Performance Improvement via Bagging Competitive Associative Nets for Multiobjective Robust Controller Using Difference Signals

Weicheng Huang, Shuichi Kurogi, and Takeshi Nishida

Kyushu Institute of Technology, Tobata, Kitakyushu, Fukuoka 804-8550, Japan
ko@kurolab2.cntl.kyutech.ac.jp,
{kuro,nishida}@cntl.kyutech.ac.jp
http://kurolab2.cntl.kyutech.ac.jp

Abstract. So far, we have shown that, using difference signals of a plant to be controlled, a single CAN2 (competitive associative net) is capable of leaning piecewise Jacobian matrices of nonlinear dynamics of the plant. Here, the CAN2 is an artificial neural net for learning efficient piecewise linear approximation of nonlinear function. Furthermore, a multiobjective robust controller is obtained by means of combining the GPC (generalized predictive controller) and a switching scheme of multiple CAN2s to cope with plant parameter change and control objective change. This paper focuses on an improvement of control performance by means of replacing single CAN2 by bagging CAN2. We analyze to show the effectiveness of the present method via numerical experiments of a crane system.

Keywords: Multiobjective robust control, Switching of multiple bagging CAN2s, Difference signals, Generalized predictive control, Jacobian matrix of Nonlinear plant.

1 Introduction

So far, we have constructed a robust controller which uses multiple CAN2s (competitive associative nets) and difference signals of the plant to be controlled [1, 2]. Here, the CAN2 is an artificial neural net introduced for learning efficient piecewise linear approximation of nonlinear function by means of competitive and associative schemes [4–6]. Thus, a CAN2 is capable of leaning piecewise Jacobian matrices of nonlinear dynamics of a plant by means of using difference signals of the plant for the input to the CAN2, In [1], we have constructed a robust controller using multiple CAN2s which have learned the differential dynamics of the plant for several parameter values. In [2], we have focused on a multiobjective robust control, where we consider two conflicting control specifications for a crane system: one is to reduce settling time with allowable overshoot and the other is to reduce overshoot with allowable settling time. Our method provides a controller to flexibly cope with those specifications by means of switching two sets of CAN2s obtained through several control and learning iterations. From the point of view of multiobjective control, there are a number of research studies [8]. However, the control of the crane itself is neither so easy nor clarified so much for such methods to be applied.

M. Lee et al. (Eds.): ICONIP 2013, Part I, LNCS 8226, pp. 319–327, 2013.

In this paper, we try to improve the control performance by means of using bagging CAN2s. Here, the bagging (bootstrap aggregation) scheme is expected to reduce the variance and the overfitting of the prediction by a single learning machine [9]. Thus, we expect that bagging CAN2s provide more stable control performance than single CAN2s in the present application. In the next section, we show a formulation of the present method. In Sect. 3, we examine the effectiveness of the method through numerical experiments applied to a nonlinear crane system involving changeable parameter values, such as rope length and load weight.

2 Multiobjective Robust Controller Using Difference Signals and Bagging CAN2s

We formulate the controller using difference signals and multiple bagging CAN2s to cope with parameter change.

2.1 Plant Model Using Difference Signals

Suppose a plant to be controlled at a discrete time $j = 1, 2, \cdots$ has the input $u_j^{[\mathrm{p}]}$ and the output $y_j^{[\mathrm{p}]}$. Here, the superscript "[p]" indicates the variable related to the plant for distinguishing the position of the load, (x, y), shown below. Furthermore, we suppose the dynamics of the plant is given by

$$y_j^{[\mathrm{p}]} = f(\boldsymbol{x}_j^{[\mathrm{p}]}) + d_j^{[\mathrm{p}]} \ , \tag{1}$$

