TotalControl

Advanced integrated supervisory and wind turbine control for optimal operation of large Wind Power Plants

Title: Machine learning approaches to wind farm control
Deliverable no.: D1.10

Delivery date: 30.04.2018
Lead beneficiary: DNV
Dissemination level: PU

This project has received funding from the European Union's Horizon 2020 Research and Innovation Programme under grant agreement No. 727680

Author(s) information (alphabetical):		
Name	Organisation	Email
Elizabeth Traiger	Garrad Hassan & Partners Ltd. / DNV GL (DNV)	elizabeth.traiger@dnvgl.com

Acknowledgements/Contributions:		
Name	Name	Name

Document information

Version	Date	Description			
			Prepared by	Reviewed by	Approved by
1	18.10.2018	Machine learning approaches to wind farm control	E. Traiger	E. *Bossanyi*	
2	17.12.2018	-	E. Traiger	J. Meyers	J. Meyers

Definitions

ANN	Artificial Neural Network
CNN	Convolutional Neural Network
GRU	Gated Recurrent Unit Neural Network
GP	Gaussian Process
LSTM	Long Short-Term Memory Network
MDP	Markov Decision Process
MLP	Multi-Layer Perceptron
NFQ	Neural Fitted Q-iteration
Pilco	Probabilistic Inference for Learning Control

TABLE OF CONTENTS

EXECUTIVE SUMMARY

There is an opportunity to autotomize data-driven new control strategies for wind turbines based on existing sensor information to optimize wind farm scale performance. Machine learning (ML) algorithms show promise for overcoming the complexity and stochastic nature of the relevant physics need for the optimization of wind energy converter performance on a project scale. Given the nonlinearity, nonstationary and high dimensionality of the of the wind farm-wide control problem, only a subset of machine learning algorithms are suitable for the task. In a grey-box modelling approach which utilizes the operational data and operating mechanism together, ML can be used to predict the next state in the system through unscented Kalman filters, detect changes that are 'significant' so the controller knows when it should or should not react with convolutional neural networks, and reduce the state space needed for simulations of input producing localized estimates using Gaussian processes. Full black box control algorithms based on data only could be achieved using Recurrent Neural Networks designed for sequence problems. Long Short-Term Memory networks (LSTM) can learn and forecast long sequences capable of learning long-term dependencies. A CNN LSTM architecture which involves using Convolutional Neural Network (CNN) layers for feature extraction on input data combined with LSTMs are very promising. They have the ability to capture both the quick and the slower dynamic components without devaluing each other as noise. This is key for wind farm control where the existence of different time scales is a prominent feature. Unfortunately, real SCADA data comes with incomplete system observations wither missing or erroneous data. Controllers work on live feeds requiring constant valid input. Given this, the unsupervised learning technique of Reinforcement learning (RL) which deals with the question of how an agent can learn optimal strategies through interaction with an external environment is a leading avenue to explore. Q-learning, a temporal difference method, allows procedures in some states to change policy before the values settle which is great given the dynamic stochastic nature of the wind farm control environment. Monte Carlo tree search could be useful. This locally adaptive tree search method reduces the search space from all possible moves to those most likely given the current position, giving performance gains in speed and compute resources. Care must be used as high model errors can be an issue for reinforcement learning from true dynamic models. PILCO (Probabilistic Inference for Learning Control) avoids this problem by including a probabilistic dynamics model in the form of a non-parametric regression GP for learning and modelling the long-term planning and policy search and is useful for data efficient reinforcement learning, in which only a few interactions with the system are available for learning. There is the element of failure in order for the system to learn in any reinforcement learning framework. For wind farm control, the safety and financial consequences of turbine or wind project-wide failure prohibit the training on a real installation. Therefore, a highly realistic wind farm simulator is required for training the control strategy in a simulated environment. The digital training environment must accurately capture the dynamic forces acting on the turbines and resulting wakes coming from changes in the control operating settings.

INTRODUCTION

We want to adapt largely autonomous data-driven new control strategies for wind turbines based on existing sensor information to optimize wind farm scale performance. The task has unique characteristics which make it quite challenging. These include complex data, which are characterized by nonlinearity, nonstationary and high dimensionality. The state space is very high-dimensional, which leads to problems in many machine learning methods need to get reliably good results even in high-dimensional state space. Solution methods must deal with incomplete system observations. SCADA has missing or erroneous data. Controllers work on live feeds. Another challenge is given by the existence of different time scales in typical real systems. In particular, both the quick and the slower dynamic components must be captured without devaluing each other as noise. Regression based models typically fail at this requirement. Machine Learning methods have evolved to handle such issues, though there exists a wide array to adopt. In this deliverable, we discuss those specific machine learning data-driven models which have the best potential to provide novel effective wind farm wide turbine control. This deliverable serves a recommendation of methods and models to pursue in further application trials with additional resources which include real operational wind farm SCADA data, simulation results and computing resources.

