

- Estimation of Errors-In-Variables Models Using a Sample Covariance Matrix Obtained from Small Data Sets”, *Recent Advances in Total Least Squares Techniques and Errors-In-Variables Modeling*, 1997, pp. 59 -68.
- [22] Sjöberg J., “Non-Linear System Identification with Neural Networks”, *Linköping Studies in Science and Technology Dissertation No. 381*, Sweden, 1995.
- [23] Sjöberg J., Ljung L., “Overtraining, Regularization and Searching for a Minimum, With Application to Neural Networks”, *International Journal on Control*, Vol. 62, 1995, pp. 1391 - 1407.
- [24] Soucek B., the IRIS group (Eds.), “Fast Learning and Invariant Object Recognition, the Sixth-generation Breakthrough”, John Wiley & Sons, Inc., 1992.
- [25] Stuart A., Ord J. K., “Kendall’s Advanced Theory of Statistics, Distribution Theory”, Vol. 1, Charles Griffin & Co., 1987.
- [26] Tresp V., Reimar H., “Missing and Noisy Data in Nonlinear Time-Series Prediction”, to be published in *Neural Networks for Signal Processing*, Vol. 5, 1995.
- [27] Twomey J. M., Smith A. E., “Bias and Variance of Validation Methods for Function Approximation Neural Networks Under Conditions of Sparse Data”, *IEEE Transactions on Systems, Man, and Cybernetics - Part C: Applications and Reviews*, Vol. 28, No. 3, August 1998, pp. 417 - 430.
- [28] Utans J., “Stopped Training via Algebraic On-line Estimation of the Expected Test-Set Error”, *ICNN '97*, IEEE International Conference on Neural Networks, Alabama, USA, August 1997, pp. 1088 - 1092.
- [29] Vandersteen G., Rolain Y., Schoukens J., Pintelon R., “On the Use of System Identification for Accurate Parametric Modeling of Nonlinear Systems Using Noisy Measurements”, *IEEE Transactions on Instrumentation and Measurement*, Vol. 45, No. 2, April 1996, pp. 605 - 609.
- [30] Van Gorp J., Schoukens J., Pintelon R., “Adding Input Noise to Increase the Generalization of Neural Networks is a Bad Idea”, *ANNIE 1998, Intelligent Engineering Systems Through Artificial Neural Networks*, Volume 8, pp. 127 - 132.
- [31] Van Gorp J., Schoukens J., Pintelon R., “The Errors-In-Variables Cost Function for Learning Neural Networks with Noisy Inputs”, *ANNIE 1998, Intelligent Engineering Systems Through Artificial Neural Networks*, Volume 8, pp. 141 - 146.

results of the NN mappings for the same type of NN is shown in Fig. 12. It comes clear that the variability of the NN has decreased significantly, while the original measurements are preserved.

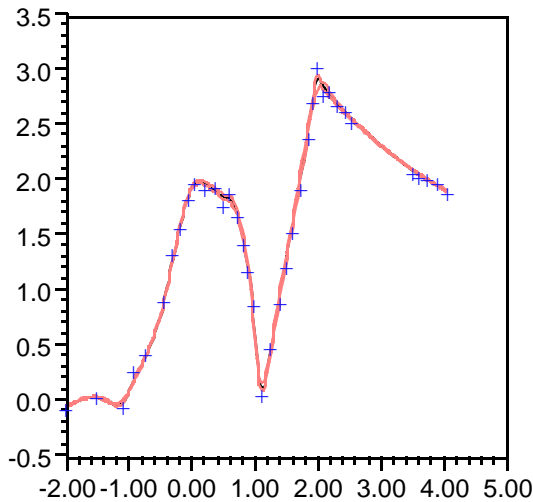


Fig. 12. Variability of the same NN structure with prior interpolation. Solid line: mean NN mapping; gray line: variability on the NN mappings.

VI. CONCLUSION

This paper presented an interpolation scheme, based upon a local linearization of the data. The combination of the different local-linear parts leads to a polynomial interpolation of the data, which has a relationship with spline fitting. The choice on different parameters for the interpolation is based upon 95% or 99% certainty boundaries. The advantages of the given method are a physical meaning of the smoothing factor, the applicability on higher dimension multi-input systems, the straightforward calculation, low bias, and a near-linear behaviour when the scheme is used for extrapolation. Examples are given to illustrate the theory.

A drawback of the given interpolation scheme is that it is hardly useable for compression of the data: typically a large number of parameters are used to represent the measurement data. The scheme is therefore not recommended for use as a black box model, but it is well useable as a filter for noisy measurements or for creating interpolated data points in the case of sparse data.

Acknowledgments

This paper presents research results of the Belgian Programme on Interuniversity Poles of Attraction (IUAP 4/2), initiated by the Belgian State, Prime Minister's Office for Science, Technology and Culture; and the Flemish Government (GOA, IMMI2). The research is supported by the Fund for Scientific Research - Flanders (FWO).

