2016 Australian Control Conference
November 3-4, 2016. Newcastle, Australia

Block-oriented nonlinear model based control for relative performance and resource management in virtualized software system

Dharma Aryani[1,2], Liuping Wang[1] and Tharindu Patikirikorala [3]

Abstract— **This paper presents an application of feedback control system for relative performance management in virtualized software system. The system dynamics are characterized in a block-oriented nonlinear system identification. The models are estimated in Hammerstein-Wiener structure. The advantage of this approach is the elimination of nonlinearity impact to the performance management by consolidating nonlinear compensators in the feedback control loop. PI controller parameters are designed based on the step response curve generated from Frequency Sampling Filter functions. The experiments conducted in a two-class of virtual machines environment have shown the control system robustness with a great performance stability in the presence of unpredictable disturbances.**

I. INTRODUCTION

The recent development of virtualization based computing technology leads to an increasing need for better Quality-of Service (QoS) guarantees. It is very crucial to provide reliable services to the users to run their applications on the shared infrastructure. In all respects, the basic concept of a shared resources environment is to accommodate sufficient resource provisioning for the users in virtual machines. However, it turns to be very challenging task when multiple virtual machines (VMs) running applications with different performance objectives and under unpredictable workloads changes. In optimization terms for resource and performance management , feedback control approach have been identified as a favorable method to maintain the performance management stability [1]. However, dealing with relative performance management objectives of the multi-class shared resource environments, the existence of nonlinear dynamics is the biggest concern. Earlier works in [2], [3] found that the relationship between the performance properties such as response time and throughput (controlled variable) and the resource allocation as the manipulated variable of a single client class is known to be nonlinear.

To proceed the integration of control engineering technique, the whole dynamics of target system should be characterized. In [4], [5], the authors demonstrated the efficacy of system identification in Hammerstein-Wiener manner to estimate the linear and nonlinear characteristics of virtualized software system. Dynamics of the system can be captured more comprehensively, by means of the linear and nonlinear behaviour are identified in a block-oriented structure. The other existing approaches for performance management in software systems are neglecting the nonlinearity dynamics. The relationship between the input and output of the software system is estimated using linear models only, such as in [6], [7]. Furthermore, several studies have demonstrated research for automated management of resources and QoS performance using linear models [8], [9], [7], [10], [11], [12], [13], [14].

[1,2]Dharma Aryani and Liuping Wang are with School of Electrical and Computer Engineering, RMIT University, Australia, s3351269@rmit.edu.au

[1]Dharma Aryani is also with The State Polytechnic of Ujung Pandang, Indonesia, dharma.aryani@poliupg.ac.id

[3]Tharindu Patikirikorala is with Faculty of Computer Engineering, Swinburne University of Technology, Australia

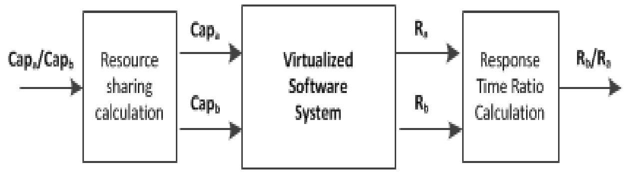

Fig. 1: Relative performance management scheme

A relative management scheme specifies the relative importance of clients and controls the ratio of QoS parameters to the desired levels. Each consecutive clients is paired and the ratios are computed between the pairs. The CPU capacity is allocated to dynamically manage the response time as the controlled variable of QoS parameter. Figure 1 illustrates the input and output variables in the framework of a relative management objectives for *I* client classes.

The main contribution of this paper is the implementation of feedback control system design for relative performance management in a shared resources environment. The control technique is applied in Hammerstein-Wiener feedback structure where the dynamics nonlinearity is compensated by the inverse function of static input and output nonlinearities. Therefore, the target system can be treated as a linear system. The advantage of this approach is the elimination of nonlinearities impact to the performance management by consolidating nonlinear compensators in the feedback control loop. PI controller parameters are designed based on the step response curve generated from Frequency Sampling Filter functions. Experiments are conducted in a two-class virtual machines environment in an equal performance differentiation scheme. Results have shown the control system robustness with a great system stability in the presence of unpredictable disturbances.

