Proceeding of the 2004 American Control Conference
Boston, Massachusetts June 30 - July 2, 2004

A Novel Method of Process Dead-time Identification: Support Vector Machine Approach

Zhu Hongdong, Shao Huihe

Abstract—**Performance and robustness of model-based control system are sensitive to the modeling error, especially to the dead-time identification error. Support vector machine (SVM) employs structure risk minimization principle to control model complexity and the upper bound of generalization risk. If the seeking dead-time contained in training data equals dead time of actual plant, the trained SVM will have the lowest complexity. The identification procedure is described as follows. Firstly, specify a dead-time seeking range based on the prior process knowledge. Secondly, construct training data sets from input-output data according to different dead times in seeking range and train SVMs respectively. Finally, the estimated dead-time can be obtained through comparing the numbers of support vectors of all trained SVMs. A lot of discrete simulations for the first order plus dead-time system have been done to illuminate the effectiveness of proposed method.**

I. INTRODUCTION

Dead-time in many industrial processes is a well-recognized phenomenon. It may be caused by transportation of materials, sampling time, requirements for human intervention. As the dead-time of an open loop transfer function increases, the crossover frequency decreases. If a process has a relatively large dead-time compared to the dominant time constant, the achievable performance of conventional feedback control (e.g. PID control) systems can be significantly degraded. In order to enhance the performance of closed loop system, dead time compensation may be necessary. The popular schemes for such compensation are the Smith Predictor and Internal Model Control scheme [1]. They are all model based control schemes. Unfortunately, their performance and stability are all sensitive to the modeling error, especially to the dead-time error. Large dead time modeling error can cause the closed loop system unstable. Therefore, it is significant for model based control system to precisely identify the process dead-time.

There are various dead-time identification methods. They may be broadly classified into time domain and frequency domain techniques. There are three main time domain methods: multiple model estimation method, gradient method, and experimental open loop method. When using frequency domain techniques, one must estimate the process frequency response firstly and the use graphical or analytical method to obtain the model parameters. [2]

In this paper, we propose a novel method for process dead-time identification: support vector machine approach. It belongs to time domain method. Applying the property of SVM, the process dead-time can be identified precisely by proposed method. The method is composed of three steps. Firstly, specify dead-time seeking range and construct training data sets from sampling data according to the different dead-time in seeking range. Secondly, train SVM using the constructed training data sets and obtain a couple of trained SVMs. Finally, compare the support vector numbers of all trained SVMs and obtain the dead time of actual plant.

This paper is organized as follows: in section 2 basic principles of SVM are introduced; the identification approach is proposed in section 3; a lot of simulations results are proposed in section 4; we draw some conclusions in last section.

II. SUPPORT VECTOR MACHINE (SVM)

SVM was first suggested by Vapnik in 1960s for classification and the whole SVM framework was entirely described in 1995. SVM, based on statistical learning theory, is a training algorithm for classification, regression, and density approximation. It holds the Structural Risk Minimization (SRM) principle, which has been shown to be superior to traditional Empirical Risk Minimization (ERM) principle, employed by conventional neural networks. SRM minimizes an upper bound on the VC dimension, as opposed to ERM that minimizes the error on the training data [4]. This principle incorporates capacity control to prevent over fitting. The parameters of SVM are obtained from solving optimal problem with a convex objective function. Therefore, the local-minimum problem is not existed in

Manuscript received September 20, 2003.

Zhu Hongdong is with Institute of Automation, Shanghai Jiao Tong University, Shanghai, 200030, PRC (phone: 86-21-6933427-13; fax: 86-21-62932138; e-mail: hongdongzhu@ sjtu.edu.cn).

Shao Huihe is with Institute of Automation, Shanghai Jiao Tong University, Shanghai, 200030, PRC (e-mail: hhshao@ sjtu.edu.cn).

0-7803-8335-4/04/$17.00 ©2004 AACC

SVM training procedure. In addition, the generalization ability can be guaranteed when employing small sizes sample data to train the SVM.

It is well known for a given sample size that there exists a model of optimal complexity corresponding to the smallest generalization error. Then, any approach for learning from finite samples needs to have some methods for complexity control. VC-theory provides a very general and powerful framework for complexity control. For regression problems with squared loss function the following bound on prediction risk holds with probability 1-η[4]:

$$R(\eta) \leq \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 / (1 - c\sqrt{\frac{h(\ln(\frac{an}{h})+1)-\ln\eta}{n}}) \quad (1)$$

where $R(\eta)$ is the generalization error, h is the VC-dimension of the set of approximating functions, c and a are constants, and n is the sample size. In practical use, a is often selected as 1. From (1), we can draw a conclusion that: for a given sample set the smaller the h is, the smaller the upper bound of generalization risk is. Hence, the VC-dimension (complexity) of approximation model reflects the generalization errors.