For more complex systems important system parameters are often limited or not measurable. In wind turbines, turbulence and wind shear caused locally cannot be determined exactly. The wind speed is measured at the nacelle height only. Thus, the sensory information available is incomplete and very noisy. Furthermore, environmental inflows and boundary conditions are very non-linear, partly chaotic (turbulence) and temporally non-stationary.

Various strategies based on neural networks have been studied to learn in varying environments. Online methods have been developed which have an adaptive learning rate control and takes changes in the environment into account. Within the covariate shift frameworks, new methods have been developed that can compensate for differences between the distributions of the training and test data by reweighting of training examples. Here we discuss specific machine learning data-driven models which may provide novel effective wind farm wide turbine control.

1.1 Machine Learning

There are four sections of models in machine learning which are used depending on what input and output data is available, these four are:

1. Supervised learning: When both input and output data is available.
2. Unsupervised learning: When only the input data is available.
3. Semi-supervised learning: When only a small amount of the inputs has a corresponding output.
4. Reinforced learning: When no historical input or output is available, the parameter learning is then based on a trial and error reward system.

Supervised learning is the easiest and fastest method to use since the selected model can be evaluated directly on historical data. This allows quicker adjustment of the model parameters or a change to another model if it does not work. It is often quite expensive to get to the stage where supervised learning can be used as it is often expensive to label all the output data. One example of this is object recognition in images where a competition has been held each year since 2010 on the data set ImageNet which as of 2016 has over 10 million hand-annotated images with labels of what is in it.

Unsupervised learning is used to infer a function to describe a hidden structure in the input data. This can for example be used in clustering, anomaly detection, neural networks, or to find hidden variables. Since it only works with the input data, it is fairly cheap to use since there will be no manual labelling of the output. It also means that it can only find things that exist in the input. For example, if the input data consist of the parameters wind speed and direction it would be impossible to predict alarms due to high vibration since there is no information about accelerations and/or forces exerted on turbines. But it would be possible to use it to detect abnormalities in the data and warn when they happen.

Semi-supervised learning often use both supervised and unsupervised learning in order to accomplish its task. First unsupervised learning is used in order to label or infer a label to the unlabelled output data and then supervised learning is used in order to accomplish its task. This means that it has the advantages of supervised learning but also the disadvantages of unsupervised learning where it is hard to see if it has correctly labelled the data. An example of where semi-supervised learning can be used is the following: A wind farm has been logging the SCADA since 2001, and 2016 it started labelling the data with the reasons for any power shutdowns. If we now want to predict any future turbine shutdowns using all of the data from 2001, semi-supervised learning can be used in order to first infer the label overflow on the data from 2001 to 2016 and then supervised learning could be used to predict the stoppages.

Reinforcement learning is concerned with the question of how an agent can learn optimal strategies through interaction with an external environment. Reinforced learning is unique in the way that it does not require any historical input or output data but instead learns from trial and error. This requires an environment where it can make mistakes and start over. A clear scheme of what is good and bad is also needed. A simulation of the problem to be solved and its environment is useful in this case. An example could be a simulation of a wind farm control where it tries to learn how to best control the turbines to optimize power and minimize fatigue loading.

1.2 White, Grey and Black Box Models

Models can be classified into three main categories including white-box, grey-box and black-box models depending on the level of machine learning integration. A white-box model is used when there is enough knowledge about the physics of the system. In this case, mathematical equations regarding dynamics of the system are utilized. No data is directly used to build in the model. White box control models can deal with dynamic equations of the system which are usually coupled and nonlinear, but no machine learning is implemented. A grey-box model combines a partial theoretical structure with data to complete the model. Here, parts of the analysis come from a dynamical physical algorithm, but a mix of machine learning algorithms are also included. A black-box model is used when no or little information is available about the physics of the system and is based on data only. In this case, the aim is to disclose the relations between variables of the system using operational or simulated operational data. This is the full machine learning based case. Artificial Neural Networks are one of the most significant methods in black-box modelling. It is a fast-growing method which has been used in different industries during recent years alongside reinforcement learning techniques.

