THE APPLICATION OF CONTROL USING NEURO-DYNAMIC PROGRAMMING WITH A FEATURE MAP

Dong Kyu Kim[*], Joeng Ho Ahn[**], Kwang Soon Lee[***], and Dae Ryook Yang[*,1]

** Department of Chemical and Biological Engineering, Korea University, Seoul, Korea*
*** The School of Applied Chemistry, Sungkyunkwan University , Suwon, Korea*
**** Department of Chemical Engineering, Sogang University, Seoul, Korea*

Abstract: For the general nonlinear processes control problem, the most rigorous approach is to use dynamic optimization. However, as the size of the problem grows, the Dynamic Programming (DP) approach is suffered from the burden of calculation which is called as 'Curse of Dimensionality'. To overcome this problem, a cost-approximator can be used to obtain the optimal control input policy minimizing the value of cost, called the Neuro-Dynamic Programming (NDP) algorithm. In this study, the NDP algorithm was applied to pH neutralization process in both simulations and experiments. A systematic approach of NDP to a pH neutralization process has been proposed and the performance of the NDP approach has been evaluated through the comparison with PI control. Also, a few related issues such as selection of approximator and a feature map of the states are investigated.

Keywords: pH neutralization process, the Neuro-Dynamic Programming (NDP), *k*-nearest neighbor method (*k*NN), Neural network (NN), Feature map

1. INTRODUCTION

As the modern computing power is enhancing due to the development of fast computer, the requirements of the control performance become tighter and the target processes has been extended to various non-conventional processes such as those in Biotechnology (BT) and Nano Technology (NT) areas. Thus, the models for control become more sophisticated and the modern model-based control techniques have to process vast amount of information to meet the high standard of control performance. In order to achieve more accurate and precise control performance, the most rigorous solution for the control of nonlinear system is to use the optimal control strategy obtained by dynamic optimization considering the details of the process characteristics such as nonlinearity and high dimensionality.

The most desirable control strategy can be obtained using standard Dynamic Programming (DP) with rigorous process model. The aim of DP is to find the optimal time-varying input policies by minimizing the objective function which is defined according to the specific control purposes and in most cases, the optimal strategy is calculated rather numerically than analytically. If the size of problem is large, the calculation load can be enormous and the solution cannot be obtained within the given sampling time even with quite fast computer. This problem is called 'Curse of Dimensionality' and this makes the on-line control using DP virtually impossible (Kaisare, *et al*., 2003). However, as the Neuro-Dynamic Programming (NDP) approach is introduced, the application of DP approach to nonlinear processes becomes feasible and the field of application for NDP is growing. This approach is to perform the vast amount of calculation offline, to learn the optimal strategy in a simple form of approximation by using the data from simulations and experiments and to calculate the optimal strategy online based on the cost function approximator which is trained offline. Cost-to-go or profit-to-go function as a performance objective function for DP can be approximated by a nonlinear function or neural network (NN) and this approach can reduce the calculation burden so that the dynamic programming approach can be applied online. But the NN requires appropriate training before use and the training of NN is not trivial for many cases. To avoid the difficulty in NN training, local approximation method could be used such as *k*-nearest neighbor (*k*NN) method.

In this study, Simulation-Approximation-Evolution (SAE) algorithm suggested by Kaisare *et al*. (2003) is

[1] Author to whom correspondence should be addressed. Email: dryang@korea.ac.kr

investigated on a pH neutralization process. The SAE algorithm is based on neuro-dynamic programming (NDP) approach. In this approach, the Bellman iteration is very important for its performance since the optimal control strategy is usually unknown. In this study to investigate the control performance of NDP approach, the training data for cost function approximator are obtained from an optimal control strategy of pH neutralization process considering the peculiar nature of the process so that the Bellman iterations are not necessary.

The cost function approximation is usually based on the state information. However, the outputs of the process are the function of states and the output equation involves process parameters which are uncertain. In this case, the control based on the state may result steady-state offset in the process output. Thus, the feature map of the process information has been added in this study to eliminate the steady-state offset. In addition to that, the various aspects of control performance of NDP approach have been investigated through the simulations and experiments of a pH neutralization apparatus.

