

SIMULATION

<http://sim.sagepub.com/>

Simulation of wind farm operations and maintenance using discrete event system specification

Eunshin Byon, Eduardo Pérez, Yu Ding and Lewis Ntaimo

SIMULATION published online 16 July 2010

DOI: 10.1177/0037549710376841

The online version of this article can be found at:

<http://sim.sagepub.com/content/early/2010/07/15/0037549710376841>

Published by:



<http://www.sagepublications.com>

On behalf of:



[Society for Modeling and Simulation International \(SCS\)](#)

Additional services and information for *SIMULATION* can be found at:

Email Alerts: <http://sim.sagepub.com/cgi/alerts>

Subscriptions: <http://sim.sagepub.com/subscriptions>

Reprints: <http://www.sagepub.com/journalsReprints.nav>

Permissions: <http://www.sagepub.com/journalsPermissions.nav>



Simulation of wind farm operations and maintenance using discrete event system specification

Eunshin Byon, Eduardo Pérez, Yu Ding and Lewis Ntaimo

Abstract

Wind farms provide a source of clean and renewable energy. However, unlike many industries where machines are operated under more or less static conditions, wind turbines suffer from stochastic loading due to the hourly or seasonal variation of wind speed and direction. The stochastic loading of wind turbines makes their degradation or failure prediction rather complex. This in turn makes the decision-making process of when and what type of maintenance action to undertake very challenging. This paper uses the discrete event system specification (DEVS) to develop a simulation model for wind farm operations and maintenance. The DEVS methodology¹ provides a formal modeling and simulation framework based on dynamical systems theory and allows for hierarchical and modular model construction. We report on implementation results based on historical data that provide useful insights into wind farm operations under two different maintenance strategies, scheduled maintenance and condition-based maintenance. The results show that condition-based maintenance enables more wind power generation by reducing wind turbine failure rates and thus increasing wind turbine available.

Keywords

DEVS, renewable energy, wind turbine, scheduled maintenance, condition-based maintenance

1. Introduction

A wind farm is a grouping of wind turbines used to generate electricity and is usually located at a site having dependable strong winds to maximize power generation. Sometimes wind turbines are stopped during periods of strong wind to prevent catastrophic failure. In the case of wind turbine failure, access to the wind farm can be difficult during harsh weather periods such as winter or storm seasons.² Offshore wind turbines are largely affected by stochastic wave climate. The stochastic loading or stress on wind turbines may cause the degradation pattern of each wind turbine to be different. However, most maintenance optimization models in the literature and existing wind farm simulation models are based on average degradation (failure) behavior. Therefore, this problem calls for new models to assist in wind farm operations and maintenance (O&M) decisions to minimize the operational costs while maintaining acceptable levels of power generation.

The mainstream of the current maintenance practice for wind farms is scheduled maintenance (SM). The frequency of SM depends largely on the manufacturer's

recommended maintenance program. However, in general wind farm operators carry out SM about twice a year. Despite the regular SM, wind farm operators still have to respond to unanticipated breakdowns, which require unscheduled or corrective maintenance. Unexpected breakdowns usually account for a significant portion of downtime. Considering today's trend toward large-scale wind farms and the long distances from the operation and monitoring centers, wind farm operators want to avoid unnecessary visits by detecting and fixing the problems before any failure occurs. In this regard, many manufacturers have recognized the benefits of condition-based maintenance (CBM) and, accordingly, most modern turbines are equipped with

Department of Industrial and Systems Engineering Texas A&M University
College Station, Texas, USA.

Corresponding author:

Lewis Ntaimo, Department of Industrial and Systems Engineering, Texas
A&M University, 3131 TAMU, College Station, TX 77843, USA
Email: ntaimo@tamu.edu

condition monitoring equipment with many sensors to help detect incipient catastrophic failures. Furthermore, condition monitoring equipment enables more effective maintenance planning and reduces the downtime involved in carrying out maintenance activities.

A viable approach towards addressing the difficult problem of wind farm O&M planning is modeling and simulation (M&S). In this work we use the discrete event M&S approach to build a generic simulation model for a wind farm for predicting wind turbines' states and assessing the maintenance actions. The simulation model is composed of all of the critical component models for a wind farm: a wind turbine model with a degradation model and power generation model, a condition monitoring sensor model, a component state evaluation model, a wind model, and a maintenance model.

