Vol. 2(1), pp. 1-6, 2023
eISSN 2976-4009

ASEAN Journal of Process Control

Research Article

Bayesian-ANN Controller for pH-Control

A.W. Hermansson[1*], S. Syafiie[2]

1. School of Engineering and Physical Sciences, Heriot-Watt University Malaysia, No. 1 Jalan Venna P5/2, Precinct 5, 62200 Putrajaya, Wilayah Persekutuan Putrajaya, MALAYSIA
2. Department of Chemical and Material Engineering, Faculty of Engineering-Rabigh, King AbdulAziz University, Jeddah 21911, KINGDOM OF SAUDI ARABIA

*Corresponding Author: a.hermansson@hw.ac.uk
Academic Editor: Jobrun Nandong & Mohd Fauzi Zanil
Received: 07 December 2023; Accepted: 14 December 2023; Published: 31 December 2023

Abstract:

The major problems using artificial neural networks (ANN) for the controller in process control are overfitting and extrapolation. The former commonly occur when unmeasured disturbances affect the process and the ANN will average out the controller output, between the disturbance case and the case where there are no disturbances. Extrapolation problem stems from the ANN suggesting controller outputs in areas where the ANN has not been trained for. This paper proposes a Bayesian weighting approach to improve the generalization performance of ANN controllers in pH control where a feedforward ANN was trained to mimic the behavior of a Robust Model Predictive Controller (RMPC). The proposed approach relies on separating the training data based on the presence and size of disturbances, as well as a Bayesian weighting scheme. The training data was generated from running multiple tests on the RMPC for different requirements and cases of pH control. The training algorithm used was the Levenberg-Marquardt algorithm. The proposed approach was applied to a MATLAB® simulated pH control system. Bayesian-ANN is a novel approach for enhancing the generalization performance of ANNs in pH control. It is straightforward to implement and can be utilized with any ANN architecture. The proposed Bayesian weighted ANN (ANN-MPC) method in controlling pH process is found to be superior to a single ANN and multiple model predictive control (MMPC). This is particularly true with regard to reducing overshoots and oscillations.

Keywords: Bayesian Weighting, Process Control, Nonlinear control, Machine Learning

1. Introduction

Artificial neural networks (ANNs) have emerged as a powerful tool for process control applications. ANNs can learn to map complex nonlinear relationships between process inputs and outputs, making them well-suited for systems with complex dynamics. Though there are some issues with ANN, most commonly overfitting and extrapolation. Overfitting occurs when the ANN cannot generalize and fits too close to the training data [1]. Which means that any process that includes uncertainties (like unmeasured disturbances) will be likely to suffer from overfitting. As a result, the neural network controller averages out the output between cases with uncertainties and those without. Extrapolation is also a problem, however, to keep simplicity, this article focuses on one problem at the time, extrapolation is not addressed.

ANN for process control can be used in two different ways. The most common one in the process control research literature is to model the process using ANNs and then use that model in a Model Predictive Control (MPC) framework [2-7]. There have also been cases where the ANN substituted the controller itself and thus the time-consuming optimization required by a nonlinear MPC can be avoided [8].

These challenges are especially pronounced in the context of pH control, where the process is affected by unmeasured disturbances. These disturbances can cause the pH of the process to fluctuate in pH process, potentially causing the ANN to overfit to the training data and extrapolating to new data outside of the range of values on which it was trained on.

In this paper, we propose a Bayesian weighting approach to improve the generalization performance, and thus reducing the chance of overfitting, of ANN controllers in pH control. The proposed approach involves partitioning the training data based on the set-point, the presence and size of disturbances, reducing the tendency of the ANN average out the uncertainties. The different ANN's are then processes in a Bayesian weighting scheme. The Bayesian weighting scheme helps to average out the predictions of the ANN in cases where the predictions diverge, and to shift to the more accurate prediction as new data becomes available.