References

- [1] Amari S.-I., Murata N., Müller K.-R., Finke M., Yang H. H., "Asymptotic Statistical Theory of Overtraining and Cross-Validation", *IEEE Transactions on Neural Networks*, Vol. 8, No. 5, September 1997, pp. 985 - 998.
- [2] Amemiya Y., "Two-Stage Instrumental Variable Estimators for the Nonlinear Errors-In-Variables model", *Journal of Econometrics*, No. 44, 1990, pp. 311 - 332.
- [3] An G., "The Effects of Adding Noise During Backpropagation Training on a Generalization Performance", *Neural Computation*, Vol. 8, 1996, pp. 643 - 674.
- [4] Baldi P. F., Hornik K., "Learning in Linear Neural Networks: a Survey", *IEEE Transactions on Neural Networks*, Vol. 6, No. 4, July 1995, pp. 837 - 858.
- [5] Bishop C. M., "Neural Networks for Pattern Recognition", Clarendon Press, Oxford, 1995.
- [6] Fausett L., "Fundamentals of Neural Networks, Architectures, Algorithms and Applications", Prentice Hall, 1994.
- [7] Freeman J. A., Skapura D. M., "Neural Networks, Algorithms, Applications and Programming Techniques", Addison-Wesley Publishing Co., 1991.
- [8] Funahashi K.-I., "On the Approximate Realization of Continuous Mappings by Neural Networks", *Neural Networks*, Vol. 2, 1989, pp. 183 - 192.
- [9] Geman S., Bienenstock E., Doursat R., "Neural Networks and the Bias/Variance Dilemma", *Neural Computation*, Vol. 4, 1992, pp. 1 - 58.
- [10] Haykin S., "Neural Networks, a Comprehensive Foundation", Prentice Hall, 1999.
- [11] Hornik K., "Approximation Capabilities of Multilayer Feedforward Networks", *Neural Networks*, Vol. 4, 1991, pp. 251-257.
- [12] Ljung L., "System Identification, Theory for the User", Prentice-Hall information and system sciences series, 1994.
- [13] Lukacs, "Stochastic Convergence", Academic Press, New York, 1975.
- [14] MacKay D. J. C., "Bayesian Interpolation", *Neural Computation*, Vol. 4, 1992, pp. 415 - 447.
- [15] MacKay D. J. C., "Bayesian Methods for Adaptive Models", PhD thesis, California Institute of Technology, Pasadena, California, 1992.
- [16] Masters T., "Practical Neural Network Recipes in C++", Academic Press, Inc., 1993.
- [17] Miller W. T. III, Sutton R. S., Werbos P. J. (Eds.), "Neural Networks for Control", Massachusetts Institute of Technology, 1990.
- [18] Nelles O., Hecker O., Isermann R., "Automatic Model Selection in Local Linear Model Trees (LOLIMOT) for Nonlinear System Identification of a Transport Delay Process", *SYSID '97, 11th IFAC Symposium on System Identification*, Japan, July 1997, pp. 727 - 732
- [19] Reed R., Marks R. J. II, Oh S., "Similarities of Error Regularization, Sigmoid Gain Scaling, Target Smoothing, and Training with Jitter", *IEEE Transactions on Neural Networks*, Vol. 6, No. 3, May 1995, pp. 529 - 538.
- [20] Sarle W. S., "Stopped Training and Other Remedies for Overfitting", *Proceedings of the 27th Symposium on the Interface*, 1995, pp. 1 - 10.
- [21] Schoukens J., Pintelon R., Rolain Y., "Maximum Likelihood

with $x \in [0.1, 3.1]$ is sampled 100 times and noise is added to the output samples with a standard deviation $\sigma = 0.2$. The data points are used to train a feedforward neural network (NN). A large number of neurons is used and the network tends to overfit the simulation data, as is shown in Fig. 9. To compensate the overfitting behaviour, noise is deliberately added to the inputs. This commonly used technique is expected to give better generalization of the NN, but also inevitably leads to bias on the NN parameters (Van Gorp et al. [31]). The remaining question is how to choose the proper level of input noise.

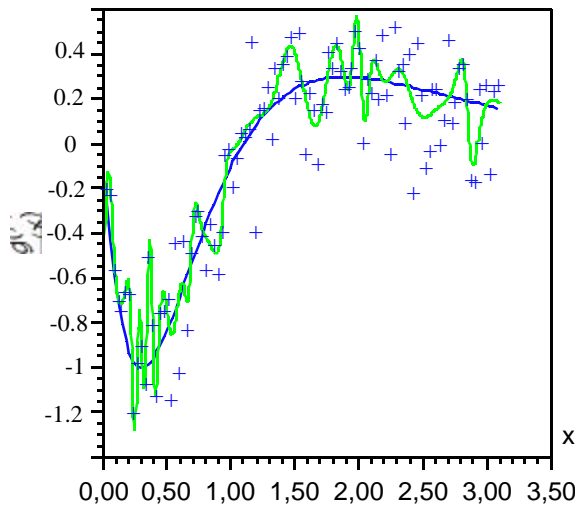


Fig. 9. Overfitting of a NN due to sample noise. Solid line: true function; crosses: simulation samples; gray line: NN mapping.

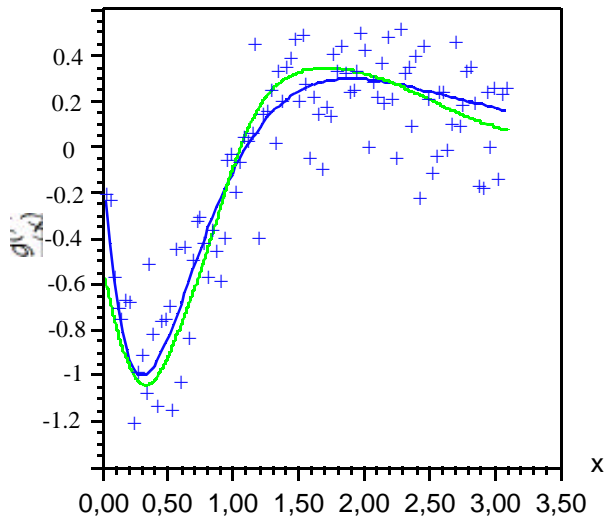


Fig. 10. Mapping of the same NN with prior interpolation of the data using the given scheme with limitation to 4 Hz. Solid line: true function; gray line: NN approximation.

The data was interpolated with a smaller grid of 1,000 samples with the previously given procedure. Based upon the data, it was assumed that the highest frequency was 4 Hz.

With $N = 100$ this gives a smoothing factor $f_s = 21.738$. The interpolated points were used for training the same neural network. The resulting network is shown in Fig. 10 and very much resembles the optimal solution given by Geman. However, in this case the hassle of finding the right amount of input noise is avoided and replaced by the easy choice of the highest frequency in the measurement data. Moreover, the training of the NN parameters took far less time when the preprocessing of the data was used, even with the added time needed for the preprocessing and taking into account that more data points were used. Typically the learning scheme for the NN parameters spends a lot of time in the overtraining phase. This overtraining phase is largely avoided with filtered data.

B. Overfitting example (Mackay [14])

A NN is trained to fit 37 input-output measurements. The input-output pairs (u, y) were not measured on an equidistant grid and gaps remain within the measurement points. Mackay noticed that overtraining led to a large variability of the NN in these gaps. To illustrate this Fig. 11. shows the mean result and variance of 100 simulations of a feedforward perceptron NN with one hidden layer and 20 neurons in the hidden layer.