1.3 Grey-Box Model Implementations

With white-box models being the current state of wind turbine control and black box models being a fully machine learning data driven solution, there are many ways in which machine learning models and be used in tandem with physical models in a grey box wind farm control system. As wind farm control deals with time series, machine learning can be used to predict the next state in the system like the current use of Kalman filters. Combinatorial neural networks can recognise parameter clustering. These clusters can identify optimal patterns of behaviour and indicate what parts of the state space needs to be modelled more accurately. Machine learning algorithms can detect changes that are practically significant, so the controller knows when it should or should not react. Note feedback on the quality of the learning method is indirect and noisy. Optimal decision strategies to cope with change, for example, to decide whether a learning system should be adapted or whether complete relearning is necessary. ML also excels at autonomous feature extraction and design of core functions. Feature extraction can reduce the state space needed for simulations of input producing localized estimates. Gaussian processes, for example, provide a probabilistic forecast model from which mean, and variance can be derived of prediction. These indicate where more information is needed to adequately model the parameters in the physical model and which parameters, or features carry more influence in final predictions. These error estimates also can help to inform which functions can be automated, and which are best left to physical modelling

SUPERVISED MACHINE LEARNING ALGORITHMS

Machine learning algorithms can be classified as, supervised, unsupervised and semi supervised. The majority of practical applications involves supervised machine learning where the response target is known, and the training set contains examples of input and the response out. For the wind farm control problem, neural networks and Kalman filters, extended and unscented, offer some promise for practical application.

2.1 NEURAL NETWORKS

A neural network model is a group of interconnected artificial units (neurons) with linear or nonlinear transfer functions. Known as Artificial neural networks (ANN), the networks consist of neurons are arranged in different layers. The number of neurons and layers in an ANN-based model depends on the degree of complexity of the system dynamics. ANNs learn the relation between inputs and outputs of the system through an iterative process called training. Each input into the neuron has its own associated weight. Weights are adjustable numbers which are determined during training the network. Figure 1 shows a simple structure of a typical ANN with three inputs and two outputs.

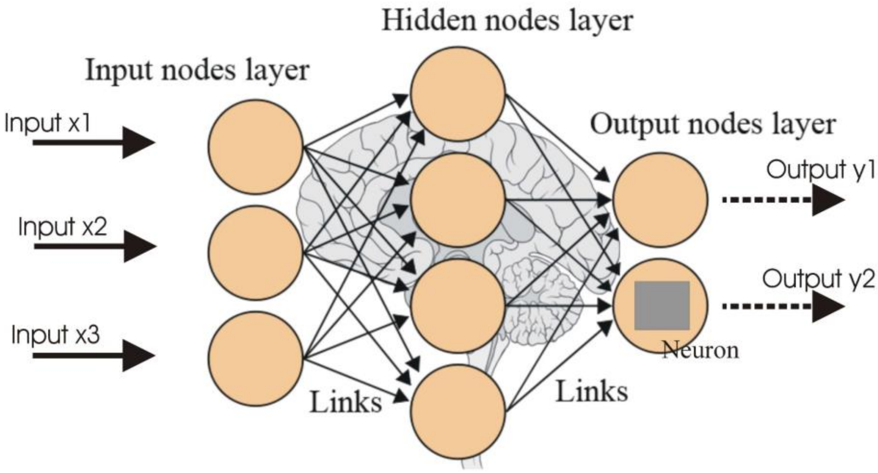

FIGURE 1– A SIMPLE STRUCTURE OF A TYPICAL ANN WITH INPUT, HIDDEN AND OUTPUT LAYERS. (ALI & AHUJA 2016)

Multilayer Perceptrons, or MLPs for short, are the classical type of neural network. They are comprised of one or more layers of neurons. Data is fed to the input layer, there may be one or more hidden layers providing levels of abstraction, and predictions are made on the output layer, also called the visible layer. MLPs are suitable for classification prediction problems where inputs are assigned a class or label. They are also suitable for regression prediction problems where a real-valued quantity is predicted given a set of inputs. They are very flexible and can be used generally to learn a mapping from inputs to outputs.

The more general artificial neural networks have capabilities which are desirable for the wind farm control problem. They include

- Robust to Noise. Neural networks are robust to noise in input data and in the mapping function and can even support learning and prediction in the presence of missing values.
- Nonlinear. Neural networks do not make strong assumptions about the mapping function and readily learn linear and nonlinear relationships.
- Multivariate Inputs. An arbitrary number of input features can be specified, providing direct support for multivariate forecasting.
- Multi-step Forecasts. An arbitrary number of output values can be specified, providing direct support for multi-step and even multivariate forecasting.

