Nonlinear system identification via data augmentation ⋆

Simone Formentin[a], Mirko Mazzoleni[b], Matteo Scandella[b], Fabio Previdi[b]

[a] *Dept of Electronics, Information and Bioengineering, Politecnico di Milano, Via G. Ponzio 34/5, 20133 Milano, Italy.*
[b] *Dept of Management, Information and Production Engineering, University of Bergamo, Via G. Marconi 5, 24044 Dalmine (BG), Italy.*

Abstract

This paper presents a novel nonparametric approach to the identification of nonlinear dynamical systems. The proposed methodology exploits the potential of manifold learning on an artificially augmented dataset, obtained without running new experiments on the plant. The additional data are employed for approximating the manifold where input regressors lie. The knowledge of the manifold acts as a prior information on the system, that induces a proper regularization term on the identification cost. The new regularization term, as opposite to the standard Tikhonov one, enforces local smoothness of the function along the manifold. A graph-based algorithm tailored to dynamical systems is proposed to generate the augmented dataset. The hyperparameters of the method, along with the order of the system, are estimated from the available data. Numerical results on a benchmark Nonlinear Finite Impulse Response (NFIR) system show that the proposed approach may outperform the state of the art nonparametric methods.

Keywords: System Identification; Semi-Supervised Learning.

1. Introduction

In the last decade, kernel methods [21] have shown their potential when used to learn dynamical systems, both in the linear and in the nonlinear framework [20, 22], as well as in time and frequency domain [10]. Unlike other widely used approaches, kernel methods work with infinite dimensional models in a nonparametric fashion. In order to avoid overfitting, they are usually equipped with a proper regularization term [11]. One of the main advantages of such methodologies is that they alleviate the model complexity selection issue. To this aim, parametric approaches like the prediction error method (PEM) [15] are often coupled with complexity criteria such as the Akaike Information Criterion (AIC) or the Bayesian Information Criterion (BIC) [15]. However, the resulting performance is usually not satisfactory, especially for short and noisy observations [19]. Instead, regularization turned out to be one of the most effective tools to manage the bias-variance trade-off of statistical models and impose a proper degree of smoothness [1].

Artificially augmented datasets can be seen as a way to induce regularization in model estimates. As an example, consider the problem of learning a parametric linear model. Ridge regression can be seen as the estimation procedure obtained via simple linear regression on an augmented dataset, where all the new outputs are set to zero [12]. Another example is the Vicinal Risk Minimization (VRM) principle [8], where additional virtual examples are drawn from a pre-defined vicinity distribution of the training examples. The authors of [8] showed how the VRM approach is an equivalent way to derive the Ridge regression as well as the Support Vector Machine (SVM) solutions of [12]. The enlargement of the available dataset is nowadays a standard tool also for training deep neural networks, in particular when performing image classification [14]. Finally, in [1], model constraints are obtained by adding artificial data that satisfy them inside the training set. Differently from previously cited methods, here the learning "hints" are designed by relying only on the independent variables, i.e., the regressors.

In this work, we investigate kernel-based estimation of nonlinear dynamical systems via regularization using artificially augmented datasets. Such an approach seems particularly promising in all applications where there is some prior knowledge about the system, but only few data are available as running new experiments is difficult or too costly, see e.g. some biomedical systems like glucose dynamics [9] or industrial plants like [27, 7]. More specifically, we will consider Nonlinear Finite Impulse Response (NFIR) systems, in that they represent a wide range of applications [2] and, for such models, augmented regressors can be generated without running new experiments on the systems. The generation of fictitious input/output data reflecting the system dynamics is not a trivial extension and will be treated in future works. The augmented regressors can be used

⋆A preliminary version of this work has been presented in [16].

Corresponding author: S. Formentin, Tel. +39 02 2399 3498.
Email addresses: simone.formentin@polimi.it (Simone Formentin), mirko.mazzoleni@unibg.it (Mirko Mazzoleni), matteo.scandella@unibg.it (Matteo Scandella), fabio.previdi@unibg.it (Fabio Previdi)

for *manifold learning*, i.e., to learn the subspace where the data lie, in order to regularize the model estimate.

Manifold learning methods have already been employed for system identification in [17]. However, in the above papers: (i) only the *transductive learning* problem is coped with, that is, the problem of estimating new outputs and not that of inferring the model equations is addressed; (ii) manifold regression and system identification on a lower-dimensional space are treated as two separate tasks, whereas in this work the final algorithm turns out to be a *one-shot* learning procedure; (iii) only parametric modeling was considered. In light of these acknowledgements, we extend the existing research with the following contributions: (i) we introduce a novel *nonparametric* method for *inductive learning* of nonlinear dynamical systems employing an artificially augmented dataset and relying upon manifold regularization. Since we will show that this problem is equivalent to a semi-supervised regression problem, we will call the overall procedure *Semi-Supervised System Identification* (S^3I) from now on; (ii) usually, in semi-supervised problems, the additional additional unsupervised data is *a-priori given*: in this work, we propose a method to artificially *generate* the unsupervised points for dynamical system identification; (iii) we optimize the hyperparameters in a rigorous way; (iv) we explicitly embed the dynamic properties of the system into the manifold regularization framework, thus also allowing a novel interpretation of the manifold regularization scheme for dynamical systems.

The remainder of the paper is organized as follows. Section 2 revisits the background on semi-supervised learning to motivate this study on dynamical systems. Section 3 provides the problem formulation. Section 4 discusses manifold learning to exploit the augmented dataset. Section 5 presents the overall approach, including a technique to generate the artificial data and an explanation of the resulting regularization rationale. A discussion about the different regularization terms is given in Section 6. Numerical results and a comparison with the state of the art are illustrated in Section 7. The paper is ended by some concluding remarks.

2. Background and motivation

Semi-supervised learning is not a new concept in data-driven function mapping and has been widely used both in classification [7] and regression [17] problems. In both cases, the aim is to learn the function that generates the output y. When, in addition to the supervised data, other inputs are available (without the corresponding output), their position in the regressors space gives additional information about the values of the unknown y's [7]. It becomes clear that, whenever the input points belong to a manifold in the regressors space, their distribution provides additional information about the function to learn. Consider a classification problem where only some (labeled) points are known to belong to a certain class, whereas the others (unlabeled) correspond to an unknown class. Intuitively, if regressors lie on a manifold, the class of unlabeled points is likely to be the same of the nearest (along the manifold) labeled ones. This rationale can be extended to dynamical systems. As an example, consider the linear Finite Impulse Response (FIR) model:

$$y(t) = u(t) + u(t-1) + e(t), \tag{1}$$

where $u(t) = 0.8u(t-1) + \eta(t)$ and $e(t), \eta(t) \sim WN(0,1)$, $\varphi(t) = [u(t), u(t-1)]^T \in \mathbb{R}^{2\times 1}$. Figure 1 depicts a random sampling of the regressors over a given time window for model (1).

