Approaches to Identification of Nonlinear Systems

Lennart Ljung[1]
1. Division of Automatic Control, Linköping University, SE-58183 Linköping, Sweden
E-mail: ljung@isy.liu.se

Abstract: System Identification for linear systems and models is a well established and mature topic. Identifying nonlinear models is a much more rich and demanding problem area. In this presentation some major approaches and concepts for that are outlined.
Key Words: System Identification, Nonlinear models, manifold learning, estimation, nonparametric methods, gray box, black box

1 INTRODUCTION

System Identification is about building mathematical models of dynamical systems based on observed input–output data. In case the sought model is linear, there exist well structured theory, methodology and algorithms. For the case of nonlinear models, the situation is more complex, and perhaps confusing. In particular, there are several intriguing techniques that have been developed outside the system identification research area, mostly in the machine learning community. Concepts like Manifold learning and Gaussian processes turn out to be valuable also for identifying nonlinear systems.
This presentation aims at an overview of various approaches to nonlinear system identification. This is of course not possible to do in comprehensive manner in such a brief text, but we refer for more details to [11] and the recent articles [13], [16] and [12].

2 BASIC PROBLEM FORMULATION

Given a (dynamical) system with input $u(t)$ and output $y(t)$. A (mathematical) model of the system is a mathematical relationship that relates the output to past inputs and outputs. For a system with (stochastic) disturbances only the *predicted* output $\hat{y}(t)$ can obey such an exact relationship.

$$\hat{y}(t) = \tilde{g}(y(t-1), u(t-1), y(t-2), u(t-2), \ldots) \quad (1)$$

The model is nonlinear if the function $\tilde{g}$ is nonlinear.
So, estimation of nonlinear models is very rich topic. A simplistic template formulation is to view it as an extended *nonlinear regression* problem: We seek a nonlinear relationship between the system outputs $y(t)$ and a known *regression vector* $\varphi(t)$:

$$y(t) = g(\varphi(t)) + e(t) \quad (2)$$

The regression vector is in principle a function of all inputs $u(s)$ and outputs $y(s)$ prior to time t:

$$\varphi(t) = h(y(t-1), u(t-1), y(t-2), u(t-2), \ldots) \quad (3)$$

It can be seen as a *state* for the system. In many of the considerations below we will assume that $\varphi(t)$ is known (measured) at time t. So, finding g is a way to compute (predictions of) future outputs. We assume that φ is d-dimensional: $\varphi \in R^d$

This work is supported by the Linnaeus Center CADICS, funded by the Swedish Research Council and by the strategic research center MOVIII, funded by the Foundation for Strategic Research, SSF.

Then the knowledge we have available about the nonlinear function g is

$$Z_{\mathrm{L}} = \{y(t), \varphi(t), t = 1, \ldots, N_{\mathrm{L}}\} \quad (4)$$

and from that information we need to find g.
There are two basic approaches for this:

- Parametric methods: g is parameterized as $g(\varphi, \theta)$ and the information Z_{L} is used to find an estimate $\hat{\theta}$ giving the function $\hat{g}(\varphi) = g(\varphi, \hat{\theta})$. The parameter θ is typically estimated by optimizing the fit $y(t) - g(\varphi(t), \theta)$ for the measured Z_{L}

- Nonparametric methods: $\hat{g}$ is formed from Z_{L} without the explicit use of parameters.

  - As a special case of non-parametric methods we may consider MOD – *model on demand* where the function $\hat{g}$ is never formed explicitly, but there is a computational scheme (using Z_{L}) that for any given φ^* computes the corresponding output $y^* = \hat{g}(\varphi^*)$

We shall give brief overviews of these basic approaches in the next sections.

3 PARAMETRIC METHODS: A PALETTE OF GREY SHADES

This has lead to a large, and sometimes confusing amount of approaches, and it is not easy to give a coherent description of the current status. Part of the problem is the negative definition ("non"-linear): it has been commented that this area is as huge as "non-elephant zoology" (quote attributed to mathematician/physicist Stan Ulam). In this section we give a brief account of the dominating concepts. It is customary in estimation to distinguish between grey-box models that are parameterizations based on physical insights, and black-box model, that are just flexible function surfaces. To bring some kind of order into nonlinear model structures we follow [13] and invoke a whole palette of grey shades from white to black. Only brief descriptions of the model classes will be given here. We refer to [13] for more details, formulas and further references.

