Technical report from Automatic Control at Linköpings universitet

Manifold-Constrained Regressors in System Identification

Henrik Ohlsson, Jacob Roll, Lennart Ljung

Division of Automatic Control

E-mail: ohlsson@isy.liu.se, roll@isy.liu.se, ljung@isy.liu.se

12th September 2008

Report no.: LiTH-ISY-R-2859

Accepted for publication in the 47th IEEE Conference on Decision and Control

Address:
Department of Electrical Engineering
Linköpings universitet
SE-581 83 Linköping, Sweden

WWW: http://www.control.isy.liu.se

AUTOMATIC CONTROL
REGLERTEKNIK
LINKÖPINGS UNIVERSITET

Abstract

High-dimensional regression problems are becoming more and more common with emerging technologies. However, in many cases data are constrained to a low dimensional manifold. The information about the output is hence contained in a much lower dimensional space, which can be expressed by an intrinsic description. By first finding the intrinsic description, a low dimensional mapping can be found to give us a two step mapping from regressors to output. In this paper a methodology aimed at manifold-constrained identification problems is proposed. A supervised and a semi-supervised method are presented, where the later makes use of given regressor data lacking associated output values for learning the manifold. As it turns out, the presented methods also carry some interesting properties also when no dimensional reduction is performed.

Keywords: Manifold, System identification

Manifold-Constrained Regressors in System Identification

Henrik Ohlsson, Jacob Roll and Lennart Ljung
Division of Automatic Control, Department of Electrical Engineering, Linköping University, Sweden
{ohlsson, roll, ljung}@isy.liu.se

***Abstract*— High-dimensional regression problems are becoming more and more common with emerging technologies. However, in many cases data are constrained to a low dimensional manifold. The information about the output is hence contained in a much lower dimensional space, which can be expressed by an intrinsic description. By first finding the intrinsic description, a low dimensional mapping can be found to give us a two step mapping from regressors to output. In this paper a methodology aimed at manifold-constrained identification problems is proposed. A supervised and a semi-supervised method are presented, where the later makes use of given regressor data lacking associated output values for learning the manifold. As it turns out, the presented methods also carry some interesting properties also when no dimensional reduction is performed.**

I. Introduction

With new applications emerging, for instance within medicine and systems biology, system identification and regression using high-dimensional data has become an interesting field. A central topic in this context is dimension reduction.

Sometimes, the system itself is such that the data are implicitly constrained to a lower-dimensional manifold, embedded in the higher dimension. In such cases, some regression algorithms do not suffer of the high dimensionality of the regressors. However it is common that regression algorithms assume that the underlying system behaves smoothly. For manifold-constrained systems this is commonly a restriction. A less conservative condition is the semi-supervised smoothness assumption [3]. Using the semi-supervised smoothness assumption the underlying system is assumed to behave smoothly along the manifold but not necessary from one part of the manifold to another, even though they are close in Euclidean distance. The semi-supervised smoothness assumption motivates the computation and use of an intrinsic description of the manifold as regressors and not the original regressors. Finding the intrinsic description is a manifold learning problem [15], [13], [1].

The resulting method is a two-step approach, where in the first step an intrinsic description of the low-dimensional manifold is found. Using this description as new regressors, we apply a regression in a second step in order to find a function mapping the new regressors to the output (see Figure 1).

This strategy for regression with manifold-constrained data was previously discussed in [12]. However, since an unsupervised manifold learning approach was used to find the intrinsic description, no guarantee could be given that the new low-dimensional regressors would give an easy identification problem. For instance, a high-dimensional linear problem could be transformed into a low-dimensional nonlinear problem.

To overcome this problem, the manifold learning step can be modified to take into account the fact that the intrinsic description in the next step will be used as regressors in an identification problem. In this paper we have chosen to extend a nonlinear manifold learning technique, *Locally Linear Embedding* (LLE) [13]. LLE finds a coordinatization of the manifold by solving two optimization problems. By extending one of the objective functions with a term that penalizes any deviation from a given functional relation between the intrinsic coordinates and the output data, we can stretch and compress the intrinsic description space in order to give an as easy as possible mapping between the new regressors, the intrinsic description, and the output. Also, as the regressors, in themselves, contain information about the manifold they are constrained to, all regressors at hand can be used to find the intrinsic description. To that end, both a supervised and a semi-supervised extension of LLE will be proposed.

As it turns out, the idea of stretching and compressing the regressor space can be useful, not only for dimension reduction purposes, but also for nonlinear system identification problems where no dimensional reduction is performed. In this way, we can move the nonlinearities from the identification problem to the problem of remapping the regressor space, and thus simplifying the identification step.