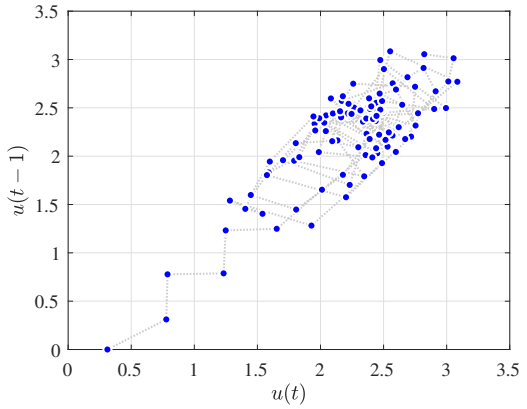

Figure 1: Regressor sampling for the system in (1).

It can be noticed that, due to the intrinsic correlation among the regressors' components in dynamical models, the position of the points within the regressors' space is not random. Instead, one may argue that the points are likely to lie on a certain manifold. This observation is confirmed if Principal Component Analysis (PCA) [12] is applied to the data of Figure 1: in fact, the first principal component can explain 91% of the data variance. This means that one dimension can be neglected without significant loss of information and, therefore, bias and variance can be effectively traded off to improve the model estimate. In this paper, the case of dynamical systems will be treated for the first time.

3. Problem formulation

Consider the NFIR Single-Input Single-Output (SISO) model be defined as:

$$\mathcal{S}: \; y(t+1) = g\big(\varphi(t)\big) + e(t), \tag{2}$$

where $y(t) \in \mathbb{R}$ denotes the system output, g is a nonlinear function, $\varphi(t) = [u(t), \cdots, u(t-m+1)]^T \in \mathbb{R}^{m\times 1}$ is the regression vector and $e(t) \in \mathbb{R}$ is an additive white noise. From now on, m will be referred to as the *model order*.

The objective of this work is to identify the system (2). An estimation of the model order m will be provided. We suppose furthermore that two different datasets are available: a *supervised* dataset $\mathcal{D}_S$ and an *unsupervised* dataset $\mathcal{D}_U$. The supervised dataset is such that:

$$\mathcal{D}_S = \left\{ \left(u_S(t), y_S(t) \right) \quad | \quad 1 \leq t \leq N_S \right\}, \tag{3}$$

where $u_S(t)$ and $y_S(t)$ are the measured input and output signals at time t and N_S is the number of supervised data.

The unsupervised dataset $\mathcal{D}_U$ has dimension of N_U and it is defined as:

$$\mathcal{D}_U = \left\{ u_U(t) \quad | \quad 1 \leq t \leq N_U \right\}, \tag{4}$$

where $u_U(t)$ is an input sequence for which the corresponding output is not available. To obtain a more compact representation, we will represent the observations and the regressors in a matrix form. Concerning the supervised dataset $\mathcal{D}_S$, we define the output vector $Y \in \mathbb{R}^{N \times 1}$:

$$Y = \begin{bmatrix} y_S(m+1) & \cdots & y_S(N_S) \end{bmatrix}^T, \tag{5}$$

where $N = N_S - m$ is the number of output samples that can be employed for the identification part, given the model order m. In the same way, it is possible to construct the N *supervised* model regressors $\varphi_S(t) \in \mathbb{R}^{m \times 1}$, for $m \leq t \leq N_S - 1$, as:

$$\varphi_S(t) = \begin{bmatrix} u_S(t) & \cdots & u_S(t-m+1) \end{bmatrix}^T. \tag{6}$$

Analogously, $N_{rU} = N_U - m + 1$ *unsupervised* model regressors $\varphi_U(t) \in \mathbb{R}^{m \times 1}$ can be collected, for $m \leq t \leq N_U - 1$, as:

$$\varphi_U(t) = \begin{bmatrix} u_U(t) & \cdots & u_U(t-m+1) \end{bmatrix}^T. \tag{7}$$

For simplicity, we define the generic regressor $\varphi(t)$ as:

$$\varphi(t) = \begin{cases} \varphi_S(t+m-1) & 1 \leq t \leq N \\ \varphi_U(t+m-N-1) & N+1 \leq t \leq N_r \end{cases} \tag{8}$$

where $N_r = N + N_{rU}$ is the total number of regressors. The $t-$th row of Y will be denoted as $y(t)$.

4. Manifold regularization

This section shows how unsupervised data can be effectively employed in a learning framework. In particular, the use of additional data is helpful for approximating the manifold where the regressors evolve. The discussion of the manifold regularization concepts will use the notation introduced in Section 3.

Section 2 gave intuitive motivations of how the geometry of the inputs space acts as an additional information that can be employed for learning. In order to embed this notion into a learning framework, we can resort to the following rationale. In the classical literature on learning from examples [25], the aim is to estimate the conditional distribution $p(y|\varphi)$ describing possible outputs values, given the corresponding input regressor φ. To do this, some samples (φ_S, y_S) are drawn from $p(\varphi, y)$ and used to build $\mathcal{D}_S$. Unsupervised examples φ_U can also be extracted according to the marginal distribution $p(\varphi)$ and used to build $\mathcal{D}_U$. The knowledge of $p(\varphi)$ can be useful if a *specific assumption is made* about the connection between the marginal and the conditional distributions [4]. For example, one may assume that, if two points φ_1, φ_2 are *close* according to some metrics in $p(\varphi)$, then the conditional distributions $p(y|\varphi_1)$ and $p(y|\varphi_2)$ are similar. In other words, the conditional probability distribution $p(y|\varphi)$ varies smoothly along the geodesics in the *intrinsic geometry* of $p(\varphi)$. The aforementioned assumption can be stated as follows [4]:

Assumption 1 (Semi-supervised smoothness). *If two regressors $\varphi(i)$ and $\varphi(j)$ in a high-density region are close, then so should be their corresponding outputs $y(i)$ and $y(j)$.*

Note that, if Assumption 1 holds, the solution is constrained to be *locally* smooth, i.e., smooth over the manifold where the regressors lie. Therefore, it can be formulated as a constraint (or an equivalent regularization term) for the learning algorithm. An effective way to write a regularization term enforcing Assumption 1 has been first proposed in [5]. In detail, if the support of $p(\varphi)$ is a compact manifold $\mathcal{G} \subset \mathbb{R}^m$, a common indicator of the degree of smoothness over the manifold is:

$$S_g = \int_{\mathcal{G}} \|\nabla \cdot g\|_2^2 \, dp(\varphi) = \int_{\mathcal{G}} g \cdot \Delta \cdot g \, dp(\varphi), \tag{9}$$

where ∇ and Δ are the gradient and the Laplace-Beltrami operators along the manifold $\mathcal{G}$, respectively. The integral is taken with respect to the marginal distribution $p(\varphi)$ [4]. The main idea behind such a manifold regularization is that, if Assumption 1 holds, the gradient of g (along $\mathcal{G}$), and so S_g, must be small. Then, minimizing S_g is a way to leverage Assumption 1. From (9), we see that the Laplacian is related to the squared norm of the gradient.