where $f(\cdot)$ is a nonlinear function which may change slowly in time and $d_j^{[\mathrm{p}]}$ represents zero-mean noise with the variance σ_d^2. The input vector $\boldsymbol{x}_j^{[\mathrm{p}]}$ consists of the input and output sequences of the plant as $\boldsymbol{x}_j^{[\mathrm{p}]} \triangleq \left(y_{j-1}^{[\mathrm{p}]}, \cdots, y_{j-k_y}^{[\mathrm{p}]}, u_{j-1}^{[\mathrm{p}]}, \cdots, u_{j-k_u}^{[\mathrm{p}]}\right)^T$, where k_y and k_u are the numbers of the elements, and the dimension of $\boldsymbol{x}_j^{[\mathrm{p}]}$ is given by $k = k_y + k_u$. Then, for the difference signals $\Delta y_j^{[\mathrm{p}]} \triangleq y_j^{[\mathrm{p}]} - y_{j-1}^{[\mathrm{p}]}$, $\Delta u_j^{[\mathrm{p}]} \triangleq u_j^{[\mathrm{p}]} - u_{j-1}^{[\mathrm{p}]}$, and $\Delta \boldsymbol{x}_j^{[\mathrm{p}]} \triangleq \boldsymbol{x}_j^{[\mathrm{p}]} - \boldsymbol{x}_{j-1}^{[\mathrm{p}]}$, we have the relationship $\Delta y_j^{[\mathrm{p}]} \simeq f_{\boldsymbol{x}} \Delta \boldsymbol{x}_j^{[\mathrm{p}]}$ for small $\|\Delta \boldsymbol{x}_j^{[\mathrm{p}]}\|$, where $f_{\boldsymbol{x}} = \partial f(\boldsymbol{x})/\partial \boldsymbol{x} \,\big|_{\boldsymbol{x}=\boldsymbol{x}_{j-1}^{[\mathrm{p}]}}$ indicates the Jacobian matrix (row vector). If $f_{\boldsymbol{x}}$ does not change for a while after the time j, then we can predict $\Delta y_{j+l}^{[\mathrm{p}]}$ by

$$\widehat{\Delta y}_{j+l}^{[\mathrm{p}]} = f_{\boldsymbol{x}} \widetilde{\Delta \boldsymbol{x}}_{j+l}^{[\mathrm{p}]} \tag{2}$$

for $l = 1, 2, \cdots$, recursively. Here, $\widetilde{\Delta \boldsymbol{x}}_{j+l}^{[\mathrm{p}]} = (\widetilde{\Delta y}_{j+l-1}^{[\mathrm{p}]}, \cdots, \widetilde{\Delta y}_{j+l-k_y}^{[\mathrm{p}]}, \widetilde{\Delta u}_{j+l-1}^{[\mathrm{p}]}, \cdots, \widetilde{\Delta u}_{j+l-k_u}^{[\mathrm{p}]})^T$, and the elements are given by

$$\widetilde{\Delta y}_{j+m}^{[\mathrm{p}]} = \begin{cases} \Delta y_{j+m}^{[\mathrm{p}]} \text{ for } m < 1 \\ \widehat{\Delta y}_{j+m}^{[\mathrm{p}]} \text{ for } m \geq 1 \end{cases} \quad \text{and} \quad \widetilde{\Delta u}_{j+m}^{[\mathrm{p}]} = \begin{cases} \Delta u_{j+m}^{[\mathrm{p}]} \text{ for } m < 0 \\ \widehat{\Delta u}_{j+m}^{[\mathrm{p}]} \text{ for } m \geq 0. \end{cases} \tag{3}$$

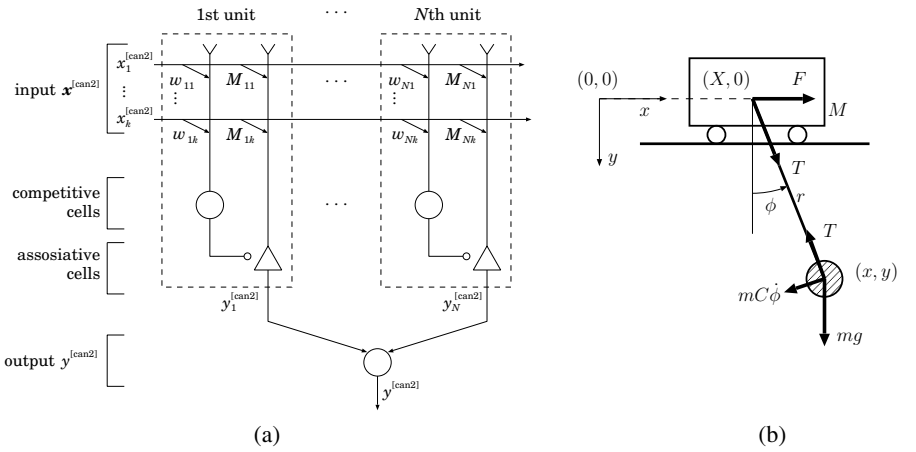

Fig. 1. Schematic diagram of (a) CAN2 and (b) overhead traveling crane system

Here, $\widehat{\Delta u}^{[p]}_{j+m}$ $(m \geq 0)$ is the predictive input (see Sect. 2.3). Then, we have the prediction of the plant output from the predictive difference signals as

$$\widehat{y}^{[p]}_{j+l} = y^{[p]}_j + \sum_{m=1}^{l} \widehat{\Delta y}^{[p]}_{j+m}. \tag{4}$$