VC-dimension is defined as: there exists a set of points x_n such that these points can be separated in all 2^n possible configurations, and that no set x_m exists where $m>n$ satisfying this property. For a set of linear functions the VC-dimension equals the number of free parameters [4].

When SVM is applied to solve the regression problem, it is called Support Vector Regression (SVR). SVR is formulated as follows. Given a set of input-output data $\{(\vec{x}_i, y_i), i = 1,...,n\}$, where $\vec{x}_i \in R^N$ is the N-dimension input vector and y_i is the corresponding system output. For linear regression, assume that the estimated regression in the set of linear functions is:

$$f(\vec{x}) = (\vec{w} \cdot \vec{x}) + b \quad (2)$$

where $\vec{w}$ and b are the weight and bias, respectively. The weight $\vec{w}$ has the following form: $\sum_{i=1}^{l}(\alpha_i - \alpha_i^*)\vec{x}_i$, where α_i and α_i^* are the Lagrange multipliers and l is the number of support vectors. For nonlinear regression, assume that the estimated regression in the set of nonlinear functions is: $f(\vec{x}) = \vec{w} \cdot \varphi(\vec{x}) + b$, where $\varphi(\vec{x})$ is a kind of kernel function that satisfies the Mercer's condition. The weight $\vec{w}$ has the following form: $\sum_{i=1}^{l}(\alpha_i - \alpha_i^*)\varphi(\vec{x}_i)$. Minimize a loss function to obtain the Lagrange multipliers and the bias term. The loss function can be defined as quadratic loss function, Huber loss function, and ϵ-insensitive loss function. In practice, ϵ-insensitive loss function is often employed. The parameters can be calculated by the following procedure. Transform the original problem to the dual problem: Minimize

$$L(\vec{w}, b) = \frac{1}{2}\|\vec{w}\|^2 + C\sum_{i=1}^{l}(\xi_i + \xi_i^*), \quad (3)$$

under the constrains:

$$\begin{aligned} y_i - f(\vec{x}_i) &\leq \varepsilon + \xi^*, & i &= 1,...,l \\ f(\vec{x}_i) - y_i &\leq \varepsilon + \xi, & i &= 1,...,l \\ \xi^* &\geq 0, & i &= 1,...,l \\ \xi &\geq 0, & i &= 1,...,l \end{aligned} \quad (4)$$

where C is the regulation control parameter and pre-specified by user, ξ^* and ξ are slack variables, and ε is the noisy control parameter. If the data is "polluted" by large magnitude noise, ε should be selected to be a large one. If one chose ε to zero, all the training vectors will be support vectors and the generalization ability of SVR will be very poor. In order to guarantee the sparseness, ε should not equal zero.

III. Process Dead-time Identification Through SVM

The key purpose in system identification is to find a suitable model structure with suitable parameters to represent the real process dynamic. It is important to select regression vector during the regression estimation.

In this paper, we study the first order plus dead time discrete system:

$$y(k+1) = a \cdot y(k) + b \cdot u(k - d) \quad (5)$$

where $y(k+1)$, $y(k)$ are respectively next step predictive values of plant output and current plant output, $u(k\text{-}d)$ is the d step before control input. a and b are the plant parameters. d is the delay step. If the sampling period is T_s, the dead-time of system will be $d \cdot T_s$. In discrete system (5), dead-time identification is equal to identifying delay step d.