Unfortunately, $p(\varphi)$ and $\mathcal{G}$ are usually unknown and the smoothness index S_g in (9) cannot be computed. One way to model the manifold is by employing a *regressor graph* [4]. The model is a weighted and completely connected graph, with the (supervised and unsupervised) regressors as its vertices. The *intrinsic structure* of the regressors space is thus revealed by both supervised and unsupervised points. The weight of each edge, where $\sigma_e \in \mathbb{R}$ is a tuning parameter, is defined as

$$w_{i,j} = \exp\left(-\frac{\|\varphi(i) - \varphi(j)\|^2}{2{\sigma_e}^2} \right).$$

A high value of $w_{i,j}$ indicates that two regressors are

similar. Notice that the concept of "smoothness over a manifold" expressed through (9) may be seamlessly translated into a discrete graph domain.

Consider the Laplacian graph matrix $L = D - W$, where $D \in \mathbb{R}^{N_r \times N_r}$ is the diagonal matrix with elements $D_{ii} = \sum_{j=1}^{N_r} w_{i,j}$, and $W \in \mathbb{R}^{N_r \times N_r}$ is the matrix composed by the weights $w_{i,j}$. It can be shown that using exponential weights leads to the convergence of L to Δ [3]. By considering *graph derivatives* [24], the rhs of (9) can be represented by the *Laplacian quadratic form* [4, 24]:

$$S_g \simeq \widetilde{F}^T \cdot L \cdot \widetilde{F}, \tag{10}$$

where $\widetilde{F} = \Big[g\big(\varphi(1)\big), \cdots, g\big(\varphi(N_r)\big)\Big]^T \in \mathbb{R}^{N_r \times 1}$ depends only upon the unknown g and the input regressors[2]. It follows that *both supervised and unsupervised datasets* can be employed for weighting S_g within a learning task for regularizing the manifold. We will refer to (10) as the *manifold regularization* term.

Remark. From the above discussion, it comes out that, if Assumption 1 is not satisfied, the use of an additional unsupervised dataset is not beneficial. However, in all cases where Assumption 1 holds, the proposed approach may take advantage of such prior information to more accurately identify the unknown system.

5. Identification with data augmentation

This section presents the proposed learning methodology, highlighting each stated contributions.

5.1. A manifold-regularized identification approach

Suppose now that g belongs to a RKHS $\mathcal{H}$ defined using a kernel K. The kernel can depend by some hyperparameters η. The typical variational formulation consists into finding the best function $\widehat{g}$ according to the criterion [23]:

$$\widehat{g} = \underset{g \in \mathcal{H}}{\arg\min} \sum_{t=1}^{N} \Big(y\,(t) - g\big(\varphi\,(t)\,\big)\Big)^2 + \lambda_T \cdot \|g\|_{\mathcal{H}}^2, \tag{11}$$

where the summation spans the available N supervised regressors, $\|g\|_{\mathcal{H}}^2$ is the Tikhonov regularization term and $\lambda_T > 0 \in \mathbb{R}$ controls the regularization strength. The solution to (11) can be found by referring to the *representer theorem* [13]:

$$\widehat{g}\big(\varphi(t)\big) = \sum_{s=1}^{N} c_s K\big(\varphi\,(t)\,, \varphi\,(s)\,\big), \tag{12}$$

[2]The structure of the regularization term in (10) is shared by many manifold learning methods, where L is substituted by other symmetric matrices [6]. The reason is that such algorithms are still based on Assumption 1, but they formalize it from different perspectives.

for a N-tuple $c = [c_1, c_2, \ldots, c_N]^T \in \mathbb{R}^{N \times 1}$. Making use of (12), the Tikhonov regularization term of (11) can be restated as $\|g\|_{\mathcal{H}}^2 = c^T \mathcal{K} c$, where $\mathcal{K} \in \mathbb{R}^{N \times N}$ is a semidefinite positive and symmetric matrix (also called Gram matrix or kernel matrix) such that $\mathcal{K}_{ij} = K\big(\varphi(i), \varphi(j)\big)$. The matrix $\mathcal{K}$ is formed by using only the supervised regressors. Using (12), we can write the minimization problem (11) in such a way that it depends only on the unknown vector $c \in \mathbb{R}^{N \times 1}$:

$$\widehat{c} = \underset{c \in \mathbb{R}^N}{\arg\min} \; \| Y - \mathcal{K} c \|_2^2 + \lambda_T \cdot c^T \cdot \mathcal{K} \cdot c. \tag{13}$$

It is then possible to find the estimate of the vector c by solving the system:

$$\Big[\mathcal{K} + \lambda_T \cdot I_N\Big] \cdot \widehat{c} = Y, \tag{14}$$

where ψ includes the hyperparameters, that - in the case of (11) - are $\psi = [\lambda_T, \eta]$.

In order to include information about the local smoothness of the function (using the unsupervised data points), it is meaningful to add the *manifold regularization term* (10) to (11), leading to [4]:

$$\begin{aligned}\widehat{g} = \underset{g \in \mathcal{H}}{\arg\min} \sum_{t=1}^{N} \Big(y\,(t) - g\big(\varphi\,(t)\,\big)\Big)^2 \\ + \lambda_T \cdot \|g\|_{\mathcal{H}}^2 + \lambda_M \cdot \widetilde{F}^T \cdot L \cdot \widetilde{F},\end{aligned} \tag{15}$$

where $\lambda_M > 0 \in \mathbb{R}$ plays the same weighting role as λ_T.