ANNs, as data-driven models, have been considered as a suitable alternative to white-box models during the last few decades either in grey-box sub processes or whole black-box solutions. ANN-based models can be created directly from the operational data of an actual WT or the simulated data from original equipment manufacturers (OEMs) performance information. Simulated data may be used when operational data are not available. The obtained data should cover the whole operational range of the system and quality controlled before the modelling process. ANN models for turbines can be created using different approaches and are flexible as to hyperparameters. Hyperparameters which can vary are the number of neurons, number of hidden layers, values of the weights and biases, type of the activation function, structure of the network, training styles and algorithms as well as data structure. The best structure is the one which can predict behaviour of the system as accurately as possible. This is problem specific and a result of trial and error of many different models. Selecting the right parameters of WTs, as inputs and outputs of the network, is very important for making an accurate and reliable model for all machine learning approaches. The availability of data for the selected parameters, system knowledge for identification of interconnections between different parameters and the objectives for making a model are basic factors in choosing appropriate inputs and outputs. Sensitivity analysis can be performed on results to indicate which parameters are of most importance.

Various strategies based on neural networks have been studied to learn in varying environments. Online methods have been developed which have an adaptive learning rate control and takes changes in the environment into account. Within the covariate shift frameworks, new methods have been developed that can compensate for differences between the distributions of the training and test data by reweighting of training examples. A well adopted approach within control systems engineering is Kalman filtering.

2.2 Kalman Filters

Learning algorithms are described for layered feedforward type neural networks, in which a unit generates a real-valued output through a logistic function. The well-known back-propagation algorithm based on gradient descent has a constant learning rate. In the wind farm control problem, the task of adjusting the weights of internal hidden units can be regarded as a problem of estimating (or identifying) constant parameters with a non-linear observation equation. In wind farm control, the use of Kalman filters for state estimation is an accepted technique, but the basic Kalman filter is limited with respect to some linearity assumptions. In machine learning, it has been shown that the use of the extended Kalman filters and the unscented Kalman filters in place of gradient descent in Neural Networks better handles applications with strong non-linearity. Extended Kalman filters accept non-linear functions in both the state transition and observation models. With the use of sine/cosine function and the exploitation of the Taylor Series, extend Kalman filters linearly approximate a non-linear function around the mean of a Gaussian distribution. There can be an issue of variance propagation through the model, though. Unscented Kalman filters on the other hand have been shown to minimize resulting errors propagating through the nonlinear system. (Wan 2000). Unscented Kalman filters utilize sigma points where a subset of points are mapped through a non-linear function to a transformed Gaussian instead of the entire function. Given the inherent noise and errors in the wind farm observations upon which a supervised neural network for wind farm control would be built, the unscented Kalman filter approach may result in a better performing algorithm over more traditional neural network methods.

Reinforcement Learning

Reinforcement learning (RL) deals with the question of how an agent can learn optimal strategies through interaction with an external environment. In practice, the environment in which the agent is working us often too complex or the state space is too high-dimensional for standard RL methods since these methods need to learn too many interactions. This is especially true for the case that Dynamic Programming (DP) is used. Approximate DP was one of the main approaches to overcome this problem, but it has issues with data efficiency, the ability to learn with just a few interactions. A promising approach is neural fitted Q-iteration algorithms. This algorithm has a special case of a neural reward regression method. Such novel combinations of RL with neural networks show, in practice, a high degree of data efficiency.

Further progress can be achieved when model-based reinforcement learning methods are associated with recurrent neural networks because they can learn accurate predictions from fewer data. Generally, in predictive applications there is an increased interest in methods based on neural networks for modelling of nonlinear systems and environments. It was shown that the results of state estimation conform to both theoretical requirements (Markov property), as well as cope with the problems that bring with them the real data (high dimensionality, incompleteness, redundancy). The recurrent neural networks are used to expand the learning skills of the agents with a neural memory. These networks, the state-space models with discrete time scale, have universal approximation properties and allow highly accurate predictions which allows the agent to act intelligently based on a model of future change. Recurrent networks for forecasting have developed to include error correction and dynamic consistency for better forecasting capability.

Reinforcement learning has both exploration and exploitation capabilities. Exploration is meta-learning exploration of the relationship between parameters and exploring the system dynamics, Exploitation is control use of the learned system dynamics. Using meta-learning exploration meta-parameters are created, not by hand, but adapted based on the learning progression and globally mapped for all states. For of the exploration control, optimal choices are decided either the instantaneous reward the environment, or the prediction error δ which is the expected discounted rewards. The control processes are state-dependent, not global, based on the sensory perception of the environment and adaptive meta parameters given by the exploration strategy. The agent learns processes in the exploratory state space. In unknown system dynamics, a higher absolute value of the prediction error δ exists and recorded as "incomplete knowledge environment" or "model uncertainty". Exploration activities in these high uncertainty areas increases knowledge, so future behaviour of the environment or the technical system can be predicted more accurately in more conditions. It is an iterative process of learning local state spaces.

