JOURNAL OF ENGINEERING SCIENCES
ЖУРНАЛ ІНЖЕНЕРНИХ НАУК
ЖУРНАЛ ИНЖЕНЕРНЫХ НАУК

Web site: http://jes.sumdu.edu.ua

DOI: 10.21272/jes.2018.5(2).e5 Volume 5, Issue 2 (2018)

UDC 681.5.09

# Kalman Filter Based Controlled Online System Identification

Ganesh E. N.

Saveetha Engineering College, Kuthambakkam, 600124 Tamil Nadu, Chennai, India

**Article info:**

| | |
|---|---|
| Paper received: | July 13, 2018 |
| The final version of the paper received: | September 24, 2018 |
| Paper accepted online: | October 3, 2018 |

***Corresponding Author's Address:**
enganesh50@gmail.com

**Abstract.** In the development of model predictive controllers a significant amount of time and effort is necessary for the development of the empirical control models. Even if on-line measurements are available, the control models have to be estimated carefully. The payback time of a model predictive controller could be significantly reduced, if a common identification tool would be available which could be introduced in a control scheme right away. In this work it was developed a control system which consists of a neural network (NN) with external recurrence only, whose parameters are adjusted by the extended Kalman filter in real-time. The output of the neural network is used in a control loop to study its accuracy in a control loop. At the moment this control loop is a NN-model based minimum variance controller. The on-line system identification with controller was tested on a simulation of a fed-batch penicillin production process to understand its behaviour in a complex environment. On every signal process and measurements noise was applied. Even though the NN was never trained before, the controller did not diverge. Although it seemed like the on-line prediction of the NN was quite accurate, the real process was not learned yet. This was checked by simulating the process with the NN obtained at the end of the batch. Nevertheless the process was maintained under control near the wanted set-points. These results show a promising start for a model predictive controller using an on-line system identification method, which could greatly reduce implementation times.

**Keywords:** Kalman filter, neural network, on-line training, variance control.

## 1 Introduction

Most companies have limited resources as most of the large central research departments have shrunken down. Advanced control projects have to compete with other cost saving projects and therefore need to have a typical payback time of two years. Once a control system is introduced, it has to be maintained as the process configuration or process conditions can change willingly or unwanted, for example catalyst decoking. If it would be possible to have a general control tool with self-tuning capabilities for system identification and control, implementation time and thus payback times could be greatly reduced.

The proposed control and system identification system consist of a neural network with external recurrence, whose weights are adjusted by the extended Kalman filter. The neural network's prediction is fed to a second extended Kalman filter which tries to obtain the set-point at the next sampling point. This system identification scheme and control structure can be seen as an adaptive non-linear minimum variance control (Astrom and Wittenmark, 1984).

## 2 Research Methodology

### 2.1 System identification

Neural networks are known to be non-linear fitters in a certain domain. A neural network with external recurrence is normally sufficient for chemical processes as they show slow dynamics compared with electrical (Haykin, 1999).

Various neural network configurations can be seen as the non-linear correspondent of known linear models such as the models NARX, ARMAX, CARMA and state space models (Haykin, 1999; Rivals and Personnaz, 1998).

The neural network weights are adapted by the extended Kalman filter, which is implemented as the MEKA algorithm as first introduced by Shah and Palmiero in 1990, as well as later by Puskoris and Feldkamp in 1991 as the decoupled Kalman filter algorithm. In this case every neuron has its own extended Kalman filter (Figure 1).

