Proceedings of the 19th World Congress
The International Federation of Automatic Control
Cape Town, South Africa. August 24-29, 2014

Control Oriented System Identification for Performance Management in Virtualized Software System

Dharma Aryani * Liuping Wang ** Tharindu Patikirikorala ***

* *School of Electrical Engineering and Computer Science, RMIT University, VIC, Australia; e-mail: dharma.aryani@rmit.edu.au*
** *School of Electrical Engineering and Computer Science, RMIT University, VIC, Australia*
*** *Faculty of Information and Communication Technologies, Swinburne University of Technology, VIC, Australia*

Abstract: Capability to manage the performance of a shared resources environment relies on the model estimation of all dynamics in the system. The main challenge is to capture the nonlinear characteristic which inherently exists in software system applications. Hammerstein-Wiener block structural model is widely regarded as a basis for description of nonlinear systems. This paper extends the existing work in system identification of software systems using Hammerstein-Wiener block structural model. As part of the estimation, the linear element is represented in terms of Frequency Sampling Filter model and the inverse static output non-linearity is estimated using B-Spline curve functions. Experimental studies show that the static nonlinearities estimated when using the B-Spline functions have less oscillations and the overall model has a better performance than the case when using polynomial functions.

Keywords: Nonlinear system identification; Software system modeling; Software system control; Hammerstein-Wiener model; Performance management.

1. INTRODUCTION

The objective to deploy control mechanisms for performance and resources allocation management between many users in shared resource applications leads to an increasing demand of control technologies and dynamic modeling investigations. Control engineering approaches have been studied to automate performance management tasks to regulate different domains such as web server systems, storage systems and data centers. The existing approaches have used linear modeling and control approaches by disregarding the dominant nonlinear dynamics of shared resource systems (Lu [2001],Lu [2002],Lu [2003]). This limitation affecting the performance management because the overall behaviour of a software system is only partially captured.

In order to design a robust controller, the nonlinearities as a part of system characteristics should also be included in the model estimation stage. System identification of a Hammerstein-Wiener model is a well-known method to approximate the system dynamics in a form of block-oriented model that consists of a linear model and static nonlinearities on input and output variables Fig.1. There are many control applications that have implemented Hammerstein-Wiener model in engineering, medical, science and process industries. The Hammerstein model consists of a nonlinear function followed by a linear model of system dynamics. Conversely, in Wiener model, the nonlinear function comes after the linear model. Both models have been deployed in a wide range of engineering applications involving significant nonlinearity issues in the process. For example, Jurado [2006] and Hong [2011].

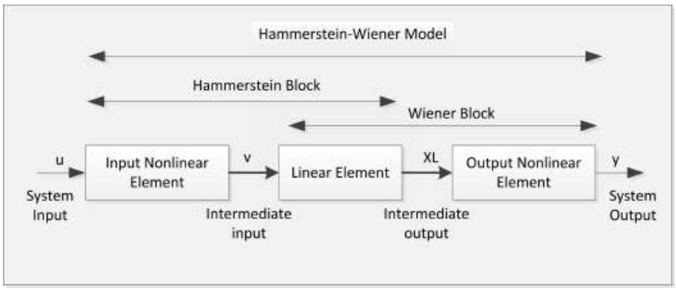


Fig. 1. Hammerstein-Wiener block structure

The identification procedures for Wiener-type model could proceed in two steps or single step. In the two-steps identification, the linear element is identified first and the second stage is required for the nonlinear model or in a reverse way like the work by Cervantes [2003]. There were different ways to represent the Wiener models. Several earliest studies initiated to use cross-correlation analysis between input and output data to identify Wiener system. Subsequent studies showed that both the linear and the static nonlinearity could be identified in their inverse form. The other research proposed the identification of direct nonlinearity by using a recursive method. Another effective way is to identify both linear and nonlinear elements in one combined estimation as in Kalafatis [1995]. Most of the existing approaches use polynomial functions to represent the nonlinear elements. The use of various spline functions in the modelling of the Wiener system have been studied such as in Kalafatis [1997]. B-splines function is constructed based on the initial finding which is widely known as "de-Boor" algorithm (DeBoor [1978]).

