Distributed Model Predictive Control of Load Frequency of Power Network

Christian Fogelberg[1] and Toru Namerikawa[1]

[1]Department of System Design Engineering, Keio University, Kanagawa, Japan
(Tel: +81-45-563-1151; E-mail: fogelberg@nl.sd.keio.ac.jp, namerikawa@sd.keio.ac.jp)

Abstract: This paper deals with control of smart grid by using distributed Model Predictive Control to control the load frequency of the power network. The control objective is to minimize the cost function while still keeping the frequency deviation constrained to a safe level. Some different control structures are proposed and control law and constraints are derived. Lastly we show the effectiveness of the different controllers by using a large scale power network simulation.

Keywords: Model predictive control, power network, frequency control.

1. INTRODUCTION

In recent years there has been an increase of interest in smart grid concept, depicted in Fig. 1, to adapt the power grid to improve the reliability, efficiency and economics of the electricity production and distribution. One of the generator side problem in this is to meet the power requirement while not wasting unnecessary power, thus keeping the cost down, which must be done while the frequency is kept in a suitable range that will not damage any equipment connected to the power grid.

Therefore, in this paper, a distributed Model Predictive Control (MPC) approach to control each power plant output frequency as to not deviate from the predefined output is proposed. The advantage of MPC is that it generalizes directly to plants being MIMO, which can be non-square, and takes process constraints into account, which eliminates the possibility of variables exceeding their predetermined limit.

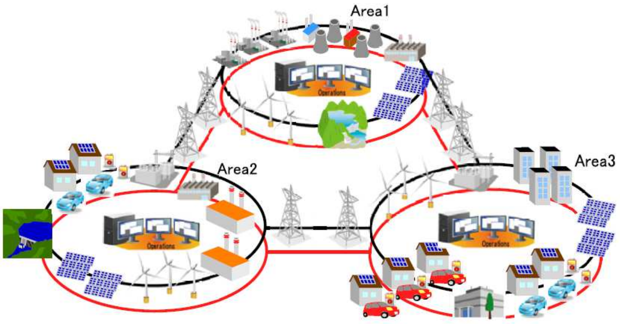

Fig. 1 Smart grid

2. PROBLEM FORMULATION

We consider an electric power network that consist of $N(\geq 2)$ in series connected subsystems.

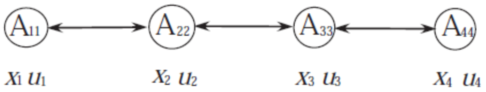

Fig. 2 Graphical representation

The normal state-space formulation for the whole system, which is assumed to be

$$x(k+1) = Ax(k) + Bu(k)$$
$$y(k) = Cx(k) \tag{1}$$

can be separated into smaller subsystem equation. Assuming that each subsystem can use information of neighboring subsystem, then the i-th subsystem is given by

$$x_i(k+1) = \sum_{j=1}^{N} A_{ij}x_j(k) + B_iu_i(k) \quad i = 1,\dots,N \tag{2}$$

where k is the time, $x_i(k) \in \mathbb{R}^{n_{xi}}$ is the states of system i, $A_{ij} \in \mathbb{R}^{n_{xj}\times n_{xj}}$, if $j = i$ then $A_{ii} \in \mathbb{R}^{n_{xi}\times n_{xi}}$, $u_i(k) \in \mathbb{R}^{n_{ui}}$ is the control signal of system i and $B = \mathrm{diag}[B_1,\dots,B_N] \in \mathbb{R}^{n_u\times n_u}$. The set $\mathcal{N}_i$ includes the subsystems that subsystem i is connected to, so when the i-th subsystem is connected to the j-th subsystem it can be written as $j \in \mathcal{N}_i$. If they are not connected it is expressed as

$$A_{ij} = 0 \quad \text{if } j \notin \mathcal{N}_i \tag{3}$$

which means that the system equation (2) also can be expressed as

$$x_i(k+1) = A_{ii}x_i(k) + \sum_{j\in\mathcal{N}_i} A_{ij}x_j(k) + B_iu_i(k). \tag{4}$$

The centralized controller uses the information of the whole system, while the decentralized controllers only uses the information of their own subsystem. The distributed controller uses its own information and the neighboring subsystems information.