In reinforcement learning, there is the element of failure in order for the system to learn. For wind farm control, the safety and financial consequences of turbine or wind project-wide failure prohibit the training on a real installation. Therefore, a highly realistic wind farm simulator is required for training the control strategy in a simulated environment. The digital training environment must accurately capture the dynamic forces acting on the turbines and resulting wakes coming from changes in the control operating settings.

3.1 Monte Carlo tree search & Markov decision process (MDP)

Monte Carlo tree search is type of reinforcement learning utilised in game playing implementations such as Alpha Go. It is a locally adaptive tree search where it reduces the search space from all possible moves to those most likely given the current position. In the game application, the algorithm searches for the most promising moves, expanding the search tree based on random sampling of the search space. The game is played many times to end by selecting moves at random, expanding the tree. The end result of each game: win, loss, final score, is used to weight the nodes in the game tree so that better nodes are more likely to be chosen in future game matches. Markov decision processes describe an environment for reinforcement learning, where the environment is fully observable. Almost all RL problems can be formalized as MDP. It is characterized by a discrete time stochastic process where the outcomes are partially random and partially under control based on the decision maker. Instead of being truly random, at each time step the process is in a given state and the decision maker choses the next action based on the options available in the current state. There is a reward at each state transition and such a probability that the process moves into each new state. This mean that for the game player there are actions which allow for choice and rewards which give motivation. MDP are used in robotic control, self-driving cars and other optimal control strategy applications. For wind farm control, MDP might result in sufficient time step dependency to achieve reasonable results. Though, there are known longer persistence signals in atmospheric, turbine fatigue and production data which may mean that an algorithm that includes an element of entropy might outperform MDP. Further investigations are required to assess the true applicability for wind farm control.

3.2 Recurrent Neural Network

Recurrent Neural Networks or RNNs are a special type of neural network designed for sequence problems. Given a standard feed-forward multilayer Perceptron network, a recurrent neural network can be thought of as the addition of loops to the architecture to allow information to persist. For example, in a given layer, each neuron may pass its signal latterly (sideways) in addition to forward to the next layer. The output of the network may feedback as an input to the network with the next input vector. And so on. The recurrent connections add state or memory to the network and allow it to learn broader abstractions from the input sequences.

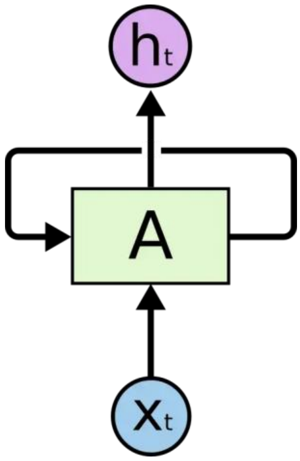

FIGURE 1 IN THE ABOVE DIAGRAM, A CHUNK OF NEURAL NETWORK, A, LOOKS AT SOME INPUT XT AND OUTPUTS A VALUE HT. A LOOP ALLOWS INFORMATION TO BE PASSED FROM ONE STEP OF THE NETWORK TO THE NEXT. (OHLAH 2015)

3.3 LSTM

The Long Short-Term Memory network or LSTM is a recurrent neural network that can learn and forecast long sequences capable of learning long-term dependencies. It can learn the order dependence between items in a sequence.

A benefit of LSTMs in addition to learning long sequences is that they can learn to make a one-shot multi-step forecast which is useful for time series forecasting.

The Long Short-Term Memory or LSTM network is a recurrent neural network that is trained using Backpropagation Through Time and overcomes the vanishing gradient problem.
As such it can be used to create large (stacked) recurrent networks, that in turn can be used to address difficult sequence problems in machine learning and achieve state-of-the-art results.
Instead of neurons, LSTM networks have memory blocks that are connected into layers.
A block has components that make it smarter than a classical neuron and a memory for recent sequences. A block contains gates that manage the block's state and output. A unit operates upon an input sequence and each gate within a unit uses the sigmoid activation function to control whether they are triggered or not, making the change of state and addition of information flowing through the unit conditional.

There are three types of gates within a memory unit:

- **Forget Gate:** conditionally decides what information to discard from the unit.
- **Input Gate:** conditionally decides which values from the input to update the memory state.
- **Output Gate**: conditionally decides what to output based on input and the memory of the unit.

Each unit is like a mini state machine where the gates of the units have weights that are learned during the training procedure.