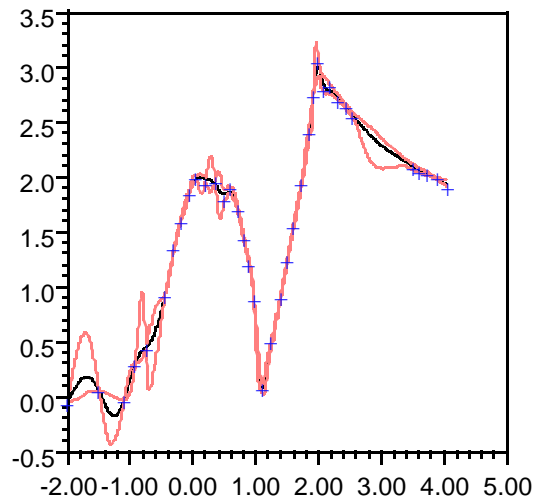


Fig. 11. Variability of NN in the case of sparse data. Solid line: mean NN mapping; gray line: variability on the NN mappings.

The given solution for this overfitting was to increase the number of “measurements” by taking a large number of copies of the data samples and add Gaussian noise to the inputs of the resulting data set. This results in a “smoothing” of the NN mapping but also creates bias on the NN parameters. Since the final goal of adding noise is to become a “smooth” behaviour of the NN, we used the interpolation scheme to fill in the gaps. The smoothing factor was chosen as $f_s = 1.97$ at the measurements were interpolated with 200 interpolated points in between the measurements. The

can be seen that the interpolated curve fits the datapoints for $f_s = 1$, but that the interpolated points are severely biased for $f_s = 10$. Further it is shown that these biasing effects are in fact the results of a low pass filter. The effects of the smoothing factor are better explained using frequency analysis. A multisine signal is sampled 1000 times on an equidistant grid. The highest frequency of the multisine signal is arbitrarily chosen as 250. The samples are interpolated with the described SISO method with $f_s = 1$, a simple linear interpolation scheme and using a cubic interpolation. The grid used for interpolation has a density which is 10 times higher than the original grid, leading to 9991 interpolated points. A detail of the interpolation is given in Fig. 6. The frequency analysis of the interpolated signals is shown in Fig. 7.

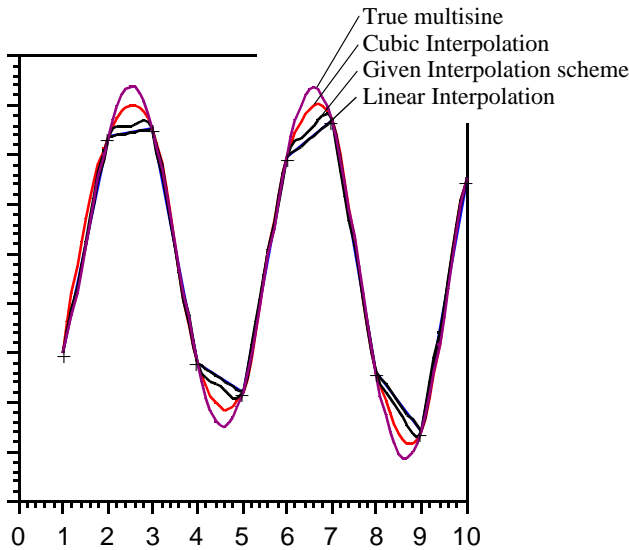


Fig. 6. Interpolation scheme compared to linear and cubic interpolation

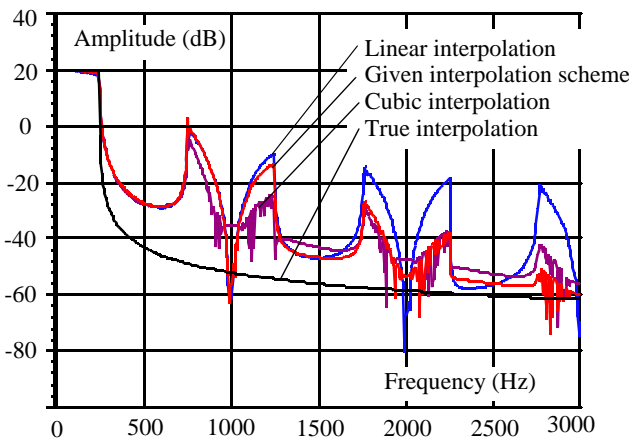


Fig. 7. Frequency analysis of the interpolated signals

From Fig. 7 it can be seen that the interpolation scheme

given in this paper, has the same distortion as a linear interpolation scheme for frequencies, while it outperforms the cubic interpolation at higher frequencies. The interpolation scheme acts as a low pass filter with a cut-off frequency that only depends on the smoothing factor. On an empirical basis we found that the -3dB frequency of the filter (f_{-3dB}) can be calculated as

$$\frac{f_{-3dB}}{N} = \frac{0.78127}{f_s} + 4.0610^{-3} \quad (36)$$

in the particular case on an equidistant grid such that $\Delta_u = 1/N$, and with values $0.01 < f_{-3dB}/N < 0.4$. The value $f_{-3dB}/N < 0.4$ showed to be a practical limit for the given optimization scheme, which yields a lower limit $f_s \geq 1.97$ for the smoothing factor. If the number of samples of the original grid N , and a certain cut-off frequency f_{-3dB} is given, the proper smoothing factor can be calculated as

$$f_s = 0.78127 \left(\frac{f_{-3dB}}{N} - 4.0610^{-3} \right)^{-1} \quad (37)$$

Consider again the example of Fig. 7, choosing the highest cut-off frequency $f_{-3dB} = 400\text{Hz}$. With $N = 1000$ the optimal smoothing factor becomes $f_s = 1.97$. The result of this interpolation is shown in Fig. 8. It is clear that the interpolation scheme causes far less distortion than the cubic interpolation scheme, while the original signal is largely preserved.