We use the *discrete event system specification* (DEVS) formalism¹ to derive the models that can be tailored to any real wind farm. DEVS is a formal M&S framework based on dynamical systems theory. We chose to use DEVS because it provides well-defined concepts for coupling components, hierarchical and modular model construction and an object-oriented substrate supporting repository reuse. Furthermore, the modular construction ability of DEVS allows the modeler to design and construct each model independently for optimum efficiency. The models can interact with each other by adhering to well-defined protocols.

Since the simulation model we propose includes many critical aspects of wind farm operations, a broad array of potential applications can be developed on this simulation platform. For example, the simulation can be used for studying wind farm site viability,³ grid connection, evaluation of the generation adequacy of wind power systems,⁴ wind power system reliability analysis,^{5,6} etc.

The rest of this paper is organized as follows. In Section 2 we review closely related work. In Section 3 we discuss model abstraction for wind farm simulation. We present DEVS models of wind farm simulation components in Section 4. We also present the overall hierarchical structure and operation of the wind farm simulation. We discuss a computer implementation and application of the simulation model to a real setting in Section 5 and end the paper with some concluding remarks and directions for further research in Section 6.

2. Related work

There have been several simulation studies regarding wind farm operations. First, we present the simulation studies directly related to wind farm O&M. Rademakers et al.⁷ describe a Monte Carlo simulation model for operations and maintenance of offshore wind farms, developed by Delft University of Technology (TU-Delft). They illustrate the features and benefits of

the model by a case study of a 100 MW wind farm. The model simulates the operations and maintenance aspects over a period of time by considering several critical factors for performing repair actions such as turbine failures and weather. The failures of turbine components are generated stochastically, based on the relevant statistics such as mean-time-to-failure (MTTF) and reliability distributions. Also, weather conditions are realized with the given summer and winter storm percentages at the specific site. The model only considers corrective maintenance, and the simulation results indicate that the revenue losses account for 55% of the total maintenance costs, mainly due to the long lead time to prepare parts and the long waiting time until favorable weather conditions are met for repair jobs. Similar studies can be found in the works of Rademakers et al.,² Van Bussel⁸ and Hendriks et al.⁹

The above-described models do not consider a degradation status for each component of a wind turbine. However, Mcmillan and Ault¹⁰ use a Monte Carlo Simulation to quantify the cost-effectiveness of condition monitoring equipment, and compare the performance of two contrasting maintenance policies, that is, SM and CBM. They employ several probabilistic models to introduce uncertainties. For example, Markov models are used to represent component degradation behavior. In their simulation model, it is assumed that the condition monitoring equipment exactly reveals the degradation status of each component. They also consider weather constraints when performing repair actions. The results over various scenarios with different wind profiles, downtime durations and replacement costs show the economic benefits of CBM versus SM for onshore turbines.

Simulation is also used for evaluation of various O&M approaches. Andrawus et al.¹¹ suggest the optimal replacement time for each component of a wind turbine using statistical approaches and evaluate the suggested strategy by Monte Carlo simulation. In their study, the Weibull distribution is used to model the failure pattern of each component before deciding on the optimal replacement cycle for each component. According to their case studies of 600 kW horizontal axis turbines, to minimize total maintenance costs, the gearbox should be replaced every six years and the generator every three years. They assess the reliability, availability and maintenance costs by simulating a wind farm with 26 turbines over a period of four years using a commercial software called ReliaSoft BlockSim-7.¹² Similarly, Hall and Strutt¹³ also develop probabilistic failure models for component reliability using Monte Carlo simulation combined with statistical analysis.

