Sparse Kernel Regression Modelling Based on L1 Significant Vector Learning

Junbin Gao
School of Information Technology
Charles Sturt University
Bathurst, NSW 2795, Australia
E-mail: jbgao@csu.edu.au

Daming Shi
School of Computer Engineering
Nanyang Technological University
Singapore 639798
E-mail: asdmshi@ntu.edu.sg

***Abstract*— A novel L1 significant vector (SV) regression algorithm is proposed in the paper. The proposed regularized L1 SV algorithm finds the significant vectors in a successive greedy process. The performance of the proposed algorithm is comparable to the OLS algorithm while it saves a lot of time complexities in implementing orthogonalization needed in the OLS algorithm.**

I. INTRODUCTION

In practical nonlinear data modelling more and more interests are paid to the basic principle of parsimonious models that ensure the smallest possible model that explains the data well. Apart from obvious computational advantage, small models often generalize better for the unseen data. In recent years the support vector machine (SVM) [18] and kernel machine models (KMM) [14], [3] considerably attract one's interests. These techniques have been gaining more and more popularity and have been regarded as the state-of-art technique for regression and classification problems with tremendously successful applications in many areas. The theoretical fundamental of SVM is the structural risk minimization principle which results in excellent generalization properties with a sparse model representation [13]. However it has been shown that the standard SVM technique is not always able to construct parsimonious models in system identification [8]. This inadequateness motivates exploring new methods for the parsimonious models under the framework of both SVM and KMM. Tipping [16] first introduced the relevance vector machine (RVM) method which can be viewed from a Bayesian learning framework of kernel machine and produces an identical functional form to the SVM/KMM. The results given in [16] have demonstrated that the RVM has a comparable generalization performance to the SVM but requires dramatically fewer kernel functions or model terms than the SVM. A drawback of the RVM algorithm is a significant increase in computational complexity, compared with the SVM method. Recently Chen et al [2], [3], [7] derived a novel method for constructing sparse kernel models based on his orthogonal least squares (OLS) algorithm [5], [6] and kernel techniques [14]. The OLS algorithm has been demonstrated as efficient learning procedure for constructing sparse regression models and gives good performances in nonlinear system identification. There are a lot of literatures concerning the problem of regressor selection, see for example [12], [10], [17], [15].

The OLS algorithm involves sequential selection of the regressors, which ensures that each new regressor vector defined on the training data is orthogonal to the previous selections. It employs the well-known Gram-Schmidt orthogonalization method in applied mathematics. In choosing the best regressor, the contribution of each regressor to the modelling error decrease is measured. Each chosen regressor maximally decreases the squared error of the model output, and the method stops when this error reaches an acceptable level or when the desired number of regressors have been chosen. Following the same idea, we found that orthogonalization procedure employed by the OLS algorithm can be removed and a comparable training result can be achieved, so that we can save a lot of computational complexity in the training procedure, see [9].

In this paper, a new approach is proposed to determine the regressors of the kernel regression modelling based on the so-called L1 significant vectors (SV). The rest of this paper is organized as follows: In section 2, the concepts of L1 significant vector analysis are given and the algorithm for finding L1 significant vectors is presented. The experiments are carried out in section 3, followed by our conclusions in Section 4.

II. THE L1 SIGNIFICANT VECTORS

To introduce our method, we follow the notations used in [7].

Consider the general discrete-time nonlinear system represented by the nonlinear model [4]:

$$\begin{aligned} y(k) =& f(y(k-1), ..., y(k-n_y), \\ & u(k-1), ..., u(k-n_u)) + e(k) \\ =& f(\mathbf{x}(k)) + e(k), \end{aligned} \tag{1}$$

where $u(k)$ and $y(k)$ are the system input and output variables, respectively, n_y and n_u are positive integers representing the lags in $y(k)$ and $u(k)$, respectively, $e(k)$ is the system noise, $\mathbf{x}(k) = [y(k-1), ..., y(k-n_y), u(k-1), ..., u(k-n_u)]^T$ denotes the system "input" vector, and f is the unknown