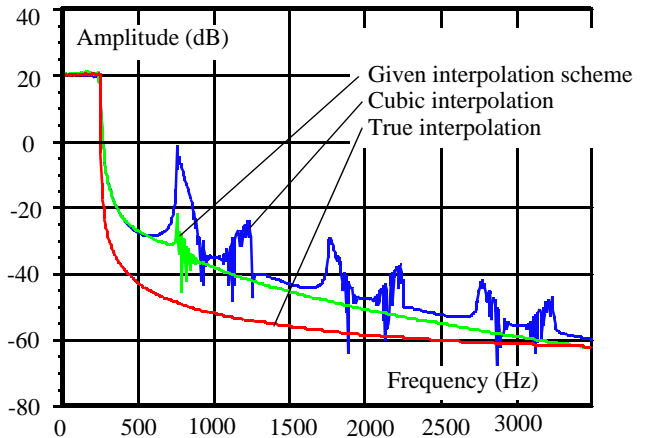


Fig. 8. Frequency analysis of the interpolation scheme with cut-off frequency at 400Hz.

V. SIMULATION RESULTS

A. Regression example (Geman et al.[9]).

A functional

$$(x) = 4.26(e^{-x} - 4e^{-2x} + 3e^{-3x}) \quad (38)$$

$$\mathbf{u}_{99} = [\mathbf{u}_1^T \ \mathbf{u}_2^T \ \dots \ \mathbf{u}_{P_{j_1, j_2, \dots, j_m}}^T]^T \in \mathbb{R}^{P_{j_1, j_2, \dots, j_m} \times m}. \quad (26)$$

In the case of noiseless inputs and noisy output measurements ($\sigma_{u, i, k} = 0$ and $\sigma_{y, k} \neq 0$), the A_j are calculated using

$$A_{j_1, j_2, \dots, j_m} = (\Sigma_y [\mathbf{u}_{99} | \mathbf{1}_{P_{j_1, j_2, \dots, j_m}}]) \backslash (\Sigma_y [\mathbf{y}_{99}]). \quad (27)$$

Σ_y is defined as

$$y = \text{diag}([\sigma_{y, 1}, \sigma_{y, 2}, \dots, \sigma_{y, P_{j_1, j_2, \dots, j_m}}]). \quad (28)$$

If both the inputs and outputs are noisy ($\sigma_{u, i, k} \neq 0$ and $\sigma_{y, k} \neq 0$) the following iterative scheme is used

$$A_{j_1, j_2, \dots, j_m, t+1} = (\Sigma_{u, y, t} [\mathbf{u}_{99} | \mathbf{1}_{P_{j_1, j_2, \dots, j_m}}]) \backslash (\Sigma_{u, y, t} [\mathbf{y}_{99}]) \quad (29)$$

with

$$\Sigma_{u, y, t} = \text{diag}([\sqrt{\sigma_{y, 1}^2 + a_{1, 1, t}^2 \sigma_{u, 1, 1}^2 + \dots + a_{1, m, t}^2 \sigma_{u, 1, m}^2}, \dots, \sqrt{\sigma_{y, j}^2 + a_{j, 1, t}^2 \sigma_{u, j, 1}^2 + \dots + a_{j, m, t}^2 \sigma_{u, j, m}^2}]). \quad (30)$$

The scheme is repeated for t until convergence is reached on all $a_{j, i, t}$ parameters. The initial parameters $a_{j, i, 0}$ can be found solving (27), which is linear in the parameter A_j .

For the final interpolation, the local linear systems $A_{P_{j_1, j_2, \dots, j_m}}$ are recombined. Denote the grid points of the interpolation grid as \mathbf{x}_i with

$$\mathbf{x}_i = [x_{i, 1}, x_{i, 2}, \dots, x_{i, m}] \quad (31)$$

and $i = 1, 2, \dots, K$. There is no need to create fine grid points that are equidistant, or ordered in any way. Define the smoothing variances

$$\sigma_{sm, i, j}^2 = \frac{f_s \Delta_{i, j}}{\Delta_{99}} \quad (32)$$

with f_s the smoothing factor¹. The fine grid is then calculated using the following equations:

$$\phi_{i, j_1, j_2, \dots, j_m} = \exp\left(-\frac{(c_{j_1} - x_{i, 1})^2}{\sigma_{sm, 1, j_1}^2} - \dots - \frac{(c_{j_m} - x_{i, m})^2}{\sigma_{sm, m, j_m}^2}\right) \quad (33)$$

1. For simplicity of notation, one overall smoothing factor is chosen here. If only some of the inputs are known to be noisy, it is possible to define different smoothing factors for each of the m input dimensions.

$$\phi_i = \sum_{j_1=1}^{K_1} \sum_{j_2=1}^{K_2} \dots \sum_{j_m=1}^{K_m} \phi_{i, j_1, j_2, \dots, j_m} \quad (34)$$

and

$$y_i = \frac{1}{\phi_i} \sum_{j_1=1}^{K_1} \dots \sum_{j_m=1}^{K_m} \phi_{i, j_1, j_2, \dots, j_m} (A_{j_1, j_2, \dots, j_m} [\mathbf{x}_i \ \mathbf{1}]^T). \quad (35)$$

The resulting y_i ($i = 1, 2, \dots, K$) are the interpolated values for the grid inputs u_i .

IV. ANALYSIS OF THE SMOOTHING FACTOR

Remark that in the above interpolation scheme, only one variable is chosen by the user: the smoothing factor. This factor directly influences the width of the Parzen windows, i.e. the region in which each local-linear section is effective. If the smoothing factor is chosen as $f_s = 1$, then each local-linear section is based upon the Δ_{99} width, i.e. each point of the fine grid has a 99% probability that it falls within the correct local-linear section. Choosing larger values for f_s means that the region of influence is enlarged accordingly, thus doubling the region of influence if the smoothing factor is chosen as $f_s = 2$. It is possible to choose $f_s < 1$ in order to fully eliminate smoothing. However, in practice care should be taken that $\sigma_{sm, i}$ is chosen large enough (typically $\sigma_{sm, i} > 10^{-3}$) to avoid calculation precision problems. For the same reason, smoothing factors should be chosen that are larger than $f_s > 0.1$. It must be stressed that choosing larger smoothing factors also increases biasing effects. This is shown more in detail in Fig. 5.