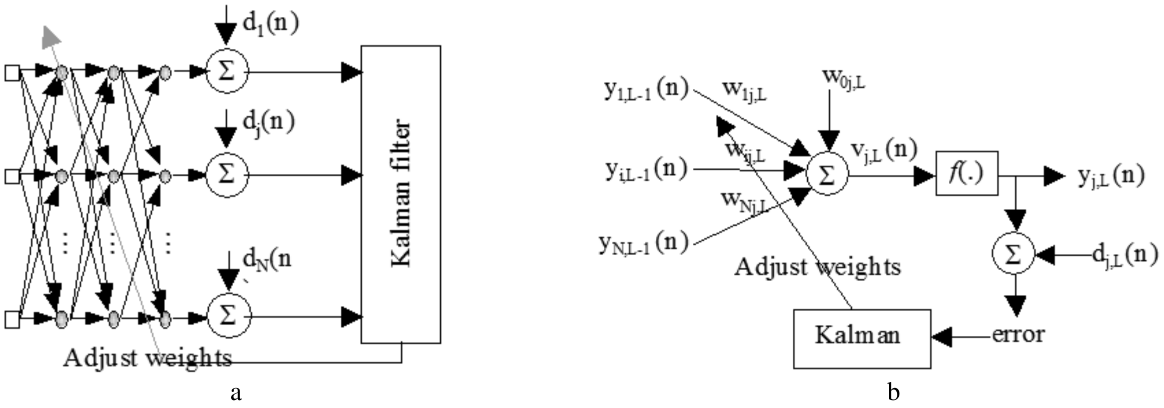

Figure 1 – Global extended Kalman Filter (a) and non-linear Kalman filtering per neuron (b)

The network weights are updated in real-time by a Multiple Extended Kalman filter algorithm to account for changes in the process, whose implementation can be found in Scheffer et al. in 2000–2001.

New development have taken place in improving de Kalman filter by replacing the derivatives for mean and variance calculations (Julier and Uhlmanm, 1997) and applying this concept to neural networks (Wan and van der Merwe, 2000).

## 2.2 The control system

The estimate of the recurrent neural network is fed to an extended Kalman filter to estimate the controller parameters or directly the manipulated variables. Here, the latter approach is chosen and the manipulated variable is directly estimated by the following dynamical system:

$$m(k+1) = m(k) + w(k); \qquad (1)$$

$$y_c(k) = y_{ann,c}(k) + v(k), \qquad (2)$$

where $m$ is the manipulated variable; $y_c$ – the controlled variable; $w$ and $v$ are variables with a Gaussian distribution of $(0, Q)$ and $(0, R)$ respectively.

The measurement $d$ of the controlled variable is the desired set-point of the controlled variable. The manipulated variable is one of the inputs of the recurrent neural network. In the application of the Kalman filter, the observation equation has to be linearised every sampling instance. Thus the derivative of the controlled variable to the manipulated variable has to be calculated, which is the derivative to the recurrent neural network. The derivative of a neural network can be calculated by applying the chain rule, which results for a two layer neural network in:

$$\frac{dy_{k,j}}{dy_{k-2,h}} = f'(v_{k,j})\cdot\sum_{i=1}^{N_{k-1}}\left[w_{k,ji}\cdot w_{k-1,ih}f'(v_{k-1,i})\right]. \qquad (3)$$

The controlled variable can now be updated by the Kalman filter:

$$m(k) = m(k) + K(k)\cdot\left[y_{c,setpoint}(k) - y_{ann,c}(k)\right]. \qquad (4)$$

## 2.3 The penicillin production process

Fed-batch processing is a typical example of a process exhibiting non-linear process dynamics. Especially, biochemical processes are known to have a lot of interaction between their state variables and are sensible to minor changes in pH, dissolved oxygen concentration, and temperature due to the sensitivity of the biochemical catalysts. PID control behaves well in case of continuous processing but in batch processing the control parameters will never maintain optimal values due to the changing process conditions. Therefore this seems to be a challenging study case as it exhibits non-linear process dynamics and a need for a self-tuning controller (Figure 2).