2.2 Single and Bagging CAN2 for Learning and Identifying Nonlinear Plants

A single CAN2 has N units. The ith unit has a weight vector $\boldsymbol{w}_i \triangleq (w_{i1}, \cdots, w_{ik})^T \in \mathbb{R}^{k\times 1}$ and an associative matrix (row vector) $\boldsymbol{M}_i \triangleq (M_{i1}, \cdots, M_{ik}) \in \mathbb{R}^{1\times k}$ for $i \in I = \{1, 2, \cdots, N\}$ (see Fig. 1(a)). For a given dataset $D^{[n]} = \{(\Delta \boldsymbol{x}^{[p]}_j, \Delta y^{[p]}_j) \mid j = 1, 2, \cdots, n\}$ obtained from the plant to be controlled, we train a CAN2 by feeding the input and output of the CAN2 as $(\boldsymbol{x}^{[can2]}, y^{[can2]}) = (\Delta \boldsymbol{x}^{[p]}_j, \Delta y^{[p]}_j)$. We employ an efficient batch learning method shown in [7]. Then, for an input vector $\Delta \boldsymbol{x}^{[p]}$, the CAN2 after the learning predicts the output $\Delta y^{[p]} = f_{\boldsymbol{x}} \Delta \boldsymbol{x}^{[p]}$ by

$$\widehat{\Delta y}^{[p]} = \boldsymbol{M}_c \Delta \boldsymbol{x}^{[p]}, \tag{5}$$

where c denotes the index of the unit selected by

$$c = \underset{i\in I}{\operatorname{argmin}} \|\Delta \boldsymbol{x}^{[p]}_j - \boldsymbol{w}_i\|^2. \tag{6}$$

For a bagging CAN2, we generate a number of bags $D^{[n\alpha^\sharp, l]}$ for $l = 1, 2, \cdots, b$ by means of resampling with replacement from $D^{[n]}$, where b is the number of bags, $n\alpha$ denotes the number of elements in each bag and α is a constant which we call

bagsize ratio. Let $\mathrm{CAN2}^{[l]}$ be a single CAN2 for learning the lth bag $D^{[n\alpha^{\sharp},l]}$. After the learning for all bags, we execute the bagging prediction of $\Delta y^{[\mathrm{p}]} = f_{\boldsymbol{x}}\Delta \boldsymbol{x}^{[\mathrm{p}]}$ by

$$\widehat{\Delta y}^{[\mathrm{p}]} = \frac{1}{b}\sum_{l=1}^{b} \widehat{\Delta y}^{[\mathrm{p}][l]} = \boldsymbol{M}^{[\mathrm{bag}]}\Delta \boldsymbol{x}^{[\mathrm{p}]}. \tag{7}$$

Here, $\widehat{\Delta y}^{[\mathrm{p}][l]} = \boldsymbol{M}_c^{[l]}\Delta \boldsymbol{x}^{[\mathrm{p}]}$ and $\boldsymbol{M}_c^{[\mathrm{bag}]} = (1/b)\sum_{l=1}^{b}\boldsymbol{M}_c^{[l]}$, where $\boldsymbol{M}_c^{[l]}$ denotes $\boldsymbol{M}_c$ in (5) selected by (6) for $\mathrm{CAN2}^{[l]}$.

As we have explained in [2], the Jacobian matrix $f_{\boldsymbol{x}}$ is not the function of $\Delta\boldsymbol{x}$, but a Jacobian matrix $f_{\boldsymbol{z}} = \partial f/\partial \boldsymbol{z}$ for an enlarged differential vector $\Delta \boldsymbol{z}_j^{[\mathrm{p}]} = (\Delta y_{j-1}^{[\mathrm{p}]}, \cdots, \Delta y_{j-k_y'}^{[\mathrm{p}]}, \Delta u_{j-1}^{[\mathrm{p}]}, \cdots, \Delta u_{j-k'}^{[\mathrm{p}]})$ for $k_y' = k + k_y$ and $k_u' = k + k_u$ is considered to be a function of $\Delta\boldsymbol{z}$ when the plant parameter does not change for a while. This indicates that we can estimate Jacobian matrix if we observe the input and output for a certain duration of time of the plant. This conjecture is supposed to be applicable to many nonlinear plants. Thus, with $\Delta\boldsymbol{x}_j^{[\mathrm{p}]}$ in (6) replaced by an enlarged $\Delta\boldsymbol{z}_j^{[\mathrm{p}]}$, we can select an appropriate unit corresponding to the Jacobian matrix.