2. NEURO-DYNAMIC PROGRAMMNG (NDP)

2.1 Dynamic Programming

A discrete-time dynamic system can be described by an *n*-dimensional state vector $x(k)$ and an *m*-dimensional input vector $u(k)$ at time step *k*. Choice of an *m*-dimensional control vector $u(k)$ determines the transition of the system from $x(k)$ state to $x(k+1)$ through the following relations (Bertsekas and Tsitsiklis, 1996; Bryson, 1999),

$$x(k+1) = F_h\left(x(k), u(k)\right) \tag{1}$$

where F_h denotes the process model equation and *h* represents the sampling time. A general dynamic optimization problem for such system is to find the optimal sequence of control vectors $u(k)$ for $k=0, \ldots, N-1$ to minimize a performance index which is related with cost-to-go function (J).

$$J = \sum_{i=0}^{N-1} \phi(x(i), u(i)) + \phi_N \tag{2}$$

where ϕ is the one-stage-cost and ϕ_N represents the final cost. Among many ways, the most popular one-stage-cost can be chosen as follows, with the weighting factors Q and R.

$$\phi(x(k), u(k)) = \left(x(k+1) - x_{sp}\right)^T Q \left(x(k+1) - x_{sp}\right) + \Delta u(k)^T R \Delta u(k) \tag{3}$$

$$\phi_N = \left(x(N+1) - x_{sp}\right)^T M \left(x(N+1) - x_{sp}\right) \tag{4}$$

where $k=0, \ldots, N-1$, $u(0)=u_0$, and x_{sp} denotes the set point. In addition to that, $\Delta u(k)=u(k)-u(k-1)$.
Then, the optimal cost-to-go can be expressed as follow at time-step *k* and.

$$J_k^*(x(k)) = \min_u \left[\sum_{i=k}^{N-1} \phi(x(i), u(i)) + \phi_N \right] \tag{5}$$

where superscript * implies the optimal value. If N is infinite, then it becomes the infinite horizon cost-to-go function. It can be expressed as a recursive form.

$$\begin{aligned} J_k^*(x(k)) &= \min_u \left[\phi(x(k), u(k)) + \sum_{i=k+1}^{N-1} \phi(x(i), u(i)) + \phi_N \right] \\ &= \min_u \left[\phi(x(k), u(k)) + J_{k+1}^*(x(k+1)) \right] \\ &= \min_u \left[\phi(x(k), u(k)) + J_{k+1}^*(F(x(k), u(k))) \right] \end{aligned} \tag{6}$$

$$J_N^*(x(N)) = \phi_N \tag{7}$$

It can be shown to satisfy the Bellman equation (Bertsekas and Tsitsiklis, 1996).

For simplicity, $J_k^*(x(k))$ will be shortened as $J^*(k)$. The final goal of DP is to find the input strategy $u(k)$, $k=1, \ldots, N-1$ so that the optimal cost-to-go function $J^*(k)$ satisfies the Bellman equation for all time-step *k*. The solution can usually be obtained numerically and it suffers from the curse of dimensionality when it involves the gridding of large state space dimension. In order to circumvent the problem, one approach suggested by Kaisare, *et al*. (2003) described in the next section can be applied.

2.2 Simulation-based dynamic programming

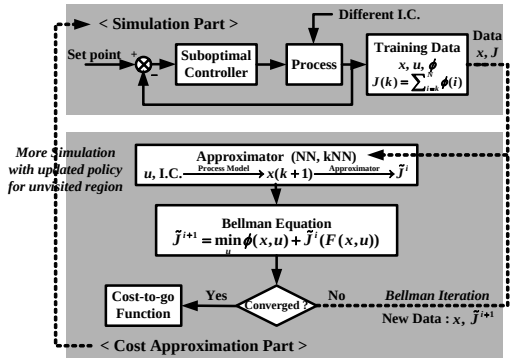

Fig. 1. Architecture for offline computation of cost-to-go approximation.