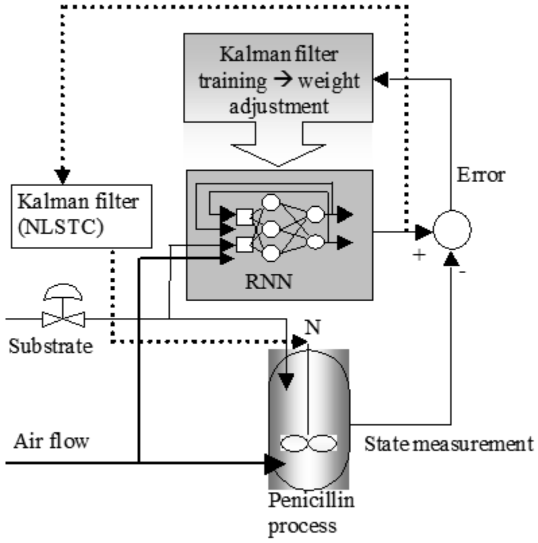

Figure 2 – The proposed non-linear self-tuning controller scheme

The emphasis is put on the production phase and not on the growing phase where an optimal feeding strategy is essential in obtaining a high concentration of penicillin. But it is essential to keep the dissolved oxygen concentration above 30 % to ensure life conditions to the fungi. In this work the feeding strategy determined by Rodrigues in 1999 is used and the control objective is to maintain the dissolved oxygen concentration at about 55 %. The dissolved oxygen concentration is controlled by manipulating the rotation speed through the mentioned non-linear self-tuning controller.

An essential part of the non-linear self-tuning controller is the recurrent neural network identification. Three input and four state variables were taken to identify the process and are the substrate feed flow, the rotation speed and the air flow as input variables and the bio-mass concentration, the substrate concentration, the penicillin concentration and the dissolved oxygen concentration as state variables. The mentioned Kalman filter algorithm will be compared to the standard backpropagation algorithm.

## 3 Results

The system identification is an important part of the control structure. It is necessary to have a good prediction of the dissolved oxygen concentration of the next measurement as this is used by the minimum variance control. Therefore the on-line prediction was studied also without the control structure to understand its prediction capabilities. The standard back-propagation algorithm (SBP) is one of the few algorithms was used as a comparison as it is one of the few other recurrent algorithms. The parameters of the standard back propagation and the Kalman filter algorithms were tuned and a selection was made by ranking them on the training error or on the simulation error of all the state variables as mentioned in the former paragraph. The simulation error was obtained by simulating again after the on-line training, which is the showing of every data-point only one time to the neural network. We would like to note that it was also tried to use the standard backpropgation algorithm with momentum, but that did not result in smaller errors than with the standard backpropagation algorithm. The results are presented in Figures 3–6.

From the Figures 3 and 4 it can be seen the known fact, that the best training errors is no assurance for a good generalization error and is clearly seen the larger discrepancy between the data and the prediction in the simulation with the one-time trained neural network. But it is interesting to see that the usage of the Kalman filter algorithm results indeed in a better general learning. Additionally it can be seen that the simplification of the global extended Kalman filter to multiple local Kalman filters in the MEKA algorithm penalizes the general learning. Additionally, the MEKA algorithm shows more instability in the simulation (Figure 4).

If we turn to the tuning of the parameters, which lead to the best simulation error (Figures 5, 6) the situation becomes even more pronounced. During the training the prediction of the dissolved oxygen concentration with the backpropagation algorithm would not be suitable for control as the concentration prediction during the prediction is not good enough.

It is interesting to see that the instability in the MEKA algorithm has been transferred from the simulations error (Figure 4) to the training error for this case (Figure 5). Still the MEKA algorithm's prediction during the training follows the real concentration.

