IOP Conference Series: Materials Science and Engineering

PAPER • OPEN ACCESS

Evaluation of Finite Control Set - Model Predictive Control for Dynamic Resources Management in A Shared-Resources System

To cite this article: D Aryani *et al* 2019 *IOP Conf. Ser.: Mater. Sci. Eng.* **619** 012023

View the article online for updates and enhancements.

Evaluation of Finite Control Set - Model Predictive Control for Dynamic Resources Management in A Shared-Resources System

D Aryani[1], L Wang[2], N A Hidayatullah[3]

[1] Department of Electrical Engineering, State Polytechnic of Ujung Pandang, Indonesia
[2] School of Engineering, RMIT University, Australia
[3] Department of Electrical Engineering, State Polytechnic of Madiun, Indonesia

E-mail: dharma.aryani@poliupg.ac.id

Abstract.

This paper presents an evaluation of Finite Control Set-Model Predictive Control (FCS-MPC) performance for dynamic resources provisioning. Realization of a shared-resources environment is experimented in software system network by virtualization technique. Control parameter regulates the resources provisioning among the users to guarantee the achievement of performance metrics in the network. The objective preferences are maintained in relative term by manipulating a finite set of operating points as the control states. Moreover, an optimization procedure is implemented based on receding horizon prediction of MPC in a feedback control scheme. Further investigation is conducted to examine the performance of optimal FCS and constrained FCS for dynamic system management. The experimental results demonstrate that constrained FCS system provides a better response stability than the optimal FCS. Correspondingly, the range and quantity of input states are significantly influential to the flow of control action. A wider range and longer control state sequence yields to a smoother control signal stream.

1. INTRODUCTION

Optimization scheme in a multi-users networked system is aimed to preserve satisfactory performance objectives by managing the infrastructure provisioning. In a shared resources environment such software system, the principal properties of performance quality are response time and service differentiation. It has been practically confirmed that the settings of output preferences and workload uncertainty are main challenges that could cause performance shortcomings in software system. Therefore, a self-adapting ability is very crucial to be integrated into control system procedure in order to enable dynamics adjustment in unpredictable circumstances [1, 2].

One of predictive control approaches that has been used extensively to control dynamic system with certain states of operating points is Finite Control Set-Model Predictive Control (FCS-MPC). Studies by [3] and [4] can be referred as successful implementation of FCS-MPC to handle the issue of dynamic resource management. The signature of FCS control method is its algorithm simplicity and straightforward application. Earlier findings by [5] and [6] have shown that FCS-MPC is a well-proven predictive techniques for power converters and drives control. In addition, based on [6], FCS-MPC is a preferable control for power electronics. The FCS-MPC technique was initially exposed in [6] and [7]. In these studies, FCS-MPC was analyzed with PWM and the hysteresis method. Similar methods have

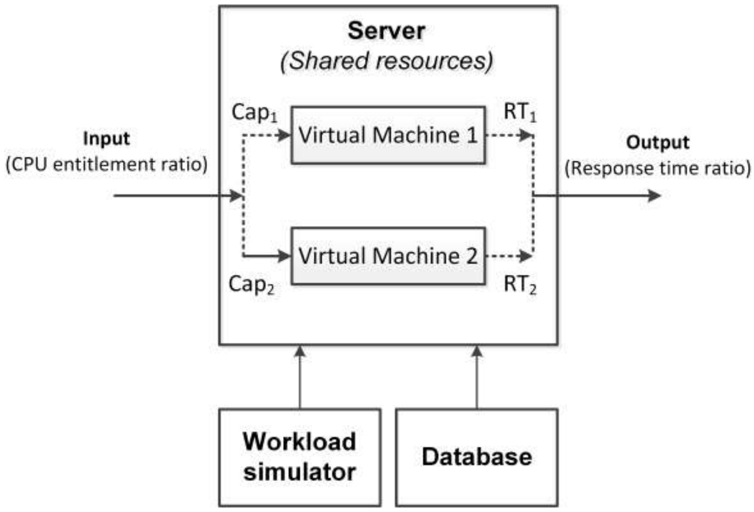

Figure 1: Structure of Virtualized software system

been investigated for various applications, such as power converters ([8, 3]), and the Permanent Magnet Synchronous Motor (PMSM) ([9, 10, 11]).