The proposed approach was applied to the problem of pH control. A feedforward ANN was trained to mimic the behavior of a Robust Model Predictive Controller (RMPC). The training data was generated from running multiple tests on the RMPC for different requirements and cases of pH control. The training algorithm used was the Levenberg-Marquardt algorithm.

The results show that the proposed Bayesian weighting approach can significantly improve the generalization performance of ANN controllers in pH control. The proposed approach is straightforward to implement and can be utilized with any ANN architecture. It has been demonstrated to be effective in pH control and is anticipated to be effective in other applications as well.

2. Materials and Methods

2.1. pH-system and MMPC controller

The process used for this case is a pH-control system as outlined in [9] and follows the same setup as in [10]. The feed stream (f_w) consist of an aqueous solution of Phosphoric acid (H_3PO_4) and control stream (f_u) consist of an aqueous solution of Calcium hydroxide ($Ca(OH)_2$) as seen in Figure 1. Due to the acid being triprotic and the formation of salts the process is severely non-linear. The system was simulated in MATLAB® based on the standard ODE solver as discussed in [10].

The controller that is used as the input is also based on the setup in [10], which is in Multi Model Predictive Control (MMPC). The MMPC is based on a set of linear models spanned by the desired set points (pH=3, 4, 5, 6 and 7). The linear models are combined using Bayesian weighting to give an average model that the MPC is using for its predictions and optimization to obtain a control sequence.

2.2. ANN

The controllers were run for the standard set points and plus three different levels for the disturbances, the concentration of the phosphoric acid (0.009, 0.01 and 0.011 mol/l). The data collected from this part were the desired set point (y_{ref}), the measured pH (y_m) and the controller input (u). The data was then clustered using the MATLAB® *kmeans* function. The disturbance was assumed to be immeasurable. The data was the modelled using ANN employing the *MATLAB® Neural Network Fitting Toolbox*. This was done both modelling the process itself as in (1) and for the controller as in (2).

$$y_m = ANN_{model,i}(y_{ref}, u) \tag{1}$$

where ($ANN_{model,i}$) is the model i.

$$u = ANN_{controller,i}(y_{ref}) \tag{2}$$

where ($ANN_{controller,i}$) is the controller i. The two ANN's use the same and data sets and clustering, but trained separately. (1) is used to determine the modelled output ($\hat{y}_\iota$), which will be used in the Bayesian weighing to determine which are the most accurate models. (2) is then used to determine the different models output ($\hat{u}_\iota$), which will be weighted together based on how accurate the models in (1) are.

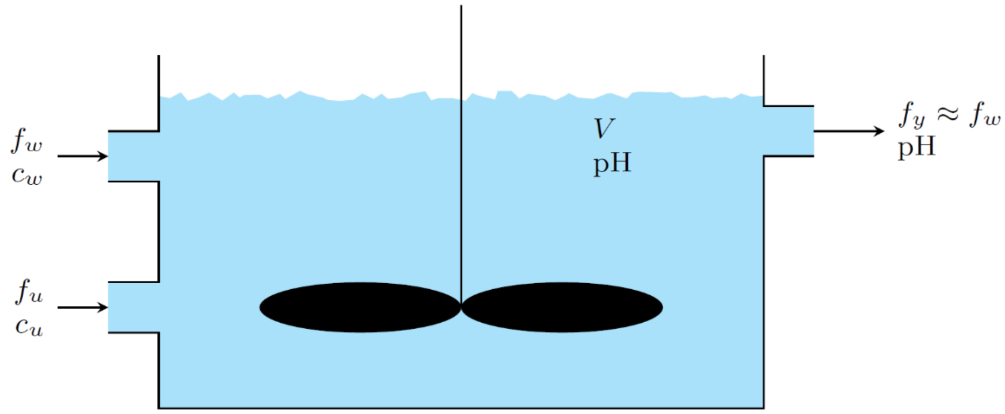

Figure 1. pH system

2.3. Bayesian Weighting

For a full description see [11, 12]