It is possible to show that the representer theorem still holds for the cost function (15) and the solution can be written by considering all $N_r = N + N_{rU}$ regressors [4]:

$$\widehat{g}\big(\varphi(t)\big) = \sum_{s=1}^{N_r} \tilde{c}_s K\big(\varphi\,(t)\,, \varphi\,(s)\,\big), \tag{16}$$

for a N_r-tuple $\tilde{c} = [\tilde{c}_1, \tilde{c}_2, \ldots, \tilde{c}_{N_r}]^T \in \mathbb{R}^{N_r \times 1}$. The vector $\widetilde{F}$ introduced in (10) can now the be rewritten as $\widetilde{F} = \widetilde{\mathcal{K}}\tilde{c}$, where $\widetilde{\mathcal{K}} \in \mathbb{R}^{N_r \times N_r}$ is the kernel matrix constructed considering both supervised and unsupervised regressors. Notice that $\widetilde{\mathcal{K}}$ depends on the kernel hyperparameters η and may depend also on some hyperparameters δ used to generate the augmented dataset. Now, by means of (16), it is possible to write the minimization problem (15) in such a way that it depends only on the unknown vector $\tilde{c} \in \mathbb{R}^{N_r \times 1}$:

$$\begin{aligned}\widehat{\tilde{c}} = \underset{\tilde{c} \in \mathbb{R}^{N_r}}{\arg\min} \; \Big\| \mathcal{Y} - P \cdot \widetilde{\mathcal{K}}\tilde{c} \Big\|_2^2 \\ + \lambda_T \cdot \tilde{c}^T \cdot \widetilde{\mathcal{K}} \cdot \tilde{c} + \lambda_M \cdot \tilde{c}^T \cdot \widetilde{\mathcal{K}} L \widetilde{\mathcal{K}} \cdot \tilde{c},\end{aligned} \tag{17}$$

where $\mathcal{Y} = \begin{bmatrix} Y^T & 0_{N_{rU}}^T \end{bmatrix}^T \in \mathbb{R}^{N_r \times 1}$, with

$0_{N_{rU}} \in \mathbb{R}^{N_{rU}\times 1}$ a column vector of zeros. The matrix

$$P = \begin{bmatrix} I_N & 0 \\ 0 & 0 \end{bmatrix},$$

that is such that $P \in \mathbb{R}^{N_r \times N_r}$, permits to select only the elements of $\widetilde{\mathcal{K}}$ explaining the N supervised data points. Since (17) is now quadratic in $\tilde{c}$, its minimization can be carried out analytically and the minimizer is found by solving the linear system:

$$\left[P \cdot \widetilde{\mathcal{K}} + \lambda_T \cdot I_{N_r} + \lambda_M \cdot L \cdot \widetilde{\mathcal{K}}\right] \cdot \widehat{\tilde{c}} = \mathcal{Y}, \qquad (18)$$

where ψ includes the hyperparameters, which - in the case of (15) - are $\psi = [\lambda_T, \lambda_M, \eta, \delta]$.

The role of additional data can be clearly seen in (18). In fact, the *unsupervised points contribute here to the overall estimated function* via the matrix $\widetilde{\mathcal{K}}$.

5.2. A criterion for data augmentation

In dynamical system identification, unlike many static semi-supervised learning applications, the unsupervised data set $\mathcal{D}_U$ should better be seen as a *design parameter*, rather than an input of the problem. In some cases, $\mathcal{D}_U$ may contain some input time series which are likely to excite the system dynamics in future operating conditions (when the model will be used). Alternatively, $\mathcal{D}_U$ could be chosen to enforce Assumption 1 to be true. Since Assumption 1 requires only that, inside the same high density region, the regressors have a similar corresponding output (namely their difference must be "small"), a reasonable method is to generate the unsupervised regressors in the neighborhood of the supervised ones, where, if the system is smooth enough, they should have a similar corresponding output. This approach will generate a regressors' set looking as the one exemplified in Figure 2, where it is possible to list N_S regions, containing a supervised regressor and some unsupervised ones.

A possible algorithm to select $\mathcal{D}_U$ as discussed above is as follows. Let $\mathcal{D}_U$ be composed of p unsupervised datasets $\mathcal{D}_U^i$, $i = 1, \ldots, p$ as $\mathcal{D}_U^i = \left\{u_U^i(t) \;\middle|\; 1 \le t \le N_S\right\}$, where $u_U^i(t) = u_S(t) + v^i(t)$, $v^i(t)$ is a random variable and p is a free parameter of the method. Each one of the p new (unsupervised) datasets contains exactly N_S unsupervised input regressors, see again Figure 2. From such p datasets, it is possible to determine the quantities defined in Section 3. Since the unsupervised points are generated in correspondence of the supervised ones, we have N employable unsupervised regressors for each of the p datasets. This leads to $N_{rU} = p \cdot N$ unsupervised regressors $\varphi_U^i(t) \in \mathbb{R}^{m\times 1}$, $i = 1, \ldots, p$. Each one of them is such that, according to (7), for $m \le t \le N_S - 1$:

$$\varphi_U^i(t) = \begin{bmatrix} u_U^i(t) & \cdots & u_U^i(t-m+1) \end{bmatrix}^T. \qquad (19)$$

The value of $v^i(t)$ determines the distance of the p unsupervised points from the supervised one. Therefore, $v^i(t)$ has to be small enough to guarantee that the system output does not vary significantly inside these regions. A reasonable criterion for its selection is to consider that the regions should not mix with each other, since this might lead to non-smooth functions. A possible way is to use a uniform distribution:

$$v^i(t) \sim \mathrm{U}(-h, h), \quad 1 \le t \le N_S, \;\; i = 1, \ldots, p \qquad (20)$$

where $h > 0$ determines the area of the unsupervised points regions. To impose distinct regions, the following inequalities must hold:

$$\left\|\varphi_U^i(t) - \varphi_S(t)\right\|_2 \le \frac{d}{2}, \qquad \begin{matrix} m \le t \le N_S - 1, \\ i = 1, \ldots, p, \end{matrix} \qquad (21)$$

where d denotes the Euclidean distance between the two closest supervised regressors. After some computations, it can be shown that (21) can be written as:

$$\sum_{j=1}^{m} \left(v^i(t-j+1)\right)^2 \le \left(\frac{d}{2}\right)^2 \qquad \begin{matrix} m \le t \le N_S - 1 \\ i = 1, \ldots, p \end{matrix} \qquad (22)$$

Since $\left|v^i(t-j+1)\right| \le h$ (it is generated from the random variable (20)), the inequalities (22) hold if $\sum_{j=1}^{m} h^2 \le \left(\frac{d}{2}\right)^2$. Recalling that $h \ge 0$, this corresponds to $h \le \frac{d}{2\sqrt{m}}$. This condition imposes a constraint for h to maintain N_S distinct regions. To make such a constraint more or less conservative, a tuning parameter $\alpha \in \mathbb{R}$ can be introduced, allowing to regulate the region maximum area, as , e.g., as follows:

$$h = \frac{d}{2\alpha\sqrt{m}}. \qquad (23)$$

In the above criterion, $\alpha = 1$ corresponds to the threshold

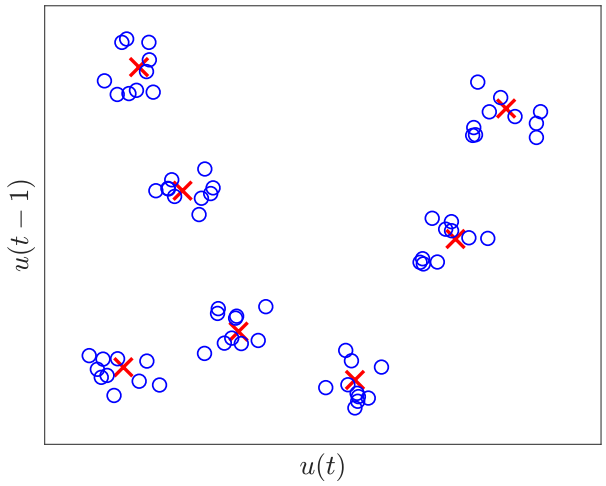

Figure 2: An example of unsupervised regressors' selection, for a system with $m = 2$ using $p = 10$. The plot represents the supervised regressors (red crosses) and the unsupervised regressors (blue circles)

between mixed regions (achieved using $\alpha < 1$) and completely distinct regions ($\alpha > 1$).