3.1 White Models

White box models are the results of diligent and extensive physical modeling from first principles. This approach consists of writing down all known relationships between relevant variables and using software support to organize them suitably. Similarly, libraries of standard components and subsystems are frequently used. The resulting model could be given as a collection of differential algebraic equations (DAEs) or in state space form as

$$\dot{x}(t) = f(x(t), u(t), w(t)) \tag{5a}$$
$$y(t) = h(x(t), u(t)) + e(t) \tag{5b}$$

Here, y, u are the output and input as before, and w is a process noise sequence, while e is measurement noise.

3.2 Off-white Models: Parameterized Physical Models

Models with lightest shade of grey are obtained when white-box models (5) contain some parameters that have unknown or uncertain numerical values.
The nonlinear identification problem is to estimate such parameters from the measured Z_{L}. In general, this is a difficult problem, that has not yet been treated in full generality, due to the prediction problems when a process noise w is present. A good reference for a deterministic setting is [22].

3.3 Smoke-Grey Models: Semi-physical Modeling

By *semi-physical modeling* we mean finding nonlinear transformations of the measured data, so that the transformed data stand a better chance to describe the system in a linear relationship. The basic rule for this process (to ensure its leisurely aspect) is that only high-school physics should be required and the work must take no more than 10 minutes. See, e.g., [11], Examples 5.1 and pages 533 - 536 for examples of this kind.

3.4 Steel-Grey Models

All the model structures so far contained an element of physical insight. The insight can also simply be that different models are needed depending on the operating point of the system, or the current *regime*, like speed and altitude of an aircraft.

3.4.1 Composite Local Models:

Nonlinear systems are often handled by linearization around a working point.
The idea behind *composite local models* is to deal with the nonlinearities by developing local models, which are good approximations in different neighborhoods, and then compose a global model from these. Often, the local models are linear, so a common name for composite models is also *local linear models*. See, e.g. [9], and [14]. If each local model is defined as a linear regression we obtain a composite model

$$\hat{y}(t|\theta, \eta) = \sum_{k=1}^{d} w_k(\rho(t), \eta)\varphi^T(t)\theta^{(k)} \tag{6}$$

here $\hat{y}^{(k)}(t) = \varphi^T(t)\theta^{(k)}$, is the predicted output if the k:th model, and w_k is weight that assigns the relevance of the k:th model. $\rho(t)$ is the known current value of the regime variable (operating point). η is a vector that governs the partitioning of the global behavior into the local models, via the weighs w_k and the regime variable ρ For fixed η is a linear regression, since the regime variable $\rho(t)$ is measured and known.

3.4.2 LPV Models:

So-called *Linear Parameter Varying (LPV)* models are closely related to composite local models. In state space form they are described by:

$$\dot{x}(t) = A(\rho(t))x(t) + B(\rho(t))u(t)$$
$$y(t) = C(\rho(t))x(t) + D(\rho(t))u(t)$$

where the *exogenous* or regime parameter $\rho(t)$ is measured during the operation of the system. Identification of such models has been the subject of rather intense recent interest. See, e.g., [10], [1], [6], and [23].

3.5 Slate-Grey Models

The steel-gray models contain certain insights and knowledge into what type of regime variables are useful. A still slightly darker shade of gray is when the mechanisms of the model shifting are not known.

3.5.1 Hybrid Models:

The model (6) is also an example of a *hybrid* model. It is piecewise linear (or affine), and switches between different modes as the "state" $\varphi(t)$ varies over the partition. The regime variable ρ is then a known function of φ. If the partition has to be estimated too, the problem is considerably more difficult, due to the discrete/logical nature of the influence of η. Methods based on mixed integer and linear (or quadratic) programming are described in [18]. See also [3] for another approach.

3.5.2 Block-oriented Models.

A very useful idea is to build up structures from simple building blocks. This could correspond both to physical insights and as a means for generating flexible structures.
There are two basic building blocks for block-oriented models: linear dynamic system and nonlinear static transformation. These can be combined in a number of ways. Some combinations are known under well-known names, like Wiener and Hammerstein models.

3.6 Black Models: Basis Function Expansion

In a black-box setting, the idea is to parameterize the function $g(x, \theta)$ in a flexible way, so that it can well approximate any feasible true functions $g(x)$. A typical choice is to use a function expansion

$$g(x, \theta) = \sum_{k=1}^{m} \alpha_k g_k(x) \tag{7a}$$

with some basis functions g_k.