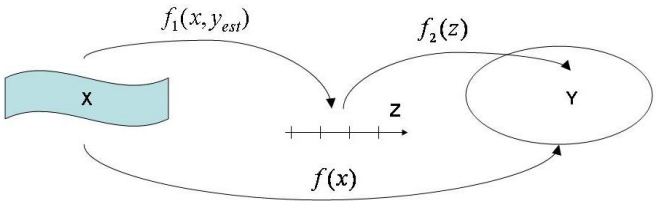

Fig. 1. Overview of the identification steps for a system having regressors constrained to a manifold. X is the regressor space with regressor data constrained to some low-dimensional manifold. Z is a space, with the same dimension as the manifold, containing the intrinsic description of the manifold-constrained regressor data. Y is the output space. Common identification schemes try to find the function $f : X \to Y$ by using the original regressors. However, the same information about the outputs can be obtained from the low-dimensional regressor space Z. With a wise intrinsic description, the low-dimensional function f_2 will be considerably easier to find than f.

Manifold learning algorithms have previously been used for classification, see for example [19]. An extension to the *Support Vector Machines* (SVM) to handle regressors constrained to manifolds has also been developed [2]. For

regression, dimension reduction has been used to find low-dimensional descriptions of data see [9], [10], [4], [11]. However, not so much has been done concerning regression with regressors constrained to manifolds. An extension of the manifold-adjusted SVM classification framework to regression is presented as well in [2]. Related ideas to the ones presented in this paper have also independently been developed by [17].

The paper is organized as follows: The problem is motivated and stated in Sections II and III, respectively. LLE is presented in Section IV and extended in Sections V and VI. The extensions are exemplified and compared to various regression methods in Section VII. We finish with conclusions in Section VIII.

II. Manifold-Constrained Data

Data constrained to manifolds often appear in areas such as medicine and biology, signal processing and image processing *etc.* Data are typically high-dimensional with static constraints giving relations between certain dimensions.

A specific example could be high-dimensional data coming from a *functional Magnetic Resonance Imaging* (fMRI) scan [14], [16], [7]. For instance, suppose that the brain activity in the visual cortex is measured using an MRI scanner. The activity is given as a $80 \times 80 \times 22$ array, each element giving a measure of the activity in a small volume (voxel) of the brain at a specific time. Furthermore, suppose that we would like to estimate in what direction a person is looking. Since this direction can be described using only one parameter, the measurements should (assuming that we can preprocess the data and get rid of most of the noise) be constrained to some manifold. For further discussions on fMRI data and manifolds, see [14], [16], [7].

Another example of data constrained to a manifold is images of faces [18]. An image can be seen as a high-dimensional point (every pixel becomes a dimension) and because every face has a nose, two eyes *etc.* the faces, or points, will be constrained to some manifold.

There is also a connection to *Differential Algebraic Equations*, DAEs [8]. In DAEs, systems are described by a combination of differential equations and algebraic constraints. Due to the latter constraints, the variables of a system governed by a DAE will naturally be forced to belong to a manifold.

III. Problem Formulation

Let us assume that we are given a set of estimation data $\{y_{\text{est},t}, x_{\text{est},t}\}_{t=1}^{N_{\text{est}}}$ generated from

$$y_t = f_0(x_t) + e_t,$$

where f_0 is a smooth unknown function, $f_0 : \mathbb{R}^{n_x} \to \mathbb{R}^{n_y}$, and e_t is i.i.d. white noise. Let $x_{\text{est},t}$ be constrained to some n_z-dimensional manifold defined by

$$g(x_t) = 0, \quad \forall t, g : \mathbb{R}^{n_x} \to \mathbb{R}^{n_x - n_z}. \tag{1}$$

Given a new set of regression vectors $x_{\text{pre},t}, t = 1, \ldots, N_{\text{pre}}$, satisfying the constraint, what would be the best way to predict the associated output values? We will in the following use subindex:

- "est" for data for which both regressors and associated outputs are known.
- "pre" for data with unknown outputs whose values should be predicted.

To facilitate, we use the notation $\mathbf{x} = [x_1, \ldots, x_N]$, for a matrix with the vectors x_i as columns and x_{ji} for the jth element in x_i. Throughout the paper it will also be assumed that the dimension of the manifold given by (1) is known. Choosing the dimension can be seen as a model structure selection problem, similar to e.g. model order selection.

IV. Locally Linear Embedding

For finding intrinsic descriptions of data on a manifold, we will use the manifold learning technique *Locally Linear Embedding* (LLE) [13]. LLE is a manifold learning technique which aims at preserving neighbors. In other words, given a set of points $\{x_1, \ldots, x_N\}$ residing on some n_z-dimensional manifold in $\mathbb{R}^{n_x}$, LLE aims to find a new set of coordinates $\{z_1, \ldots, z_N\}$, $z_i \in \mathbb{R}^{n_z}$, satisfying the same neighbor-relations as the original points. The LLE algorithm can be divided into two-steps:

Step 1: Define the w_{ij}:s – the regressor coordinatization

Given data consisting of N real-valued vectors x_i of dimension n_x, the first step minimizes the cost function

$$\varepsilon(\mathbf{w}) = \sum_{i=1}^{N} \left\| x_i - \sum_{j=1}^{N} w_{ij} x_j \right\|^2 \tag{2a}$$

under the constraints

$$\begin{cases} \sum_{j=1}^{N} w_{ij} = 1, \\ w_{ij} = 0 \text{ if } \|x_i - x_j\| > C_i(K) \text{ or if } i = j. \end{cases} \tag{2b}$$

Here, $C_i(K)$ is chosen so that only K weights w_{ij} become nonzero for every i. In the basic formulation of LLE, the number K and the choice of lower dimension $n_z \leq n_x$ are the only design parameters, but it is also common to add a regularization

$$F_r(\mathbf{w}) \triangleq \frac{r}{K} \sum_{i=1}^{N} [w_{i1}, \ldots, w_{iN}] \begin{bmatrix} w_{i1} \\ \vdots \\ w_{iN} \end{bmatrix} \sum_{j: w_{ij} \neq 0}^{N} ||x_j - x_i||^2$$

to (2a), see [13].

Step 2: Define the z_{ij}:s – the manifold coordinatization

In the second step, let z_i be of dimension n_z and minimize

$$\Phi(\mathbf{z}) = \sum_{i=1}^{N} \left\| z_i - \sum_{j=1}^{N} w_{ij} z_j \right\|^2 \tag{3a}$$

with respect to $\mathbf{z} = [z_1, \ldots, z_N]$, and subject to

$$\frac{1}{N} \sum_{i=1}^{N} z_i z_i^T = \mathbf{I} \tag{3b}$$

using the weights w_{ij} computed in the first step. The solution $\mathbf{z}$ to this optimization problem is the desired set of low-dimensional coordinates which will work as an intrinsic

description of the manifold for us. By expanding the squares we can rewrite $\Phi(\mathbf{z})$ as

$$\Phi(\mathbf{z}) = \sum_{i,j}^{N}(\delta_{ij} - w_{ij} - w_{ji} + \sum_{l}^{N} w_{li}w_{lj})z_i^T z_j$$
$$\triangleq \sum_{i,j}^{N} M_{ij} z_i^T z_j = \sum_{k}^{n_z}\sum_{i,j}^{N} M_{ij} z_{ki} z_{kj} = Tr(\mathbf{z}M\mathbf{z}^T)$$

with M a symmetric $N \times N$ matrix with the ijth element M_{ij}. The solution to (3) is obtained by using Rayleigh-Ritz theorem [6]. With v_i the unit length eigenvector of M associated with the ith smallest eigenvalue,

$$\left[\ v_1, \ldots, v_{n_z}\ \right]^T = \arg\min_{\mathbf{z}} \Phi(\mathbf{z}) \quad \text{s.t. } \mathbf{z}\mathbf{z}^T = N\mathbf{I}.$$

LLE is an unsupervised method that will find an intrinsic description without using any knowledge about y_t. However, since our purpose is to use the intrinsic description as new regressors, there might be better coordinatizations of the manifold, that could be found by taking observed output values into account.

V. Smoothing using Weight Determination by Manifold Regularization (WDMR)

In this section, we extend LLE by including the knowledge of observed outputs in order to get a description that will facilitate a subsequent identification step. The result will be a smoothing filter with a weighting kernel adjusted to the manifold-constrained regressors. To avoid poor intrinsic descriptions, we modify the optimization problem (3) in the second step of the LLE algorithm into

$$\min_{\mathbf{z}} \lambda Tr(\mathbf{z}M\mathbf{z}^T) + (1-\lambda)||\mathbf{y}_{\text{est}} - f_2(\mathbf{z})||_F^2 \qquad (4)$$
$$\text{subject to } \mathbf{z}\mathbf{z}^T = N_{\text{est}}\mathbf{I}.$$

Here, $||\cdot||_F$ is the Frobenius norm and f_2 is a function mapping from the intrinsic description, z_t, to the output, y_t, see Figure 1. The parameter λ is a design parameter which can be set to values between 0 and 1. $\lambda = 1$ gives the same intrinsic description as LLE and $\lambda = 0$ gives an intrinsic description satisfying $f_2(z_t) = y_t$. The function f_2 can be:

- Chosen beforehand.
- Numerically computed, by for example alternating between minimizing (4) w.r.t. z_t and f_2. However, it is unclear if optimizing over f_2 would improve the results or if there is enough flexibility with a fixed f_2.

We choose to fix $f_2(z_t) = z_t$[1] and believe that the intrinsic description will adapt to this. Using this particular choice, the constraint on z_t can be relaxed since the second term of (4) $(1-\lambda)||\mathbf{y}_{\text{est}} - f_2(\mathbf{z})||_F^2$ will keep z_t from becoming identically zero. The problem is then simplified considerably while many of the properties are still preserved.

