Sparse Regression Modelling Using an Incremental Weighted Optimization Method Based on Boosting with Correlation Criterion

S. Chen ‡, X.X. Wang † and D.J. Brown †
‡ School of Electronics and Computer Science
University of Southampton, Southampton SO17 1BJ, U.K.
E-mail: sqc@ecs.soton.ac.uk

† Department of Creative Technologies
University of Portsmouth, Portsmouth PO1 3HE, U.K.

ABSTRACT

A novel technique is presented to construct sparse Gaussian regression models. Unlike most kernel regression modelling methods, which restrict kernel means to the training input data and use a fixed common variance for all the regressors, the proposed technique can tune the mean vector and diagonal covariance matrix of individual Gaussian regressor to best fit the training data based on the correlation between the regressor and the training data. An efficient repeated weighted optimization method is developed based on boosting with the correlation criterion to append regressors one by one in incremental regression modelling. Experimental results obtained using this construction technique demonstrate that it offers a viable alternative to the existing state-of-art kernel modelling methods for constructing parsimonious regression models.

Index Terms — Regression, construction algorithm, correlation, mean square error, boosting

1 INTRODUCTION

A basic principle in practical nonlinear data modelling is the parsimonious principle of ensuring the smallest possible model that explains the training data. Forward selection using the orthogonal least square (OLS) algorithm [1]–[4] is popular for nonlinear data modelling practicians, for the reason that the algorithm is simple and efficient, and is capable of producing parsimonious linear-in-the-weights nonlinear models. Recently, the state-of-art sparse kernel modelling techniques, such as the support vector machine and relevant vector machine [5]–[7], have widely been adopted in data modelling applications. In most of these sparse regression modelling techniques, a fixed common variance is used for all the regressor kernels and the kernel centers or means are placed at the training input data.

We present a flexible construction method for Gaussian kernel models. The correlation between a Gaussian regressor and the training data is used as the criterion in positioning (mean adjustment) and shaping (diagonal covariance adjustment) the regressor. To incrementally append regressor one by one, a repeated weighted optimization search algorithm is developed, which is based on the idea from boosting [8]–[10]. Because kernel means are not restricted to the training input data and each regressor has an individually tuned diagonal covariance matrix, our method can produce very sparse models. The proposed repeated weighted optimization algorithm is simple, robust and easy to implement. Experimental results are used to demonstrate the effectiveness of this incremental construction algorithm.

2 THE PROPOSED CONSTRUCTION METHOD

We consider the modelling problem of approximating the N pairs of training data $\{\mathbf{x}_l, y_l\}_{l=1}^N$ with the regression model

$$\hat{y}(\mathbf{x}) = \sum_{i=1}^{M} w_i g_i(\mathbf{x}) \tag{1}$$

where $\mathbf{x}$ is the m-dimensional input variable; w_i, $1 \leq i \leq M$, denote the model weights; M is the number of regressors; and $g_i(\bullet)$, $1 \leq i \leq M$, denote the regressors. We allow the regressor to be chosen as the general Gaussian function $g_i(\mathbf{x}) = G(\mathbf{x}; \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$ with

$$G(\mathbf{x}; \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i) = e^{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_i)^T \boldsymbol{\Sigma}_i^{-1} (\mathbf{x}-\boldsymbol{\mu}_i)} \tag{2}$$

where $\boldsymbol{\Sigma}_i = \text{diag}\{\sigma_{i,1}^2, \cdots, \sigma_{i,m}^2\}$. We will adopt an incremental approach to build up the regression model (1) by appending regressors one by one.

2.1 Correlation criterion for selecting regressor

Let us first introduce the following notation

$$\left.\begin{array}{l} y_i^{(0)} = y_i \\ y_i^{(k)} = y_i^{(k-1)} - w_k g_k(\mathbf{x}_i) \end{array}\right\} 1 \leq i \leq N \tag{3}$$

