested on a benchmark two-dimensional nonlinear system to compare to existing methods. Second, the proposed ENMPC is used to control an 8-dimensional nonlinear model to illustrate the computational efficacy of the method on higher dimensions without a significant degradation in performance. We offer conclusions in Section VII. The Appendix contains the pseudo-code for implementation of the proposed ENMPC.

## II. Notation

We denote $\mathbb{N}$ for the set of natural numbers and $\mathbb{R}^n$ for the $n$-tuples with real components. We denote $\mathcal{C}^1(\mathbb{R}^n)$ for the space of continuously differentiable functions on $\mathbb{R}^n$, $\mathbb{L}^1(\mathbb{X})$ for the space of Lebesgue integrable functions on the set $\mathbb{X}$ and $\mathbb{L}^2(\mathbb{X})$ for the space of square integrable functions on $\mathbb{X}$ equipped with inner product $\langle\cdot,\cdot\rangle$. Also, $\ell^2$ denotes the square summable sequence space. We use boldface to distinguish vectors/matrices from scalars, that is, $\boldsymbol{x}\in\mathbb{R}^n$ for some $n\in\mathbb{N}$ indicates that $\boldsymbol{x}$ is a vector while $x$ is a scalar. We denote a sequence of scalars with the notation $\{x_i\}$ for $i\in\mathbb{N}$ and a finite set of $N$ samples is denoted $\{x_i\}_{i=1}^N$ where each $x_i\in\mathbb{R}$. The cardinality of a set $K$ is denoted by $\mathrm{card}(K)$ and the Lebesgue measure of the set is denoted $\mathrm{Vol}(K)$. For a closed set $K$, we denote the boundary as $\partial K$, its complement as $K^c$ and the open set consisting of interior points of $K$ as $\mathrm{int}\, K$. The notation $\mathcal{O}(\cdot)$ is the standard 'big-O' notation used in complexity analysis of algorithms. For two sets $A, B$ in a metric space $(X,d)$, we denote the distance $d(A,B)=\inf_{x\in A, y\in B} d(x,y)$ and the diameter of the set $A$ is denoted $\rho(A)=\sup\{d(x,y): x,y\in A\}$. The open ball of radius $\varepsilon$ centered at $\boldsymbol{x}\in X$ is denoted by $\mathbb{B}_\varepsilon(\boldsymbol{x})$. For a square, symmetric matrix $\boldsymbol{P}=\boldsymbol{P}^\top$, we denote the quadratic form $\|\boldsymbol{x}\|_{\boldsymbol{P}}^2=\boldsymbol{x}^\top\boldsymbol{P}\boldsymbol{x}$.

## III. Problem Statement

### A. Model Description

We consider a class of nonlinear dynamical systems modeled as,

$$\dot{\boldsymbol{x}}=\boldsymbol{f}(\boldsymbol{x},\boldsymbol{u}), \tag{1}$$

where $\boldsymbol{x}\in\mathbb{X}\subset\mathbb{R}^{n_x}$ is a state-vector constrained to the state-space polytope $\mathbb{X}$, $\boldsymbol{u}\in\mathbb{U}\subset\mathbb{R}^{n_u}$ is the control-vector constrained to the input space polytope $\mathbb{U}$, and $\boldsymbol{f}:\mathbb{X}\times\mathbb{U}\to\mathbb{R}^{n_x}$ is the nonlinear model.

We make the following assumptions.

**Assumption 1.** *The set $\mathbb{U}$ is convex, compact and contains the origin in its interior. The set $\mathbb{X}$ is a product of closed intervals and has nonempty interior.*

**Remark 1.** Without loss of generality, we assume that $\mathbb{X}$ is a $n_x$-dimensional hypercube.

**Assumption 2.** *The function $\boldsymbol{f}$ is continuously differentiable, that is, $\boldsymbol{f}\in\mathcal{C}^1$ with a Lipschitz constant $\mathfrak{L}_f^x$ with respect to $\boldsymbol{x}$ and a Lipschitz constant $\mathfrak{L}_f^u$ with respect to $\boldsymbol{u}$. This implies*

$$\|\boldsymbol{f}(\boldsymbol{x}_1,\boldsymbol{u})-\boldsymbol{f}(\boldsymbol{x}_2,\boldsymbol{u})\|\le\mathfrak{L}_f^x\|\boldsymbol{x}_1-\boldsymbol{x}_2\|$$

*and*

$$\|\boldsymbol{f}(\boldsymbol{x},\boldsymbol{u}_1)-\boldsymbol{f}(\boldsymbol{x},\boldsymbol{u}_2)\|\le\mathfrak{L}_f^u\|\boldsymbol{u}_1-\boldsymbol{u}_2\|$$

*for any $\boldsymbol{x}_1,\boldsymbol{x}_2\in\mathbb{X}$ and any $\boldsymbol{u}_1,\boldsymbol{u}_2\in\mathbb{U}$.*

**Assumption 3.** *The pair $(\boldsymbol{x}_e,\boldsymbol{u}_e)$ is an equilibrium pair of the nominal system, that is, $\boldsymbol{f}(\boldsymbol{x}_e,\boldsymbol{u}_e)=\boldsymbol{0}$ and the linearized model at the equilibrium is stabilizable.*

**Remark 2.** Without loss of generality, we assume that $(\boldsymbol{x}_e,\boldsymbol{u}_e)=(\boldsymbol{0},\boldsymbol{0})$.

**Assumption 4.** *The nonlinear model (1) has unique solutions for any initial conditions and for any admissible piece-wise continuous controllers $u(t)\in\mathbb{U}$ for all $t$.*

We now propose a construction methodology for the ENMPC controller using deterministic sampling.

### B. Deterministic Sampling for ENMPC Construction

Our control objective is to construct a fast, scalable stabilizing ENMPC controller $\boldsymbol{u}(\boldsymbol{x})$ for nonlinear systems of the form (1) under state and input constraints. To this end we sample the space $\mathbb{X}$ and solve a terminal region based finite horizon optimal control problem [34] at each sampled node. The nodes/samples are extracted from a low-discrepancy sequence constructed on a multilevel sparse-grid. We use the standard notion of low-discrepancy sequences.

**Definition 1 (Low-Discrepancy Sequence).** *The **discrepancy** of a sequence $\{\boldsymbol{x}_j\}_{j=1}^N\subset\mathbb{X}$ is defined as*

$$D_N(\{\boldsymbol{x}_j\}_{j=1}^N)\triangleq\sup_{X\in J}\left|\frac{\#X_N}{N}-\frac{\mathrm{Vol}(X)}{\mathrm{Vol}(\mathbb{X})}\right|, \tag{2}$$

*where $J$ is the set of $n_x$-dimensional intervals of the form*

$$\prod_{j=1}^{n_x}[a_j,b_j)=\{\boldsymbol{x}\in\mathbb{X}:a_j\le x_j<b_j\}$$

*where* $\prod_{j=1}^{n_x}[a_j, b_j) \subset \mathbb{X}$ *and*

$$\#X_N \triangleq \mathrm{card}\{j \in \{1, \ldots, N\} : \boldsymbol{x}_j \in X\}.$$

*For a* ***low-discrepancy*** *sequence,*

$$\lim_{N\to\infty} D_N(\{\boldsymbol{x}_j\}_{j=1}^N) = 0.$$

We now discuss the construction of the ENMPC using $N$ samples. Suppose the $j$th sampled node is denoted as $\boldsymbol{x}_j$. We make an assumption which ensures that the samples $\boldsymbol{x}_j$ are distributed sufficiently evenly over $\mathbb{X}$.

**Assumption 5.** *The sequence* $\{\boldsymbol{x}_j\}_{j=1}^\infty$ *is a low-discrepancy sequence on* $\mathbb{X}$*, in the sense of Definition 1.*

**Remark 3.** Assumption 5 includes several common sampling schemes, including: (i) when the $\{\boldsymbol{x}_j\}_{j=1}^N$ consists of grid nodes in a multi-level sparse grid in state space (using equispaced points to generate the grid) and (ii) when $\{\boldsymbol{x}_j\}_{j=1}^N$ is quasi-random, such as the nodes in the Sobol or Halton sequences. Implementation of low-discrepancy sequences are widely available online for `MATLAB` and `C/C++`.

Upon fixing a sampling sequence on $\mathbb{X}$, we solve the following constrained finite horizon optimal control problem at each sample in the sequence $\{\boldsymbol{x}_j\}_{j=1}^N$:

$$\min_{\boldsymbol{u}} \left( \|\boldsymbol{x}(T_f)\|_{\boldsymbol{P}}^2 + \int_0^{T_f} \|\boldsymbol{x}(t)\|_{\boldsymbol{Q}}^2 + \|\boldsymbol{u}(t)\|_{\boldsymbol{R}}^2 \,\mathrm{d}t \right) \tag{3a}$$

subject to:

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}), \tag{3b}$$
$$\boldsymbol{x}(0) = \boldsymbol{x}_j, \tag{3c}$$
$$\boldsymbol{x}(T_f) \in \mathbb{X}_T, \tag{3d}$$
$$\boldsymbol{x}(t) \in \mathbb{X}, \tag{3e}$$
$$\boldsymbol{u}(t) \in \mathbb{U} \; \forall \, t \in [0, T_f]. \tag{3f}$$

Let the corresponding minimizer be denoted $\boldsymbol{u}^*(\boldsymbol{x}_j)$. Here $\boldsymbol{P} = \boldsymbol{P}^\top \succ 0$ is the terminal penalty matrix, $\boldsymbol{Q} = \boldsymbol{Q}^\top \succeq 0$, $\boldsymbol{R} = \boldsymbol{R}^\top \succ 0$ are weighting matrices for the stage cost, $T_f$ is the prediction horizon, and $\mathbb{X}_T$ is an open terminal set within which a feasible (albeit, perhaps suboptimal) stabilizing state-feedback controller exists, see for example, [34].

An important property of this terminal set is that it is control invariant with respect to a pre-computed auxiliary state-feedback controller $\boldsymbol{u}_T(\boldsymbol{x})$. That is, if $\boldsymbol{x}(t_0) \in \mathbb{X}_T$, then the solutions of the dynamical system (1) with the auxiliary controller $\boldsymbol{u}_T$ satisfy $\boldsymbol{x}(t) \in \mathbb{X}_T$ and $\boldsymbol{u}_T(\boldsymbol{x}(t)) \in \mathbb{U}$ for all $t \geq t_0$. We now define our notion of feasibility.

**Definition 2 (Feasible State).** *A state* $\boldsymbol{x}_0 = \boldsymbol{x}(t_0) \in \mathbb{X}$ *is feasible if there exists an admissible control history, that is,* $\boldsymbol{u}(t) \in \mathbb{U}$ *for all* $t \geq t_0$*, that drives the system (1) from* $\boldsymbol{x}(t_0)$ *to* $\boldsymbol{x}(T_f) \in \mathbb{X}_T$ *while ensuring state constraints* $x(t) \in \mathbb{X}$ *for all* $t \in [t_0, T_f]$.

That is, the $j$th sample $\boldsymbol{x}_j$ is feasible if a feasible solution to (3) exists with initial condition $\boldsymbol{x}(0) = \boldsymbol{x}_j$.

**Definition 3 (Feasible Region).** *The set of all feasible states within the state space polytope* $\mathbb{X}$ *is called the feasible region, denoted* $\mathbb{F}$.

By Definition 3, $\mathbb{F} \subseteq \mathbb{X}$. We denote $\partial\mathbb{F}$ as the boundary of $\mathbb{F}$ and make the following assumption.

**Assumption 6.** *There exists a function* $\zeta \in \mathcal{C}^1(\mathbb{X})$ *such that the feasible domain restricted to within* $\mathbb{X}$ *can be represented as the zero superlevel-set of* $\zeta$*, that is,*

$$\mathrm{int}\ \mathbb{F} \cap \mathrm{int}\ \mathbb{X} = \{\boldsymbol{x} \in \mathrm{int}\ \mathbb{X} : \zeta(\boldsymbol{x}) > 0\}.$$

We now present a support vector machine classification (SVM) method that can be employed to estimate the feasible region boundary function $\zeta$ using the low-discrepancy samples $\{\boldsymbol{x}_j\}_{j=1}^N$.

## IV. Feasible Region Boundary Estimation Using Support Vector Machines

In this section, we first present the principle of SVM classification and then apply this method to estimate the boundary function $\zeta$ described in Assumption 6. Detailed discussions of kernel based classification methods (specifically, SVM) can be found in [29], [35].

The SVM bi-classifier is a supervised learning scheme which efficiently solves a two-class pattern recognition problem. Suppose that the vector $\boldsymbol{x}_j \in \mathbb{X}$ is the $j$th sample to be classified and its label is denoted by $y_j$, where $y_j \in \{-1, +1\}$ and $j = 1, 2, ..., N$. The goal of the SVM is to construct a decision function $\psi(\boldsymbol{x}) : \mathbb{X} \to \mathbb{R}$ that can accurately classify an arbitrary state $\boldsymbol{x} \in \mathbb{X}$ as feasible (labeled '+1') or infeasible (labeled '-1'). This is equivalent to reconstructing the feasible region boundary function $\zeta$.