The z_t coordinate now acts as an estimate of y_t and we therefore write

$$\hat{\mathbf{y}} = \arg\min_{\mathbf{z}} \lambda Tr(\mathbf{z}M\mathbf{z}^T) + (1-\lambda)||\mathbf{y}_{\text{est}} - f_2(\mathbf{z})||_F^2 \qquad (5)$$

[1]The n_y first components of z_t if $n_z > n_y$. In the continuation we assume $n_z = n_y$. However, expressions can be generalized to hold for $n_z > n_y$ with minor adjustments.

which can be shown to be minimized by

$$\hat{\mathbf{y}}_{\text{est}}^T = (1-\lambda)\left(\lambda M + (1-\lambda)\mathbf{I}\right)^{-1}\mathbf{y}_{\text{est}}^T.$$

$\hat{\mathbf{y}}_{\text{est}}$ becomes a smoothed version of $\mathbf{y}_{\text{est}}$. The filtering method takes into account that the output is associated with some regressors and aims to make two outputs close to each other if associated regressors are close. The design parameter λ reflects how much we rely on the measured outputs. For a $\lambda = 1$, the information in the measured output is considered worthless. Using a $\lambda = 0$, the output is thought to be noise-free and obtained as the estimate from the filter.

A nice way to look at the two-step scheme is by seeing the term $Tr(\mathbf{z}M\mathbf{z}^T)$ in (5) as a regularization (*cf.* ridge regression [5]). The regularization incorporates the notion of a manifold and makes outputs similar if their regressors are close on the manifold, well consistent with the semi-supervised smoothness assumption. Since the scheme produce a weighting-kernel defined by $(\lambda M + (1-\lambda)\mathbf{I})^{-1}$ we name the algorithm *Weight Determination by Manifold Regularization* (WDMR). We summarize the WDMR filter in Algorithm 1.

Algorithm 1 WDMR smoothing filter

Let N_{est} be the number of estimation regressors. For a chosen K, r and λ,

1) Find the weights w_{ij} minimizing

$$\sum_{i=1}^{N_{\text{est}}}\left\|x_i - \sum_{j=1}^{N_{\text{est}}} w_{ij}x_j\right\|^2 + F_r(\mathbf{w}),$$
$$\text{subject to } \begin{cases} \sum_{j=1}^{N_{\text{est}}} w_{ij} = 1, \\ w_{ij} = 0 \text{ if } |x_i - x_j| > C_i(K) \text{ or if } i = j. \end{cases}$$

2) With $M_{ij} = \delta_{ij} - w_{ij} - w_{ji} + \sum_k^{N_{\text{est}}} w_{ki}w_{kj}$ the filtered output is given by

$$\hat{\mathbf{y}}^T = (1-\lambda)\left(\lambda M + (1-\lambda)\mathbf{I}\right)^{-1}\mathbf{y}_{\text{est}}^T.$$

VI. Regression using Weight Determination by Manifold Regularization (WDMR)

In this section we examine the possibilities to extend LLE to regression. The WDMR filter is a smoothing filter and can therefore be used to reduce noise from measurements. With new regressors at hand, the filtered outputs can be utilized to find estimates of the outputs. To generalize to regressors with unknown outputs nearest neighbor or an affine combination of the closest neighbors could for example be used.

With x_t constrained to some manifold, however, also the regressors x_t themselves, regardless of knowledge of associated output y_t, contain information about the manifold. We could therefore use this information and include all regressors at hand, even though the output is unknown, when trying to find an intrinsic description. As we will see, including regressors with unknown outputs also gives us a way to generalize and compute an estimate for their outputs.

Hence we apply the first step of the LLE algorithm (2) to all regressors, both x_{est} and x_{pre}. The optimization problem (3) in the second step of the LLE algorithm takes the form

$$\min_{\mathbf{z}_{\text{est}},\mathbf{z}_{\text{pre}}} \lambda Tr([\mathbf{z}_{\text{est}}\,\mathbf{z}_{\text{pre}}]M\begin{bmatrix}\mathbf{z}_{\text{est}}^T\\ \mathbf{z}_{\text{pre}}^T\end{bmatrix})+(1-\lambda)||\mathbf{y}_{\text{est}}-f_2(\mathbf{z}_{\text{est}})||_F^2 \quad (6)$$

subject to $[\mathbf{z}_{\text{est}}\,\mathbf{z}_{\text{pre}}][\mathbf{z}_{\text{est}}\,\mathbf{z}_{\text{pre}}]^T = (N_{\text{est}}+N_{\text{pre}})\mathbf{I}$.