The weighting is based on first computing the residual:

$$\varepsilon^i(k) = y_m(k) - \hat{y}_\iota(k) \tag{3}$$

Follow by the calculation of the probability (P^i) that the ANN model i is the correct model:

$$P^i(k) = \frac{\exp\left(-0.5\varepsilon^i(k)K\varepsilon^i(k)\right) \cdot P^i\ (k-1)}{\sum_{j=1}^{n} \exp(-0.5\varepsilon^j(k)K\varepsilon^j(k)) \cdot P^j\ (k-1)} \tag{4}$$

where $P^i(k\text{-}1)$ is the historical probability from the following time step and K is a convergence parameter used to select speed of which the weights evolve. It should be noted that as (4) is a recursive process, if P^i becomes zero, it would remain zero until reset. Hence, P^i_{hist} are limited by a minimum of 0.02. The weights are obtained by normalizing the probabilities computed in (4) to give:

$$\omega^i = \frac{P^i}{\sum_{j=1}^{n} \mathrm{P^j}} \tag{5}$$

Thus, the ANN-MPC controller output can be computed as:

$$u(k) = \sum_{j=1}^{n} \omega^j(k) \cdot \hat{u}_j(k) \tag{6}$$

Where the only tuning variable available after the ANN's have been obtained is the convergence parameter, K, in (4).

2.4. Clustering and Modelling

The pH-control system was run through various step changes of pH and during various disturbances as described in 2.2. The data was than clustered using MATLAB® *kmeans*. The ideal number of clusters were studied by comparing the within-clusters sums of point-to-centroids. The centroid distance sums were summed together over all the clusters. The sum rapidly decreased until five clusters per disturbance value or if run altogether to fifteen clusters. The clustered data sets were then fed into the *MATLAB® Neural Network Fitting Toolbox* where one ANN were obtained for each cluster thus making n=15 for the weighting and control according to equations (3)-(6).

3. Results and Discussion

The ANN-MPC was implemented based on the model obtained according to section 2.4 and implemented according to section 2.3. The process was simulated in MATLAB® R2022a in an Intel Core i7-6700, 3.40GHz computer with a 64-bit Windows 10 operating system. The process was simulated using their ode45 solver. The simulations were set up with a white Gaussian noise ±2.5% of the pH measurement. The controller based on the obtained ANN models and the Bayesian weights were also implemented in MATLAB®, where the tuning variable K in (4) was given the value of 5 and the initial probabilities were uniformly distributed to all be 1/15.

3.1. Set-Point Tracking

Firstly, a case of set-point changing was applied. Starting from steady state at pH=3, the pH was stepped up and down at 40 minutes intervals following the sequence 3-4-5-6-7-6-5-4-3 as seen in Figure 2. A single ANN based MPC is included to demonstrate the problems with overfitting, where the general trained ANN averages out the different inputs it has been trained on. The more valid comparison in this case is how the ANN-MPC is comparing the multi-model MPC [10]. For the stepping up part they behave similarly, the set points are reached and generally with similar settling time. The difference is that the ANN-MPC does lead to less overshoots, though it appears at a pH of 6 and 7. The controllers perform more differently when stepping down, though very similar at a pH of 6. At a pH of 5 the ANN-MPC is faster compared to when increasing, while the MMPC is slower, while the trend is reversed for the rest, stepping down to 4 and 3. The behavior of the ANN-MPC is due to the closeness of the ANN models at a pH of 4, which causes the weighting spread its weight among more models, which affect the case at 5 while increasing and 3 when decreasing. The MMPC is more affected by the steep gradients effect on the integral action part.

