Identification of Structured Dynamical Systems in Tensor Product Reproducing Kernel Hilbert Spaces

Marco Signoretto[1] and Johan A. K. Suykens[1]

***Abstract*— Recent research on statistical learning methods leveraged multilinear algebra and tensor-based models in reproducing kernel Hilbert spaces (RKHSs). We study implications of this framework for the identification of discrete-time dynamical systems that depend upon a (spatial) indexing variable. When this variable is discrete and the system of interest is linear, the proposed technique corresponds to learn a finite dimensional tensor under a constraint on the multilinear rank, hereby reducing the number of parameters in the estimation. More generally, the approach requires to learn a static mapping within a RKHS. The choice of reproducing kernel leads to different classes of models, which might depend nonlinearly on past inputs and outputs.**

I. INTRODUCTION

A key ingredient to improve the generalization of statistical learning algorithms is to incorporate structural information, either by choosing appropriate input representations or by tailored regularization schemes. Recent research on statistical learning methods leveraged multilinear algebra and tensor-based models to achieve this goal [15], [13]. In [14] this approach has been generalized within the context of reproducing kernel Hilbert spaces. The arising framework comprises existing problem formulations, such as tensor completion [7], as well as novel functional formulations. The approach is based on a class of regularizers, termed *multilinear spectral penalties*, that is related to spectral regularization for operator estimation [1]. In this paper we study implications of this framework for system identification. We propose an identification approach for a class of discrete-time dynamical systems with input-output representation. We are interested in the case where the parameters of the system are functions of a indexing variable x. This situation arises, in particular, in a number of environmental modeling applications. We show that, when the indexing variable is discrete and the system of interest is linear, a convenient approach prescribes to learn a finite dimensional tensor under a constraint on the multilinear rank. Equivalently, this task can be casted as a statistical learning problem with a multilinear spectral penalty. The goal of this learning problem is the estimation of a static mapping within a Hilbert space of functions with a certain reproducing kernel k. This gives rise to a flexible modelling tool; in fact, modifying k leads to different classes of models, which might depend nonlinearly on past inputs and outputs.

The paper is organized as follows. In the next section we introduce the class of problems of interest. In Section III we show how these problems can be tackled by learning a model in a space of tensor product functions. Section IV deals with multilinear spectral regularization. We present case studies in Section V and draw our concluding remarks in Section VI.

[1]ESAT-STADIUS, KU Leuven, Kasteelpark Arenberg 10 B-3001 Leuven (BELGIUM) marco.signoretto@esat.kuleuven.be and johan.suykens@esat.kuleuven.be

II. BACKGROUND AND PROBLEM SETTING

In the following we denote by $[I]$ the set of integers up to and including the integer I. We write $\times_{m=1}^{M}[I_m]$ to mean $[I_1]\times[I_2]\times\cdots\times[I_M]$, i.e., the cartesian product of such index sets, the elements of which are M-tuples $(i_1, i_2, \ldots, i_M)$. For a generic set $\mathcal{X}$, we write $\mathbb{R}^{\mathcal{X}}$ to mean the set of mappings from $\mathcal{X}$ to $\mathbb{R}$.

A. Systems with Linear Parameter Structure

For the sake of presentation we will start from the case where the system of interest is assumed to have a convenient linear parameter structure; we will then relax this assumption to include more general classes of models. The AutoRegressive model with eXogenous (ARX) input is:

$$y_s(x) = -\sum_{l=1}^{L} A_l(x)y_{s-l}(x) + \sum_{m=0}^{M} B_m(x)u_{s-m}(x) + \epsilon_s(x) \tag{1}$$

where s is the time index. Note that the exogenous input $u_s(x) \in \mathbb{R}^{N_u}$, as well as both the output and the noise process $y_s(x), \epsilon_s(x) \in \mathbb{R}^{N_y}$, depend upon the indexing variable x. Finally, $\{A_l\}_{l=1}^{L}$ and $\{B_m\}_{m=1}^{M}$ are matrix-valued functions of x with static dependence. In the following for simplifying the discussion and without loss of generality, we will assume that x represents a spatial coordinate.

Discrete-time systems in line with (1) are usually conceived to model (physical) processes that are continuous functions of time and space. Typical examples are transportation phenomena of mass or energy, such as heat transmission and/or exchange, humidity diffusion or concentration distributions. These systems are intrinsically distributed parameter systems whose description from first principles requires the introduction of partial differential equations which are necessarily an approximation of the underlying phenomena.