The output of the system that is to be controlled is the deviation of the frequency from the normal frequency of the whole system.

3. MODEL PREDICTIVE CONTROL

To specify how many steps ahead the controller should take into account, the prediction horizon and control horizon is defined.

Definition 1: We denote the prediction horizon as N_p and the control horizon as N_c. The prediction horizon defines how many steps ahead the controller should predict the states. The control horizon dictates the number of steps the controller should try to complete the control objective in.

Also, an assumption about the control horizon is needed such that it does not cause problems in the calculations.

Assumption 1: It is assumed that the control horizon N_c is chosen to be less than or equal to the prediction horizon N_p

$$N_c \leq N_p$$

since it is not possible to predict a control trajectory without having predicted the states at that time instant.

Based on the state-space model Eq. (1), the future state variables are calculated sequentially using the future control parameters. By substituting the previous row into the next one, we can get a predicted state estimate at a certain time with calculations only depending on the current states $x(k)$ and the control input u.

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) \\ x(k+2) &= Ax(k+1) + Bu(k+1) \\ &= A^2x(k) + ABu(k) + Bu(k+1) \\ &\vdots \\ x(k+N_p) &= A^{N_p}x(k) + A^{N_p-1}Bu(k) + \\ &+ A^{N_p-2}Bu(k+1) + \cdots + \\ &+ A^{N_p-N_c}Bu(k+N_c-1) \end{aligned} \tag{5}$$

From the above equation and the original state-space model Eq. (1) we can get the predicted output variables, by substitution, so all predicted variables are formulated in terms of current state variable information $x(k)$ and the future control movement $u(k+t)$, where $t = 0, 1, \ldots, N_c - 1$.

$$\begin{aligned} y(k+1) &= CAx(k) + CBu(k) \\ y(k+2) &= CA^2x(k) + CABu(k) + CBu(k+1) \\ y(k+3) &= CA^3x(k) + CA^2Bu(k) + \\ &+ CABu(k+1) + CBu(k+2) \\ &\vdots \\ y(k+N_p) &= CA^{N_p}x(k) + CA^{N_p-1}Bu(k) + \\ &+ CA^{N_p-2}Bu(k+1) + \cdots + \\ &+ CA^{N_p-N_c}Bu(k+N_c-1) \end{aligned} \tag{6}$$

Rearranging these into matrices thus gives the system as

$$\begin{aligned} \underset{\rightarrow}{x} &= P_x x(k) + H_x \underset{\rightarrow}{u} \\ \underset{\rightarrow}{y} &= P_y x(k) + H_y \underset{\rightarrow}{u} \end{aligned} \tag{7}$$

where

$$\begin{aligned} &\underset{\rightarrow}{x} = \begin{bmatrix} x(k+1) \\ x(k+2) \\ x(k+3) \\ \vdots \\ x(k+N_p) \end{bmatrix}, \underset{\rightarrow}{u} = \begin{bmatrix} u(k) \\ u(k+1) \\ u(k+2) \\ \vdots \\ u(k+N_c-1) \end{bmatrix}, \\ &P_x = \begin{bmatrix} A \\ A^2 \\ A^3 \\ \vdots \\ A^{N_p} \end{bmatrix}, P_y = \begin{bmatrix} CA \\ CA^2 \\ CA^3 \\ \vdots \\ CA^{N_p} \end{bmatrix}, \\ &H_x = \begin{bmatrix} B & 0 & \ldots & 0 \\ AB & B & \ldots & 0 \\ A^2B & AB & \ldots & 0 \\ \vdots & \vdots & \ddots & 0 \\ A^{N_p-1}B & A^{N_p-2}B & \ldots & A^{N_p-N_c}B \end{bmatrix}, \end{aligned} \tag{8}$$

$$H_y = \begin{bmatrix} CB & 0 & \ldots & 0 & \\ CAB & CB & \ldots & 0 & \\ CA^2B & CAB & \ldots & 0 & \\ \vdots & \vdots & \vdots & \ddots & 0 \\ CA^{N_p-1}B & CA^{N_p-2}B & \ldots & CB & \end{bmatrix}.$$