II. VIRTUALIZED SOFTWARE SYSTEM

Real system to represent virtualized software sytem is established by using RUBiS application. It is an online auction site benchmark for multi-tiers application of e-commerce website to modeling the behavior of ebay.com. RUBiS has

978-1-922107-90-9

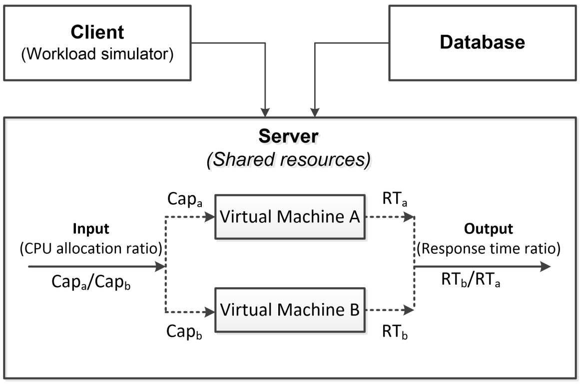

Fig. 2: Virtualized software system [5]

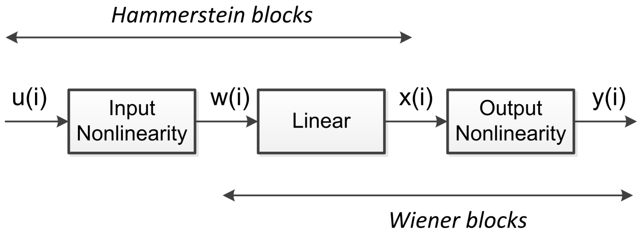

Fig. 3: Hammerstein-Wiener structure [16]

been used in some earlier studies in the field of software system management (eg. [12], [15]). The virtual machines are constructed using $Xen2.6$ hypervisor. This hypervisor comes with a credit-based scheduler, a default scheduler for proportional CPU allocation to each VM. All the shared resources are belong to the server machine and will be shared among the clients. Figure 2 shows the structure of the testbed which consists of a server, a database and a client simulator machine. The server is functioning as the host machine while the two VMs are the guest machines.

The input and output data pairs for system identification are generated from built the testbed of an environment of two virtual machines (VM_a and VM_b). To represent relative management scheme, input and output variables are defined as the ratio values of the related variables from two virtual machines. The input parameter is resource allocation ratio representing the CPU capacity entitlement for VM_a over the entitlement for VM_b $u = \frac{Cap_a}{Cap_b}$ while the output is the response time ratio from the time measurement of each VMs to respond to the workload requests $y = \frac{RT_b}{RT_a}$. These parameters are the main metrics which indicate the end-users experience. The portion of resource sharing is in the percentage of total CPU capacity where full CPU capacity equals to 100%. In order to prevent resources shortage when workload requests are very high, the CPU allocation for each VM is constrained to a minimum capacity. For instance, $Cap_{a,min}, Cap_{b,min} = 20$, then the possible operating points configuration will be $n = 61$ points $(u = u_1, u_2, u_3, \cdots, u_{n-1}, u_n)$ with the following sequence,

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

. The order of these points is nonsymmetrical because the deviations between these points are unequal. This condition cause the nonlinear characteristic in input side. Similar behaviour exist in output variable which led by the ratio function in y.

III. Block-Oriented Model

Block-oriented nonlinear model is a type of model representation where Linear Time Invariant subsystem is cascaded with static nonlinear element. The most well-known structures are Hammerstein, Wiener or the combination of them, so called Hammerstein-Wiener (HW). In this paper, the system dynamics are configured in HW framework (Figure 3). The measured variables are input $u(k)$ and output $y(k)$, while $w(k)$ and $x(k)$ are denoted as the intermediate variables representing unmeasurable input and output of linear element. Both of the nonlinearity models capture the nonlinear dynamics on input and output in their inversion function. Further, these inverse static nonlinearity elements will be functioning as compensators for the nonlinear behaviour of the system. The rest of the dynamics are characterized in the linear element. This approach will be advantageous for control system design because by integrating the inverse static nonlinearities in input and output elements, the presence of nonlinear dynamics is sufficiently eliminated and the control system can be designed as a linear system [4]. The inverse static nonlinear models are estimated in polynomial functions and the linear model is formulated in Frequency Sampling Filter function. The details of system identification referred to the earlier findings of the authors ([16], [5]).