B. Curse of Dimensionality and Reduced Rank Regression

In general, it is well known that the estimation of multivariate time series models from data is hindered by the curse of dimensionality [5]. With reference to (1) consider the case where, for simplicity, $L = M$ and there is no dependence with respect to x; then the dimension of the parameter space is proportional to $N_y^2 + N_y N_u$ and therefore

scales quadratically with respect to the number of inputs and outputs. In *reduced-rank* (RR) regression [10], [12] the problem of reducing the number of parameters is approached through a low-rank assumption. Consider the case where $\mathcal{X} = [I_1] \times [I_2]$, i.e., the spatial index is discrete and consisting of two coordinates. In this case the model can be stated as:

$$y_s(i_1,i_2) = C(i_1,i_2)\, x_s(i_1,i_2) + \epsilon_s(i_1,i_2) : \ (i_1,i_2) \in [I_1] \times [I_2] \tag{2}$$

in which $C(i_1,i_2) \in \mathbb{R}^{N_y \times (LN_y + MN_u)}$ is given by:

$$C(i_1,i_2) = \big[A_1(i_1,i_2), \cdots, A_L(i_1,i_2), B_1(i_1,i_2), \cdots, B_M(i_1,i_2)\big] \tag{3}$$

and $x_s(i_1,i_2)$ is a vector of delayed inputs and outputs:

$$x_s(i_1,i_2) = \big[y_{s-1}^\top(i_1,i_2), \ldots, y_{s-L}^\top(i_1,i_2), u_s^\top(i_1,i_2), \ldots, u_{s-M}^\top(i_1,i_2)\big]^\top . \tag{4}$$

We refer to (2) as the *global* model. This consists of $I_1 I_2$ *local* models, each of which is indexed by a pair of indices (i_1,i_2). The ***RR estimator*** of $C(i_1,i_2)$, denoted by $\hat{C}_{RR}(i_1,i_2)$, is now:

$$\hat{C}_{RR}(i_1,i_2) = \sum_{r=1}^{R^*} \sigma_r \, u_r \, v_r^\top \tag{5}$$

where $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_{R^*} > 0$ and $\{u_r \ : \ r \in [R^*]\}$ / $\{v_r \ : \ r \in [R^*]\}$ are the $R^* < \min\{N_y, LN_y + MN_u\}$ leading *singular values* and left/right *singular vectors* [8] obtained from an estimate of $C(i_1,i_2)$ computed under no rank restrictions; see [10], [12] for additional details.

III. Identification of Tensor-based Models

A. Parameter Structure and Low Multilinear Rank Tensors

In general, RR regression has been shown to improve over alternative techniques, see for instance [6] for a case study in financial econometrics. Within the present context, however, the estimation of the $I_1 I_2$ local models is decoupled; the spatial structure of the global system is not accounted for. In order to introduce an approach that overcomes this limitation, we note the following. Identifying the global system (2) requires to find a *tensor*:

$$\gamma \in \mathbb{R}^{I_1 \times I_2 \times I_3 \times I_4} \ \text{ with } I_3 := N_y \text{ and } I_4 := LN_y + MN_u \, , \tag{6}$$

entry-wise given by:

$$\gamma_{i_1 i_2 i_3 i_4} = [C(i_1,i_2)]_{i_3 i_4} \ . \tag{7}$$

The reader is referred to [9] for a survey on tensors and tensor calculus. Here we recall that the *order* of a tensor is the number of dimensions, also known as *modes*. The tensor in (6)-(7), in particular, has 4 modes. The *mode-p unfolding* of a generic Pth order tensor $\gamma \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_P}$, is the matrix $M_p(\gamma) \in \mathbb{R}^{I_p \times I_1 I_2 \cdots I_{p-1} I_{p+1} \cdots I_P}$ constructed by stacking the column vectors obtained from $\gamma_{i_1 i_2 \cdots i_P}$ by fixing all but the pth index[1] i_p. The *multilinear rank* of γ, denoted by $\mathrm{mlrank}(\gamma)$, is now the P-tuple $(R_1, R_2, \ldots, R_P)$ entry-wise defined by $R_p := \mathrm{rank}(M_p(\gamma))$. Going back to the problem of interest, it is clear that finding a global model requires the estimation of a considerable number of parameters, each of which corresponds to an entry of γ in (6)-(7). A convenient approach to overcome this issue consists of constraining $\mathrm{mlrank}(\gamma)$. This amounts at imposing additional structure on the global system, as detailed in the following.

