

Learning Robustly Stabilizing Explicit Model Predictive Controllers: A Non-Regular Sampling Approach

Cristiano Cervellera, Danilo Macciò, and Thomas Parisini

Abstract—Off-line supervised learning from data of robustly-stabilizing nonlinear explicit model predictive controllers (EMPC) is dealt with in this letter. The learning procedure relies on the construction of a suitably large set of specifically chosen sampling points of the state space in which the values of the optimal EMPC control function have to be computed. When bounding the magnitude of approximation errors is important for stability or performance specifications, regular gridding techniques are not feasible due to the curse of dimensionality arising from the structural exponential growth of the number of points with the state dimension. In this note, we consider non-regular sampling techniques – namely, i.i.d. sampling with uniform distribution, low-discrepancy sequences and lattice point sets – that offer a good covering of the state space without suffering from an unfeasible growth of the number of points, while preserving at the same time the method guarantees in terms of robustness and stability. Some theoretical properties of the proposed sampling schemes are briefly discussed, and their successful application is showcased in a practically-relevant optimal heating problem involving a 21-dimensional state space that rules out the use of regular gridding techniques.

I. INTRODUCTION

The model predictive control (MPC) scheme is the most successful multivariable control approach in industrial applications owing to its properties in terms of optimization of a complex cost functional for nonlinear and possibly large-scale systems, while simultaneously satisfying state and control constraints. The reader is referred to the vast literature on the subject (see, for instance the very recent book [1] and the well-known papers [2], [3], [4] as well as the references cited therein).

To cope with the high computational burden of an on-line optimization to compute the MPC control action at each time-stage, an approach known in the literature as *explicit MPC* (EMPC) has been devised, which shifts most of the required computation in a off-line phase (see, for instance the survey paper [5] as well as the references cited therein). More specifically, the basic idea of EMPC is the off-line construction of an explicit function μ that generates in a fast way the same control vector u_t we would obtain applying the on-line model predictive control (MPC) procedure, as a function

of the current state x_t , i.e., $u_t = \mu(x_t)$. Unfortunately, as is well known, in the general nonlinear and/or constrained case, it is impossible to determine an analytical form for such MPC control function. Thus, the implementation of the EMPC approach must rely on an approximation μ^* of the true MPC control function μ .

The procedure to obtain such approximate function can be stated in a typical context of supervised learning from data. According to this framework, we look for the approximate function within a class of parameterized models. The best element inside such class is chosen minimizing the difference with respect to the true MPC control computed over a finite set of elements of the state space (see the seminal paper [6] and the very recent book [7]). However, unlike the classic learning setting, in the EMPC framework considered here the above elements of the state space (the “patterns”, in machine learning parlance) are not given by an external source, but yet they need to be carefully selected. Indeed, the cardinality of the set of these specific elements of the state space is also a key aspect for the EMPC procedure to be applicable in realistic settings.

In [8] it was proved that the Input-to-State Stability (ISS) of the controlled system can be guaranteed (under suitable regularity conditions) when the sampling points correspond to standard regular grids with even-spaced values for each state component. Unfortunately, this kind of sampling of the state space is affected by the *curse of dimensionality*, since the number of grid points is structurally bound to grow exponentially with the dimension of the state space. Hence, the practical feasibility of approximate robustly-stabilizing EMPC is limited to very low-dimension control scenarios.

In this note, we analyze the use of *non-regular sampling* methodologies to learn approximate robustly-stabilizing EMPC schemes in high-dimensional settings that are not feasible for sampling algorithms based on regular gridding of the state space. We deal with a slightly different framework than the one presented in [9] where a specific kind of non-regular sampling technique has been used in a disturbance-free setting (see also [10] and the references therein for other learning and optimal control scenarios using non-regular sampling approaches).

More specifically, we investigate three different sampling algorithms, namely the classic independent and identically distributed (i.i.d.) random sampling with uniform distribution, low-discrepancy sequences and the method of good lattice points [11], [12]. A key common feature of these non-regular sampling schemes is that they all guarantee that the stability properties of the approximate EMPC controller are preserved, while not being subject to the structural curse of dimensionality of regular grids. In particular, we show the con-

This work has been partially supported by the European Union’s Horizon 2020 research and innovation programme under grant agreement No 739551 (KIOS CoE) and by the Italian Ministry for Research in the framework of the 2017 Program for Research Projects of National Interest (PRIN), Grant no. 2017YKXYXJ.