It turns out that a powerful choice of basis functions is to let them be generated from one and the same "mother function" $\kappa(x)$ and scale and translate it according to

$$g_k(x) = \kappa(\beta_k(x - \gamma_k)) \tag{7b}$$

(here written as if x is a scalar.) The basis functions are thus characterized by the scale (dilation) parameters β_k and the location (translation) parameters γ_k. The resulting structure (7) is very flexible and very much used. Depending on how the particular options are chosen, this contains, for example, radial basis neural networks, one-hidden-layer sigmoidal neural networks, neuro-fuzzy models, wavenets, least-squares support vector machines etc. See e.g [11], Chapter 5.

4 NON-PARAMETRIC REGRESSION

5 Nonparametric methods

According to (2), the function values are observed in additive noise. If many observations were made for the same value of $\varphi(k)$ it would thus be possible to estimate $g(\varphi(k))$ by averaging over the corresponding $y(k)$. This is the basic idea behind nonparametric methods: To average over relevant observations $y(k)$ to form an estimate of the function at a particular value φ^*. A general reference to nonparametric regression is [8].

5.1 Kernel Methods

The averaging or smoothing of observations takes the basic form

$$\hat{g}_N(\varphi^*) = \sum_{k=1}^{N} w_k y(k) \tag{8a}$$

$$\sum_{k=1}^{N} w_k = 1 \tag{8b}$$

The weights w_k will depend both on the target point φ^* and the observation point $\varphi(k)$:

$$w_k = C(\varphi^*, \varphi(k)) \tag{9a}$$

Typically, they depend only on the distance between the two points:

$$C(\varphi^*, \varphi(k)) = \frac{K_h(\varphi^* - \varphi(k))}{\sum_{j=1}^{N} K_h(\varphi^* - \varphi(j))} \tag{9b}$$

$$K_h(\tilde{\varphi^*}) = K(\tilde{\varphi^*}/h) \tag{9c}$$

where h is a parameter that scales the function K. This is an example of a *kernel method*, more precisely the *Nadaraya-Watson estimator*, [15]. Typical choices of the kernel function K are

$$K(\tilde{x}) = \frac{1}{\sqrt{2\pi}} e^{-\tilde{x}^2/2} \text{ (Gaussian)} \tag{10a}$$

$$K(\tilde{x}) = \frac{3}{4} \max\{1 - \tilde{x}^2, 0\} \text{ (Epanechnikov)} \tag{10b}$$

If the kernel is (essentially) zero for $|\tilde{x}| > 1$, observations that are further away than h (the *bandwidth*) from the target point φ^* in (8) will not be used in the function estimate.

It is obvious that the bandwidth parameter in this case is what controls the bias-variance trade-off: A small bandwidth gives few data to average over and hence a large variance. A large bandwidth means that the averaging takes place over a large area, where the true function may change quite a bit, thus leading to large bias.

5.2 Local Polynomial Methods

In a kernel estimator, the function value is estimated as a mean over a local neighborhood. A more sophisticated approach would be to compute a more advanced estimate within the neighborhood. For example, the function could be approximated as a polynomial within the chosen neighborhood. The coefficients of the polynomial are computed using a weighted least squares fit, the weights typically chosen as a kernel $K_h(u)$, (9c)-(10), giving more weight to the observations close to the target value φ^*. The estimate $\hat{g}(\varphi^*)$ would then be this polynomial's value at φ^*. This is the *local polynomial method*, see, e.g. [5]. Clearly, the Nadaraya-Watson estimator corresponds to a local polynomial approach with polynomials of zero order. It also follows that the local polynomial method is closely related to *local composite models*, (Section 3.4.1), often used in control applications.

5.3 Direct Weight Optimization

A very direct approach to determine the weights w_k in a nonparametric estimator (8) would be to choose them so that the mean square error (MSE) between the model output and the true output at the target point φ^*, is minimized w.r.t. w_k. Let M denote the MSE:

$$M_N(\varphi^*) = E|g(\varphi^*) - \hat{g}_N(\varphi^*)|^2$$

This value depends on g and the weights (and the probability distribution of e in (2)). To carry out the minimization, the true function $g(\varphi^*)$ needs to be known. To handle that, first a maximization of the MSE is carried out w.r.t. a function family $\mathcal{G}$ that g is assumed to belong to:

$$\hat{g} = \sum_{k=1}^{N} w_k y(k) \tag{11a}$$

$$\sum_{k=1}^{N} w_k = 1 \tag{11b}$$

$$w_k = \arg\min_{w_k} \max_{g_0 \in \mathcal{G}} M_N(\varphi^*) \tag{11c}$$

This method is described in [19]. The result depends, of course, on the function family $\mathcal{G}$. If the family consists of Lipschitz continuous functions

$$\mathcal{G}_2(L) = \{g; |g(x_1) - g(x_2)| \leq L|x_1 - x_2|\} \tag{12}$$

the resulting estimate (11) is a kernel type estimator, typically with the Epanechnikov kernel, and a bandwidth that is automatically selected from L, the assumed noise level, and the available observations. See also [21]. This method is an example of "model on demanad" (see Section 2): To find $\hat{g}(\varphi^*)$ the optimization problem (11) must be solved for each desired φ^*.

6 SEMI-SUPERVISED REGRESSION: WDMR

See [16] for a more comprehensive account of the concept and algorithm discussed in this section.

6.1 The concept of semi-supervised regression

We now introduce a new element into the discussion that brings the topic closer to machine learning and manifold learning. Suppose that in addition to the measurements (4) we have also measured N_{U} regression vector values without corresponding y ("unlabeled regressors"):

$$Z_{\mathrm{U}} = \{\varphi(t), \quad t = N_{\mathrm{L}} + 1, \dots N_{\mathrm{L}} + N_{\mathrm{U}}\} \tag{13}$$

Estimating g from Z_{L} *and* Z_{U} is called a *semi-supervised* regression problem: Z_{L} are "supervised" observations (the "labels" $y(t)$ are known and Z_{U} are unsupervised data. Clearly, there is information of any value in Z_{U} only if the regressors are confined to some *a priori* unknown region, like a manifold of R^d

Since we in the following will make no difference between the unlabeled regressor φ^* and $\{\varphi(t)\}_{t=N_{\mathrm{L}}+1}^{N_{\mathrm{L}}+N_{\mathrm{U}}}$, we simply include φ^* in the set of unlabeled regressors to make the notation a bit less cluttered. We introduce the simplified notation

$$\hat{g}_t \sim g(\varphi(t)) \tag{14}$$

for the estimates of $g(\varphi(t))$. In the following we will also need to introduce kernels

$$k(\varphi_1, \varphi_2)$$

as distance measure in the regressor space. To simplify the notation, we will use K_{ij} to denote a kernel $k(\cdot, \cdot)$ evaluated at the regressor pair $(\varphi(i), \varphi(j))$i.e.,

$$K_{ij} \triangleq k(\varphi(i), \varphi(j))$$

. A popular choice of kernel is the Gaussian kernel

$$K_{ij} = e^{-\|\varphi(i)-\varphi(j)\|^2/2\sigma^2}. \tag{15}$$

Since we will consider regressors constrained to certain regions of the regressor space (often manifolds), kernels constructed from manifold learning techniques, see Sect. 6.3, will be of particular interest. Notice however that we will allow us to use a kernel like

$$K_{ij} = \begin{cases} \frac{1}{K}, & \text{if } \varphi(j) \text{ is one of the } K \text{ closest} \\ & \text{neighbors of } \varphi(i), \\ 0, & \text{otherwise,} \end{cases} \tag{16}$$

and K_{ij} will therefore not necessarily be equal to K_{ji}. We will also always use the convention that $K_{ij} = 0$ if $i = j$.

We shall use the *semi-supervised (or manifold) smoothness assumption*: the function g is smooth (on the manifold) where the φ live. This means that we would like the estimates belonging to two regressors which are close in this region to have similar values. Using a kernel, we can express this as

$$\hat{g}_t \approx \sum_{i=1}^{N_{\mathrm{L}}+N_{\mathrm{U}}} K_{ti}\hat{g}_i, \quad t = 1 \dots N_{\mathrm{L}} + N_{\mathrm{U}} \tag{17}$$

where K_{ti} is a kernel giving a measure of distance between $\varphi(t)$ and $\varphi(i)$, relevant to the assumed region.