In FCS-MPC, a control action is selected from optimal switching states based on error value between the predicted and real output value. FCS-MPC calculates the output prediction, and computes cost function based on the finite control states. The optimal solution is the switching state which minimizes prediction error in objective function. This method features simple implementation and leads to a robust performance. In regards to relative management scheme of software system, control parameter is CPU capacity of the server which needs to be managed dynamically in a certain degree. The relative relationship is interpreted by pairing consecutive clients to get the ratio value.

The main contribution of this paper is the implementation of Finite Control Set-Model Predictive Control (FCS-MPC) to optimize dynamic management of a shared-resources system. Two schemes are evaluated, the optimal formulation and constrained FCS. Experiments are carried out in a two-class virtual machines environment for an equal performance differentiation setting. The results exhibit control system robustness with a better system's stability when the constrained FCS algorithm is implemented into the feedback control loop.

The remaining topics of this paper are presented into five sections. Section 2 covers the description of virtualized software system and its model structure. Section 3 presents the formulation of FCS-MPC formulation which comprises of optimal FCS-MPC and constrained FCS-MPC. The experimental results are evaluated in Section 4, with a brief conclusion in Section 5.

2. System Dynamic and Characterization

2.1. Virtualized software System

A test bench for experimentation of a shared-resources environment is attained in a software system network based on *RUBiS* application. *RUBiS* is a site benchmark of e-commerce for multi-tiers applications to emulate the system dynamics of *ebay.com*. The physical structure comprises of a server, database, and workload simulator as depicted in Figure 1. Two virtual machines (VM_1 and VM_2) are installed in the server and act as the clients that will utilize all resources in the server. For the purpose of virtualization, $Xen2.6$ hypervisor is deployed by considering its credit-based scheduler which can be used to manipulate the CPU allocation proportionally to each VM. To execute an automated control in a feedback control method, an actuator is included to actuate the system with the estimated control signal. In addition, a sensor is integrated to measure the time that server needs in order to respond the incoming requests of each client.