Using the cost function

$$J = \underset{\rightarrow}{x}^T Q \underset{\rightarrow}{x} + \underset{\rightarrow}{u}^T R \underset{\rightarrow}{u}, \tag{9}$$

where $Q \geq 0$ and $R > 0$ is the weighting matrices, the minimization in regards to $\underset{\rightarrow}{u}$ using the prediction system from Eq. (7) becomes

$$\begin{aligned} \min_{\underset{\rightarrow}{u}} J = (P_x x(k) + H_x \underset{\rightarrow}{u})^T Q (P_x x(k) + H_x \underset{\rightarrow}{u}) + \\ + \underset{\rightarrow}{u}^T R \underset{\rightarrow}{u}, \end{aligned} \tag{10}$$

and from the minimization that the derivative should be zero, we get that

$$\frac{dJ}{d\underset{\rightarrow}{u}} = 0 \Rightarrow -(H_x^T Q H_x + R)\underset{\rightarrow}{u} = H_x^T Q P_x x(k), \tag{11}$$

from which it is given that the optimal control law is

$$\begin{aligned} \underset{\rightarrow}{u} &= -(H_x^T Q H_x + R)^{-1} H_x^T Q P_x x(k) \\ &= -Kx(k), \end{aligned} \tag{12}$$

where $K = (H_x^T Q H_x + R)^{-1} H_x^T Q P_x$. From Eq. (8) and Eq. (12) it can also be seen that K only depends on the system parameters, hence is a constant matrix that can be calculated offline.

Even though $\underset{\rightarrow}{u}$ contains the predicted control signal for N_c steps ahead, since the calculation is made in every sample only the first $u(k)$ is used, which is called Receding Horizon Control since the horizon is always moving away. This ensures that the most recent data is used, which gives a more precise control calculation and a faster response to new changes that might occur, depicted in Fig. 3.

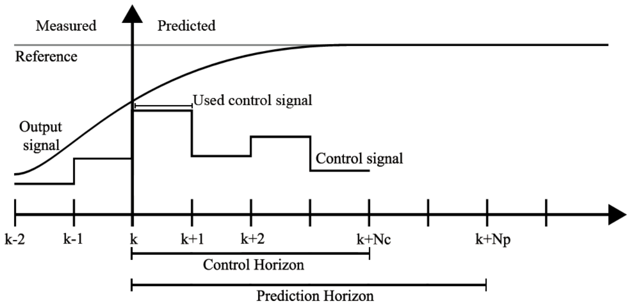

Fig. 3 Receding Horizon Control

So from the complete set of predicted control signals

$$\underset{\rightarrow}{u} = \begin{bmatrix} u(k) \\ u(k+1) \\ u(k+2) \\ \vdots \\ u(k+N_c) \end{bmatrix} \tag{13}$$

we only want the most relevant control signal for the next control correction

$$u(k) = \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_{n_u} \end{bmatrix}, \tag{14}$$

n_u being the number of control signals to the plant.

Since we only take the first element of $\underset{\rightarrow}{u}$, we can write the control signal as

$$u(k) = e_I \underset{\rightarrow}{u} = -e_I K x(k), \tag{15}$$

where $e_I = [\underbrace{1\ 1\ \dots\ 1}_{n_u}\ \ 0\ \ 0\ \ \dots\ \ 0]$ eliminates all elements in K except for the first control sequence. Thus the state equation can be written as

$$\begin{aligned} x(k+1) &= Ax(k) - Be_I K x(k) \\ &= (A - Be_I K)x(k). \end{aligned} \tag{16}$$