[1] In the tensor jargon, this means stacking the *mode-p fibers* of γ [11].

Proposition III.1. *With reference to* (6)-(7) *assume that* $\mathrm{mlrank}(\gamma) = (R_1, R_2, R_3, R_4)$. *Then there exist matrices with orthonormal vectors:*

$$U^{(p)} = \Big[u_1^{(p)}, u_2^{(p)}, \cdots, u_{R_p}^{(p)}\Big] \in \mathbb{R}^{I_p \times R_p}, \ p = 1, \ldots, 4 \tag{8}$$

and a tensor $\sigma \in \mathbb{R}^{R_1 \times R_2 \times R_3 \times R_4}$, *so that the local model* $C(i_1,i_2)$ *in* (2) *admits the representation:*

$$C(i_1,i_2) = \sum_{(l,m) \in [R_3] \times [R_4]} W_{lm}(i_1,i_2) u_l^{(3)} u_m^{(4)\top} \tag{9}$$

where

$$W_{lm}(i_1,i_2) = \sum_{(r_1,r_2) \in [R_1] \times [R_2]} \sigma_{r_1 r_2 lm} U^{(1)}_{i_1 r_1} U^{(2)}_{i_2 r_2} \ . \tag{10}$$

It is clear from (9) that $\mathrm{rank}(C(i_1,i_2)) \leq \min(R_3, R_4)$; therefore, a low-mlrank global model entails low-rank local models, as obtained in (5) by RR regression. Additionally, (9) shows that all local models are linear combinations of the same dictionary of rank-1 terms $\mathcal{T} := \{u_l^{(3)} u_m^{(4)\top} \ : \ (l,m) \in [R_3] \times [R_4]\}$. This clarifies that learning a global model with constraints on the multilinear rank amounts at simultaneously finding a compact dictionary $\mathcal{T}$ as well as the set of local weights $\{W(i_1,i_2) \ : \ (i_1,i_2) \in [I_1] \times [I_2]\}$.

B. From Linear To Nonlinear Models

Our next goal is to expand the class of models for the global dynamical system by relaxing linearity. To this end, denote by $\langle \gamma, \sigma \rangle$ the inner product between the Pth order tensors $\gamma, \sigma \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_P}$:

$$\langle \gamma, \sigma \rangle := \sum_{i_1 \in [I_1]} \cdots \sum_{i_p \in [I_P]} \gamma_{i_1 i_2 \cdots i_P} \sigma_{i_1 i_2 \cdots i_P} \ . \tag{11}$$

Let δ be the indicator function, $\delta(x) = 1$ if $x = 0$ and $\delta(x) = 0$ otherwise. One can check that (2) can be equivalently stated in terms of the parameter tensor (6)-(7) as:

$$[y_s(i_1,i_2)]_{i_3} = \big\langle \gamma, e_{i_1}^{(1)} \otimes e_{i_2}^{(2)} \otimes e_{i_3}^{(3)} \otimes x_s(i_1,i_2) \big\rangle + [\epsilon_s(i_1,i_2)]_{i_3} \, , \tag{12}$$

in which $e_{i_1}^{(1)} \otimes e_{i_2}^{(2)} \otimes e_{i_3}^{(3)} \otimes x_s(i_1,i_2)$ is the *rank-1 tensor* entry-wise defined by[2]:

$$\big[e_{i_1}^{(1)} \otimes e_{i_2}^{(2)} \otimes e_{i_3}^{(3)} \otimes x_s(i_1,i_2)\big]_{j_1 j_2 j_3 j_4} := x_s(i_1,i_2) \prod_{p=1}^{3} \delta(j_p - i_p) \, . \tag{13}$$

[2] The tensor $e_{i_1}^{(1)} \otimes e_{i_2}^{(2)} \otimes e_{i_3}^{(3)} \otimes x_s(i_1,i_2)$ corresponds to the outer product of the vector $x_s(i_1,i_2)$ and the canonical basis vectors $e_{i_p}^{(p)} \in \mathbb{R}^{I_p}, p = 1,2,3$ with $\big[e_{i_p}^{(p)}\big]_j = \delta(i_p - j)$.

