Materials Today: Proceedings xxx (xxxx) xxx

Contents lists available at ScienceDirect

Materials Today: Proceedings

journal homepage: www.elsevier.com/locate/matpr

Machine learning and its impact on control systems: A review

Prabhat Dev [a], Siddharth Jain [b], Pawan Kumar Arora [b], Harish Kumar [c,*]

[a] *National Institute of Technology Delhi, Delhi, India*
[b] *Galgotias College of Engineering and Technology, Greater Noida, G.B.Nagar, India*
[c] *Meerut Institute of Engineering and Technology, Meerut, India*

ARTICLE INFO

Article history:
Received 28 November 2020
Received in revised form 29 January 2021
Accepted 9 February 2021
Available online xxxx

Keywords:
Control system
PID
Neural network
RNN
LSTM
System identification

ABSTRACT

Control systems play a vital role in our day to day lives, from mobile phones to autopilot systems that precisely navigate airplanes, they can be found anywhere. The primary role of any control system is to achieve a set point defined by the user and provide stability to the system while doing so. There are various approaches that are used for the design of control systems such as PID (Proportional, Integral and Derivative control) algorithms, Fuzzy logic controller, Neural Network controllers etc. Machine learning (ML) is a key tool in analysing time series data and can be used to predict the future states of any dynamic system, however, sufficient historic data is required. This key feature of ML based predictive algorithms can be employed in modelling a wide range of non-linear dynamic systems for ensuring system stability as well as seeking continuous improvement in system response. This paper gives a brief introduction of conventional methods used in the design of control systems with focus on Neural Network based control strategies which employ Recurrent Neural Network (RNN) and Long Short Term Memory (LSTM) networks for system identification and predictive control of dynamic processes.

Selection and peer-review under responsibility of the scientific committee of the 3rd International Conference on Computational and Experimental Methods in Mechanical Engineering.

1. Introduction

Control systems can be found anywhere around us. They have played a crucial role in development of technology in the industrial revolution. Every automated system we see is equipped with a control system. The primary goal any control system is to eliminate the need for constant user input. Control theory is a domain of Mathematics and Engineering which deals with controlling various dynamical systems to reach a desired set point defined by user also knows as Set point Value (SV). This is achieved by altering one or more inputs of the system that has an influence on the output and simultaneously keeping track of the current state of the system, also known as the Process Value (PV). The algorithm which is designed to do this operation is known as control algorithm. The control algorithm constantly keeps track of the error (SV-PV) before making any changes to the input to the system. Typical examples of control systems include Air conditioners, Auto pilot systems, maintaining chemical compositions etc. Control theory provides a systematic approach to designing closed loop systems [1]. There have been many methods that have been used in the past for the design of control system. Rapid advances in digital system technology have influenced the control design options to a great extent [2].Fig. 1Fig. 2Fig. 3Fig. 4Fig. 5

2. Conventional control and optimization techniques

More than 90% of the control systems use PID (Proportional, Integral and Derivative control) in their control systems. PID controllers can handle multiple DOF systems where there us more than one input and output. The PID algorithm works based on the following governing equation

$$u(t) = k_p e(t) + k_i \int_o^t e(t)dt + k_d \frac{de(t)}{dt}$$

Where $u(t)$ is the control signal, $e(t)$ is the error term (SV-PV) and k_p, k_i *and* k_d are tuning parameters which influence the dynamics of the control

PID controllers may be suited for most of the industrial applications but they are less efficient when it comes to handling non-linear systems where the system might behave in an unpredicted

* Corresponding author.
E-mail address: harishkumar@nitdelhi.ac.in (H. Kumar).

https://doi.org/10.1016/j.matpr.2021.02.281
2214-7853/
Selection and peer-review under responsibility of the scientific committee of the 3rd International Conference on Computational and Experimental Methods in Mechanical Engineering.

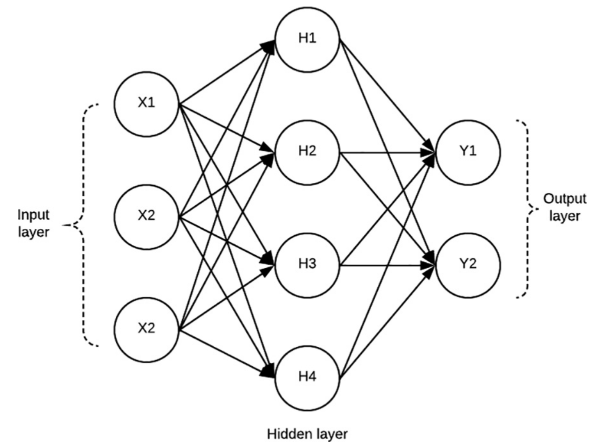

Fig. 1. A simple feed forward neural network.