In a shared resources environment with unpredictable workload variation, another challenge to manage performance guarantees is to provide different levels of performance guarantees to each client classes depending on the service preferences while dynamically allocating the available resources. Relative guarantee management has been used as a solution scheme for service differentiation based on the level of importance of client classes (Lu [2003], Patikirikorala [2011]).

This paper contributes to the improvement of the previous work in feedback control for resource management using Hammerstein-Wiener model where the Wiener model was identified in two-steps estimation (see Patikirikorala [2011]). In this work, the Wiener model is estimated in a straightforward manner where the linear and the inverse static nonlinear model are identified simultaneously. The linear model is represented by non-parametric Frequency Sampling Filter (FSF) model and the inverse static nonlinear model in polynomial function and B-Spline terms.

The structure of the paper is outlined as follows. The process description of the Virtual Machine (VM) environment used in this work is described in Section 2. Section 3 covers the identification algorithms and the identification results using experimental data are given in Section 4. The conclusion is in Section 5.

2. PROCESS DESCRIPTION

Three physical machines (M_1,M_2 and M_3) are described in Figure 2 to represent the structure of target system in this study. It is used to generate input and output data pairs for system identification. The shared infrastructures are deployed in M_1 which is functioning as the server for the system. Within the server, two virtual machines are built using Xen Hypervisor and operate as two individual machines. The server and VMs run on CentOS and Apache Httpd 2.2 server deployed in order to establish customer application. In the client side, RUBiS benchmark is used. It is a three tier e-commerce website modeling the behavior of ebay.com. The database of RUBiS benchmark was deployed in another machine M_3 and the two client simulators which generate workloads for the server are deployed in a M_2. All three machines are connected in an isolated network using a network switch.

The dynamic characteristic of software system in this system is nonlinear because there is an inverse relationship occurs between response time and session allocations and its stability is affected by workloads change. Since the proposed performance management based on relative guarantee scheme, the ratio of response time as the controlled performance property will be maintained proportionally to the service preferences while allocating dynamic proportional resources between client classes. Assume that R_0, R_1 are the response time to the workloads of VM_0 and VM_1 respectively and S_0, S_1 are the CPU capacity allocated for each VM. In this situation, the total capacity (S_{total}) $= S_0 + S_1$ with a constraint in the minimum number of capacity (for example, $S_0, S_1 \geq 20$). For system identification and control analysis, the input variable of the system is the ratio of CPU allocations ($u = S_0/S_1$) and the output variable is the ratio of average response time of VMs ($y = R_1/R_0$).

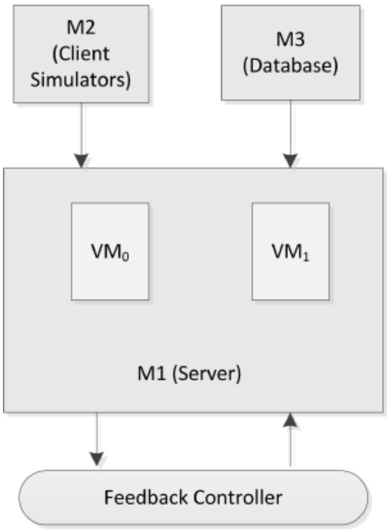


Fig. 2. Virtualized software system

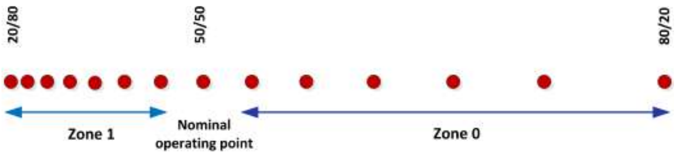


Fig. 3. Description of input nonlinearity

3. IDENTIFICATION ALGORITHMS

This section provides the steps and all equations to generate Hammerstein-Wiener model with a straightforward estimation for the Wiener model.

