Proceedings of the ASME Turbo Expo 2013: Power for Land, Sea and Air
GT2013
June 3-7, 2013, San Antonio, Texas, USA

GT2013-94300

SIMULATION OF WIND FARM OPERATIONS AND MAINTENANCE

Eduardo Pérez
Ingram School of Engineering
Texas State University
San Marcos, Texas 78640
Email: eduardopr@txstate.edu

Lewis Ntaimo
Yu Ding*
Industrial & Systems Engineering
Texas A&M University
College Station, Texas, 77843
Email: ntaimo@tamu.edu
Email: yuding@iemail.tamu.edu

ABSTRACT

We develop a discrete event-based simulation framework that mimics the operations of a commercial size wind farm. Each turbine is treated as separate module, so that the simulation can be easily scaled up to more than one hundred turbines for a farm. Each turbine module includes a structural element sub-module, degradation sub-module, power generation sub-module, sensing and maintenance scheduling sub-module. The simulator is specially designed to handle a large number of unorganized random events (turbine failures, waiting for parts, weather disruptions) and reflect in the simulator's outputs the variation from parameters and operations. We report on implementation results and provide insights into wind farm operations under different maintenance strategies.

NOMENCLATURE

X_M Set of input ports.
Y_M Set of output ports.
S Set of sequential states.
δ_{ext} External transition function.
δ_{int} Internal state transition function.
δ_{con} Confluent transition function.
λ Output function.
ta Time advance function.
Q Set of total states.

INTRODUCTION

A commercial size wind farm usually houses more than one hundred turbines spread over a large geographical area located remotely from any population centers. Caring for this fleet of turbines is not a trivial undertaking. To our best knowledge, the current work on wind turbines operation and maintenance (O&M) primarily focuses within a single turbine, and oftentimes, even a single component of a turbine [1, 2]. When a farm is considered, simplified assumptions are commonly used, for example, assuming that all the turbines are identical not only at the initial time but also throughout their life time. Such assumptions are apparently not realistic.

We believe that injecting realism into the modeling and analysis of a wind farm's operation is of critical importance. Top on the needs is the development of wind farm simulation method that can mimic the operation of a hundred plus turbines, governed by the models pertaining to the turbines' stochastic degeneration and behaviors. Such a simulation platform can be a valuable tool to test any control logics or maintenance scheduling strategies for assessing their effectiveness with a good degree of realism in it.

That is indeed our undertaking in this research, i.e., we develop a discrete event-based simulation framework for representing the operations of a commercial size wind farm. Establishing the simulation framework entails two major efforts: (1) the logic framework that interconnects components in wind power systems (such as turbines) and mimics the execution and operation in a virtual, cyberspace environment; (2) detailed models

*Address all correspondence to this author.

that characterize turbine's dynamic responses and deterioration, maintenance scheduling, and local wind prediction.

A viable approach for modeling complex large-scale wind power systems is to use the discrete event system specification (DEVS) formalism [3, 4]. We select DEVS because it allows for modeling multi-scale (time, space and decisions) complex systems. DEVS is a formal modeling and simulation framework based on dynamical systems theory, providing well-defined concepts for coupling atomic and coupled models, hierarchical and modular model construction, and an object-oriented substrate supporting repository reuse. DEVS allows us to build a simulation platform for wind power operations at different granularities: wind turbine, wind farm and power grid and network. In this research, our emphasis is on the turbine and farm level modeling and simulation without getting into the details of how to connect to the power grid.

Specifically, our development involves the following: (a) building wind farm DEVS atomic models; (b) coupling the atomic models to create complex coupled models; (c) building the experimental frame to allow for a suite of simulation experimental choices; (d) computer implementation of the models; and (e) testing and validation of the simulation models. The simulator is specially designed to handle a large number of unorganized random events (turbine failures, waiting for parts, weather disruptions) and reflect in the simulator's outputs the stochasticity from parameters and operations.

The rest of the paper is organized as follows: In the next section, we review related work on wind turbine and wind farm simulations. After that, we provide the details on our simulation model. We also present the overall hierarchical structure and operation of the wind farm simulation. Subsequently, we discuss alternative maintenance scheduling strategies and practices. Finally, we discuss the computational results and end the paper with some concluding remarks and directions for further research.