Obviously, $y_i^{(k)}$ is the modelling error at $\mathbf{x}_i$ after the kth regressor has been fitted. At the kth stage of incremental modelling, the regressor $g_k(\mathbf{x})$ is fitted to the training data set $\{y_i^{(k-1)}, \mathbf{x}_i\}_{i=1}^N$. The correlation function between the regressor and the training data set given by

$$C_k = \frac{\sum_{i=1}^N g_k(\mathbf{x}_i) y_i^{(k-1)}}{\sqrt{\sum_{i=1}^N g_k^2(\mathbf{x}_i)} \sqrt{\sum_{i=1}^N \left(y_i^{(k-1)}\right)^2}} \quad (4)$$

defines the "similarity" between the regressor $g_k(\mathbf{x})$ and the training data $\{y_i^{(k-1)}, \mathbf{x}_i\}_{i=1}^N$. The larger value of $|C_k|$ is, the more similar they are.

Regressor positioning and shaping. With the Gaussian regressor $g_k(\bullet) = G(\bullet; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$, the correlation C_k is a function of the kernel mean $\boldsymbol{\mu}_k$ and covariance matrix $\boldsymbol{\Sigma}_k$. Thus, the correlation criterion (4) can be used for positioning and shaping $g_k(\bullet)$ by maximizing $|C_k|$ with respect to $\boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}_k$.

Calculation of the model weight. After the determination of the kth regression $g_k(\mathbf{x})$, the corresponding model weight w_k is calculated by minimizing the mean square error (MSE) for the k-term regression model

$$MSE_k = \frac{1}{N} \sum_{i=1}^N \left(y_i^{(k-1)} - w_k g_k(\mathbf{x}_i)\right)^2 \quad (5)$$

This leads to the usual least square solution

$$w_k = \frac{\sum_{i=1}^N y_i^{(k-1)} g_k(\mathbf{x}_i)}{\sum_{i=1}^N g_k^2(\mathbf{x}_i)} \quad (6)$$

We now prove that selecting regressors by incrementally maximizing $|C_k|$ is equivalent to that by incrementally minimizing the modeling MSE (5). In fact, substituting w_k in (5) by the least square solution (6), it can be shown that

$$MSE_k = \left(\frac{1}{N} \sum_{i=1}^N \left(y_i^{(k-1)}\right)^2\right) \left(1 - C_k^2\right) \quad (7)$$

Obviously, maximizing $|C_k|$ with respect to $\boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}_k$ is identical to minimizing MSE_k with respect to $\boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}_k$.

2.2 Repeated weighted search method

It is seen that at each increment regression stage, the basic task is to maximize some function $f(\mathbf{u})$ over $\mathbf{u} \in U$, where $f(\mathbf{u}) = |C_k(\mathbf{u})|$, and $\mathbf{u}$ contains $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$. We use the following simple search method to perform this optimization. Given s points of $\mathbf{u}$, $\mathbf{u}_1, \cdots, \mathbf{u}_s$, let $\mathbf{u}_{best} = \arg\max\{f(\mathbf{u}_i), 1 \le i \le s\}$ and $\mathbf{u}_{worst} = \arg\min\{f(\mathbf{u}_i), 1 \le i \le s\}$. A $(s+1)$th value is generated by a weighted combination of $\mathbf{u}_i$, $1 \le i \le s$. Because this weighted combination is a convex combination, the point $\mathbf{u}_{s+1}$ is always within the convex hull defined by the s values. A $(s+2)$th value is then generated as the mirror image of $\mathbf{u}_{s+1}$, with respect to $\mathbf{u}_{best}$, along the direction defined by $\mathbf{u}_{best} - \mathbf{u}_{s+1}$. The best of $\mathbf{u}_{s+1}$ and $\mathbf{u}_{s+2}$ then replaces $\mathbf{u}_{worst}$. The process is repeated until it converges. Obviously, the weighting values used to perform this combination are critical, and we adopt the idea of boosting [8]-[10] to adapt these weightings. This leads to the following basic weighted optimization algorithm.

WeightedOp

Initialization: Given $\{\mathbf{x}_i, y_i^{(k-1)}\}_{i=1}^N$ and the s initially chosen values for $\mathbf{u}$, $\mathbf{u}_1, \mathbf{u}_2, \cdots, \mathbf{u}_s$, set iteration index $t = 0$ and $\delta_i^{(t)} = \frac{1}{s}$ for $1 \le i \le s$.