Simulation-Approximation-Evolution (SAE) algorithm (Kaisare, *et al*., 2003) is one of the reinforcement learning methods and it involves computation of the converged cost-to-go approximation offline, which is described in Fig. 1. The SAE algorithm is roughly composed of two parts. The first part is "Simulation Part". Simulation is performed with suboptimal control law to make training data set which is used for the calculation of the infinite horizon cost-to-go function in (6) for each state visited during the simulation, and the suboptimal cost-to-go function is calculated as follow.

$$J(k) = \sum_{i=k}^{N} \phi(i) \quad (8)$$

where N is sufficiently large for the system to reach new steady state. The second part is "Cost Approximation Part." In this part, the cost-to-go function approximation is performed by fitting a neural network or other function approximator to the data from "Simulation Part." In addition to that, Bellman iteration and policy update procedure is performed to improve the approximation of the cost-to-go function (Kaisare, *et al*., 2003).

Since the optimal control law is not available to begin with, a suboptimal control policy is used for the cost-to-go approximation and the resulting control law has to be suboptimal. To improve the approximation, the cost or value iteration can be performed until convergence based on the Bellman equation (Kaisare, *et al*., 2003).

$$\tilde{J}^{i+1} = \min_{u} \phi(x,u) + \tilde{J}^{i}\left(F_h(x,u)\right) \quad (9)$$

This step may impose an enormous computational burden, but it is performed offline.

2.3 Approximator

In the algorithms related to NDP, the performance of the approximator for the cost-to-go approximation is crucial. As approximators, the global approximator and the local approximator can be considered. Global approximators like neural network (NN), polynomial, and etc. are the parametric approximators which require extensive offline training. The training of NN is very difficult procedure to converge and the training of NN is quite critical to the performance of the NDP approaches. The local approximators such as *k*-nearest neighbor (*k*NN), kernel-based approximator, and so on are nonparametric approximators which require extensive online querying instead of offline training. If both methods provide comparable accuracy of approximation, the choice of approximator will not affect the control performance of the NDP approach.

2.4 Feature maps

If the states are not measured, rather estimated and the set points for states are obtained from a model of output equation, the NDP approach will track the calculated set points which do not exactly represent the important process outputs. In this case, the NDP approach may result in the steady-state offset. The remedy of this problem is to use feature map so that the set points for actual process output can be defined. The feature map in this case can be the output equation relating the states and outputs. Then, the set points for the objective function in (3) can be the one for actual output instead of ones for estimated states.

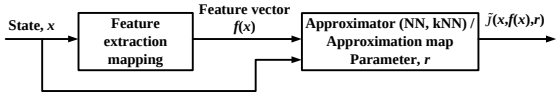

Fig. 2. The approximation architecture that uses feature map.

3. pH NEUTRALIZATION PROCESS

The pH neutralization process has long been taken as a representative benchmark problem of nonlinear chemical process control due to its nonlinearity and time-varying nature. In this study, the pH neutralization process is selected as the control target system with NDP approach.

3.1 pH neutralization process

The neutralization is a chemical reaction. The control objectives are to drive the system to a different pH conditions (tracking control) or to regulate the effluent pH value despite the disturbance by manipulating the flow rate of titrating stream (Henson and Seborg, 1994, 1997). The process is illustrated in Fig. 3 and the operating conditions are shown in Table 1.

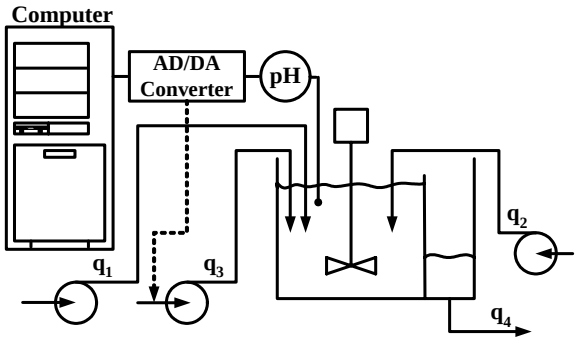

Fig. 3. The pH neutralization process.