LITERATURE REVIEW

The phrase *simulation* is used broadly in the wind energy literature. It generally refers to using computational models and running them on computers as a surrogate or replacement of running physical experiments and operations. The simulation models and tools can be categorized into two schools of thought: (a) Monte Carlo-based simulations, which start with random number generations and are based on probability distributions of certain operation characteristics; (b) Numerical simulation based on physical principles such as Finite Element Analysis (FEA) or Computational Fluid Dynamics (CFD) models or tools. The two schools of thought can be in principle combined, even though actually doing so is not very common, because Monte Carlo-based simulations required a large number of replications of running a computer model, while each individual running of a FEA or CFD model is already computationally expensive. As a result, the two schools of thought have so far proceeded more or less in parallel.

Our research in this paper is in the line of Monte Carlo-based simulations rather than of the physics-based numerical simulations. The latter is almost always conducted for individual turbines (more precisely, individual components of a turbine). It is difficult to imagine, in any foreseeable future, the feasibility of running a farm size FEA or CFD model. Nonetheless, readers interested in the physics-based numerical simulation should be aware that National Renewable Energy Lab in the US and Risø National Lab in Denmark, the two leading organizations in wind energy technology, have developed their own aeroelastic computer simulation tools [5–7].

Monte Carlo-based simulations are based, quite naturally, on collecting and analyzing of the historical failure and operation data to elucidate a turbine's failure probability and operation characteristics. The idea is to use the historical data to fit certain probability distributions, which then yield a number of commonly used statistics such as the mean time to failure (MTTF) [8–14] . The popular distribution here, as in other reliability analysis, is the Weibull distribution. Recent work by Tavner et al. [15, 16] and Guo et al. [17] used the non-homogeneous Poisson process to handle the cases where the number of failures is provided but the actual time when a failure takes place is missing. Existing simulations were conducted generally for studying a specific objective, including evaluating the effectiveness of maintenance actions [18–20], assessing the impact of turbine reliability on power generation [21, 22], comparing turbine siting choices [23], and validating operational strategies [11, 18].

The limitations of the current Monte Carlo-based simulation can be summarized as: (a) The current models are generally oversimplified. Most were performed on a single turbine, or a farm but with the assumption of identical turbines. Stochasticity resulting from different weather profiles, wind turbine types, operational strategies were generally not considered but should have been. (b) The existing work lacks decision-making ability inside the simulation models. That is to say, there is no integrated framework for wind farm operations in which simulations can interact with decision-making modules during simulation runs. As a result, people can evaluate the impact of a fixed schedule maintenance which does not change throughout the turbine's life cycle but could not do so for anything more sophisticated. To our best knowledge, the only exception is our most recent work, attempting on modeling with sufficient granularity a wind farm's operation [24]. The limitation of that work, since it is the first of this kind, is that each turbine was simplified to the consideration of a single component, or equivalently, assuming that only one major component deteriorates over the time, while other components remain unchanged.

In this paper, we extend our previous work by modeling four major components for each turbine, a much more realistic treatment than that in [24]. Not only are extra models needed for representing the additional components, more importantly, the corresponding atomic models for maintenance scheduling and dispatch are different and need to be developed in this paper.

SIMULATION MODEL

DEVS allows for building complex models using a bottom-up approach, starting with *atomic* models and then coupling them to create *coupled* models. Each atomic model has input ports and output ports through which the couplings are done. These couplings allow for exchange of messages between coupled models. The inputs ports allow for the atomic model to receive inputs from outside the model, process the inputs, and generate output through the output ports. The dynamic behavior is captured within the atomic model using a set of *states* and *functions*. We use Parallel DEVS which allows for processing multiple inputs simultaneous. More formally, a Parallel DEVS atomic model is a structure and is defined as follows:

$$DEVS = (X, Y, S, \delta_{ext}, \delta_{int}, \delta_{con}, \lambda, ta), \tag{1}$$

where

$X = \{(p, v) | p \in IPorts, v \in X_p\}$ is the set of input ports and values, where *IPorts* is the set of input ports;

$Y = \{(p, v) | p \in OPorts, v \in Y_p\}$ is the set of output ports and values;

S is the set of sequential states;

$\delta_{ext} : Q \times X^b \to S$ is the *external transition* function, where X^b is a set of *bags* over elements in X and Q is the set of total states;

$\delta_{int} : S \to S$ is the *internal state transition* function;

$\delta_{con} : Q \times X^b \to S$ is the *confluent transition* function;

$\lambda : S \to Y^b$ is the output function;

$ta : S \to R^+_{0,\infty}$ is the *time advance* function; and

$Q := \{(s, e) | s \in S, 0 \le e \le ta(s)\}$ is the set of total states, where s is the state and e is the elapsed time.