The main goal in this paper is to applying SVM to find the delay step d. Identification principle will be introduced as follows. When we approximate this dynamic system using SVM, the first problem is how to construct the training data, $\{(\vec{x}_i, y_i), i = 1,...,n\}$. From (5), the optimal selection of regression vector $\vec{x}_i$ should be $[y(k)\ u(k\text{-}d)]^T$. Then, the training data set is composed of regression vectors and corresponding plant outputs, $\{[\ y(k)\ u(k\text{-}d)]^T,\ y(k+1)\ \}$, $k=d+1, ..., d+n$. Unfortunately, the dead time of plant is unknown. On the other hand, the goal of training SVM is to obtain the cause-effect relationships hiding behind the training data. In this problem, the "cause" is the regression vector, $[y(k)\ u(k\text{-}d)]^T$, and the "effect" is the corresponding plant output $y(k+1)$. If we choose a delay step d' which equals the real plant dead time (correct cause) to construct training data set and train SVM, the trained SVM will have minimal complexity. Remember that the VC-dimension for linear function set equals the number of the free parameters. Therefore, for linear SVR the VC-dimension equals $l+1$ if number of support vectors of trained SVR equals l. Then, the SVM, trained by appropriate constructed training data, has minimal number of support vectors. Otherwise, the SVM will increase more complexity (the number of support vectors) to fit for the wrong cause-effect relationship hiding in the inappropriate constructed training data. According to the above explanation, we summary the dead time identification procedure as follows. Firstly, specify a dead time seeking range [d'_{min}, d'_{max}] around the real plant delay step and construct different cause-effect training data sets according to the different d' in the seeking range. d'_{min} and d'_{max}, which are specified by user based on the prior plant knowledge, are the lower bound and upper bound of delay step, respectively. The constructed training sets have following form: $\{(\vec{x}_i^k, y^k{}_i), i=1,...,n; k=1,...,m\}$, where n and m are the numbers of training data and training sets, respectively. Secondly, train the SVM using the m different training sets. Finally, compare the numbers of support vectors of all the trained SVMs and find the optimal SVM with minimal complexity, i.e. minimal number of support vectors. Then the delay step d' corresponding to the optimal SVM is the real delay step.

IV. Simulation Results

In this section, we present eight simulations for a first order plus dead time plant with eight different delay steps in order to verify the proposed idea. The plant is

$$y(k+1) = 0.9512 \cdot y(k) + 0.07316 \cdot u(k - d) \quad (6)$$

The sampling period is 0.1 second. The nine different delay steps and corresponding delay step seeking ranges are displayed on Table I.

1) The simulation procedure is described as follows.
2) Choose a delay step d from Table I. Substitute d to (6).
3) Employ signal $1+sin(10t+5)+sin(2t+1)$ as plant input. Stimulate the plant and record the input and output data.
4) Construct d'_{max}-d'_{min} training sets according to the different d' within seeking range. Each training set has 50 elements.
5) Train SVM with the d'_{max}-d'_{min} training sets.
6) Compare the numbers of support number of trained SVMs and obtain the real delay step.
7) Change the plant delay step and repeat the step 2 to step 6 till all nine plants are simulated.

TABLE I
Delay Step of Plant and their corresponding seeking ranges

d	0	4	10	20	30	44	60	80
d'_{min}	0	0	5	15	25	39	65	75
d'_{max}	10	10	15	25	35	49	65	85

Note that in SVM framework there are two open parameters, ε and regulation parameter C. It means that C and ε are selected by users according to the different application. In this paper we chose ε=0.01 according to our experience. In order to study the influence of C, ten simulations have been done for each plant delay step under the conditions that C is selected as 0.5, 4, 9, 15, 25, 60, 100, 500, 1000, *infinite*.

The simulations results are shown in appendix table II to VIII. Each table contains the results for a plant delay step with different C. The elements in the first row of each table are the seeking delay steps, while the first column elements of each table are the different C. The SV numbers of SVM trained by different training sets and different C are shown in each table. The least numbers of support vectors are marked by bold font. In order to display the results intuitively, Fig.1 shows the curves of SV numbers in the cases that the real plant delay step is 20 and C is set to 25 and 100.

From appendix tables and figure 1, we can get that:

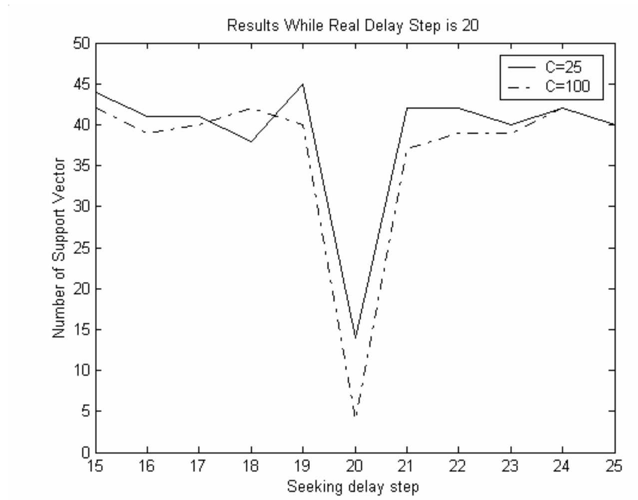

Fig. 1. Results while actual delay step is 20 (solid line: C=25; dashdoted line: C=100)

1) The results are stable in respect of C while C is larger than 9. When C is smaller than 9, we cannot use this method to identify the delay step (dead time).
2) When the seeking delay step equals real plant dead time, the SV number is decreased dramatically. The proposed method is sensitive to identifying the delay step (dead-time) through detecting the model complexity.