Letting $f(i_1, i_2, i_3, x_s(i_1, i_2)) = \langle \gamma, e^{(1)}_{i_1} \otimes e^{(2)}_{i_2} \otimes e^{(3)}_{i_3} \otimes x_s(i_1, i_2) \rangle$ suggests now that estimating the global system could be phrased as the problem of learning a static mapping:

$$f : I_1 \times I_2 \times I_3 \times \mathbb{R}^{I_4} \to \mathbb{R} \tag{14}$$

in a suitable space of functions. In particular, consider the *tensor product reproducing kernel Hilbert space* (TP-RKHS) $\mathcal{H}_k$ with the reproducing kernel[3] [14]:

$$k\left((i_1, i_2, i_3, x), (i'_1, i'_2, i'_3, x')\right) = k^{(4)}(x, x') \prod_{p=1}^{3} \delta(i_p - i'_p) \ . \tag{15}$$

If we let $k^{(4)}(x, x') = x^\top x'$ in the latter, it can be shown that estimating γ in (12) corresponds to finding a function in $\mathcal{H}_k$. Additionally it is not difficult to see that, if

$$M_4(\gamma) = [w_1, w_2, \ldots, w_{I_1 I_2 I_3}] \tag{16}$$

and $\iota : [I_1] \times [I_2] \times [I_3] \to [I_1 I_2 I_3]$ denotes a one-to-one mapping between index sets, then we have:

$$f(i_1, i_2, i_3, x_s) = w^\top_{\iota(i_1, i_2, i_3)} x_s(i_1, i_2) \ . \tag{17}$$

This emphasizes the linear dependence on $x_s(i_1, i_2)$, in line with the model (2) from which we started from. A nonlinear dependence, on the other hand, can be obtained by working in a different TP-RKHS. One approach is to replace $k^{(4)}(x, x') = x^\top x'$ with the RBF-Gaussian kernel:

$$k^{(4)}(x, x'; \eta) = \exp\left(-\|x - x'\|^2 / \eta^2\right) \tag{18}$$

where η is a user defined parameter. This amounts at considering the case where

$$f(i_1, i_2, i_3, x_s) = g_{\iota(i_1, i_2, i_3)}(x_s(i_1, i_2)) \tag{19}$$

and $g_{\iota(i_1, i_2, i_3)}$ belongs to the space of nonlinear functions generated by the RBF-Gaussian kernel[4] [16].

Finally note that the approach is immediately extended to the continuous setting. Suppose that $\mathcal{X} = \mathbb{R} \times \mathbb{R}$ rather than $[I_1] \times [I_2]$, so that i_1, i_2 represent continuous spatial coordinates. We can encode a notion of proximity by replacing the indicator functions within (15), with kernels defined on continuous sets. Using again the RBF-Gaussian kernel, in particular, one has:

$$k\left((i_1, i_2, i_3, x), (i'_1, i'_2, i'_3, x')\right) = k^{(4)}(x, x'; \eta) \\ k^{(4)}(i_1, i'_1; \eta_1) k^{(4)}(i_2, i'_2; \eta_2) \delta(i_3 - i'_3) \ . \tag{20}$$

[3] We recall that a Hilbert space of functions $f : \mathcal{X} \to \mathbb{R}$ endowed with an inner product $\langle \cdot, \cdot \rangle$ is a RKHS, denoted by $\mathcal{H}_k$, if there exists a function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ (the reproducing kernel) such that [2]: *a.)* $k(\cdot, x) \in \mathcal{H}$, $\forall x \in \mathcal{X}$ and *b.)* $\langle f, k(\cdot, x) \rangle = f(x)$ $\forall x \in \mathcal{X}$, $\forall f \in \mathcal{H}$ (*reproducing property*). Note that *a* and *b* imply that k is symmetric. Additionally, $\mathcal{H}$ is called *tensor product RKHS* (TP-RKHS) if its reproducing kernel is further given by the point-wise product of a finite family of reproducing kernels $k^{(i)} : \ i \in [I]$, $k(x, x') = \prod_{i \in [I]} k^{(i)}(x, x')$.

[4] Note that, if $k^{(4)}_x := t \mapsto k^{(4)}(t, x)$, then the rank-1 tensor $e^{(1)}_{i_1} \otimes e^{(2)}_{i_2} \otimes e^{(3)}_{i_3} \otimes k^{(4)}_x$ is the *Riesz representation* [4] of the evaluation functional $L_{(i_1, i_2, i_3, x)}$ on $\mathcal{H}_k$, defined by $L_{(i_1, i_2, i_3, x)} f := f(i_1, i_2, i_3, x)$.