Table 1 Operating conditions of pH neutralization process

Symbols	Values	Symbols	Values
V	2500 [ml]	$[q_1]$	0.003 M HNO_3 5.0×10^{-5} M H_2CO_3
q_1	9.0 [ml/s]		
q_2	0.6 [ml/s]	$[q_2]$	0.01 M $NaHCO_3$
q_3	8.5 [ml/s]	$[q_3]$	0.003 M NaOH 5.0×10^{-5} M $NaHCO_3$

The reactor type of the neutralization process is a continuous stirred tank reactor (CSTR) with baffles, which has a volume of 2.5L. The inlet stream consists of a strong acid stream (q_1: feed solution), a

weak acid stream (q_2: buffer solution) and a strong base stream (q_3: titrating solution), which are pumped to the reactor. It is assumed that the perfect mixing in tank and the complete dissociation in solution at 25°C are reached (Yoo, *et al*., 2004).

3.2 Model for pH neutralization Process

Generally, the strong acid-base reaction is always assumed to reach equilibrium in water solution almost instantly. This implies the reaction rates approach infinity. So, the reaction rate terms can be ignored in process model which can be simplified. From this approach, Gustafsson and Waller proposed a model using reaction invariants (Gustafsson and Waller, 1983; Yoo, *et al*., 2004).

As the strong acid and base solutions are completely dissociated into ions, the chemical reactions with a weak acid solution reach equilibrium state. The chemical reactions in the system are as follows (Yoo, *et al*., 2004).

$$\begin{aligned} &H_2CO_3 \leftrightarrow HCO_3^- + H^+, \\ &HCO_3^- \leftrightarrow CO_3^{2-} + H^+, \\ &H_2O \leftrightarrow OH^- + H^+. \end{aligned} \tag{10}$$

The total amount of the reaction invariant is not affected by the degree of chemical. According to this fact, the reaction invariants can be derived from the stoichiometry. As Gustafsson and Waller (1992) proposed, two kinds of reaction invariant variables are defined in this process. The first reaction invariant (state variable) is the concentration of charge related ions. The other reaction invariant (another state variable) is the total concentrations related to carbonate ions. The relationship between pH and the reaction invariants is given by a nonlinear equation (Gustafsson and Waller, 1992; Yoo, *et al*., 2004).

Reaction invariants for this process are defined as:

$$\begin{aligned} W_{ai} &= [H^+]_i - [OH^-]_i - [HCO_3^-]_i - 2[CO_3^{2-}]_i \ , \\ W_{bi} &= [H_2CO_3]_i + [HCO_3^-]_i + [CO_3^{2-}]_i \ . \end{aligned} \tag{11}$$

where W_a denotes the charge related reaction invariant, W_b denotes the carbonate ion related reaction invariant, and i=1,2,3,4 for each stream in Fig. 3. The pH value is the negative logarithm of the hydrogen ion concentration ($\mathrm{pH} = -\log[H^+]$), so the pH value can be determined if W_a and W_b are known.

If it is assumed that the flow rates and the concentrations of the feed and buffer streams are known except for two properties, W_{a1} and W_{b2} and they consists of unknown parameters (θ). From this conditions, the following state space model can be constructed (Yoo, *et al*., 2004):

$$\begin{aligned} &\dot{x} = f(x,t) + g(x,t)u + F_\theta(t)\theta, \\ &c(x,y) = 0 \end{aligned} \tag{12}$$

where

$$f(x,t) = \frac{1}{V}\begin{bmatrix} q_2(W_{a2} - x_1) - q_1 x_1 \\ q_1(W_{b1} - x_2) - q_2 x_2 \end{bmatrix}, \ g(x,t) = \frac{1}{V}\begin{bmatrix} W_{a3} - x_1 \\ W_{b3} - x_2 \end{bmatrix},$$

$$F_\theta(t) = \frac{1}{V}\begin{bmatrix} q_1 & 0 \\ 0 & q_2 \end{bmatrix}, \ \theta = \begin{bmatrix} W_{a1} & W_{b2} \end{bmatrix}^T, \ x = \begin{bmatrix} W_{a4} & W_{b4} \end{bmatrix}^T,$$