### A. Linearly Separable Data

For linearly separable data, the separating hyperplane function is of the form $\psi(\boldsymbol{x}) = \boldsymbol{\omega}^\top \boldsymbol{x} + b$, where $\boldsymbol{\omega}, b$ are parameters that determine the orientation of the separating hyperplane. If a state $\boldsymbol{x}$ belongs to the class '+1', then $\psi(\boldsymbol{x}) \geq +1$ and if it belongs to '−1' then $\psi(\boldsymbol{x}) \leq -1$. In [35], the classification problem is formulated as the following constrained optimization problem,

$$\min \; \frac{1}{2}\boldsymbol{\omega}^\top \boldsymbol{\omega}$$

subject to

$$y_i(\boldsymbol{\omega}^\top \boldsymbol{x}_j + b) \geq 1, \;\; \forall \, j = 1, 2, \cdots, N.$$

In order to avoid over-fitting of the data, we utilize a regularization term into the cost function. The modified optimization problem then takes the form

$$\min \; \left( \frac{1}{2}\boldsymbol{\omega}^\top \boldsymbol{\omega} + L \sum_{j=1}^N s_j \right) \tag{4a}$$

subject to

$$y_i(\boldsymbol{\omega}^\top \boldsymbol{x}_j + b) \geq 1 - s_j, \tag{4b}$$
$$s_j \geq 0 \tag{4c}$$

for all $i, j = 1, \ldots, N$. where $L > 0$ is a regularization parameter, and the $s_i$'s are slack variables introduced to relax separability constraints. The minimizer of (4) is denoted $\boldsymbol{\omega}^*$

and the **separating margin** is $\mu^* = 1/\|\boldsymbol{\omega}^*\|_2$. This margin $\mu^* > 0$ is sometimes called the '1-norm soft margin' and denotes the separation between the two classes of data being classified. The SVM cannot classify the data with guaranteed accuracy within this margin.

### *B. Nonlinearly Separable Data*

If the data is not linearly separable in the feature space, a common approach is to map the data to a higher-dimensional space where the data is linearly separable. This is sometimes called the 'nonlinear SVM', see for example [35].

Let $\Gamma : \mathbb{X} \to \mathbb{H}$ be a map from the feature space $\mathbb{X}$ to the higher-dimensional Hilbert space $\mathbb{H}$. The coefficients of the separating hyperplane in $\mathbb{H}$ are obtained by solving the following primal problem

$$\text{minimize} \quad \left(\frac{1}{2}\|\boldsymbol{\omega}\|^2_{\mathbb{H}} + L\sum_{j=1}^{N} s_j\right) \tag{5a}$$

$$\text{subject to}$$

$$y_i(\langle \boldsymbol{\omega}, \Gamma(\boldsymbol{x}_j)\rangle_{\mathbb{H}} + b) \geq 1 - s_j, \tag{5b}$$

$$s_j \geq 0. \tag{5c}$$

This can be reformulated as the following dual problem:

$$\beta^*_N = \arg\max_{\boldsymbol{\beta}} \sum_{k=1}^{N}\beta_k - \frac{1}{2}\sum_{k=1}^{N}\sum_{j=1}^{N}\beta_k\beta_j y_k y_j \mathcal{K}(\boldsymbol{x}_k, \boldsymbol{x}_j) \tag{6a}$$

$$\text{subject to:}$$

$$\sum_{k=1}^{N}\beta_k y_k = 0, \tag{6b}$$

$$0 \leq \beta_k \leq L, \ \ \forall\, k = 1, 2, \cdots, N, \tag{6c}$$

where $\mathcal{K}(\cdot,\cdot)$ is the so-called SVM **kernel function** and $L > 0$ is the regularization parameter for the primal problem. Then the SVM decision function is

$$\boxed{\psi(\boldsymbol{x}, \boldsymbol{\beta}^*_N) = \sum_{k=1}^{N}\beta^*_k y_k \mathcal{K}(\boldsymbol{x}_k, \boldsymbol{x}),} \tag{7}$$

and the estimated **feasibility region boundary** or estimated **separating manifold** is given by

$$\boxed{\mathcal{S} \triangleq \{\boldsymbol{x} \in \mathbb{R}^{n_x} : \psi(\boldsymbol{x}, \boldsymbol{\beta}^*_N) = 0\}.} \tag{8}$$

**Remark 4.** The kernel function avoids expensive computation of the inner product $\langle\Gamma(\boldsymbol{x}_k), \Gamma(\boldsymbol{x}_j)\rangle_{\mathbb{H}}$, as discussed in [29], [35]. If the kernel satisfies Mercer's condition, then $\mathcal{K}(\boldsymbol{x}_k, \boldsymbol{x}) \equiv \langle\Gamma(\boldsymbol{x}_k), \Gamma(\boldsymbol{x})\rangle_{\mathbb{H}}$. This leads to an efficient solution of (6).

### *C. Inner and Outer Approximations of $\partial\mathbb{F}$ using SVM*

In this subsection, we are concerned with constructing strict inner ($\mathbb{F}^-$) and outer ($\mathbb{F}^+$) approximations of the feasibility region. This idea has been discussed previously in [36]. Bi-classifier based inner and outer approximations are proposed in [37]; however, no guarantees are provided for error convergence. Herein, we propose a novel algorithm for constructing strict inner and outer approximations of the feasible region. Furthermore, we provide convergence guarantees for the approximation error.

Our algorithm for constructing $\mathbb{F}^-$ and $\mathbb{F}^+$ is as follows. We begin by solving (6) to obtain $\boldsymbol{\beta}^*_N$. Next, we collect the samples which are labeled feasible ('+1'), and infeasible ('-1'), respectively. To this end, we define the sets

$$\mathcal{X}^+_N = \{\boldsymbol{x}_k \in \{\boldsymbol{x}_j\}_{j=1}^N : \boldsymbol{x}_k \text{ is labeled 'feasible'}\},$$

and

$$\mathcal{X}^-_N = \{\boldsymbol{x}_k \in \{\boldsymbol{x}_j\}_{j=1}^N : \boldsymbol{x}_k \text{ is labeled 'infeasible'}\}.$$

The main idea of the proposed algorithm is to choose super- (sub-) level sets of $\psi(\boldsymbol{x}, \boldsymbol{\beta}^*_N)$ and ensure they contain no feasible (infeasible) samples. The inner approximation

$$\mathbb{F}^+ \triangleq \{\boldsymbol{x} \in \mathbb{R}^{n_x} : \psi(\boldsymbol{x}, \boldsymbol{\beta}^*_N) > \varepsilon^+\},$$

is obtained by solving

$$\varepsilon^+ = \arg\min_{\varepsilon > 0} \ \varepsilon \tag{9a}$$

$$\text{subject to:} \quad \psi(\boldsymbol{x}_k, \boldsymbol{\beta}^*_N) < \varepsilon, \quad \forall\, \boldsymbol{x}_k \in \mathcal{X}^-_N. \tag{9b}$$

Similarly, the outer approximation has the form

$$\mathbb{F}^- \triangleq \{\boldsymbol{x} \in \mathbb{R}^{n_x} : \psi(\boldsymbol{x}, \boldsymbol{\beta}^*_N) < \varepsilon^-\}.$$

where $\varepsilon^-$ is obtained by solving

$$\varepsilon^- = \arg\max_{\varepsilon < 0} \ \varepsilon \tag{10a}$$

$$\text{subject to:} \quad \psi(\boldsymbol{x}_k, \boldsymbol{\beta}^*_N) > \varepsilon, \quad \forall\, \boldsymbol{x}_k \in \mathcal{X}^+_N. \tag{10b}$$

**Remark 5.** It is important to note that with small number of samples $N$, our inner approximation $\mathbb{F}^-$ is based only on the feasibility information obtained by the samples $\{x_j\}_{j=1}^N$. Hence, $\mathbb{F}^-$ will not generally be an inner approximation of the actual feasible region. We prove in the sequel, however, that as $N$ increases, our inner approximation $\mathbb{F}^-$ converges to a strict inner approximation of the actual feasible region $\mathbb{F}$.

**Remark 6.** A possible method for increasing the accuracy of classification near the feasible region boundary is to sample more densely around the optimal SVM classification boundary $\psi(\boldsymbol{x}, \boldsymbol{\beta}^*_N) = 0$. One algorithm which can be used to do this is discussed in [38], where the authors use adaptive sampling.

### *D. Convergence of SVM-based Estimator*

Now we discuss convergence properties for the SVM employing a class of kernel functions. We use the following definition of universal kernels presented in [39].

**Definition 4 (Universal Kernel).** *A continuous kernel $\mathcal{K}$ is universal if the space of all functions induced by $\mathcal{K}$ is dense in the space of all continuous functions defined on $\mathbb{X}$.*

Note that a function $\psi_{\mathcal{K}}(\boldsymbol{x}, \boldsymbol{\beta}) : \mathbb{X} \to \mathbb{H}$ is induced by a kernel $\mathcal{K}$ implies that there is an element $\Gamma(\boldsymbol{x}) \in \mathbb{H}$ such that $\psi_{\mathcal{K}} = \langle\Gamma(\boldsymbol{x}), \Gamma(\cdot)\rangle_{\mathbb{H}}$. From Definition 4, we deduce that for every continuous function $\psi$ and $\varepsilon > 0$, there exists a function $\psi_{\mathcal{K}}$ induced by $\mathcal{K}$ such that $|\psi - \psi_{\mathcal{K}}|_\infty < \varepsilon$. As discussed in [39], the induced function $\psi_{\mathcal{K}}$ can be written in the form (7).

**Remark 7.** A commonly used universal kernel function that fulfills Mercer's condition is the Gaussian Radial Basis Function (GRBF) with kernel variance $\sigma^2$ [39] of the form

$$\mathcal{K}(\boldsymbol{x}_k, \boldsymbol{x}) = \exp\left(-\frac{\|\boldsymbol{x}_k - \boldsymbol{x}\|^2}{2\sigma^2}\right). \tag{11}$$

The following important property of universal kernels justifies the application of the SVM to the feasibility boundary estimation problem.

**Proposition 1.** *Every universal kernel separates all compact subsets of $\mathbb{X}$.*

*Proof.* See [39]. □

It immediately follows from Proposition 1 that the GRBF kernel can separate any pairwise disjoint compact subsets in $\mathbb{X}$. Specifically, there exists some $\boldsymbol{\beta}$ and $\nu > 0$ such that the decision function $\psi_{\mathcal{K}}$ induced by $\mathcal{K}$ separates the inner and outer approximations, that is

$$\psi(\boldsymbol{x}, \boldsymbol{\beta}) \ge +\nu \;\; \forall \boldsymbol{x} \in \mathbb{F}^+,$$
$$\psi(\boldsymbol{x}, \boldsymbol{\beta}) \le -\nu \;\; \forall \boldsymbol{x} \in \mathbb{F}^-.$$

Proposition 1 guarantees that the universal kernel induces a separating function $\psi$, but does not provide a computational method for determining $\psi$. Herein, we provide guarantees regarding the separating function $\psi$ defined in (8) by solving the convex program (6) with feasibility information obtained from low-discrepancy samples. Specifically, we verify that increasing the number of samples $N$ for the SVM with a universal kernel results in an increasingly accurate estimate of the feasible region, and reduces the conservativeness of the inner and outer approximations. To this end, we modify the arguments presented in [39] to the case when the training data $\{\boldsymbol{x}_j\}_{j=1}^N$ is extracted from a low-discrepancy sequence.

Recall that $\zeta(\boldsymbol{x})$ is the true boundary function described in Assumption 6 and $\psi(\boldsymbol{x}, \boldsymbol{\beta}_N)$ is the separating function with coefficient vector $\boldsymbol{\beta}_N$.

**Theorem 1.** *Suppose Assumptions 1, 5 and 6 hold. Let $\mathcal{K}$ be a universal kernel on $\mathbb{X}$ and $\psi(\boldsymbol{x}, \boldsymbol{\beta}_N^*)$ be the SVM decision function described in (7) with $\boldsymbol{\beta}_N^*$ obtained by solving (6). Then for every regularization parameter $L > 0$ in the optimization problem (5) and compact subsets*

$$F^+ \subset \{\boldsymbol{x} \in \mathbb{X} : \zeta(\boldsymbol{x}) > 0\} \text{ and } F^- \subset \{\boldsymbol{x} \in \mathbb{X} : \zeta(\boldsymbol{x}) < 0\},$$

*there exists a sufficiently large number of samples $N_0$ such that if $N \ge N_0$, then*

$$\begin{cases} \psi(\boldsymbol{x}, \boldsymbol{\beta}_N^*) > 0 & \text{for } \boldsymbol{x} \in F^+ \\ \psi(\boldsymbol{x}, \boldsymbol{\beta}_N^*) < 0 & \text{for } \boldsymbol{x} \in F^-. \end{cases} \tag{12}$$

*Proof.* See Appendix. □

**Remark 8.** We provide a lower bound on the number of samples $N$ which results in correct classification of the each of the sets $A$ in the covering used in the proof of Theorem 1. Let $r \in (0, 1)$ and choose $N$ such that

$$D_N \le r\frac{\mathrm{Vol}(A)}{\mathrm{Vol}(\mathbb{X})} \quad \text{and} \quad N \ge \frac{\lfloor 2/L(\mu^*)^2 \rfloor}{1-r}\frac{\mathrm{Vol}(\mathbb{X})}{\mathrm{Vol}(A)}.$$

Then, the condition $\#A_N \ge \lfloor 2/L(\mu^*)^2 \rfloor = m$ is satisfied.