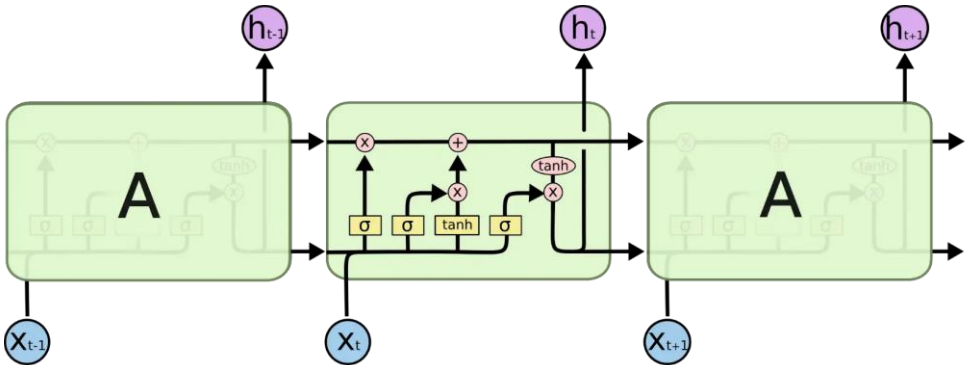

FIGURE 2 LONG SHORT-TERM MEMORY NETWORK (BROWNLEE 2018)

3.4 RNN LSTM

A promising method is to use Recurrent Neural Network (RNN) or Long Short-Term Memory (LSTM)-network which is the RNN's extension. These are very useful in time series predictions. The RNN works by letting the previous output of the network be used together with the new input to create a new output. When doing this it weighs how much of the previous output it should keep in order to create an accurate new output. This network has the same problem as the Markov assumption, the further away in time the worse performance. The LSTM-network solves this by adding a memory into the network that can remember the important events which can cause certain control issues and forget the unimportant ones. The LSTM-network also has the advantage that it solves the vanishing gradient problem which made it harder to build deep networks. It does however have the problem of being slow to train.

3.5 CNN LSTM

The CNN LSTM architecture involves using Convolutional Neural Network (CNN) layers for feature extraction on input data combined with LSTMs to support sequence prediction. CNN LSTMs were developed for visual time series prediction problems and the application of generating textual descriptions from videos. The feature extraction means they are able to deal with noisy input data. The LSTM can incorporate information from longer sequences. These two characteristics make CNN LSTM's extremely attractive for use in wind farm control. CNN LSTMS do require large amounts of input data, adequate coverage of the input space and are extremely computationally intensive to train. Extensive compute resources are required.

3.6 NFQ

The NFQ algorithm, Neural Fitted Q-iteration, is a data efficient RL-process which can learn a control strategy for complex systems based less interaction data. The goal is to learn a policy that guides an agent under various circumstances. It does not require a model of the environment and can handle problems with stochastic transitions and rewards, without requiring adaptations. This method is limited in that it can be applied only to Markov decision models with discrete actions. However, many technical systems have continuous control variables, such as motor commands for turbine. A discretization of these manipulated variables would lead to the optimal discrete action which often does not deliver the same performance as the optimal continuous action.

In project ALICE, the NFQ was further developed into two methods with continuous manipulated variables, NFQG and NFQCA. The NFQG (NFQ with gradient descent) adds gradient descent manipulation to a manipulated variable a where the state control value pair (s, a) applied as input to the neural network and the ideal Q- Value, Q (s, a) = 1. In NFQG this process is carried out for all interaction data of batches with different starting positions a (Minimum search), and ultimately the improved action strategy is approximated using a separately-to-learn strategy net and generalized. The NFQCA process (NFQ with Continuous Actions) combines the basic idea of data efficient NFQ Learning and the gradient descent NFQG-in task space to a common network architecture. Here, the computationally intensive step of the minimum search is replaced in the task space by the coupling of the two neural networks (Q- and strategy Network) as a triangular mesh, whereby a closed learning method arises.

3.7 RL With Gaussian Processes

A Gaussian process (GP) is a system method for predicting a function with a limited amount of training data based on the theory of probability. If only small amount of training data exists, a large uncertainty arises regarding the underlining function that generated the data. A GP can explicitly model and quantify this uncertainty. For a given input vector of limited test data, a GP returns a probability distribution over the function with values at the points with predicted mean and standard deviations. This explicit consideration of model uncertainties makes different Gaussian processes from most other regression methods. By adding data points form the higher areas of uncertainty, the function reins in the set of possible final functions and brings the model closer to the unobserved real process.

Data efficient reinforcement learning, in which only a few interactions with the system are available for learning, typically requires a model of the system dynamics. Only limited data of the system are available, and the model of the system dynamics can only be learned with great uncertainty. High model errors can be an issue for reinforcement learning from true dynamic models.

