Available online at www.sciencedirect.com

ScienceDirect

IFAC PapersOnLine 50-1 (2017) 7795–7800

Data-Driven Predictive Control with Nonlinear Compensation for Performance Management in Virtualized Software System

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

* *School of Electrical and Computer Engineering, RMIT University, Australia (e-mail: s3351269@student.rmit.edu.au). Department of Electrical Engineering, The State Polytechnic of Ujung Pandang, Indonesia*

** *School of Electrical and Computer Engineering, RMIT University, Australia (e-mail: liuping.wang@rmit.edu.au)*

*** *Faculty of Information and Communication Technology, Swinburne University of Technology, Australia(e-mail: tpatikirikorala@swin.edu.au)*

Abstract: This paper deals with data-driven predictive control for relative performance management in virtualized software system. The system dynamics are characterized in Hammerstein-Wiener structure to capture nonlinear and linear characteristics. The proposed control approach is the implementation of Subspace-based Predictive Control with the integration of nonlinear compensation. The compensator functions are inverse static input and output nonlinearity models from the Hammerstein-Wiener system identification. The subspace predictors are formulated from the linear model input and output of Wiener block. The experimental results from three scenarios of performance objectives show the reliability of Subspace-based Predictive Control to manage the virtualized software system.

Keywords: data-driven control, subspace predictive control, Hammerstein-Wiener, nonlinear compensation, virtualized software system

1. INTRODUCTION

Data-driven predictive control is a synthesized technique of system identification with control system design which is also called as Subspace-based Predictive Control (SPC). This approach incorporates the subspace state space estimation into Model Predictive Control (MPC) structure. Series of earlier works in this class of control and identification approach have been conducted by Di Ruscio and Foss (1998); Favoreel et al. (1998); Kadali et al. (2003); Barry and Wang (2004); Mardi and Wang (2009). The novelty of SPC is the use of subspace linear predictor to estimate the future output of the system without performing explicit parameterization of the conventional model. Subspace identification method identifies certain matrices to capture the relationship between the process inputs and outputs in non-parametric coefficients form. Then, the future output of the system is predicted by a linear function of past input, past output, and future input values. In addition, SPC is numerically robust and very attractive for on-line implementation since it uses QR-decomposition to generate the subspace coefficients directly from I/O data.

Model Predictive Control (MPC) is a model-based control algorithm which objective is to find the future control input in a finite-time prediction horizon. The control algorithm is formulated upon a numerical minimization of a cost function in a receding horizon principle. The design of MPC use the system matrices obtained from subspace system identification with a guarantee of reducing design complexity for MPC gain calculations from the real experiment data. The system matrices of dynamic model are not explicitly composed since SPC only implementing the subspace predictor variable.

In a virtualized software systems, the provider serves multiple customers by managing a single physical environment to deliver the required performance properties. The main control objective is to perform dynamic resource management where the resources can be allocated efficiently among the clients during runtime. The performance management could be carried out for absolute or relative management objective. In the case of relative scheme, the preferences consideration of performance properties and resource provisioning between the client classes lead to a severe nonlinear dynamics which can be observed at the system input and output. Some constituted factors in software systems, namely demand changes and complex preferences for performance objectives, could provoke noisy characteristics to the environment in management implementation. For noisy system, block-oriented Hammerstein-Wiener structure is considered as a favorable and generic class for dynamic model estimation (Lennart (1999)) since the approach is reliable in system identification for a process with significant nonlinearity issues. To proceed control engineering technique in virtualized software systems, the linear and nonlinear dynamics should be characterized. A useful approach to compensate the nonlinear dynamics in Hammerstein-Wiener structure is by estimating the nonlinearities in their inversion functions (Kalafatis et al. (1997)). Studies by the authors in Patikirikorala et al. (2012); Aryani et al. (2014, 2016b) exhibited the efficacy of system identification in Hammerstein-Wiener manner to identify the linear and nonlinear characteristics of shared resources environment .

Peer review under responsibility of International Federation of Automatic Control.
10.1016/j.ifacol.2017.08.1054

The main contribution of this paper is the implementation of control system design using Subspace Predictive Control with nonlinear compensation in Hammerstein-Wiener structure for virtualized software system. Input and output data set for this study are generated from the experimental testbed of two-clients virtualized software system. The linear model is represented by non-parametric Frequency Sampling Filter (FSF) model and the input and output nonlinearities are formulated in polynomial functions. The estimation of linear and nonlinear models parameters in Wiener block is performed in a straightforward manner. Nonlinear elements are estimated in inverse form since these functions are used as pre-input and post-output nonlinear compensators in the control system loop. This approach could reduce the impact of nonlinearities for relative performance management in virtualized software system.