Next, we show that Theorem 1 can be applied to ensure that strict inner approximations of the feasible region are obtained by sufficiently sampling $\mathbb{X}$ using low-discrepancy sequences.

**Corollary 1.** *Let $\varepsilon > 0$. The construction in Theorem 1 produces a classifier that classifies all points in $\{\boldsymbol{x} \in \mathbb{X} : \zeta(\boldsymbol{x}) \ge \varepsilon\}$ as feasible, and all infeasible points as infeasible.*

The proof for a strict outer approximation is identical and uses sub-level sets instead of super-level sets.

**Remark 9.** As seen in the proof of Corollary 1, there is some region between the feasible and infeasible region that cannot be classified using the SVM. This is due to the margin $\mu^*$ that is inherent in the SVM. As $N$ increases, $\mu^*$ decreases.

From the results presented in this section, we conclude that the SVM classifier equipped with universal kernels can approximate continuous feasible region boundaries with arbitrarily high accuracy. We also propose an algorithm for obtaining a strict inner/outer approximation of $\mathbb{F}$ and provide performance guarantees for the same.

## V. Explicit Model Predictive Controller Design

In general, the approximation of an arbitrary discontinuous function is a challenging problem. From [40], we know that the NMPC $\boldsymbol{u}^*$ may not be continuous everywhere on $\mathbb{F}$. Thus, we begin by restricting the class of NMPC control surfaces considered in this paper.

In particular, let

$$\mathcal{D}_u = \{\boldsymbol{x} \in \mathbb{F} : \boldsymbol{u}^* \notin \mathcal{C}^1 \text{ at } \boldsymbol{x}\}.$$

Then, we define $\mathcal{W}$ as the set of $\boldsymbol{x}_0 \in \mathbb{F}$ such that the NMPC-controlled trajectories with initial condition $\boldsymbol{x}_0$ *do not intersect* $\mathcal{D}_u$. We refer to $\mathcal{W}$ as the **feasible subregion**.

The algorithm employed for constructing the ENMPC $\hat{\boldsymbol{u}}(\boldsymbol{x})$ depends on whether $\mathcal{W}$ is a regular domain (boxes/hypercubes) or an arbitrary domain. If $\mathcal{W}$ is regular, we can directly use sparse-grid interpolation. If $\mathcal{W}$ is an arbitrarily-shaped domain, we use a regression surface to construct the ENMPC. To proceed, we make assumptions on the NMPC control surface and the feasible subregion.

**Assumption 7.** *The set $\mathcal{D}_u$ is closed, and the NMPC control $\boldsymbol{u}^*$ is $\mathcal{C}^1$ on some neighborhood of the origin. That is, there exists some $\varepsilon > 0$ such that $\boldsymbol{u}^* \in \mathcal{C}^1$ for all $\boldsymbol{x} \in \mathbb{B}_\varepsilon(\mathbf{0})$.*

This assumption implies that any two points in $\mathcal{W}$ can be driven to the terminal region using the NMPC without intersecting $\mathcal{D}_u$, which implies that the set $\mathcal{W}$ is path-connected.

**Assumption 8.** *The feasible subregion $\mathcal{W}$ is open and dense in $\mathbb{F}$.*

For control surfaces that are $\mathcal{C}^1$ except for jump discontinuities, a sparse-grid based algorithm to identify a neighborhood of $\mathcal{D}_u$ is discussed in detail in [41], [42]. Once a neighborhood of $\mathcal{D}_u$ is estimated, we can determine $\mathcal{W}$ by labeling a sample $\boldsymbol{x}_j$ as '-1' if the NMPC controlled trajectories intersect this neighborhood of $\mathcal{D}_u$, or are infeasible; and, '+1' if it can be controlled to the terminal region without intersecting this neighborhood of $\mathcal{D}_u$ within the predictive horizon. Based on these labels, an inner approximation of $\mathcal{W}$ can be constructed using the SVM as discussed in Section IV.

**Remark 10.** If the NMPC is $\mathcal{C}^1$ on all of $\mathbb{F}$, then the feasible domain is $\mathcal{W} = \mathbb{F}$. However, in case of discontinuities on the NMPC surface, we have to construct $\mathcal{W}$ as discussed here.

### *A. Sparse-grid interpolation-based ENMPC on regular domains*

In the case where the feasible region is regular, we use sparse-grid interpolation schemes to approximate the ENMPC control surface. We present our implementation of the sparse-grid interpolation algorithm.

Without loss of generality, we consider $\mathcal{W} = [-1,1]^{n_x}$ as in [43]. We wish to approximate the NMPC control law $\boldsymbol{u}^*(\boldsymbol{x})$ on $\mathcal{W}$.

Let the $j$th component of $\boldsymbol{u}^*(\boldsymbol{x})$ be denoted $\boldsymbol{u}_j^*(\boldsymbol{x}) : \mathcal{W} \to \mathbb{R}$, where $j = 1,\ldots,n_u$. We consider a sequence of approximations $\{\mathcal{V}_i\}$ of $\boldsymbol{u}_j^*(\boldsymbol{x})$, where $i$ belongs to some index set $\mathcal{I}$. The accuracy of the approximation improves with increasing $i$. These $\mathcal{V}_i$ are defined by the type of sparse-grid selected and parameterized by $m_i$ number of nodes. For simplicity, suppose $n \triangleq n_x = 1$. The $i$th interpolated function in the sequence $\{\mathcal{V}_i\}$ has a basis expansion,

$$\mathcal{V}_i(\boldsymbol{u}_j^*, x) = \sum_{k=1}^{m_i} \boldsymbol{u}_j^*(x_k^i)\mathcal{L}_k^i(x), \tag{13}$$

where $i \in \mathbb{N}$, $\boldsymbol{u}_j^*(x_k^i)$ is the optimal NMPC control action obtained by solving (3) at the $k$th node of the sparse grid, $x_k^i \in [-1,1]$ and $\mathcal{L}_k^i(\boldsymbol{x})$ is a Lagrange polynomial such that $\mathcal{L}_k^i(x_j^i) = \delta_{kj}$, the Dirac delta function.

When $n > 1$, tensor products of (13) in higher dimensions yields,

$$\begin{aligned}\mathcal{V}_{\boldsymbol{i}}(\boldsymbol{u}_j^*, \boldsymbol{x}) &= (\mathcal{V}_{i_1}\otimes\cdots\otimes\mathcal{V}_{i_n})(\boldsymbol{u}_j^*, \boldsymbol{x}), \qquad (14)\\ &= \sum_{k_1=1}^{m_{i_1}}\cdots\sum_{k_n=1}^{m_{i_n}} \boldsymbol{u}_j^*(x_{k_1}^{i_1},\cdots,x_{k_n}^{i_n})(\mathcal{L}_{k_1}^{i_1}\otimes\cdots\otimes\mathcal{L}_{k_n}^{i_n})(\boldsymbol{x}),\\ &\triangleq \sum_{k_n=1}^{m_{i_n}} \boldsymbol{u}_j^*(x_{k_1}^{i_1},\cdots,x_{k_n}^{i_n})\mathbb{T}_{\boldsymbol{k}}^{\boldsymbol{i}}(\boldsymbol{x})\end{aligned}$$

where $\boldsymbol{i} = (i_1,\ldots,i_n)$ and $\boldsymbol{k} = (k_1,\ldots,k_n)$ are multi-index vectors. Linear combinations of these formulas produce the Smolyak formulas [43], [44]. Let $U^0 = 0$ and

$$\Delta^{i_k}(\boldsymbol{u}_j^*, \boldsymbol{x}) = \mathcal{V}_{i_k}(\boldsymbol{u}_j^*, \boldsymbol{x}) - \mathcal{V}_{i_k-1}(\boldsymbol{u}_j^*, \boldsymbol{x})$$

for $i_k \in \mathbb{N}$, $|\boldsymbol{i}| = \sum_{k=1}^n i_k$. Let the sparse-grid approximated function be $\hat{\boldsymbol{u}}_j(q,n) = \sum_{n\le|\boldsymbol{i}|\le q}(\Delta^{i_1}\otimes\cdots\otimes\Delta^{i_n})(\boldsymbol{u}_j^*, \boldsymbol{x})$, where $q \ge n$ is a parameter that determines the configuration of the nodes in the sparse grids. It is worth noting that to compute $\hat{\boldsymbol{u}}(q,n)$, one only needs to know function values at the sparse grid $\bigcup_{n<|\boldsymbol{i}|\le q}\mathcal{X}^{i_1}\otimes\cdots\otimes\mathcal{X}^{i_n}$, where $\mathcal{X}^i$ denotes the set of points used by $\mathcal{V}_i$.

**Theorem 2.** *If Assumption 7 holds, then*

$$\|\hat{\boldsymbol{u}}_j(q,n) - \boldsymbol{u}_j^*\| \le \frac{c_n}{N}\left(\|u_j^*\|_\infty + \|\nabla\boldsymbol{u}_j^*\|_\infty\right)(\ln N)^{3n-2},$$

*where $N \triangleq N(q,n)$ is the number of points in the sparse grid employed for constructing $\hat{\boldsymbol{u}}_j(\boldsymbol{x})$.*

*Proof:* See [43]. ■

**Remark 11.** Theorem 2 implies that the error in the approximating function decreases roughly in the order of the inverse of the number of sampled points (up to a logarithmic factor). If $\boldsymbol{u}_j^* \in \mathcal{C}^k$, then the convergence is $\mathcal{O}(1/N^k)$ up to a logarithmic factor.

**Remark 12.** Although (13) is written in terms of Lagrange polynomials to emphasize the role of function evaluations at the nodes $x_k^i$, it has been demonstrated in [45] that Legendre polynomials provide an alternative basis with an improvement in computational times for sparse-grid interpolation.

Next, we discuss the case when the feasible region is not a box or hypercube.

### *B. Regression-based ENMPC on domains of arbitrary shapes*

An arbitrary region can be modeled as a regular grid with missing information at a subset of the nodes. This case generally arises when $\mathcal{W} \neq \mathbb{X}$ and so no optimal NMPC control actions are computed for nodes in $\mathbb{X}\cap\mathcal{W}^c$. To overcome the difficulty of constructing an interpolant with missing function values, which is a serious obstacle for standard sparse-grid interpolation, we sample (more densely, if required) within the domain $\mathcal{W}$ using a low-discrepancy sequence. Even then there are subtleties to consider since low-discrepancy sequences are defined in terms of product intervals and convergence results in higher dimensions are fairly delicate. Nevertheless, in this subsection, we demonstrate that if we select a sufficient number of low-discrepancy samples in $\mathcal{W}$, a linear regression based ENMPC $\hat{\boldsymbol{u}}(\boldsymbol{x})$ will converge to the optimal NMPC $\boldsymbol{u}^*(\boldsymbol{x})$ on $\mathcal{W}$, provided Assumptions 7 and 8 hold.

Let $\mathbb{T}_{\boldsymbol{i}}^{\boldsymbol{k}}$ denote a tensored polynomial associated with a predefined sparse-grid on a regular domain $\mathbb{X}$ given by

$$\mathbb{T}_{\boldsymbol{k}}^{\boldsymbol{i}}(\boldsymbol{x}) \triangleq \left(\mathfrak{C}_{k_1}^{i_1}\otimes\cdots\otimes\mathfrak{C}_{k_n}^{i_n}\right)(\boldsymbol{x}), \tag{15}$$

where $\mathfrak{C}_k^i(\cdot)$ denotes the Chebyshev polynomial basis element associated with indices $i,k \in \mathbb{N}$. We construct a sequence of basis functions by assigning each pair $(\boldsymbol{i},\boldsymbol{k})$ of $\mathbb{T}_{\boldsymbol{i}}^{\boldsymbol{k}}$ to a positive integer $r \in \mathbb{N}$. This is done over all possible multi-indices $(\boldsymbol{i},\boldsymbol{k})$ and we denote each $\mathbb{T}_{\boldsymbol{i}}^{\boldsymbol{k}}$ as $\mathbb{T}_r$. We express the regression based ENMPC as a linear combination of these $M$ basis functions. Hence, for $\boldsymbol{c} \in \mathbb{R}^M$, we have

$$\hat{\boldsymbol{u}}_M(\boldsymbol{x}) = \sum_{j=1}^M \boldsymbol{c}_j\mathbb{T}_j(\boldsymbol{x}). \tag{16}$$

Next, we define the matrices

$$\boldsymbol{G}_{N,M} \triangleq \begin{bmatrix} \mathbb{T}_1(\boldsymbol{x}_1) & \dots & \mathbb{T}_M(\boldsymbol{x}_1) \\ \vdots & \ddots & \vdots \\ \mathbb{T}_1(\boldsymbol{x}_N) & \dots & \mathbb{T}_M(\boldsymbol{x}_N) \end{bmatrix} \text{ and } \boldsymbol{u}^*_N \triangleq \begin{bmatrix} \boldsymbol{u}^*(\boldsymbol{x}_1) \\ \vdots \\ \boldsymbol{u}^*(\boldsymbol{x}_N) \end{bmatrix}$$

for all $M, N \in \mathbb{N}$.