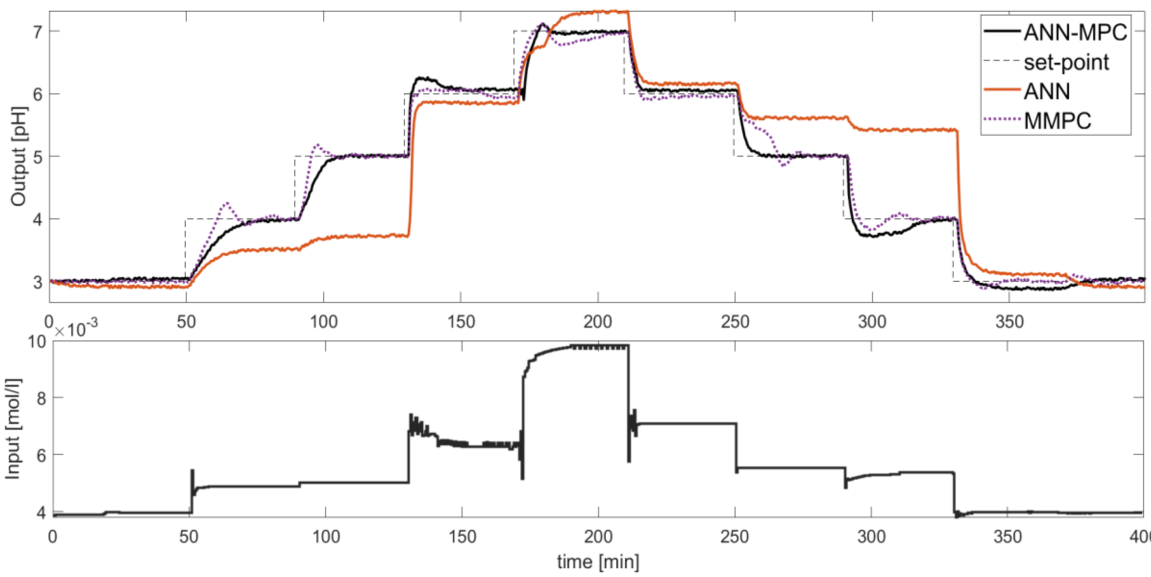

Figure 2. Set-point tracking through the sequence 3-4-5-6-7-6-5-4-3.

3.3. Disturbance rejection

The set-point in this case was kept constant, while a 10% increase in the disturbance variable happened at 50 minutes, while a 10% decrease occurred at 100 minutes, followed by no disturbance at 150 minutes. The responses for the ANN-MPC and MMPC are shown in Figure 3. The single ANN was not included as it kept staying around a pH of 3.6. At the first instance of disturbance at 50 minutes the ANN-MPC performs considerably better than the MMPC with less deviation, less oscillations and faster settling. The disturbance change at 100 minutes creates more problem for the ANN-MPC, while the MMPC is performing fairly like the previous change. The difference for the ANN-MPC is that another model is initially predicting the same output as the most appropriate model (the model that is describing the particular disturbance). The Root mean square error (RMSE) between the different methods are fairly similar with 0.18 for ANN and 0.26 for the MMPC. Where the major contributor for the ANN is the slow convergence after the change at 100 min. The convergence parameter K could be increased for faster convergence to the appropriate model, but that increases the fluctuations at any changes, which was considered unfavorable.