V. Conclusions

Under SVM frameworks, the SV number reflects the model complexity for linear regression. It is important for approximating plant model to select appropriate regression vectors. Inappropriate selection of regression vector will decrease generalization ability of trained model. When the delay step of training data does not equal the real plant delay step, the SVM has to increase complexity to fit the wrong cause and effect relationship.

The selection of *C* is robust for proposed method. When *C* varies in a wide range, the proposed method can identify the dead time of real plant precisely.

The proposed method is sensitive to identifying the dead time of real plant.

The novel delay step identification method has been verified by lots of simulations.

APPENDIX

TABLE II
SIMULATION RESULTS WHERE DELAY STEP OF ACTUAL PROCESS IS 0

C \ *d'*	**0**	1	2	3	4	5	6	7	8	9	10
0.5	48	49	49	48	49	46	47	49	48	47	49
4	48	47	47	48	46	47	48	48	48	47	46
9	**34**	44	47	46	46	45	43	45	46	50	48
15	**20**	39	43	43	45	41	46	47	49	49	48
25	**14**	42	42	40	42	40	47	47	44	47	47
60	**6**	39	39	41	40	39	48	42	43	45	42
100	**4**	37	39	39	42	40	46	42	46	48	42
500	**4**	35	40	41	43	39	44	43	47	47	41
1000	**4**	35	38	41	43	38	44	43	46	46	41
Inf	**4**	41	36	36	43	44	45	45	46	43	40

TABLE III
SIMULATION RESULTS WHERE DELAY STEP OF ACTUAL PROCESS IS 4

C \ *d'*	0	1	2	3	**4**	5	6	7	8	9	10
0.5	49	50	48	50	48	49	49	48	49	46	47
4	49	45	49	47	48	47	47	48	46	47	48
9	47	45	42	45	**34**	44	47	46	46	45	43
15	43	46	40	44	**20**	39	43	43	45	41	46
25	41	41	38	45	**14**	42	42	40	42	40	47
60	39	39	41	40	**6**	39	39	41	40	39	48
100	39	40	42	40	**4**	37	39	39	42	40	46
500	37	39	43	40	**4**	35	40	41	43	39	44
1000	37	38	43	39	**4**	35	38	41	43	38	44
Inf	43	41	37	39	**4**	41	36	36	43	44	45

TABLE IV
SIMULATION RESULTS WHERE DELAY STEP OF ACTUAL PROCESS IS 10

C \ *d'*	5	6	7	8	9	**10**	11	12	13	14	15
0.5	50	49	50	48	50	48	49	49	48	49	46
4	47	49	45	49	47	48	47	47	48	46	47
9	47	47	45	42	45	**34**	44	47	46	46	45
15	45	43	46	40	44	**20**	39	43	43	45	41
25	44	41	41	38	45	**14**	42	42	40	42	40
60	44	39	39	41	40	**6**	39	39	41	40	39
100	42	39	40	42	40	**4**	37	39	39	42	40
500	44	37	39	43	40	**4**	35	40	41	43	39
1000	45	37	38	43	39	**4**	35	38	41	43	38
Inf	40	43	41	37	39	**4**	41	36	36	43	44

TABLE V
SIMULATION RESULTS WHERE DELAY STEP OF ACTUAL PROCESS IS 20

C \ *d'*	15	16	17	18	19	**20**	21	22	23	24	25
0.5	50	49	50	48	50	48	49	49	48	49	46
4	47	49	45	49	47	48	47	47	48	46	47
9	47	47	45	42	45	**34**	44	47	46	46	45
15	45	43	46	40	44	**20**	39	43	43	45	41
25	44	41	41	38	45	**14**	42	42	40	42	40
60	44	39	39	41	40	**6**	39	39	41	40	39
100	42	39	40	42	40	**4**	37	39	39	42	40
500	44	37	39	43	40	**4**	35	40	41	43	39
1000	45	37	38	43	39	**4**	35	38	41	43	38
Inf	40	43	41	37	39	**4**	41	36	36	43	44