Then, an estimate for the coefficients of $\hat{\boldsymbol{u}}_{N,M}$ with $N$ samples and $M$ basis functions is computed by solving the convex problem:

$$\boldsymbol{c}^* = \arg\min_{\boldsymbol{c}} \|\boldsymbol{G}_{N,M}\boldsymbol{c} - \boldsymbol{u}^*_N\|_\infty, \tag{17a}$$

$$\text{subject to:} \quad \|\boldsymbol{c}\|_\infty \leq \bar{c} \tag{17b}$$

for some $\bar{c} \in (0, \infty)$. Hence, the ENMPC action can be computed as:

$$\boxed{\hat{\boldsymbol{u}}^*_{N,M}(\boldsymbol{x}) = \sum_{j=1}^{M} \boldsymbol{c}^*_j \mathbb{T}_j(\boldsymbol{x}).}$$

For the following convergence result, we restrict the $\boldsymbol{x}_j$'s to some slightly smaller compact set.

**Theorem 3.** *Suppose Assumptions 5–8 hold. Let $\mathcal{W}_0$ and $\mathcal{W}_1$ be compact sets satisfying $\mathcal{W}_0 \subset \operatorname{int} \mathcal{W}_1 \subset \mathcal{W}_1 \subset \mathcal{W}$. Then there exists a scalar $\bar{c} > 0$ such that for every $\varepsilon > 0$, there exist $M_0, N_0 \in \mathbb{N}$ such that the solution to (17) using only samples in $\mathcal{W}_1$ satisfies*

$$\|\hat{\boldsymbol{u}}^*_{N,M} - \boldsymbol{u}^*\|_{\mathbb{L}^\infty(\mathcal{W}_0)} < \varepsilon$$

*for all $M \geq M_0$ and $N \geq N_0$.*

*Proof.* See Appendix. □

The difficulty in getting uniform convergence of the ENMPC $\hat{\boldsymbol{u}}_{N,M}(\boldsymbol{x})$ to the NMPC $\boldsymbol{u}^*(\boldsymbol{x})$ in $\mathcal{W}$ using a polynomial basis is due to the arbitrary shape of $\mathcal{W}$. We first restrict $\boldsymbol{u}^*(\boldsymbol{x})$ to a compact subset of $\mathcal{W}$, and extend the function to be smooth on a hypercube, which implies the existence of a uniformly convergent Chebyshev series. However, this series is not unique since the extension is not unique. Moreover, in practice, we can take samples only within a compact subset of the feasible subregion, which means that we have no way of recovering any such Chebyshev series from samples of the optimal NMPC $\boldsymbol{u}^*(\boldsymbol{x})$. Instead, we minimize the maximum error on sampled points using an *a priori* upper bound on maximum coefficient in a Chebyshev expansion. The existence of a uniformly convergent Chebyshev series guarantees that this can be done with one single upper bound, independent of the number of basis elements and the number of sampled points. Using properties of the low-discrepancy sequence, for a fixed finite set of basis polynomials we are able to increase $N$ to obtain a uniform error on the compact subset of $\mathcal{W}$ that is at most that of the corresponding truncation of the uniformly convergent Chebyshev series. Increasing the number of basis elements gives us the uniform approximation we seek.

**Remark 13.** From an implementation perspective, the Chebyshev polynomials may be replaced with other orthogonal polynomial basis functions such as Hermite, Jacobi or Legendre polynomials.

### C. Feasibility and stability guarantees

In this subsection, we use the notation $\hat{\boldsymbol{u}}_{N,M}$ for the ENMPC controller (irrespective of construction using interpolation/regression) computed with $N$ samples and $M$ basis functions. Recall $\boldsymbol{u}^*$ denotes the optimal NMPC. We provide feasibility and stability guarantees of the closed-loop system (1). In order to guarantee input feasibility, we use a projection of $\hat{\boldsymbol{u}}_{N,M}$ onto the closed, convex set $\mathbb{U}$ of admissible control actions. The **nearest-point projection operator** onto $\mathbb{U}$ is denoted by $\operatorname{Proj}_{\mathbb{U}}(\hat{\boldsymbol{u}}_{N,M}) = \arg\min_{\boldsymbol{u}\in\mathbb{U}} \|\hat{\boldsymbol{u}}^*_{N,M} - \boldsymbol{u}\|$.

**Theorem 4.** *Suppose Assumptions 1–8 hold and let $\mathcal{W}_0$ be compact with*

$$\bar{\mathbb{X}}_T \subset \operatorname{int} \mathcal{W}_0 \subset \mathcal{W}_0 \subset \mathcal{W}.$$

*Then for sufficiently large $M$ and $N_0 \in \mathbb{N}$, and $N \geq N_0$, the closed-loop trajectories of the plant (1) with control $\boldsymbol{u} = \operatorname{Proj}_{\mathbb{U}}(\hat{\boldsymbol{u}}^*_{N,M})$ and initial condition $\boldsymbol{x}_0 \in \mathcal{W}_0$ are feasible for all $t \geq t_0$ and are asymptotically stable to the origin.*

*Proof.* See Appendix. □

**Remark 14.** We present our result for compact subsets in the interior of $\mathbb{F}$ in order to ensure that there is a non-zero distance $\varepsilon$ between $\mathcal{W}_0$ and the infeasible region. By ensuring that the deviation between the optimal NMPC controlled trajectories and the ENMPC controlled trajectories are within $\varepsilon$, we can provide feasibility guarantees for the ENMPC driven system.

### D. Implementation considerations

For implementation on digital systems, suppose the ENMPC is applied in a piecewise constant manner with a sampling time of $\delta > 0$, that is,

$$\hat{\boldsymbol{u}}^\delta(\boldsymbol{x}(t)) = \hat{\boldsymbol{u}}^*_{N,M}(\boldsymbol{x}(m\delta)), \quad \text{for } t \in [m\delta, (m+1)\delta), \tag{18}$$

where $m \in \mathbb{N}$ and $\boldsymbol{u}_N$ is an ENMPC constructed using $N$ samples. It remains to be shown that if we use the piecewise constant ENMPC controller $\hat{\boldsymbol{u}}^\delta$, then there exists a sufficiently small sampling time $\delta$ such that this feasibility result holds for the closed-loop trajectories of (1).

**Corollary 2.** *Suppose the conditions of Theorem 4 hold and $M$ is fixed. Then there exists some large $N_0 \in \mathbb{N}$ and small $\delta > 0$ such that the closed-loop trajectory $\boldsymbol{x}(t)$ controlled by the ENMPC $\operatorname{Proj}_{\mathbb{U}}(\hat{\boldsymbol{u}}^\delta_{N,M})$ is feasible for all $t \geq t_0$, $\delta' \in (0, \delta]$ and all $N \geq N_0$.*

*Proof.* See Appendix. □

Corollary 2 implies that with a sufficiently small sampling time, we can implement our piecewise constant ENMPC without losing feasibility guarantees. A pseudo-code to assist implementation is provided in the Appendix.

## VI. Examples

In this section, we test our proposed control scheme on two nonlinear systems. All computations are performed on `MATLAB R2014b`. All convex problems are solved using the `YALMIP` toolbox [46]. Off-line solutions of the FHOCP are computed using the `GODLIKE` toolbox [47]. SVM classification is performed using MATLAB's in-built `svmtrain` and `svmclassify` functions and the `LibSVM` toolbox [48] for larger datasets. Sparse-grid construction and interpolation are performed using the `Sparse-Grid Interpolation Toolbox` [49]. The parameters required to compute $N$ are the grid-type, the maximum depth and a user-defined upper and lower bounds on the number of nodes. We employ the Newton-Cotes grid to ensure that the sequence of sparse-grid nodes is a low-discrepancy sequence as reported in [50]. The Newton-Cotes grid comprises of nested, equidistant nodes which demonstrate slow growth with increasing state-space dimension [51].

### A. *Example 1: 2-D Model*

We use a dynamical system model that has been explored previously in [8], [34], [52], [53] and is a modification of the model proposed in [54]. The system is described by,

$$\begin{bmatrix}\dot{x}_1\\ \dot{x}_2\end{bmatrix} = \begin{bmatrix}0 & 1\\ 1 & 0\end{bmatrix}\begin{bmatrix}x_1\\ x_2\end{bmatrix} + \begin{bmatrix}0.5+0.5x_1\\ 0.5-2x_2\end{bmatrix}u. \tag{19}$$

The state-space polytope is given by

$$\mathbb{X} = \{\boldsymbol{x} \in \mathbb{R}^2 : \|\boldsymbol{x}\|_\infty \le 1\},$$

and the space of admissible control actions is

$$\mathbb{U} = \{u \in \mathbb{R} : |u| \le 2\}.$$

To estimate the feasible region $\mathbb{F}$, we first require a terminal region $\mathbb{X}_T$. Several methods exist for deriving the terminal region, for example [34], [55]. Our feasible stabilizing controller gain within $\mathbb{X}_T$ is given by $\boldsymbol{K} = \begin{bmatrix}2 & 2\end{bmatrix}$, and the terminal region is parametrized by,

$$\boldsymbol{x}^\top \boldsymbol{P}\boldsymbol{x} = \boldsymbol{x}^\top \begin{bmatrix}1.3333 & 0.6667\\ 0.6667 & 1.3333\end{bmatrix}\boldsymbol{x} \le 0.0762.$$

We use a predictive horizon of $T_f = 2$, and solve the nonlinear constrained optimization problem (3) with $\boldsymbol{Q} = \boldsymbol{I}_2$ and $R = 0.1$. In order to pose the optimization problem (3) as a finite-dimensional problem, we use a piecewise continuous (spline interpolated) NMPC with optimization variables sampled every $\delta = 0.1$ s. For feasible region approximation, we use a GRBF kernel SVM (11) with $\sigma = 0.8$. A comparison of average classification accuracy and average classification time with increasing number of nodes for 1000 runs is provided in Table I. We observe the rapid decrease in classification error without a significant rise in the computation time required for constructing the SVM. We test the accuracy of the method and demonstrate the estimated inner and outer approximations of the feasible region with $N = 1792$ nodes in Figure 1. We perform sparse-grid based regression with $N = 1792$ nodes as described in Section V with the stored values of control values at the grid nodes. The constructed ENMPC surface is shown in Figure 1.

TABLE I

COMPUTATIONAL RESULTS OF FEASIBILITY BOUNDARY ESTIMATOR ON EXAMPLE 1 AND 2. THE SCALAR $T_f$ IS THE PREDICTIVE HORIZON IN SECOND, $N$ IS THE TOTAL NUMBER OF SAMPLES USED, AND 'TIME' DENOTES THE TIME REQUIRED TO CONSTRUCT THE SVM.

| $T_f$ (s) | Max Depth | $N$ | Time (s) |
|---|---|---|---|
| 4 | 5 | 80 | 0.141 |
| 4 | 6 | 176 | 0.291 |
| 4 | 7 | 384 | 0.354 |
| 4 | 8 | 832 | 0.474 |
| 4 | 9 | 1792 | 0.858 |

We now simulate the closed-loop system from 80 different initial points selected randomly within the inner approximation of the feasible region. The results of the simulations are provided in Figure 1. We observe that the control scheme is successful and drives all 80 initial conditions to the origin successfully. It remains to be seen how much optimality is lost using the ENMPC in comparison with an optimal NMPC with identical $\boldsymbol{Q}, \boldsymbol{R}, \mathbb{X}_T, T_f$. In order to compare the performance of these two controllers, we select 50 random, unique initial conditions which do not intersect with pre-selected nodes (so that the $u(\boldsymbol{x})$ at each point is an approximation, not a pre-calculated optimal value) and simulate the controlled uncertain system with an on-line NMPC and our ENMPC. The NMPC takes $13.6 \pm 0.5$ (mean ± standard deviation) seconds to solve (3) using the population-based global search algorithm `GODLIKE`, whereas the sparse-regression based ENMPC required $11 \pm 2$ $\mu$s to compute a control action. The trade-off is a reduction in optimality of the control solution. The squared-error stage cost over all 50 trajectories from the origin are 5.24 and 4.97 for the ENMPC and NMPC, respectively. Thus, for a large reduction in on-line computation time, we sacrifice a reasonable $10\%$ of our optimal cost.

### B. *Example 2: 8-D Model*

This example illustrates the efficacy of the controller on a higher-dimensional system with multiple control inputs. We aim to stabilize the following randomly generated eight-dimensional (8-D) uncertain nonlinear system with two control inputs,

$$\dot{\boldsymbol{x}} = \boldsymbol{A}_0\boldsymbol{x} + \boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{B}_0\boldsymbol{u}, \tag{23}$$

where $\boldsymbol{A}_0, \boldsymbol{f}, \boldsymbol{B}_0$ are defined in (20). The constrained state space is $\mathbb{X} = \{\boldsymbol{x} \in \mathbb{R}^8 : \|\boldsymbol{x}\|_\infty \le 1\}$ and $\mathbb{U} = \{\boldsymbol{u} \in \mathbb{R}^2 : \|\boldsymbol{u}\|_\infty \le 2\}$. We use $N = 3088$ samples and construct the inner and outer approximations of the feasible region with a GRBF kernel SVM with $\sigma = 0.9$. We construct two ENMPC control surfaces, (since $m = 2$). The construction of the ENMPC with a predictive horizon of $T_f = 2$, $\boldsymbol{Q} = \boldsymbol{I}_8$ and $\boldsymbol{R} = 0.1\boldsymbol{I}_2$ requires 2 hours off-line. Of the total 3088 nodes, 2857 were classified as feasible. The SVM classification times for varying samples is reported in Table II.