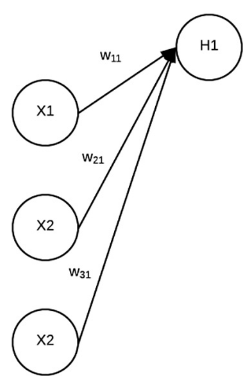

Fig. 2. Activation in single neuron.

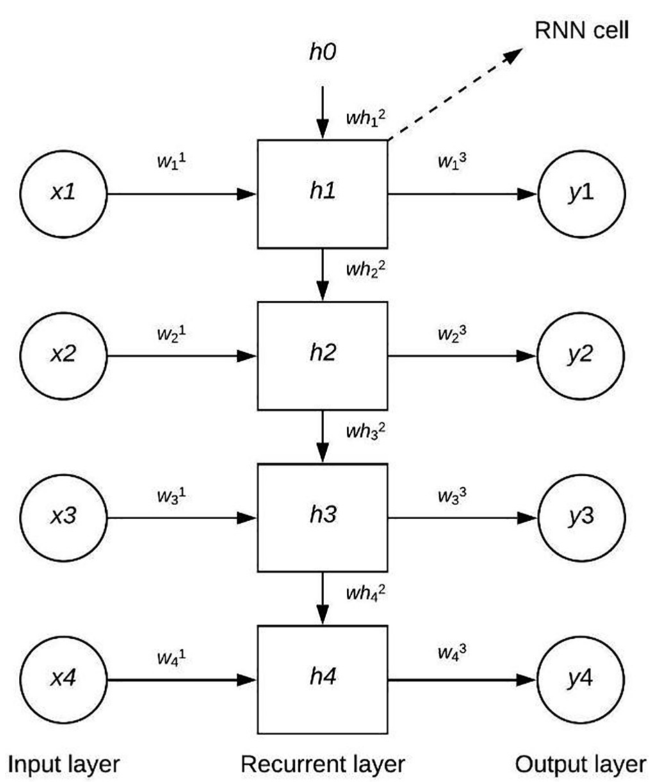

Fig. 3. Network architecture of Recurrent Neural Network.

manner or when external disturbances are involved. This is where various optimization methods comes in, which alter the tuning parameters in such a way that the system would be able to handle uncertainties in an efficient way. There have been various methods which are used for the optimizing the gain parameters for obtaining better performance such as Particle Swarm Optimization (PSO) [3] technique, Genetic Algorithm (GA) optimization [4]. These optimization techniques definitely provide better results but when it to comes to applications where advanced control is required such as design of Biomechanical systems, Chemical reaction plants, Autonomous driving etc., conventional controllers fail to deliver efficiently. This brought forth the need to develop control systems that are precise and at the same time better adapting to the changes in the dynamics of the system.

2.1. Machine learning

The development of modern control systems integrated with microcontrollers not only allowed users to have a better interface but also helped in collecting large amount of data about how the system responds to various inputs and disturbances. This allowed researchers to better understand the dynamics of the system and to develop systems that could predict the behaviour of the system. This led to the integration of Machine Learning into control systems. Machine Learning can be employed in wide range of applications such as Image recognition, security systems, autonomous vehicles, Natural Language Processing (NLP), Genetics, medical diagnosis, robotics, classification tasks, stock exchange etc. [5–9]. It forms the base framework on which Artificial Intelligence (AI) is built on. Hence it is no wonder it can be used in the design of intelligent control systems. There are many ways to represent a system model among those Transfer function models, state-space model and differential equation models are most frequently used in the design of control systems. In the recent development of microprocessors and huge increase in computing power researchers are now focusing on Neural Network models for representing non-linear dynamic systems.