There is another group of studies which are not specific to wind farm O&M, but from which we could gain

insights about reliability and cost issues of wind energy under the framework of the whole electric power system. These studies also provide the broad applicability of the presented simulation model to other classes of problems associated with wind power operations. Karki and Billinton⁴ use a Monte Carlo simulation to help determine appropriate wind power penetration in an existing power system from both reliability and economic aspects. The overall power generating system is divided into subsystems of wind turbine generators and conventional generators. In simulating wind speeds to determine the generated power from the wind turbines, they use an autoregressive moving average (ARMA) time series model. A simple failure model is considered in this study, that is, the time to failure of each generator including wind turbines and conventional generators is assumed to be exponentially distributed and the MTTF of each generator is obtained from historical data. Based on the case study on a typical small power generating system, the authors present the procedure to help determine an appropriate wind penetration level in a power system with both reliability and cost criteria. Karki and Patel⁵ later extend the study to determine appropriate transmission line size and evaluate the reliability of the combined wind generation and transmission systems. Several other studies also use Monte Carlo simulation to evaluate the reliability and/or availability of power generation including wind power.^{14,15,16} For a detailed review of recent reliability assessment studies on wind power, we refer the reader to Wen et al.⁶

In addition to the simulation studies, recently several analytical models have been proposed to investigate the advantages of CBM for wind farms. Although these models are not based on simulation, they provide insights on how condition monitoring information can be utilized. Byon et al.¹⁷ examine optimal repair strategies for wind turbines operated under stochastic weather conditions. The problem is formulated as a partially observed Markov decision process for cost-effective sequential decision-making. Several unique factors in wind turbine operations, such as weather effects, long lead time to assemble maintenance crew and parts upon unanticipated failures, and the corresponding revenue losses are considered. Nilsson and Bertling¹⁸ present an asset life cycle cost analysis to discuss the benefits of condition monitoring with a case study of two actual wind farms. They break down the entire maintenance costs into several cost components including corrective maintenance costs, preventive maintenance costs, installment and operating costs of condition monitoring equipment, and then examine the cost reduction by utilizing condition monitoring information.

Discrete event modeling and simulation^{1,19} is an operational research technique that has been used in

the past to study the characteristics of many applications. Despite the benefits of discrete event simulation, the studies using discrete event simulation are scarce in wind farm simulation studies as depicted by our literature review. The contributions of this work include, to the best of our knowledge, the first discrete event simulation model for wind farm operations. The simulation model also belongs to the category of Monte Carlo simulation in a sense that it uses random number generators to characterize the stochastic aspects of wind farm operations. One of the main differences of the previous work from the proposed model is that in the existing models time evolution is not important and the models focus on obtaining estimates for performance measures. In contrast, our model considers dynamic wind turbine state changes as a result of stochastic events to enable us to gain detailed insights into the lifetime evolution of wind turbine systems in addition to gathering performance measures.

The current simulation models are generally oversimplified without sufficient granularity representing wind farm operations. In general, the condition of each wind turbine component is not considered in the literature except in the work of Mcmillan and Ault.¹⁰ Also, most studies assume independence among wind turbines in terms of power generation, while in actuality there is high dependency. This is because there is a spatial correlation of wind speeds at wind turbine sites and wind turbines at different elevations produce different amounts of power. The morphology of the terrain where wind farms are located should also be considered in wind farm simulations. Our simulation model considers all the above-mentioned factors and represents actual wind farm operations with sufficient details and provides interesting insights into the stochastic behavior of wind power systems. Detailed descriptions of the simulation model are presented in the next section.

3. Simulation model abstraction and performance measures

In order to build a simulation model of a wind farm, we need to consider all the critical components associated with wind farm operations and maintenance. This includes a model of the condition-based monitoring system for the wind farm. In this section, we address model abstraction and performance measures for wind farm operations and maintenance, and derive the DEVS models of the components in Section 4.

3.1. Model abstraction

We consider model abstraction for the following wind farm operations and maintenance models: a power generation model, a wind speed model, a wind turbine with

component degradation model, a sensor model, a smart sensor model, a state evaluation model, and a maintenance model.

Power generation model. Power generated from wind turbines mainly depends on wind speed and can be calculated using a power curve as shown in Figure 1. Wind turbines are designed to start generating power at the cut-in wind speed V_{ci} . The power output increases nonlinearly as the wind speed increases from the cut-in wind speed to the rated wind speed V_r . However, at higher wind speeds than the cut-out wind speed V_{co} turbines are shut down to avoid damage to the structure due to excessive mechanical loads. These parameter values depend on the type of wind turbine and are usually specified by the wind turbine manufacturer.