0-7803-9422-4/05/$20.00 ©2005 IEEE

system mapping. The system identification involves in construct a function (model) to approximate the unknown mapping f based on an N-sample observation data set $\mathbf{D} = \{\mathbf{x}(k), y(k)\}_{k=1}^{N}$, i.e., the system input-output observation data $\{u(k), y(k)\}$. The most popular class of such approximating functions is the kernel regression model of the form:

$$y(k) = \widehat{y}(k) + e(k) = \sum_{i=1}^{N} w_i \phi_i(k) + e(k), \qquad 1 \le k \le N \tag{2}$$

where $\widehat{y}(k)$ denotes the "approximated" model output, w_i's are the model weights, and $\phi_i(k) = k(\mathbf{x}(i), \mathbf{x}(k))$ are the regressors generated from a given kernel function $k(\mathbf{x}, \mathbf{y})$, see [14]. If we choose $k(\mathbf{x}, \mathbf{y})$ as the Gaussian kernel, then (2) describes a RBF network with each data as a RBF center and a fixed RBF width. The model (2) can be made more general if we choose each ϕ_i as different function regressors, such that it can include, for example, all the kernel based models, the polynomial-based models and all the generalized linear nonlinear model (i.e., linear-in-the-weight models). But in this paper we will focus on the case in which all the regressors ϕ_i are generated from a single kernel function just as defined in (2). Our analysis in this paper can be easily applied to all the other cases.

Let

$$\Phi_i = [\phi_i(1), ..., \phi_i(N)]^T = [k(\mathbf{x}(i), \mathbf{x}(1)), ..., k(\mathbf{x}(i), \mathbf{x}(N))]^T; \tag{3}$$

$$\mathbf{\Phi} = [\Phi_1, ..., \Phi_N];$$
$$\mathbf{w} = [w_1, w_2, ..., w_N]^T;$$
$$\mathbf{y} = [y(1), y(2), ..., y(N)]^T;$$
$$\mathbf{e} = [e(1), e(2), ..., e(N)]^T$$

then the regression model (2) can be written in the following matrix form

$$\mathbf{y} = \mathbf{\Phi}\mathbf{w} + \mathbf{e}. \tag{4}$$

The goal of modelling data is to find the best linear combination of the columns of $\mathbf{\Phi}$ (i.e. the best value for $\mathbf{w}$) to explain $\mathbf{y}$ according to some criterium. The popular criterium is to minimize the sum of squared errors,

$$E = \mathbf{e}^T\mathbf{e} \tag{5}$$

where the solution $\mathbf{w}$ is called the least squares solution to the above model, which is equivalent to finding the orthogonal projection of the target vector $\mathbf{y}$ in the subspace spanned by all the regressor vectors Φ_i given by (3).

In this paper we are interested in the cost criterium

$$E_{L1} = |\mathbf{e}| = \sum_{k=1}^{N} |e(k)|. \tag{6}$$

instead of using squared errors. What interests us in the above model is that the model can be used to fit and explain the Laplacian noises in $\mathbf{e}$ rather than a Guassian noise. In the case of outliers on the data or non-Gaussian noise distributions with heavy tails it becomes important then to reduce the influence of the outliers or points of the tails on the estimate. For this purpose we investigate the L1 error criterium defined in (6)

Using the L1 error criterium increases the burden of optimizing E_{L1} with respect to the weight vector $\mathbf{w}$. However as we pointed out in [9] by analyzing the OLS algorithm, the regressor vector should give the maximal similarity to the target vectors that mostly explain the data under a given criterium. This suggests that we could define a set of L1 significant vectors which mostly explain the data in a forward greedy procedure.

At the beginning the set of L1 significant vectors is empty. Denote $\mathbf{y}^{(0)} = \mathbf{y}$. Then for each regressor vector Φ_i from the columns of $\mathbf{\Phi}$ we solve a one-parameter L1 estimate problem defined as

$$\min_{\omega_i^{(1)}} |\mathbf{y} - \omega_i^{(1)}\Phi_i| = \sum_{k=1}^{N} |y(k) - \omega_i^{(1)}\Phi_i(k)|$$

Suppose the solution to the above L1 regression is $\omega_i^{(1)^*}$. Then find the first significant regressor vector Φ_{i_1} from

$$\Phi_{i_1} = \min_{\Phi_i \in \mathbf{\Phi}} |\mathbf{y} - \omega_i^{(1)^*}\Phi_i|$$