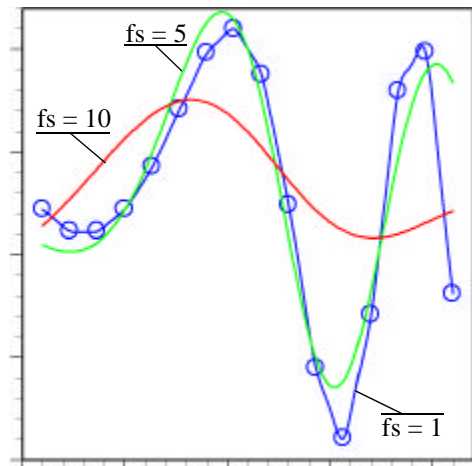


Fig. 5. Biasing effects caused by choosing larger smoothing factors

The above example is repeated with noiseless samples and using smoothing factors $f_s = 1$, $f_s = 5$ and $f_s = 10$. It

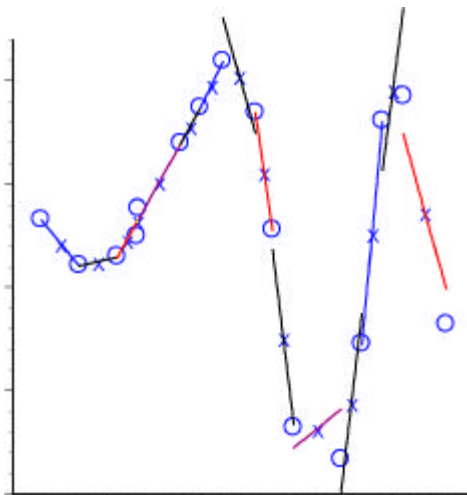


Fig. 3. Local-linear sections. Circles: measurement samples; crosses: section centers, lines: local-linear interpolation.

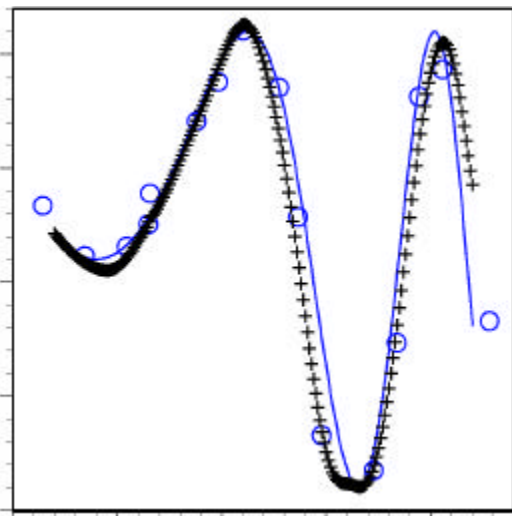


Fig. 4. Interpolation results. Solid line: true function, circles: measurement data, crosses: interpolated samples with $f_s = 2$.

III. INTERPOLATION OF SPARSE DATA FOR MIMO SYSTEMS

Assumption 3: The outputs of the MIMO system are mutually uncorrelated. We consider therefore that each output is only determined by the inputs and that none of the outputs is used as an input (feedforward system). \square

Under assumption 3, it is possible to consider the MIMO system as a concatenation of n MISO systems, such that further the discussion is restricted to MISO systems, following the lines of section II. The N input and output measurements are denoted as

$$\mathbf{U} = \begin{bmatrix} u_{1,1}, u_{1,2}, \dots, u_{1,m} \\ u_{2,1}, u_{2,2}, \dots, u_{2,m} \\ \dots \\ u_{N,1}, u_{N,2}, \dots, u_{N,m} \end{bmatrix} \in \mathbb{R}^{N \times m} \quad (20)$$

and

$$\mathbf{y} = [y_1, y_2, \dots, y_N]^T \in \mathbb{R}^{N \times 1}. \quad (21)$$

We consider that the measurements are noisy with known (or experimentally determined) variances $\sigma_{u,i,j}$ and $\sigma_{y,i}$, and not ordered in any way. A coarse grid is defined, e.g. by dividing each input dimension i in K_i equidistant sections, thus creating a total of $K_1 \times K_2 \times \dots \times K_m$ hypercubes with corners q_{j_1, j_2, \dots, j_m} ($j_i = 0, 1, \dots, K_i$) and hypercube centers c_{j_1, j_2, \dots, j_m} ($j_i = 1, 2, \dots, K_i$). Each center has coordinates $c_{j_1, j_2, \dots, j_m} = (c_{j_1}, c_{j_2}, \dots, c_{j_m})$. The length of each hypercube side is denoted as $\Delta_{i,j}$ with $i = 1, 2, \dots, m$, $j = 1, 2, \dots, K_i$ and

$$\Delta_{i,j} = q_{j_1, \dots, j_i = j, \dots, j_m} - q_{j_1, \dots, j_i = j-1, \dots, j_m} \quad (22)$$

Another method to divide the input space in Local-linear spaces, is based on the measurement data properties, and described by Nelles et al. [18]. For each hypercube, the input measurements are determined that lie within a “neighborhood” of the hypercube. The selected measurements either lie within the hypercube, or have 95% or 99% boundaries that intersect with the hypercube. Any point $\mathbf{x} = (x_1, x_2, \dots, x_m)$ of a hypercube lays within the 95% or 99% boundary of an input measurement $\mathbf{u}_i = [u_{i,1}, u_{i,2}, \dots, u_{i,m}]$ if it lies within the hyperellipsoid formed by the standard deviations of the data points, i.e. if the following equation holds:

$$\frac{(x_{i,1} - x_1)^2}{\sigma_{u,i,1}^2} + \frac{(u_{i,2} - x_2)^2}{\sigma_{u,i,2}^2} + \dots + \frac{(u_{i,m} - x_m)^2}{\sigma_{u,i,m}^2} \leq \Delta. \quad (23)$$

Denote the number of selected measurements for a hypercube as P_{j_1, j_2, \dots, j_m} . For the noiseless case, the parameters of the local-linear hyperplane, denoted as $\mathbf{A}_{j_1, j_2, \dots, j_m} \in \mathbb{R}^{m+1}$, are calculated as

$$\mathbf{A}_{j_1, j_2, \dots, j_m} = [\mathbf{u}_{99} | \mathbf{1}_{P_{j_1, j_2, \dots, j_m}}] \setminus [\mathbf{y}_{99}] \quad (24)$$

with

$$\mathbf{A}_{j_1, j_2, \dots, j_m} = [a_{j_1, j_2, \dots, j_m, 1}, \dots, a_{j_1, j_2, \dots, j_m, m}, b_{j_1, j_2, \dots, j_m}] \quad (25)$$

and \mathbf{u}_{99} is defined as

Lemma 1: The local-linear parameters A_k that are found using (8), (9) and (11) are locally unbiased, i.e. are unbiased with respect to the \hat{u}_k and \hat{y}_k vectors.