Static Input Nonlinearity

Nonlinear model in Hammerstein block is the inverse static input nonlinearity function which performs as pre-input compensator in the feedback control loop. The model captures the relationship between input signal u and intermediate input w. Firstly, the operating points u are transformed to a range of evenly spaced operating points. The technique was formulated by defining an intermediate variable w where $w_{min} \leq w \leq w_{max}$. To have an equally spaced operating points, a fixed deviation is determined by $\delta w = \dfrac{w_{max} - w_{min}}{n-1}$.

$$w = w_1, w_2, w_3, \cdots, w_n$$

where $w_1 = w_{min}, w_2 = w_1 + \delta w, w_3 = w_2 + \delta w$ and $w_n = w_{max}$. Secondly, the respective operating points of input u_i is mapped to the w_i points. Furthermore, the relationship between data pairs from the mapping is estimated in a polynomial function $u(i) = f^{-1}(w(i))$,

$$u(i) = \alpha_0 + \alpha_1 w(i) + \alpha_2 w(i)^2 + \cdots + \alpha_a w(i)^a \qquad (1)$$

Accordingly, the coefficients of $u(i)$ function are approximated by using least squares method, see [5],

$$\hat{\Theta} = (\Phi^T \Phi)^{-1} (\Phi^T U) \qquad (2)$$

where θ is parameter vector and and ϕ is data vector.

Linear: The linear model is represented in Frequency Sampling Filters (FSF) model. FSF model estimation only requires prior knowledge about settling time of the process with an assumption that the system is stable, linear, and time-invariant process. The FSF model is formulated as follows

$$x(i) = \sum_{l=-\frac{m-1}{2}}^{\frac{m-1}{2}} G(e^{j\omega_l m}) f_l(i) w(i) \tag{3}$$

where $f_l(i) = (\frac{1}{M}\frac{1-e^{-j\omega_l}}{1-e^{j\omega_l m}e^{-j\omega_l}})$
Equation 3 is associated as the j-th FSF with $\omega_l = \frac{2\pi l}{M}$ as the center frequency in $l = 0, \pm 1, \pm 2, \cdots, \pm\frac{m-1}{2}$. m is the effective order which indicates the significant parameters of FSF model and M represents the order of individual FSF filters corresponding to the process settling time $M = T_s/\delta t$ and δt is a sampling interval. m should be much smaller than M and in odd number.

Static Output Nonlinearity: Nonlinear model in Wiener block is assigned to get the relationship between output y and intermediate output x. It is expressed in inverse function $x(i) = g^{-1}(y(i))$ and will be used as a compensator for the output nonlinear characteristics. In a similar way with the pre-input compensator, the post-output compensator is estimated in a polynomial function

$$x(i) = \beta_0 + \beta_1 y(i) + \beta_2 y(i)^2 + \cdots + \beta_b y(i)^b \tag{4}$$

Furthermore, model parameters of linear and inverse static output nonlinearity models are approximated in a one-step manner by equating the $x(i)$ functions from (3) and (4). It leads to a process output function. With an assumption that inverse static function is a single-valued smooth function and $\beta_1 = 1$, the output function is formulated as follows

$$\hat{y}(i) = \sum_{l=-\frac{m-1}{2}}^{\frac{m-1}{2}} G(e^{j\omega_l}) f_l(i) w(i) - \beta_0 - \beta_2 y(i)^2 - \cdots - \beta_b y(i)^b \tag{5}$$

Afterwards, the model parameters of linear and inverse static output nonlinearity are obtained by implementing least squares method (Eq.2).

IV. Feedback Control Design

The target system is software system, the performance metrics to be controlled is response time which is monitored by sensors. The control input signal to push the system gives different response is resources provisions. The allocation will be adjusted through the actuator. The sensor and actuator are establised in a sort of algorithms for proper incorporation to the real software system. The main objective of control implementation is to maintain the outputs of the system sufficiently close to the reference value, by adjusting the control inputs. Proportional-Integral (PI) is considered to be the suitable controller for this SISO system on account of its simplicity and robustness for disturbance rejection [2]. Over the integration of nonlinearity compensators, the target system is assumed as linear system (Fig. 4). Therefore, the system can be controlled by any type of linear controller without concerns of nonlinearity issues. In this structure, the output of the controller is the intermediate input (w) signal. Practically, this control signal should be converted to the real target system input (u) by passing the signal through the inverse static input nonlinear block model. Consequently, the output of the feedback system is the real output y. Then by passing through $g^{-1}(y)$, the output is transformed to x which is then compared with the transformed version of setpoint r.