3.8 PILCO

Pilco (Probabilistic Inference for Learning Control) avoids this problem by including a probabilistic dynamics model in the form of a non-parametric regression GP for learning and modelling the long-term planning and policy search. The long-term planning consists of predicting the future state sequences in the current policy. The propagation of the uncertainties over several time steps is carried out by means of deterministic approximate inference. Pilco is suitable for continuous state and action spaces.

The input space is extended by adding additional a-priori system knowledge introduced as a mean value function of the GP. This is particularly advantageous when only a few data points are available for a higher-dimensional dynamic model (for example, 200 data points of a 18-dimensional function). Most applications of Pilco relate to the learning of robot controllers and show promise for the wind turbine control problem.

CONCLUSIONS

Future energy generation plants, through supervised machine learning and reinforcement learning, could learn independently advantageous decision strategies through interaction with the environment and continually optimize these strategies through a continuous learning process. The aim is to autonomously try unknown but promising strategies and learn from the success or failure of a control system. There are promising developments in NFQ, Gaussian processes coupled with long short-term networks when historical wind farm information, or reliable simulation data are available. In the Reinforcement learning framework, adaptive Markov-tree models which could be suitable for use in wind farm control as the state space is reduced at various time step. The proof of concept has been demonstrated in the single turbine case by various researchers. The wind farm project-wide case has been considered by the Horizon 2020 ALICE project. There is the element of failure in order for the system to learn in a machine learning framework. However, these so-called explorations must be done in a safe manner such that neither the system to be controlled nor its energy production is significantly impaired. For wind farm control, the safety and financial consequences of turbine or wind project-wide failure prohibit the training on a real installation. Therefore, a highly realistic wind farm simulator is required for training the control strategy in a simulated environment. The digital training environment must accurately capture the dynamic forces acting on the turbines and resulting wakes coming from changes in the control operating settings. The inclusion of real farm production data will be key as a machine learning model based purely on simulated data will lean the simulated environment only. A grey-box model with machine learning models in complement with deterministic physical models offers a level of transparency and explainablity with potential for improved results over traditional white-box models. Full black box models, while requiring extensive data and computations resources to create, may offer further advantages as all simplifications and assumptions inherent in the physical models is removed.

REFERENCES

1. Ali, S.D., Ahuja, R.(2016). The Evolution and Core Concepts of Deep Learning & Neural Networks. Available at: https://www.analyticsvidhya.com/blog/2016/08/evolution-core-concepts-deep-learning-neural-networks/. Accessed 7 Sep. 2018.

2. Brownlee, J. (2018). Machine Learning Mastery. Available at: https://machinelearningmastery.com/blog/. Accessed 7 Sep. 2018.

3. Hauptmann, W., Hentschel, A., Otte, C., et al ."ALICE: Autonomes Lernen in komplexen Umgebungen : Abschlussbericht zur Veröffentlichung : Berichtszeitraum: 01.07.2010-30.06.2014" https://doi.org/10.2314/GBV:857321463 , *Siemes AG,* 2015.

4. Mika, Sebastian. "ALICE: Autonomes Lernen in komplexen Umgebungen : Schlussbericht Teilvorhaben" https://doi.org/10.2314/GBV:837610982. *idalab GmbH,* 2014.

5. Olah, C. (2015). Understanding LSTM Networks. *colah's blog*. Available at: http://colah.github.io/posts/2015-08-Understanding-LSTMs/Accessed 7 Sep. 2018

6. L. F. Ochoa, G. P. Harrison, "Minimizing energy losses: Optimal accommodation and smart operation of renewable distributed generation", *IEEE Trans. Power Syst.,* vol. 26, no. 1, pp. 198-205, Feb. 2011.

7. Kravchyshyn, V., Medykovskyy, M.. Melnyk, R, & Dilai, M. "Optimizing wind farm structure control", *Advances in Intelligent Systems and* Computing, vol. 689, pp. 320-333, 2018.

8. Wan, E.A., van der Merwe, R.. "The unscented Kalman filter for nonlinear estimation." Paper presented at the meeting of the Symposium on Adaptive Systems for Signal Processing, Communication and Control, Lake Louise, Alberta, Canada, 2000.

9. Wikipedia contributors. Q-learning. Wikipedia, The Free Encyclopedia. September 22, 2018, 16:56 UTC. Available at: https://en.wikipedia.org/w/index.php?title=Q-learning&oldid=860727273. Accessed 23 Sep 2018.

10. Z. Z. Zhang, H. B. Xu, J. X. Zou, G. Zheng, "Sliding mode control-based active power control for wind farm with variable speed wind generation system", *Proc. Inst. Mech. Eng. J. Mech. Eng. Sci*., vol. 227, no. 3, pp. 449-458, Oct. 2012.