3.1 Hammerstein Model

For Hammerstein model, the input signal is formulated as

$$\frac{S_0}{S_1} = \frac{S_{total} - S_1}{S_1} = \frac{S_{total}}{S_1} - 1 \tag{1}$$

It is assumed that at least one request to a VM arrives from client in each sampling time. The input variable is $u = \dfrac{S_0}{S_1}$. The nonlinearities caused by static restrictions of the operating points in input variable result in the dynamics of static input nonlinearities. For instance, if $S_{total} = 100$ and $S_{min} = 20$, then the operating points will be

$$u = \frac{20}{80}, \frac{21}{79}, \ldots, \frac{50}{50}, \ldots, \frac{79}{21}, \frac{80}{20}$$

These points describe the nonlinear input behavior of the system, see Fig. 3

Transforming the restricted operating points to equally spaced operating points is a proper technique to capture static nonlinearity characteristic. This technique was proposed in Patikirikorala [2011] by defining an intermediate variable v where $v_{min} \leq v \leq v_{max}$ with $v_1 = v_{min}, v_2 = v_1 + \delta v, \cdots, v_{i+1} = v_i + \delta v, \cdots, v_p = v_{max}$.

The inverse static input nonlinear element is approximated as

$$u(i) = f^{-1}(v) = \alpha_0 + \alpha_1 v(i) + \alpha_1 v(i)^2 + \cdots + \alpha_m v(i)^m \tag{2}$$

Then the coefficient of inverse static nonlinear input is estimated by least square method,

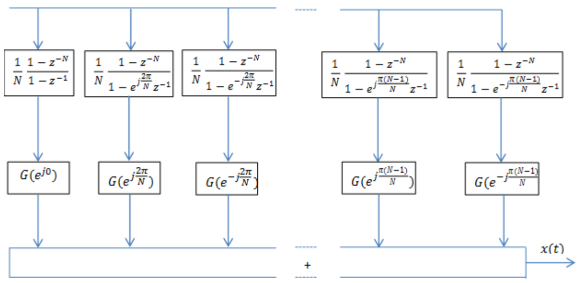


Fig. 4. Diagram of Frequency Sampling Filters

$$\hat{\Theta} = (\Phi^*\Phi)^{-1}(\Phi^* U) \tag{3}$$

where the parameter vector $\theta = [\alpha_0\ \alpha_1\ \cdots\ \alpha_m]^T$ and data vector $\phi(i) = [1\ v(i)\ \cdots\ v(i)^m]^T$

When the inverse static nonlinear input has been modeled and integrated into the system as adapter, the system is reduced to a Wiener-type model.

3.2 Wiener block

Consider the Wiener block structure shown in Fig.1 where $u(t), v(t), x(t)$ and $y(t)$ denote the process input, the unmeasurable intermediate input variable, intermediate output variable and the measured process output, respectively. Two basic assumptions take place in this identification. Firstly the dynamic linear element is stable with finite settling time T_s and secondly the static nonlinearity is continuous and single-valued in the range of input-output data. The input signal to the Wiener block is the intermediate input variable v.

The linear subsystem is described by the non-parametric frequency sampling filter (FSF) model (Bitmead [1981]). In FSF model estimation, the coefficients of regressor vector are constructed by passing the input signal of the system through the set of narrow band-limited frequency sampling filters. The FSF model of a stable, linear, time-invariant process has a z-transfer function as

$$x(t) = \sum_{k=-\frac{n-1}{2}}^{k=\frac{n-1}{2}} G(e^{-j\omega_k}) f_k(t) \tag{4}$$

where $G(e^{-j\omega_k})$ is the discrete frequency response of the linear subsystem at $\omega_k = 2\pi k/N$. $f_k(t)$ is the output of frequency sampling filters which are described by

$$f_k(t) = H_k(z)u(t) = \frac{1}{N}\frac{1-z^{-N}}{1-e^{-j\omega_k}z^{-1}}u(t) \tag{5}$$

where N is the number of samples corresponding to the process settling time $N = T_s/\delta t$ and δt is a sampling interval. The novel findings by Wang [1994] and Wang [1997] concluded that the number of parameters associated with FSF model is much smaller than an FIR model. Referring to the second assumption that the inverse of static nonlinear element is a single-valued smooth function, polynomial where B-Spline model can be used in this context.