2.2. Neural network

Neural networks can be considered as a computing system inspired by biological neurons. In simple words it can be defined as a black box which takes one or more inputs and gives one or more outputs as defined by the user. A neural network consists of multiple layers of neurons which are interconnected by weights and by adjusting these weights in the right way, the network can be used to represent any complex system. This gives a mathematical relation between the inputs and the output. The process of adjusting the weights is known as training the neural network. There are various optimization methods which are used for the training operation. A simple Feed Forward 3-layer neural network is shown below.

The above shown neural network has an input layer, a hidden layer and an output layer. Consider a single neuron 'H1' in the hidden layer

Where w_{11}, w_{21} and w_{31} are the weights connecting the input layers to the 'H1' neuron. The output this neuron is calculated as

$$h1 = f(x1.w_{11} + x2w_{21} + x3.w_{31})$$

Where f is the activation function of the layer which is usually a non-linear in nature

The entire network can be represented in matrix form. The output of the hidden layer can be represented as

$$H = f_1\left(X.W^1\right)$$

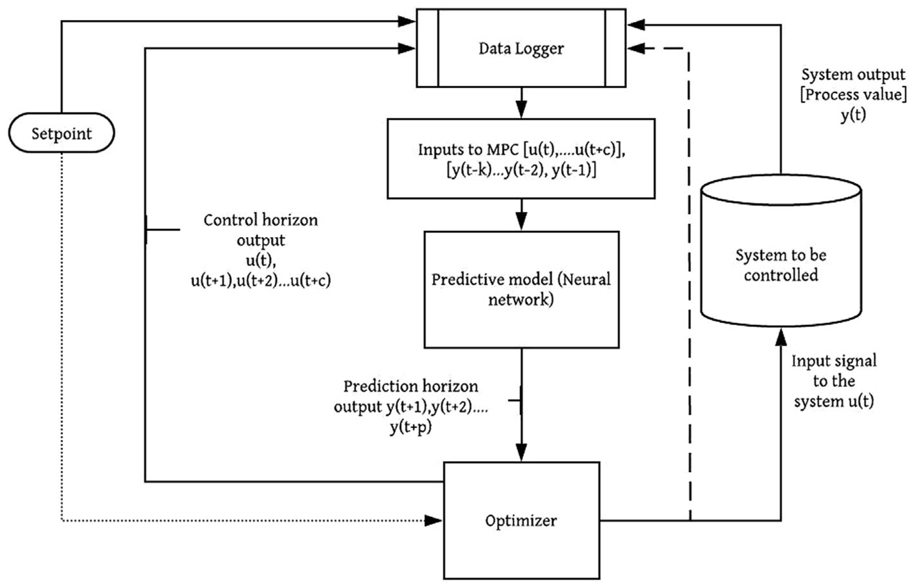

Fig. 4. General layout of Model Predictive Control (MPC).

f_1 is the activation function of hidden layer, X is the matrix containing inputs $x1$, $x2$,$x3$ and W^1 is the weight matrix containing the weights between input and the hidden layer.

Output of the network can be calculated as

$$Y = f_2\left(H.W^2\right)$$

f_2 is the activation function of the output layer and W^2 is the weight matrix containing weights between hidden and the output layer.

2.3. Recurrent neural network (RNN)

RNN networks are very effective in analysing time series data. It can be thought as a neural network in which the input(s) to the network is the output(s) from the previous step combined with other inputs. This way the network has memory about the previous states of the system and can be used to make a prediction about the future states of the system with greater accuracy. Consider the following network

It can be seen from Fig that the network consists of a recurrent layer in which each recurrent cell has two inputs, one from the input layer and the other from the previous state of the system. An RNN can consist of a single or multiple recurrent hidden layer based on the complexity of the problem to be modelled.

The output of a single RNN cell in the above network is calculated as,

$$h1 = f_1\left(x1.w_1^1 + h0.wh_1^2\right)$$

where f_1 is the activation function of the input layer, w_1^1 is the weight between input and recurrent cell, $wh_1^2$is the weight between