According to Karki and Patel,⁵ the mathematical relationship between the wind speed V and the generated power P can be given as

$$P = \begin{cases} 0, & \text{if } 0 \leq V < V_{ci} \\ P_r(a + b \cdot V + c \cdot V^2) & \text{if } V_{ci} \leq V < V_r \\ P_r, & \text{if } V_r \leq V < V_{co} \\ 0, & \text{if } V_{co} \leq V \end{cases} \quad (1)$$

where P_r is the rated power output of the wind turbine. The parameters a , b , and c in Equation (1) are obtained from the following equations:

$$a = \frac{1}{(V_{ci} - V_r)^2} \left[V_{ci}(V_{ci} + V_r) - 4V_{ci}V_r \left(\frac{V_{ci} + V_r}{2V_r} \right)^3 \right] \quad (2)$$

$$b = \frac{1}{(V_{ci} - V_r)^2} \left[4(V_{ci} + V_r) \left(\frac{V_{ci} + V_r}{2V_r} \right)^3 - (3V_{ci} + V_r) \right] \quad (3)$$

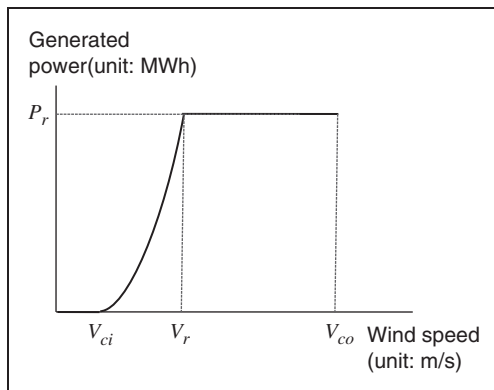


Figure 1. Power curve.⁵

$$c = \frac{1}{(V_{ci} - V_r)^2} \left[2 - 4 \left(\frac{V_{ci} + V_r}{2V_r} \right)^3 \right] \quad (4)$$

Wind speed model. When anemometers are installed inside turbines, wind speed models at turbine locations can be built using temporal models. However, when wind speeds are measured at stations near wind turbine sites, the spatial correlations between wind speeds at the stations and those at the wind turbine sites have to be considered. The latter case happens when evaluating a new wind farm site where the wind speeds are not available but the wind speeds at the stations near the target wind farm site are available. This is also true when wind turbines do not have anemometers, which can be encountered often in old, small-sized wind turbines. A spatio-temporal model that considers both temporal and spatial variations of wind speeds is required in this case.

In this paper we develop a spatio-temporal model in a hierarchical manner to generate synthetic sequences of wind speeds at turbine locations. This hierarchical construction makes the model flexible so that when the wind speeds are available at the turbine site one only needs the temporal model. The generated sequences should represent possible realizations of wind speeds at the turbine sites. Therefore, each of the components included in a spatio-temporal model has a physical interpretation and includes diurnal cycle, yearly seasonality, and the spatial variations among different locations.

First, a time series model is constructed at each station which measures wind speeds on a regular basis. In the model, diurnal cycle and yearly seasonality are modeled using Fourier series.^{20,21} Let us denote the wind speed at time t at station x_i , $i=1, 2, \dots, I$, by $S_t(x_i)$. Then, $S_t(x_i)$ at each station, $i=1, \dots, I$, can be modeled as follows:

$$S_t(x_i) = \alpha_0^i + \sum_{d=1}^{n_d} \left[\beta_{1d}^i \sin\left(d \frac{2\pi t}{\omega_d}\right) + \beta_{2d}^i \cos\left(d \frac{2\pi t}{\omega_d}\right) \right] + \sum_{s=1}^{n_s} \left[\gamma_{1s}^i \sin\left(s \frac{2\pi t}{\omega_s}\right) + \gamma_{2s}^i \cos\left(s \frac{2\pi t}{\omega_s}\right) \right] + \epsilon_t^i \quad (5)$$