As for the WDMR filter, $f_2(z_t) = z_t$ is an interesting choice. Relaxing $[\mathbf{z}_{\text{est}}\,\mathbf{z}_{\text{pre}}][\mathbf{z}_{\text{est}}\,\mathbf{z}_{\text{pre}}]^T = (N_{\text{est}}+N_{\text{pre}})\mathbf{I}$ using the same motivation as in the WDMR filter, (6) has a solution

$$\begin{bmatrix}\hat{\mathbf{y}}_{\text{est}}^T\\ \hat{\mathbf{y}}_{\text{pre}}^T\end{bmatrix}=(1-\lambda)\left(\lambda M+(1-\lambda)\begin{bmatrix}\mathbf{I}_{N_{\text{est}}\times N_{\text{est}}} & 0_{N_{\text{est}}\times N_{\text{pre}}}\\ 0_{N_{\text{pre}}\times N_{\text{est}}} & 0_{N_{\text{pre}}\times N_{\text{pre}}}\end{bmatrix}\right)^{-1}\begin{bmatrix}\mathbf{y}_{\text{est}}^T\\ 0_{N_{\text{pre}}\times n_y}\end{bmatrix}.$$

Notice that we get an estimate of the unknown outputs along with the filtered estimation outputs. The algorithm is as for the WDMR filter, an algorithm for computing a weighting-kernel. The kernel account for the manifold and is well consistent with the semi-supervised smoothness assumption. We summarize the WDMR regression algorithm in Algorithm 2.

Algorithm 2 WDMR Regression

Let x_t be the tth element in $[\mathbf{x}_{\text{est}},\mathbf{x}_{\text{pre}}]$, N_{est} the number of estimation regressors and N_{pre} the number of regressors for which a prediction is searched. For a chosen K, r and λ,

1) Find the weights w_{ij} minimizing

$$\sum_{i=1}^{N_{\text{est}}+N_{\text{pre}}}\left\|x_i-\sum_{j=1}^{N_{\text{est}}+N_{\text{pre}}}w_{ij}x_j\right\|^2+F_r(\mathbf{w}),$$

$$\text{subject to } \begin{cases}\sum_{j=1}^{N_{\text{est}}+N_{\text{pre}}} w_{ij}=1,\\ w_{ij}=0 \text{ if } |x_i-x_j|>C_i(K) \text{ or if } i=j.\end{cases}$$

2) With $M_{ij}=\delta_{ij}-w_{ij}-w_{ji}+\sum_k^{N_{\text{est}}+N_{\text{pre}}} w_{ki}w_{kj}$ the estimated output is given by

$$\begin{bmatrix}\hat{\mathbf{y}}_{\text{est}}^T\\ \hat{\mathbf{y}}_{\text{pre}}^T\end{bmatrix}=(1-\lambda)\left((\lambda M)+(1-\lambda)\begin{bmatrix}\mathbf{I}_{N_{\text{est}}\times N_{\text{est}}} & 0_{N_{\text{est}}\times N_{\text{pre}}}\\ 0_{N_{\text{pre}}\times N_{\text{est}}} & 0_{N_{\text{pre}}\times N_{\text{pre}}}\end{bmatrix}\right)^{-1}\begin{bmatrix}\mathbf{y}_{\text{est}}^T\\ 0_{N_{\text{pre}}\times n_y}\end{bmatrix}.$$

VII. Examples

To illustrate the WDMR smoothing filter and regression algorithm, four examples are given. The three first examples illustrates the ability to deal with regressors on manifolds and the last example shows the algorithm without making use of the built-in dimension reduction property. Comparisons with classical identification approaches, without any dimensional reduction, and LapRLSR [2], adjusted for manifold-constrained data, are also given.

Example 1: Consider the system

$$x_{1,t} = 8v_t\cos 8v_t,$$
$$x_{2,t} = 8v_t\sin 8v_t,$$
$$y_t = \sqrt{x_{1,t}^2+x_{2,t}^2} = 8v_t.$$

Assume that the output y_t is measured with some measurement error, *i.e.*,

$$y_t^m = y_t + e_t, \quad e_t \sim \mathcal{N}(0,\sigma_e^2)$$

and that a set of regressor data is generated by the system by v-values uniformly distributed in the interval $[2,3.2]$. The regressors, $[x_{1,t}\,x_{2,t}]$, are situated on a one-dimensional manifold, a spiral. Figure 2 shows 25 regressors along with associated measured outputs. Even though the dimensionality is not an issue in this particular example, the manifold-constrained regression data makes it a suitable example.

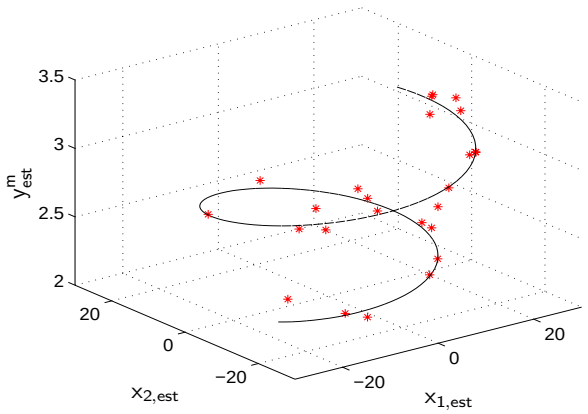

Fig. 2. Estimation data for Example 1. The measured outputs, showed with '$*$', were measured from the underlying system (dashed line) using $\sigma_e = 0.07$.