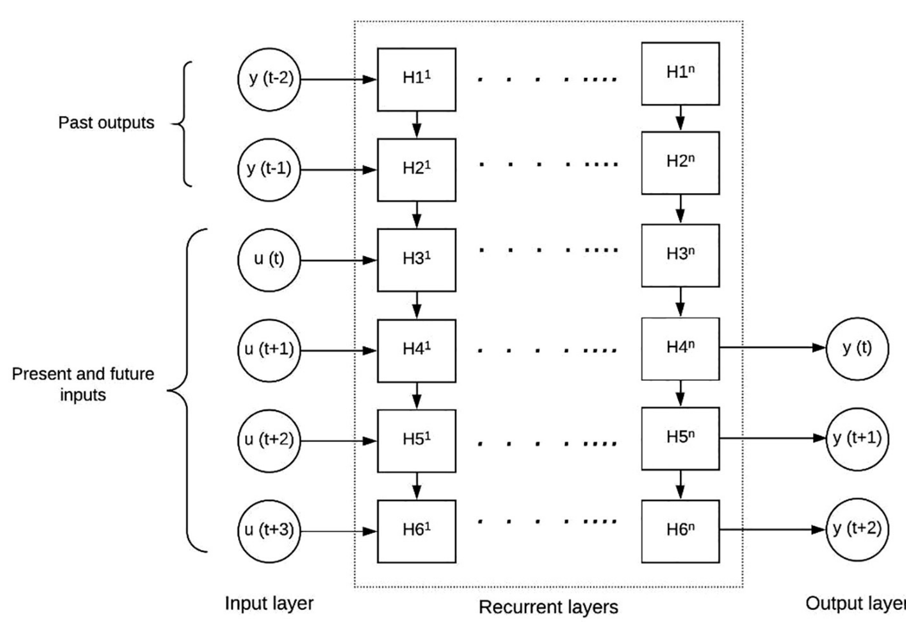

Fig. 5. Predictive model using a Recurrent Neural Network.

each recurrent cells and $h0$ is the initial state. The subscripts in the weight denote the cell from which they originate and the superscripts denote the layer in which they are present.

Considering linear activations in the output layer, the network output is calculated as,

$$y1 = f_2(h1.w_1^3)$$

Where f_2 is the activation function in recurrent layer and w_1^3 is the weight between the recurrent layer and the output layer.

The main drawback of RNN comes while training the network. Unlike normal feedforward networks, in RNN, the network is trained by 'Back propagation through time' since the current output of the network is also dependent on the output from previous timestamps. Updating weights can be quite difficult [10] and there is also the problem of gradient getting stuck at local minima [11]. While training RNN usually there is a problem of vanishing gradient where the gradient becomes small with every epoch and the network doesn't converge. Similarly there's also the problem of exploding gradient where the gradient updates are huge with each epoch. Usually these are rectified by setting gradient limits which are set before training the network. A special type of RNN network called LSTM networks can overcome the gradient problem. Even though training the RNN can be difficult it makes up for the drawback when it comes to mapping. An RNN can map any dynamic system better than feed forward networks [12] which makes it best suited for control system design. In [13], the author has explained how a simple RNN can tackle uncertainties that arise during system operation.

LSTM stands for Long Short Term Memory, which is nothing but a modified RNN with gates. It has 3 gates namely input gate, output gate and forget gate. The input gate decides which feature from the input values should be let through. This is usually decided by using a sigmoid activation function and weightage is given using a tanh activation function. The forget gate decides which information is worth passing on through the next stage. The forget gate takes previous state h_{t-1} and the current input as inputs which is then passed through sigmoid activation to decide if the memory should be disregarded or kept and passed on to the next cell. The output gate works with memory status from the forget gate and the current input, and passes it through sigmoid activation which is then multiplied with one more activation usually tanh to produce the output. LSTM networks thus possess extraordinary mapping ability and can also tackle vanishing gradient problems which conventional RNN can't do. This makes them well suited for control system applications.

2.4. System identification

One of the crucial steps in the design of control systems is system identification. It is important to represent the process that needs to be controlled using some form of mathematical expression which relates the input and output. Based on this mathematical model control parameters are developed. System identification is a technique used for developing a mathematical model to represent a process or a system. It also helps in the study of how the system behaves under various conditions and how external disturbances affect the system response. As discussed above, researchers are now focusing more on exploiting the power of neural networks for control applications. This is because of the network's ability to model a wider range of complex nonlinear systems which otherwise could be difficult to represent using differential equations. This is where neural networks come in handy with their exceptional nonlinear mapping ability, especially Recurrent Neural Networks. Researchers in the past have proved this ability of a neural network to identify and map nonlinear systems [14,15]. RNN can approximate complex nonlinear systems with reasonable accuracy while taking short time to develop [16]. Also an RNN with smaller network size has approximation capability of larger feed forward networks [17] which makes it more suited to be used for system identification. In [18], the author discusses the qualities a neural network should possess in order to obtain servo level control.