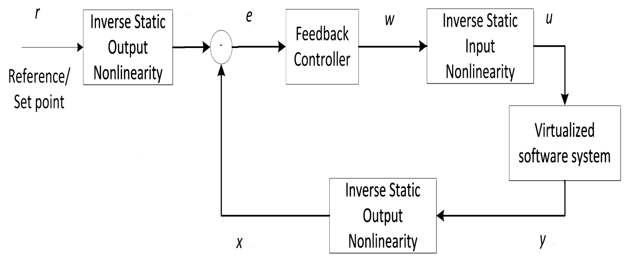

Fig. 4: Feedback control loop

Consequently, in order to apply the basic tuning rules of PI gains, the linear model (FSF functions) is transfered to step response model. Refer to [17], the step response at sampling instant $'l'$, where $l = 0, 1\cdots, M-1$ can be formulated as the equation below

$$\hat{g}_l = \sum_{i=0}^{l-1} \hat{f}_i \tag{6}$$

where the relationship of impulse response coefficients $(\hat{f}_o, \hat{f}_1, \hat{f}_2, \cdots, \hat{f}_{l-1})$ with the frequency response of FSF is linked to the following equation

$$\hat{f}_i = \frac{1}{M} \sum_{l=-\frac{m-1}{2}}^{\frac{m-1}{2}} \hat{G}(e^{j\omega_l}) e^{j\omega_l i} \tag{7}$$

The estimated step response coefficients is obtained by substituting Eq. 6 int Eq. 7,

$$\hat{g}_l = \sum_{l=-\frac{m-1}{2}}^{\frac{m-1}{2}} \hat{G}(e^{j\omega_l}) \frac{1}{M} \frac{1-e^{-j\omega_l(l+1)}}{1-e^{j\omega_l}} \tag{8}$$

A. Estimated Models

400 samples of input and output data are collected from the test-bed and used in a block-oriented nonlinear system identification approach as explained in Section 3.

Two system identification procedures are performed using different settings of FSF function parameters. The objective is to evaluate the performance of feedback controls when the system is estimated in lower and higher model order. Each SID gives inverse static input nonlinearity, inverse static output nonlinearity and a transfer function of linear model. The first setting is $M = 39$ and $m = 3$ (Model A), and the the second estimation is $M = 100$ and $m = 13$ (Model B). Meanwhile, the polynomial order for all nonlinear models in both settings is $a = b = 5$. The inverse static input nonlinear

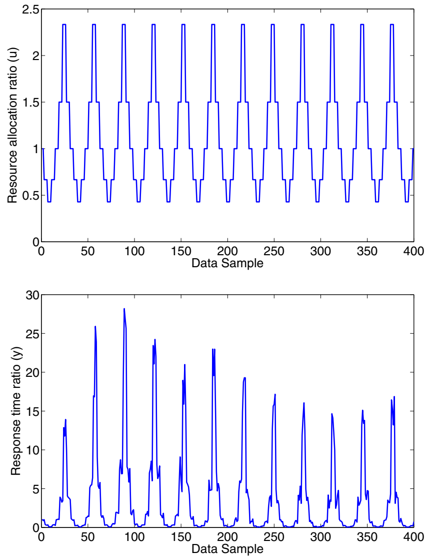

Fig. 5: Data set for system identification

model is similar for both settings since the input operating points u and the intermediate input variable w are maintained the same as in [5]. The identified mathematical models are

- Inverse static input nonlinearity

$$u(k) = 4.17e^{-7}w(k)^5 + 9.34e^{-6}w(k)^4 + 1.018e^{-4}w(k)^3 + 0.0028w(k)^2 + 0.08w(k) + 1.0045 \tag{9}$$

- Linear Model
Step responses of the estimated FSF functions in Eq. (3) are presented in Fig. 6 for model A and Fig. 7 for model B.