2.3 GPC Using Difference Signals

The GPC (Generalized Predictive Control) is an efficient method for obtaining the predictive input $\widehat{u}_j^{[\mathrm{p}]}$ which minimizes the following control performance index [10]:

$$J = \sum_{l=1}^{N_y}\left(r_{j+l}^{[\mathrm{p}]} - \widehat{y}_{j+l}^{[\mathrm{p}]}\right)^2 + \lambda_u \sum_{l=1}^{N_u}\left(\widehat{\Delta u}_{j+l-1}^{[\mathrm{p}]}\right)^2, \tag{8}$$

where $r_{j+l}^{[\mathrm{p}]}$ and $\widehat{y}_{j+l}^{[\mathrm{p}]}$ are desired and predictive output, respectively. The parameters N_y, N_u and λ_u are constants to be designed for the control performance. We obtain $\widehat{u}_j^{[\mathrm{p}]}$ by means of the GPC method as follows; the CAN2 at a discrete time j can predict $\Delta y_{j+l}^{[\mathrm{p}]}$ by (2) and then $\widehat{y}_{j+l}^{[\mathrm{p}]}$ by (4). Then, owing to the linearity of these equations, the above performance index is written as

$$J = \|\boldsymbol{r}^{[\mathrm{p}]} - \boldsymbol{G}\Delta\boldsymbol{u}^{[\mathrm{p}]} - \overline{\boldsymbol{y}}^{[\mathrm{p}]}\|^2 + \lambda_u\|\widehat{\Delta\boldsymbol{u}}\|^2 \tag{9}$$

where $\boldsymbol{r}^{[\mathrm{p}]} = \left(r_{j+1}^{[\mathrm{p}]}, \cdots, r_{j+N_y}^{[\mathrm{p}]}\right)^T$ and $\widehat{\Delta\boldsymbol{u}}^{[\mathrm{p}]} = \left(\widehat{\Delta u}_j^{[\mathrm{p}]}, \cdots, \widehat{\Delta u}_{j+N_u-1}^{[\mathrm{p}]}\right)^T$. Furthermore, $\overline{\boldsymbol{y}}^{[\mathrm{p}]} = \left(\overline{y}_{j+1}^{[\mathrm{p}]}, \cdots, \overline{y}_{j+N_y}^{[\mathrm{p}]}\right)^T$ and $\overline{y}_{j+l}^{[\mathrm{p}]}$ is the natural response $\widehat{y}_{j+l}^{[\mathrm{p}]}$ of the system (1) for the null incremental input $\widehat{\Delta u}_{j+l}^{[\mathrm{p}]} = 0$ for $l \geq 0$. Here, we actually have $\overline{y}_{j+l}^{[\mathrm{p}]} = y_j^{[\mathrm{p}]} + \sum_{m=1}^{l}\overline{\Delta y}_{j+m}^{[\mathrm{p}]}$ from (4), where $\overline{\Delta y}_{j+l}^{[\mathrm{p}]}$ denotes the natural response of the difference system of (2) with $\boldsymbol{J}_f$ replaced by $\boldsymbol{M}_c$. The ith column and the jth row of the matrix $\boldsymbol{G}$ is given by $G_{ij} = g_{i-j+N_1}$, where g_l for $l = \cdots, -2, -1, 0, 1, 2, \cdots$ is the unit step response $y_{j+l}^{[\mathrm{p}]}$ of (4) for $\widehat{y}_{j+l}^{[\mathrm{p}]} = \widehat{u}_{j+l}^{[\mathrm{p}]} = 0$ $(l < 0)$ and $\widehat{u}_{j+l}^{[\mathrm{p}]} = 1 (l \geq 0)$. It is easy to derive that the unit response g_l of (4) is obtained as the impulse response of (2). Then, we have $\widehat{\Delta\boldsymbol{u}}^{[\mathrm{p}]}$ which minimizes J by $\widehat{\Delta\boldsymbol{u}}^{[\mathrm{p}]} = (\boldsymbol{G}^T\boldsymbol{G} + \lambda_u\boldsymbol{I})^{-1}\boldsymbol{G}^T(\boldsymbol{r}^{[\mathrm{p}]} - \overline{\boldsymbol{y}}^{[\mathrm{p}]})$, and then we have $\widehat{u}_j^{[\mathrm{p}]} = u_{j-1}^{[\mathrm{p}]} + \widehat{\Delta u}_j^{[\mathrm{p}]}$.

2.4 Control and Training Iterations

To improve the control performance, we execute iterations of the following phases.

(i) **control phase:** Control the plant by a default control schedule at the first iteration, and by the GPC using the bagging CAN2s obtained by the training phase otherwise.
(ii) **training phase:** Train the bagging CAN2s with the dataset $D^{[n]} = \{(\Delta \boldsymbol{x}_j^{[\mathrm{p}]}, \Delta y_j^{[\mathrm{p}]} | j = 1, 2, \cdots, n)\}$ obtained in the control phase.

The control performance at an iteration depends on the bagging CAN2 obtained at the previous iterations. So, for the actual control of the plant, we use the best bagging CAN2 obtained through a number of iterations. We store and selectively use multiple best bagging CAN2s for multiple objectives in multiobjective control.