The performance of the ENMPC is illustrated on the closed-loop system in Figure 2 for 50 randomly selected initial conditions inside $F^-$. We note that the ENMPC successfully drives the system to the origin. Next, we compare the computational time and performance of the ENMPC with the NMPC (applied to the nominal system) with identical design parameters $T_f, \delta, \boldsymbol{Q}$, and $\boldsymbol{R}$. The ENMPC control action was computed in an average of $0.28 \pm 0.07$ ms, whereas a similar computation takes $67.2 \pm 4.1$ s if computed with `GODLIKE`

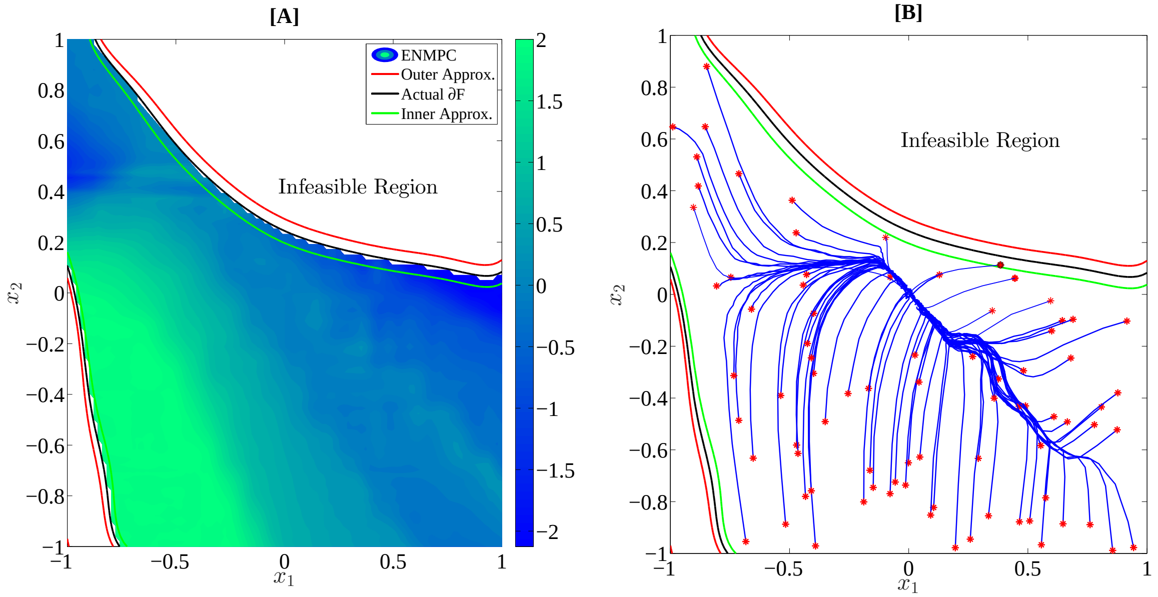

Fig. 1. **A.** SVM-generated $\partial\mathbb{F}$ employing $N = 1792$. The actual feasibility region boundary $\partial\mathbb{F}$ is shown with the black lines determined using a uniform grid with $N = (100)^2 = 10^4$ points. The inner approximation using the SVM classifier is depicted with green lines, and the outer approximation is depicted with red. The ENMPC surface constructed using regression is illustrated within the feasible region. **B.** Controlled trajectories for the nonlinear system of Example 1. The red '*' indicate the 80 randomly selected initial states, and dark-blue lines are controlled trajectories on the state-space.

$$\boldsymbol{A}_0 = \begin{bmatrix} -0.72 & -0.61 & 0.19 & -0.59 & 0.24 & -0.58 & 0.03 & 0.03 \\ 0.02 & -0.48 & -0.23 & 0.00 & 0.07 & -0.01 & -0.30 & -0.17 \\ 0.89 & 0.63 & -0.17 & 1.31 & -0.39 & 0.85 & 0.83 & 0.56 \\ 0.57 & 0.76 & 0.26 & 0.32 & -0.15 & 0.90 & 0.44 & 0.41 \\ -1.05 & -0.29 & -0.82 & -1.13 & -1.13 & -0.96 & -0.79 & -0.87 \\ -0.40 & -0.23 & -0.11 & -0.35 & 0.01 & -1.01 & -0.21 & -0.22 \\ -0.61 & -0.67 & -0.43 & -1.16 & 0.13 & -0.95 & -1.33 & -0.47 \\ 0.51 & 0.40 & 0.24 & 0.79 & 0.21 & 0.88 & 0.39 & -0.22 \end{bmatrix}, \quad \boldsymbol{f} = \begin{bmatrix} -x_2x_5 \\ 0 \\ 0 \\ 0 \\ 0 \\ x_3^2 \\ 0 \\ 0 \end{bmatrix}, \quad \boldsymbol{B}_0 = \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix}, \tag{20}$$

$$\boldsymbol{P} = \begin{bmatrix} 1.11 & 0.09 & -0.79 & -0.45 & 0.56 & 0.09 & 0.71 & -0.19 \\ 0.09 & 1.11 & -0.78 & -0.42 & 0.55 & 0.08 & 0.70 & -0.18 \\ -0.79 & -0.78 & 7.14 & 3.41 & -4.35 & -0.67 & -5.54 & 1.45 \\ -0.45 & -0.42 & 3.41 & 2.90 & -2.41 & -0.37 & -3.07 & 0.81 \\ 0.56 & 0.55 & -4.35 & -2.41 & 4.08 & 0.47 & 3.92 & -1.03 \\ 0.09 & 0.08 & -0.67 & -0.37 & 0.47 & 1.07 & 0.60 & -0.16 \\ 0.71 & 0.70 & -5.54 & -3.07 & 3.92 & 0.60 & 5.99 & -1.31 \\ -0.19 & -0.18 & 1.45 & 0.81 & -1.03 & -0.16 & -1.31 & 1.35 \end{bmatrix}, \tag{21}$$

$$\boldsymbol{K} = \begin{bmatrix} 3.79 & 4.14 & 2.15 & 5.86 & -1.02 & 3.88 & 0.63 & 0.87 \\ 0.87 & -1.80 & 2.53 & 1.03 & -0.88 & 0.20 & 1.94 & 3.52 \end{bmatrix}. \tag{22}$$

and $4.1 \pm 0.7$ s using local methods such as MATLAB's `fmincon`. Thus, we achieve high computational speeds with a $10\%$ degradation in optimal cost using the ENMPC in this 8-D model over 50 simulation runs.

TABLE II
COMPUTATIONAL RESULTS OF FEASIBILITY BOUNDARY ESTIMATOR ON EXAMPLE 2.

| $T_f$ (s) | Max Depth | $N$ | Time (s) |
|---|---|---|---|
| 2 | 3 | 704 | 0.457 |
| 2 | 4 | 3088 | 5.699 |
| 2 | 5 | 11776 | 22.478 |

## VII. CONCLUSIONS

In this paper, we develop a support vector machine-informed methodology to construct explicit model predictive controllers for the stabilization of nonlinear systems with feasibility and stability guarantees. The low storage and computational costs make this ENMPC design methodology attractive for implementation in low-memory devices to stabilize nonlinear systems with fast dynamics. Additionally, from an implementation perspective, our proposed method offers a separation between the computation of the NMPC control actions and the construction of the ENMPC. For example, the NMPC control actions could be computed using non-quadratic cost functions or time varying constraints. Our method for constructing the ENMPC using regression requires only the control values at

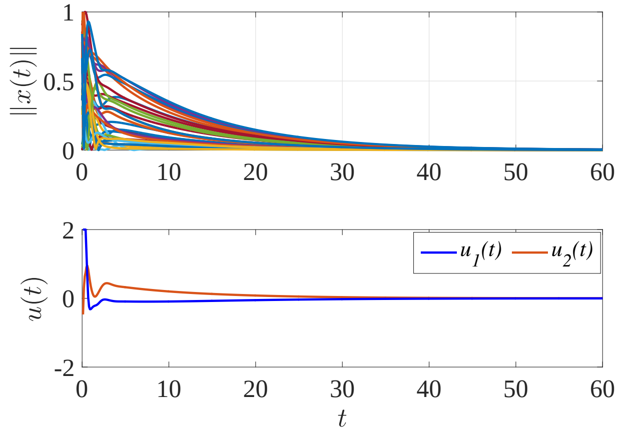

Fig. 2. (*Top*) Trajectories of the controlled 8-D system for 50 randomly selected initial conditions within $\mathbb{F}^-$. (*Bottom*) An exemplar ENMPC control history. Note that the control actions are constrained within the specified bounds.

the sampled nodes, irrespective of how the control actions were computed. Therefore, the proposed method offers a degree of flexibility in controller design. Of course, a trade-off of using such flexible cost functions would be the lack of guaranteed closed-loop stability.

## Acknowledgments

The authors would like to thank the Associate Editor and the anonymous reviewers for their valuable comments and suggestions. This research was supported partially by a National Science Foundation (NSF) CAREER Award ECCS0846572, and NSF grant DMS-0900277.

## References

[1] L. Magni, D. M. Raimondo, and F. Allgöwer, *Nonlinear Model Predictive Control: Towards new challenging applications*. Springer, 2009, vol. 384.

[2] A. Szucs, M. Kvasnica, and M. Fikar, “A memory-efficient representation of explicit MPC solutions,” in *Decision and Control and European Control Conference (CDC-ECC), 2011 50th IEEE Conference on*, Dec 2011, pp. 1916–1921.

[3] A. Alessio and A. Bemporad, “A survey on explicit model predictive control,” in *Nonlinear Model Predictive Control*, ser. Lecture Notes in Control and Information Sciences, L. Magni, D. M. Raimondo, and F. Allgöwer, Eds. Springer Berlin Heidelberg, 2009, vol. 384, pp. 345–369.

[4] A. Bemporad, M. Morari, V. Dua, and E. N. Pistikopoulos, “The explicit linear quadratic regulator for constrained systems,” *Automatica*, vol. 38, no. 1, pp. 3–20, 2002.

[5] ——, “The explicit solution of model predictive control via multiparametric quadratic programming,” in *Proc. of the American Control Conference.*, vol. 2, 2000, pp. 872–876.

[6] D. Limon, T. Alamo, D. Raimondo, D. M. de la Pena, J. M. Bravo, A. Ferramosca, and E. F. Camacho, “Input-to-state stability: a unifying framework for robust model predictive control,” in *Nonlinear model predictive control*. Springer, 2009, pp. 1–26.

[7] E. N. Pistikopoulos, “Perspectives in multiparametric programming and explicit model predictive control,” *AIChE Journal*, vol. 55, no. 8, pp. 1918–1925, 2009.

[8] S. Summers, C. N. Jones, J. Lygeros, and M. Morari, “A multiresolution approximation method for fast explicit model predictive control,” *IEEE Transactions on Automatic Control*, vol. 56, no. 11, pp. 2530–2541, 2011.

[9] A. Domahidi, M. N. Zeilinger, M. Morari, and C. N. Jones, “Learning a feasible and stabilizing explicit model predictive control law by robust optimization,” in *50th IEEE Conference on Decision and Control and European Control Conference (CDC-ECC)*. IEEE, 2011, pp. 513–519.

[10] D. M. Raimondo, S. Riverso, S. Summers, C. N. Jones, J. Lygeros, and M. Morari, “A set-theoretic method for verifying feasibility of a fast explicit nonlinear model predictive controller,” in *Distributed Decision Making and Control*. Springer, 2012, pp. 289–311.

[11] D. M. Raimondo, O. Huber, M. Schulze Darup, M. Monnigmann, and M. Morari, “Constrained time-optimal control for nonlinear systems: a fast explicit approximation,” *Nonlinear Model Predictive Control*, vol. 4, no. 1, pp. 113–118, 2012.

[12] J. Oravec, S. Blazek, M. Kvasnica, and S. Di Cairano, “Polygonic representation of explicit model predictive control,” in *Proc. IEEE 52nd Annual Conference on Decision and Control (CDC)*. IEEE, 2013, pp. 6422–6427.

[13] D. Axehill, T. Besselmann, D. M. Raimondo, and M. Morari, “A parametric branch and bound approach to suboptimal explicit hybrid mpc,” *Automatica*, vol. 50, no. 1, pp. 240–246, 2014.

[14] M. Rubagotti, D. Barcelli, and A. Bemporad, “Robust explicit model predictive control via regular piecewise-affine approximation,” *International Journal of Control*, vol. 87, no. 12, pp. 2583–2593, 2014.

[15] W. H. Chen, D. J. Ballance, and P. J. Gawthrop, “Optimal control of nonlinear systems: a predictive control approach,” *Automatica*, vol. 39, no. 4, pp. 633–641, 2003.

[16] T. A. Johansen, “Approximate explicit receding horizon control of constrained nonlinear systems,” *Automatica*, vol. 40, no. 2, pp. 293–300, 2004.