C. Cervellera and Danilo Macciò are with the Institute of Marine Engineering, National Research Council (CNR-INM), Italy (Email: {cristiano.cervellera, danilo.maccio}@cnr.it).

T. Parisini is with the Dept. of Electrical and Electronic Engineering at the Imperial College London, UK, with the KIOS Research and Innovation Centre of Excellence, University of Cyprus, and also with the Dept. of Engineering and Architecture at the University of Trieste, Italy (Email: t.parisini@gmail.com).

vergence guarantees taking advantage of quantities typical of number-theoretic methods and numerical integration, namely, the *dispersion* and the *discrepancy*.

In order to show the effectiveness in practice of the proposed non-regular sampling learning procedure, all the three proposed sequences are implemented for the design of an EMPC law to impose a desired temperature profile to a copper plate. The spatial discretization of partial-differential equations describing the heat transfer model leads to a 21-dimensional state space that rules out the use of regular gridding techniques. Our numerical results show that all the proposed non-regular grids yield EMPC controllers (with neural networks as learning models) able to approximate well the MPC solution, while the classic regular grid performs very poorly and cannot be improved by a more accurate sampling due to the exponential growth in the computational complexity.

II. FORMULATION OF THE LEARNING PROBLEM FOR EXPLICIT MODEL PREDICTIVE CONTROL

Consider the discrete-time nonlinear dynamical system

$$x_{t+1} = f(x_t, u_t, \vartheta_t), \quad t = 0, 1, \dots$$

where $x_t \in X \subset \mathbb{R}^n$ is the state vector, $u_t \in U \subset \mathbb{R}^m$ is the control vector and $\vartheta_t \in \Theta \subset \mathbb{R}^r$ is an exogenous disturbance input. We assume that the sets X and U are compact Euclidean subsets of \mathbb{R}^n and \mathbb{R}^m , respectively, containing the origin as an interior point.

The stabilizing control design is based on the *nominal* model

$$x_{t+1} = \hat{f}(x_t, u_t) + d_t, \quad t = 0, 1, \dots$$

where $d_t = f(x_t, u_t, \vartheta_t) - \hat{f}(x_t, u_t)$ denotes the discrete-time state transition uncertainty, where we assume that $\hat{f}(0, 0) = 0$.

Definition 1: Consider a positive integer N , a given initial state x_0 and a generic sequence of control vectors $\{u_0, \dots, u_{N-1}\}$. The N -stage cost is defined as

$$J_{\text{FH}}(x_0, u_0, \dots, u_{N-1}) = \sum_{k=0}^{N-1} h(x_k, u_k) + h_N(x_N), \quad (1)$$

where h, h_N are the stage and final stage costs, respectively, and, for every $k = 0, \dots, N-1$, $x_{k+1} = \hat{f}(x_k, u_k)$. \square

In the usual MPC framework [1], the control action at time stage t when the system is in the generic state x_t is obtained by minimizing J_{FH} with $x_0 = x_t$. Then, only the first optimal control vector u_t^* is actually applied to drive the system to the next state x_{t+1} . The EMPC problem consists in constructing the control function given in the following definition.

Definition 2: For every $x_t \in X$, consider the sequence $u_t^\circ, \dots, u_{t+N-1}^\circ$ that minimizes the cost (1). Then, the EMPC function μ° is defined as $\mu^\circ(x_t) = u_t^\circ$ for all $x_t \in X$. \square

In this note, we address the off-line construction of an empirical approximation $\mu^* \in \Gamma$ of the EMPC function given in Definition 2 by a *learning from data* approach, where Γ is a suitably “rich” class of parametrized approximating models (for instance, feedforward neural networks, kernel-based approximating models, deep neural structures, etc.). The availability of the EMPC function μ^* allows to obtain online

the approximate control action $u_t^* = \mu^*(x_t)$ for any observed state x_t , without the need to perform at each time instant the minimization of J_{FH} [6], [7].

To this end, the first step is to construct a suitable data set

$$X_L = \{x^{(1)}, \dots, x^{(L)}\}, \quad x^{(l)} \in X, \quad l = 1, \dots, L,$$

and the corresponding set of MPC control actions

$$U_L = \{u^{(1)}, \dots, u^{(L)}\},$$

where $u^{(l)} = \mu^\circ(x^{(l)})$, $x^{(l)} \in X_L$, $l = 1, \dots, L$. Next, the appropriate parametric complexity p of the functions $\mu^*(x, w) \in \Gamma$ is selected¹, where $w \in \Lambda \subset \mathbb{R}^p$ denotes the parameter vector of the parametrized function $\mu^*(x, w)$.