2.5 Switching Multiple Bagging CAN2s for Robustness to Parameter Change

To cope with parameter change of the plant, we employ the following method to switch bagging CAN2s. Let $\mathrm{CAN2}^{[\mathrm{bag}][\theta_s]}$ denote the best bagging CAN2 obtained for the plant with parameter θ_s ($s \in S = \{1, 2, \cdots, |S|\}$) through the control and training iterations as described above.

Step 1: At each discrete time j in the control phase, obtain $\boldsymbol{M}^{[\mathrm{bag}][\theta_s]}$ for all $s \in S$. where $\boldsymbol{M}^{[\mathrm{bag}][\theta_s]}$ denotes $\boldsymbol{M}^{[\mathrm{bag}]}$ given in (7) of $\mathrm{CAN2}^{[\mathrm{bag}][\theta_s]}$.

Step 2: Select the s^*th bagging CAN2, or $\mathrm{CAN2}^{[\mathrm{bag}][\theta_{s^*}]}$, which provides the minimum MSE (mean square prediction error) for the recent N_e outputs,

$$s^* = \underset{s \in S}{\operatorname{argmin}} \frac{1}{N_e} \sum_{l=0}^{N_e - 1} \| \Delta y_{j-l}^{[\mathrm{p}]} - \widehat{\Delta y}_{j-l}^{[\mathrm{p}][s]}) \|^2 \ , \tag{10}$$

where $\widehat{\Delta y}_{j-l}^{[\mathrm{p}][s]}$ is the predictive output of $\mathrm{CAN2}^{[\mathrm{bag}][\theta_s]}$ at time $j - l$.

3 Numerical Experiments of Crane System

In order to examine the effectiveness of the present method, we execute numerical experiments on the following crane system shown in [2].

3.1 Overhead Traveling Crane System

We consider the overhead traveling crane system shown in Fig. 1(b). From the figure, we have the position and motion equations as

$$(x, y) = (X + r \sin\phi, r \cos\phi) \tag{11}$$

$$m(\ddot{x}, \ddot{y}) = (-T \sin\phi - mC\dot{\phi} \cos\phi, mg - T \cos\phi - mC\dot{\phi} \sin\phi) \tag{12}$$

$$M\ddot{X} = F + T \sin\phi \tag{13}$$

where (x, y) and m are the position and the weight of the suspended load, $(X, 0)$, M and F are the position, weight and driving force of the trolley, r and ϕ are the length and the angle of the rope, T is the tension of the rope, and C is the viscous damping coefficient. From (11) and (12), we have the nonlinear second order differential equation of ϕ given by $r\ddot{\phi} + (C + 2\dot{r})\dot{\phi} + g\sin\phi + \ddot{X}\cos\phi = 0$. Thus, with (13), the transition of the state $\boldsymbol{x} = \left(\phi, \dot{\phi}, X, \dot{X}\right)^T$ is given by

$$\dot{\boldsymbol{x}} = h(\boldsymbol{x}) = \begin{bmatrix} \dot{\phi} \\ -\dfrac{C + 2\dot{r}}{r}\dot{\phi} - \dfrac{g}{r}\sin\phi - \dfrac{F + T\sin\phi}{rM}\cos\phi \\ \dot{X} \\ \dfrac{F + T\sin\phi}{M} \end{bmatrix}, \tag{14}$$

where $T = m\sqrt{(\ddot{x} + C\dot{\phi}\cos\phi)^2 + (\ddot{y} - g + C\dot{\phi}\sin\phi)^2}$ is also a function of $\boldsymbol{x}$. The control objective is to move the horizontal position of the load, $x = X + r\sin\phi$, to a destination position x_d by means of operating F.

3.2 Parameter Settings

The control objective is to move the load position of the crane from $x = 0$ to the destination position $x_d = 5$m within the overshoot $x_{\rm OS}$ less than 100mm. We obtain discrete signals by $u_j^{[\rm p]} = F(jT_v)$ and $y_j^{[\rm p]} = x(jT_v)$ with (virtual) sampling period $T_v = 0.5$s. Here, we use virtual sampling method shown in [3], where the discrete model is obtained with T_v (virtual sampling period) while the observation and operation are executed with shorter actual sampling period $T_a = 0.01$s. We use $k'_y = k'_u = 4$ for enlarged input vector $\Delta\boldsymbol{x}_j^{[\rm p]}$, and $N_y = 20$, $N_u = 1$ and $\lambda_u = 0.01$ for the GPC. We used $N_e = 8$ samples for (10).