At any given time, a DEVS atomic model is in some state s and if no external events occur, the model remains in state s for a time $ta(s) \in [0, \infty]$. When this time elapses the system outputs the value, $\lambda(s)$, and transitions to a state $s' = \delta_{int}(s)$. The model remains in the current state for ever (passive) if $ta(s) = \infty$. In a DEVS atomic model an output can only be generated after an internal transition. If an external event $x \in X$ occurs when the model is state (s, e) with $e \le ta(s)$, it transitions to state $s' = \delta_{ext}(s, e, x)$. The *external transition* function determines the new state when an external event occurs, while the *internal transition* function determines new state when no events occur since the last transition. The *confluent function* decides the next state in cases when there is a collision, that is, when there is an external event exactly when an internal transition has to occur.

The DEVS specification includes external interface, components (atomic or coupled models), and the coupling relations to enable constructing models from components. Formally, let *EIC*, *EOC* and *IC* respectively denote the external input coupling, external output coupling and internal coupling. Then a coupled model N is defined mathematically as follows:

$$N = (X, Y, D, \{M_d \mid d \in D\}, EIC, EOC, IC), \tag{2}$$

where D is the set of component names, and for each $d \in D$,

$$M_d = (X_d, Y_d, S, \delta_{ext}, \delta_{int}, \delta_{con}, \lambda, ta)$$

is a DEVS model with

$$X_d = \{(p, v) | p \in IPorts_d, v \in X_p\}$$

and

$$Y_d = \{(p, v) | p \in OPorts_d, v \in Y_p\}.$$

The external input coupling, *EIC*, connect external inputs to component inputs:

$$EIC \subseteq \{((N, ip_N), (d, ip_d)) \mid ip_N \in IPorts, d \in D, ip_d \in IPorts_d\}.$$

Similarly, the external output coupling, *EOC*, connect external outputs to component outputs:

$$EOC \subseteq \{((N, op_d), (N, op_N)) \mid op_N \in OPorts, d \in D, op_d \in OPorts_d\}.$$

Finally, the internal coupling, *IC*, connect component outputs to component inputs:

$$IC \subseteq \{((a, op_a), (b, ip_b)) \mid a, b \in D, op_a \in OPorts_a, ip_b \in IPorts_b\}.$$

DEVS does not allow for an output port of a component to be connected to an input port of the same component. Thus in DEVS $((a, op_a), (b, ip_b) \in IC$ implies $a \neq b$. In other words, no direct feedback loops are allowed for each component.

Using the characterizations of DEVS atomic and coupled models in (1) and (2), we abstract a wind farm as a dynamical system with components that interact to produce wind power. To