Considering a loss function ℓ as introduced in [7], the following supervised learning from data problem is stated.

Problem 1: Find

$$w^* = \arg \min_{w \in \Lambda} \frac{1}{L} \sum_{l=1}^L \ell[u^{(l)}, \mu^*(x^{(l)}, w)]. \quad \square$$

On the Construction of X_L

The off-line construction of the set U_L involves a significant computational load since it requires the solution of an optimization problem for every state sample $x^{(l)} \in X_L$. Indeed, the choice of the cardinality L of X_L and the selection of the specific values taken on by $x^{(l)}$ is crucial in order to ensure that the mismatch between the EMPC function μ° and its learned approximation μ^* is not too large in order to provide the required robust stability properties (see Section III). In this respect, when the dimension n of the state space takes on large values, the construction of the set X_L through a uniform gridding procedure may turn out to be unfeasible due to the well-know *curse of dimensionality*. In fact, the number of points of a regular grid with b discretization levels for each component of the state is equal to b^n , which yields an unfeasible number L of samples when n is large, even for small values of b . This motivates the use of non-regular sampling approaches to construct X_L (see Section IV).

III. THE SET X_L AND ROBUST STABILITY PROPERTIES

As mentioned at the end of Section II, the construction of the sample set X_L is key not only from a computational complexity perspective, but also to ensure that the learned EMPC control policy is equipped with suitable robust stability properties.

The following general result on the ISS robust stability properties of the closed-loop control system driven by the approximate EMPC control law $\mu^*(x^{(l)}, w)$ can be easily obtained from the specific results given in [7], [8] for a set X_L constructed by a regular gridding procedure.

Proposition 1: Assume that the following conditions are satisfied.

¹The approximation properties of the function space Γ in terms of the dimension n of the state space, the complexity p of the approximating models and the smoothness characteristics of the target function μ° is a crucial issue that is out of the scope of this letter. The reader is referred to [7] for an up-to-date comprehensive treatment of this topic.

- 1) The nominal function $\hat{f}(x, u)$ is Lipschitz continuous with respect to x , with Lipschitz constant denoted as L_x , and is uniformly continuous in u .
- 2) The exogenous disturbance ϑ_t is bounded, and the uncertainty d_t is such that $|d_t| \leq g(|\vartheta_t|)$ for all t , where g is a continuous, strictly increasing function zero-valued in 0.
- 3) The approximate MPC control $\mu^*(x, w)$ is locally Lipschitz continuous with respect to x for any w , with Lipschitz constant denoted as $L_\mu(x, w)$.

Then, a sufficient condition for system \hat{f} driven by μ^* to be ISS on X is:

$$L_\mu(x, w)\bar{q} + |\mu^*(x, w) - \mu^\circ(x)| \leq \varepsilon^*(x), \quad x \in X_L, \quad (2)$$

where \bar{q} is such that for any $y \in X$ there exists $x \in X_L$ such that $|y - x| < \bar{q}$, and $\varepsilon^*(x)$ is a suitable constant depending on L_x and the bound on d_t . \square

The first term in the left hand-side of (2) is where the choice of the discretization set X_L affects the stability. In particular, the parameter \bar{q} represents the maximum distance that is allowed between any point in X and its closest point in the set X_L , meaning that smaller values of \bar{q} correspond to higher denseness of the sampling grid.

The condition on having a small \bar{q} can be expressed in terms of the *dispersion* of the set of points X_L , a quantity commonly employed in number-theoretic and numerical integration methods, defined (see [11]) as

$$\delta^*(X_L) = \sup_{x \in X} \min_{1 \leq l \leq L} \|x - x^{(l)}\|. \quad (3)$$

In fact, for any value \bar{q} that is suitable to satisfy (2), it is sufficient for the set X_L to have $\delta^*(X_L) \leq \bar{q}$. As a straightforward consequence, the ISS stability of the system is guaranteed if we generate X_L through an algorithm that ensures the convergence to 0 of the dispersion as L grows. From now on, assume, without loss of generality, that $X = [0, 1]^n$, i.e., the closed unitary n -dimensional cube (the generic hyperrectangle X can be obtained from $[0, 1]^n$ by simple scaling).