IV. Learning Models by Multilinear Spectral Regularization

In Section III-A we have shown that, when the system of interest is linear, a convenient identification approach prescribes to learn the finite dimensional tensor (6)-(7) under a constraint on the multilinear rank. More generally, one can use *multilinear spectral regularization* [14] to learn parsimonious tensor-based models from observational data in a TP-RKHS $\mathcal{H}_k$. This applies, in particular, to functions of the type (14). Depending on the choice of reproducing kernel $k^{(4)}$ one obtains different model classes for systems with input-output representation. The kernel function $k^{(4)}(x, x') = x^\top x'$ leads to linear systems of the type (1), and the approach ultimately corresponds to learn the parameter tensor γ in (6)-(7).

A. Penalized Empirical Risk Minimization Problem

In general, a learning approach based on multilinear spectral regularization requires solving penalized empirical risk minimization problems [18] of the type:

$$\min_{f \in H} E(f; \mathcal{D}) + \lambda \, \Omega(f; \mathcal{M}) \tag{21}$$

where H is a suitable subset of $\mathcal{H}_k$ and $E(f; \mathcal{D})$ is the empirical risk of f, i.e., the model misfit measured upon observational data $\mathcal{D}$. Other than the dataset, the approach depends upon a family of *unfolding operators* $\mathcal{M} = \{M_p \ : \ p \in [P]\}$ used within the function Ω, termed multilinear spectral penalty. The operators in $\mathcal{M}$ represent the natural generalization towards infinite-dimensional spaces of functions, of the notion of matrix unfolding given in Section (III-A). The penalty Ω is now constructed based upon the singular values of $M_p(f), \ p \in [P]$. As a specific example, define $R_p(f) := \min\{r : \sigma_{r+1}(M_p(f)) = 0\}$. A special case of multilinear spectral penalty is:

$$\Omega_R(f; \mathcal{M}) = \begin{cases} 0, & \text{if } R_p(f) \leq R^*_p, \ p \in [P] \\ \infty, & \text{otherwise} \end{cases} \tag{22}$$

where $(R^*_p \ : \ p \in [P])$ is a user-defined tuple. For this choice it can be shown that when the kernel $k^{(4)} = (x, x') = x^\top x'$ is employed within (15), and the penalty (21) is based upon a suitably defined family $\mathcal{M}$, solving (21) is equivalent to learning γ under a low multilinear-rank constraint. An alternative penalty, which combines the *nuclear* (a.k.a. *trace*) norm of the functional unfoldings with an upper-bound on the multilinear rank is given by:

$$\Omega_T(f; \mathcal{M}) = \begin{cases} \sum\limits_{p \in [P]} \sum\limits_{r \in [R_p(f)]} \sigma_r(M_p(f)), & \text{if } \Omega_R(f; \mathcal{M}) < \infty \\ \infty, & \text{otherwise} \ . \end{cases} \tag{23}$$

For (22), one has to fine-tune the tuple $(R^*_p \ : \ p \in [P])$, which could be a difficult model-selection task for some problems[5]. In contrast, when using (23) one can establish a loose upper-bound on the multilinear rank and then tune λ in (21) to obtain the desired level of sparsity, see [14].

[5] Note that, if we let $R_p \leq S$ for any $p \in [P]$ and some global upper bound S, there are S^P different tuples to choose from.

B. Finite Dimensional Optimization and Out-of-Sample Evaluations

In general, (21) is an infinite dimensional problem; nevertheless, the representer theorem for multilinear spectral penalties [14] shows that solutions admit a finite dimensional representation. Specifically, consider the problem of finding (14) in a TP-RKHS $\mathcal{H}_k$ with reproducing kernel (15). The dataset consists of N pairs of observations:

$$\mathcal{D} = \left\{ \left(g^{(n)}, z^{(n)} \right) \ : \ n \in [N] \right\} \subset (I_1 \times I_2 \times I_3 \times \mathbb{R}^{I_4}) \times \mathbb{R} \tag{24}$$

in which $g^{(n)}$ is a quadruple with ith entry denoted by $g_i^{(n)}$. We assume that the dataset is consistent with the behavior of the system, i.e., that:[6]

$$\mathcal{D} \subseteq \{((i_1, i_2, i_3, x_s(i_1, i_2)), [y_s(i_1, i_2)]_{i_3}) \ : \\ s \in [S], \ (i_1, i_2, i_3) \in [I_1] \times [I_2] \times [I_3]\} \tag{25}$$

where $x_s(i_1, i_2)$ is the vector of delayed inputs and outputs defined in (4). Let now $K^{(4)}$ be the $N \times N$ kernel matrix:

$$\left[K^{(4)} \right]_{mn} := k^{(4)} \left(g_4^{(m)}, g_4^{(n)} \right) \tag{26}$$

and assume that for a $N \times Q$ matrix $F^{(4)}$ one has the factorization[7] $K^{(4)} = F^{(4)} F^{(4)\top}$. Moreover, for $p \in [3]$, denote by $F^{(p)}$ the $N \times I_p$ matrix entry-wise defined by $[F^{(p)}]_{nj} = 1$ if $g_p^{(n)} = j$ and $[F^{(p)}]_{nj} = 0$, otherwise. The next result follows from Theorem 2 and Proposition 3 in [14], see also [14, Section 4.3].

Proposition IV.1. *If $\hat{f}$ is a solution to* (21)*, then there exists a tensor $\hat{\alpha} \in \mathbb{R}^{I_1 \times I_2 \times I_3 \times Q}$ such that, for any $g \in I_1 \times I_2 \times I_3 \times \mathbb{R}^{I_4}$, we have:*

$$\hat{f}(g) = \langle \hat{\alpha}, z \rangle \tag{27}$$

in which $z = \left(\bar{k}^{(2)}(g) F^{(2)\ddagger} \right) \otimes \cdots \otimes \left(\bar{k}^{(4)\top}(g) F^{(4)\ddagger} \right)$, we denoted by $F^{(p)\ddagger}$ the transpose of the pseudo-inverse of the matrix $F^{(p)}$ and $\bar{k}^{(p)}(g)$ is entry-wise defined by:

$$\left[\bar{k}^{(p)}(g) \right]_n := \begin{cases} \delta\left(g_p - g_p^{(n)} \right), & \text{if } p \in [3] \\ k^{(4)}\left(g_4, g_4^{(n)} \right), & \text{otherwise} \ . \end{cases} \tag{28}$$

Note that the tensor $\hat{\alpha}$ in (27) can be computed by solving a finite dimensional optimization problem. The nature of this problem, in turn, depends upon the choice of E and Ω in (21), see [14] for details. In the following we consider the case where the empirical risk is based on the *quadratic loss*:

$$E(f; \mathcal{D}) = \sum_{(g,z) \in \mathcal{D}} \left(f(g) - z \right)^2 \tag{29}$$

and the multilinear spectral penalty Ω combines an upper bound on the multilinear rank with the nuclear norm of the functional unfoldings, as in (23). The resulting finite dimensional optimization problem can be solved via the block-descent algorithm given in [14, Section 5.3], and termed MLR-SNN.

V. Case Studies

In this section we compare the approach illustrated in Section IV against an alternative method that does not keep into account the spatial structure of the underlying system.

A. Synthetic Problem

For the first test case we consider a synthetic dataset generated by a linear single-input and single-output dynamical system. With reference to (1), we took $L = 2$ and $M = 3$ with the spatial index ranging in $[20] \times [20]$; $\epsilon_s, \ u_s$ were taken to be independently generated white Gaussian noises with mean zero and variance $\sigma_\epsilon^2 = 0.0025$ and $\sigma_u^2 = 1$, respectively. We let $y_1(i_1, i_2) = y_0(i_1, i_2) = 0$ and generated output measurements according to (2) for $s = 2, \ldots, 300$. For each value of $(i_1, i_2, i_4) \in [20] \times [20] \times [6]$, we let $[C(i_1, i_2)]_{i_4} = \gamma_{i_1 i_2 i_4}$ where γ was a randomly generated tensor with $\mathrm{mlrank}(\gamma) = (3, 3, 3)$. Note that, whereas in (7) γ was a fourth order tensor, here it is a third order tensor; this results from removing the singleton dimension corresponding to $I_3 = 1$. Only a subset of measurements collected up to $s = 90$ were considered for training. More precisely, the measurements obtained at time s at the location (i_1, i_2) were included in the training set with probability p, where different values of p were considered. We compared MLR-SNN with $k^{(4)}(x, x') = x^\top x'$ - denoted as LIN-MLR-SNN - with Least Squares Support Vector machine for Regression (LS-SVR) with linear kernel [17], denoted as LIN-LS-SVR. Note that LS-SVR was used to learn the local models independently on each over. The regularization parameter within the two procedures was determined by cross-validation. The results on test data ($s > 90$) in terms of Mean Squared Error (MSE $\times 10^{-3}$) of 1-step ahead predictions is reported on Table I; specifically, we reported the mean and the standard deviation across 15 Monte Carlo runs.