Remark. The regressors $\varphi_U^i(t)$ may improve the quality of the supervised estimate only if they lie on the same manifold spanned by the $\varphi_S(t)$. This is indeed not difficult to obtain. Suppose that the input signal $u_S(t)$ is a zero-mean white noise with variance of γ^2, i.e. $u_S(t) \sim WN(0, \gamma^2)$. We have that the regressors $\varphi_S(t)$ are composed by lagged version of the white noise $u_S(t)$. Now, assume that $u_U^i(t) = u_S(t) + v^i(t)$, with $u_S(t) \perp v^i(s)\ \forall t, s, i$, and $v^i(t) \perp v^i(s)\ \forall t \neq s$. Then, it follows that $u_U^i(t) \sim WN(0, \widetilde{\gamma}^2)$, with $\widetilde{\gamma}^2 = \gamma^2 + 4h^2/12$. Therefore, $\varphi_U^i(t)$ will span the same manifold of $\varphi_S^i(t)$, but, since the underlying process has greater variance, the additional regressors will cover a greater area of the regressors' manifold. Thus, the use of additional regressors is useful to better approximate the manifold. The same reasoning applies when $u_S(t)$ is a stationary zero-mean stochastic process and the independence assumptions hold.

5.3. Estimating hyperparameters and model order

In [4], no explicit guidelines for hyperparameters tuning is given. In this work, the hyperparameters vector ψ is estimated via Generalized Cross Validation (GCV) [12], by relying on the available data. This formulation computes an approximation of the Leave One Out Cross-Validation (LOOCV) score in the following way. Recall that, in Tikhonov-regularized estimation, the model prediction $\widehat{Y} \in \mathbb{R}^{N\times 1}$ can be computed by referring to (12) and (14) as $\widehat{Y} = \mathcal{K}\cdot\widehat{c} = S_\psi \cdot Y$, where S_ψ is given from the expression of $\widehat{c}$. In the case of the semi-supervised approach, the prediction $\widehat{Y} \in \mathbb{R}^{N\times 1}$ can be cast by referring to (16) and (18) as $\widehat{Y} = \widetilde{P}\widetilde{\mathcal{K}}\cdot\widehat{\widetilde{c}} = \widetilde{S}_\psi \cdot Y$, where $\widetilde{P} = \begin{bmatrix} I_N & 0 \end{bmatrix} \in \mathbb{R}^{N\times N_r}$ is used to select only the supervised components, and $\widetilde{S}_\psi$ comes from the expression of $\widehat{\widetilde{c}}$. Following [12], the number of effective degrees of freedom of a linear smoother, as in our case, can be found as:

$$\nu(\psi) = \mathrm{Tr}\left(\bar{S}_\psi\right), \qquad \bar{S}_\psi = \left\{S_\psi, \widetilde{S}_\psi\right\}. \tag{24}$$

The quantity in (24) can be used to efficiently compute the GCV score. The hyperparameters estimate is then computed as:

$$\begin{aligned}\widehat{\psi}_m &= \arg\min_\psi \frac{1}{N}\sum_{t=1}^{N}\left(\frac{y(t)-\widehat{y}(t)}{1-\nu(\psi)/N}\right)^2, \\ &= \arg\min_\psi J_m(\psi),\end{aligned} \tag{25}$$

where y and $\widehat{y}$ are the observed output and prediction at a specific time instant t. The subscript m on $J_m(\psi)$ and $\widehat{\psi}_m$ is used to highlight the dependency on the model order m. Since the model order is a discrete variable, the optimization becomes hybrid. For this reason, it is estimated as described in [22]. Specifically, the estimated order $\widehat{m}$ is obtained by computing $J_m(\psi)$ for a grid of chosen order values, such that:

$$\widehat{m} = \arg\min_m J_m(\widehat{\psi}_m). \tag{26}$$

In light of the same rationale, we fixed the value of p (the number of additional datasets) in our simulations.

5.4. Graph topology selection

The method presented in Section 5.1 is strongly related to the well-known approach for manifold regularization in [4]. In such a paper, it was implicitly assumed that *all* the regressors are connected. In this work, instead, the role of the dynamic dependency among the regressors can be *explicitly* taken into consideration to determine the most suitable *structure of the graph* describing the manifold[3].

To this end, firstly we need to distinguish between[4]:

1. *Spatial connections*: among different regressors in the regressor space, they are used to constrain the outputs corresponding to close regressors to be similar;
2. *Temporal connections*: among different time samples of $g(\varphi(t))$, they are used to constrain the time trajectories to be smooth.

Following the above distinction, we connect each additional regressor $\varphi_U^i(t)$ to its "parent" $\varphi_S(t)$, and each $\varphi_U^i(t)$ to its "brothers" $\varphi_U^j(t)$, $j \neq i$, for every time instant t. The output that corresponds to the unsupervised regressors $\varphi_U^i(t)$ is forced to be "close" to the output of the supervised regressor $\varphi_S(t)$ from which they are generated. Consider now the time dimension and assume that the input $u_S(t)$ of the considered NFIR system is a zero-mean white noise signal. Then, each regressor $\varphi_S(t)$ is correlated to the $m-1$ regressors $\varphi_S(t+1), \ldots, \varphi_S(t+m-1)$, as well as to the $m-1$ regressors $\varphi_S(t-1), \ldots, \varphi_S(t-m+1)$. Thus, we also need to connect the supervised regressors at different time instants according to the system memory (i.e. model order).

Figure 3 shows an example of how regressors can be connected according to the proposed approach (considering both spatial and temporal connections).

Remark. It is worth to point out that the proposed rationale is only *one* possible scheme for connecting the regressors. One may also connect the unsupervised regressors *at different time instants*, e.g. $\varphi_U^i(t)$ with $\varphi_U^i(t-1)$ and $\varphi_U^i(t+1)$ in Figure 3. However, these additional links in the regressors graph may impose a too strong condition on the set of possible functions to

[3] Recall that (10) penalizes the variations of the unknown function among the connected nodes (i.e., the regressors), thus the choice of the graph topology plays a key role to enforce smoothness.

[4] In the case of static systems, only spatial connections are meaningful, in that there is no time shift (nor correlation) among the regressors and the outputs.