It is easy to see that the regular uniform grid satisfies the convergence property. In fact, if we consider a discretization of each state component by b values, the dispersion is equal to half the diagonal of the elementary cube forming the grid, i.e., $\delta^*(X_L) = \frac{\sqrt{n}}{2} L^{-1/n}$. Yet, as said, the number of points of such a regular grid is bound to be equal to b^n , which makes it unfeasible when the state dimension n is large.

IV. NON-REGULAR SAMPLING GRIDS

We recall the definition of *discrepancy*, a quantity that is closely related to the dispersion. The discrepancy of a set X_L in X is defined [11] as:

$$D(X_L) = \sup_{J \in \mathcal{J}} \left| \frac{A(X_L, J)}{L} - \lambda(J) \right| \quad (4)$$

where \mathcal{J} is a family of Lebesgue-measurable subsets of X , λ denotes the Lebesgue measure and $A(X_L, J)$ denotes the number of points of X_L contained in J . Here in particular we consider the case in which the family \mathcal{J} is the set of all the subintervals $J \in X$ having form $\prod_{i=1}^n [a_i, b_i] \subset \mathbb{R}^n$, where

$a_i \leq b_i$. Dispersion and discrepancy are closely related. In fact, it can be easily proved [11] that $\delta^*(X_L) \leq D(X_L)^{1/n}$, which implies that a generating algorithm ensuring convergence of the discrepancy also ensures also the convergence of the dispersion (notice that the converse is not true).

Remark 1: It is worth noting that, while there are several methods in the literature explicitly designed to provide sample sets characterized by favourable discrepancy rates, there are no principled solutions available for the dispersion. This is the main reason why we consider the former quantity for the rest of the analysis.

In the following, three different examples of uniform space-covering sequences are given that can be used as feasible alternatives to the regular uniform grid.

A. Uniform Random i.i.d. Sequences

The simplest solution to generate uniform points in the n -dimensional unit hypercube is a sequence of L points drawn randomly with uniform distribution. From the point of view of the discrepancy, it can be shown [13] that an i.i.d. sequence of L points drawn with uniform distribution is characterized by a convergence rate for the discrepancy of order $O((\log \log L)^{1/2}/\sqrt{n})$ almost surely. Thus, a random i.i.d. uniform sequence satisfies the condition of $\delta^*(X_L) \rightarrow 0$ concerning the convergence of the dispersion, with probability one. However this convergence guarantee is only probabilistic, which can be regarded as a drawback in our context of robust EMPC.

Another drawback is that the resulting sampling is often not well uniformly scattered in the state space, leading to the formation of clusters of points that may leave other zones of the state space undersampled. This means that the learning model has no information regarding the true EMPC function in those regions, and the resulting approximate control coming from the learning procedure becomes less accurate when the state visits those portions of the input space. Figure 1a illustrates an example of sampling of the 2-dimensional unit hypercube with $L = 500$ i.i.d. points, in which the clustering effect, together with the related undersampling of portions of the input space, is well noticeable.

B. Low-Discrepancy Sequences

The so-called ‘‘low-discrepancy sequences’’ are a family of algorithms for the generation of points born in the field of quasi-random numerical integration. Examples of such sequences are the Sobol’, the Faure, the Niederreiter and the Halton sequence [11], [14]. The aim of these algorithms is to provide point sets that cover the unit hypercube uniformly, in the sense that any subset contains a share of points that is proportional to its volume (following the definition of discrepancy), in a deterministic way. Figure 1b illustrates an example, namely 500 points from the 2-dimensional Sobol’ sequence. The construction of LDSs can be generalized through the concept of ‘‘digital (T, n) -sequences’’ (see, e.g., [14] for a comprehensive introduction and treatment).

More in detail, the construction of a (T, n) -sequence in base b , for b prime, is based on a set of n ($\mathbb{N} \times \mathbb{N}$) matrices