6.2 The WDMR Method

So the sought estimates $\hat{g}_i$ should be such that they are smooth over the region. At the same time, for regressors in Z_{L} with measured y:s, the estimates should be close to those, meaning that

$$\sum_{t=1}^{N_{\mathrm{L}}} (y(t) - \hat{g}_t)^2 \tag{18}$$

should be small. The two requirements (17) and (18) can be combined into a criterion

$$\lambda \sum_{i=1}^{N_{\mathrm{L}}+N_{\mathrm{U}}} (\hat{g}_i - \sum_{j=1}^{N_{\mathrm{L}}+N_{\mathrm{U}}} K_{ij}\hat{g}_j)^2 + (1-\lambda)\sum_{t=1}^{N_{\mathrm{L}}} (y(t) - \hat{g}_t)^2 \tag{19}$$

to be minimized with respect to $\hat{g}_t$, $t = 1, \dots, N_{\mathrm{L}} + N_{\mathrm{U}}$. The scalar λ decides how trustworthy our labels are and is seen as a design parameter.

The criterion (19) can be given a Bayesian interpretation as a way to estimate $\hat{g}$ in (18) with a "smoothness prior" (17), with λ reflecting the confidence in the prior.

Note that (19) is quadratic in $\hat{g}_j$ so the minimization is straightforward.

When the kernel K is chosen by local linear embedding (see next section 6.3, $K_{ij} = \ell ij$ in (20)), we call the estimation method (19) *Weight Determination by Manifold Regularization* (WDMR, [17]). It is clearly another example od Model on Demand: Each new φ^* it has to be included in Z_{U} and the problem (19) re-solved. In this case the unlabeled regressors are used to get a better knowledge for what parts of the regressor space that the function g varies smoothly in.

Similar methods to the one presented here has also been discussed in [7, 25, 4, 2, 24].

6.3 LLE: A Way of Selecting the Kernel in WDMR

Local Linear Embedding, LLE, [20] is a technique to find lower dimensional manifolds to which an observed collection of regressors belong. A brief description of it is as follows:

Let $\{\varphi(i), i = 1, \dots, N\}$ belong to $U \subset R^d$ where U is an unknown manifold of dimension n_z. A coordinatization $z(i)$, $(z(i) \in R^{n_z})$ of U is then obtained by first minimizing the cost function

$$\varepsilon(l) = \sum_{i=1}^{N} \left\| \varphi(i) - \sum_{j=1}^{N} l_{ij}\varphi(j) \right\|^2 \tag{20a}$$

under the constraints

$$\begin{cases} \sum_{j=1}^{N} l_{ij} = 1, \\ l_{ij} = 0 \text{ if } \|\varphi(i) - \varphi(j)\| > C_i(K) \text{ or if } i = j. \end{cases} \tag{20b}$$

Here, $C_i(K)$ is chosen so that only K weights l_{ij} become nonzero for every i. K is a design variable. It is also common to add a regularization to (20a) not to get degenerate solutions.

Then for the determined l_{ij} find $z(i)$ by minimizing

$$\sum_{i=1}^{N} \left\| z(i) - \sum_{j=1}^{N} l_{ij} z(j) \right\|^2 \tag{21a}$$

wrt $z(i) \in R^{n_z}$ under the constraint

$$\frac{1}{N}\sum_{i=1}^{N} z(i)z(i)^T = I_{n_z \times n_z} \tag{21b}$$

$z(i)$ will then be the coordinate for $\varphi(i)$ in the lower dimensional manifold.

The link between WDMR and LLE is now clear: If we pick the kernel K_{ij} in (19) as l_{ij} from (20) and have no labeled regressors ($N_{\text{L}} = 0$) and add the constraint on $\hat{g}_i$ corresponding to (21b) minimization of the WDMR criterion (19) will yield $\hat{g}_i$ as the LLE coordinates $z(i)$.

In WDMR with labeled regressors, the addition of the criterion (18) in (19) will replace the constraint corresponding to (21b) as an anchor to prevent a trivial zero solution. Thus WDMR is a natural semi-supervised version of LLE, [17].

See [16] for more details and experimental tests of the suggested approach.

7 CONCLUDING REMARKS

The major approaches to nonlinear system identification can be divided into parametric and nonparametric methods. We have given a quick and superficial account of the most common parametric models used. These correspond to the main thrust of the control community's activities in the area. The nonparametric approaches (Section 4) may correspond to interests in the statistical community in the past two decades. The manifold learning and semi-supervised techniques, where we gave most technical details correspond to active and current interests in the machine learning community. Not much of that has spread into the conventional system identification literature. It is an exciting question for the future to see how many methods of practical relevance for our control area will come out of this.