Proof: see Ljung [12] and Schoukens et al. [21]. \square

The recombination of the $N - 1$ sections is done using a Parzen-window, as defined by Nelles et al. [18]. The width of each section is defined as

$$\Delta_i = u_{i+1} - u_i \quad (13)$$

and the centers of the sections are calculated as

$$c_i = u_i + \Delta_i / 2 \quad (14)$$

with $i = 1, 2, \dots, N - 1$ and $c = [c_1, c_2, \dots, c_{N-1}]^T$. The fine grid with the interpolated inputs is denoted as $x = [x_1, x_2, \dots, x_K]^T$. Define the smoothing variance

$$\sigma_{sm,i}^2 = \frac{f_s \Delta_i}{\Delta_{99}} \quad (15)$$

with f_s the smoothing factor. The recombination of the sections is then done with the definition of

$$\phi_{i,j} = \exp\left(-\frac{(c_j - x_i)^2}{2\sigma_{sm,i}^2}\right) \quad (16)$$

with $i = 1, \dots, K$ and $j = 1, \dots, N - 1$, and

$$\phi_i = \sum_{j=1}^{N-1} \phi_{i,j} \quad (17)$$

such that finally

$$= \frac{1}{\phi_i} \sum_{j=1}^{N-1} \phi_{i,j} (a_j x_i + b_j; i = 1, 2, \dots, K. \quad (18)$$

Illustrative example

To fix the ideas, consider the regression example of An [3], where an arbitrary functional

$$f = \sin(3(u + 0.8)^2) \text{ with } u \in [-1, 1] \quad (19)$$

is sampled at 16 points. Noise is added on the inputs and outputs with standard deviations $\sigma_u = 0.05$ and $\sigma_y = 0.05$. The sampling is repeated 10 times, such that the sample variances of the inputs u and outputs y can be calculated. This results in 16 input-output measurements that are sorted for u . These samples are shown in Fig. 1. The solid line is the true function and the sample means are shown with circles. The 99% interval based on the sample variance is shown with a horizontal line. The calculation of the local-linear section 2-3 is shown in detail in Fig. 2. The 99% boundaries of samples 4 and 5 overlap with the section 2-3 and are used for the calculation of the local-linear piece 2-3.

The vectors \hat{u}_2 and \hat{y}_2 therefore each contain 4 elements, such that $\hat{u}_2 = [u_2, u_3, u_4, u_5]^T$ and $\hat{y}_2 = [y_2, y_3, y_4, y_5]^T$.

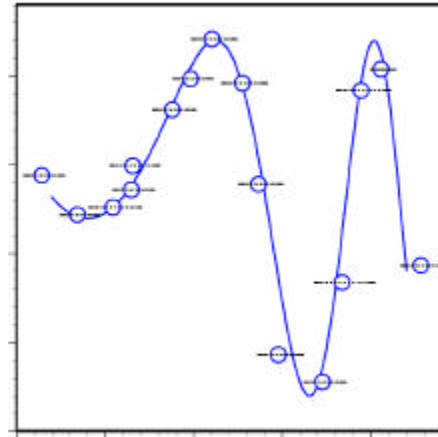


Fig. 1. Regression example. Solid line: true function; circles: samples; dots: 99% boundaries based upon input sample variances.

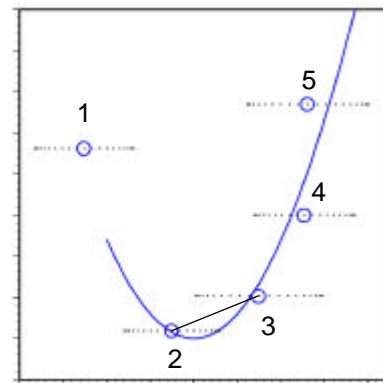


Fig. 2. Selection of measurements for calculation of the local-linear section 2-3. Based upon the 99% boundaries of the sample variances on the input, also samples 4 and 5 are used.

Since both $\sigma_{u,k} \neq 0$ and $\sigma_{y,k} \neq 0$ the iterative scheme (11) is used to calculate the local-linear sections. is based upon a Least Squares (LS) cost function. For the given example, the local-linear sections based upon (11) are depicted in Fig. 3. Remark that the sections don't necessarily form a first-order interpolation between the different measurements, since each section is based upon all measurements that lay within a "neighborhood". For recombination of the different local-linear sections, we choose an arbitrary value for the smoothing factor that equals $f_s = 2$. In the sequel of the paper a more detailed analysis of the smoothing factor is given. The results of the recombination are shown in Fig. 4. The solid line is again the true function, while the interpolated points are shown as crosses.

II. INTERPOLATION OF SPARSE DATA FOR SISO SYSTEMS

In order to simplify the explanation of the method used, consider a single-input single output (SISO) system with N input measurements

$$\mathbf{u} = [u_1, u_2, \dots, u_N]^T \in \mathbb{R}^N \quad (5)$$

and output measurements

$$\mathbf{y} = [y_1, y_2, \dots, y_N]^T \in \mathbb{R}^N. \quad (6)$$

Assumption 1: The input and output measurements are contaminated by additive noise

$$\begin{aligned} u_i &= u_{i,0} + \Delta u_i & \Delta u_i &\in \mathbb{R} \\ y_i &= y_{i,0} + \Delta y_i & \Delta y_i &\in \mathbb{R} \end{aligned} \quad (7)$$

in which the $u_{i,0}$ and $y_{i,0}$ are the true, but unknown input and output values and Δu_i and Δy_i are independent, mutually uncorrelated, zero mean Gaussian distributed random variables with standard deviations $\sigma_{u,i}$, $\sigma_{y,i}$ and $\sigma_{uy,i} = 0; \forall i$. \square

Assumption 2: The input measurements are ordered. This assumption can easily be met for single-input systems by sorting \mathbf{u} . For multi-input systems the data is expected to be sorted per dimension, i.e. that the measurements are performed over one dimension while the values for other dimensions are kept constant. If a multiple-input system cannot be measured following a strict grid, it is assumed that a (non-equidistant) grid can be given that encloses the data with a sufficient resolution. \square

As an example, assume a two-input system $y = f_{nl}(u_1, u_2)$ with $u_1 \in [-1, 1]$ and $u_2 \in [-2, 2]$ that is excited on a random basis. Then it is assumed that the experimenter can give a coarse grid (v_i, v_j) with $i = 1, 2, \dots, N_1$ and $\forall i; j = 1, 2, \dots, N_2$. The elements of the coarse grid $v_i \in [-1, 1]$ and $v_j \in [-2, 2]$ are each sorted.