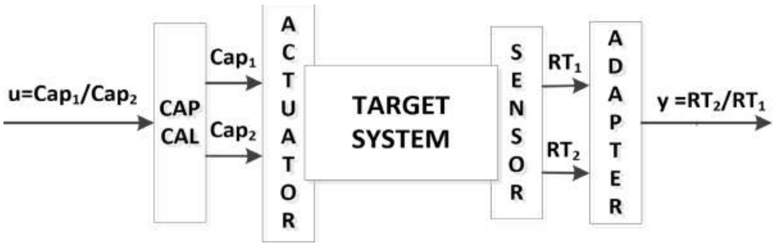

Figure 2: Relative Management Scheme for Two Users

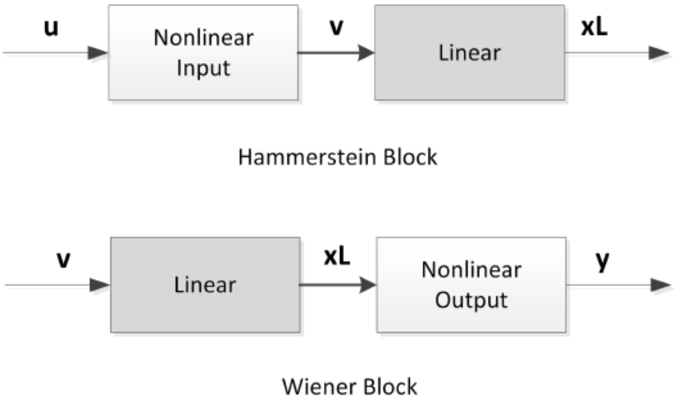

Figure 3: Hammerstein-Wiener Structure

Maintaining the performance objective in relative mode requires a dynamic resources management. In a condition where the level of CPU allocation for a client is higher than the other, it means that more resources have been provisioned to serve the workload (request) from this client. The CPU capacity entitlement ratio for VM_1 over VM_2 ($u(k) = \frac{Cap_1(k)}{Cap_2(k)}$) is input variable, while the full CPU capacity is numerically quantified as 100%. Furthermore, output parameter is response time ratio which is expressed as $y(k) = \frac{RT_1(k)}{RT_2(k)}$, where $RT_1(k)$ and $RT_2(k)$ are the measured time of the server response to the client requests. Figure 2 illustrates the framework of a relative management objectives for two clients.

2.2. Dynamic Model Estimation

Model estimation using a system identification approach refers to the study in [12] and [13]. Characterization of virtualized software system obtained by a system identification for nonlinear dynamics in a form of block-oriented model. Figure 3 shows a Hammerstein-Wiener structure to formulate the dynamic model estimation.

Nonlinear models in input and output element are identified in their inverse function in order to use the models as compensator for the nonlinear characteristic. Then, the rest of system dynamic is taken as linear function. System identification is proceeded by using a set of input and output data which is generated from the experimental testbed. A multi-level sinusoidal input with minimum 100 requests/s workloads for each client class is given into the system input, and the output response for 400 data samples are measured (see Figure 4).

For Hammerstein block, an inverse static input nonlinearity model is a polynomial function from least squares optimization. It is a function of $u(k) = f^{-1}(v(k))$ which is estimated by setting $v(k)$ values in

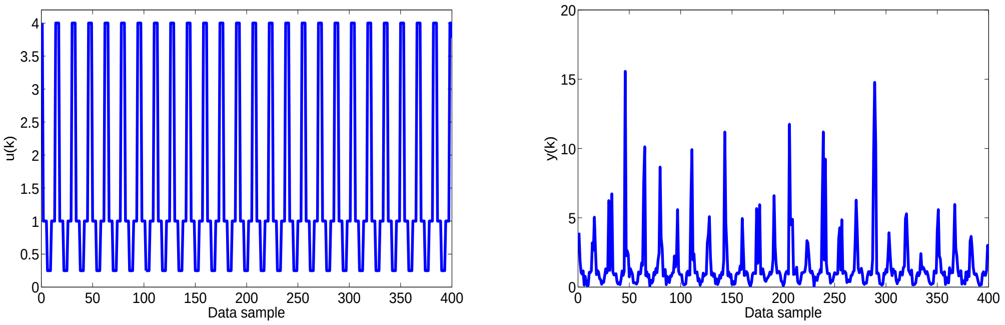

Figure 4: Input and Output Data for System Identification

range $v_{min} = -15$, $v_{max} = 15$ and $\delta v = 0.5$. The model is formulated as in eq. 1.

$$u(k) = 4.17e^{-7}v(k)^5 + 9.34e^{-6}v(k)^4 + 1.018e^{-4}v(k)^3 + 0.0028v(k)^2 + 0.08v(k) + 1.0045 \tag{1}$$

Subsequently, a linear model ($xL(k) = f(v(k))$) is formulated in an ARX (Auto-Regressive eXogenous) model,

$$xL(k+1) = 0.4201xL(k) + 0.0430v(k) \tag{2}$$

In Wiener block, another inverse nonlinear model is approximated as a function of $xL(k) = f^{-1}(y(k))$ which is estimated in B-spline curve function $B(y(k))$ with parameter N_i as basis function, the curve order ρ , and Cp_i as control points of the curve. The mathematical equation is written as follows,

$$xL(k) = \sum_{i=0}^{s} N_{i,\rho}((y(k))Cp_i \tag{3}$$

3. Finite Control Set - Model Predictive Control

Control signal in FCS-MPC utilizes a finite number of input states. In each sampling time, the input state that minimizes cost function is selected and implemented as the control action. Cost function (J) is evaluated for each input state. Then, the optimal control signal is the one with the least error in cost function optimization.