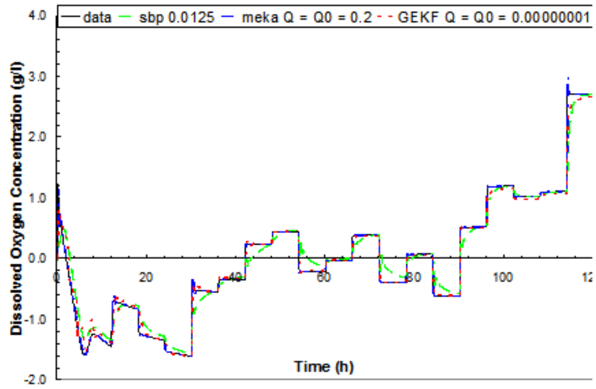

Figure 3 – Dissolved oxygen prediction of NN during the on-line training (best training error and normalized data)

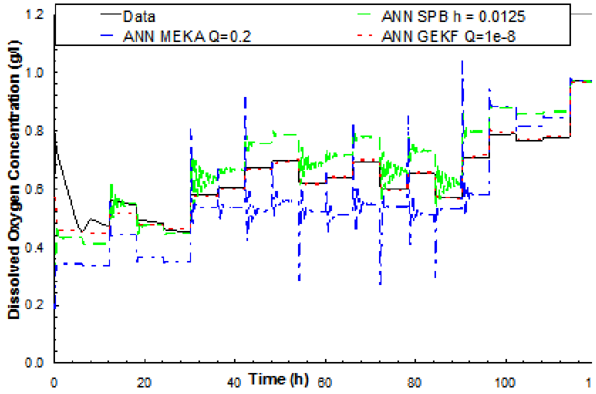

Figure 4 – Simulation of the same dissolved oxygen data with the on-line trained NN (best training error)

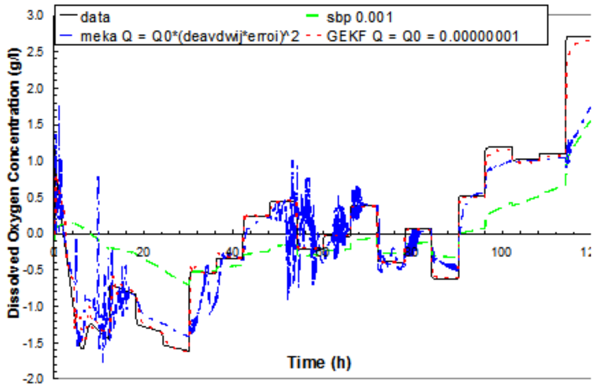

Figure 5 – Dissolved oxygen prediction of NN during the on-line training (best simulation error and normalized data)

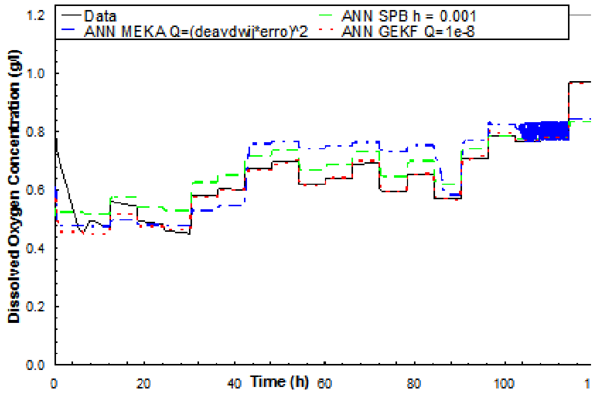

Figure 6 – Simulation of the same dissolved oxygen data with the on-line trained NN (best simulation error)

The GEKF algorithm outperforms in this case very much both algorithms. Therefore there is encouragement to implement the unscented Kalman filter (Julier et al., 1997) also, as this is an improvement over the extended Kalman filter.

In Table 1 it is shown the errors for both cases. The error summed over all the state variables (bio-mass concentration, the substrate concentration, the penicillin concentration and the dissolved oxygen concentration. From the table, it seems that the MEKA algorithm performs better than the GEKF, but that is not true due to the instabilities as shown in Figures 4, 5.

In Table 2 it is shown the calculation times of the different algorithms. Although it seems that the GEKF has a much higher computational costs, it still means that the calculation per data point is done in less than 1 second. The data is probably not faster available from the on-line measurements and therefore the GEKF would be a good candidate also for on-line control scheme or even be the favourite on-line neural network training algorithm.