B-Spline B-Spline is a computationally favorable estimation of spline functions where an order k B-spline is constructed by joining several piecewise polynomials of degree $k-1$ in a variable u. In B-spline form, a curve is represented by combining the control points and the basis functions. A basis function to represent B-spline model is:

$$x(t) = \sum_{i=0}^{n} N_{i,p}(\bar{y}(t))P_i \tag{6}$$

where $N_{i,p}(\bar{y}(t))$ is the pth-degree basis function of B-Spline and P_i is the ith control point. The input to a B-spline approximation is a set of parameters selected along data points range. The parameters determine the shape of the curve because inappropriate selection of parameters could lead to an unpredictable curve estimation. There are two basic ways to select these parameters, uniform and non-uniform knot vector. The knots are uniform if they are equally spaced (i.e., $u_{i+1} - u_i$ is constant for $0 \le i \ge m-1$), otherwise, it is called non-uniform knots vector. The knot vector is generated using "deBoor" formula. The knots can be defined as division points that subdivide the interval $[u_o, u_m]$ into knot spans and the basis functions should have their domain on the knot vectors range. Suppose $n+1$ parameters are used $\{t_0, t_1, \ldots, t_n\}$ and the B-spline degree is p, there will be $m+1$ knots to be used in the curve modeling, where $m = n+p+1$. Knots vector $U = \{u_0, u_1, \ldots, u_m\}$ is computed using the formula:

$$\begin{aligned} u_0 = u_1 = \cdots = u_p = 0 \\ u_{j+p} = \frac{1}{p}\sum_{i=j}^{j+p-1} t_i \quad for j = 1, 2, \ldots, n-p \\ u_{m-p} = u_{m-p+1} = \cdots = u_m = 1 \end{aligned} \tag{7}$$

The values of basis functions with degree p are defined recursively as follows:

$$N_{i,0}(\bar{y}(t) = \begin{cases} 1 & \text{if } u_i \le \bar{y}(t) < u_{i+1} \\ 0 & \text{otherwise} \end{cases} \tag{8}$$

$$\begin{aligned} N_{i,p}(\bar{y}(t)) = & \frac{\bar{y}(t) - u_i}{u_{i+p} - u_i} N_{i,p-1}(\bar{y}(t)) \\ & + \frac{u_{i+p+1} - \bar{y}(t)}{u_{i+p+1} - u_{i+1}} N_{i+1,p-1}(\bar{y}(t)) \end{aligned} \tag{9}$$

In order to compute a point of B-Spline curve at a fixed point of the input u value, the knot span where u point lies should be found and then, the nonzero basis functions are calculated, directly multiplying the value with the corresponding control points. See (10).

$$x(t) = N_{0,p}(\bar{y}(t))P_0 + N_{1,p}(\bar{y}(t))P_1 + \cdots + N_{i,p}(\bar{y}(t))P_i \tag{10}$$

From (4) and (18) and the assumption that $b_0 = 1$, the process output can be approximated as

$$\begin{aligned} N_{0,p}(\bar{y}(t))P_0 = & \sum_{k=-\frac{n-1}{2}}^{k=\frac{n-1}{2}} G(e^{-j\omega k} f_k(t)) - N_{1,p}(\bar{y}(t))P_1 \\ & - N_{2,p}(\bar{y}(t))P_2 - \cdots - N_{i,p}(\bar{y}(t))P_i \end{aligned} \tag{11}$$