The general idea behind the interpolation scheme is that the ordered measurements, described in assumption 2, are resampled using a fine grid. There is no need that the fine grid uses equidistant points. For the SISO case, the sorted measurement points are used as the grid points. The interpolation scheme starts with the calculation of the local linear sections. Since the inputs are sorted, it is very well possible that measurements skip the boundaries of a local linear section due to noise displacements. For that reason, all measurements that lay “in a neighborhood” of the calculated section are used for the calculation of the section parameters. This procedure is also used when an arbitrary

coarse grid has been chosen. The selection of the points that lay within a “neighborhood” is based upon the 95% or 99% percent intervals of the measurement points. For the 99% interval, this means that a measurement (u_i, y_i) participates to the calculation of the a local linear section if its 99% interval $u_{i,i} = [u_i - \sigma_{u,i}\Delta_{99}, u_i + \sigma_{u,i}\Delta_{99}]$ with $\Delta_{99} = 2.577$ overlaps with the segment $[u_k, u_{k+1}]$ to be interpolated ($\Delta_{95} = 1.955$ for the 95% interval). Define $\hat{\mathbf{u}}_k$ as the vector of all input measurements that lie in this segment and define $\hat{\mathbf{y}}_k$ as the corresponding vector of output measurements. The number of rows in the vector $\hat{\mathbf{u}}_k$ is denoted as P_k . In the case that $\sigma_{u,k} = 0$ and $\sigma_{y,k} = 0$, the calculation of the local-linear sections is based on a Least Squares (LS) cost function and the parameters of the linear system can easily be determined as

$$A_k = [\hat{\mathbf{u}}_k | 1_{P_k}] \backslash [\hat{\mathbf{y}}_k] \quad (8)$$

in which the operator “ \backslash ” solves the LS problem using QR decomposition and 1_{P_k} is a unity vector with the same number of rows as $\hat{\mathbf{u}}_k$. The $N-1$ vectors $A_k = [a_k, b_k]^T$, $k = 1, 2, \dots, N-1$, form the coefficients of a first order interpolation between the noiseless data points. In the case of exact input measurements and noisy output measurements ($\sigma_{u,k} = 0$ and $\sigma_{y,k} \neq 0$), the A_k are calculated using a Weighted Least Squares (WLS) cost function, such that (8) becomes

$$a_k = (\Sigma_y [\hat{\mathbf{u}}_k | 1_{P_k}]) \backslash (\Sigma_y [\hat{\mathbf{y}}_k]) \quad (9)$$

in which the matrix of output variances Σ_y is calculated as

$$\Sigma_y = \text{diag}([\sigma_{y,k}, \sigma_{y,k+1}]). \quad (10)$$

If both the inputs and outputs are noisy (case $\sigma_{u,k} \neq 0$ and $\sigma_{y,k} \neq 0$) the Errors-In-Variables (EIV) cost function is used. This results in an iterative scheme

$$a_{k,t+1} = (\Sigma_{uy,t} [\hat{\mathbf{u}}_k | 1_{P_k}]) \backslash (\Sigma_{uy,t} [\hat{\mathbf{y}}_k]) \quad (11)$$

with

$$\Sigma_{uy,t} = \text{diag}([\sqrt{\sigma_{y,1}^2 + a_{k,t}^2 \sigma_{u,1}^2}, \sqrt{\sigma_{y,2}^2 + a_{k,t}^2 \sigma_{u,2}^2}, \dots, \sqrt{\sigma_{y,P_k}^2 + a_{k,t}^2 \sigma_{u,P_k}^2}]). \quad (12)$$

The scheme is repeated for t until convergence is reached on the $a_{i,t}$ parameter. The initial parameter $a_{k,0}$ can be found by solving (9). Equation (11) has only one global minimum and can be solved in polynomial time.

- 3) **Regularization** adds a penalty term in the cost function. E. g. Sjöberg [22] adds an increasing penalty for an increasing number of parameters. The way how regularization is reached heavily depends on the specific properties of the penalty term (An [3]). The property that is usually asked from the penalty term, is a low variance of the BB model. Translating this into a regularization term is not a trivial task.
- 4) **Curvature-driven smoothing** (Bishop [5]) is based on a penalty term that uses the second order derivatives of the model. It is also called **Bias correction** and is described for linear systems (Amemiya [2]) and nonlinear systems (Santhanam, Vandersteen et al. [29]). It is our observation that bias correction leads to a vast number of local minima during the minimization of the cost function, while it has limited effects to overcome sparse data problems.
- 5) **Nearest-neighbor regression** (Geman et al. [9]) calculates a response output vector y for any input vector u , based on the average of the k measurements that are the closest to u . The value of k should be somewhere in between one and the total number of measurements to obtain a “reasonable” amount of smoothing. In this paper, a variant of the nearest-neighbor driven smoothing and a rule of thumb to select k based on the variance of the measurements is derived. A similar method is mentioned using splines in Geman et al. [9] and Mackay [14], but not worked out.
- 6) **Parzen-window regression** resembles the the nearest-neighbor regression, but the neighbor u_i of an input u is multiplied with a weight which is based on a Parzen-window

$$N(u) = \left(\frac{1}{\sqrt{2\pi}\sigma} \right)^d \exp\left(-\frac{1}{2} \frac{|u - u_i|^2}{\sigma^2} \right), \quad (4)$$

with d a smoothing factor. The choice of σ determines the allowable model variance (Bishop [5]). This paper also uses Parzen windows for the recombination of local-linear systems. The width of the windows will be based on 95% or 99% certainty bounds.