Step 1: Boosting

1. Calculate the loss of each point, namely

$$cost(j) = 1 - |C_k(\mathbf{u}_j)|, \ \ 1 \le j \le s$$

2. Find $\mathbf{u}_{best} = \arg\min\{cost(j), \ 1 \le j \le s\}$

$$\mathbf{u}_{worst} = \arg\max\{cost(j), \ 1 \le j \le s\}$$

3. Normalize the loss

$$loss(j) = \frac{cost(j)}{\sum_{l=1}^s cost(l)}, \ 1 \le j \le s$$

4. Compute a weighting factor β_t according to

$$\epsilon_t = \sum_{j=1}^s \delta_j^{(t)} loss(j) \quad \beta_t = \frac{\epsilon_t}{1 - \epsilon_t}$$

5. Update the weighting vector

$$\delta_j^{(t+1)} = \begin{cases} \delta_j^{(t)} \beta_t^{loss(j)} & \text{for } \beta_t \le 1, \\ \delta_j^{(t)} \beta_t^{1-loss(j)} & \text{for } \beta_t > 1, \end{cases}$$

where $j = 1, \cdots, s$

6. Normalize the weighting vector

$$\delta_j^{(t+1)} = \frac{\delta_j^{(t+1)}}{\sum_{l=1}^s \delta_l^{(t+1)}}, \ 1 \le j \le s$$

Step 2: Parameter updating

1. Construct the $(s+1)$th point using the formula

$$\mathbf{u}_{s+1} = \sum_{i=1}^s \delta_i^{(t+1)} \mathbf{u}_i$$

2. Construct the $(s+2)$th point using the formula

$$\mathbf{u}_{s+2} = \mathbf{u}_{best} + (\mathbf{u}_{best} - \mathbf{u}_{s+1})$$

3. Choose a better point from $\mathbf{u}_{s+1}$ and $\mathbf{u}_{s+2}$ to replace $\mathbf{u}_{worst}$.

Repeat from *Step 1* until the $(s+1)$th value changes very little compared with the last round, or a preset maximum number of iterations has been reached.