Parameter estimation for linear subsystem and inverse static nonlinear is performed in least squares scheme. The regression form is structured based on process output equation (12):

$$\Theta^T = \left[G(e^{j0})\ G(e^{j\omega})\ \cdots\ G(e^{j\frac{n-1}{2}\omega})\ G(e^{-j\frac{n-1}{2}\omega})\ P_1\ \cdots\ P_i\right] \tag{12}$$

and the corresponding regression vector as

$$\phi(k)^T = [A(t)\ B(t)] \tag{13}$$

where

$$A(t) = \left[f_0(t)\ f_1(t)\ f_{-1}(t)\ \dots\ f_{\frac{n-1}{2}}(t)\ f_{-\frac{n-1}{2}}(t) \right] \quad (14)$$

$$B(t) = [-N_{1,p}(\bar{y}(t))\ -N_{2,p}(\bar{y}(t))\ \dots\ -N_{i,p}(\bar{y}(t))] \quad (15)$$

for data samples 0 to t_n, the vector yields

$$\Phi = \begin{bmatrix} f_0(0) & \cdots & f_{-\frac{n-1}{2}}(0) & -N_{1,p}(y(0)) & \cdots & -N_{i,p}(y(0)) \\ f_0(1) & \cdots & f_{-\frac{n-1}{2}}(1) & -N_{1,p}(y(1)) & \cdots & -N_{i,p}(y(1)) \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ f_0(t_n) & \cdots & f_{-\frac{n-1}{2}}(t_n) & -N_{1,p}(y(t_n)) & \cdots & -N_{i,p}(y(t_n)) \end{bmatrix} \quad (16)$$

The output vector is

$$Y^T = [y(0)\ y(1)\ \cdots\ y(n)] \quad (17)$$

Polynomial model Another alternative model to represent the inverse static output nonlinearity in one step identification is to use a polynomial function. The structure of the polynomial model is

$$x(t) = b_0 + b_1\bar{y}(t) + b_2\bar{y}(t)^2 + b_3\bar{y}(t)^3 + \dots + b_M\bar{y}(t)^M \quad (18)$$

where M is the polynomial order. With the assumption that $b_1 = 1$, the process output can be approximated as the function below:

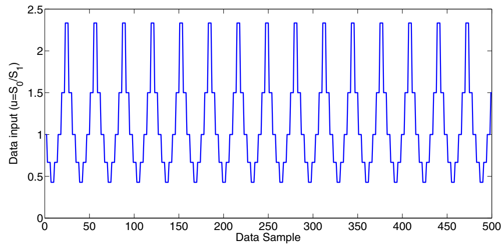
$$\bar{y}(t) = \sum_{k=-\frac{n-1}{2}}^{k=\frac{n-1}{2}} G(e^{-j\omega k} f_k(t)) - b_0 - b_2\bar{y}(t)^2 - \dots - b_M\bar{y}(t)^M \quad (19)$$

The estimation steps are similar to B-Spline, the only change is the inverse model in regression matrix equation (13) replaced by polynomial form. The parameters of linear and nonlinear models for both in Hammerstein and Wiener block identification are estimated in Least Squares method using Prediction Sum of Square (PRESS) residuals which based on the orthogonal decomposition algorithm. All model estimations are executed in MATLAB program using experimental data collected from a test bed.

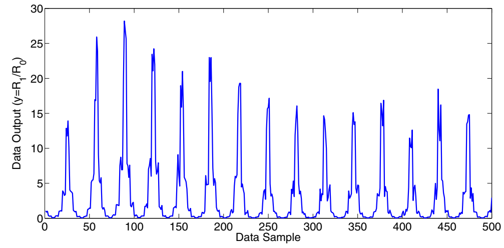
4. IDENTIFICATION RESULTS

Data samples for identification are a set of input-output data which are observed from 500 samples using multi level sinusoidal input with 20 requests/s workloads for each client class. Data Set can be seen in Fig. 5 and the data are divided for model estimation and model validation purposes. The data set is collected from an experimental test-bed used in Patikirikorala [2013] which representing a virtualized software system environment as described in Section 2.

The validation of the estimated models is based on the value of coefficient of determination (R^2) which provides a fitness measurement of predicted model response over the real response. The input signal for the prototype is calculated based on Equation (1) for total available CPU capacity $S_{total} = 100$ and the minimum capacity allocation for a VM is 20 CPU caps. The signal u takes value at $\frac{20}{80}, \dots, 1, \dots, \frac{80}{20}$. The intermediate variable v defined by selecting $v_{min} = -15$, $v_{max} = 15$ and $\delta v = 0.5$. The values of v are $-15, -14.5, \dots, 14.5, 15$.