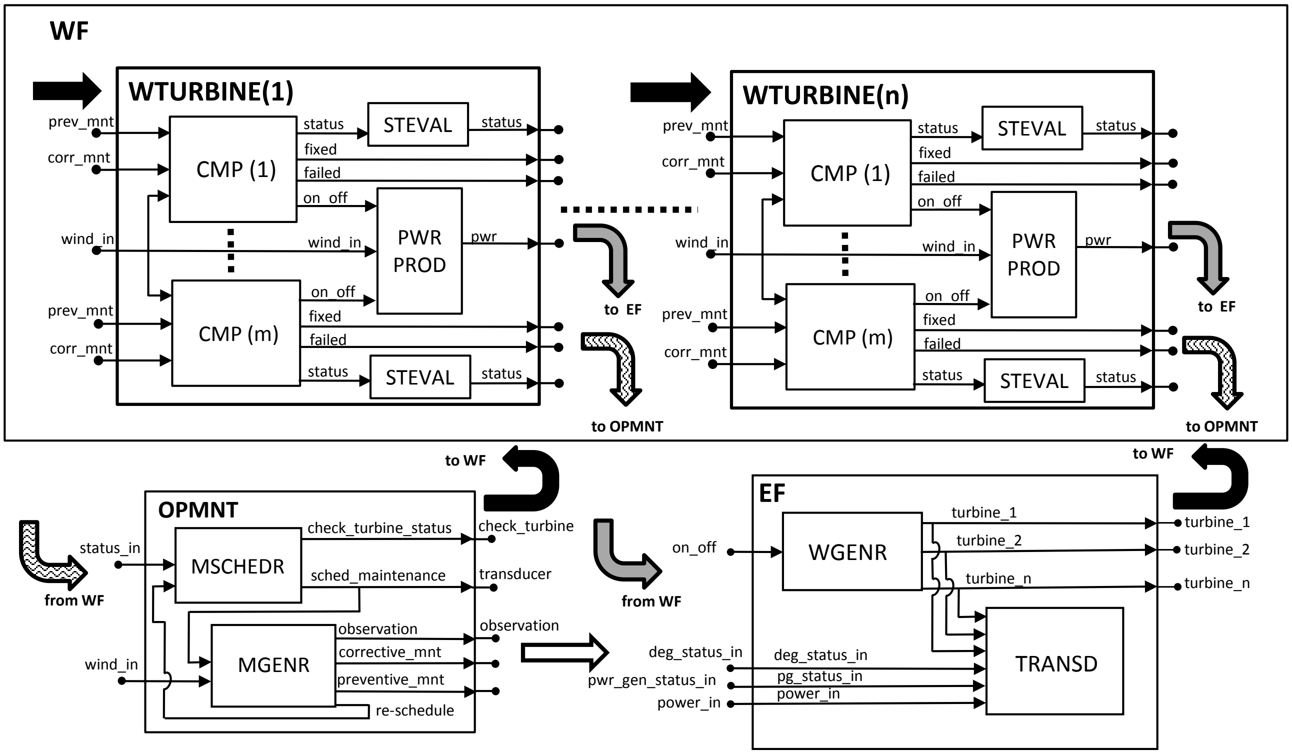

FIGURE 1. DEVS wind farm simulation model

build a simulation model of a wind farm, we consider all the critical components associated with wind farm operations and maintenance. Our DEVS wind farm simulation model is depicted in Figure 1. This model comprises three coupled models: wind farm (WF), operations maintenance (OPMNT) and experimental frame (EF). The wind farm coupled model has n wind turbines, labeled WTURBINE(1) to WTURBINE(n) in the figure. Each wind turbine has m components (labeled CMP(1) to CMP(m)) that make up the wind turbine. In particular, we model a wind turbine as having four ($m = 4$) components: a gearbox (CMP(1)), power generator (CMP(2)), blades (CMP(3)), and a control system (CMP(4)). We model these components as DEVS atomic models with a set of states and state transition functions based on the actual operations of each component.

Since we are considering condition-based monitoring, we assume that each wind turbine also has smart sensors that allow for evaluating the state/condition of each of the four turbine components. So we use a state evaluation (STEVAL) atomic model to capture this behavior. Finally, each turbine has a power production (PWR PROD) atomic model to compute the power generated by the wind turbine based on the prevailing wind speed, and the conditions of all the turbine components.

To model turbine maintenance activities, we use an operations maintenance coupled model OPMNT. This model comprises two atomic models: the maintenance scheduler (MSCHEDR) and maintenance generator (MGENR). As the name implies, MSCHEDR implements different algorithms for scheduling maintenance operations. In particular, we study both scheduled maintenance and condition-based maintenance. The MSCHEDR atomic model communicates with the MGENR atomic model when a maintenance procedure is scheduled. The MGENR atomic model is responsible for generating maintenance jobs at the scheduled times.

The experimental frame EF is a coupled model and is an important part of the simulation model. It is coupled to both WTURBINE and OPMNT and is used for designing and running experiments of interest by the user/modeler. For example, EF allows the user to specify experimental parameters and performance measures to compute. During simulation runs, the EF collects information of interest such as the amount of power generated, capacity factor, availability and turbine failures. As shown in Figure 1, the EF model is composed of two atomic models: a wind generator (WGENR) and a transducer (TRANSD). The WGENR atomic model is in charge of generating wind speed information for each one of the turbines the wind farm model WF. It allows to compute the wind speed for each WTURBINE based on its location in the wind farm and its height. Finally, the TRANSD atomic model collects information of interest from both WTURBINE and OPMNT, and computes and reports the performance measures specified by the user.

Due to limitation in space, we skip details of the operation of each component of our wind farm simulation model and simply provide a description of the operation of one of the components, the power generator atomic model. This is component CMP(2) and its block diagram is shown in Figure 2. This model has two input ports, "prev_mnt" and "corr_mnt"; and four output ports, "on_off", "fixed", "failed", and "status". CMP(2) is coupled to STEVAL and PWR PROD to enable component status verification and calculation of power produced, respectively.