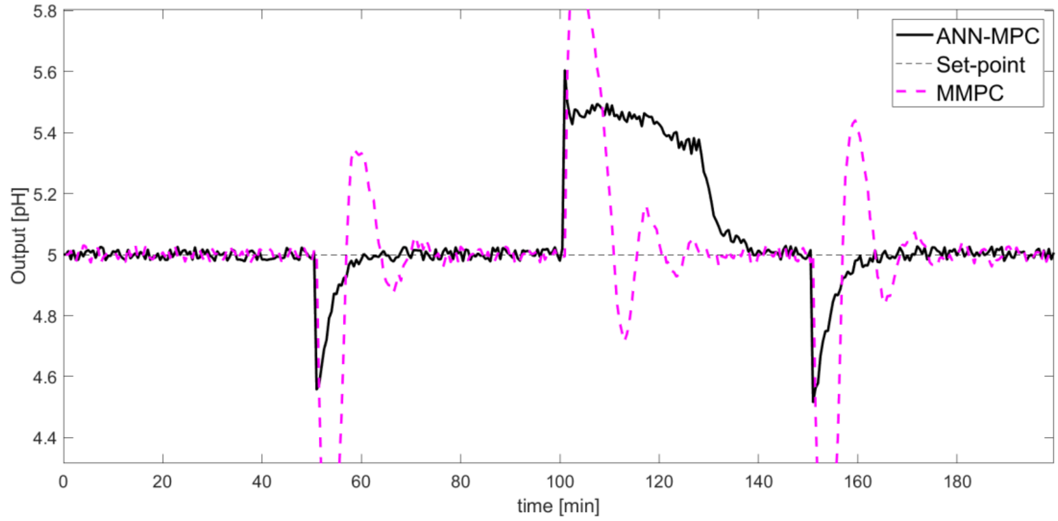

Figure 3. Disturbance rejection test

3.3. Set-point tracking with disturbance.

The test in 3.1 is repeated, but there is a disturbance occurring at 10 minutes and staying throughout the step changes. The results are quite like the stepping without disturbance. There is an additional problem for the ANN-MPC and that is the slow settling into one model that also occurred in section 3.2. This could be improved by increasing the convergence parameter, K, in the Bayesian weighting. That however has a similar effect to reducing the integral time in a PID-controller. There is also an increased time for settling and increased overshoots as well. In the case of surrounding a pH of 6, neither ANN-MPC nor MMPC are settling.

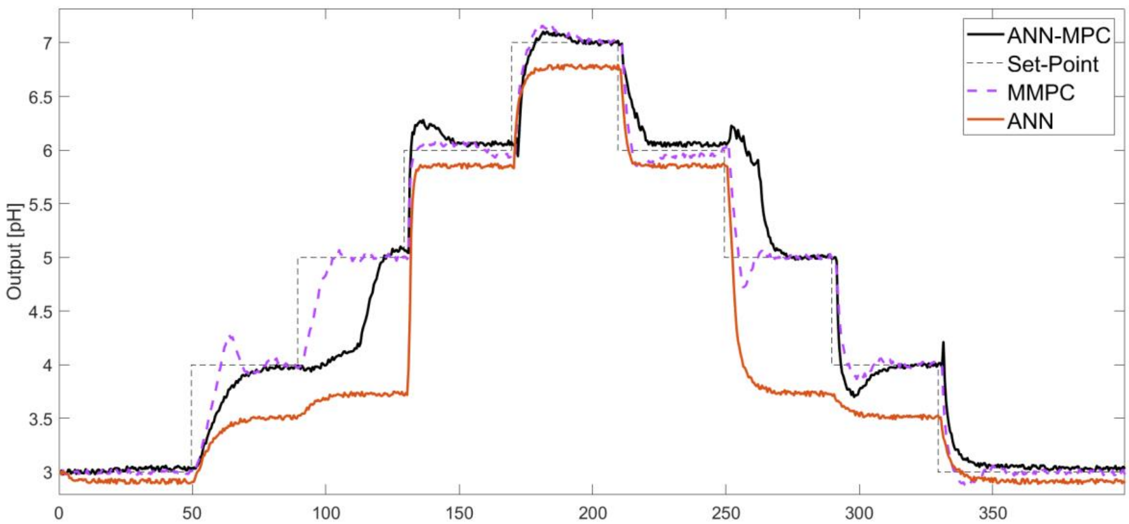

Figure 4. Set-point tracking while a constant disturbance is present.