TABLE VI
SIMULATION RESULTS WHERE DELAY STEP OF ACTUAL PROCESS IS 30

C \ *d'*	25	26	27	28	29	**30**	31	32	33	34	35
0.5	50	49	50	48	50	48	49	49	48	49	46
4	47	49	45	49	47	48	47	47	48	46	47
9	47	47	45	42	45	**34**	44	47	46	46	45
15	45	43	46	40	44	**20**	39	43	43	45	41
25	44	41	41	38	45	**14**	42	42	40	42	40
60	44	39	39	41	40	**6**	39	39	41	40	39
100	42	39	40	42	40	**4**	37	39	39	42	40
500	44	37	39	43	40	**4**	35	40	41	43	39
1000	45	37	38	43	39	**4**	35	38	41	43	38
Inf	40	43	41	37	39	**4**	41	36	36	43	44

TABLE VII
SIMULATION RESULTS WHERE DELAY STEP OF ACTUAL PROCESS IS 44

C \ *d'*	39	40	41	42	43	**44**	45	46	47	48	49
0.5	50	49	50	48	50	48	49	49	48	49	46
4	47	49	45	49	47	48	47	47	48	46	47
9	47	47	45	42	45	**34**	44	47	46	46	45
15	45	43	46	40	44	**20**	39	43	43	45	41
25	44	41	41	38	45	**14**	42	42	40	42	40
60	44	39	39	41	40	**6**	39	39	41	40	39
100	42	39	40	42	40	**4**	37	39	39	42	40
500	44	37	39	43	40	**4**	35	40	41	43	39
1000	45	37	38	43	39	**4**	35	38	41	43	38
Inf	40	43	41	37	39	**4**	41	36	36	43	44

TABLE VIII
SIMULATION RESULTS WHERE DELAY STEP OF ACTUAL PROCESS IS 60

C \ *d'*	55	56	57	58	59	**60**	61	62	63	64	65
0.5	50	49	50	48	50	48	49	49	48	49	46
4	47	49	45	49	47	48	47	47	48	46	47
9	47	47	45	42	45	**34**	44	47	46	46	45
15	45	43	46	40	44	**20**	39	43	43	45	41
25	44	41	41	38	45	**14**	42	42	40	42	40
60	44	39	39	41	40	**6**	39	39	41	40	39
100	42	39	40	42	40	**4**	37	39	39	42	40
500	44	37	39	43	40	**4**	35	40	41	43	39
1000	45	37	38	43	39	**4**	35	38	41	43	38
Inf	40	43	41	37	39	**4**	41	36	36	43	44

TABLE IX
SIMULATION RESULTS WHERE DELAY STEP OF ACTUAL PROCESS IS 80

C \ *d'*	75	76	77	78	79	**80**	81	82	83	84	85
0.5	50	49	50	48	50	48	49	49	48	49	46
4	47	49	45	49	47	48	47	47	48	46	47
9	47	47	45	42	45	**34**	44	47	46	46	45
15	45	43	46	40	44	**20**	39	43	43	45	41
25	44	41	41	38	45	**14**	42	42	40	42	40
60	44	39	39	41	40	**6**	39	39	41	40	39
100	42	39	40	42	40	**4**	37	39	39	42	40
500	44	37	39	43	40	**4**	35	40	41	43	39
1000	45	37	38	43	39	**4**	35	38	41	43	38
Inf	40	43	41	37	39	**4**	41	36	36	43	44

REFERENCES

[1] K.K. Tan, T.H. Lee, F.M Leu, "Predictive PI versus Smith control for dead-time compensation," ISA Trans.,vol.40, pp. 17-29,2001

[2] O'Dwyer, A, "Time delayed process model parameter estimation: a classification of techniques," in *Proc. of UKACC International Conference on Control* 2000, Cambridge, UK, 2000, Available: http://www.dit.ie/DIT/engineering/csee/staff/aodwyer/research/refpubs/index.html

[3] Vladimir Cherkassky, Xuhui Shao,Filip M. Mulier,et.al., "Model Complexity Control for Regression Using VC Generalization Bounds," *IEEE Trans.* Neural Networks, vol. 10, pp. 1075-1089,1999

[4] Vladimir N. Vapnik, The Nature of Statistical Learning Theory, New York: Springer-Verlag,1999