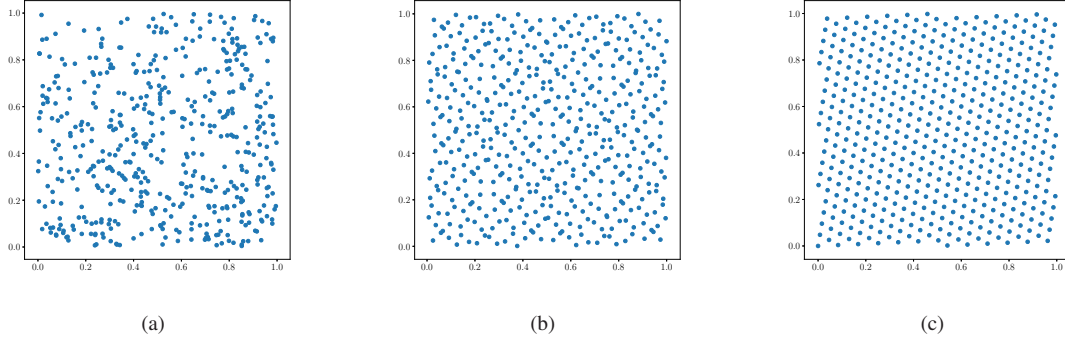


Fig. 1: Sampling of the 2-D unit cube with $L = 500$ points from the uniform distribution (a), the Sobol' sequence (b) and the R1LPS lattice with $z = [1, 131]$ (c).

C_1, \dots, C_n having elements in $Z_b = \{0, \dots, b-1\}$. Then, to generate the j -th point x_j of the sequence, we write j in its b -adic expansion, i.e., $j = \sum_{i=0}^{\infty} v_i b^i$ with digits $v_i \in Z_b$, and consider the infinite vector $v = (v_0, v_1, v_2, \dots)^T$. The p -th component of x_j is obtained first by multiplying the p -th matrix C_p by v , yielding the vector $(y_{j,p,1}, y_{j,p,2}, \dots)^T = C_p \cdot v$ where each element is again in Z_b , and finally taking $x_{j,p} = y_{j,p,1} b^{-1} + y_{j,p,2} b^{-2} + \dots$. Notice that, in practice, the infinite vector v can be truncated after a few elements.

The above-described procedure is not computationally intensive, and many of the aforementioned special instances, like the Sobol' sequence, can be found already implemented for the most common numerical software environments.

For the purpose of the EMPC method considered in this paper, the most noteworthy property of (T, n) -sequences is that they guarantee the convergence of the discrepancy to zero as L grows [11], with a rate of order $O(L^{-1}(\log L)^n)$. Furthermore, since the construction procedure is fixed, this convergence rate is deterministic. This eventually means that LDS sequences satisfy the condition stated in (2), thus ensuring that the robustness of the procedure.

C. Rank-1 Lattice Sample Sets

This kind of sequence can be generically ascribed to the family of digital sequences seen in the previous section, even if they were proposed independently for numerical integration [12]. Geometrically, a lattice consists of the infinite repetition of elementary unit cells, defined by a linear combination of n n -dimensional vectors. Within the unitary hypercube (or a closed hypercube in general) the points of the lattice can be expressed conveniently based on a single n -dimensional integer vector. More specifically, consider an integer vector $z \in \mathbb{Z}^n$ having no factor in common with L . Then, a lattice point set X_L of L points can be obtained as

$$\Sigma_t^L(z) = \left\{ \left\langle \frac{j}{L} z \right\rangle \text{ for } j = 0, \dots, L-1 \right\}, \quad (5)$$

where $\langle \cdot \rangle$ denotes the component-wise fractional part.

Such a formulation usually takes the name of *Rank-1 Lattice Point Set* (R1LPS) [12]. A standard choice for z , following Koroobov's rule [15], is to take $z = [1, w, w^2 \bmod L, \dots, w^{n-1} \bmod L]$ for an integer $w \leq \lfloor L/2 \rfloor$.

In Figure 1c a R1LPS lattice is shown, with $z = [1, 131]$.

Although their construction is very easy and straightforward to implement in any numerical software environment, the variability due to the choice of z implies that not all R1LPS necessarily offer an efficient covering of the state space. However, "good" lattice points satisfy the condition in (2). In fact, it can be proved [11] that the average rate of convergence of the discrepancy over z of a R1LPS is $O(L^{-1}(\log L)^n)$, and that there exists at least a z^* that yields this rate exactly.

To help in the choice of good vectors, various quantities have been devised in the literature to evaluate a priori the uniformity of a lattice depending on z . For instance, the P_α index (see, e.g., [11]) is related to the discrepancy in such a way that sets with low values of P_α are expected to be good also from the discrepancy (and, as a consequence, dispersion) point of view. Furthermore, such index can be explicitly computed (see, e.g., [12] Chapter 10.5), thus it can be used for a search over z to isolate "good" generating vectors for the R1LPS in the dimension n .

Remark 2: In general, some hints can be obtained on the most convenient non-regular scheme to be applied for the problem at hand, borrowing from the theoretical results available for the use of such methods in numerical integration. In particular, LDS points are expected to perform particularly well when the MPC function belongs to a weighted Sobolev space [14] (roughly speaking, when the involved functions depend strongly only on just on a few of the n components), while the LPS sets are particularly effective when the involved function exhibits strong spatial regularity (in the extreme case, when it is periodic in space) [12].