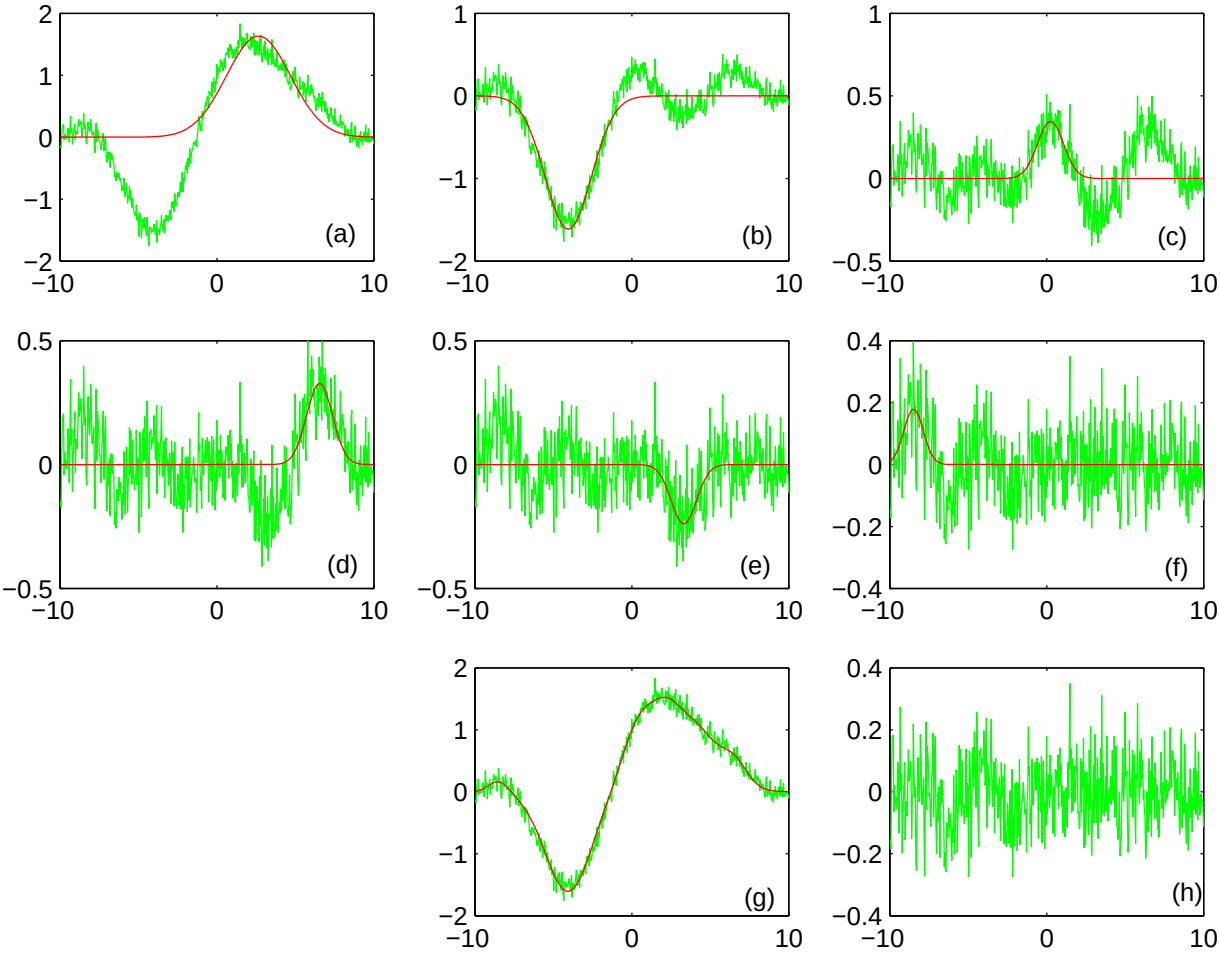


Fig. 1. Incremental modelling results for Example 1: in (a)–(f), the light curves are the modelling errors of the previous stage, $y_i^{(k-1)}$, and the dark curves are the fitted current regressors, $w_k g_k(x_i)$, for $1 \leq k \leq 6$, respectively; in (g), the light curve is the noisy training data y_i and the dark curve is the final 6-term model $\hat{y}$; and (h) shows the final modelling errors.

The above **WeightedOp** algorithm performs a guided random search and solution obtained may depend on the initial choice of the population. To derive a robust algorithm that ensures a stable "global" solution, we simply augment the algorithm into the following **ReWeightedOp** algorithm.

ReWeightedOp

Repeat loop: For $l = 1 : M_R$
Set $\mathbf{u}_1^{(l)} = \mathbf{u}_{best}^{(l-1)}$, and randomly generate the other $s-1$ points $\mathbf{u}_i^{(l)}$ for $2 \leq i \leq s$
Call the **WeightedOp** algorithm to obtain a solution $\mathbf{u}_{best}^{(l)}$
If $\|\mathbf{u}_{best}^{(l-1)} - \mathbf{u}_{best}^{(l)}\| < \xi_r$, Exit loop
End for
The solution is $\mathbf{u}_{best}^{(l)}$

In the **ReWeightedOp**, M_R is the number of maximum repeating runs and ξ_r the termination threshold value.

2.3 Incremental regression modelling

The completed algorithm for incremental regression modelling can now be summarized. Choose a preset modelling accuracy ξ, and set $k = 0$.

Do: $k = k + 1$
1. Call **ReWeightedOp** to determine $\boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}_k$, the position and diagonal covariance matrix of the kth regressor
2. Calculate the weight w_k for the kth regressor according to (6) and compute the modelling errors $y_i^{(k)} = y_i^{(k-1)} - w_k g_k(\mathbf{x}_i)$, $1 \leq i \leq N$
While $MSE_k > \xi$

The termination of the model construction process can also be decided using cross validation [11],[12]. A simple method is to have a separate validation data set. The model construction is based on the training data set, while the performance of the selected model, the MSE, is monitored over the validation data set. The construction process is terminated when the MSE over the

TABLE I

INCREMENTAL MODELLING PROCEDURE FOR EXAMPLE 1.

regression step k	mean μ_k	variance σ_k^2	weight w_k	MSE MSE_k
0	–	–	–	0.8431
1	2.6905	4.2488	1.6088	0.3703
2	-4.0837	2.1853	-1.6019	0.0341
3	0.2982	0.6000	0.3781	0.0243
4	6.6062	0.6610	0.3116	0.0173
5	3.4162	0.6091	-0.2242	0.0138
6	-8.4780	0.4295	0.1787	0.0119

validation data set stops improving. Alternatively, the Akaike information criterion [13], the optimal experimental design criteria [4] and the leave-one-out generalization criterion [14] may be adopted to automatically terminate the model construction process without the need for a separate validation data set.