4. CONSTRAINTS

The Quadratic Programming algorithm used to recalculate the control signal in case of constraints, optimizes the problem on the form

$$\min_u (f^T u + \frac{1}{2} u^T H u) \tag{17}$$

under the constraints such that

$$A_{QP} u \leq b, \tag{18}$$

where in our case u are the control signal $\underset{\rightarrow}{u}$, and H and f is from the optimal control law in Eq. (12)

$$H = H_x^T Q H_x + R, \quad f = H_x^T Q P_x x(k), \tag{19}$$

where f depends on the current state value, thus are time-varying. The constraints are formulated into A_{QP}, which is a matrix of linear constraint coefficients, and b, which is a time-varying vector. Constraints on the control signal, such as

$$-0.5 \leq \underset{\rightarrow}{u} \leq 0.5 \tag{20}$$

would be rearranged into

$$\begin{bmatrix} -1 \\ 1 \end{bmatrix} \underset{\rightarrow}{u} \leq \begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix}. \tag{21}$$

Constraint on the output signal, such as

$$-0.2 \leq \underset{\rightarrow}{y} \leq 0.2 \tag{22}$$

needs to be rewritten in terms of u, which by using Eq. (7)

$$\underset{\rightarrow}{y} = P_y x(k) + H_y \underset{\rightarrow}{u} \tag{23}$$

becomes

$$-0.2 \leq P_y x(k) + H_y \underset{\rightarrow}{u} \leq 0.2. \tag{24}$$

Rearranging Eq. (24) gives the boundaries as

$$\begin{bmatrix} -H_y \\ H_y \end{bmatrix} \underset{\rightarrow}{u} \leq \begin{bmatrix} 0.2 + P_y x(k) \\ 0.2 - P_y x(k) \end{bmatrix}. \tag{25}$$

Constraints on the states would be rewritten simmilarly as constraints on output signal, but using $\underset{\rightarrow}{x}$ from Eq. (7) instead of $\underset{\rightarrow}{y}$.

By combining the constrains on the control signal in Eq. (21) and the constraints on the output signal in Eq. (25), we get the complete constraint matrix Eq. (18) as

$$\underbrace{\begin{bmatrix} -1 \\ 1 \\ -H_y \\ H_y \end{bmatrix}}_{A_{QP}} \underset{\rightarrow}{u} \leq \underbrace{\begin{bmatrix} 0.5 \\ 0.5 \\ 0.2 + P_y x(k) \\ 0.2 - P_y x(k) \end{bmatrix}}_{b}. \tag{26}$$

From this we can use the Quadratic Programming algorithm to get the new optimized control signal u from Eq. (17) under the constraints from Eq. (26).

5. KALMAN FILTER

Since the plants only output is the frequency deviation, and the above described MPC uses all the states, a state observer is needed.

For this a state estimate with Kalman filter is implemented as

$$\begin{aligned} \hat{x}(k+1) = &A\hat{x}(k) + Bu(k) + \\ &+ K_f(y(k) - C\hat{x}(k)), \end{aligned} \tag{27}$$

where K_f is given by

$$K_f = (APC^T)(CPC^T + R_f)^{-1}, \tag{28}$$

in which R_f is the weights and P is the symmetric positive semidefinite solution of the algebraic Riccati equation

$$AP + PA^T - (PC^T)R_f^{-1}(PC^T)^T = 0. \tag{29}$$

The decentralized controller has one Kalman filter for each subsystem, the centralized controller uses these to build up a complete estimate of the system, and each of the distributed controller uses the ones it needs for the region it controls.

6. CONTROLLER IMPLEMENTATION

The MPC controller was implemented both as separate decentralized controllers, as a centralized controller, and as distributed controllers.

The centralized controller uses the state-space shown in Eq. (1), while the decentralized controller instead uses the matrices A_{ii}, B_{ii} and C_{ii}.