[17] A. Grancharova and T. Johansen, “Computation, approximation and stability of explicit feedback min–max nonlinear model predictive control,” *Automatica*, vol. 45, no. 5, pp. 1134–1143, 2009.

[18] A. Bemporad and C. Filippi, “Suboptimal explicit receding horizon control via approximate multiparametric quadratic programming,” *Journal of optimization theory and applications*, vol. 117, no. 1, pp. 9–38, 2003.

[19] G. Pin, M. Filippo, A. Pellegrino, and T. Parisini, “Approximate off-line receding horizon control of constrained nonlinear discrete-time systems,” in *Proc. European Control Conference*, 2009, pp. 2420–2431.

[20] G. Pin, M. Filippo, F. A. Pellegrino, G. Fenu, and T. Parisini, “Approximate model predictive control laws for constrained nonlinear discrete-time systems: analysis and offline design,” *Int. J. Control*, vol. 86, no. 5, pp. 804–820, 2013.

[21] Y. Wang and S. Boyd, “Fast model predictive control using online optimization,” *IEEE Transactions on Control Systems Technology*, vol. 18, no. 2, pp. 267–278, 2010.

[22] T. Parisini and R. Zoppoli, “A receding-horizon regulator for nonlinear systems and a neural approximation,” *Automatica*, vol. 31, no. 10, pp. 1443–1451, 1995.

[23] T. Parisini, M. Sanguineti, and R. Zoppoli, “Nonlinear stabilization by receding-horizon neural regulators,” *Int. J. Control*, vol. 70, no. 3, pp. 341–362, 1998.

[24] B. M. Akesson and H. T. Toivonen, “A neural network model predictive controller,” *Journal of Process Control*, vol. 16, no. 9, pp. 937–946, 2006.

[25] Q. Miao and S. F. Wang, “Nonlinear model predictive control based on support vector regression,” in *International Conference on Machine Learning and Cybernetics*, vol. 3, 2002, pp. 1657–1661.

[26] A. Aswani, H. Gonzalez, S. S. Sastry, and C. J. Tomlin, “Provably safe and robust learning-based model predictive control,” *Automatica*, 2013.

[27] C. J. Ong, D. Sui, and E. G. Gilbert, “Enlarging the terminal region of nonlinear model predictive control using the support vector machine method,” *Automatica*, vol. 42, no. 6, pp. 1011–1016, 2006.

[28] C. Cortes and V. Vapnik, “Support-vector networks,” *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995.

[29] V. Vapnik, *The nature of statistical learning theory*. Springer, 2000.

[30] S. Dumais and H. Chen, “Hierarchical classification of web content,” in *Proc. 23rd Annual International ACM SIGIR Conf. on Research and Development in Information Retrieval*. ACM, 2000, pp. 256–263.

[31] R. Lambert, “Approximation methodologies for explicit model predictive control of complex systems,” Ph.D. dissertation, Imperial College London, 2014.

[32] I. Steinwart, “Consistency of support vector machines and other regularized kernel classifiers,” *IEEE Transactions on Information Theory*, vol. 51, no. 1, pp. 128–142, 2005.

[33] A. Chakrabarty, V. Dinh, G. T. Buzzard, S. H. Żak, and A. E. Rundell, “Robust explicit nonlinear model predictive control with integral sliding mode,” in *Proc. IEEE American Control Conference (ACC)*. IEEE, 2014.

[34] H. Chen and F. Allgöwer, "A quasi-infinite horizon nonlinear model predictive control scheme with guaranteed stability," *Automatica*, vol. 34, no. 10, pp. 1205–1217, 1998.

[35] C. J. C. Burges, "A tutorial on support vector machines for pattern recognition," *Data Mining and Knowledge Discovery*, vol. 2, no. 2, pp. 121–167, 1998.

[36] M. Schulze Darup and M. Monnigmann, "Low complexity suboptimal explicit nmpc," in *Nonlinear Model Predictive Control*, vol. 4, no. 1, 2012, pp. 406–411.

[37] G. Deffuant, L. Chapel, and S. Martin, "Approximating viability kernels with support vector machines," *IEEE Transactions on Automatic Control*, vol. 52, no. 5, pp. 933–937, 2007.

[38] V. Dinh, A. E. Rundell, and G. T. Buzzard, "Effective sampling schemes for behavior discrimination in nonlinear systems," *International Journal for Uncertainty Quantification*, vol. 4, no. 6, 2014.

[39] I. Steinwart, "On the influence of the kernel on the consistency of support vector machines," *Journal of Machine Learning Research*, vol. 2, pp. 67–93, 2001.

[40] E. S. Meadows, M. A. Henson, J. W. Eaton, and J. Rawlings, "Receding horizon control and discontinuous state feedback stabilization," *International Journal of Control*, vol. 62, no. 5, pp. 1217–1229, 1995.

[41] J. D. Jakeman, R. Archibald, and D. Xiu, "Characterization of discontinuities in high-dimensional stochastic problems on adaptive sparse grids," *Journal of Computational Physics*, vol. 230, no. 10, pp. 3977–3997, 2011.

[42] J. D. Jakeman, A. Narayan, and D. Xiu, "Minimal multi-element stochastic collocation for uncertainty quantification of discontinuous functions," *Journal of Computational Physics*, vol. 242, pp. 790–808, 2013.

[43] V. Barthelmann, E. Novak, and K. Ritter, "High dimensional polynomial interpolation on sparse grids," *Advances in Computational Mathematics*, vol. 12, no. 4, pp. 273–288, 2000.

[44] S. Smolyak, "Quadrature and interpolation formulas for tensor products of certain classes of functions," in *Dokl. Akad. Nauk SSSR*, vol. 4, 1963, pp. 240–243.

[45] G. T. Buzzard, "Efficient basis change for sparse-grid interpolating polynomials with application to T-cell sensitivity analysis," *Computational Biology Journal*, 2013.

[46] J. Löfberg, "YALMIP: A toolbox for modeling and optimization in MATLAB," in *2004 IEEE International Symposium on Computer Aided Control Systems Design*. IEEE, 2004, pp. 284–289.

[47] R. Oldenhuis and J. Vandekerckhove, "GODLIKE—a robust single- and multi-objective optimizer," *MATLAB Central, MathWorks, Ismaning, Germany*, 2009.

[48] C.-C. Chang and C.-J. Lin, "LIBSVM: a library for support vector machines," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 2, no. 3, p. 27, 2011.

[49] A. Klimke, "Sparse grid interpolation toolbox," *Universität Stuttgart*, 2008.

[50] N. Nauryzbayev and N. Temirgaliyev, "An exact order of discrepancy of the smolyak grid and some general conclusions in the theory of numerical integration," *Foundations of Computational Mathematics*, vol. 12, pp. 139–172, 2012.

[51] A. Klimke and B. Wohlmuth, "Algorithm 847: Spinterp: piecewise multilinear hierarchical sparse grid interpolation in MATLAB," *ACM Transactions on Mathematical Software (TOMS)*, vol. 31, no. 4, pp. 561–579, 2005.

[52] S. Summers, C. N. Jones, J. Lygeros, and M. Morari, "A multiscale approximation scheme for explicit model predictive control with stability, feasibility, and performance guarantees," in *Proc. of the Conference on Decision and Control, 2009 held jointly with the 2009 28th Chinese Control Conference.*, 2009, pp. 6327–6332.

[53] M. S. Darup and M. Monnigmann, "Approximate explicit NMPC with guaranteed stability ensured by a simple auxiliary controller," in *Proc. IEEE International Symposium on Intelligent Control (ISIC)*, 2012, pp. 270–275.

[54] D. Q. Mayne and H. Michalska, "Receding horizon control of nonlinear systems," *IEEE Trans. on Automatic Control*, vol. 35, no. 7, pp. 814–824, 1990.

[55] W. H. Chen, J. O'Reilly, and D. J. Ballance, "On the terminal region of model predictive control for non-linear systems with input/state constraints," *Int. J. of adaptive control and signal processing*, vol. 17, no. 3, pp. 195–207, 2003.

[56] H. Whitney, "Analytic extensions of differentiable functions defined in closed sets," *Transactions of the American Mathematical Society*, vol. 36, no. 1, pp. 63–89, 1934.

[57] J. C. Mason and D. C. Handscomb, *Chebyshev polynomials*. CRC Press, 2002.

[58] R. Bellman, "The stability of solutions of linear differential equations," *Duke math. J*, vol. 10, no. 4, pp. 643–647, 1943.

[59] H. H. Bauschke and P. L. Combettes, *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*. Springer Science & Business Media, 2011.

## Appendix

### Pseudo-Code for Proposed ENMPC Design

### Proof of Theorem 1

Recall that $\mathcal{S}$ is the set of points on the separating manifold defined in (8). From Proposition 1, there exists some

$$\mu = \max\left(d(F^+,\mathcal{S}), d(F^-,\mathcal{S})\right),$$

where $d$ denotes a distance metric. Let $m \triangleq \left\lfloor 2/L(\mu^*)^2 \right\rfloor$, where $\lfloor\cdot\rfloor$ is the floor function, $\mu^*$ is the separating margin corresponding to $\boldsymbol{\beta}_N^*$, and $L$ is the regularization parameter in the SVM optimization problem (5). We construct open covers $\mathcal{P}_0^+$ of $F^+$ and $\mathcal{P}_0^-$ of $F^-$ such that every set $A \in \mathcal{P}_0^+ \cup \mathcal{P}_0^-$ has diameter $\rho(A) = \mu^*/2$ and is a hypercube aligned with the co-ordinate axes.

We claim that for $N$ sufficiently large, there are at least $m$ samples out of $N$ in each set $A \in \mathcal{P}_0^+ \cup \mathcal{P}_0^-$. To see this, note that by definition of the discrepancy of a low-discrepancy sequence,

$$\sup_{X\in J}\left|\frac{\#X_N}{N} - \frac{\mathrm{Vol}(X)}{\mathrm{Vol}(\mathbb{X})}\right| = D_N(\{\boldsymbol{x}_j\}_{j=1}^N).$$

Hence, for any $A \in \mathcal{P}_0^+ \cup \mathcal{P}_0^-$, we get

$$\left|\frac{\#A_N}{N} - \frac{\mathrm{Vol}(A)}{\mathrm{Vol}(\mathbb{X})}\right| \le D_N(\{\boldsymbol{x}_j\}_{j=1}^N)$$

and hence,

$$\frac{\mathrm{Vol}(A)}{\mathrm{Vol}(\mathbb{X})} - \frac{\#A_N}{N} \le D_N(\{\boldsymbol{x}_j\}_{j=1}^N).$$

Solving for $\#A_N$ we get

$$\#A_N \ge N\left(\frac{\mathrm{Vol}(A)}{\mathrm{Vol}(\mathbb{X})} - D_N(\{\boldsymbol{x}_j\}_{j=1}^N)\right).$$

Since $\{\boldsymbol{x}_j\}_{j=1}^N$ is a low discrepancy sequence, $D_N \to 0$, and hence $\#A_N \to \infty$ as $N \to \infty$. Therefore, we can choose $N_0$ large enough to satisfy $\#A_N \ge m$ for all $N \ge N_0$ and all $A \in \mathcal{P}_0^+ \cup \mathcal{P}_0^-$.