The degrading behavior of CMP(2) is illustrated using the state transition diagram in Figure 3. For any given turbine component, its health status is categorized at four levels: "normal", the perfect operation state; "alert", a deteriorated state but still safe to operate; "alarm", a deteriorated state that could fail soon; and "fail", the state that the turbine component is no longer functioning. CMP(2) is initialized in state "normal", and its stochastic deterioration is characterized by a probability transition matrix P_2 calculated based on historical data of a wind turbine generator (similar approach is used in weather forecast to characterize the transition from "sunny" to "cloudy" to "overcast" to "raining" etc.):

$$P_2 = \begin{pmatrix} 0.995 & 0.004 & 0.001 & 0.000 \\ 0.000 & 0.985 & 0.010 & 0.005 \\ 0.000 & 0.000 & 0.985 & 0.015 \\ 0.000 & 0.000 & 0.000 & 1.000 \end{pmatrix}.$$

Assuming no maintenance is performed, the transition matrix shows that after some stochastic time duration the model remains in state "normal" with probability 0.995, or transitions from "normal" to "alert" with probability 0.004, or to "alarm" with probability 0.001. When in state "alert", the model remains in this state with probability 0.985, or transitions to either "alarm" or "fail", with probability 0.010 and 0.005, respectively. The model remains in "alarm" state with probability 0.985 and transitions from "alarm" to "fail" with probability 0.015. Once in the "fail" state, the model remains in this state with probability one. Only after corrective maintenance will the model be re-initialized to "normal" state. While in the operational states "normal", "alert" and "alarm", the model transitions to "off" when the wind speed reaches the cut out limit. When the wind speed is within the turbine operational limits, the model transitions from the "off" state to "checkSate", and then returns to the state it was in before.

Preventive maintenance can be performed while the model is in any of the operational states. In this case, the model transitions to "preventiveM". After preventive maintenance is completed, the model goes to "off", then to "checkState", and finally restarted as new in the "normal" state. Corrective maintenance can only be performed after failure. Thus the model transitions from "fail" to "correctiveM", and is restarted as new in the "normal" state only after corrective maintenance is completed.

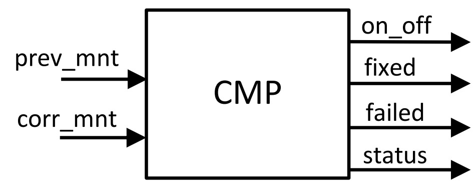

FIGURE 2. CMP block diagram

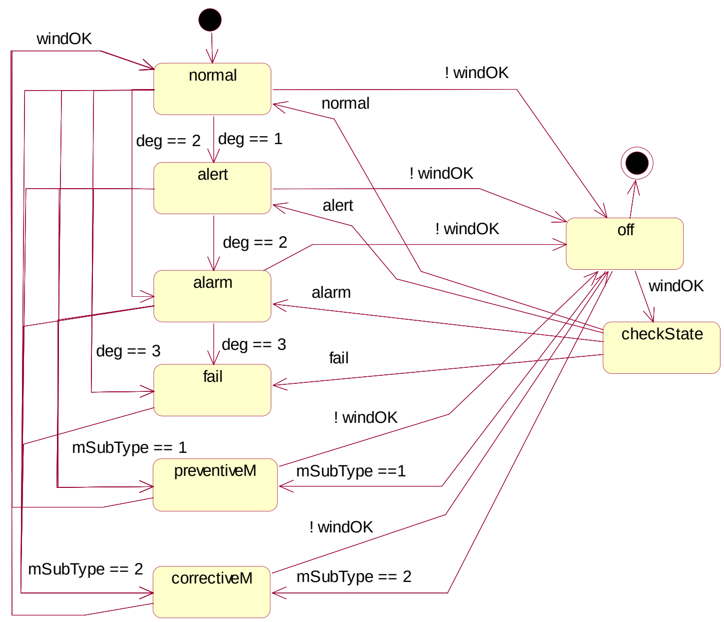

FIGURE 3. CMP state transition diagram

Please note that since we now model four major components within a turbine, there are three other transition matrices, P_1, P_3, and P_4, affiliating with CMP(1), CMP(3), and CMP(4), respectively. The structures of P_1 and P_3 are the same as P_2, namely that they have also four deterioration status and therefore the matrices have a dimension of 4×4. The specific transition probabilities between the states are different, though, for different components, but can still be calculated by using historical operational data. The structure of P_4 (for the control system) is different – it has only two states, either working or failure. As such, the transition matrix P_4 is of dimension 2×2. This different transition matrix is used for the control systems because its hardware is generally reliable and does not exhibit a gradual decline pattern. Rather sudden failures are the predominant failure mode observed for the control system.

In terms of maintenance scheduling, we considered the operation of the wind farm model under two maintenance strategies: scheduled maintenance (SM) and condition-based monitoring (CBM). Under SM, maintenance preventive actions are scheduled before hand and are performed twice a year. In addition, corrective maintenance corrective actions are performed for unanticipated breakdowns. CBM strategy conducts maintenance preventive actions only when condition monitoring equipment report alarm signals and corrective actions when unanticipated breakdowns occur.