Using 25 labeled regressors (output measurements distorted using $\sigma_e = 0.07$), the WDMR framework was applied to predict the outputs of 200 validation regressors. The performance of the prediction was evaluated by computing the mean fit[2] for 50 examples, like the one just described. The result is summarized in Table I. For all 50 experiments $K = 11$, $r = 10$ and $\lambda = 0.9$. A comparison to LapRLSR [2], which also adjusts to manifolds, is also given. Figure 3 shows the prediction computed in one of the 50 runs.

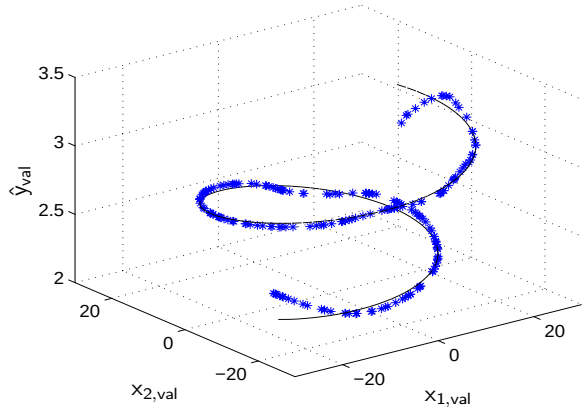

Fig. 3. Validation regressors together with predicted outputs for Example 1. The function from which the estimation data was measured is shown with a dashed line.

[2] $fit = (1-\frac{\|\mathbf{y}-\hat{\mathbf{y}}\|}{\|\mathbf{y}-\frac{1}{N}\sum_t y_t\|})\times 100$

Figure 4 shows the weighting kernel associated with a validation regressor for WDMR regression. It is nice to see how the kernel adapts to the manifold.

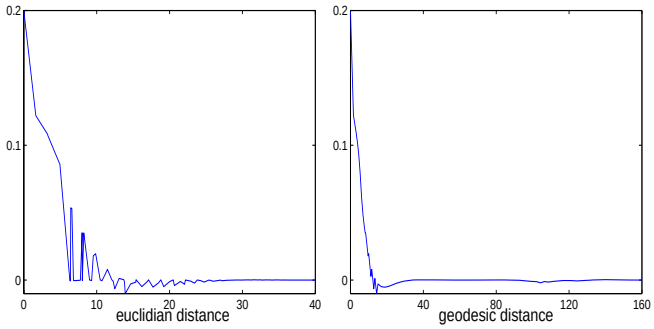

Fig. 4. Weighting kernel associated with a validation regressor used in Example 1. Left figure: Kernel as a function of Euclidean distance. Right figure: Kernel as a function of geodesic distance.

To test the smoothing properties of the WDMR framework, a WDMR smoothing filter ($K = 4$, $r = 10$ and $\lambda = 0.7$) and a Gaussian filter (weighting together 3 closest neighbors and the measurement itself) were applied to 50 labeled regressors (output measurements distorted using $\sigma_e = 0.1$). Figure 5 shows the filtered outputs. The Gaussian filter runs into problems since it weights together the 3 closest neighbors, not making use of the manifold. The WDMR filter, on the other hand, adjusts the weighting kernel to the manifold and thereby avoid to weight together measurements from different parts of the manifold.

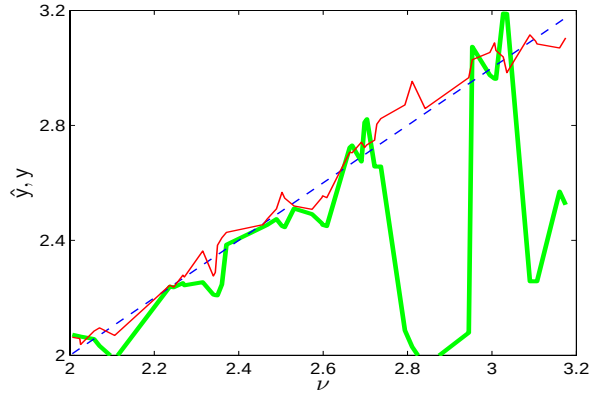

Fig. 5. Outputs filtered by WDMR and a Gaussian filter in Example 1. WDMR filter (thin solid line), Gaussian filter (thick solid line) and noise free outputs (dashed line).

Example 2: To exemplify the behavior for a high-dimensional case, the previous example was extended as follows. $x_{1,t}$ and $x_{2,t}$ from Example 1 were used to compute

$$[\tilde{x}_1,\ \tilde{x}_2,\ \tilde{x}_3,\ \tilde{x}_4,\ \tilde{x}_5,\ \tilde{x}_6] = [x_2 e^{x_1},\ x_1 e^{x_2},\ x_2 e^{-x_1},\ x_1 e^{-x_2}, \log|x_1|,\ \log|x_2|],$$

(t has been neglected for simplicity) which were used as the new regressors. Using the same estimation and validation procedure ($N_{\text{est}} = 25$, $N_{\text{val}} = 200$, $\sigma_e = 0.07$) as in Example 1, WDMR regression was applied to predict the unknown outputs of the validation regressors. The result is shown in Table I using ($K = 16$, $r = 10$, $\lambda = 0.999$).