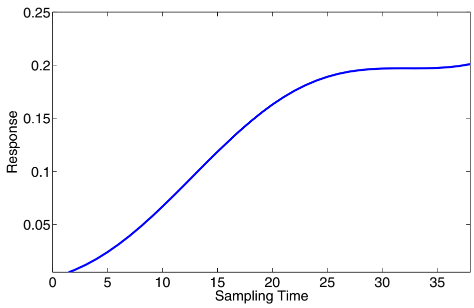

Fig. 6: Step Response Curve of Linear Model A

- Inverse static output nonlinearity
From estimation Model A

$$x(k) = 1.6e^{-4}y(k)^5 - 0.006y(k)^4 + 0.078y(k)^3 - 0.43y(k)^2 + y(k) - 0.57 \tag{10}$$

From estimation Model B

$$x(k) = 1.7e^{-4}y(k)^5 - 0.006y(k)^4 + 0.08y(k)^3 - 0.44y(k)^2 + y(k) - 0.56 \tag{11}$$

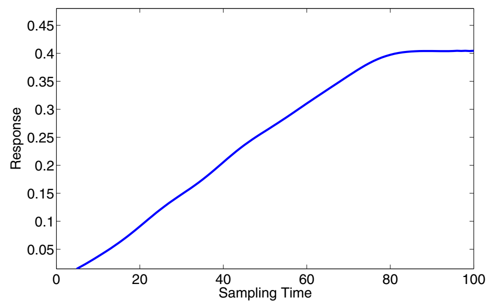

Fig. 7: Step Response Curve of Linear Model B

B. Tuning PI gains

The PI controller gains are tuned using the tunning rules of Wang-Cluett (see [18]). The referred method calculates the controller parameter K_p and τ_I using the value of steady-state gain (K_{ss}), time constant (T) and time delay (d) which are attained from the step response plot.

- Model A $K_{ss} = 0.2034$, $T = 19$ and $d = 3$
- Model B $K_{ss} = 0.4096$, $T = 63$ and $d = 8$

Using Wang-Cluett PI tuning rules, the PI controller gains are calculated. The gain values are in Table I.

TABLE I: PI Controller Parameter

FSF function	K_p	τ_I
ModelA	16.6	17.67
ModelB	10.8	43.48

V. Experimental Results

This section compares the performance of feedback controllers which are designed based on two different frequency response models. Throughout the experiments, the performance differentiation ratio is equal. Thus, VM_a and VM_b are in the same importance level. By means of the setpoint as reference value during runtime is set to 1.The experiments are carried out in two conditions; nominal workloads and varying high workloads. The intention is to evaluate the disturbance rejection capability of the controller.

A. Case I: Nominal workloads

Workloads are simulated 100 concurrent users for each VM in all sampling time. Fig. 8 and Fig. 9 illustrated the output response and control signal from the feedback control experiment. It can be seen that the feedback system with Model B gives better performance than Model A. Controller from Model B has more stable control signal which leads to more steady output response compare to Model A.

B. Case II: Separate high workloads

In the first 60 samples, 100 concurrent users are simulated for both VM_a and VM_b. However, after the 30th sample, workloads of VM_a increases suddenly to 200. At the 95th sample, the number of users of VM_b increases to 200 while the concurrent users for VM_a reduce back to the nominal