APPLICATION

The simulation model was implemented in DEVS-JAVA which is a java based platform for DEVS. The model simulated a 100-unit wind farm located in West Texas. The wind behavior was modeled using the parameters obtained from the West Texas Mesonet [25].

A computational study was performed to gain management insight regarding wind farm operations and maintenance. Our experiments compared the two maintenance strategies using four performance measurement: systems availability, power generated, capacity factor, and number of failures. The capacity factor is a measurement of the productivity of the wind farm which compares the wind farm production over a given period of time with amount of power the wind farm would have produced at full capacity under ideal condition over the same period of time. We run twenty simulation replications for each maintenance strategy using a time horizon of twenty years. Twenty years is the estimated average lifespan of a wind turbine.

Table 1 reports the simulation results for the average power generation and capacity factor for both maintenance strategies SM and CBM. The mean, standard deviation (Std. Dev.), and 95% confidence interval bounds are reported for each performance measure. The table shows that CBM provides on average a 6.78% higher power generation and capacity factor for the 20-year period compare to SM. In terms of computational run time, the simulation for both maintenance strategies lasted about 0.9 hours on average for a 20-year planning horizon. The experiments were conducted on a Dell X5355 computer with 2 Intel(R) Xeon(R) X processors at 2.66 GHz each with 12.0 GB of RAM.

The average power generated per year for both SM and CBM is depicted in Figure 4. Under CBM the system is able to generate more power every year. The figure shows an initial decreasing pattern for both maintenance strategies. The highest power generation for both strategies occur on the first year because all wind turbines are new and maintenance interruption are limited. CBM achieves a more steady pattern after the fifth year keeping the power generation per year around 640,000 MW.

Table 2 shows the simulation results for the wind farm availability, failures, and maintenance costs under SM and CBM, respectively. The results indicate that CBM provides a 6.75% higher availability of the wind farm for the 20 years time period. The table also shows a lower average number of failures per turbine under CBM. Figure 5 shows the average availability per wind turbine for the 20-year time period. The graphs shows

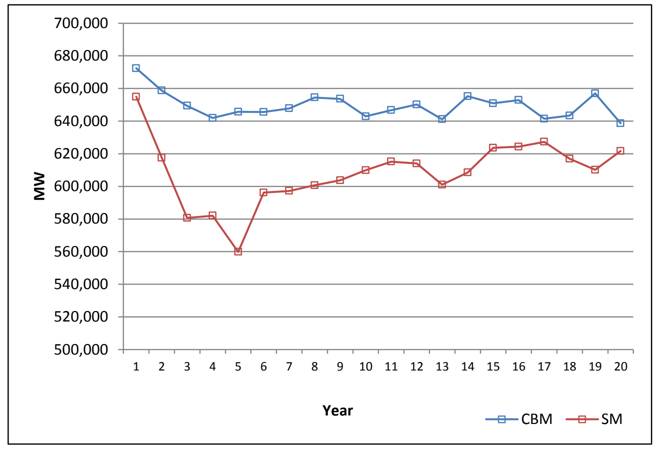

FIGURE 4. Average power generated annually

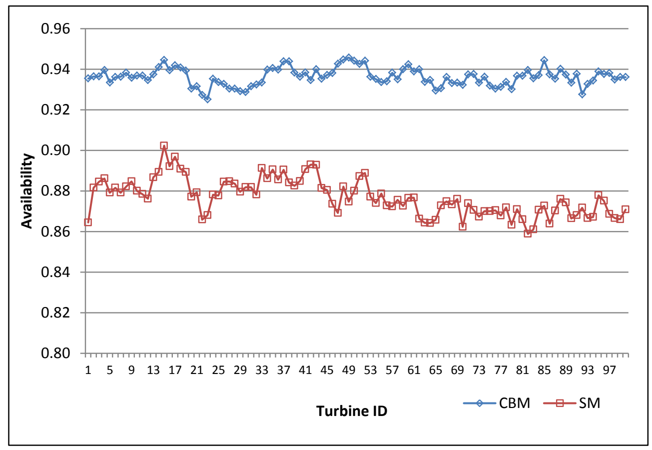

FIGURE 5. Average availability per turbine

that all the turbines were available for a longer period of time under the CBM. The average availability per turbine under CBM was about 94% for the 20-year period. Figure 6 shows the average number of failures per year experienced by the wind farm under both maintenance strategies. An non-decreasing pattern is observed for both maintenance strategies during the first five years. This can be explained by the fact that turbines are new at the beginning of the time period. After five years the average number of failures achieve a steady state and under CBM a lower average number of failures is achieved.