The distributed control model were split into sections that encompass data that each subsystem have access to. The matrices for each distributed control section can then be taken from the complete power network system model as shown below on the A matrix Eq. (30).

$$A = \begin{bmatrix} A11 & A12 & 0 & 0 \\ & & & \\ A21 & A22 & A23 & 0 \\ A_{D1} & & & \\ 0 & A32 & A33 & A34 \\ A_{D2} & & & \\ 0 & 0 & A43 & A44 \end{bmatrix} \tag{30}$$
$$\qquad\qquad A_{D3} \qquad A_{D4}$$

Thus the state-space for the distributed controller uses the matrices A_{Di}, B_{Di} and C_{Di}.

The distributed controller implementation assumes that the tie-line power flow deviation ΔP_{tie_i} is known. From this and the plants own frequency deviation Δf_i, the connected subsystems frequency deviation Δf_j can be calculated if it is only connected to one other subsystem, as shown in (31), and then all states of the connected subsystem are estimated with a Kalman filter.

$$\Delta f_j = \frac{P_{tie_i}}{T_{ij}} - \Delta f_i \tag{31}$$

This method works well when connected to one other subsystem, like A_{D1} and A_{D4}. But the two other, A_{D2} and A_{D3}, has connections to two other subsystems, which means this method will not work since the contribution from each connected subsystem can not be assumed to be equal.

Instead the distributed controller implementation uses a slightly time delayed value of the real value of the connected subsystems frequency deviation, as if the power plants shares its information with the other plants over for example an internet connection. As long as the time delay is not to long, or some big changes happens to the connected subsystem, the calculated control signal are accurate enough to give a good result.

In cases with long distances or slow information transfer, where the time delay might become to great, it would be reasonable to time stamp the information when sending it, so that the receiving controller can check if it is relevant. If the information is to old, it can instead use a previous calculated control signal from $\underrightarrow{u}$ that used relevant information, or ignore the connected subsystem altogether and calculate a decentralized control signal instead.

7. SYSTEM SETUP

The setup shown in Fig. 4 shows four subsystems connected into a system, and the following system equations are acquired from it. The B, C and A_{ij} matrices are only given for the first subsystem, the others are similarly constructed. The states x are the tie-line power flow deviation ΔP_{tie_i}, frequency deviation Δf_i, output of the gas turbine generator ΔP_{g_i}, governor input of the gas turbine generator Δx_{g_i}, output of Battery Energy Storage System ΔP_{E_i}, output of thermal system ΔP_{H_i} and the demand U_{AR_i}.

Area 1 is set up with all generators present. Area 2 only has the battery system, thermal system and wind system. Area 3 has gas, thermal and wind system. Area 4 has gas, battery system and wind system. Since wind power is a non-controllable generator source it is not included in the system model, but instead is modeled as a added noise source in the simulation model.

Parameters used can be seen in Table 1 and below.

$$x_1 = \begin{bmatrix} \Delta P_{tie_i} \\ \Delta f_i \\ \Delta P_{g_i} \\ \Delta x_{g_i} \\ \Delta P_{E_i} \\ \Delta P_{H_i} \\ U_{AR_i} \end{bmatrix}, \; x_2 = \begin{bmatrix} \Delta P_{tie_i} \\ \Delta f_i \\ \Delta P_{E_i} \\ \Delta P_{H_i} \\ U_{AR_i} \end{bmatrix}, \; x_3 = \begin{bmatrix} \Delta P_{tie_i} \\ \Delta f_i \\ \Delta P_{g_i} \\ \Delta x_{g_i} \\ \Delta P_{H_i} \\ U_{AR_i} \end{bmatrix},$$

$$x_4 = \begin{bmatrix} \Delta P_{tie_i} \\ \Delta f_i \\ \Delta P_{g_i} \\ \Delta x_{g_i} \\ \Delta P_{E_i} \\ U_{AR_i} \end{bmatrix}, \; A_{12} = \begin{bmatrix} 0 & T & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \; A_{11} =$$