Generating training data is a crucial step in developing a neural network model. Training data consist of input signals and corresponding output signals from the plant. This is usually done by performing an open loop test on the plant where there is no control governing the plant. In [19], it is shown that data obtained from the plant under some form of control (P/PI/PID) gives better training data than open loop data but there isn't sufficient research to come to a conclusion. The efficient way would be to experiment with network architecture and training data to ultimately obtain the least complex model to represent the system. The goal of system identification is not only to generate a model for the system, but also to do it in an efficient way such that it can be used for multiple systems with minimal changes to the network. Cost of model development along with testing and validation requires a great deal of resources when it comes to industrial applications [20]. Improvement in the identification technique reduces testing time [21], which saves a lot of resources when it comes to industrial implementation. In [22], the author has used a convex-based LSTM network for reducing processing time and increased efficiency. Various factors come into play when it comes to using neural networks for system identification such as sample time, network structure, excitation signal etc., all of which, can affect the network's mapping ability [23]. Haiquan Zhao and Jiashu Zhang have developed a novel method for reducing computational cost using a pipelined functional link artificial recurrent neural network [24] for nonlinear system identification. In [25] an RNN with multi-segment piecewise linear connection weight was used for system identification where each weight in the connection has multiple values dependent on the input to the network. Various iterations have been done in the past in order to exploit the complete potential of neural networks for system identification processes. The main purpose of system identification is to use the identified model for control system development. One such important implementation of the model is the development of predictive control systems.

2.5. Predictive control systems based on neural networks

Model Predictive Control (MPC) is a control logic by which a mathematical model is used to predict the future states of the system and based on these states future set of control signals are optimized for increased control efficiency. It can be considered as solving an optimization problem at each sampling interval (control step) in the future horizon. MPC requires a model which needs to be capable of predicting the future output of the system with available historical data. Since RNN can be used to predict the future response of a dynamic system, it can be used for implementing MPC.

In the above figure u is the control signal, y is the process value and k is the limit set by the user during the development of the prediction model for best performance.

MPC predicts the output of the system as shown above for the time periods $y(t + 1)$, $y(t + 2)\ldots\ldots y(t + p)$. This range of time period ($t = 0$ to $t = p$) for which the system predicts the values is known as the Prediction horizon. Based on this predicted system output, the optimizer compares the values with the set point and generates control signals $u(t)$, $u(t + 1)$, $u(t + 2)\ldots\ldots u(t + c)$. This range of the time period ($t = 0$ to $t = c$) ahead in the future where the control signals are altered is known as Control horizon. Even though more than one control signal value is generated in the control horizon, only one control signal u (t), the control signal at present instant, is fed into the system. This process is repeated all over again at each sampling interval.

Shown above is a predictive model based on RNN with prediction horizon length of $t = 2$ sec, control horizon length of $t = 3$ sec, $k = 2$ and sample time of 1 sec. These parameters depend on the characteristics of plant that needs to be controlled, such as process gain, delay, dead time etc... Other factors such as computational effort required, memory storage capacity of the data logger also come into play while designing the predictive model.

Usually MPC controllers have an offline model, meaning the model is pre-trained to predict the system output for a set of inputs. This may cause problems when the system experiences some unknown disturbances which was not considered during the predictive model development. In order to overcome this issue, Adaptive MPC can be designed which has an online predictive model that gets updated as the plant runs. This requires a lot of computing power as the control system needs to handle both the online model and solve the optimization problem at the same time. Recent development in the electronics industry which has made computing power cheaper than before has enabled many researchers to work on Adaptive MPC. One such model development using neural networks has been discussed and implemented in [26].

The efficiency of the MPC model primarily depends on two factors: The accuracy of the prediction model [27] and the optimization technique used for updating control signals. Using a neural network as a prediction model can be very efficient as we have discussed the network's ability to approximate any nonlinear system (system identification) with higher accuracy. Moreover computational expense depends upon the complexity of the network not controlling the horizon's length. Hence larger control horizons are possible with neural networks [28]. The efficiency of the optimization method impacts the feasibility of using MPC for process control [29]. Wide range of complex nonlinear control problems can be solved by constrained optimization [30]. Apart from using neural networks for system identification, it can also be used to solve optimization problems. YunpengPan, and Jun Wang have used two neural networks, one for system identification and the other one for optimization of control signals [31]. This explains a lot about the flexibility of a neural network model.