In Equation (5), the first term α_0^i is the average wind speed and the second term $\beta_{1d}^i \sin(d2\pi t/\omega_d) + \beta_{2d}^i \cos(d2\pi t/\omega_d)$ is known as the d th harmonic to represent the diurnal cycle, where ω_d is the number of wind speed measurements during a day.²² Similarly, s in the third term $\gamma_{1s}^i \sin(s2\pi t/\omega_s) + \gamma_{2s}^i \cos(s2\pi t/\omega_s)$ is the s th yearly harmonic to represent yearly seasonality, where ω_s is the number of measurements during a year. In many environmental studies, n_d and n_s are typically set^{20,22,23,24} to values of 1 or 2. The parameter ϵ_t^i in (5) is a stochastic

component of the time series model, which can be formulated as the following autoregression and moving average (ARMA) model:

$$\epsilon_t^i - \sum_{p=1}^P \phi_p^i \epsilon_{t-p}^i = a_t^i - \sum_{q=1}^{Q-1} \theta_q^i a_{t-q}^i \quad (6)$$

where a_t^i is an independent and identically distributed random variable with distribution $N(0, \sigma_i^2)$. The appropriate degree (P, Q) of the ARMA model can be selected by the Akaike information criterion (AIC) or the Bayesian information criterion (BIC).²⁵

The wind speed at a turbine site x at time t , denoted $W_t(x)$, can be generated considering the spatial correlations between the stations and wind turbine sites, using a spatial model, called *Kriging*.²⁶ Suppose that wind speeds at the stations $S_t(x_1), \dots, S_t(x_I)$ are simulated (or observed) at spatial locations x_1, \dots, x_I at time t . Then, the wind speed at a wind turbine site x at time t is given by

$$W_t(x) = \sum_{j=1}^J \varphi_j f_j(x) + r^\top R^{-1} (S_t - \varphi \cdot e) \quad (7)$$

where $f_j(x)$, $j = 1, \dots, J$, are known regression functions taking spatial locations x as input. In this study, simple Kriging is used with $\sum_{j=1}^J \varphi_j f_j(x) = \varphi$, which is constant. $S_t = [S_t(x_1), \dots, S_t(x_I)]^\top$ is the $I \times 1$ matrix, each element of which denotes a wind speed at each station at time t , and $e = [1, 1, \dots, 1]^\top$ is a $I \times 1$ vector. R is the correlation matrix between the wind speeds at the stations, whose (k, l) th element is as follows:

$$R(x_k, x_l) = \exp\left\{-\sum_{u=1}^U |(x_{ku} - x_{lu})/\nu_u|^\psi\right\} \quad (8)$$

Here, U is the dimension of x_i , $i = 1, \dots, I$, and $R(k, l)$ denotes the correlation function between two locations, $x_k = [x_{k1}, \dots, x_{kU}]^\top$ and $x_l = [x_{l1}, \dots, x_{lU}]^\top$, $k, l = \{1, \dots, I\}$. ψ is the shape parameter representing the correlation function shape, whereas ν_u is the scale parameter denoting the effect of distance in each dimension u , $u = 1, \dots, U$. Finally, $r = [R(x, x_1), \dots, R(x, x_I)]^\top$ in (7) is a $I \times 1$ dimensional vector, each element of which reflects the correlation between the wind turbine site and the station.

Note that $W_t(x)$ in (7) is the estimated wind speed at the anemometer height. If the anemometer height does not coincide with the turbine hub height, the wind speed has to be scaled based on the turbine hub height. There are different formulas for the wind speed adjustment,^{27,28} among which the following

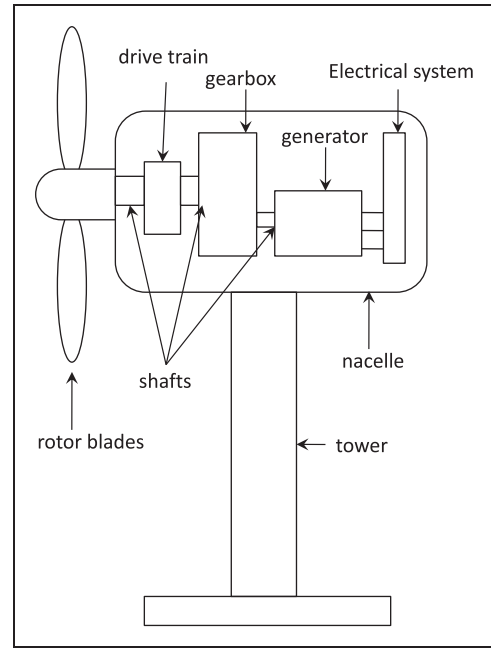


Figure 2. Wind turbine components.³⁰

wind power law has been recognized as a useful tool to transfer the anemometer data to the desired hub center in many studies:^{27,29}