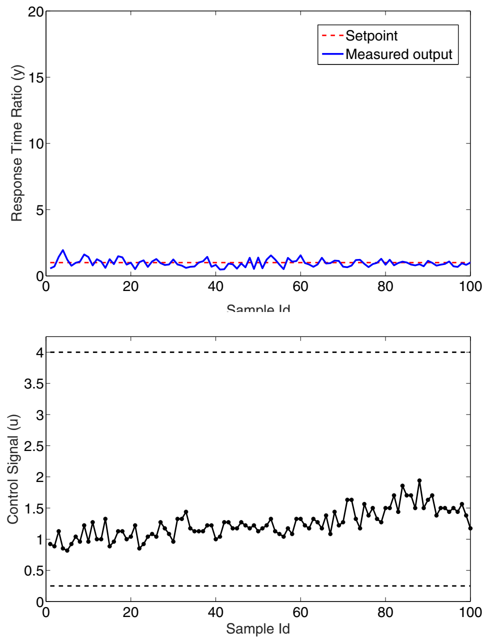

Fig. 8: Output response and control signal of Model A in Case I

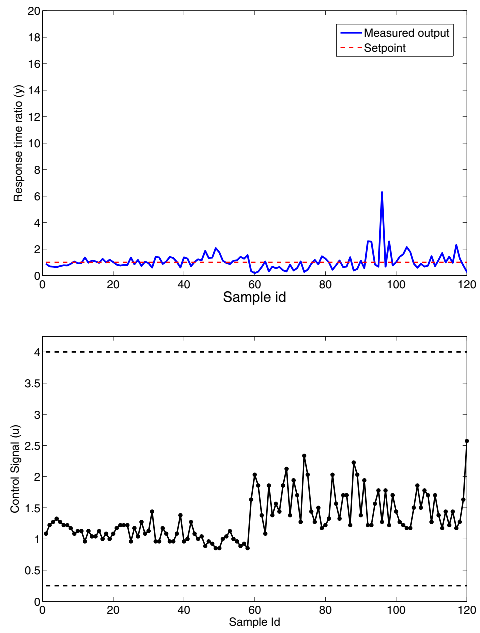

Fig. 10: Output response and control signal of Model A in Case II

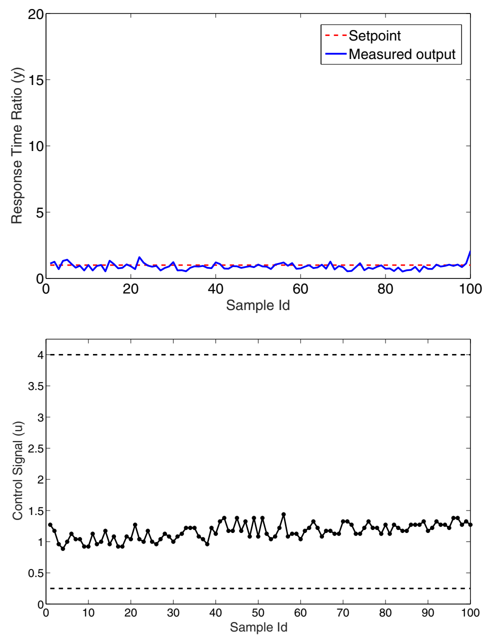

Fig. 9: Output response and control signal of Model B in Case I

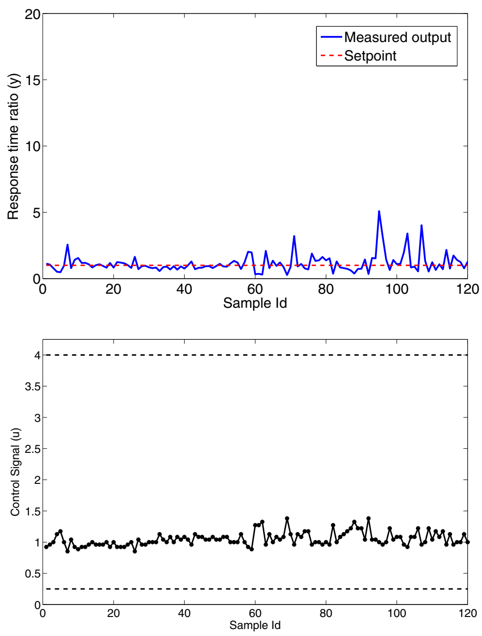

Fig. 11: Output response and control signal of Model B in Case II

workloads (100). This setting is the presentation of a scenario when unbalance high resource demand arrive to the virtual machines and the equal performance differentiation level is set as the reference value. This condition requires efficient CPU management to assist the output response of target system to follow the reference value. Figure 10 and 11 exhibit the feedback control results for this case.

The output response from both models achieve the set point with small errors during the balance nominal workloads. Then, when the workload for VM_a and VM_b alternately increased, the output response of both models show some overshoots. In details, feedback control of model A gives a better output response when sudden disturbances appear in the VM environment. However, its control signal is more oscillatory (bigger slew rate) compare to model B which has smoother control signal.

VI. CONCLUSIONS

Feedback control application based on a structure of block-oriented nonlinear system identification has been implemented in virtualized software system. PI controller parameters are designed based on the step response curve of the FSF functions. The experimental results prove the robustness of the controller. The performance of feedback control system with the integration of pre-input and post-output compensators shows great satisfactory in steady state performance.

REFERENCES

[1] T. Patikirikorala, A. Colman, J. Han, and L. Wang, "A systematic survey on the design of self-adaptive software systems using control engineering approaches," in *Proceedings of the 7th International Symposium on Software Engineering for Adaptive and Self-Managing Systems*. IEEE Press, 2012.