V. SIMULATION RESULTS

In this section, we present an application of the proposed sampling schemes to a 21-dimensional case study, involving a rectangular plate made of some thermally conductive material (e.g., copper) and subject to 4 heat sources that we control in order to impose a desired temperature profile. This kind of system is very common in industry (inductive heating on steel plants) and in the oil/gas sector.

Specifically, the purpose of the control is to raise or lower the temperature of the plate initially assumed to be equal to the

ambient temperature, using heat sources located on each side of the rectangle. Here we assume that each pair of opposite sides of the plate is controlled by the same source, eventually yielding two freely assignable control values. Let $\varphi(t, x, y)$ be the temperature of the plate at time $t > 0$ and point (x, y) , with $0 \leq x \leq L_x$ and $0 \leq y \leq L_y$. The classic heat equation governing the system is given by a partial differential equation of the form $\partial_t \varphi = \alpha(\partial_{xx} \varphi + \partial_{yy} \varphi)$. In order to apply the MPC scheme, we need to discretize the heat equation with respect to both time and space: we employ a forward time centered space approach, with a sample time $\Delta t = 0.05$ s and spatial stepsizes equal to $\Delta x = \Delta y = 0.1$ m. Therefore, the heat equation can be written in the form

$$\begin{aligned} \varphi(t+1, i, j) &= \varphi(t, i, j) \\ &+ \frac{\alpha \Delta t}{\Delta x^2} (\varphi(t, i-1, j) - 2\varphi(t, i, j) + \varphi(t, i+1, j)) \quad (6) \\ &+ \frac{\alpha \Delta t}{\Delta y^2} (\varphi(t, i, j-1) - 2\varphi(t, i, j) + \varphi(t, i, j+1)), \end{aligned}$$

where $\varphi(t, i, j) = \varphi(t \cdot \Delta t, i \cdot \Delta x, j \cdot \Delta y)$, $\alpha = 1.11 \cdot 10^{-4}$ m²/s (the conductivity of copper), $i = 1, \dots, n$ and $j = 1, \dots, m$. In the simulations, we consider $n = 8$ and $m = 5$ whereby $L_x = 0.7$ m and $L_y = 0.4$ m.

Boundary conditions must be added to make equation (6) defined for every temporal stage. More specifically, we assume $\varphi(t+1, 3, 1) = \varphi(t+1, 3, 5) = u_1(t)$ and $\varphi(t+1, 1, 3) = \varphi(t+1, 8, 3) = u_2(t)$, while all the other entries in the boundaries are kept fixed at the ambient temperature. The controls u_1 and u_2 are, therefore, able to set the temperature on two discretization points and they are constrained to assume values in the range $|u_1(t) - \varphi(t, 3, 1)| \leq 0.1$ °C and $|u_2(t) - \varphi(t, 1, 3)| \leq 0.05$ °C. This constraint reflects the requirement of not changing the plate temperature too abruptly. Furthermore, we want to limit the magnitude of the controls.

The aim of the control is to bring the temperature of the plate from an initial ambient temperature as close as possible to a target temperature using the four heat sources. Without loss of generality, we can always assume that the target temperature is zero, since the dynamics described by the heat equation are defined up to an additive constant. Therefore, we adopt the following functional cost over 5 temporal stages:

$$\begin{aligned} J(\varphi(0), u(0), \dots, u(4)) &= \sum_{k=0}^4 \|\tilde{\varphi}(k+1)\|^2 \\ &+ 10^{-3} (|u_1(k)|^2 + |u_2(k)|^2) \end{aligned}$$

such that

$$|u_1(k) - \varphi(k, 3, 1)| \leq 0.1, \quad k = 0, \dots, 4,$$

$$|u_2(k) - \varphi(k, 1, 3)| \leq 0.05, \quad k = 0, \dots, 4,$$

where $\tilde{\varphi}(t) = \varphi(t, i, j)$ for $i = 2, \dots, n-1$ and $j = 2, \dots, m-1$.