3 EXPERIMENTAL RESULTS

Example 1. The 500 points of training data were generated from

$$y(x) = 0.1x + \frac{\sin x}{x} + \sin 0.5x + e$$

with $x \in [-10,\ 10]$, where e was a Gaussian white noise with zero mean and variance 0.01. The population size and the number of maximum repeating times used in **ReWeightedOp** were $s = 9$ and $M_R = 10$. With the modelling accuracy set to $\xi = 0.012$, the incremental regression modelling produced 6 Gaussian regressors, as summarized in Table I, and the construction process is also illustrated graphically in Fig. 1 (a)–(f). In Fig. 1 (g), the model output from the constructed 6-term model is superimposed on the noisy training data, and the final modelling errors are shown in Fig. 1 (h).

Example 2. This example constructed a model for the gas furnace data set (Series J in [15]). The data set contained 296 pairs of input-output points, where the input $u(t)$ was the coded input gas feed rate and the output $y(t)$ represented CO_2 concentration from the gas furnace. The input-output data are depicted in Fig. 2 (a) and (b), respectively. The training data set was constructed with $y_i = y(i)$ and

$$\mathbf{x}_i = [y(i-1)\, y(i-2)\, y(i-3)\, u(i-1)\, u(i-2)\, u(i-3)]^T$$

for $i = 4, 5, \cdots, 296$. In the previous work [14], it was found out that various existing state-of-art kernel modeling techniques required at least 28 model terms to achieve a modelling accuracy of $\xi = 0.054$ for this data set.

With a population size $s = 147$ and the number of maximum repeating times $M_R = 20$ in **ReWeightedOp**, and a preset modelling accuracy of $\xi = 0.054$, the proposed incremental modelling procedure produced 18 Gaussian regressors, and the resulting model is listed in Table II. Fig. 2 (b) depicts the model prediction $\hat{y}(t)$ for this 18-term model, in comparison with the system output $y(t)$. The corresponding model prediction error $\epsilon(t) = y(t) - \hat{y}(t)$ is shown in Fig. 2 (c). It is seen that the proposed approach is able to produce a sparser regression model with an equally good modeling performance over the various existing state-of-art kernel modeling methods.

4 CONCLUSIONS

A construction algorithm has been proposed to incrementally fit sparse Gaussian regression models. The algorithm has the ability to tune the mean vector and diagonal covariance matrix of individual Gaussian regressor to best fit the training data based on the correlation between the regressor and the training data. A repeated weighted optimization search method has been developed based on boosting with the correlation criterion to append regressors one by one in incremental regression modelling. Experimental results presented have demonstrated the effectiveness of the proposed technique.

REFERENCES

[1] S. Chen, S.A. Billings and W. Luo, "Orthogonal least squares methods and their application to non-linear system identification," *Int. J. Control*, Vol.50, No.5, pp.1873–1896, 1989.

[2] S. Chen, C.F.N. Cowan and P.M. Grant, "Orthogonal least squares learning algorithm for radial basis function networks," *IEEE Trans. Neural Networks*, Vol.2, No.2, pp.302–309, 1991.

[3] S. Chen, Y. Wu and B.L. Luk, "Combined genetic algorithm optimisation and regularised orthogonal least squares learning for radial basis function networks," *IEEE Trans. Neural Networks*, Vol.10, No.5, pp.1239–1243, 1999.

[4] S. Chen, X. Hong and C.J. Harris, "Sparse kernel regression modelling using combined locally regularized orthogonal least squares and D-optimality experimental design," *IEEE Trans. Automatic Control*, Vol.48, No.6, pp.1029–1036, 2003.