3.4. Computational time

The major reason for moving away from the MPC is the optimization that must be carried out. By utilizing the ANN we do not have to set up the optimization calculation in this case as well as doing ANN modelling rather than the physical models in [10,12]. Though, we still have to setup the Baysesian weighting (3-6) but that is of less complexity than the optimization. Also, by removing the optimization needed for MPC, the average time for the control calculation for ANN-MPC is 0.15 seconds irrespective of the three cases, while the MMPC takes 0.28 seconds on average.

4. Conclusions

The proposed Bayesian ANN-MPC approached has been simulated to maintain pH levels for references tracking, disturbance rejection and reference tracking with disturbances rejection. The proposed technique is numerically compared to the existing methodologies such as ANN and MMPC in performing possible condition in real-life such reference tracking and disturbances rejection. The numerical simulation of maintaining pH for Phosphoric acid (H_3PO_4) and Calcium hydroxide ($Ca(OH)_2$) system shows that the proposed Bayesian ANN-MPC methodology out performs of ANN and MMPC. It can be remarked that the proposed ANN-MPC approach can handle the problem of overfitting that occurs with just single ANN without having to retraining when new uncertainties occur. The major problem with the proposed method is that it on occasion can be quite slow in "selecting" the best model thus creating a slow response. Thus, further studies would deal with improving the convergence of the Bayesian weighting.

References

1. Ying, X., 2019, February. An overview of overfitting and its solutions. In *Journal of physics: Conference series* (Vol. 1168, p. 022022). IOP Publishing.
2. Draeger, A., Engell, S. and Ranke, H., 1995. Model predictive control using neural networks. *IEEE Control Systems Magazine*, *15*(5), pp.61-66.
3. Patwardhan, R.S., Lakshminarayanan, S. and Shah, S.L., 1998. Constrained nonlinear MPC using Hammerstein and Wiener models: PLS framework. *AIChE Journal*, *44*(7), pp.1611-1622.
4. Lawryńczuk, M., 2010. Computationally efficient nonlinear predictive control based on neural Wiener models. *Neurocomputing*, *74*(1-3), pp.401-417..
5. Sena, H.J., da Silva, F.V. and Fileti, A.M.F., 2021. ANN model adaptation algorithm based on extended Kalman filter applied to pH control using MPC. *Journal of Process Control*, *102,* pp.15-23.
6. Durairaj, S., Xavier, J., Patnaik, S.K. and Panda, R.C., 2023. Deep Learning Based System Identification and Nonlinear Model Predictive Control of pH Neutralization Process. *Industrial & Engineering Chemistry Research*, *62*(33), pp.13061-13080.
7. Tharakan, L.G., Benny, A., Jaffar, N.E. and Jaleel, J.A., 2013, March. Neural network based pH control of a weak acid—Strong base system. In *2013 International Mutli-Conference on Automation, Computing, Communication, Control and Compressed Sensing (iMac4s)* (pp. 674-679). IEEE.
8. Pua, Z.Y., Hermansson, A.W. and Lim, C.H., 2022, October. ANN based Multi Model Predictive Control for pH-Control. In *IOP Conference Series: Materials Science and Engineering* (Vol. 1257, No. 1, p. 012035).
9. Gustafsson, T.K., Skrifvars, B.O., Sandström, K.V. and Waller, K.V., 1995. Modeling of pH for control. *Industrial & Engineering Chemistry Research*, *34*(5), pp.820-827.
10. Hermansson, A.W. and Syafiie, S., 2020, April. Offset-free control of a pH system using Multiple Model Predictive Control. In *IOP Conference Series: Materials Science and Engineering* (Vol. 778, No. 1, p. 012072). IOP Publishing.
11. Aufderheide, B. and Bequette, B.W., 2003. Extension of dynamic matrix control to multiple models. *Computers & chemical engineering*, *27*(8-9), pp.1079-1096.
12. Hermansson, A.W., Syafiie, S. and Noor, S.M., 2010, December. Multiple model predictive control of nonlinear pH neutralization system. In *2010 IEEE International Conference on Industrial Engineering and Engineering Management* (pp. 301-304). IEEE.