3.1. Optimal Finite Control Set (OFCS)

The basic algorithm of FCS is the a receding horizon approach with one-step ahead prediction and on-line optimization. The value of $y(k)$ for prediction uses the updated value from current measurements of output response. This updating procedure is closed-loop feedback mechanism in FCS method. The discrete model of an SISO system is defined as,

$$\begin{bmatrix} y_1(k+1) \\ y_2(k+1) \end{bmatrix} = A \begin{bmatrix} y_1(k) \\ y_2(k \end{bmatrix} + B \begin{bmatrix} u_1(k) \\ u_2(k) \end{bmatrix} \tag{4}$$

and the cost function to be solved as optimization problem is the squared error between reference and the output values:

$$J = \begin{bmatrix} r_1(k) - y_1(k+1) & r_2(k) - y_2(k+1) \end{bmatrix} \begin{bmatrix} r_1(k) - y_1(k+1) \\ r_2(k) - y_2(k+1) \end{bmatrix} \tag{5}$$

Furthermore, by defining $f(k)$ as,

$$\begin{bmatrix} f_1(k) \\ f_2(k) \end{bmatrix} = \begin{bmatrix} r_1(k) \\ r_2(k) \end{bmatrix} - A \begin{bmatrix} y_1(k) \\ y_2(k \end{bmatrix} \tag{6}$$

the objective function can be rewritten in the following notation:

$$J = \left(\begin{bmatrix} f_1(k) & f_2(k) \end{bmatrix} - B \begin{bmatrix} u_1(k) & u_2(k) \end{bmatrix} \right) x \left(\begin{bmatrix} f_1(k) \\ f_2(k) \end{bmatrix} - B \begin{bmatrix} u_1(k) \\ u_2(k) \end{bmatrix} \right) \tag{7}$$

Transforming the J function into a quadratic form yields to:

$$J = \begin{bmatrix} f_1(k) & f_2(k) \end{bmatrix} \begin{bmatrix} f_1(k) \\ f_2(k \end{bmatrix} - 2 \begin{bmatrix} u_1(k) & u_2(k) \end{bmatrix} B^T \begin{bmatrix} f_1(k) \\ f_2(k \end{bmatrix} + \begin{bmatrix} u_1(k) & u_2(k) \end{bmatrix} B^T B \begin{bmatrix} u_1(k) \\ u_2(k) \end{bmatrix} \tag{8}$$

Consequently, the system optimization can be solved as a least squares problem. The first derivative of J function from (8) in respect to control signal u is

$$\frac{\delta J}{\delta u_{12}(k)} = -2B^T \begin{bmatrix} f_1(k) \\ f_2(k) \end{bmatrix} + 2B^T B \begin{bmatrix} u_1(k) \\ u_2(k) \end{bmatrix} \tag{9}$$

where the minimum value of J is acquired when (9) is equal to 0. Control signal with the minimum value of J is written as follows,

$$\begin{bmatrix} u_1(k) \\ u_2(k) \end{bmatrix} = (B^T B)^{-1} B^T \begin{bmatrix} f_1(k) \\ f_2(k) \end{bmatrix} \tag{10}$$

Next, the selected control action is obtained by substituting (6) into (10). It leads to the function below:

$$\begin{bmatrix} u_1(k)^{opt} \\ u_2(k)^{opt} \end{bmatrix} = B^{-1} \left(\begin{bmatrix} r_1(k) \\ r_2(k) \end{bmatrix} - A \begin{bmatrix} y_1(k) \\ y_2(k) \end{bmatrix} \right) \tag{11}$$

This $u(k)$ value is the optimal solution of one-step ahead prediction algorithm in FCS.