[5] V. Vapnik, *The Nature of Statistical Learning Theory*. New York: Springer-Verlag, 1995.

[6] V. Vapnik, S. Golowich and A. Smola, "Support vector method for function approximation, regression estimation, and signal processing," in: M.C. Mozer, M.I. Jordan and T. Petsche, Eds., *Advances in Neural Information Processing Systems 9*. Cambridge, MA: MIT Press, 1997, pp.281–287.

[7] M.E. Tipping, "Sparse Bayesian learning and the relevance vector machine," *J. Machine Learning Research*, Vol.1, pp.211–244, 2001.

[8] Y. Freund and R.E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *J. Computer and System Sciences*, Vol.55, No.1, pp.119–139, 1997.

[9] R.E. Schapire, "The strength of weak learnability," *Machine Learning*, Vol.5, No.2, pp.197–227, 1990.

[10] R. Meir and G. Rätsch, "An introduction to boosting and leveraging," in: S. Mendelson and A. Smola, eds., *Advanced Lectures in Machine Learning*. Springer Verlag, 2003, pp.119–184.

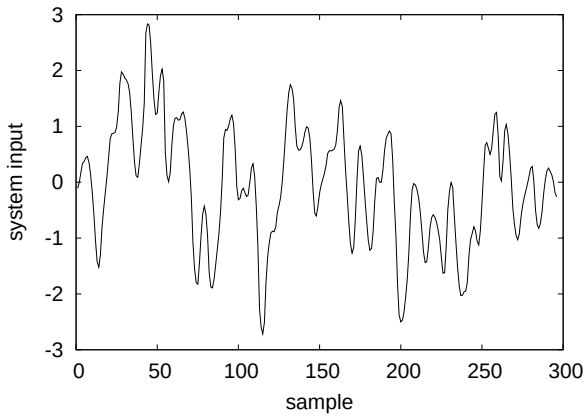
[11] M. Stone, "Cross validation choice and assessment of statistical predictions," *J. Royal Statistics Society Series B*, Vol.36, pp.117–147, 1974.

[12] R.H. Myers, *Classical and Modern Regression with Applications*. 2nd Edition, Boston: PWS-KENT, 1990.

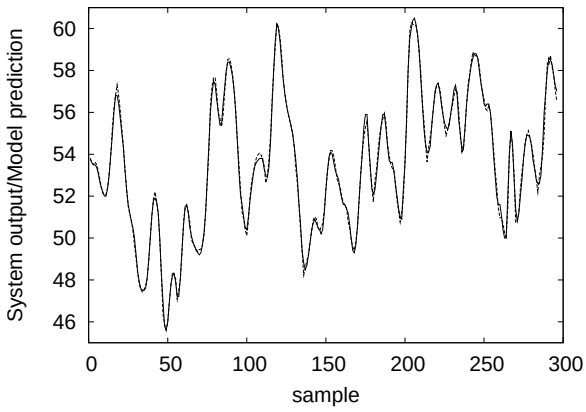
[13] H. Akaike, "A new look at the statistical model identification," *IEEE Trans. Automatic Control*, Vol.AC-19, pp.716–723, 1974.

[14] S. Chen, X. Hong, C.J. Harris and P.M. Sharkey, "Sparse modelling using orthogonal forward regression with PRESS statistic and regularization," *IEEE Trans. Systems, Man and Cybernetics, Part B*, Vol.34, No.2, pp.898–911, 2004.

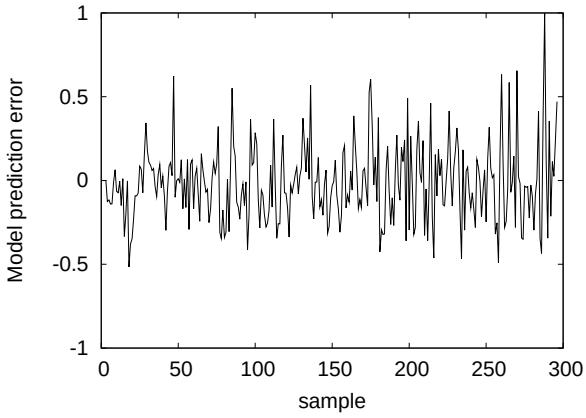
[15] G.E.P. Box and G.M. Jenkins, *Time Series Analysis, Forecasting and Control*. Holden Day Inc., 1976.