$$\begin{bmatrix} 0 & -T & 0 & 0 & 0 & 0 & 0 \\ \frac{1}{M_i} & -\frac{D}{M_i} & \frac{1}{M_i} & 0 & \frac{1}{M_i} & -\frac{1}{M_i} & 0 \\ 0 & 0 & -\frac{1}{T_{di}} & \frac{1}{T_{di}} & 0 & 0 & 0 \\ 0 & -\frac{1}{T_{gi}R_{gi}} & 0 & -\frac{1}{T_{gi}} & 0 & 0 & \frac{a_g}{T_g} \\ 0 & 0 & 0 & 0 & -\frac{1}{T_{Ei}} & 0 & \frac{a_E}{T_E} \\ 0 & 0 & 0 & 0 & 0 & -\frac{1}{T_{Hi}} & \frac{a_H}{T_H} \\ K_i & -B_iK_i & 0 & 0 & 0 & 0 & 0 \end{bmatrix},$$

$$B_{11} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ \frac{1}{T_{gi}} & 0 & 0 \\ 0 & \frac{1}{T_{Ei}} & 0 \\ 0 & 0 & \frac{1}{T_{Hi}} \\ 0 & 0 & 0 \end{bmatrix}, \tag{32}$$

$$C_{11} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

The off-diagonal matrices of B and C are zero.

The control weights are set to

$$R = [\overbrace{8 \;\; 0.83 \;\; 12}^{A1} \; \overbrace{0.83 \;\; 8}^{A2} \; \overbrace{10 \;\; 1}^{A3} \; \overbrace{8 \;\; 0.83}^{A4}] * I, \tag{33}$$

$$Q = I \tag{34}$$

The implementations is set to fulfill the constraints as

$$-0.2 \leq y \leq 0.2 \qquad (35)$$

Table 1 Power network parameters

Parameter	Symbol	Value
Inertia constant [puMW · s/Hz]	M	0.2
Damping constant [puMW/Hz]	D	0.26
Governor time constant [s]	T_g	0.2
Gas turbine constant [s]	T_d	5.0
BESS time constant [s]	T_E	0.2
Thermal system time constant [s]	T_H	4.5
Gas turbine capacity [p.u.]	a_G	0.8
BESS capacity [p.u.]	a_E	0.15
Thermal system capacity [p.u.]	a_H	0.5
Regulation constant [Hz/puMW]	R	2.5
Synchronizing coefficient [puMW]	T	0.50
Sampling time [s]	T_s	0.1
Prediction horizon	N_p	60
Control horizon	N_c	10