Now drop Φ_{i_1} from the columns of $\mathbf{\Phi}$ and denote the remnant by $\mathbf{\Phi}^{(1)}$. Then the second significant regressor vector will be chosen from the columns of the new $\mathbf{\Phi}^{(1)}$. As there is no orthogonality between the columns of $\mathbf{\Phi}$, the selection criterium should be rectified. It is better for us to choose the second significant regressor vector such that it will mostly explain the residual between the target vector and the first significant regressor vector, i.e., $\mathbf{y}^{(1)} = \mathbf{y}^{(0)} - \omega_{i_1}^{(1)^*}\Phi_{i_1}$. Thus we define the second significant regressor vector as

$$\Phi_{i_2} = \min_{\Phi_i \in \mathbf{\Phi}^{(1)}} \min_{\omega_i^{(2)}} |\mathbf{y}^{(1)} - \omega_i^{(2)}\Phi_i|$$

Generally, suppose that, at the time m, we have a set of significant regressor vectors $\{\Phi_{i_1}, \Phi_{i_2}, ..., \Phi_{i_{m-1}}\}$. Denote by $\mathbf{y}^{(m-1)}$ the residual vector incurred by the L1 approximation by $\{\Phi_{i_1}, \Phi_{i_2}, ..., \Phi_{i_{m-1}}\}$ and $\mathbf{\Phi}^{(m)}$ the remnant regressor vectors from $\mathbf{\Phi}$ by removing $\{\Phi_{i_1}, \Phi_{i_2}, ..., \Phi_{i_{m-1}}\}$. Then the mth significant regressor vector is defined by

$$\Phi_{i_m} = \min_{\Phi_i \in \mathbf{\Phi}^{(m)}} \min_{\omega_i^{(m)}} |\mathbf{y}^{(m-1)} - \omega_i^{(m)}\Phi_i| \tag{7}$$

where we need to solve $N-m+1$ one-parameter L1 problems.

It is very easy to solve each one-parameter L1 problem defined in (7). To see this, let $\mathbf{y} = (y(1), y(2), ..., y(N))^T$ and $\mathbf{u} = (u(1), u(2), ..., u(N))^T$ be two N-dimension vectors, and ω be an undetermined weight. The one-parameter L1 problem is defined by

$$\min_{\omega} \sum_{k=1}^{N} |y(k) - \omega u(k)| \tag{8}$$

The algorithm for the problem is very simple.

1) Throw away $u(k) = 0$, those components don't matter.
2) Compute $z(k) := y(k)/u(k)$, and $c(k) = |u(k)|$.
3) Order the $z(k)$ in increasing order, if several $z(k)$ coincide, coalesce them but redefine the corresponding $c(k)$ as the sum of all $c(j)$ with $z(j) = z(k)$.
   Thus, after step 3, we can assume that
$$z(1) < z(2) < ... < z(n), n <= N,$$
   and $c(k) > 0$ where the same notation is used for the situation after re-ordering only for simplicity.
4) Compute $s(k) = \sum_{j=k}^{m} c(j)$, and define the index k_0 where the sequence $t(k) = 2s(k) - s(1)$ changes sign, i.e.,
$$t(k_0 - 1) \geq 0 \geq t(k_0)$$
5) set $\omega = z(k_0)$.

The value $\omega = z(k_0)$ given by the algorithm is the solution to the one-parameter L1 problem defined by (8). The $N-m+1$ one-parameter L1 problems in (7) can be solved parallelly. One may note that finding the solution to a one-parameter L1 problem only needs $2N$ multiplications plus a sorting, where N is the length of vector, thus the total number of multiplications for finding the mth significant vector is $O(N*(N-m+1))$. Overall for selecting the first m_0 regressor vectors the complexity is roughly $O(N^2 m_0)$.

In this paper, the above procedure for selecting L1 significant regressor vectors is called the *L1 Significant Vector* (SV) algorithm. The input vectors $\{\mathbf{x}(i_1), ..., \mathbf{x}(i_m)\}$ corresponding to $\{\Phi_{i_1}, \Phi_{i_2}, ..., \Phi_{i_m}\}$ are called the *L1 Significant Vectors*.