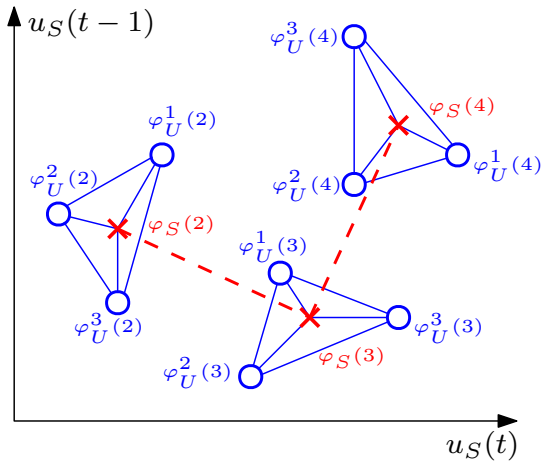

Figure 3: Example of connections in the regressor space setting the structure of the graph, with $m = 2$, $p = 3$ and $N_S = 3$: temporal connections (dashed red), spatial connections (solid blue).

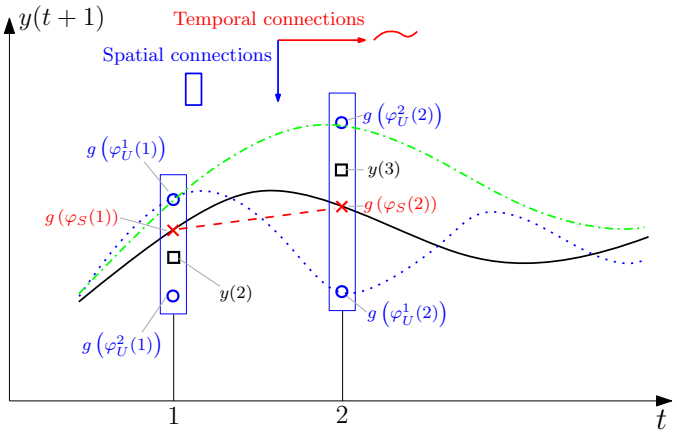

Figure 4: Representation of spatial and temporal connections in the time domain: true output (black bold line), measured output (black squares), output at supervised regressors (red crosses), output at unsupervised regressors (blue circles), possible output trajectory in case of temporal connections among supervised regressors (blue dotted line) and possible trajectory in case of temporal connections among both supervised and unsupervised regressors (green dash-dotted line).

be learnt. In fact, consider Figure 4, where the solid line represents the true output, while the measurements are denoted by $y(t)$. Since each unsupervised regressor $\varphi_U^i(t)$ is connected to its supervised "parent" $\varphi_S(t)$, their outputs are constrained to be similar, i.e. $g\left(\varphi_S(t)\right) \approx g\left(\varphi_U^i(t)\right)$. Temporal connections between $\varphi_S(t)$, $\varphi_S(t-1)$ and $\varphi_S(t+1)$ can also be imposed to constrain the output of the function g to be smooth in time. However, since the unsupervised regressors $\varphi_U^i(t)$ are generated by *randomly perturbing* the input sequence $u_S(t)$ (see again Section 5.2), temporal dependence may be partially lost, e.g., an admissible output behaviour could turn out to be the dotted blue curve of Figure 4 (which varies more rapidly than the observed one). Therefore, the output at $g\left(\varphi_U^i(1)\right)$ and $g\left(\varphi_U^i(2)\right)$ should *not* be required to be smooth in time, but only to be similar to $g\left(\varphi_S(1)\right)$ and $g\left(\varphi_S(2)\right)$, respectively. Connecting $\varphi_U^i(t)$ at different time instants may instead lead to the dash-dotted green curve of Figure 4, which could be not acceptable, unless additional prior knowledge on the output dynamics is available.

6. A discussion on global and local regularization

We now discuss the different impact of the two regularization terms in (15). Consider a static unknown function $g(x)$ that presents a discontinuity point at $x = 0$, and let the employed kernel be the Gaussian kernel

$$K\big(\varphi(t), \varphi(s)\big) = \exp\left(-\frac{\|\varphi(t) - \varphi(s)\|^2}{\sigma^2}\right),$$

where $\sigma > 0 \in \mathbb{R}$ regulates the Gaussian dispersion. In this section, we will use simple examples to show that the Tikhonov term enforces a global smooth behaviour, while the manifold term strives for local smoothness corresponding to the additional points. Figure 5 shows the results of a regularization network that employs only the Tikhonov regularization for different values of λ_T and σ. In this case, the unsupervised points are of no use, and

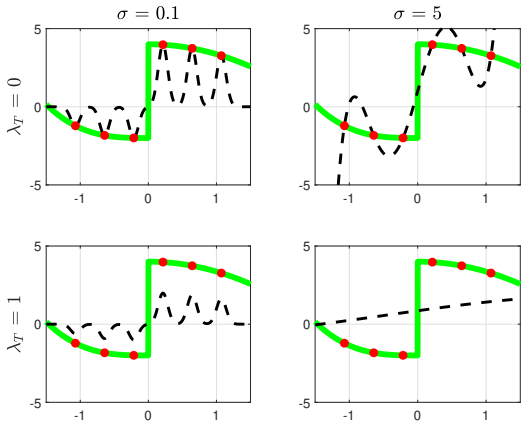

Figure 5: Sensitivity to the hyperparameters λ_T and σ when employing only Tikhonov regularization. The plots depict the true unknown function (solid green line), the supervised data (red dots) and the estimated function (dotted black line)

therefore are not depicted. When $\lambda_T = 0$, the Tikhonov term is missing, and the estimated function interpolates each one of the supervised points. Choosing a low value of σ, we are defining a function space that admits also non-smooth functions [26].

As σ grows, the estimated function gets smoother, fitting worse and worse the discontinuous region of the true underlying function. In all of these cases, given the global nature of the imposed regularization, the estimated function fails to approximate well the discontinuity region. The estimation example using only the manifold regularization term is depicted in Figure 6, where the generated unsupervised points are equally

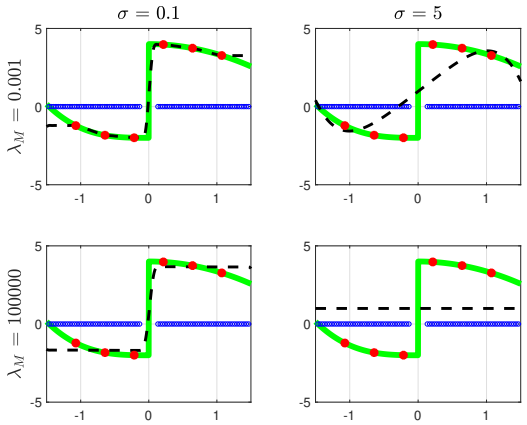

Figure 6: Sensitivity to the hyperparameters λ_M and σ when employing only manifold regularization. The plots depict the true unknown function (solid green line), the supervised data (red dots), the unsupervised data (blue dots), and the estimated function (dotted black line). The hyperparameter σ_e is fixed to 0.01.