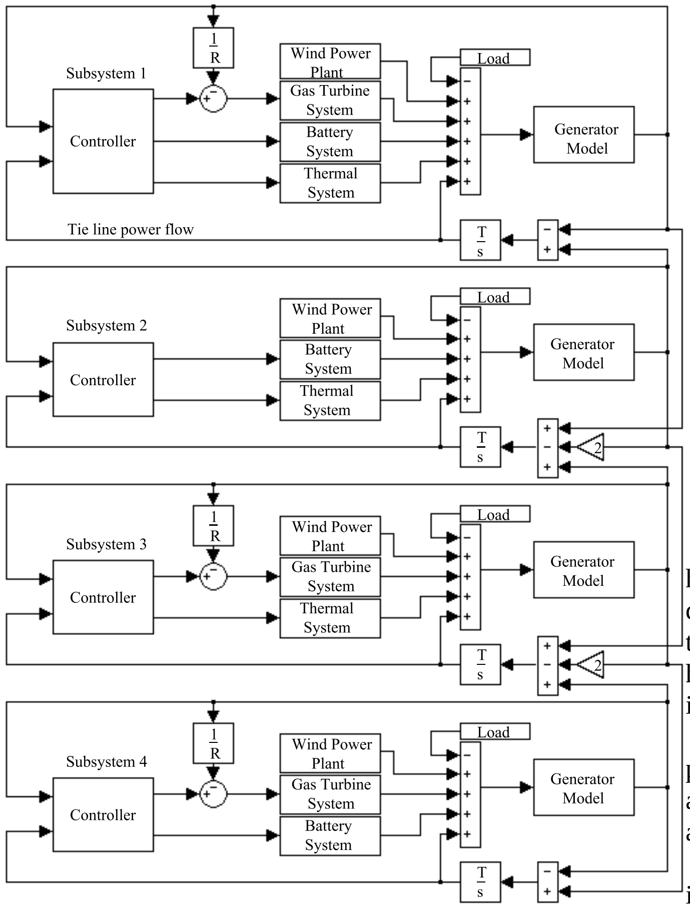

Fig. 4 Model of power network

8. RESULTS

The results shows the response to a load frequency change of 0.1 Hz at the time 0.1s.

The results from different settings on the prediction horizon can be seen in Fig. 5 - 6.

Result from the different controller setups can be seen in Fig. 7 - 8.

The effect of the difference between the real value and the delayed value to the distributed controller is shown in Fig. 9 - 8.

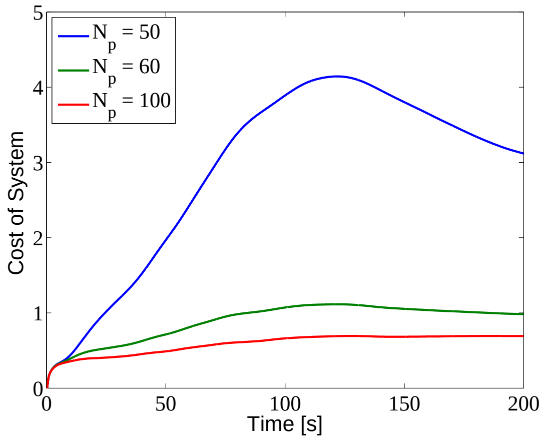

Fig. 5 N_c effect on centralized controller cost

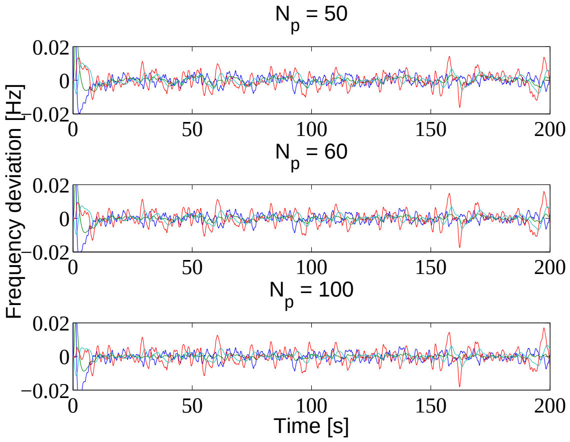

Fig. 6 N_c effect on centralized controller frequency

Fig. 5 - 6 shows that although an increased prediction horizon N_p has almost no visible impact on the frequency deviation, it has a large impact on the system cost to a certain point. The trade-off being that the longer prediction horizon increases of size of the equation matrices, which in turn increases the calculation time.

Fig. 7 - 8 shows that the distributed controller as expected is better than the decentralized one and not as good as the centralized one, both in the aspect of system cost and frequency deviation.

Fig. 9 - 8 shows that the delayed signal only slightly increases the cost while the frequency deviation stays almost the same.