One of interesting questions is that when the above procedure should be terminated, i.e., how many significant vectors should be selected such that the resulting model can be generalized better. One simple criterium is that the procedure would be terminated if the residual vector satisfies a given threshold or a given number of the significant vectors has been achieved. In fact we know the relative error $\frac{|e^{(m)}|}{\sqrt{\mathbf{y}^{(m-1)^T}\mathbf{y}^{(m-1)}}}$ at the mth step. Similar to the stopping criteria used in our first work [9], we terminate the procedure if for one of the remnant regressor vectors Φ_i,
$$\frac{|e^{(m)}|}{\sqrt{\mathbf{y}^{(m-1)^T}\mathbf{y}^{(m-1)}}} < \xi_0$$
where $0 < \xi_0$ is a pre-specified credit value.

Obviously the resulting model may cause an overfitting problem. To avoid overfitting one may use more sophisticated stopping criteria, for example the one introduced in [12] which can be used in our procedure, or one may use a regularized technique described as follows.

For the sake of simplicity, we suppose the procedure is terminated at the time m and the selected L1 significant vectors are $\mathbf{\Phi}^{(m)} = \{\Phi_1, \Phi_2, ..., \Phi_m\}$. Then we solve an overall LASSO L1 problem defined as
$$\min_{\mathbf{w}^{(m)}} |\mathbf{y} - \mathbf{\Phi}^{(m)}\mathbf{w}^{(m)}| + \lambda|\mathbf{w}^{(m)}| \tag{9}$$
where $\mathbf{w}^{(m)} = (\omega_1, \omega_2, ..., \omega_m)^T$ is the weight vector corresponding to the L1 significant vectors and λ is a given regularization factor. The second term in (9) is called the lasso penalty [11]. The shrinkage resulting from the L1 lasso penalty is better suited to sparse situations, where some overfitted L1 significant vectors could be shrunk out. We can convert problem (9) into an optimization problem by using the similar technique for support vector regression algorithm, see [15].

Further we can apply the significant vector algorithm to the support vector machine regression involving with the so-called ϵ-insensitive error function, see [15],
$$E = |\mathbf{e}|_\epsilon = \sum_{k=1}^{N} |e_k|_\epsilon$$
where
$$|e_k|_\epsilon = \begin{cases} 0 & \text{if } |e_k| \leq \epsilon \\ |e_k| - \epsilon & \text{if } |e_k| > \epsilon \end{cases}$$

III. Modelling Examples

We are now in a position to compare our algorithm with both Chen's LROLS algorithm [3] and L2 SV algorithm [9]. Our modelling simulation is conducted on the three examples used in [3] for the purpose of comparison.

Example 1: In this example we use a Gaussian radial basis function (RBF) network to model the scalar function
$$f(x) = \sin(2\pi x), \qquad 0 \leq x \leq 1.$$
The width of RBF kernel function is 0.2, i.e., $\sigma^2 = 0.04$. A set of training data $\mathcal{D} = \{(x_k, t_k)\}_{k=1}^{100}$ is generated for the input x_k by drawing from the uniformly distribution over $[0, \quad 1]$ and the target noise within t_k was given by Laplacian with zero mean and the deviation 0.2. The target is quite noisy compared to the maximal target values ± 1 and there are some outliers. The full RBF model is defined by all the RBF regressors with centers at each input training data, thus $N = 100$. As we have pointed out in section II without regularization the constructed models suffered from a serious over-fitting problem. The regularized significant vector algorithm derived in the last section was able to overcome this problem and produced a sparser seven-term model, with the means square error (MSE) values over the noisy training set and the noise-free testing set being 0.22143 and 0.000823, respectively.

Table 1 compares the MSE values over the training and testing sets for the models constructed by the LROLS [3], the regularized significant vector algorithm (RSV) and L1 RSV. Obviously the result of L1 RSV algorithm is better than that of other algorithms due to the existence of Laplacian noise. The result given by The model map of the 7-term model produced by the RSV algorithm is shown in Figure 1 (b) where the significant vectors (or selected regressors) are marked as $+$.