3. Future scope

Development in the electronics industry and subsequent increase in the processing capability of computers have carved a pathway for development of more sophisticated methods of process control. J. Brian Froisy in 1994 mentioned in his review paper [32] that the use of nonlinear models and neural networks will continue as computers become more powerful. In [33] the authors have explained how control systems constantly evolve according to the needs in the marketplace. We are seeing this happen right now with more research now being focused on the development of automated and intelligent control systems. Machine learning is at the core of this development phase. Machine learning integrated control systems has applications in Robotics, Bionics, Material science, Traffic control, autonomous vehicles, manufacturing, surgical systems, power plants and many more.

4. Conclusion

Machine learning and its application in control systems have been discussed in this review paper with more focus towards system identification, neural network modelling and how it can be used in designing predictive control systems. Various challenges that arise during the development of control systems using neural networks and how many researchers have found solutions to work around those issues were also discussed. The field of electronics and computer science is evolving at a far greater pace than ever. This opens a wide range of opportunities for researchers to experiment with various ideas and develop novel methods for effectively solving control problems. We are just beginning to explore the capabilities of machine learning and artificial intelligence. In the upcoming years, the development will be rapid which will carve a pathway for designing state of the art control systems. Such systems will application in an enormous number of fields where precise control is required and safety is crucial.

CRediT authorship contribution statement

Prabhat Dev: Investigation. **Siddharth Jain:** Methodology. **Pawan Kumar Arora:** Writing - original draft. **Harish Kumar:** Writing - review & editing.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

Authors are thankful to Hon. Director, National Institute of Technology Delhi, Delhi, India.

References

[1] T. Abdelzaher, Y. Diao, J.L. Hellerstein, C. Lu, X. Zhu, Introduction to Control Theory And Its Application to Computing Systems, in: Z. Liu, C.H. Xia (Eds.), Performance Modeling and Engineering, Springer, Boston, MA, 2008.
[2] S. Simrock, "Control theory", DESY, Hamburg, Germany.
[3] Mahmud IwanSolihin, Lee Fook Tack, Moey Leap Kean, "Tuning of PID controller using Particle Swarm Optimization (PSO)", Proc. Int. Conf. Adv. Sci. Eng. Inf. Technol. 2011.
[4] AndriMirzal, Shinichiro Yoshii, Masashi Furukawa, "PID Parameters Optimization by Using Genetic Algorithm", Graduate School of Information Science and Technology Hokkaido University Sapporo, Japan.
[5] Ian Goodfellow, Yoshua Bengio, Aaron Courville, "Deep Learning", Adaptive computation and machine learning series, MIT press (2017).
[6] I. Kononenko, Machine learning for medical diagnosis: history, state of the art and perspective, Artif. Intell. Med. 23 (1) (2001) 89–109.
[7] Maxwell W. Libbrecht, William Stafford Noble, Machine learning applications in genetics and genomics, Nat. Rev. Genetics 16 (2015) 321.
[8] Sreelekshmy Selvin, R. Vinayakumar, E.A. Gopalakrishnan, Vijay Krishna Menon, K.P. Soman, "Stock price prediction using LSTM, RNN and CNN-sliding window model", International Conference on Advances in Computing, Communications and Informatics (ICACCI), IEEE Xplore (2017).
[9] Weiyu Wang, KengSiau, "Artificial intelligence, machine learning, automation, robotics, future of work and future of humanity: A review and research agenda", J. Database Manage., Volume 30, Issue 1, January-March 2019.
[10] Xiao D. Ji, Babajide O. Familoni, A diagonal recurrent neural network-based hybrid direct adaptive SPSA control system, IEEE Trans. Automatic Control 44 (7) (1999).
[11] G.M. Zenga, X.S. Qinb, L. Hea, G.H. Huanga,b, H.L. Liua, Y.P. Linb, "A neural network predictive control system for paper mill wastewater treatment", G.M. Zeng et al. / Engineering Applications of Artificial Intelligence 16 (2003) 121–129
[12] Tommy W.S. Chow, Yong Fang, A recurrent neural-network-based real-time learning control strategy applying to nonlinear systems with unknown dynamics, IEEE Trans. Ind. Electron. 45 (1) (1998).
[13] Faa-Jeng Lin, Rong-Jong Wai, Wen-Der Chou, Shu-Peng Hsu, Adaptive backstepping control using recurrent neural network for linear induction motor drive, IEEE Trans. Ind. Electron. 49 (1) (2002).
[14] Shou-Ju Li, Ying-Xi Liu, "An Improved Approach to Nonlinear Dynamical System Identification using PID Neural Networks", Freund Publishing House Ltd. International Journal of Nonlinear Sciences and Numerical Simulation 7 (2), ,177-182, 2006.
[15] Xiao-Dong Li, John K.L. Ho, Tommy W.S. Chow, Approximation of dynamical time variant systems by continuous-time recurrent neural networks, IEEE Trans. Circuits Syst.—II: Express Briefs 52 (10) (2005).