11. H. R. Zhao, Q. W. Wu, Q. L. Guo, H. B. Sun, Y. S. Xue, "Distributed model predictive control of a wind farm for optimal active power control—Part I: Clustering-based wind turbine model linearization", *IEEE Trans. Sustain. Energy,* vol. 6, no. 3, pp. 831-839, Jul. 2015.

MACHINE LEARNING CONTROL OF SINGLE TURBINE RESEARCH

1. Barbade, S.A., & Kasliwal, P.”Neural network based control of Doubly Fed Induction Generator in wind power generation” *International Journal of Advancements in Research &Technology*, vol.1, no. 2, pp. 83-88, July 2012.

2. A. Boubakir, S. Laliod, F. Boudjema, "A stable self-tuning proportional-integral-derivative controller for a class of multi-input-multi-output nonlinear systems", *J. Vib. Control*, vol. 18, no. 2, pp. 228-239, Jul. 2012.

3. B. Boukhezzar, H. Siguerdudjane, "Nonlinear control of variable speed wind turbines for power regulation", *Proc. IEEE Conf. Control Appl.,* pp. 1085-1992, Sep. 2005, [online] Available: http://dx.doi.org/10.1109/CCA.2005.1507110.

4. L. Jizhen, G. Junlin, H. Yang, W. Juan and L. Hong, "Dynamic modeling of wind turbine generation system based on grey-box identification with genetic algorithm," *2017 36th Chinese Control Conference (CCC),* Dalian, pp. 2038-2042, 2017.

5. W. C. Meng, Q. M. Yang, Y. Ying, Y. Sun, Z. Y. Yang, Y. X. Sun, "Adaptive power capture control of variable-speed wind energy conversion systems with guaranteed transient and steady-state performance", *IEEE Trans. Energy Convers*., vol. 28, no. 3, pp. 716-725, Sep. 2013.

6. L. F. Ochoa, G. P. Harrison, "Minimizing energy losses: Optimal accommodation and smart operation of renewable distributed generation", *IEEE Trans. Power Syst.,* vol. 26, no. 1, pp. 198-205, Feb. 2011.

7. U. Ozbay, E. Zergeroglu, S. Sivrioglu, "Adaptive backstepping control of variable speed wind turbines", *Int. J. Control,* vol. 81, no. 6, pp. 910-919, Jun. 2008.

8. M. Pucci, M. Cirrincione, "Neural MPPT control of wind generators with induction machines without speed sensors", I., vol. 58, no. 1, pp. 37-47, Jan. 2011.

9. B. B. Ren, S. S. Ge, K. P. Tee, T. H. Lee, "Adaptive neural control for output feedback nonlinear systems using a barrier Lyapunov function", *IEEE Trans. Neural Netw*., vol. 21, no. 8, pp. 1339-1345, Aug. 2010.

10. R. Shahnazi, M.-R. Akbarzadeh-T, "Pi adaptive fuzzy control with large and fast disturbance rejection for a class of uncertain nonlinear systems", *IEEE Trans. Fuzzy Syst.,* vol. 16, no. 1, pp. 187-197, Feb. 2008.

12. Q. Song, Y. D. Song, "Generalized PI control design for a class of unknown nonaffine systems with sensor and actuator faults", *Syst. Control Lett.,* vol. 64, no. 4, pp. 86-95, Feb. 2014.

11. Song, Yongduan & Liang, Liyuan & Tan, Mi. “Neuroadaptive Power Tracking Control of Wind Farms Under Uncertain Power Demands”, *IEEE Transactions on Industrial Electronics*. PP. 1-1. 10.1109/TIE.2017.2682789. March 2017.

12. Y. D. Song, B. B. Zhang, K. Zhao, "Indirect neuroadaptive control of unknown MIMO systems tracking uncertain target under sensor failures", *Automatica,* vol. 77, pp. 103-111, Mar. 2017.

13. A. Uehara, T. Senjyu, E. A. T. Kaneko, "Output power dispatch control for a wind farm in a small power system", *Wind Energy,* vol. 13, no. 7, pp. 671-684, Mar. 2010, [online] Available: http://dx.doi.org/10.1002/we.388.

14. Q. Wu, H. Zhao, Q. L. Guo, H. B. Sun, Y. S. Xue, "Distributed model predictive control of a wind farm for optimal active power control—Part II: Implementation with clustering-based piece-wise affine wind turbine model", *IEEE Trans. Sustain. Energy,* vol. 6, no. 3, pp. 840-849, Jul. 2015.

13. J.M Zamarreño, P Vega, L.D Garcia, & M Francisco, “State-space neural network for modelling, prediction and control,” *Control Engineering Practice,* vol. 8, no 9, pp 1063-1075, Aug 2000.