The derived algorithms in all sections throughout this paper represent multi input-multi output system. In the structure of relative performance management of a virtualized software system in Figure 2, the input and output signal are formulated in the ratio value from the two virtual machines. Therefore, the identified dynamic model is a single input-single output function (SISO), as follows

$$y(k+1) = ay(k) + bu(k) \tag{12}$$

Furthermore, all corresponding equations for the controller design should be treated as a SISO system. For instance, the optimal control signal from (11) is simplified to,

$$u(k)^{opt} = b^{-1}(r(k) - ay(k)) \tag{13}$$

and the cost function for SISO system is:

$$J = (r(k) - y(k+1))^2 \tag{14}$$

Control law for Optimal FCS-MPC (OFCS) is sorted in the following steps:

(i) At sampling time k, find the reference value $r(k)$ and measure output $y(k)$.

(ii) Calculate the control solution for all the optimal control signal candidates v_n, $n = 1, 2, 3, \dots, N$ using (13), where N is the total number of control states.

(iii) Compute the objective function J based on the squares of error between the predicted value $y(k)$ and the reference value $r(k)$ using (14).

(iv) Find J_n, which has the smallest value and get its corresponding index n. The control signal with this index (v_n) is the optimal control solution to be implemented in the feedback control system at sampling time k.

(v) Repeat from step 1 for the next sampling time $(k+1)$.

3.2. Candidates of Control States

In this section, the effect of control states length is examined using Optimal FCS. The feedback control system is evaluated with two different length of input states. The available resources are assumed to be 100% for the total capacity and the minimum allocation for each virtual machine is 20%. By setting $\delta u = 1$ for each Cap_1 and Cap_2, the input signal ($u = \dfrac{Cap_1}{Cap_2}$) can be arranged in a sequence of 61 input operating points [13].

$$u = \left\{ \frac{20}{80}, \frac{21}{79}, \ldots, \frac{50}{50}, \ldots, \frac{79}{21}, \frac{80}{20} \right\} \tag{15}$$

For Hammerstein-Wiener feedback control structure, input signal is the intermediate variable v. The values of v are assigned as the set of possible control states in FCS control scheme. To transform the input u so that the operating points are equally spaced, the intermediate variable $v(k)$ values are set arbitrarily by $v_{min} = -15$, $v_{max} = 15$ where $\delta v = 0.5$. Thus, the number of v operating points is 61 with the sequence of $v = \{-15, -14.5, \ldots, 0, \ldots, 14.5, 15\}$. The sequence of v based on the selection of δv determines the length of input states. The smaller the δv value, the longer the operating input set.

The OFCS algorithm is implemented to control the system with reference value $r = 1$ and nominal workloads of 100 requests/second for both clients. The control scheme is in Hammerstein-Wiener structure which is designed based on the linear model in eq. 2. Then, using eq. 12, parameter a and b are determined, $a = 0.4201$, and $b = 0.0430$.

The values of $\delta v = 0.5$ for 61 points and $\delta v = 0.1$ for 301 points are chosen. It can be seen that the performance of the FCS control system with 301 control states is significantly better than the one with 61 states. The output responses are presented in Figures 5 and 6. The output response from the controlled system with a longer sequence of control states has a less steady state error and the change of control signal trajectory is more stable than the one with fewer control states. It can be analyzed that with the longer control states, the rate of change in control signal will be smoother because the system is provided with a wide range of control state candidates.