Table 1 – Training errors for the best training and best simulation cases of the different algorithms

| Algorithm | SBP | SBP | MEKA | MEKA | GEKF | GEKF |
|---|---|---|---|---|---|---|
| Parameter | Best training | Best simulation | Best simulation | Best training | Best simulation | Best training |
| Quality, $Q$ | $1.25 \cdot 10^{-2}$ | $1.0 \cdot 10^{-3}$ | $(error \cdot d\_error/dw_{ij})^2$ | 0.20 | $1.0 \cdot 10^{-8}$ | $1.0 \cdot 10^{-8}$ |
| Relative error | $5.09 \cdot 10^{6}$ | $2.34 \cdot 10^{6}$ | $1.95 \cdot 10^{5}$ | $2.24 \cdot 10^{5}$ | $1.35 \cdot 10^{6}$ | $7.39 \cdot 10^{5}$ |

Table 2 – Calculation times of the different algorithms for 1 200 data points

| Algorithm | SPB | MEKA | GEKF |
|---|---|---|---|
| Calculation time, s | 0.08 | 27 | 409 |

In Figures 7 and 8 it is shown the minimum variance control of the dissolved oxygen concentration with the on-line training of the recurrent neural network at the same time. For the moment the MEKA algorithm has been used and therefore it can still be gained from using the GEKF algorithm. It can be clearly seen that the minimum variance control is better with the MEKA algorithm. The Kalman filter training algorithm assure that the neural network training is better which results in the much smaller deviations at the end of the batch run.

## 4 Conclusions

The on-line training of recurrent neural networks should be done preferably with extended Kalman filter algorithms. Localizing the Kalman filter to the neuron level (MEKA) affects the learning of the neural network and therefore the implementation of the unscented Kalman filter algorithm could lead even to further improvements for on-line training of neural networks. However, the simulations show that not always a good generalization is obtained. The application of the minimum variance controller with the on-line training of the neural network showed that control could be obtained on the neural networks prediction without divergence of the algorithm. Therefore, implementation or payback time could be reduced by applying on-line training.

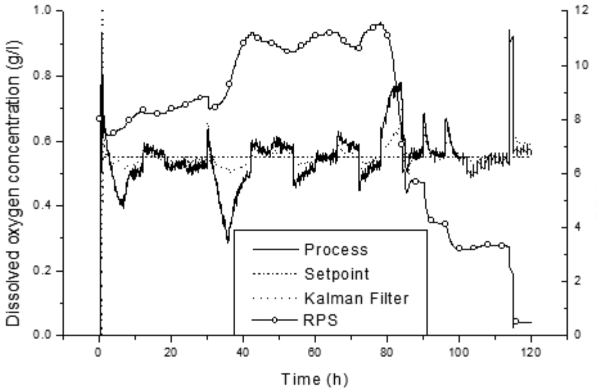

Figure 7 – Estimation and minimum variance control of the dissolved oxygen concentration concentration (the recurrent neural network is trained with the MEKA filter algorithm)

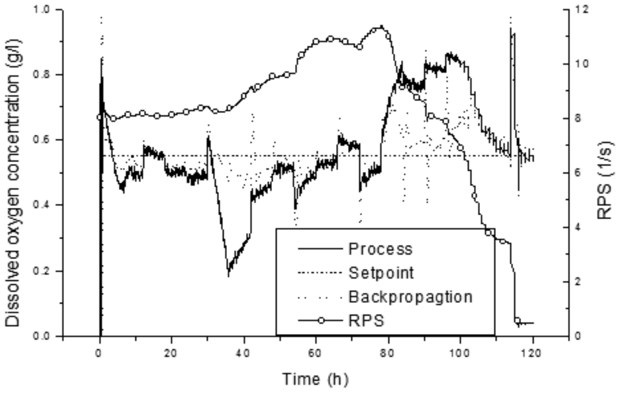

Figure 8 – Estimation and minimum variance control of the dissolved oxygen concentration concentration (the recurrent neural network is trained with the backpropagation algorithm)