Example 2: This is a two-dimensional simulated nonlinear time series given by
$$\begin{aligned} y(k) =& (0.8 - 0.5\exp\{-y^2(k-1)\})y(k-1) \\ & - (0.3 + 0.9\exp\{-y^2(k-1)\})y(k-2) \\ & + 0.1\sin(\pi y(k-1)) + e(k) \end{aligned} \tag{10}$$

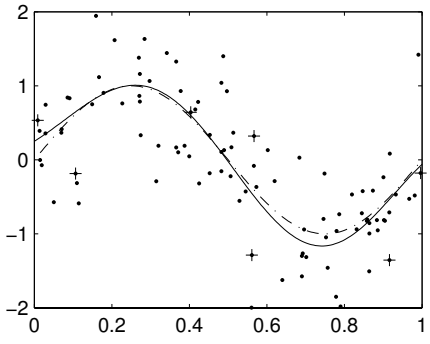


Fig. 1. The results for the simple scalar function modelling problem: dots are the noise training data, the dash-curve is the underlying function $f(x)$, the solid curves are models generated from the proposed L1 RSV algorithm and the marker $+$ indicates the L1 significant vectors selected by L1 RSV algorithm

TABLE I

MEAN SQUARE ERRORS (INTEGER IS THE NO OF REGRESSORS)

Methods	LROLS (11)	RSV (13)	L1 SV (7)
Training MSE	0.21581	0.11559	0.22143
Test MSE	0.007204	0.01634	0.000823

where the noise $e(k)$ is Laplacian with zero mean and variance 0.08. We generated one thousand noisy samples with the initial conditions $y(0) = y(1) = 0.0$. The first 500 data points were used for training, and the other 500 samples were used for possible cross-validation. The underlying noise-free system was specified by a limit circle, as shown by the one thousand samples given in Figure 2 (b) with initial value $y(0) = y(-1) = 0.1$. We use a Gaussian RBF model in the form

$$\widehat{y}(k) = \widehat{f}_{\text{RBF}}(\mathbf{x}(k)) \quad \text{with} \quad \mathbf{x}(k) = [y(k-1), y(k-2)]^T$$

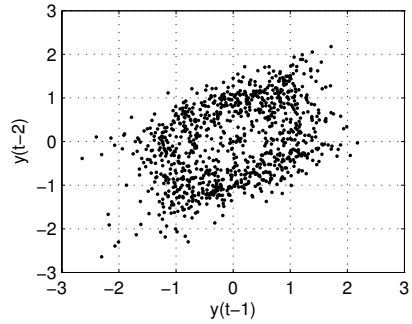
The modelled results with 18 significant vectors are shown in Figure 2. Figure 2(c) plots the result generated by one-step prediction from the learnt model and Figure 2(d) shows the model output generated by iterative model prediction. The training MSE is, respectively, 0.16438 for the L1 RSV algorithm and 0.20752 for the LROLS algorithm while the test MSE is, respectively, 0.08978 by the L1 RSV and 0.09356 by the LROLS, see Table 2

Example 3: The third example is a practical modelling problem. In this example, we are about to construct a model representing the relationship between the fuel rack position (input) and the engine speed (output) for a Leyland TL11 turbocharged, direct inject diesel engine operated at low engine speed. Detailed system description and experimental setup can

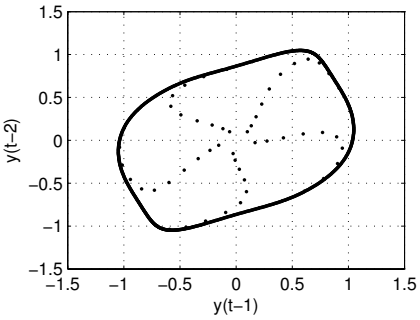
TABLE II

MEAN SQUARE ERRORS (INTEGER IS THE NO OF REGRESSORS)

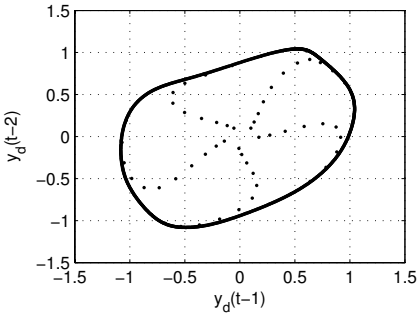
Methods	LROLS (17)	L1 RSV (18)	Random (18)
Training MSE	0.20752	0.16438	0.23435
Test MSE	0.09356	0.08978	0.12037