(a) Data Input



(b) Data Output

Fig. 5. Data set for system identification

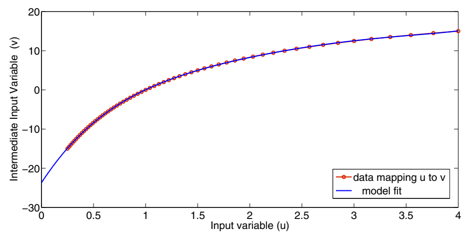
4.1 Hammerstein Model

Using least squares estimation algorithm based on (3), the static input nonlinear model of $v = f(u)$ and the inverse static input nonlinear $u = f^{-1}(u)$ are represented in polynomial function.

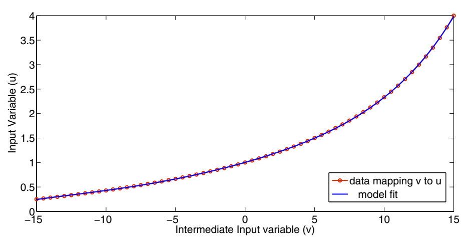
$$v(t) = 0.129u(t)^5 - 1.679u(t)^4 + 8.768u(t)^3 - 24.248u(t)^2 + 40.784u(t) - 23.708 \quad (20)$$

$$u(t) = 4.17e^-7v(t)^5 + 9.34e^{-6}v(t)^4 + 1.018e^{-4}v(t)^3 + 0.0028v(t)^2 + 0.08v(t) + 1.0045 \quad (21)$$

Both of the models can achieve fitness value $R^2 = 0.99$ and the model fit can be seen in Fig.(6).



(a) $v = f(u)$



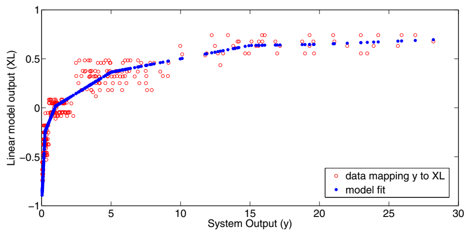
(b) $u = f^{-1}(v)$

Fig. 6. Model fit of Hammerstein model identification

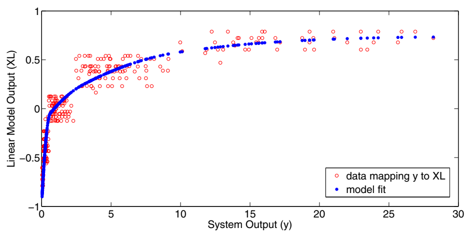
4.2 Wiener Model

To perform Wiener model estimation, some parameters need to be predetermined. For linear FSF model, there are two parameters should be chosen, variable N which is set to 30 and the model order of linear dynamics n is 7. These parameters give the best result among several experiments that have been undertaken by changing the parameter value of N and n.

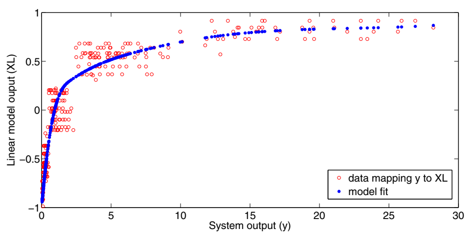
B-Spline function In the inverse model estimation using B-Spline, selecting the knot points in vector parameter can be performed in two ways: uniform and non-uniform knot points. For the case of static nonlinearity in the software system dynamics, the non-uniform knots are the most suitable parameters selection. By evaluating the relationship between FSF linear model output (XL) and system output(y), the vector parameter for knot series derivation for this estimation is a vector consisting of 7 points which are chosen along the range of y values, that is $t_n = [0\ \ 0.25\ \ 1\ \ 5\ \ 15\ \ 20\ \ 30]$. The degree of B-splines is 1 for linear spline, 2 for quadratic spline and 3 for cubic spline.