We compared our results to those reported in the literature in Table 3. It seems that our capacity factor are slightly higher than the industry upper bound, and our availability is slightly lower. We believe this difference is caused by still a number of simplifications used in our simulation, for instance, we used the ideal (published) power curve to calculate the power generation under a given wind speed, we so far account for the failures from

TABLE 1. Simulation results for power generation and capacity factor

	SM		CBM	
	Mean	StDev	Mean	StDev
Generated Power (MW)	12,165,137.72	8,701.08	12,989,792.63	7,688.64
Capacity Factor	0.427	0.001	0.456	0.001
CPU Time (secs)	3,290.05	50.55	3,219.47	41.05

TABLE 2. Simulation results for availability and failures

	SM		CBM	
Performance measure	Mean	StDev	Mean	StDev
Availability	0.877	0.001	0.936	0.001
Failures per wind turbine per year	0.545	0.004	0.232	0.011

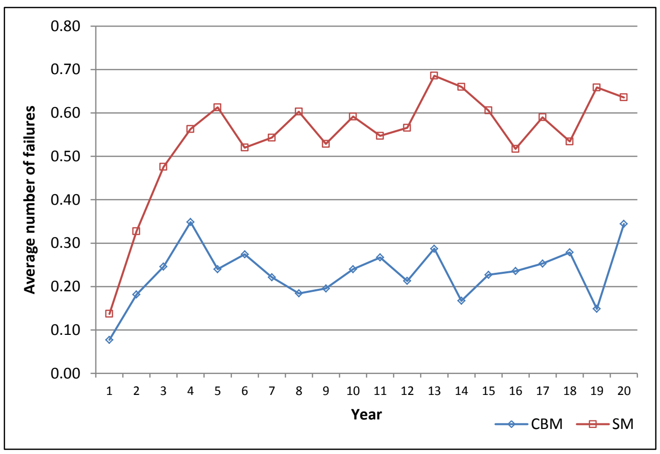

FIGURE 6. Average number of failures per year

only four major turbine components, and we did not consider the possibility of curtailment yet.

DISCUSSION AND CONCLUSIONS

Wind farms contain a group of individual wind turbines which use wind power to generate electricity and provide a source of clean and renewable energy. A commercial wind farm usually accommodates more than one hundred turbine in a remotely located area with high speed winds throughout the year. Due to rough weather conditions and seasonal variations, wind turbines experience stochastic forces that lead to components degradation and failures which lead to costly repairs. Consequently, scheduling maintenance actions to avoid wind turbine component failures is very critical.

In this paper we present a DEVS wind farm simulation model. This simulation consider the degradation of multiple component within the wind turbine model and consider multiple turbines. The simulation allows for computing several performance measures such as power generation, capacity factor, wind farm availability, and failure rates. The simulation model provides a useful tool for selecting effective operation and maintenance actions based on scheduled maintenance (SM) and condition-based maintenance (CBM), respectively. Computational results based on a real setting using historical wind data are consistent with the values encountered in practice. Our computational study shows that CBM provides on average 6.78% higher generation and capacity factor over a 20-year period compared to SM.

The extra power generated presents the benefit of using CBM for wind farm maintenance. Of course, people need to be aware that using CBM is usually a little more expensive. So the ultimate economic benefit will have to be decided when the extra cost is also accounted for. This cost-benefit analysis can be done as long as the cost data are available. So far the type of data is scarce in the public domain or in the literature. We hope that this analysis can be addressed in the future work.

REFERENCES

[1] Byon, E., Ntaimo, L., and Ding, Y. "Optimal maintenance strategies for wind power systems under stochastic weather conditions". *IEEE Transactions On Reliability*, ***59***, pp. 393–404.

[2] Byon, E., and Ding, Y., 2009. "Season-dependent condition-based maintenance for a wind turbine using a partially observed Markov decision process". *IEEE Trans-*

TABLE 3. Figures in industry

Criteria	Figures in industry	SM Mean	SM Std. Dev.	SM CI L	SM CI U	CBM Mean	CBM Std. Dev.	CBM CI L	CBM CI U
Capacity factor	0.25-0.40	0.43	0.001	0.43	0.43	0.46	0.001	0.46	0.46
Availability	0.98	0.88	0.001	0.88	0.88	0.94	0.001	0.94	0.94
Failures per year	0.05-2.29	0.55	0.004	0.55	0.55	0.23	0.011	0.23	0.24

actions on Power Systems, ***25***, pp. 1823 – 1834.