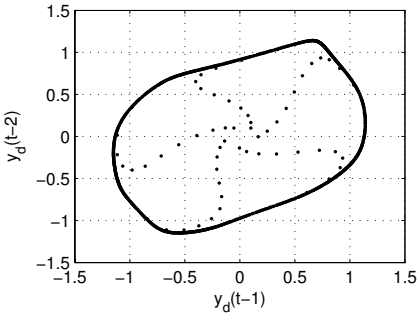
(a) Phase plot of noisy training data set ($y(0) = y(-1) = 0$)



(b) Phase plot of noise-free data generated by (10) without $e(k)$



(c) Phase plot of one-step model prediction



(d) Phase plot of iterative model prediction

Fig. 2. The results for modelling the nonlinear system defined by (10)

TABLE III

MEAN SQUARE ERRORS (INTEGER IS THE NO OF REGRESSORS)

Methods	LROLS (34)	L1 RSV (38)	Random (38)
Training MSE	0.000439	0.000576	0.003673
Test MSE	0.000485	0.000512	0.002538

be found in [1]. The data set consists of 410 samples. We use the first 210 data points as training data in modelling and the last 200 points in model validation. An RBF model of the form

$$\widehat{y}(k) = \widehat{f}_{\text{RBF}}(\mathbf{x}(k)) \tag{11}$$

but this time the input vector $\mathbf{x}(k)$ is defined as

$$\mathbf{x}(k) = [y(k-1), u(k-1), u(k-2)]^T \tag{12}$$

where u means the fuel input. The variance of the RBF kernel function was chosen to be 1.69. The total number of regressors is $N = 210$ in the initial stage. By running L1 RSV algorithm a model with 38-term significant regressors was constructed with MSE values over the training and testing data were 0.000576 and 0.000512 respectively, see Table 3. The result is still comparable to the one given by LROLS [3].

The constructed RBF model by the L1 RSV algorithm was used to generate the one-step prediction $\widehat{y}(k)$ of the system output according to (12). The iterative model output $\widehat{y}_d(k)$ was also produced by (11) with

$$\mathbf{x}_d(k) = [\widehat{y}_d(k-1), u(k-1), u(k-2)]^T \tag{13}$$

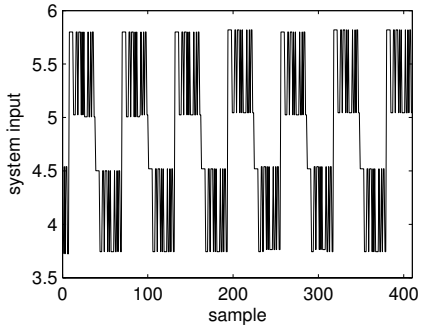
The one-step model prediction and iterative model output for this 38-term model selected by L1 RSV algorithm are shown in Figure 3 in comparison with the system output.

IV. CONCLUSIONS

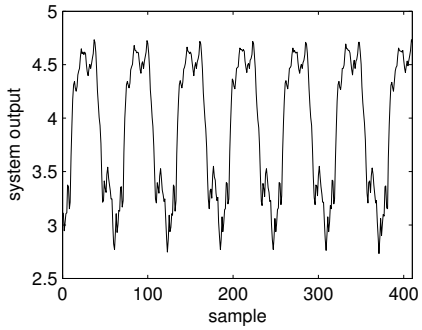
The L1 regularized significant vector algorithm has been proposed for nonlinear system identification using the kernel regression model. Compared to the LROLS algorithm the new algorithm has less computational complexity by removing the orthogonalization procedure employed in LROLS while the overall performance offered by the L1 RSV algorithm is considerably comparable to the results given by the LROLS algorithm, which has been demonstrated by three modelling problems. The L1 RSV can handle non-Gaussian noise as shown by Example 1 and 2. The computational requirements of this iterative model algorithm are very simple and its implementation is straightforward. The core idea can be easily extended to other cases such as robust loss measures and error/loss functions for classification problems.