It remains to show that the SVM decision function $\psi(\boldsymbol{x}, \boldsymbol{\beta}_N^*)$ classifies

$$\bigcup_{A\in\mathcal{P}_0^+} A \quad\text{and}\quad \bigcup_{A\in\mathcal{P}_0^-} A$$

correctly. This can be argued exactly as in the proof of [39, Theorem 18]. □

```
Algorithm 1 ENMPC Construction
Require: State and input bounds: $\mathbb{X}, \mathbb{U}$
Require: Model, number of controllers: $\boldsymbol{f}, n_u$
Require: Equilibrium pair: $\boldsymbol{x}_e, \boldsymbol{u}_e$
Require: User-defined range of samples allotted to construct
    ENMPC: $N_0^{\max}, N_0^{\min}$
Require: Number of polynomial basis elements: $M$ as described in (16)
Require: Model predictive control parameters: $\boldsymbol{Q}, \boldsymbol{R}, T_f, \delta$
 1: Compute number of sparse grid nodes: $N \leftarrow$ Sparse-grid depth, max points, min points
Learning the feasible region boundary, $\partial\mathbb{F}$
 2: Compute terminal region $\mathbb{X}_T$
 3: $\{\boldsymbol{x}_i\}_{i=1}^N \leftarrow$ Extract $N$ nodes from a low-discrepancy sequence
 4: for $i$=1:$N$ do
 5:     Solve the terminal region based NMPC problem (3) with $\boldsymbol{x}(0) = \boldsymbol{x}^i$ for each $\boldsymbol{x}_i \in \{\boldsymbol{x}_i\}_{i=1}^N$
 6:     if Problem (3) has a feasible solution then
 7:         Store as feasible data sample $y_i \leftarrow +1$
 8:     else
 9:         Store as infeasible data sample $y_i \leftarrow -1$
10:     end if
11:     Store optimal $\boldsymbol{u}^*(\boldsymbol{x})$ for ENMPC surface construction
12: end for
13: Use SVM to compute decision function $\psi(\boldsymbol{x}, \boldsymbol{\beta}_N^*)$ as in (7)
14: $\partial\mathbb{F} \leftarrow \{\boldsymbol{x} : \psi(\boldsymbol{x}, \boldsymbol{\beta}_N^*) = 0\}$
15: Strict inner approximation: $\mathbb{F}^+ \leftarrow$ Solve (9)
16: Strict outer approximation: $\mathbb{F}^- \leftarrow$ Solve (10)
Constructing ENMPC control surface, $\boldsymbol{u}(\boldsymbol{x})$
17: for $j$=1:$n_u$ do
18:     Check number of data points $\hat{N}$ available within $\mathbb{F}$
19:     if $\hat{N} < N_0^{\min}$ or $\hat{N} > N_0^{\max}$ then
20:         Return to step 3 and sample densely from within current estimate of $\mathbb{F}$ until $\hat{N} \in [N_0^{\min}, N_0^{\max}]$ from a low-discrepancy sequence
21:     end if
22:     $N \leftarrow \hat{N}$
23:     if $\mathbb{F}$ is a regular domain then
24:         Construct $j$th control $\boldsymbol{u}^j(\boldsymbol{x})$ from $\{\boldsymbol{x}_i\}_{i=1}^N$ using sparse-grid interpolation (see Section V-A)
25:     else
26:         Construct $j$th control $\boldsymbol{u}^j(\boldsymbol{x})$ from $\{\boldsymbol{x}_i\}_{i=1}^N$ using regression with Chebyshev/Legendre/Hermite polynomials (see Section V-B)
27:     end if
28: end for
```

## Proof of Theorem 3

In order to provide guarantees on the approximation performance using multivariate Chebyshev bases, without loss of generality, we consider $\mathbb{X} = [-1, 1]^{n_x}$. By the Whitney Extension Theorem (see [56]), there exists $\boldsymbol{u} \in \mathcal{C}^1(\mathbb{X})$ such that $\boldsymbol{u} = \boldsymbol{u}^*$ on $\mathcal{W}_1$, where $\boldsymbol{u}^*$ is the optimal NMPC. Invoking [57, Theorem 5.10], we get that $\boldsymbol{u}$ has a multivariate Chebyshev expansion that converges uniformly on $\mathbb{X}$.

Uniform convergence implies that $\|\hat{\boldsymbol{u}}_M - \boldsymbol{u}^*\|_{\mathbb{L}^\infty(\mathcal{W}_1)} \to 0$ as $M \to \infty$, where $\hat{\boldsymbol{u}}_{\boldsymbol{c}_M}$ has the same basis expansion as described in (16).

Let $\boldsymbol{w}(\boldsymbol{x})$ denote the multivariable Chebyshev weight function that makes the Chebyshev polynomials orthonormal. Orthogonality and the Cauchy-Schwarz inequality imply

$$\begin{aligned} |c_j| &= \left| \int_{\mathbb{X}} \boldsymbol{u}(x)\mathbb{T}_j(\boldsymbol{x})\boldsymbol{w}(\boldsymbol{x})\, d\boldsymbol{x} \right|, \\ &\leq \|\boldsymbol{u}\|_{\mathbb{L}^2(\mathbb{X},\boldsymbol{w})} \|\mathbb{T}_j\|_{\mathbb{L}^2(\mathbb{X},\boldsymbol{w})}, \\ &\leq \|\boldsymbol{u}\|_{\mathbb{L}^2(\mathbb{X},\boldsymbol{w})}. \end{aligned}$$

Defining $\bar{c}_0 \triangleq \|\boldsymbol{u}\|_{\mathbb{L}^2(\mathbb{X},\boldsymbol{w})}$, we conclude $\|\boldsymbol{c}_M\|_\infty \leq \bar{c}_0$ is independent of $M$. Note that for $\bar{c} \geq \bar{c}_0$ and $M$ fixed, the set

$$\Xi_M(\bar{c}) = \{\hat{\boldsymbol{u}}_M : \|\boldsymbol{c}\|_\infty \leq \bar{c}\}$$

is a compact set of polynomials, hence is equicontinuous on the hypercube $\mathbb{X}$. With $\bar{c}_0$ chosen above, let $\varepsilon > 0$ and $\bar{c} \geq \bar{c}_0$. Note that for such $\bar{c}$, the vector $\boldsymbol{c}$ is a feasible solution to the minimization problem (17). Hence, for any $j = 1, \ldots, N$, with $\boldsymbol{x}_j \in \mathcal{W}_1$, we have

$$\|\hat{\boldsymbol{u}}_{N,M}(\boldsymbol{x}_j) - \boldsymbol{u}^*(\boldsymbol{x}_j)\| \leq \|\hat{\boldsymbol{u}}_M - \boldsymbol{u}^*\|_{\mathbb{L}^\infty(\mathcal{W}_1)} \triangleq \varepsilon_M. \tag{24}$$

Since $\varepsilon_M \to 0$ as $M \to \infty$, there exists an $M_0 \in \mathbb{N}$ such that if $M \geq M_0$, then $\varepsilon_M < \varepsilon/3$. Note that for any $\boldsymbol{y} \in \mathcal{W}_1$ and any $k \in \mathbb{N}$ such that $\boldsymbol{x}_k \in \mathcal{W}_1$, the triangle inequality yields

$$\begin{aligned} \|\hat{\boldsymbol{u}}_{N,M}^*(\boldsymbol{y}) - \boldsymbol{u}^*(\boldsymbol{y})\| \leq\ & \|\hat{\boldsymbol{u}}_{N,M}^*(\boldsymbol{y}) - \hat{\boldsymbol{u}}_{N,M}^*(\boldsymbol{x}_k)\| \\ &+ \|\hat{\boldsymbol{u}}_{N,M}^*(\boldsymbol{x}_k) - \boldsymbol{u}^*(\boldsymbol{x}_k)\| \\ &+ \|\boldsymbol{u}^*(\boldsymbol{x}_k) - \boldsymbol{u}^*(\boldsymbol{y})\|. \end{aligned} \tag{25}$$

By equicontinuity of $\Xi_M(\bar{c}) \cup \{\boldsymbol{u}\}$, there exists some $\delta_0 > 0$ such that for all $\boldsymbol{y}_1, \boldsymbol{y}_2 \in \mathbb{X}$ with $\|\boldsymbol{y}_1 - \boldsymbol{y}_2\| < \delta_0$ we have $\|\hat{\boldsymbol{u}}(\boldsymbol{y}_1) - \hat{\boldsymbol{u}}(\boldsymbol{y}_2)\| < \varepsilon/3$ for any $\hat{\boldsymbol{u}} \in \Xi_{M,\bar{c}} \cup \{\boldsymbol{u}\}$. Without loss of generality, we reduce $\delta_0$ so that $\delta_0 < d(\mathcal{W}_0, \mathcal{W}_1^c)$.

Now, we take a finite cover of $\mathcal{W}_0$ by balls of the form $\mathbb{B}_{\delta_0}(\boldsymbol{y})$ for $\boldsymbol{y} \in \mathcal{W}_0$. The fact that we select a low-discrepancy sequence to sample $\mathcal{W}_0$ implies that for any $\boldsymbol{y} \in \mathcal{W}_0$, there is a point $\boldsymbol{x}_k \in \mathbb{B}_{\delta_0}(y)$. Choose $N_0$ to be the maximum index of the corresponding points $\boldsymbol{x}_k$ in the finitely many balls covering $\mathcal{W}_0$.

Using the inequality (24), and noting that for each $\boldsymbol{y} \in \mathcal{W}_0$ there is a point $\boldsymbol{x}_k$ with $k \leq N_0$ and $\|\boldsymbol{y} - \boldsymbol{x}_k\| \leq \delta_0$, each term of (25) is less than $\varepsilon/3$. Since this is true for any $\boldsymbol{y} \in \mathcal{W}_0$, we are done. □

## Proof of Theorem 4

We divide our proof into two parts. First, we show that for sufficiently close approximation of the NMPC $\boldsymbol{u}^*$ by the ENMPC the image of the compact set $\mathcal{W}_0$ is contained in the terminal region $\mathbb{X}_T$ within the predictive horizon $T_f$. Second, we demonstrate that the image of the terminal region under the ENMPC is invariant after time $T_f$ for sufficiently close approximation of the NMPC control law.

We now proceed with the first part. Consider the closed-loop system $\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u})$, where $\boldsymbol{u}$ is the optimal NMPC.

The ENMPC (constructed with $N$ samples) controlled closed loop system is $\dot{\hat{\boldsymbol{x}}} = \boldsymbol{f}(\hat{\boldsymbol{x}}, \hat{\boldsymbol{u}})$, where $\hat{\boldsymbol{u}} = \hat{\boldsymbol{u}}^*_{N,M}$.

We want to show that for every $\varepsilon > 0$, there exists an $\varepsilon' > 0$ such that $\|\boldsymbol{u} - \hat{\boldsymbol{u}}\| \le \varepsilon'$ implies that

$$\|\boldsymbol{x}(t) - \hat{\boldsymbol{x}}(t)\| \le \varepsilon, \tag{26}$$

for all $t \in [t_0, T_f]$ with identical initial conditions $\boldsymbol{x}(t_0) = \hat{\boldsymbol{x}}(t_0) \in \mathcal{W}_0$.

We know

$$\boldsymbol{x}(t) - \hat{\boldsymbol{x}}(t) = \int_{t_0}^{t} \boldsymbol{f}(\boldsymbol{x}(s), \boldsymbol{u}(\boldsymbol{x}(s))) - \boldsymbol{f}(\hat{\boldsymbol{x}}(s), \hat{\boldsymbol{u}}(\boldsymbol{x}(s)))\, ds.$$

Let $\Delta\boldsymbol{x} = \boldsymbol{x}(t) - \hat{\boldsymbol{x}}(t)$. This implies

$$\|\Delta\boldsymbol{x}\| \le \int_{t_0}^{t} \|\boldsymbol{f}(\boldsymbol{x}(s), \boldsymbol{u}(\boldsymbol{x}(s))) - \boldsymbol{f}(\hat{\boldsymbol{x}}(s), \hat{\boldsymbol{u}}(\boldsymbol{x}(s)))\|\, ds. \tag{27}$$

Since $\boldsymbol{u} \in \mathcal{C}^1$ on $\mathcal{W}$, it is locally Lipschitz on $\mathcal{W}$. We refer to the Lipschitz constant as $\mathfrak{L}_u$, with respect to $\boldsymbol{x}$. Using the Lipschitz continuity property of $\boldsymbol{f}$ and $\boldsymbol{u}$

$$\begin{aligned} &\|\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}(\boldsymbol{x})) - \boldsymbol{f}(\hat{\boldsymbol{x}}, \hat{\boldsymbol{u}}(\hat{\boldsymbol{x}}))\| \\ &\le \|\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}(\boldsymbol{x})) - \boldsymbol{f}(\hat{\boldsymbol{x}}, \boldsymbol{u}(\boldsymbol{x}))\| + \|\boldsymbol{f}(\hat{\boldsymbol{x}}, \boldsymbol{u}(\boldsymbol{x})) - \boldsymbol{f}(\hat{\boldsymbol{x}}, \hat{\boldsymbol{u}}(\hat{\boldsymbol{x}}))\| \\ &\le \mathfrak{L}^x_f \|\Delta\boldsymbol{x}\| + \mathfrak{L}^u_f \|\boldsymbol{u}(\boldsymbol{x}) - \hat{\boldsymbol{u}}(\hat{\boldsymbol{x}})\| \\ &\le \mathfrak{L}^x_f \|\Delta\boldsymbol{x}\| + \mathfrak{L}^u_f \|\boldsymbol{u}(\boldsymbol{x}) - \boldsymbol{u}(\hat{\boldsymbol{x}})\| + \mathfrak{L}^u_f \|\boldsymbol{u}(\hat{\boldsymbol{x}}) - \hat{\boldsymbol{u}}(\hat{\boldsymbol{x}})\| \\ &\le \mathfrak{L}^x_f \|\Delta\boldsymbol{x}\| + \mathfrak{L}^u_f \mathfrak{L}_u \|\Delta\boldsymbol{x}\| + \mathfrak{L}^u_f \varepsilon'. \end{aligned} \tag{28}$$

For a fixed scalar $\varepsilon'$, the last inequality follows from Theorem 3 for sufficiently large $N_1 \in \mathbb{N}$ for the regression based ENMPC and from Theorem 2 for the sparse-grid interpolated ENMPC, respectively.

Thus, we conclude

$$\|\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}(\boldsymbol{x})) - \boldsymbol{f}(\hat{\boldsymbol{x}}, \hat{\boldsymbol{u}}(\hat{\boldsymbol{x}}))\| \le \rho_0 \varepsilon' + \rho_1 \|\Delta\boldsymbol{x}\| \tag{29}$$

for positive scalars $\rho_0 = \mathfrak{L}^u_f$ and $\rho_1 = \mathfrak{L}^x_f + \mathfrak{L}^u_f \mathfrak{L}_u$.