spaced. Assumption 1 is required to hold wherever there is a regressor point. Here, we suppose that unsupervised points are *not* put only along an arbitrarily small neighbourhood of the discontinuity point. The method should then *not* regularize the model in this region, in order to allow non-smooth (rapid) variation of the estimated function, if needed, and enforce smoothness elsewhere. Notice that the desired high-variation of the function can be permitted by a suitable choice of the hyperparameter σ of the kernel. By choosing an appropriate low value of σ, it is possible to fit the function even in the discontinuity region. High values of σ or λ_M make the estimate smoother, like λ_T controlling the Tikhonov regularization term. Increasing λ_M, in turn, translates into making each domain point similar to the others, and the estimated function reduces to the mean of the supervised points when σ is sufficiently high. When σ is small, a high value of λ_M makes the resulting function similar to the mean output obtained considering the connected regressors in a smaller region, with respect to the whole function domain (see again the bottom-right plot of Figure 6). Furthermore, the lower σ, the less the impact of λ_M is on the estimated function. A careful tuning of both these hyperparameters is therefore needed for suitably tackling a specific learning problem.

Remark. The aim of the examples is to compare the effects of the two regularization terms. However, since Figure 6 implies the knowledge of the discontinuity region, it is of interest to observe what happens when we *do not have* such information. Figure 7 depicts the case where the same number of unsupervised regressors is generated, but now *equally spaced*. It is possible to observe (upper left plot) that the manifold regularization still well approximates the true function. When λ_M is high (bottom-left plot), the estimated function is no longer the mean of the connected regions, but assumes a value towards the mean of all the points. Figure 8 shows the case where the regressors are sampled *randomly from a uniform distribution* $\mathrm{U}(-1.5, 1.5)$. Only the results with low λ_M are reported for brevity, since with higher λ_M the estimated function is the same of Figure 7. Even in this case, with a proper tuning of σ, the manifold regularization achieves a good approximation of the true function (if the added points form a dense region). The reason is that local regularization generally yields more freedom (i.e., less constraints) in the choice of the function.

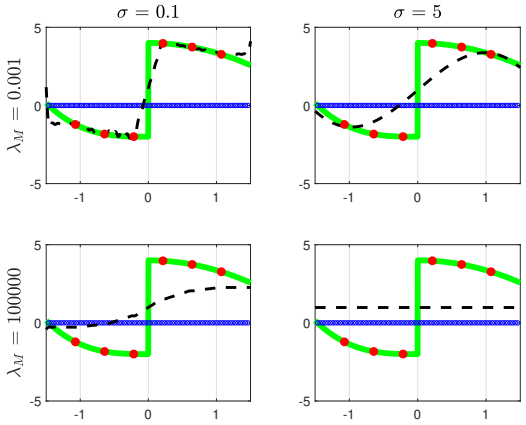

Figure 7: Manifold regularization estimates, when equally spaced unsupervised regressors are generated. The discontinuity region is not known by the method.

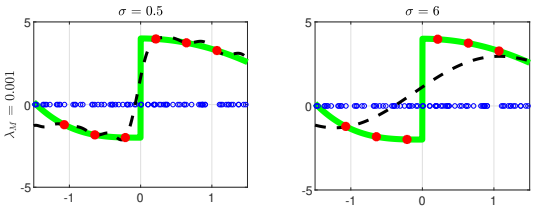

Figure 8: Manifold regularization estimates, when unsupervised regressors are randomly drawn from a uniform distribution. The discontinuity region is not known by the method.

7. A numerical case study

We test the presented methodologies on the following NFIR system taken from [22]

$$\begin{aligned} y(t) = {} & u(t-1) + 0.6u(t-2) + 0.35u(t-3) + 0.9u(t-4) + \\ & + 0.35u(t-5) + 0.2u(t-6) + 0.2u(t-7) + \\ & + 0.5u^2(t-1) - 0.25u^2(t-4) + 0.75u^3(t-3) + \\ & + 0.25u(t-1)u(t-2) + 0.5u(t-1) \cdot u(t-3) + \\ & - u(t-2)u(t-3) + 0.5u(t-2)u(t-4) + e(t), \end{aligned} \tag{27}$$

where $e(t) \sim \text{WGN}(0, 0.2)$ and $u(t) \sim \text{WGN}(0, 1)$. We employ the Gaussian kernel

$$K\big(\varphi(t), \varphi(s)\big) = \xi \cdot \exp\left(-\frac{\|\varphi(t) - \varphi(s)\|^2}{\sigma^2}\right),$$

where $\eta = [\sigma, \xi] > 0$ are the kernel hyperparameters (see [18] for a discussion about the BIBO-stability properties induced by Gaussian kernels). In particular, the following approaches are compared:

1. **Tikhonov regression**, as in (11), $\psi = [\lambda_T, \eta]$, $\eta = [\sigma, \xi]$;
2. **The approach of [4]**, where the hyperparameters are estimated via a *grid search* strategy using a part of the data set for validation. The final model is estimated using the optimal hyperparameters and all the available data. We assume that *we know the true model order*;
3. **The Kernel-based approach of [22]**;
4. **The proposed approach**, as in (15), where $\psi = [\lambda_T, \lambda_M, \eta, \delta]$, $\eta = [\sigma, \xi]$ and $\delta = [\sigma_e, \alpha]$.

The hyperparameter p, that governs how many unsupervised datasets to generate, is fixed to $p = 3$. The SNR was set to 5dB. In order to assess the overall performance of the estimation methods, a supervised testing dataset $\mathcal{D}_T$ of $N_T = 10000$ points is employed, generated analogously to $\mathcal{D}_S$. Using $\mathcal{D}_T$, it is possible to evaluate the NMAE (Normalized Mean Absolute Error) metric:

$$NMAE = \frac{\sum_{t=1}^{N_T} \left|\widehat{y}(t) - y_T(t)\right|}{\sum_{t=1}^{N_T} \left|y_T(t) - \bar{y}_T\right|}, \tag{28}$$

where $\widehat{y}(t)$ is the predicted test output in correspondence of a test regressor, $y_T(t)$ is the true test output, and $\bar{y}_T$ is the mean value of the test outputs. A Monte Carlo simulation is carried out to show the statistical significance of the proposed methodology, using 1000 runs. At each run, a different generation of the random noise was considered. The hyperparameters of the proposed method were estimated on the training set via GCV. The experimental setup problem is highly challenging: in fact, only $N_S = 30$ supervised data are available for training. The hyperparameters of the first and third approach are estimated via marginal likelihood optimization [21, 22], according to the original formulations of the methods. For the second approach, we used $N_V = 10$ data for validation (drawn from the original dataset). Once the hyperparameters are estimated, the model is identified on all the available data.