ACKNOWLEDGEMENTS

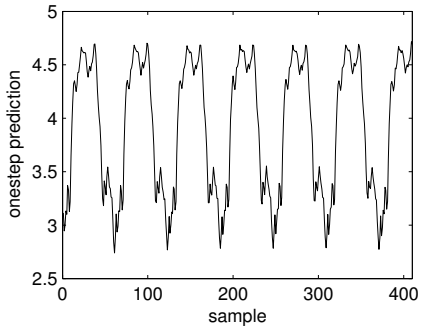
The authors thank for the supports from the grant (No. 60373090) from the National Natural Science Foundation of China (NSFC), the ARC DP Development Grant from Charles Sturt University and the University Research Grant from the University of New England.



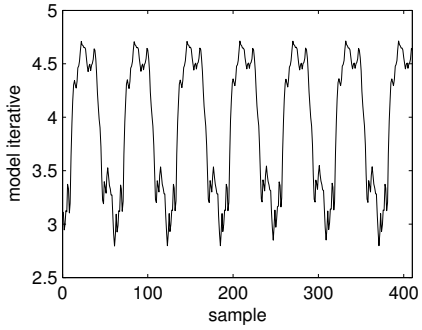
(a) Engine input data $u(k)$



(b) Engine output data $y(k)$



(c) Output plot of one-step model prediction



(d) Output plot of iterative model prediction

Fig. 3. The results for modelling the relationship between the engine speed and the fuel rack position

REFERENCES

[1] S.A. Billings, S. Chen, and R.J. Backhouse. The identification of linear and nonlinear models of a turbocharged automotive diesel engine. *Mech. Syst. Signal Processing*, 3(2):123–142, 1989.

[2] S. Chen. Locally regularized orthogonal least squares for the construction of sparse kernel regression models. In *Proceeding of 6th Int. Conf. Signal Processing*, volume 2, pages 1229–1232, Beijing, China, 2002.

[3] S. Chen. Local regularization assisted orthogonal least squares regression. *International Journal of Control*, 2004.

[4] S. Chen and S.A. Billings. Representations of nonlinear systems: the NARMAX model. *International Journal of Control*, 49(3):1013–1032, 1989.

[5] S. Chen, S.A. Billings, and W. Luo. Orthogonal least squares methods and their application to non-linear system identification. *International Journal of Control*, 50(5):1873–1896, 1989.

[6] S. Chen, C.F. Cowan, and P.M. Grant. Orthogonal least squares learning algorithm for radial basis function networks. *IEEE Trans. Neural Networks*, 2:302–309, 1991.

[7] 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*, 48(6):1029–1036, 2003.

[8] P.M.L. Drezet and R.F. Harrison. Support vector machines for system identification. In *Proceeding of UKACC Int. Conf. Control'98*, pages 688–692, Swansea, U.K., 1998.

[9] J.B. Gao, D.M. Shi, and X.M. Liu. Critical vector learning to construct sparse kernel regression modelling. *Neural Networks*, submitted.

[10] T.V. Gestel, M. Espinoza, J.A.K. Suykens, C. Brasseur, and B. deMoor. Bayesian input selection for nonlinear regression with LS-SVMS. In *Proceedings of 13th IFAC Symposium on System Identification*, pages 27–29, Totterdam, The Netherlands, 2003.

[11] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer Series in Statistics. Springer, Berlin, 2001.

[12] B.J. Kruif and T.J.A. Vries. Support-Vector-based least squares for learning non-linear dynamics. In *Proceedings of 41st IEEE Conference on Decision and Control*, pages 10–13, Las Vegas, USA, 2002.

[13] T. Poggio and F. Girosi. A sparse representation for function approximation. *Neural Computation*, 10:1445–1454, 1998.

[14] B. Schölkopf and A.J. Smola. *Learning with Kernels*. The MIT Press, Cambridge, Massachusetts, 2002.

[15] J.A.K. Suykens, T. van Gestel, J. DeBrabanter, and B. DeMoor. *Least Square Support Vector Machines*. World Scientific, Singapore, 2002.

[16] M.E. Tipping. Sparse Bayesian learning and the relevance vector machine. *J. Machine Learnign Research*, 1:211–244, 2001.

[17] J. Valyon and G. Horváth. A generalized LS-SVM. In J. Principe, L. Gile, N. Morgan, and E. Wilson, editors, *Proceedings of 13th IFAC Symposium on System Identification*, Rotterdam, The Netherlands, 2003.

[18] V.N. Vapnik. *Statistical Learning Theory*. Wiley, New York, 1998.