REFERENCES

[1] B. Bamieh and L. Giarré. Identification of linear parameter varying models. *Int. Journal of Robust and Nonlinear Control*, 12:841–853, 2002.

[2] Mikhail Belkin, Partha Niyogi, and Vikas Sindhwani. Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *J. Mach. Learn. Res.*, 7:2399–2434, 2006.

[3] A. Bemporad, A. Garulli, S. Paoletti, and A. Vicino. A bounded-error approach to piecewise affine system identification. *IEEE Transactions on Automatic Control*, 50(10):1567–1580, October 2005.

[4] Yoshua Bengio, Olivier Delalleau, and Nicolas Le Roux. Label propagation and quadratic criterion. In Olivier Chapelle, Bernhard Schölkopf, and Alexander Zien, editors, *Semi-Supervised Learning*, pages 193–216. MIT Press, 2006.

[5] J. Fan and I. Gijbels. *Local Polynomial Modelling and Its Applications*. Number 66 in Monographs on Statistics and Applied Probability. Chapman & Hall, 1996.

[6] A. Fujimori and Lennart Ljung. Model identification of linear parameter varying aircraft systems. *Proc. Inst. Mechanical Engineers, Part G Journal of Aerospace Engineering*, 220(G4):337–346, August 2006.

[7] Andrew B. Goldberg and Xiaojin Zhu. Seeing stars when there aren't many stars: Graph-based semi-supervised learning for sentiment categorization. In *HLT-NAACL 2006 Workshop on Textgraphs: Graph-based Algorithms for Natural Language Processing*, 2006.

[8] W. Härdle. *Applied Nonparametric Regression*. Cambridge Univeristy Press, Cambridge, UK, 1990.

[9] T. A. Johansen and B. A. Foss. Identification of nonlinear-system structure and parameters using regime decomposition. *Automatica*, 31(2):321–326, 1995.

[10] L. Lee and K. Poolla. Identification of linear parameter-varying systems using non-linear programming. *ASME Journal of Dynamic Systems, Measurement and Control*, 121:71–78, 1999.

[11] L. Ljung. *System Identification - Theory for the User*. Prentice-Hall, Upper Saddle River, N.J., 2nd edition, 1999.

[12] Lennart Ljung. Some aspects on nonlinear system identification. In *Proc 14th IFAC Symposium on System Identification*, Newcastle, Australia, March 2006.

[13] Lennart Ljung. Perspectives on system identification. *Annual Reviews in Control*, 34(1), March 2010.

[14] R. Murray-Smith and T. A. Johansen, editors. *Multiple Model Approaches to Modeling and Control*. Taylor and Francis, London, 1997.

[15] E. Nadaraya. On estimating regression. *Theory of Prob. and Applic.*, 9:141–142, 1964.

[16] Henrik Ohlsson and Lennart Ljung. Semi-supervised regression and system identification. In B. Wahlberg X. Hu, U. Jonsson and B.Ghosh, editors, *Three Decades of Progress in Systems and Control*. Springer Verlag, January 2010.

[17] Henrik Ohlsson, Jacob Roll, and Lennart Ljung. Manifold-constrained regressors in system identification. In *Proc. 47st IEEE Conference on Decision and Control*, pages 1364–1369, December 2008.

[18] J. Roll, A. Bemporad, and L. Ljung. Identification of piecewise affine systems via mixed-integer programming. *Automatica*, 40(1):37–50, Jan 2004.

[19] J. Roll, A. Nazin, and L. Ljung. Non-linear system identification via direct weight optimization. *Automatica*, 41(3):475–490, Mar 2005.

[20] Sam T. Roweis and Lawrence K. Saul. Nonlinear Dimensionality Reduction by Locally Linear Embedding. *Science*, 290(5500):2323–2326, 2000.

[21] J. Sacks and D. Ylvisaker. Linear estimatiob for approximately linear models. *The Annals of Statistics*, 6(5):1122–1137, 1978.

[22] K. Schittkowski. *Numerical Data Fitting in Dynamical Systems*. Kluwer Academic Publishers, Dordrecht, 2002.

[23] R. Toth, T. S. C. Heuberger, and P. M. J. van den Hof. Asymptotically optimal orthogonal basis functions for lpv system identification. *Automatica*, 45(6):1359–1370, 2009.

[24] Fei Wang and Changshui Zhang. Label propagation through linear neighborhoods. *Knowledge and Data Engineering, IEEE Transactions on*, 20(1):55–67, Jan. 2008.

[25] 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.