(a)



(b)



(c)

Fig. 2. Example 2, the gas furnace data set: (a) system input $u(t)$, (b) model output $\hat{y}(t)$ (dashed) superimposed on system output $y(t)$ (solid), and (c) model prediction error $\epsilon(t) = y(t) - \hat{y}(t)$.

TABLE II

INCREMENTAL MODELLING PROCEDURE FOR EXAMPLE 2.

k	mean vector $\boldsymbol{\mu}_k$ diagonal covariance matrix $\boldsymbol{\Sigma}_k$						weight w_k	MSE_k
0	–						–	2.8443×10^3
	–							
1	61.0000	61.0000	50.9525	3.3340	-3.2160	-3.2160	59.276	0.8268
	6.0783	25.0000	25.0000	16.6549	2.4340	9.1178		
2	60.2279	45.1000	45.1000	3.3340	3.3340	3.3340	4.0451	0.5470
	0.0217	5.9837	2.8963	23.3208	19.4850	9.1388		
3	46.8987	58.9860	57.1320	-1.2067	-2.4251	0.3321	-71.404	0.2941
	0.0936	0.0762	6.2630	0.0322	6.7020	0.0262		
4	45.1000	45.1000	45.1000	3.3340	-3.2160	3.3340	2.4931	0.1389
	5.7572	0.0476	7.9013	19.0917	12.3381	10.1693		
5	57.9404	51.5699	47.5973	-3.2160	3.3340	3.3340	29.190	0.1138
	0.0293	22.1583	0.0476	11.1085	8.9510	18.3801		
6	58.0935	45.1000	45.1000	3.3340	3.3340	1.9810	58.659	0.1065
	0.0101	0.2753	3.8145	24.6911	0.0512	0.0164		
7	56.3757	49.5204	60.1222	2.2450	2.5864	-1.2469	21.472	0.0972
	25.0000	0.0100	25.0000	0.0100	0.0123	0.0100		
8	45.1000	45.1000	51.6922	-0.0144	-3.2160	3.3340	-5.2669	0.0804
	0.1052	1.5136	0.0501	0.0100	0.0550	11.9954		
9	46.0303	58.7537	54.9481	-2.6996	1.5488	-0.2903	-1.4557	0.0760
	25.0000	0.0385	25.0000	0.0100	25.0000	0.0100		
10	58.7624	55.1555	54.8745	2.5233	2.2267	-1.6638	46.918	0.0700
	0.0100	0.0100	0.1178	0.0113	25.0000	24.5476		
11	53.9952	54.8025	52.8236	-1.7173	3.1207	2.1370	-1.7379	0.0675
	25.0000	0.0100	0.0100	25.0000	25.0000	0.0100		
12	45.1000	47.1594	61.0000	3.3340	3.3340	3.3340	2.1679	0.0631
	25.0000	0.0620	25.0000	0.0100	0.0100	25.0000		
13	54.7481	48.7561	50.0140	-0.9630	0.8991	1.1693	48.259	0.0589
	0.0100	0.0100	0.0100	15.7145	0.0100	25.0000		
14	46.2857	46.4039	54.9041	-2.7555	-2.7218	2.6338	1.5930	0.0578
	25.0000	0.0992	0.0100	25.0000	25.0000	25.0000		
15	53.5265	49.5718	48.3381	1.9158	0.1292	1.7822	7.4529	0.0570
	0.0100	0.0100	25.0000	0.0100	0.0100	0.0100		
16	55.4362	54.9274	57.2423	0.1605	0.1057	1.8326	-0.9918	0.0550
	0.0100	0.0100	25.0000	0.0100	0.0100	0.0100		
17	53.7756	51.3648	51.6104	0.7165	-0.1010	0.6642	-0.0845	0.0542
	25.0000	0.0100	0.0100	25.0000	25.0000	25.0000		
18	45.1000	45.1000	45.1000	-3.2160	-3.2160	-3.2160	0.2511	0.0538
	1.0087	17.3402	25.0000	10.6542	25.0000	0.0100		