We use a model crane with trolley weight $M = 100$kg, damping coefficient $C = 0.5$m/s, maximum driving force $F_{\max} = 10$N. We denote the crane with the rope length r and load weight m by $\mathrm{CRANE}^{[r,m]}$ or $\mathrm{CRANE}^{[\theta]}$ for $\theta = (r, m)$. We examine the robustness to 90 combinations of $r = 2, 3, \cdots, 10$ [m] and $m = 10, 20, \cdots, 100$ [kg]. Before this examination, we train CAN2s with $\mathrm{CRANE}^{[\theta_s]}$ for $\theta_s = (2, 10)$, (2,100), (10,10), (10,100) for $s = 1$, 2, 3, 4, respectively. Let $\mathrm{CAN2}_{\rm OS}^{[\theta_s]}$ and $\mathrm{CAN2}_{\rm ST}^{[\theta_s]}$ denote the best CAN2 which have achieved smallest overshoot and settling time, respectively, for $\mathrm{CRANE}^{[\theta_s]}$ through 20 control and training iterations. Here, at each iteration, we use the data of the current and previous iteration for the dataset to train the CAN2 because the number of obtained data becomes huge and time consuming as the number of iterations increases and the control performance does not seem improved even if we use all data. In order to uniquely select the CAN2, the overshoot $x_{\rm OS}$ and the settling time $t_{\rm ST}$ are ordered by $x_{\rm OS} + \epsilon t_{\rm S}$ and $t_{\rm ST} + \epsilon x_{\rm OS}$, respectively, with small $\epsilon = 10^{-5}$. We have used the set of CAN2s, or $\mathrm{CAN2}_{\rm OS}^{[\theta_S]} = \{\mathrm{CAN2}_{\rm OS}^{[\theta_s]} | s \in S\}$ and $\mathrm{CAN2}_{\rm ST}^{[\theta_S]} = \{\mathrm{CAN2}_{\rm ST}^{[\theta_s]} | s \in S\}$, for the present switching controller. We similarly obtain the sets of bagging CAN2s, i.e. $\mathrm{CAN2}_{\rm OS}^{[\rm bag][\theta_S]}$ and $\mathrm{CAN2}_{\rm ST}^{[\rm bag][\theta_S]}$. We use the number of bags to

Table 1. Statistical summary of overshoot and settling time obtained in the control of the crane for 90 combinations of rope lengths and load weights. The boldface figures indicate the best (smallest) result in each block.

CAN2 employed	tuned N for $(\theta_1,\theta_2,\theta_3,\theta_4)$	settling time $t_{\rm ST}$ [s]				overshoot $x_{\rm OS}$[mm]			
		mean	min	max	std	mean	min	max	std
$\mathrm{CAN2}_{\rm ST}^{[\theta_S]}$	(20,30,20,8)	**21.4**	19.6	**24.1**	**0.8**	22.4	**0**	67	13.9
$\mathrm{CAN2}_{\rm ST}^{[\mathrm{bag}][\theta_S]}$	(30,20,6,8)	22.0	**19.1**	26.0	1.2	**11.4**	**0**	**57**	**12.3**
$\mathrm{CAN2}_{\rm OS}^{[\theta_S]}$	(30,20,10,6)	27.1	22.4	36.4	2.6	3.3	**0**	**38**	5.6
$\mathrm{CAN2}_{\rm OS}^{[\mathrm{bag}][\theta_S]}$	(20,30,20,8)	**26.1**	**19.6**	**29.7**	**2.4**	**0.7**	**0**	**38**	**4.1**
$\mathrm{CAN2}_{\rm OS}^{[\mathrm{bag}][\theta_S^+]}$	(20,30,20,8)	**25.2**	21.3	30.7	**1.9**	**0.2**	**0**	**5**	**0.8**

be $b = 10$ and the bag size ratio $\alpha = 0.7$. We have optimized the number of units for each single and bagging CAN2 from $N = 40$, 30, 20, 10, 8, 6, 4, which indicates the number of piecewise regions of piecewise linear approximation by the CAN2.

3.3 Results and Analysis

A statistical summary of achieved overshoot $x_{\rm OS}$ and settling time $t_{\rm ST}$ is shown in Table 1, and four examples of time course of the input F and the output X and x for the best and the worst control using bagging CAN2s is shown in Fig. 2. From Fig. 2, we can see the performance in time by the best (top) and the worst (bottom) controllers for reducing settling time (left) and overshoot (right), respectively.

From Table 1, we can see that all controller has achieved overshoot less than the allowable 100mm. The top two rows indicate the results by the controllers for reducing settling time, and we can see that the mean, maximum (max) and standard deviation (std) of the settling time are not improved by the bagging $\mathrm{CAN2}_{\rm ST}^{[\mathrm{bag}][\theta_S]}$, while other performance is improved. The third and fourth rows indicate the results by the controllers for reducing overshoot, and we can see that all values are improved (reduced) by bagging $\mathrm{CAN2}_{\rm OS}^{[\mathrm{bag}][\theta_S]}$ from single $\mathrm{CAN2}_{\rm OS}^{[\theta_S]}$.