Figure 9 shows the simulation results over all the Monte Carlo runs. In this critical example, the proposed approach statistically outperforms all the state of the art methods, thus showing the effectiveness of the approach in the considered setting.[5]

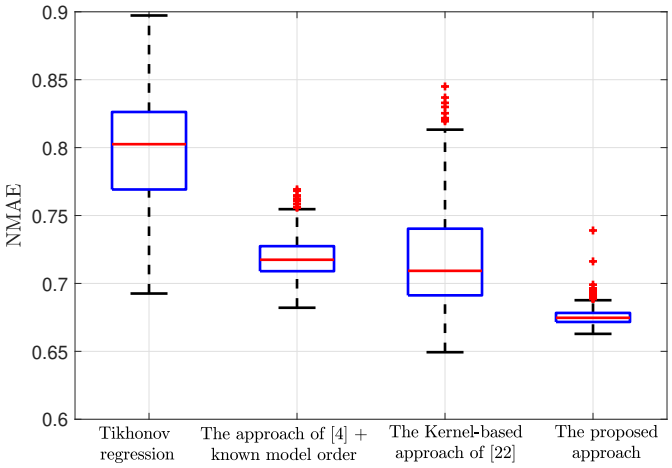

Figure 9: A numerical comparison of the proposed approach with the state of the art methods.

8. Conclusions

In this paper, we presented a method for learning nonlinear dynamical system by employing augmented datasets. The additional data are generated by perturbing the measured regressors. In order to leverage such information, manifold regularization is employed, which uses additional information on the distribution of the input regressors. The dynamical structure of the NFIR systems has been taken into consideration to best select the graph connections. Numerical results showed that the proposed approach may outperform the state of the art methods. Future research will be devoted to: (i) an extensive numerical assessment of the method; (ii) the extension of the approach to models with auto-regressive terms; (iii) the development of a data-driven graph topology selection policy.

References

[1] Y.S. Abu-Mostafa. Learning from hints. *Journal of Complexity*, 10(1):165–178, 1994.

[2] E. Bai. Identification of nonlinear additive fir systems. *Automatica*, 41(7):1247 – 1253, 2005.

[3] M. Belkin. *Problems of Learning on Manifolds*. PhD thesis, 2003. AAI3097083.

[4] M. Belkin, P. Niyogi, and V. Sindhwani. Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *Journal of machine learning research*, 7:2399–2434, 2006.

[5] V. Castelli and T.M. Cover. The relative value of labeled and unlabeled samples in pattern recognition with an unknown mixing parameter. *IEEE Transactions on information theory*, 42(6):2102–2117, 1996.

[5] For the sake of space, we do not discuss here in detail how the full graph connection discussed in the remark of Section 5.4 would work. Anyway, it is worth mentioning here that the results turn out to be less satisfactory due to the additional (too strict) time smoothness constraints.

[6] L. Cayton. Algorithms for manifold learning. *Univ. of California at San Diego Tech. Rep*, 12:1–17, 2005.
[7] O. Chapelle, B. Schlkopf, and A. Zien. *Semi-Supervised Learning*. The MIT Press, 2010.
[8] O. Chapelle, J. Weston, L. Bottou, and V. Vapnik. Vicinal risk minimization. pages 416–422, 2001.
[9] C. Cobelli, C. Dalla Man, G. Sparacino, L. Magni, G. De Nicolao, and B.P. Kovatchev. Diabetes: models, signals, and control. *IEEE reviews in biomedical engineering*, 2:54–96, 2009.
[10] M.A.H. Darwish, J. Lataire, and R. Tóth. Bayesian frequency domain identification of lti systems with obfs kernels. *20th IFAC World Congress, Toulouse*, 50(1):6238–6243, 2017.
[11] T. Evgeniou, M. Pontil, and T. Poggio. Regularization networks and support vector machines. *Advances in computational mathematics*, 13(1):1–50, 2000.
[12] J. Friedman, T. Hastie, and R. Tibshirani. *The elements of statistical learning*, volume 1. Springer series in statistics New York, 2001.
[13] G. Kimeldorf and G. Wahba. Some results on tchebycheffian spline functions. *Journal of mathematical analysis and applications*, 33(1):82–95, 1971.
[14] A. Krizhevsky, I. Sutskever, and G.E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012.
[15] L. Ljung. *System Identification: Theory for the User*. Prentice-Hall, Inc., 1986.
[16] M. Mazzoleni, M. Scandella, S. Formentin, and Fabio Previdi. Semi-supervised learning of dynamical systems: a preliminary study. 218. 17th IEEE European Control Conference (ECC 2018), Lymassol, Cyprus, 12 - 15 June.
[17] H. Ohlsson, J. Roll, and L. Ljung. Manifold-constrained regressors in system identification. In *2008 47th IEEE Conference on Decision and Control*. Institute of Electrical and Electronics Engineers (IEEE), 2008.
[18] G. Pillonetto. System identification using kernel-based regularization: New insights on stability and consistency issues. *Automatica*, 93:321 – 332, 2018.
[19] G. Pillonetto, A. Chiuso, and G. De Nicolao. Prediction error identification of linear systems: a nonparametric gaussian regression approach. *Automatica*, 47(2):291–305, 2011.
[20] G. Pillonetto and G. De Nicolao. A new kernel-based approach for linear system identification. *Automatica*, 46(1):81–93, 2010.
[21] G. Pillonetto, F. Dinuzzo, T. Chen, G. De Nicolao, and L. Ljung. Kernel methods in system identification, machine learning and function estimation: A survey. *Automatica*, 50(3):657–682, 2014.
[22] G. Pillonetto, M.H. Quang, and A. Chiuso. A new kernel-based approach for nonlinear system identification. *IEEE Transactions on Automatic Control*, 56(12):2825–2840, 2011.
[23] T. Poggio and F. Girosi. Networks for approximation and learning. *Proceedings of the IEEE*, 78(9):1481–1497, 1990.
[24] D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst. The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains. *IEEE Signal Processing Magazine*, 30(3):83–98, May 2013.
[25] V. Vapnik. *Statistical learning theory*, volume 1. Wiley New York, 1998.
[26] R. Vert and J. Vert. Consistency and convergence rates of one-class svms and related algorithms. *Journal of Machine Learning Research*, 7(May):817–854, 2006.
[27] J. Yang, H. Wei, V. Kadirkamanathan, and X. Lin. System identification from small data sets using an output jittering method with application to model estimation of bioethanol production. In *Machine Learning and Cybernetics (ICMLC), 2012 International Conference on*, volume 3, pages 949–955. IEEE, 2012.