TABLE I
MSE $\times 10^{-3}$ FOR THE SYNTHETIC PROBLEM

	p		
	0.1	0.2	0.3
LIN-LS-SVR	7.9 (0.73)	3.9 (0.08)	3.3 (0.02)
LIN-MLR-SNN	**2.9 (0.05)**	**2.7 (0.04)**	**2.6 (0.01)**

B. Temperatures Prediction

The second task that we considered amounts at predicting temperatures at 22 different cities scattered across Europe[8]. Each location coincides with a meteorological station where

[6]Note that for each $s \in [S]$ we require only a subset (possibly empty) of measurements from the global system, hereby allowing for missing observations in the data.

[7]Note that, for the linear kernel $k^{(4)}(x, x') = x^\top x'$, a factorization is readily given:

$$F^{(4)} = \left[g_4^{(1)}, \ g_4^{(2)}, \cdots, g_4^{(N)} \right]^\top \in \mathbb{R}^{N \times I_4} \ .$$

[8]The cities were, in alphabetical order: Amsterdam, Antwerp, Athens, Berlin, Brussels, Dortmund, Dublin, Eindhoven, Frankfurt, Groningen, Hamburg, Liege, Lisbon, London, Madrid, Milan, Nantes, Paris, Prague, Rome, Toulouse, Vienna.

daily data between January 1st, 2012 and October 16th, 2013 were collected from www.wunderground.com. The data consists of relevant continuous variables including humidity, wind direction and speed, pressure and min/max/average temperature measured at each location. The goal is to perform 1-day ahead predictions of the daily average temperature at each location. We followed the approach described above and deal with this task by learning a function $f : (i_1, i_2, x_s) \mapsto \hat{y}_s(i_1, i_2)$ within a TP-RKHS. In the latter, i_1 and i_2 represented, respectively, the latitude and longitude of each city, corresponding to a local model. Note that, differently than in (4), the ordering or regressors was kept constant across different locations, i.e., $x_s(i_1, i_2) = x_s(i'_1, i'_2) = x_s$ for any $i_1, i'_1 \in [22]$ and $i_2, i'_2 \in [22]$; x_s was a $1361-$dimensional vector of regressors given by the collection of relevant meteorological variables recorded at the times indexed by $s-1$, $s-2$ and $s-3$. Within MLR-SNN we used the kernel:

$$k\left((i_1, i_2, x), (i'_1, i'_2 x')\right) = k^{(4)}(x, x'; \eta) \\ k^{(4)}(i_1, i'_1; \eta_1) k^{(4)}(i_2, i'_2; \eta_2) \quad (30)$$

where we set $\eta_1^2 = \eta_2^2 = d$, and d was taken to be the equal to the median squared distance between the cities. Similarly, η^2 was taken to be equal to the median squared distance of regressors in the third mode. We refer to this approach as RBF-MLR-SNN. The regularization parameter used within RBF-MLR-SNN was chosen according to $5-$fold cross-validation. This approach was compared against LS-SVR with linear kernel[9], denoted as LIN-LS-SVR, and used to learn the local models independently on each over. Training data consisted of a fraction p of input output pairs, selected at random. Specifically, for each p, $\mathcal{D}^{(p)} \subseteq \{((i_1, i_2, x_s), y_s(i_1, i_2)) \ : \ s \in [S], \ (i_1, i_2) \in [22] \times [22]\}$ where $[S]$ indexed the daily data between January 1st, 2012 and June 6th, 2013. The full set of data corresponding to the remaining days were considered for testing purposes. The results in Table II refer to the median value measured across the different cities of the MSE of 1-step ahead predictions on test data.

TABLE II

MSE FOR THE TEMPERATURES PREDICTION PROBLEM

	p		
	0.1	0.2	0.3
LIN-LS-SVR	7.31 (0.48)	6.43 (0.29)	6.38 (0.22)
RBF-MLR-SNN	**4.58** (0.33)	**4.25** (0.31)	**4.06** (0.26)

VI. CONCLUSIONS

We have presented an identification approach for a class of discrete-time dynamical systems, which depends upon a spatial index. The problem is dealt with by learning a function within a TP-RKHS. Specifically, learning is performed based on multilinear spectral regularization, which allows one to accounts for the spatial structure of the system. For the linear case, in particular, learning a global model with constraints on the multilinear rank amounts at simultaneously finding a compact dictionary as well as the set of local weights, hereby reducing the number of parameters needed for the global model.