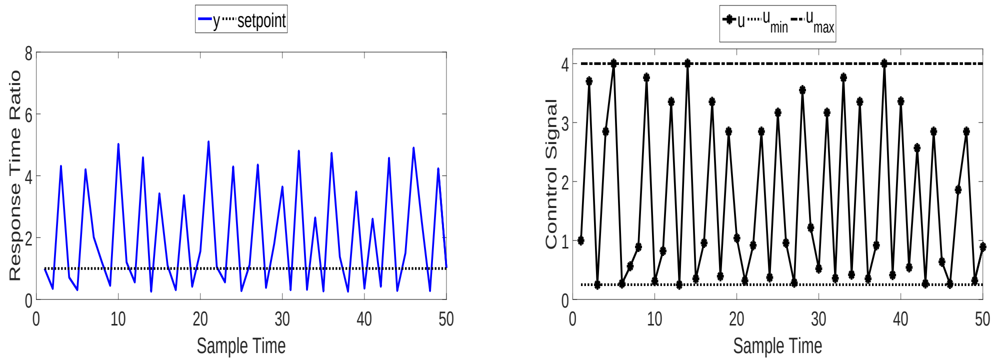

Figure 5: Output response and input signal u of OFCS-MPC for 61 control states of w

3.3. Constrained Finite Control Set (CFCS)

This section presents the procedure for finding the actual control signal $u(k)$ among the available candidates. The optimal control signal is not automatically equal to one of the control signal candidates. Therefore, constraint is deployed in the prediction of the OFCS control system to revise the searching procedure to calculate the actual control signal. The objective function for the constrained FCS-MPC

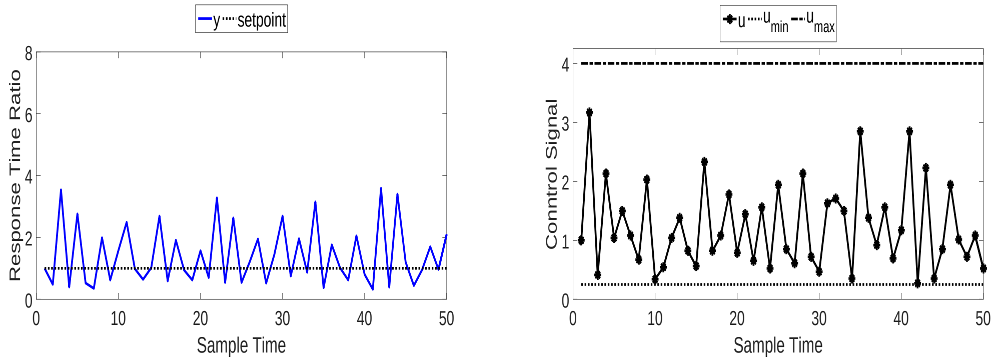

Figure 6: Output response and input signal u of OFCS-MPC for 301 control states of w

(CFCS) is converted to:

$$J = \left(\begin{bmatrix} u_1(k) \\ u_2(k) \end{bmatrix} - \begin{bmatrix} u_1(k)^{opt} \\ u_2(k)^{opt} \end{bmatrix} \right)^T (B^T B) \left(\begin{bmatrix} u_1(k) \\ u_2(k) \end{bmatrix} - \begin{bmatrix} u_1(k)^{opt} \\ u_2(k)^{opt} \end{bmatrix} \right) \tag{16}$$

This function is derived based on the value of the optimal control signal as in (11). Then, the objective function J can be presented in the formulation below:

$$J = B^2(u_1(k) - u_1(k)^{opt})^2 + B^2(u_2(k) - u_2(k)^{opt})^2 \tag{17}$$

It is sufficient to evaluate this function by using only:

$$J = (u_1(k) - u_1(k)^{opt})^2 + (u_2(k) - u_2(k)^{opt})^2 \tag{18}$$

Furthermore, to apply a constrained FCS-MPC method in relative performance management, the algorithm of CFCS should be adjusted in compatible formulations for a single input-single output function (see (12) and (13 for the output function and optimal control signal respectively). Likewise, the cost function from (18) becomes

$$J = (u(k) - u(k)^{opt})^2 \tag{19}$$

The control procedures of the constrained optimal finite control set are summarized as follows:

(i) At sampling time k, measure output $y(k)$ and get the reference value $r(k)$.

(ii) Calculate the optimal control signal $u(k)^{opt}$ using (13).