Using (27) and (29), we can write

$$\begin{aligned} \|\Delta\boldsymbol{x}(t)\| &\le \int_{t_0}^{t} (\rho_0 \varepsilon' + \rho_1 \|\Delta\boldsymbol{x}(s)\|)\; ds \\ &= \rho_0 \varepsilon'(t - t_0) + \rho_1 \int_{t_0}^{t} \|\Delta\boldsymbol{x}(s)\|\, ds. \end{aligned}$$

Employing the integral form of Grönwall's Lemma [58], we have

$$\begin{aligned} \|\Delta\boldsymbol{x}(t)\| &\le \rho_0 \varepsilon'(t_0 - t_0) + \rho_0 \varepsilon' \int_{t_0}^{t} \left( \exp\left( \int_{s}^{t} \rho_1\, dr \right) \right) ds \\ &= \rho_0 \varepsilon' \exp(\rho_1 t) \int_{t_0}^{t} \exp(-\rho_1 s)\, ds. \end{aligned}$$

Since this integral is bounded above for $t \in [t_0, T_f]$, we can choose $\varepsilon'$ small enough to get the desired inequality (26).

Let $\mathcal{X}(\mathcal{W}_0, [t_0, t])$ map any state $\boldsymbol{x}(t_0) \in \mathcal{W}_0$ to $\boldsymbol{x}(t)$ obtained by solving the closed-loop system (1) with $\boldsymbol{u}$ being the optimal NMPC controller. Since the map $\mathcal{X}$ is continuous, the set $\mathcal{W}_1 \triangleq \mathcal{X}(\mathcal{W}_0, [0, T_f]) \cup \mathbb{X}_T$ is compact and contained in $\mathcal{W} \subset \mathbb{F}$. By construction of $\mathcal{W}_1$, if $\boldsymbol{x}(t_0) \in \mathcal{W}_0$ this implies $\boldsymbol{x}(t) \in \mathcal{W}_1$ with the optimal NMPC action for all $t \ge t_0$. By compactness of $\mathcal{W}_1$, there exists some $\varepsilon > 0$ and compact set $\mathcal{W}_2 \subset \mathcal{W}$ such that $d(\mathcal{W}_1, \mathcal{W}_2^c) > \varepsilon$. If our ENMPC generated state trajectory $\hat{\boldsymbol{x}}(t)$ satisfies $\|\boldsymbol{x}(t) - \hat{\boldsymbol{x}}(t)\| \le \varepsilon$, then $\hat{\boldsymbol{x}}(t) \in \mathcal{W}_2$. This is satisfied by choosing $N$ large enough to ensure $\|\boldsymbol{u} - \hat{\boldsymbol{u}}\| \le \varepsilon'$ on $\mathcal{W}_2$, as discussed in (26). As $\mathcal{W}_2 \subset \mathcal{W} \subset \mathbb{F}$, the state trajectory of the closed-loop system with initial condition $\boldsymbol{x}(t_0)$ using ENMPC control is feasible for all $t \ge t_0$.

Now, we prove the second part. By definition of the feasible set, the trajectory $\boldsymbol{x}(t)$ enters $\mathbb{X}_T$ within $T_f$. Thus, for $t \ge T_f$, the image of $\mathcal{W}_0$ under NMPC control is a compact set, say $\mathcal{W}_T$, contained in the open set $\mathbb{X}_T$. This implies that there is a positive distance between $\mathcal{W}_T$ and the complement of $\mathbb{X}_T$, that is $d(\mathcal{W}_T, \mathbb{X}_T^c) > \varepsilon_T$. Hence, by the same arguments as above, we can choose $N_2 \in \mathbb{N}$ large enough so that the image of $\mathcal{W}_0$ under ENMPC control is within $\varepsilon_T$ of $\mathcal{W}_T$, hence contained inside $\mathbb{X}_T$, hence contained inside $\mathcal{W}_0$ by assumption. By selecting $N_0 = \max(N_1, N_2)$, we ensure that $\hat{\boldsymbol{x}}(t) \in \mathcal{W}_0$ for all $t \ge T_f$. Combining this with the feasibility for $t \in [0, T_f]$ we conclude the proof of feasibility.

For proving stability, it follows from the uniform convergence of the ENMPC to the optimal NMPC that as $N \to \infty$, $\hat{\boldsymbol{u}}_N \to \boldsymbol{u}$ on $\mathcal{W}$. This implies that the nominal stability guarantees of the terminal region based NMPC proposed in [34] extends to the ENMPC constructed in this paper.

To prove that we satisfy input constraints, we note that the projection to a convex set is non-expansive on Hilbert spaces [59, Proposition 4.8]. Therefore, selecting $N, M$ large enough to ensure $\|\hat{\boldsymbol{u}}^*_{N,M} - \boldsymbol{u}\| \le \varepsilon'$ implies $\|\mathrm{Proj}_{\mathbb{U}}(\hat{\boldsymbol{u}}^*_{N,M}) - \boldsymbol{u}\| \le \varepsilon'$. Thus, the ENMPC asymptotically stabilizes the plant (1) without violating state or input constraints. This concludes the proof. □

## Proof of Corollary 1

Let $\nu = \varepsilon/2$. We obtain $F^+ = \{\boldsymbol{x} \in \mathbb{X} : \zeta(\boldsymbol{x}) \ge \varepsilon\} \subset \{\boldsymbol{x} \in \mathbb{X} : \zeta(\boldsymbol{x}) \ge \nu\}$, and $F^- = \{\boldsymbol{x} \in \mathbb{X} : \zeta(\boldsymbol{x}) \le 0\} \subset \{\boldsymbol{x} \in \mathbb{X} : \zeta(\boldsymbol{x}) \le \nu\}$. Note that $F^-$ is the set of infeasible points. Applying Theorem 1, we obtain a classifier of the form (12) which separates $F^+$ and $F^-$. This concludes the proof. □

## Proof of Corollary 2

We define $\Delta\boldsymbol{x} = \boldsymbol{x} - \hat{\boldsymbol{x}}$. We will show that for every $\varepsilon$ there is a sufficiently small $\delta$ such that $\|\boldsymbol{u}(\boldsymbol{x}) - \hat{\boldsymbol{u}}^{\delta}(\hat{\boldsymbol{x}})\|$ is arbitrarily small. To this end, as in (27), we have

$$\begin{aligned} \|\boldsymbol{u}(\boldsymbol{x}) - \hat{\boldsymbol{u}}^{\delta}(\hat{\boldsymbol{x}})\| &\le \|\boldsymbol{u}(\boldsymbol{x}) - \hat{\boldsymbol{u}}(\hat{\boldsymbol{x}})\| + \|\hat{\boldsymbol{u}}(\hat{\boldsymbol{x}}) - \hat{\boldsymbol{u}}^{\delta}(\hat{\boldsymbol{x}})\| \\ &\le \mathfrak{L}^u_f \varepsilon' + \mathfrak{L}_u \|\Delta\boldsymbol{x}\| + \|\hat{\boldsymbol{u}}(\hat{\boldsymbol{x}}) - \hat{\boldsymbol{u}}^{\delta}(\hat{\boldsymbol{x}})\| \\ &= \mathfrak{L}^u_f \varepsilon' + \mathfrak{L}_u \|\Delta\boldsymbol{x}\| + \|\hat{\boldsymbol{u}}(\hat{\boldsymbol{x}}(t)) - \hat{\boldsymbol{u}}(\hat{\boldsymbol{x}}(t - \delta'))\| \end{aligned}$$

for some $\delta' \in [0, \delta]$.

Since $\hat{\boldsymbol{u}}$ is constructed using (differentiable) polynomial basis elements, we define $\mathfrak{L}_{\hat{u}}$ as its Lipschitz constant with respect to $\boldsymbol{x}$ within $\mathbb{X}$. Then,

$$\|\boldsymbol{u}(\boldsymbol{x}) - \hat{\boldsymbol{u}}^{\delta}(\hat{\boldsymbol{x}})\| \le \mathfrak{L}^u_f \varepsilon' + \mathfrak{L}_u \|\Delta\boldsymbol{x}\| + \mathfrak{L}_{\hat{u}} \|\hat{\boldsymbol{x}}(t) - \hat{\boldsymbol{x}}(t - \delta')\|.$$

Since $\boldsymbol{f}$ is continuous and bounded on $\mathbb{X} \times \mathbb{U}$, and $\dot{\hat{\boldsymbol{x}}} = \boldsymbol{f}(\hat{\boldsymbol{x}}, \hat{\boldsymbol{u}})$, the function $\|\hat{\boldsymbol{x}}(t) - \hat{\boldsymbol{x}}(t - \delta')\|$ satisfies a

Lipschitz condition uniformly in $t$. Call the Lipschitz constant $\mathfrak{L}_{\hat{x}}$. Therefore,

$$\|\boldsymbol{u}(\boldsymbol{x}) - \hat{\boldsymbol{u}}^{\delta}(\hat{\boldsymbol{x}})\| \leq \mathfrak{L}_f^u \varepsilon' + \mathfrak{L}_u \|\Delta \boldsymbol{x}\| + \mathfrak{L}_{\hat{u}} \mathfrak{L}_{\hat{x}} \delta'. \tag{30}$$

The rest of the proof is similar to that of the proof of Theorem 4. We can see this by substituting (30) into inequality (28), which yields, $\|\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}(\boldsymbol{x})) - \boldsymbol{f}(\hat{\boldsymbol{x}}, \hat{\boldsymbol{u}}^{\delta}(\hat{\boldsymbol{x}}))\| \leq \rho_0 \varepsilon' + \rho_1 \|\Delta \boldsymbol{x}\|$ for some positive scalars $\rho_0 = \mathfrak{L}_f^u + \mathfrak{L}_{\hat{u}} \mathfrak{L}_{\hat{x}} \delta' / \varepsilon'$ and $\rho_1 = \mathfrak{L}_f^x + \mathfrak{L}_f^u \mathfrak{L}_u$. Hence, the result of Theorem 4 immediately applies to the piecewise constant ENMPC $\hat{\boldsymbol{u}}^{\delta}$. □

**Ankush Chakrabarty** earned his Bachelor degree in Electrical Engineering from Jadavpur University, Kolkata, India with first-class honors. He is currently a Ph. D. candidate at the School of Electrical and Computer Engineering at Purdue University, West Lafayette, IN. He is interested in nonlinear control, unknown input observers, biomedical control, and machine learning.

**Vu Dinh** is currently a post-doctoral researcher at the Matsen Group in the Program in Computational Biology at Fred Hutchinson Cancer Research Center. He has a special interest in evolution and phylogenetics, with a focus on studying viral evolution and the immune system. Prior to coming to the Hutch, Vu earned a Bachelor degree from the University of Science, Ho Chi Minh City and a Ph. D. in Mathematics from Purdue University, West Lafayette.

**Martin J. Corless** is currently a Professor in the School of Aeronautics and Astronautics at Purdue University, West Lafayette, Indiana, USA. He is also an Adjunct Honorary Professor in the Hamilton Institute at The National University of Ireland, Maynooth, Ireland. He received a B.E. from University College Dublin, Ireland and a Ph.D. from the University of California at Berkeley; both degrees are in mechanical engineering. He is the recipient of a National Science Foundation Presidential Young Investigator Award. His research is concerned with obtaining tools which are useful in the robust analysis and control of systems containing significant uncertainty and in applying these results to aerospace and mechanical systems and to sensor and communication networks.

**Ann E. Rundell** received the B.S. degree in electrical engineering from the University of Pennsylvania, Philadelphia, in 1988, and the M.S. and Ph.D. degrees in electrical and computer engineering at Purdue University, West Lafayette, IN, in 1993 and 1997, respectively. She is a Professor in the Weldon School of Biomedical Engineering, Purdue University. Her research interests apply systems and control theory to control cellular and physiological processes with an emphasis on model-based experiment design. Dr. Rundell is a senior member of the IEEE, SIAM and ASEE. She has coauthored more than 30 peer reviewed articles and received the NSF CAREER Award.

**Stanisław H. Żak** (M81) received the Ph.D. degree from the Warsaw University of Technology, Warsaw, Poland, in 1977. He was an Assistant Professor with the Institute of Control and Industrial Electronics, Warsaw University of Technology, from 1977 to 1980. From 1980 to 1983, he was a Visiting Assistant Professor with the Department of Electrical Engineering, University of Minnesota, Minneapolis, MN, USA. In 1983, he joined the School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN, USA, where he is currently a Professor. He has been involved in various areas of control, optimization, fuzzy systems, and neural networks. He has co-authored *Topics in the Analysis of Linear Dynamical Systems* (Warsaw, Poland: Polish Scientific Publishers, 1984) and *An Introduction to Optimization*4th Edition (New York, NY, USA: Wiley, 2001) and has authored *Systems and Control* (London, U.K., Oxford University Press, 2003). Prof. Zak was the Associate Editor of *Dynamics and Control* and the IEEE TRANS. NEURAL NETWORKS.

**Gregery T. Buzzard** received the B.S. degree in computer science and the B.Mus. degree in violin performance in 1989, the M.S. degree in mathematics in 1991 all from Michigan State University, East Lansing, and the Ph.D. degree in mathematics in 1995 from the University of Michigan, Ann Arbor. He is a Professor and Head of the Department of Mathematics, Purdue University, West Lafayette, IN. His research interests include dynamical systems, mathematical biology, and methods for approximation and optimization. Dr. Buzzard is a member of SIAM and a winner of the Spira Teaching Award.