It is worth noting that the dimension of the state vector is 21: we have $\varphi(t, i, j)$ for $i = 2, \dots, n-1$ and $j = 2, \dots, m-1$ corresponding to the 18 variables ruled by equation (6); on the boundary, we have that $\varphi(t+1, 3, 1) = \varphi(t+1, 3, 5)$ and $\varphi(t+1, 1, 3) = \varphi(t+1, 8, 3)$, which amounts to other 2 variables,

LDS	LPS	IID	REG
0.028	0.033	0.025 (0.007)	2.53

TABLE I: Comparison of mean absolute errors between the MPC control obtained minimizing J and the output of the learned neural networks.

while all the other states on the boundary collapse into the ambient temperature (represented by the last state variable).

As the baseline reference for the training set X_L we have considered a regular grid of samples in the 21-dimensional hypercube with lower bounds -10 and upper bounds 10 . Specifically, we have considered the regular grid consisting of two discretization levels (-3.5 and 3.5) for each dimension. Since we have 21 dimensions, the total number of points L of this grid is $2^{21} = 2097152$. Notice that this is the only regular grid we can employ here in practice, since using three discretization levels would already yield more than 10 billion points. To test the non-regular discretization schemes proposed in Section IV, we have considered sets having the same number $L = 2^{21}$ points as the regular grid. In particular, in the case of LDS sampling the Sobol' sequence was employed [16]. For the LPS case we have followed Korobov's rule to obtain the generator vector z , choosing as w the value that minimized the index P_α (for $\alpha = 2$) mentioned in Section IV over 100 trial values. In case of i.i.d. uniform random sequences, due to their intrinsically probabilistic nature, 10 different sequences were generated.

In order to implement the EMPC procedure described in Section II, the class of two-hidden layers sigmoidal neural networks with 10 hidden units in every layer has been chosen to learn the MPC control. For the implementation, we used the Python scikit-learn package. The employed performance criterion for the optimization of the model parameters is the mean squared error with L_2 regularization of the weights, in order to keep the Lipschitz constant of the network bounded.

To quantify the performance of the networks we evaluate the generalization ability of the prediction model to correctly approximate the true MPC control law in the points of a test set, defined as a collection of samples not belonging to the training set. To this purpose, a test set made of 10000 points randomly generated in the range $[-10, 10]$ has been employed.

The mean absolute error (MAE) between the MPC control obtained minimizing J and the output of the various neural networks is displayed in Table I. Notice that in the case of the IID sequences the MAE reported is the average over the 10 realizations of the points of the various grids, while the standard deviation is reported in the parentheses.

Figure 2 compare: i) an example of the temporal evolution of the temperature over the plate (starting from the ambient temperature equal to -5 °C) obtained with the true MPC control, (ii) the approximation obtained with a non-regular sampling (specifically, the LDS one) and (iii) the approximation using the regular uniform grid.

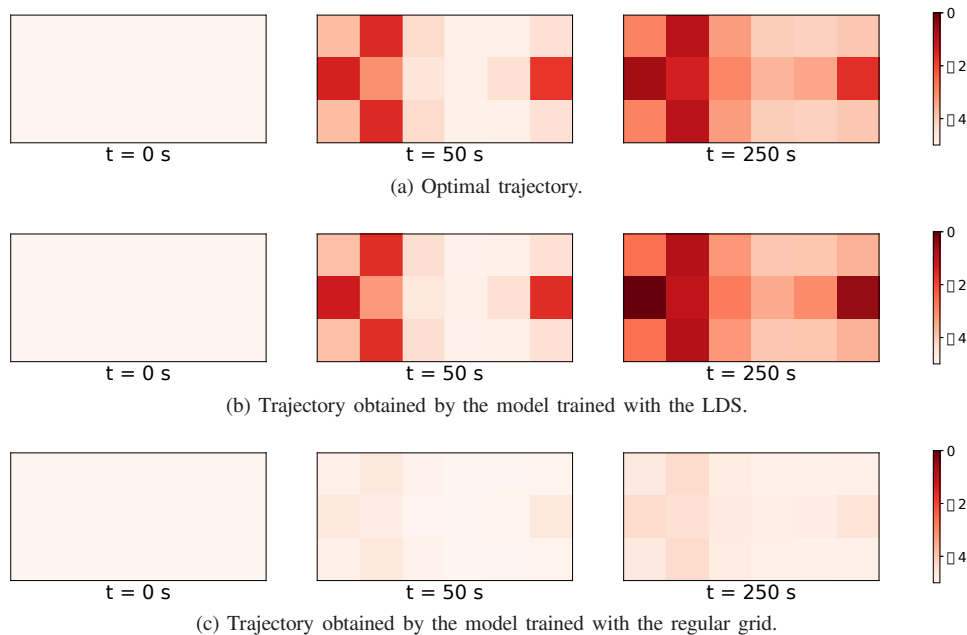


Fig. 2: Temporal evolution of the temperature profile of the plate, obtained with (a) the true MPC control (b) the approximate controls generated using a LDS non-regular scheme and (c) the approximate controls generated using the regular grid.