$u = q_3$, y=pH$_4$, $pK_1 = -\log K_{a1}$, $pK_2 = -\log K_{a2}$,

$$c(x,y) = x_1 + 10^{y-14} - 10^{-y} + x_2 \frac{1 + 2\times 10^{y-pK_2}}{1 + 10^{pK_1 - y} + 10^{y-pK_2}} = 0 \ .$$

3.3 Optimal Control Strategy

In the "Simulation Part" of SAE algorithm (Fig. 1), the suboptimal control law is necessary to get good training data sets. Furthermore, the training data can be improved through the Bellman iterations and the training of the approximator has to be performed for each Bellman iteration. This procedure requires enormous offline calculation burden. However, if the optimal control strategy is known, the improvement of cost-to-go function by Bellman iteration is not necessary. Fortunately, for this process, an optimal control can be devised from a simple principle. From the steady-state balance, the required flow rate, u_f, of titrating stream to make the mixture of inlet streams at the desired pH value can be calculated. Next, to minimize the transient period, the additional amount of titrating stream, S, should be injected in a shortest possible time to make the holdups of the CSTR at the desired pH value (Fig. 4). To solve this problem, an LP problem can be set up as follows.

$$\begin{aligned} &\min_{\Delta u_i} n \\ &\text{subject to } \sum_{i=1}^{n} \Delta u_i = u_f - u_0 \\ &\qquad \sum_{i=1}^{n} (1-i)\Delta u_i \Delta t = S \\ &\qquad -\Delta u_{\max} \le \Delta u_i \le \Delta u_{\max} \end{aligned} \tag{13}$$

where u_0 is the initial steady-state value of input and n is an integer which represents the number of sampling intervals to reach new steady state. In this manner, the effluent pH value can be reached to the desired value in shortest time without overshoot or undershoot as shown in Fig. 4. This control law may not produce exactly optimal strategy due to the residence time of the effluent stream considering the constraints of the flow rates but it is close enough as an optimal control law. Moreover, the required amount of additional injection of the titrating stream, S, can be adjusted to make the performance better. By using this optimal strategy, the laborious Bellman iteration can be omitted in this case.

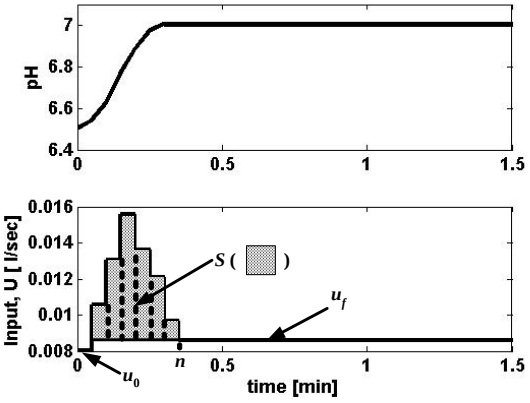

Fig. 4. Optimal control strategy in pH neutralization experiment

4. RESULTS AND DISCUSSIONS

The NDP approach is applied to the pH neutralization process with NN as a global approximator and *k*NN as a nonparametric local optimization. In addition to that, feature map has been used for improved performance of NDP algorithm.

4.1 Choice of Approximator

The different choices of approximator for cost function, a multilayer feedforward NN and a local approximator using *k*-Nearest Neighbor are investigated. The NN can approximate the entire data with a single scheme and it does not require to store the training data while the training of NN takes long time and can be quite difficult. Meanwhile, the *k*NN approach does not require training while the training data have to be stored and it takes time to query nearest data points. In this study, there are many difficulties in training NN for pH neutralization process, it is recommended to use the *k*NN approach. In terms of control performance, both methods do not exhibit any difference as long as the approximation has same degree of precision as shown in Fig. 5.