Note that in this example the LLE algorithm reduces the dimension from six to one compared to from two to one in the previous example.

Example 3: We mentioned fMRI data as an example of manifold-constrained data in the introduction. The dimensionality and the signal-to-noise ratio make fMRI data very tedious to work with. Periodic stimulus is commonly used to be able to average out noise and find areas associated with the stimulus. However, in this example, measurements from an $8 \times 8 \times 2$ array covering parts of the visual cortex were gathered with a sampling period of 2 seconds. To remove noise, data was prefiltered by applying a spatial and temporal Gaussian filter. The subject in the scanner was instructed to look away from a flashing checkerboard covering 30% of the field of view. The flashing checkerboard moved around and caused the subject to look to the left, right, up and down. Using an estimation data set (40 time points, 128 dimensions) and a validation set of the same size, the WDMR regression algorithm was tuned ($K = 6$, $r = 10^{-6}$, $\lambda = 0.2$). The output was chosen to 0 when the subject was looking to the right, $\pi/2$ looking up, π looking to the left and $-\pi/2$ looking down. The tuned WDMR regression algorithm could then be used to predict the direction in which the subject was looking. The result from applying WDMR regression to a test data set is shown in Figure 6.

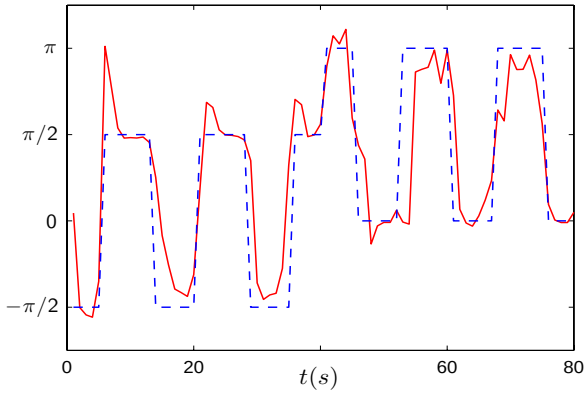

Fig. 6. WDMR regression applied to brain activity measurements (fMRI) of the visual cortex in order to tell in what direction the subject in the scanner was looking, Example 3. Dashed line shows the direction in which the subject was looking (adjusted in time for the time delay expected) and solid line, the predicted direction by WDMR regression.

Example 4: Previous examples have all included dimensional reduction. However, nothing prevents us from applying the WDMR framework to an example where no dimensional reduction is necessary. The dimensional reduction is then turned into a simple stretching and compression of the regressor space. Data was generated from

$$y_t^m = 0.08 x_t^4 + e_t \tag{7}$$

where x_t was sampled from a uniform distribution

TABLE I

RESULTS FOR EXAMPLE 1, 2 AND 4. THE MEAN FIT (BASED ON 50 EXPERIMENT) FOR WDMR REGRESSION, AFFINE COMBINATION, NARX WITH SIGMOIDNET OF DIFFERENT ORDERS (ONLY THE BEST PERFORMING NARX IS SHOWN) AND LAPRLSR.

Ex.	WDMR Regression	affine comb.	NARX	LapRLSR
1	75.7%	54.7%	57.0%	66.6%
2	80.2%	72.3%	41.9%	57.4%
4	74.5%	51.1%	74.7%	

$U(-10,10)$ and $e_t \sim \mathcal{N}(0,\sigma_e^2)$, $\sigma_e = \sqrt{30}$. Table I shows the result applying WDMR regression with $N_{\text{est}} = 10$, $K = 24$, $r = 10^6$ and $\lambda = 0.9$.

To exemplify the smoothing properties of the WDMR filter, 35 measurements were generated using (7) with $\sigma_e = \sqrt{60}$. Figure 7 shows the measured output along with the filtered version.

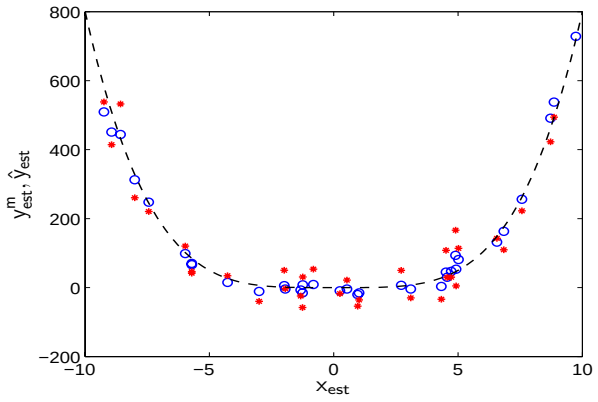

Fig. 7. Measured outputs together with filtered outputs (WDMR filter) for Example 4. $*$ marks the 35 measurements, $\circ$ marks the filtered measurements. Dashed line: the function which was used to generate the measurements.