[16] A.G. Parlos, K.T. Chong, A.F. Atiya, Application of recurrent multilayer perceptron in modelling of complex process dynamics, IEEE Trans. Neural Netw. 5 (2000) 255–266.
[17] K. Hornik, M. Stinchcombe, H. White, Multilayer feedforward networks are universal approximators, Neural Netw. 2 (5) (1989) 359–366.
[18] Takayuki Yamada, Tetsuro Yabuta, ''Dynamic system identification using neural networks'', IEEE Trans. Syst. Man Cybernetics, VOL. 23, NO. 1, JANUARY/FEBRUARY 1993.
[19] Andreas Draeger, Sebastian Engell, Horst Ranke, ''Model predictive control using neural networks'', IEEE Control Syst., 0272- 1708/95/$04.000, October 1995.
[20] Kwaku O. Temeng, Phillip D. Schnelle, Thomas J. McAvoy, Model predictive control of an industrial packed bed reactor using neural networks, J. Proc. Cont. 5 (1) (1995) 19–27.
[21] S. Joe Qina, Thomas A. Badgwell, ''A survey of industrial model predictive control technology'', Control Eng. Pract. 11 (2003) 733–764, 2002.
[22] Yu Wang, ''A New Concept using LSTM Neural Networks for Dynamic System Identification'', 2017 American Control Conference, Sheraton Seattle Hotel, May 24–26, 2017, Seattle, USA.
[23] S.K. Doherty, J.B. Gomm, D. Williams, Experiment design considerations for non-linear system identification using neural networks, Comput. Chem. Eng. 21 (3) (1997) 327–346.
[24] Haiquan Zhao, Jiashu Zhang, Nonlinear dynamic system identification using pipelined functional link artificial recurrent neural network, Neurocomputing 72 (13–15) (2009) 3046–3054.
[25] Arit Thammano, Phongthep Ruxpakawong, Nonlinear dynamic system identification using recurrent neural network with multi-segment piecewise-linear connection weight, Memetic Comp. 2 (2010) 273–282, https://doi.org/10.1007/s12293-010-0042-7, Springer-Verlag 2010.
[26] Vincent A. Akpan, George D. Hassapis, Nonlinear model identification and adaptive model predictive control using neural networks, ISA Trans. 50 (2011) 177–194.
[27] Dale E. Seborg, Thomas F. Edgar, Duncan A. Mellichamp, Process dynamics and control, Am. Inst. Chem. Eng. (AIChE) (2008).
[28] Bernt M. Akesson, Hannu T. Toivonen, A neural network model predictive controller, J. Process Control 16 (2006) (2006) 937–946.
[29] Zheng Yan, Jun Wang, Model predictive control for tracking of underactuated vessels based on recurrent neural networks, IEEE J. Oceanic Eng. 37 (4) (2012).
[30] Stephen Piche, Bijan Sayyar-Rodsari, Doug Johnson, Arid Mark Gerules, Nonlinear model predictive control using neural networks, IEEE Control Syst. Magaz. 20 (3) (2000) 53–62.
[31] Yunpeng Pan, Jun Wang, Model predictive control of unknown nonlinear dynamical systems based on recurrent neural networks, IEEE Trans. Ind. Electron. 59 (8) (2012).
[32] J. Brian Froisy, Model predictive control: Past, present and future, ISA Trans. 33 (1994) 235–243.
[33] Carlos E. Garcia, David M. Prett, Manfred Morari, ''Model predictive control: theory and practice a survey'', Automatica, Vol. 25, No. 3, pp. 335-338, 1989.