Presentation in the paper is structured in seven sections. Section 2 covers the dynamic description of virtualized software system, Section 3 presents the Hammerstein Wiener system identification, then the formulation of Subspace-based Predictive Control is addressed in Section 4. The identification results and SPC design are shown in Section 5, followed by the feedback control results in section 6. Section 7 delivers conclusion based on the results from experiments in several runtime scenarios.

2. DYNAMIC SYSTEM DESCRIPTION

2.1 Virtualized software System

A real system of virtualized software environment is established using RUBiS application. It is a multi-tiers application of e-commerce website which channeling the dynamics of *ebay.com*. RUBiS has been a favourable application in the studies of software system management (eg. Patikirikorala et al. (2012)). The computing infrastructures are shared among the installed virtual machines. Three elements are operated and connected on an isolated network. A server machine, a database and a client simulator are set up to run the application. The virtualization using $Xen2.6$ hypervisor which comes with a credit-based scheduler for allocating the resources for the VMs proportionally. To support this scheduler, an actuator was installed to send the preferred ratio of resources to the system and a sensor component was added to each VM to calculate the response time of incoming requests. For this study purposes, the virtualized software system experiments are carried out in relative performance management scheme. Figure 1 shows the testbed structure.

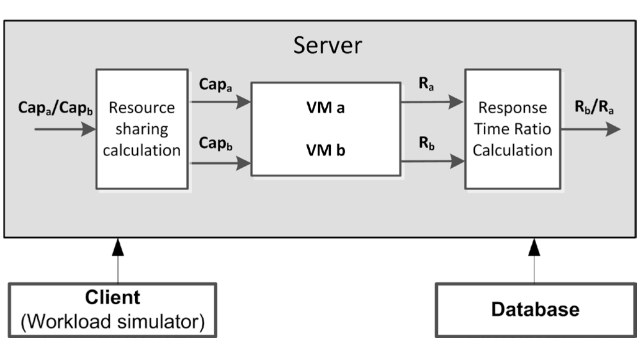

Fig. 1. Virtualized software system

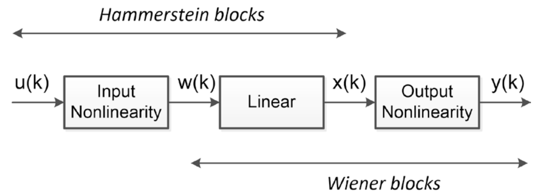

Fig. 2. Hammerstein-Wiener structure

2.2 Dynamic Nonlinearities

In relative guarantee management scheme, input and output variable setting are defined as the ratio values of the two VMs variables. $Cap_a(k)$ and $Cap_b(k)$ are the CPU allocation VM_a and VM_b respectively, where the total CPU capacity $Cap_{total} = Cap_a + Cap_b$. The input variable is $u(k) = \dfrac{Cap_a(k)}{Cap_b(k)}$. The portion of resource sharing is in the percentage of total CPU capacity where full CPU capacity equals to 100%. Therefore, to prevent a shortage of resources when workload requests suddenly increase in unpredictable condition, CPU share is constrained to a minimum capacity. In this experiment, $Cap_{a,min}(k), Cap_{b,min}(k) = 20$ and $Cap_{total} = 100$. Output variable $y(k)$ is the measured response time from each virtual machine. $RT_a(k)$, $RT_b(k)$ are response time to the workloads of VM_a and VM_b respectively. The output variable is $y(k) = \dfrac{RT_b(k)}{RT_a(k)}$.
It is clear that nonlinearities in input and output variables are caused by the ratio formulation between the VMs variable in relative scheme.

3. SYSTEM IDENTIFICATION

This section gives a summary of the Hammerstein-Wiener system identification for virtualized software system dynamics. Figure 2 shows the block structure where nonlinear memory-less blocks are sandwiched by a linear dynamic block. In the Hammerstein block, a nonlinear model is assigned to get the relationship between input signal u and intermediate input w in the form of inverse static nonlinearity function. The estimated model will be employed as compensator for the nonlinear characteristic of the input element. In Wiener block, the linear model is represented in Frequency Sampling Filters function. Moreover, the output nonlinear model is estimated in terms of inverse static nonlinearities by assigning a polynomial function as the predicted nonlinear model.

3.1 Linear model

The linear model is estimated in Frequency Sampling Filters (FSF) function. This model is used to deal with high dimensionality estimation issue when using Finite Impulse Response (FIR). FSF coefficients of the linear model are captured in frequency domain which is acquired from a linear transformation of FIR and composed of narrow bandpass filters (Wang and Cluett (1997)).

Let $f_l(k) = (\dfrac{1}{M}\dfrac{1-z^{-M}}{1-e^{j\omega_l}z^{-1}})w(k)$,

$$x(k) = \sum_{l=-\frac{m-1}{2}}^{\frac{m-1}{2}} G(e^{j\omega_l}) f_l(k) \qquad (1)$$