[3] Zeigler, B., Kim, T., and Praehofer, H., 2000. *Theory of Modeling and Simulation*. Academic Press, Inc. Orlando, FL, USA.

[4] Zeigler, B., and Sarjoughian, H., 2003. *Introduction to DEVS Modeling & Simulation with JAVA: Developing Component-Based Simulation Models*. Tucson/Tempe, Arizona, USA.

[5] Jonkman, B. J., 2009. *TurbSim User's Guide: version 1.50*. Golden CO.

[6] Jonkman, B. J., and Buhl Jr., M. L., 2005. *FAST User's Guide*. Golden CO.

[7] Larsen, T., and Hansen, A. M., 2007. *How 2 HAWC2, the user's manual*. Risø-R-1597, Technical Report Risø National Laboratory, Roskilde, Denmark.

[8] Ribrant, J. "Reliability performance and maintenance - a survey of failures in wind power systems". Master's thesis.

[9] Vittal, S., and Teboul, M., 2005. "Performance and Reliability Analysis of Wind Turbines Using Monte Carlo Methods Based on System Transport Theory". In Proceedings of the 46th AIAA Structures, Structural Dynamics and Materials Conference.

[10] Basumatary, H., Sreevalsan, E., and Sasi, K. K., 2005. "Weibull parameter estimation-a comparison of different methods". *Wind Engineering,* ***29***, pp. 309–316.

[11] Andrawus, J., Watson, J., and Kishk, M., 2007. "Modeling system failures to optimise wind turbine maintenance". *Wind Engineering,* ***31***(6), pp. 503–522.

[12] Hahn, B., Durstewitz, M., and Rohrig, K., 2007. *Reliability of wind turbines, experiences of 15 years with 1,500 WTs in wind energy*. Springer, Berlin.

[13] Jiang, W., Yan, Z., and Feng, D., 2009. "A review on reliability assessment for wind power". *Renewable and Sustainable Energy Reviews,* ***13***, pp. 2485 – 2494.

[14] Coolen, F. P. A., 2010. "On modelling of grouped reliability data for wind turbines". *IMA Journal of Management Mathematics,* ***21***, pp. 363–372.

[15] Tavner, P. J., Edwards, C., Brinkman, A., and Spinato, F., 2006. "Influence of wind speed on wind turbine reliability". *Wind Engineering,* ***30***, pp. 55 – 72.

[16] Tavner, P. J., Xiang, J., and Spinato, F., 2007. "Reliability analysis for wind turbines". *Wind Energy,* ***10***, pp. 1–8.

[17] Guo, H., Watson, S., Tavner, P., and Xiang, J., 2009. "Reliability analysis for wind turbines with incomplete failure data collected from after the date of initial installation". *Reliability Engineering & System Safety,* ***94***, pp. 1057 – 1063.

[18] McMillan, D., and Ault, G., 2008. "Condition monitoring benefit for onshore wind turbines: sensitivity to operational parameters". *IET Renewable Power Generation,* ***2***(1), pp. 60–72.

[19] Rademakers, L., Braam, H., and Verbruggen, T., 2003. R and d needs for o and m of wind turbines. Tech. Rep. *ECN-RX–03-045*, *ECN Wind Energy*.

[20] Rademakers, L., Braam, H., Zaaijer, M., and van Bussel, G. Assessment and optimisation of operation and maintenance of offshore wind turbines. Tech. rep.

[21] Foley, J. T., and Gutowski, T. G., 2008. "Turbsim: reliability-based wind turbine simulator". In Proceedings of the 2008 IEEE International Symposium on Electronics and the Environment, pp. 1–5.

[22] Nilsson, J., and Bertling, L., 2007. "Maintenance management of wind power systems using condition monitoring systems-life cycle cost analysis for two case studies". *IEEE Transactions on Energy Conversion,* ***22***(1), pp. 223–229.

[23] Marmidis, G., Lazarou, S., and Pyrgioti, E., 2008. "Optimal placement of wind turbines in a wind park using Monte Carlo simulation". *Renewable Energy,* ***33***, pp. 1455 – 1460.

[24] Byon, E., Pérez, E., Ding, Y., and Ntaimo, L., 2011. "Simulation of wind farm operations and maintenance using DEVS". *Simulation - Transactions of the Society for Modeling and Simulation International,* ***87***, pp. 1093–1117.

[25] West Texas Mesonet, 2008. http://www.mesonet.ttu.edu/.