(iii) Compute the objective function J_n for all the control candidates ($n = 1, 2, \cdots, N$) using (19), where N is the total number of control states.

(iv) Find J_n, which has the smallest value, and get its corresponding index n. The control signal with this index (v_n) is the optimal control solution to be implemented in the feedback control system at sampling time k.

(v) Repeat from step 1 for the next sampling time $(k+1)$.

4. Evaluation of Optimal FCS and Constrained FCS

In this section, the performance of Constrained FCS-MPC and Optimal FCS-MPC are evaluated in an experiment where the reference value is $r = 1$. Feedback controller with an addition of pre-input and post-output nonlinear compensator is presented in Figure 7.

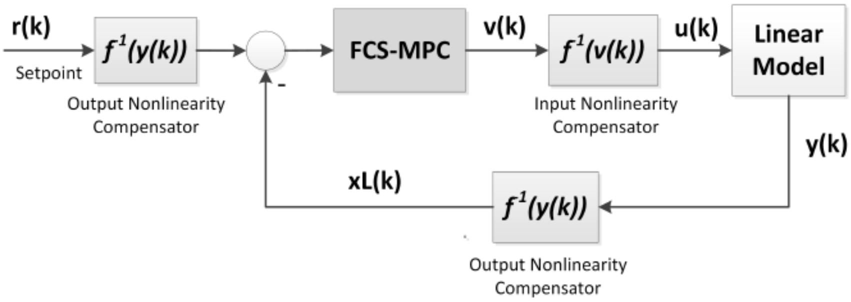
r(k)
Setpoint
$f^{1}(y(k))$
Output Nonlinearity Compensator
-
FCS-MPC
v(k)
$f^{1}(v(k))$
Input Nonlinearity Compensator
u(k)
Linear Model
y(k)
xL(k)
$f^{1}(y(k))$
Output Nonlinearity Compensator

Figure 7: Feedback control loop with nonlinear compensations

301 control states are used for this experiment. The disturbance for the control system is the workload changes of *client*$_1$ and *client*$_2$. During the first 40 samples, workloads for all clients are 100 requests/second, and then they are increased simultaneously to 200 requests/second. Experimental results for OFCS and CFCS are given in Figures 8 and 9. In addition, the statistics for output response are presented in Table 1.