VI. COMMENTS AND CONCLUDING REMARKS

The results from Table I and Figure 2 show how the approximation of the MPC control is very good with the three different non-regular sampling schemes, while the uniform regular grid is not able to provide a successful approximation. In fact, Table I shows how, despite the same number L of points, the error from the regular grid turns out to be 2 orders of magnitude larger than the error from the non-regular ones. Accordingly, the temporal evolution in Figure 2 is almost identical in case of the true MPC control and the LDS approximation, reaching the best possible achievable temperature profile, while the approximation from the uniform grid never converges to a satisfactory profile. Overall, the simulation results prove how an alternative to the regular grid, by means of an algorithm able to spread the points well over the input space, is mandatory in high-dimensional contexts.

Among the non-regular schemes, Table I shows how the i.i.d. sampling performs slightly better than the rest in average, but the reported standard deviation shows that some of the i.i.d. grids in the tests perform worse than those from the other two sampling schemes. Overall, all the proposed sampling schemes yield actually very close results in Table I, showing that they all are viable options.

REFERENCES

- [1] J. B. Rawlings, D. Q. Mayne, and M. M. Diehl, *Model Predictive Control: Theory, Computation, and Design*. Nob Hill Publishing, 2018.
- [2] A. Bemporad and M. Morari, “Robust Model Predictive Control: A Survey,” *Lectures Notes in Control and Information Sciences*, vol. 245, pp. 207–226, 1999.
- [3] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. M. Scokaert, “Constrained Model Predictive Control: Stability and Optimality,” *Automatica*, vol. 36, pp. 789–814, 2000.
- [4] D. Raimondo, D. Limon, M. Lazar, D., L. Magni, and E. F. Camacho, “Min-Max Model Predictive Control of Nonlinear Systems: A Unifying Overview on Stability,” *European Journal of Control*, vol. 15, pp. 1–17, 2009.
- [5] 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. Raimondo, and F. Allgöwer, Eds. Springer, 2009, vol. 384, pp. 345–369.
- [6] 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.
- [7] R. Zoppoli, M. Sanguineti, G. Gnecco, and T. Parisini, *Neural Approximations for Optimal Control and Decision*. Springer Nature, 2020.
- [8] G. Pin, M. Filippo, F. Pellegrino, G. Fenu, and T. Parisini, “Approximate Model Predictive Control Laws for Constrained Nonlinear Discrete-Time Systems: Analysis and Offline Design,” *International Journal of Control*, vol. 86, no. 5, pp. 804–820, 2013.
- [9] A. Chakrabarty, V. Dinh, M. Corless, A. Rundell, S. Zak, and G. Buzard, “Support Vector Machine Informed Explicit Nonlinear Model Predictive Control Using Low-Discrepancy Sequences,” *IEEE Transactions on Automatic Control*, vol. 62, no. 1, pp. 135–148, 2016.
- [10] C. Cervellera, M. Gaggero, and D. Macciò, “Lattice Point Sets for State Sampling in Approximate Dynamic Programming,” *Optimal Control, Applications and Methods*, vol. 38, pp. 1193–1207, 2017.
- [11] H. Niederreiter, *Random Number Generation and Quasi-Monte Carlo Methods*. Philadelphia: SIAM, 1992.
- [12] I. H. Sloan and S. Joe, *Lattice Methods for Multiple Integration*. New York: Oxford Univ. Press, 1994.
- [13] K. L. Chung, “An Estimate Concerning the Kolmogoroff Limit Distribution,” *Trans. Amer. Math. Soc.*, vol. 67, pp. 36–50, 1949.
- [14] J. Dick and F. Pillichshammer, *Digital Nets and Sequences: Discrepancy Theory and Quasi-Monte Carlo Integration*. Cambridge, UK: Cambridge University Press, 2010.
- [15] N. M. Korobov, “The Approximate Computation of Multiple Integrals,” *Dokl. Akad. Nauk SSSR*, vol. 124, pp. 1207–1210, 1959.
- [16] I. M. Sobol’, “The Distribution of Points in a Cube and the Approximate Evaluation of Integrals,” *Zh. Vychisl. Mat. i Mat. Fiz.*, vol. 7, pp. 784–802, 1967.