In order to examine these results precisely, we show the settling time and overshoot for each of 90 parameter values in Fig. 3. From the figure on the left, we can see that single $\mathrm{CAN2}_{\rm ST}^{[\theta_S]}$ has achieved better performance than bagging $\mathrm{CAN2}_{\rm ST}^{[\mathrm{bag}][\theta_S]}$ on average and it does not seem reasonable to improve the performance from the result of the bagging $\mathrm{CAN2}_{\rm ST}^{[\mathrm{bag}][\theta_S]}$. On the other hand, the result of overshoot on the right of Fig. 3 shows that bagging $\mathrm{CAN2}_{\rm OS}^{[\mathrm{bag}][\theta_S]}$ has achieved no overshoot ($x_{\rm OS} = 0$) except 8 cases, and the biggest overshoot $x_{\rm OS} = 38$mm occurs at $(r, m) = (2, 80)$. Thus, we train to make a new bagging $\mathrm{CAN2}_{\rm OS}^{[\mathrm{bag}][2,80]}$ and add it to $\mathrm{CAN2}_{\rm OS}^{[\mathrm{bag}][\theta_S]}$ as $\mathrm{CAN2}_{\rm OS}^{[\mathrm{bag}][\theta_S^+]} = \mathrm{CAN2}_{\rm OS}^{[\mathrm{bag}][\theta_S]} \cup \mathrm{CAN2}_{\rm OS}^{[\mathrm{bag}][2,80]}$. The performance achieved by $\mathrm{CAN2}_{\rm OS}^{[\mathrm{bag}][\theta_S^+]}$ is shown in the bottom row of Fig. 1 and in the right of Fig. 3. As we can see that the mean, maximum (max) and standard deviation (std) are improved from $\mathrm{CAN2}_{\rm OS}^{[\mathrm{bag}][\theta_S]}$, and the number of cases accompanied with positive overshoot are reduced to 5 from 8.

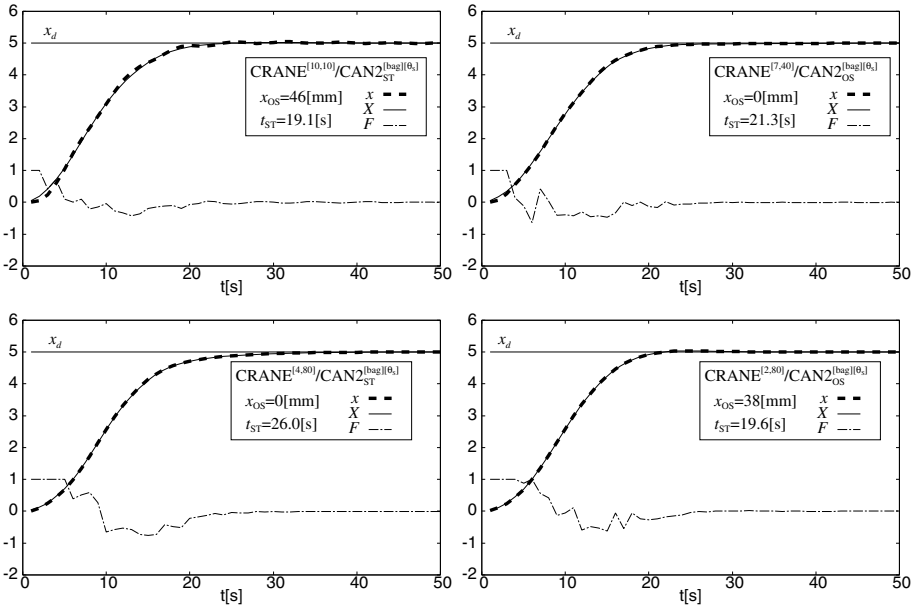

Fig. 2. Examples of time course of x[m], X[m] and F[10N]. Among the control for 90 parameter values, the results of the smallest (best) and biggest (worst) settling time by $\mathrm{CAN2}_{\mathrm{ST}}^{[\mathrm{bag}][\theta_S]}$ are shown in top left and bottom left, respectively, and those of the smallest and biggest overshoot by $\mathrm{CAN2}_{\mathrm{OS}}^{[\mathrm{bag}][\theta_S]}$ are shown in top right and bottom right, respectively.