[9]The RBF-Gaussian kernel led to worse results, which are not reported.

ACKNOWLEDGMENT

The authors thank Jeroen De Haas, Gervasio Puertas and Rocco Langone for collecting and preprocessing the data used within Section V-B. The scientific responsibility is assumed by the authors. Research supported by the Research Foundation of Flanders (FWO), Research Council KUL: GOA/10/09 MaNet, PFV/10/002 (OPTEC); CIF1 STRT1/08/23; Flemish Government: IOF: IOF/KP/SCORES4CHEM, FWO: projects: G.0588.09 (Brain-machine), G.0377.09 (Mechatronics MPC), G.0377.12 (Structured systems), G.0427.10N (EEG-fMRI), IWT: projects: SBO LeCoPro, SBO Climaqs, SBO POM, EUROSTARS SMART iMinds 2013, Belgian Federal Science Policy Office: IUAP P7/19 (DYSCO, Dynamical systems, control and optimization, 2012-2017), EU: FP7-EMBOCON (ICT-248940), FP7-SADCO (MC ITN-264735), ERC ST HIGHWIND (259 166), ERC AdG A-DATADRIVE-B (290923). COST: Action ICO806: IntelliCIS.

REFERENCES

[1] J. Abernethy, F. Bach, T. Evgeniou, and J.P. Vert. A new approach to collaborative filtering: Operator estimation with spectral regularization. *Journal of Machine Learning Research*, 10:803–826, 2009.

[2] N. Aronszajn. Theory of reproducing kernels. *Transactions of the American Mathematical Society*, 68:337–404, 1950.

[3] A. Berlinet and C. Thomas-Agnan. *Reproducing Kernel Hilbert Spaces in Probability and Statistics*. Kluwer Academic Publishers, 2004.

[4] J.B. Conway. *A Course in Functional Analysis*. Springer, 1990.

[5] M. Deistler, B.D.O. Anderson, A. Filler, and W. Chen. Modelling high dimensional time series by generalized factor models. In *Proceedings of the 19th International Symposium on Mathematical Theory of Networks and Systems–MTNS*, volume 5, 2010.

[6] M. Deistler and E. Hamann. Identification of factor models for forecasting returns. *Journal of Financial Econometrics*, 3(2):256–281, 2005.

[7] S. Gandy, B. Recht, and I. Yamada. Tensor completion and low-n-rank tensor recovery via convex optimization. *Inverse Problems*, 27(2), 2011.

[8] G. H. Golub and C. F. Van Loan. *Matrix Computations*. Johns Hopkins University Press, third edition, 1996.

[9] Wolfgang Hackbusch. *Tensor spaces and numerical tensor calculus*, volume 42. Springer, 2012.

[10] Alan Julian Izenman. Reduced-rank regression for the multivariate linear model. *Journal of multivariate analysis*, 5(2):248–264, 1975.

[11] T.G. Kolda and B.W. Bader. Tensor decompositions and applications. *SIAM review*, 51(3):455–500, 2009.

[12] Gregory C Reinsel and Rajabather Palani Velu. *Multivariate reduced-rank regression: theory and applications*. Springer New York, 1998.

[13] B. Romera-Paredes, H. Aung, N. Bianchi-Berthouze, and M. Pontil. Multilinear multitask learning. In *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, pages 1444–1452, 2013.

[14] M. Signoretto, L. De Lathauwer, and J. A. K. Suykens. Learning tensors in reproducing kernel Hilbert spaces with multilinear spectral penalties. *arXiv:1310.4977v1*, 2013.

[15] M. Signoretto, Q. Tran Dinh, L. De Lathauwer, and J. A. K. Suykens. Learning with tensors: a framework based on convex optimization and spectral regularization. *Machine Learning, DOI 10.1007/s10994-013-5366-3*, 2013.

[16] I. Steinwart, D. Hush, and C. Scovel. An explicit description of the reproducing kernel Hilbert spaces of Gaussian RBF kernels. *IEEE Transaction on Information Theory*, 52:4635–4643, 2006.

[17] J. A. K. Suykens, T. Van Gestel, J. De Brabanter, B. De Moor, and J. Vandewalle. *Least squares support vector machines*. World Scientific, 2002.

[18] V. Vapnik. *Estimation of dependences based on empirical data (translated by Samuel Kotz)*. Springer-Verlag New York, 1982.