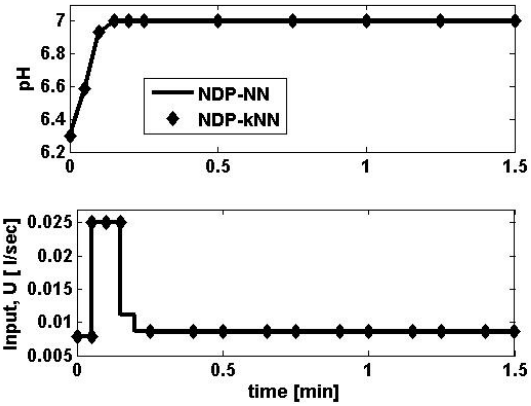

Fig. 5. Comparison of results between NDP approaches using *k*NN and NN with respect to set point change (pH 6.3→7).

4.2 Performance Comparison of NDP Approach

The case of multiple step changes in set point shows that the NDP using *k*NN outperforms the well-tuned PI control as expected as shown in Fig. 6. Also, for the disturbance in feed composition, the NDP approach can reject the disturbance effect very efficiently compared to PI control as shown in Fig. 7.

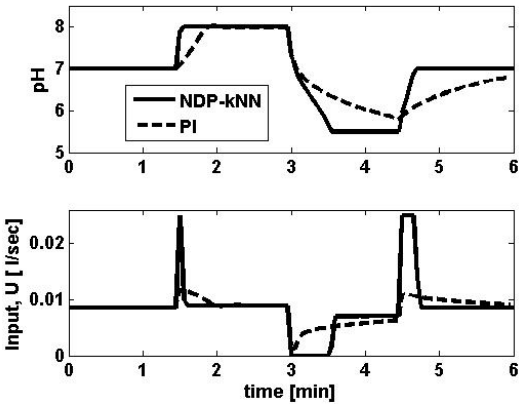

Fig. 6. Comparison of results between PI control and NDP using *k*NN with respect to multi-step set point change (7→8→5.5→7).

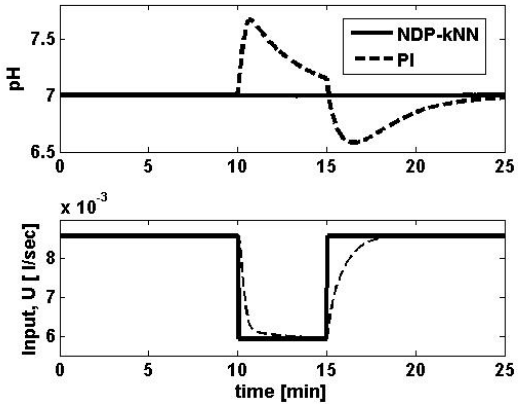

Fig. 7. Comparison of results between PI control and NDP using *k*NN with respect to disturbance (15% decrease in W_{a1} at time=10 min.).

4.3 Effect of Feature Map

The NDP approach requires the cost-to-go function values calculated based on the states. In pH neutralization process, the states are not measured and they are estimated by extended Kalman filter. Since the measured variable is the pH value, the set points for objective function in NDP should be calculated using the model. If there are mismatch between the model and process, the calculated set points may not coincide with the desired pH value. As shown in Fig. 8, steady-state offset can be observed without feature maps while the use of feature map can eliminate the steady-state offset.

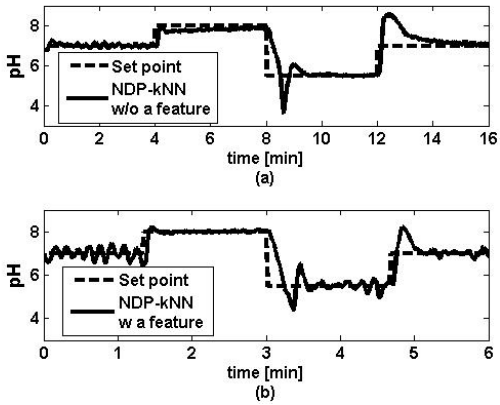

Fig. 8. Experimental comparison between using NDP using *k*NN with and without a feature map with respect to multi-step set point change (7→8→ 5.5→7).