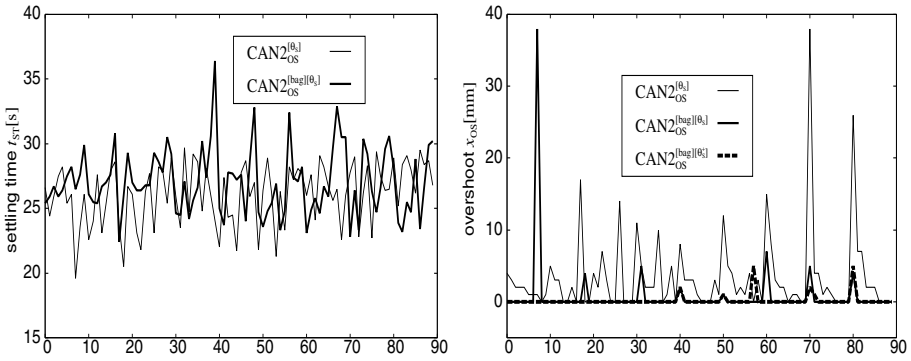

Fig. 3. Experimental result of settling time t_{ST} [s] (left) and overshoot x_{OS} [mm] (right) in the control of the crane for 90 combinations of rope lengths r[m] and load weights m[kg]. The horizontal axis indicates the parameter values ordered as $(r, m) = (2, 10), (2, 20), \cdots, (10, 100)$.

Here, note that we have a different result in our previous study [2], where the mean and the maximum overshoot are 0.02 and 2, respectively, but the mean settling time are 32.0s by $\mathrm{CAN2}_{\mathrm{OS}}^{[\theta_S]}$. The difference is owing that a constant number of units $N = 20$ for all $\mathrm{CAN2}_{\mathrm{OS}}^{[\theta_s]}$ $(s = 1, 2, 3, 4)$ is used in [2] which derives smaller overshoot but larger settling time on average. By means of tuning N for each θ_s in the present experiments, we have achieved smaller settling time while the overshoot is reduced by augmenting the set of bagging CAN2s as shown above.

4 Conclusion

We have focused on an improvement of control performance by means of replacing single CAN2s by bagging CAN2s of the robust controller using difference signals which we are developing. Via numerical experiments of a crane system, we have shown the effectiveness of the present method. From the point of view of multiobjective control, two objectives to reduce settling time and overshoot have different properties. Namely, in the present method, the settling time is reduced by tuning the number of units, N, of the CAN2s, while the overshoot is reduced by using bagging CAN2s replacing single CAN2s and the augmentation of bagging CAN2(s) for reducing plant-parameter-specific overshoot(s). We would like to analyze the present method much more in our future research, especially the effect and the role of the enlargement of the dimensionality of the input vector to the CAN2.

Acknowledgments. This work was supported by JSPS KAKENHI Grant Number 24500276.

References

1. Kurogi, S., Yuno, H., Nishida, T., Huang, W.: Robust control of nonlinear system using difference signals and multiple competitive associative nets. In: Lu, B.-L., Zhang, L., Kwok, J. (eds.) ICONIP 2011, Part III. LNCS, vol. 7064, pp. 9–17. Springer, Heidelberg (2011)
2. Huang, W., Kurogi, S., Nishida, T.: Robust controller for flexible specifications using difference signals and competitive associative nets. In: Huang, T., Zeng, Z., Li, C., Leung, C.S. (eds.) ICONIP 2012, Part V. LNCS, vol. 7667, pp. 50–58. Springer, Heidelberg (2012)
3. Kurogi, S., Nishida, T., Sakamoto, T., Itoh, K., Mimata, M.: A simplified competitive associative net and a model-switching predictive controller for temperature control of chemical solutions. In: Proc. of ICONIP 2000, pp. 791–796 (2000)
4. Kurogi, S., Ren, S.: Competitive associative network for function approximation and control of plants. In: Proc. NOLTA 1997, pp. 775–778 (1997)
5. Kohonen, T.: Associative Memory. Springer (1977)
6. Ahalt, A.C., Krishnamurthy, A.K., Chen, P., Melton, D.E.: Competitive learning algorithms for vector quantization. Neural Networks 3, 277–290 (1990)
7. Kurogi, S., Sawa, M., Ueno, T., Fuchikawa, Y.: A batch learning method for competitive associative net and its application to function approximation. In: Proc. of SCI 2004, vol. 5, pp. 24–28 (2004)
8. Deb, K.: Multi-Objective Optimization Using Evolutionary Algorithms. John Wiley & Sons (2009)
9. Optiz, D., Maclin, R.: Popular ensemble methods: an empirical study. Journal of Artificial Intelligence Research 11, 169–198 (1999)
10. Clarki, D.W., Mohtadi, C.: Properties of generalized predictive control. Automatica 25(6), 859–875 (1989)