(a) Linear B-Spline



(b) Quadratic B-Spline



(c) Cubic B-Spline

Fig. 7. Model fit of inverse static output nonlinearity in B-Splines form

From the regressor vector in equation (16), the parameters of both linear model and inverse static output nonlinearity model are estimated. Referring to Fig. 1, by using one-step identification to identify both linear model and inverse static output nonlinearity, the intermediate output point (XL) represents the output of linear model (FSF model) and the output of inverse static output nonlinearity model. See Fig 8 for the intermediate output variable using cubic spline for the inverse static output nonlinear model. The estimated inverse static output model in B-Spline form gives the best Mean Squared Error (MSE) value of 0.02 when using a cubic spline. Moreover, the inverse relationship between system output y and linear model output (XL) can be obtained based on the curve fit of each B-splines model in Figure 7 and their R^2 values can be found in Table 1. It is clear that the best model estimation is the cubic spline representation.

Table 1. R^2 of B-Spline model for inverse output nonlinearity

Model	Linear	Quadratic	Cubic
R^2	0.90	0.91	0.92

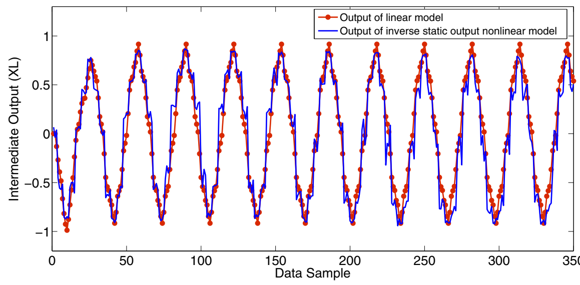


Fig. 8. Model fit in intermediate output variable(XL) using B-Spline form for the inverse static output nonlinearity

Polynomial function For polynomial estimation, only the order of polynomial function related to the estimation of inverse static output nonlinearity dynamic is assumed. The MSE value for the predicted inverse static output model of Wiener model is 0.035 for the 5th order and the value will decrease in the higher orders. Next, for the inverse model of static output nonlinearity, see Table 2 for the R^2 value of three different models and Figure (9) for the model fits with respect to three different orders of the polynomials. The best model fit in polynomial form is the 6th order and the equation is formulated in Eq. (22).

$$\begin{aligned} x(t) = & -0.7015 + \bar{y}(t) - 0.3016\bar{y}(t)^2 + 0.0401\bar{y}(t)^3 \\ & -0.0026\bar{y}(t)^4 + 7.8e^{-5}\bar{y}(t)^5 - 9.03e^{-7}\bar{y}(t)^6 \end{aligned} \tag{22}$$

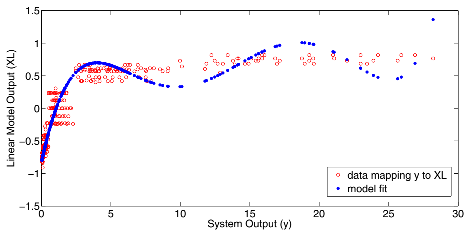
Table 2. R^2 of polynomial model

Model order	5	6	7
R^2	0.88	0.89	0.87

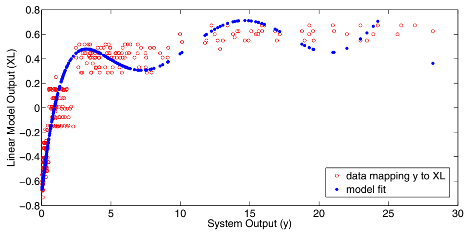
The model fit at intermediate output point (XL) of Wiener block in Fig. 1 as the estimation result between linear model (FSF model) output and the inverse static output nonlinear model output using 6th order polynomial form can be seen in Fig.10.

5. CONCLUSION

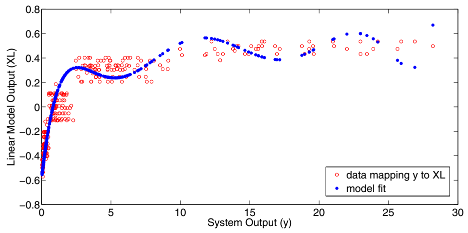
A straightforward estimation for Wiener model in Hammerstein-Wiener system identification of software system dynamics are more accurate if the static nonlinearities are represented in B-splines terms. It is clear that employing FSF and B-spline functions in Wiener model estimation could produce less error in model prediction. In addition, the proposed approaches from this study confirmed that B-spline function has the ability to capture nonlinear characteristics which naturally occur in software system.