VIII. CONCLUSIONS

The paper discusses an emerging field within system identification. High-dimensional data sets are becoming more and more common with the development of new technologies in various fields. However, data are commonly not filling up the regressors space but are constrained to some embedded manifold. Finding the intrinsic description of the regressors, this can be used as new regressors when finding the mapping between regressors and the output. Furthermore, in order to find an as good intrinsic description of the manifold as possible, we could use all regression vectors available, even if the associated output might be unknown.

Proposed is a two-step approach suitable for manifold-constrained regression problems. The first step finds an intrinsic description of the manifold-constrained regressors, and the second maps the new regressors to the output. A filter and regression version of the approach were discussed and exemplified with good results.

The approach showed promising results even without utilizing the built in dimensionality reduction property. The first step is then turned into a stretching and compression of the regressor space. This can be seen as a relocationing of the nonlinearity to the regression space.

IX. ACKNOWLEDGMENT

This work was supported by the Strategic Research Center MOVIII, funded by the Swedish Foundation for Strategic Research, SSF.

REFERENCES

[1] Mikhail Belkin and Partha Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation*, 15(6):1373–1396, 2003.

[2] Mikhail Belkin, Partha Niyogi, and Vikas Sindhwani. Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *Journal of Machine Learning Research*, 7:2399–2434, 2006.

[3] O. Chapelle, B. Schölkopf, and A. Zien, editors. *Semi-Supervised Learning (Adaptive Computation and Machine Learning)*. The MIT Press, September 2006.

[4] Kenji Fukumizu, Francis R. Bach, and Michael I. Jordan. Dimensionality reduction for supervised learning with reproducing kernel hilbert spaces. *Journal of Machine Learning Research*, 5:73–99, 2004.

[5] T. Hastie, R. Tibshirani, and J. H. Friedman. *The Elements of Statistical Learning*. Springer, August 2001.

[6] R. A. Horn and C. R. Johnson. *Matrix Analysis*. Cambridge University Press, 1990. Cambridge.

[7] Jin Hu, Jie Tian, and Lei Yang. Functional feature embedded space mapping of fMRI data. In *Conf Proc IEEE Eng Med Biol Soc.*, 2006.

[8] Peter Kunkel and Volker Mehrmann. *Differential-Algebraic Equations – Analysis and Numerical Solution*. EMS Publishing House, Zürich, 2006.

[9] K.-C. Li. Sliced inverse regression for dimension reduction. *Journal of the American Statistical Association*, 86(414):316–327, 1991.

[10] K.-C. Li. On principal hessian directions for data visualizations and dimension reduction: Another application of Stein's lemma. *Journal of the American Statistical Association*, 87(420):1025–1039, 1992.

[11] D. Lindgren. *Projection Techniques for Classification and Identification*. PhD thesis, Linköpings universitet, 2005. Dissertation no 915.

[12] Henrik Ohlsson, Jacob Roll, Torkel Glad, and Lennart Ljung. Using manifold learning in nonlinear system identification. In *Proceedings of the 7th IFAC Symposium on Nonlinear Control Systems (NOLCOS)*, Pretoria, South Africa, August 2007.

[13] S. T. Roweis and L. K. Saul. Nonlinear dimensionality reduction by local linear embedding. *Science*, 290(5500):2323–2326, 2000.

[14] Xilin Shen and François G. Meyer. *Analysis of Event-Related fMRI Data Using Diffusion Maps*, volume 3565/2005 of *Lecture Notes in Computer Science*. Springer Berlin / Heidelberg, July 2005.

[15] J. B. Tenenbaum, V. de Silva, and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, 2000.

[16] Bertrand Thirion and Olivier Faugeras. Nonlinear dimension reduction of fMRI data: The laplacian embedding approach. *IEEE International Symposium on Biomedical Imaging: Nano to Macro*, 2004.

[17] Xin Yang, Haoying Fu, Hongyuan Zha, and Jesse Barlow. Semi-supervised nonlinear dimensionality reduction. In *ICML '06: Proceedings of the 23rd international conference on Machine learning*, pages 1065–1072, New York, NY, USA, 2006. ACM.

[18] Junping Zhang, Stan Z. Li, and Jue Wang. Manifold learning and applications in recognition. *Intelligent Multimedia Processing with Soft Computing*, 2004. Springer-Verlag, Heidelberg.

[19] Qijun Zhao, D. Zhang, and Hongtao Lu. Supervised LLE in ICA space for facial expression recognition. In *Neural Networks and Brain, 2005. ICNN&B '05. International Conference on*, volume 3, pages 1970–1975, October 2005. ISBN: 0-7803-9422-4.