[2] J. L. Hellerstein, Y. Diao, S. Parekh, and D. M. Tilbury, *Feedback control of computing systems*. John Wiley & Sons, 2004.

[3] L. Chenyang, T. F. Abdelzaber, J. A. Stankovic, and S. H. Son, "A feedback control approach for guaranteeing relative delays in Web servers," in *Real-Time Technology and Applications Symposium, 2001. Proceedings. Seventh IEEE*, 2001.

[4] T. Patikirikorala, L. Wang, A. Colman, and J. Han, "Hammerstein–Wiener nonlinear model based predictive control for relative QoS performance and resource management of software systems," *Control Engineering Practice*, vol. 20, no. 1, 2012.

[5] D. Aryani, L. Wang, and T. Patikirikorala, "An improved hammerstein-wiener system identification with application to virtualized software system," in *Control Applications (CCA), 2015 IEEE Conference on*. IEEE, 2015, pp. 1552–1557.

[6] C. Lu, T. F. Abdelzaber, J. A. Stankovic, and S. H. Son, "A feedback control approach for guaranteeing relative delays in web servers," in *Real-Time Technology and Applications Symposium, 2001. Proceedings. Seventh IEEE*. IEEE, 2001, pp. 51–62.

[7] Y. Lu, T. Abdelzaher, C. Lu, L. Sha, and X. Liu, "Feedback control with queueing-theoretic prediction for relative delay guarantees in web servers," in *Real-Time and Embedded Technology and Applications Symposium, 2003. Proceedings. The 9th IEEE*. IEEE, 2003, pp. 208–217.

[8] Y. Lu, A. Saxena, and T. F. Abdelzaher, "Differentiated caching services; a control-theoretical approach," in *Distributed Computing Systems, 2001. 21st International Conference on*. IEEE, 2001, pp. 615–622.

[9] Y. Lu, T. Abdelzaher, C. Lu, and G. Tao, "An adaptive control framework for qos guarantees and its application to differentiated caching," in *Quality of Service, 2002. Tenth IEEE International Workshop on*. IEEE, 2002, pp. 23–32.

[10] M. Karlsson, X. Zhu, and C. Karamanolis, "An adaptive optimal controller for non-intrusive performance differentiation in computing services," in *Control and Automation, 2005. ICCA'05. International Conference on*, vol. 2. IEEE, 2005.

[11] D. Kusic and K. Nagarajan, "Risk-Aware Limited Lookahead Control for Dynamic Resource Provisioning in Enterprise Computing Systems," in *Autonomic Computing, 2006. ICAC '06. IEEE International Conference on*, 2006, pp. 74–83.

[12] P. Padala, K. G. Shin, X. Zhu, M. Uysal, Z. Wang, S. Singhal, A. Merchant, and K. Salem, "Adaptive control of virtualized resources in utility computing environments," in *ACM SIGOPS Operating Systems Review*, vol. 41, no. 3. ACM, 2007, pp. 289–302.

[13] W. Pan, D. Mu, H. Wu, and L. Yao, "Feedback control-based QoS guarantees in web application servers," in *High Performance Computing and Communications, 2008. HPCC'08. 10th IEEE International Conference on*. IEEE, 2008, pp. 328–334.

[14] K. Wu, D. J. Lilja, and H. Bai, "The applicability of adaptive control theory to QoS design: limitations and solutions," in *Parallel and Distributed Processing Symposium, 2005. Proceedings. 19th IEEE International*, apr 2005, pp. 8 pp.–.

[15] T. Patikirikorala, "Using Nonlinear Control of Resources to Achieve Differential Performance Objectives in Software Systems," vol. PhD Thesis, 2012.

[16] D. Aryani, L. Wang, and T. Patikirikorala, "Control oriented system identification for performance management in virtualized software system," *IFAC Proceedings Volumes*, vol. 47, no. 3, pp. 4122–4127, 2014.

[17] L. Wang and W. R. Cluett, "A more efficient way to estimate step response coefficients using the FSF model," in *American Control Conference, 1994*, vol. 1. IEEE, 1994, pp. 532–535.

[18] L. Wang and W. Cluett, "Tuning pid controllers for integrating processes," in *Control Theory and Applications, IEE Proceedings-*, vol. 144, no. 5. IET, 1997, pp. 385–392.