(a) 5^{th} order



(b) 6^{th} order



(c) 7^{th} order

Fig. 9. Model fit of inverse static output nonlinearity in polynomial form

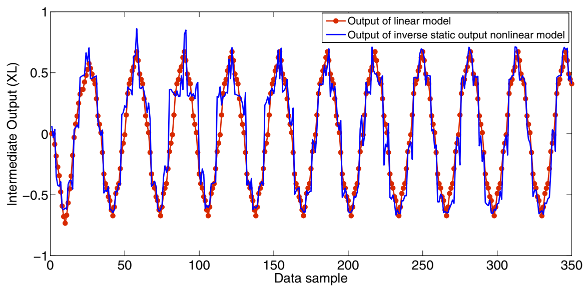


Fig. 10. Model fit in intermediate output variable(XL) using polynomial form for the inverse static output nonlinearity

REFERENCES

R. R. Bitmead, . Anderson. *Adaptive frequency sampling filters. IEEE Transactions on Circuits and Systems*, vol 28, pages 524-534, 1981.

A.L. Cervantes, O.E. Agamennoni, J.L. Figueroa. *A nonlinear model predictive control system based on wiener piecewise linear models. Journal of Process Control*, vol 13, pages 655-666, 2003.

C. Lu, T.F. Abdelzaher, J.A. Stankovic, and S.H. Son. A feedback control approach for guaranteeing relative delays in web servers. *IEEE Real-Time Technology and Applications Symposium*, pages 51–56, 2001.

C. DeBoor. *A Practical Guide to Splines. Applied Mathematical Sciences*, Springer-Verlag, New York, 1978.

X. Hong, Y. Gong, S. Chen. *A Wiener model for high power amplifiers using B-spline function approximation. International Conference of Digital Signal Processing*, pages 1-5, 2011.

F. Jurado. *Hammerstein-model-based predictive control of micro-turbines. International Journal of Energy Research*, vol 7, pages 511-521, 2006.

A.D. Kalafatis, N. Arifin, L. Wang, W.R. Cluett. *A new approach to identification of pH processes based on Wiener model. Chemical Engineering Science*, vol 50, pages 3693-3701, 1995.

A.D. Kalafatis, L. Wang, W.R. Cluett. *Identification of Wiener-type nonlinear systems in a noisy environment. International Journal of Control*, vol 66, pages 923-941, 1997.

C. Lu, Y. Lu, T. Abdelzaher, C. Lao, and G. Tao. *An adaptive control framework for QoS guarantees and its application to differentiated caching services. IEEE International Workshop Quality of Service*, pages 23-32, 2002.

Y. Lu, T. Abdelzaher, C. Lu, L. Sha, and X. Liu. *Feedback control with queueing-theoretic prediction for relative delay guarantees in web servers. IEEE Real-Time and Embedded Technology and Applications Symposium*, pages 208–217, 2003.

T. Patikirikorala, L. Wang, A. Colman, J. Han. *Hammerstein-wiener nonlinear model based predictive control for relative QoS performance and resource management of software systems. Control Engineering Practice*, vol 20, pages 49-61, 2011.

T. Patikirikorala, L. Wang, A. Colman, J. Han. *Differentiated performance management in virtualized environments using nonlinear control.* `http://www.ict.swin.edu.au/personal/tpatikirikorala/docs/tharindu_pds_2013.pdf`, 2013.

L. Wang, W. R. Cluett. *A more efficient way to estimate step response coefficients using the FSF model. Proceedings of 1994 American Control Conference*, vol 1, pages 532-535, 1994.

L. Wang, W. R. Cluett. *Frequency sampling filters: an improved model structure for step response identification. Automatica*, vol 33, pages 939-944, 1997.