## References

1. Astrom, K. J., & Wittenmark, B. (1984). *Computer Controlled Systems*. Prentice-Hall Inc., New Jersey, USA.
2. Haykin, S. (1999). *Neural networks: a comprehensive foundation*. Prentice-Hall Inc., New Jersey, USA.
3. Julier, S., & Uhlmann, J. K. (1997). A new extension of the Kalman filter to nonlinear systems. *Proceedings of the 11th International Symposium on Aerospace/Defence Sensing, Simulation and Controls*.
4. Puskorius, G. V., & Feldkamp, L. A. (1991). Decoupled extended Kalman filter training of feedforward layered networks. *Proceedings of the International Joint Conference on Neural Networks*.
5. Rivals, I., & Personnaz, L. (1998). A recursive algorithm based on the extended Kalman filter for the training of feedforward neural networks. *Neurocomputing*, Vol. 20(1–3), pp. 279–294.
6. Rodrigues, J. A. D., & Filho, R. M. (1999). Production optimisation with operating constraints for a fed-batch reactor with DMC predictive control. *Chemical Engineering Science*, Vol. 54(13–14), pp. 2745–2751.
7. Shah, S., & Palmieri, F. (1990). MEKA – A fast, local algorithm for training feedforward neural networks. *Proceedings of the International Joint Conference on Neural Networks*, pp III-41–45.
8. Scheffer, R., & Filho, R. M. (2000). Training a Recurrent Neural Network by the Extended Kalman Filter as an Identification Tool. *Escape-10 Symposium Proceedings*, pp. 223–228.
9. Scheffer, R., Filho, R. M. (2001). Process identification of a fed-batch penicillin production process - training with the extended kalman filter. *Application of Neural Network and Other Learning Technologies in Process Engineering*.
10. Wan, E. A., van der Merwe, R. (2000). The unscented Kalman filter for nonlinear estimation. *Proceedings of the Symposium 2000 on Adaptive Systems for Signal Processing, Communication adn Control (AS-SPCC)*, pp. 153–159.

## Контрольована онлайн-система ідентифікації на основі фільтра Калмана

Ганеш Е. Н.

Інженерний коледж ім. Савеета, м. Кутгембаккем, 600124, Індія

**Анотація.** Для розроблення прогностичних моделей контролерів витрачається відносно велика кількість часу та зусиль. Час окупності контролера з функцією інтелектуального прогнозування можна суттєво зменшити за наявності загальний інструменту ідентифікації, введеного до схеми контролю. У роботі була розроблена система керування, яка складається з нейронної мережі (НМ) з виключно зовнішнім повторенням, параметри якої регулюються розширеним фільтром Калмана у режимі реального часу. Вихід нейронної мережі використовується в контрольній ланці для вивчення точності контуру керування. На даний момент цей контур є контролером мінімальної дисперсії на основі НМ-моделі. Ідентифікація он-лайн системи з контролером протестована шляхом моделювання процесу виробництва пеніциліну з метою розуміння поведінки у складному середовищі. На кожному сигнальному процесі та вимірюванні застосовувався шум. Незважаючи на те, що НМ ніколи раніше не навчалась, розходження контролера не спостерігалось. Хоча онлайн-прогноз НМ був достатньо точним, реальний процес залишається досі не вивченим. Він був перевірений імітацією процесу із застосуванням НМ для отриманого кінцевого продукту. Тим не менш, цей процес тримався під контролем поруч із встановленими «розумними» точками. Отримані результати свідчать про перспективні спроби моделювання інтелектуальних контролерів із використанням методу он-лайн ідентифікації, що значно вкорочує час реалізації.

**Ключові слова:** фільтр Калмана, нейронна мережа, онлайн-навчання, варіативний контроль.