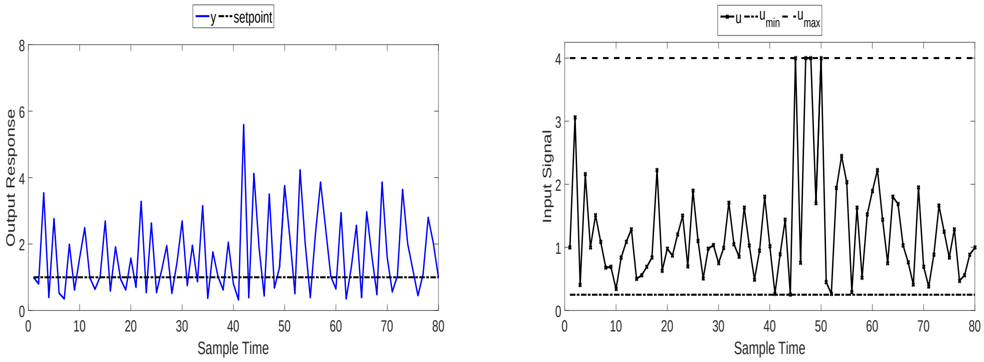

Figure 8: Output response and control signal of OFCS

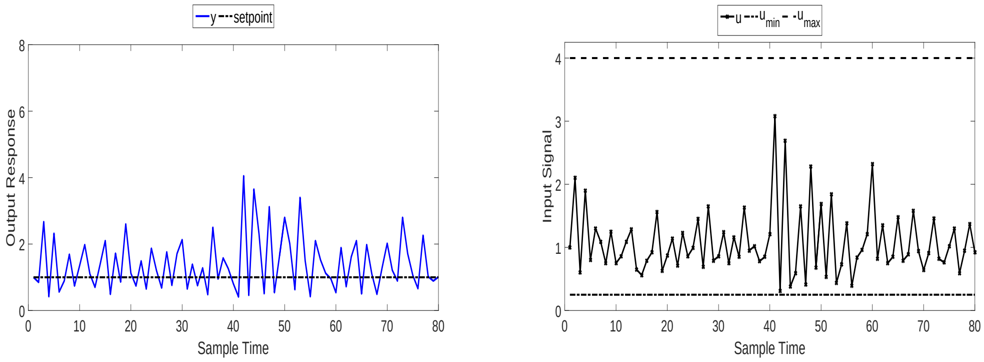

Figure 9: Output response and control signal of CFCS

Table 1: Output response statistics of OFCS and CFCS

Experiment	MSE	Minimum value	Maximum value
OFCS	1.8095	0.32	5.59
CFCS	0.8217	0.41	4.05

It can be inferred from the results that the improved procedure of control state searching by adding constraint in the CFCS algorithm makes the movement of control signals less aggressive than in OFCS. Likewise, the output response of CFCS has a smaller steady state error compare to the OFCS response. The MSE value of OFCS is 1.8095, which is two times bigger than the CFCS. This error caused by the oscillations of output response in OFCS with a higher magnitude of overshoots compare to the CFCS response.

5. CONCLUSIONS

FCS-MPC has been studied and implemented for dynamic resource provisioning in one type of shared-resources environment. Based on the experimental results, it can be concluded that a wider range of states selection provides a smoother dynamic of control signal. Moreover, constrained FCS-MPC demonstrates a better performance where its output response has smaller steady state error than the optimal FCS-MPC.

References

[1] Patikirikorala T, Wang L, Colman A and Han J 2012 *Control Engineering Practice* **20** 49–61

[2] Padala P, Shin K G, Zhu X, Uysal M, Wang Z, Singhal S, Merchant A and Salem K 2007 Adaptive control of virtualized resources in utility computing environments *ACM SIGOPS Operating Systems Review* vol 41 (ACM) pp 289–302

[3] Vargas R, Ammann U and Rodriguez J 2009 *IEEE Transactions on Power Electronics* **24** 894–902

[4] Quevedo D E, Aguilera R P, Perez M A, Cortés P and Lizana R 2012 *IEEE Transactions on Power Electronics* **27** 3128–3136

[5] Cortés P, Kazmierkowski M P, Kennel R M, Quevedo D E and Rodríguez J 2008 *IEEE Transactions on industrial electronics* **55** 4312–4324

[6] Kouro S, Cortés P, Vargas R, Ammann U and Rodríguez J 2009 *IEEE Transactions on Industrial Electronics* **56** 1826–1838

[7] Rodriguez J, Pontt J, Silva C A, Correa P, Lezana P, Cortés P and Ammann U 2007 *IEEE Transactions on Industrial Electronics* **54** 495–503

[8] Vargas R, Cortés P, Ammann U, Rodríguez J and Pontt J 2007 *IEEE Transactions on Industrial Electronics* **54** 2697–2705

[9] Morel F, Lin-Shi X, Rétif J M, Allard B and Buttay C 2009 *IEEE Transactions on Industrial Electronics* **56** 2715–2728

[10] Chai S and Wang L 2012 Finite control set model predictive control of 2lvsi-pmsm using interpolated switching states *IECON 2012-38th Annual Conference on IEEE Industrial Electronics Society* (IEEE) pp 1799–1804

[11] Preindl M and Bolognani S 2013 *IEEE Transactions on Power Electronics* **28** 1007–1015

[12] Aryani D, Wang L and Patikirikorala T 2017 *IFAC-PapersOnLine* **50** 7795–7800

[13] Aryani D, Wang L and Patikirikorala T 2016 *International Journal of Systems Science* **48** 1146–1161