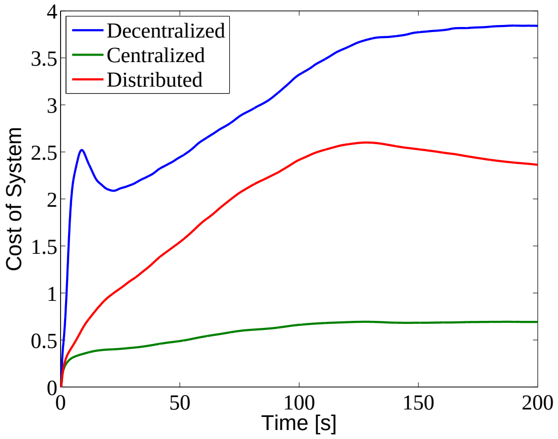

Fig. 7 MPC cost

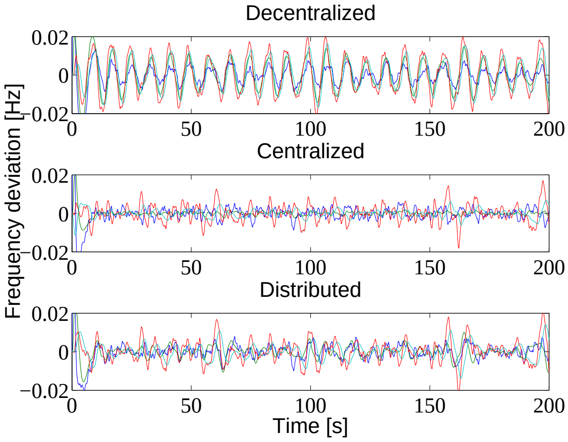

Fig. 8 MPC frequency deviation

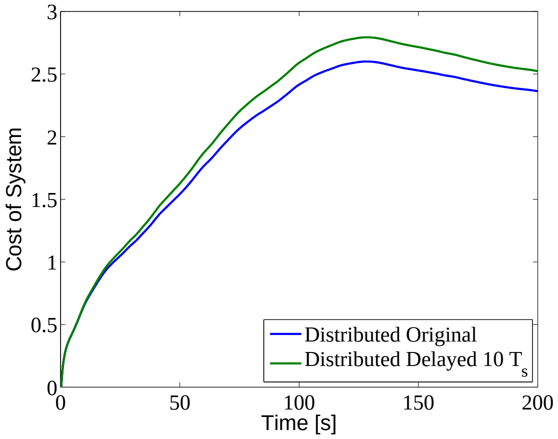

Fig. 9 Delay effect on distributed controller cost

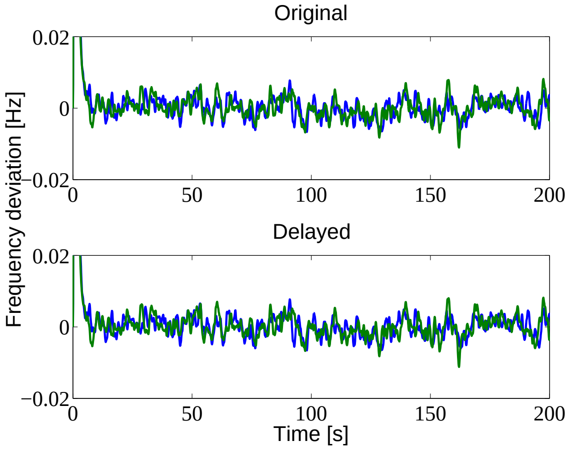

Fig. 10 Delay effect on distributed frequency deviation

9. CONCLUSION

In this paper we applied three different controllers on the smart grid power network by using Model Predictive Control to control the load frequency of the system. The control law and constraints were derived from the original system formulation, and Kalman filters were implemented as state estimator for the controllers. Lastly we implemented the controllers on a four subsystem simulation setup and showed the effectiveness of the different controllers on the power network.

It was shown that current power network system can be held inside of the normal constraint range of ±0.2 Hz with the proposed control method.

It was also shown that the implemented method of distributed controller using a delayed signal value to calculate the control signal gave a very good result, not too far from the original value. This shows that the distributed controller can give good results even if delays occur in the power network.

REFERENCES

[1] P. Giselsson, A. Rantzer. ”Distributed Model Predictive Control with Suboptimality and Stability Guarantees”, 49th IEEE Conference on Decision and Control, 2010.

[2] J.A. Rossiter. ”Model-Based Predictive Control”, CRC PRESS, 2005.

[3] C. Schmid, L.T. Biegler. ”Quadratic programming methods for reduced hessian SQP”, Computers & Chemical Engineering. Vol. 18, Number 9, 1994, pp. 817-832.