4.4 Experimental Results for pH Neutralization with NDP

To verify NDP algorithm for real system, pH neutralization experiments were performed. The control by NDP algorithm showed faster and better control performance than PI controller in Fig. 9. Note that the NDP approach showed some overshoot at the later parts of the experiment even though the first step change showed almost perfect performance. This performance degradation seemed to be caused by the imperfect modeling of the pH neutralization process. The improvement on the model accuracy should be investigated further in the next research.

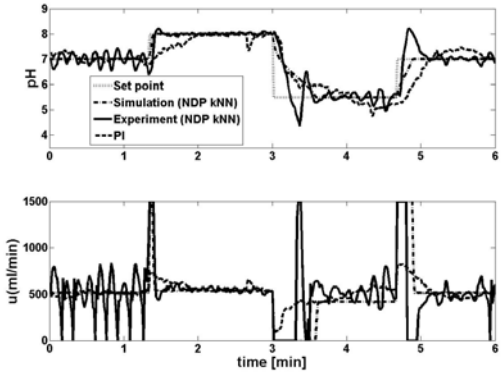

Fig. 9. Experimental comparison between using NDP and PI controllers with simulation with respect to multi-step set point change (7→8→ 5.5→7).

5. CONCLUSIONS

From the simulation and the experiments of a pH neutralization process, NDP algorithm using either the global approximator (NN) or the local approximator (*k*NN) outperforms the well-tuned PI control. These results are not surprising because NDP method uses much more information and computation. However, if the process is quite complex, this approach can achieve precise optimal control performance without excessive online computational burden. In this study, the NDP approach is applied to a chemical process of pH neutralization which is a representative benchmark nonlinear process and the possibility of applying DP concept even with short sampling period to complex nonlinear chemical processes is verified. In terms of offline preparation of NDP approach, the local approximators such as *k*NN are preferred over global approximators in the light of cost-to-go approximation. The local approximator can avoid the hard problem of training of approximator. Also, the remedy for the cases of uncertainty in output equation is suggested. The verification of NDP performance for real system was investigated by pH neutralization experiment. Through the simulations and experiment, the control performance using NDP approach is proved to be applicable to complicated nonlinear processes with very high performance.

6. ACKNOWLEDGEMENTS

This work was supported by the Korea Science and Engineering Foundation (R01-2002-000-00574-0).

7. REFERENCES

Bertsekas, D.P. and J.N. Tsitsiklis (1996). *Neuro-Dynamic Programming*. Athena Scientific, Massachusetts.

Bryson, A.E. Jr (1999). *Dynamic Optimization.* Addison-Wesley Longman, Inc., California.

Gustafsson, T. K. and K.V. Waller (1983). Dynamic Modelingand Reaction Invariant Control of pH. *Chem. Eng. Sci*., **38**, 389-397.

Gustafsson, T. K. and K.V. Waller (1992). Nonlinear and Adaptive Control of pH. *Ind. Eng. Chem. Res*., **31**, 2681-2693.

Henson, M.A. and D.E. Seborg (1994). Adaptive Nonlinear Control of a pH Neutralization Process. *IEEE Trans. on Control Systems Technology*, **2**, 169-182.

Henson, M.A. and D.E. Seborg (1997). Adaptive Input-Output Linearization of a pH Neutralization Process. *Int. J. Adapt. Control Signal Process*, **11**, 171-200.

Kaisare, N.S., J.M. Lee and J.H. Lee (2003). Simulation based strategy for nonlinear optimal control : Application to a microbial cell reactor. *Int. J. Robust Nonlinear Control*, 347-363.

Yoo, A., T.C. Lee and D.R. Yang (2004). Experimental Simultaneous State and Parameter Identification of a pH Neutralization Process Based on an Extended Kalman Filter. *Korean Journal of Chemical Engineering*, **21(4)**,753-760.