- 7) **Adding parameter noise** adds random noise to the weights of a BB model, i.e. its parameter space θ . The inevitable result of this is that the obtained parameters differ from the true minimizer of the cost function. It is shown that this results in regularization (An [3]), but it is highly doubtful that the optimal parameters θ^* that describe the true plant, are reached.
- 8) **Injecting data noise** (An [3]), also called jitter (Reed et al. [19]). The results of the doubtfull, but much used, technique of adding data noise inevitably leads to biasing of the NN parameters (Bishop [5]). The

technique of adding noisy input samples, which has become popular within NN, is actually a form of zero-order interpolation, and was proven to be of low effect on generalization (Van Gorp et al. [30]).

All of the above techniques demand a “smooth” behaviour of the resulting models in the case of sparse or noisy data. In practise, it is our finding that the only good technique to fill in gaps, is to provide extra “measurement” samples that fill the gap, based on a smoothing demand. Regardless of the choice of a too high number of BB model parameters, the model is forced to fit the data, thus decreasing variability. A well known technique for such interpolation is the use of splines. The drawback of using splines, however, is the lack of a physical meaning of the smoothing factor, and the fact that it is not defined for high-order dimensions of the input. Before introducing an interpolation method, let us therefore first define general demands for an interpolation technique:

- 1) The user must be able to impose a smoothing factor. This smoothing factor must have a physical meaning, such that the user has clear insight upon the limitations he puts on the model.
- 2) The interpolation technique must be useable for noisy and noiseless measurement data. In the case of noiseless data, an interpolation from one point to the other should be possible. In the case of noisy data, the interpolation should make use of the variance of the individual measurement points.
- 3) The interpolation should be applicable for models with a higher dimension of the inputs (Multi-Input models).
- 4) The variance of the interpolated data points must be lower than, or equal to, the variance of the measurement data.
- 5) The overall mapping of the interpolation technique must be unbiased, if no special smoothing is demanded.
- 6) It is preferred that the interpolation technique is useable with non-equidistant grids. The backlaying idea is that certain regions can be fitted in more detail.
- 7) It is preferred that the technique is feedforward, i.e. that the interpolation technique doesn’t suffer from local minima that inheritantly come with minimizing complex cost functions.
- 8) When used for extrapolation (instead of interpolation), the method must not “explode” the data, i.e. by resorting to a linear extrapolation. While the risk for extrapolation is low for SISO systems, this demand is certainly necessary for high-dimensional multi-input systems that are not persistently measured.
- 9) It is preferred that the interpolation technique has low memory and calculation needs. Since the goal is interpolation, an increase in complexity of $O(x^n)$ with n the input dimension, is acceptable.

An Interpolation Technique for Learning With Sparse Data

Jürgen Van Gorp* and Yves Rolain**

Abstract — The occurrence of sparse data is a stringent problem for Black Box system identification. The lack of measurement data leads to a high modeling error in the resulting gaps. There are a number of techniques described in the literature to fill in these gaps. Some of these techniques are only useable for a low input count (splines). Others suffer from noisy data (interpolation), severe parameter biasing (adding input noise) or time-consuming recursive calculations (Errors-In-Variables). This paper describes a linear feedforward interpolation technique that uses probability techniques, based on 95% or 99% certainty bounds. A smoothing factor can be used, based upon probability theory.

Keywords — System Identification, Sparse Data, Interpolation, Modeling.

I. INTRODUCTION

Consider a general Multi-Input Multi-Output sampled data system F with m inputs and n outputs. Consider also that the m inputs and n outputs are sampled N times simultaneously at a sampling rate T_s . At the k -th sampling instant the vectors

$$\mathbf{u}_k = [u_{1,k}, u_{2,k}, \dots, u_{m,k}] \in \mathbb{R}^m \quad (1)$$

and

$$\mathbf{y}_k = [y_{1,k}, y_{2,k}, \dots, y_{n,k}] \in \mathbb{R}^n \quad (2)$$

represent the sampled data. In estimation, the input-output pairs $(\mathbf{u}_k, \mathbf{y}_k)$ are used to identify the parameters θ of a linear or nonlinear model f_{bb} . In this paper a Black Box (BB) model

$$f_{BB}: \mathbb{R}^m \rightarrow \mathbb{R}^n: \mathbf{y} = f_{BB}(\mathbf{u}, \theta) \quad (3)$$

is chosen. Consider the problem where a plant is persistently excited but not regularly measured, i.e. gaps remain in between the measurements. The goal of the measurements is to identify a black box (BB) model, such as a Neural

Network (NN), Radial Basis Function Network (RBFN) or Fuzzy Logic System (FLS). It is a well known problem that black box models can have an unpredictable behaviour (high variability) in the measurement gaps.

Definition 1: In the sequel of this paper, the following definitions are used to describe the behaviour of a nonlinear mapping. NN mappings are said to have a large **variability** if different NN mappings tend to have a large variance when compared to each other. The **variance** of a BB model is used when a NN mapping has large variations compared to the true function, e.g. caused by overfitting on noisy data. A NN mapping is called **biased** when the mean output of an infinite number of NN mappings does not map the true function. \square

The most direct way to prevent a high variability of the BB models, is to perform better experiments and fill in the gaps in the measurements. This is not always possible, such that the experimenter is deemed to make some assumptions concerning the plant or controller. In most cases a smoothness demand is put upon the system, usually called “generalization”. The most used techniques for generalization, are

- 1) The use of a **validation set**, also called early stopping (Sjöberg [22] [23]). For early stopping typically large and representative sets are needed (An [3]), and gaps in the measurements are worsened, since the data has to be split in multiple data sets.
- 2) **Model selection methods** (Amari et al. [1], Twomey et al. [27]) The idea is that an optimal choice is made on the number of parameters in the model. Although this seems a trivial solution, this selection is an open problem for BB models. E.g. in case of Neural Networks, typically a large number of neurons is chosen to improve the possibility to fit details in the data. The result is an increased danger for high variability. The problem is then to reduce the number of neurons and to suppress non-significant parameters without compromising the quality of the modeling.

The authors are with the Vrije Universiteit Brussel, Pleinlaan 2, B-1050 Brussel, Belgium, E-mail: *Jurgen.Van.Gorp@vub.ac.be **Yves.Rolain@vub.ac.be