$$W_t^H(x) = W_t(x) \left(\frac{z}{z_0}\right)^\alpha \quad (9)$$

where $W_t^H(x)$ is the wind speed at anemometer height z at turbine location x and $W_t(x)$ is the wind speed at hub height z_0 from Equation (7). The parameter α is a wind speed power law coefficient, whose value mainly depends on the local geographical terrain.

Wind turbine components with degradation model.

Although there are different designs of wind turbines, they share basic common features and the names of the components are general.³⁰ Figure 2 shows the different components of a wind turbine.^{30,31} A wind turbine consists of a tower, two or three-bladed rotors, and a nacelle which houses several critical components such as the drive train, gearbox, generator, and the electrical system.

There are several mathematical models to represent component degradation. In this study, a Markov model is applied because of its flexibility and popularity in many applications including the modeling of the devices used in power systems.^{17,32} In a Markov model, the degradation status, or system condition, is divided into a finite number of states, $1, 2, \dots, J$, where 1 indicates the best condition and J is the worst deteriorated operational status. Conceptually, J can be any number. In this study, four different states are considered, namely, 'normal', 'alert', 'alarm', and 'failed'. In a

Markov model, the probability that the system condition moves from the current state to any other state during a fixed time is assumed to be constant. Suppose that the transition probability from state i to state j is p_{ij} . Then, the transition matrix $P = [p_{ij}]_{i,j=1,\dots,J}$ represents the Markovian deterioration process of the component.

Sensor model. There are several techniques to detect abnormal system behavior based on sensor information, among which vibration analysis is the dominant condition monitoring technique used for gearbox fault detection.³³ Other techniques include debris analysis, oil condition analysis, etc. Although these techniques have been proved to be reliable and accurate in fault diagnosis, there are still problems of measurement noises, and the problem that a specific value of monitoring data could be coming from different conditions of the target system.^{17,34} More importantly, the fault diagnosis based on sensor measurements is non-trivial because wind turbines operate under non-steady and irregular operating conditions.¹⁷ Therefore, often one cannot conclude the exact state of the system from sensor measurements, but should estimate the state in a probabilistic sense.

The hidden Markov model (HMM) has been widely used to reflect the incomplete information from ‘a sensor well’³² and, thus, HMM is adopted in this study. In a regular Markov model, the state is directly visible to the observer. However, in a HMM, the state is not directly visible, but output (here, sensor output) dependent on the state is visible. Suppose that one can categorize the sensor results into a finite number of outputs, $k = 1, 2, \dots, K$. Then, the probability of getting sensor output k when the current state is j is $b_j(k)$, where $\sum_{k=1}^K b_j(k) = 1$ for each j . Therefore, the sequence of sensor outputs generated by a HMM gives information about the system degradation in a probabilistic sense. Figure 3 illustrates the basic idea of HMM.

State evaluation model. A system state can be evaluated based on sensor streams. Let us denote the current

system state by Z_t at time t . Based on sensor streams, the system state Z_{t_0} can be estimated using the *Viterbi* algorithm.³² Given the sequences of sensor outputs, k_1, \dots, k_t over time up to time t , the algorithm proceeds as follows:

1. *Initialization:* Set $\alpha_1(j) = \pi_j b_j(k_1)$, $\forall j \in \{1, 2, \dots, J\}$.
2. *Recursion:* For, $2 \leq n \leq t$ compute $\alpha_n(j) = \max_i \{\alpha_{n-1}(i) p_{ij} b_j(k_n)\}$, $\forall j \in \{1, 2, \dots, J\}$.

Then, the estimated current state \hat{Z}_t is given by

$$\hat{Z}_t = \operatorname{argmax}_{j \in \{1, 2, \dots, J\}} \{\alpha_t(j)\} \quad (10)$$

Smart sensor model. In general, the above-described sensors are relatively cheap but may be unreliable. In practice, there are different means to evaluate the exact deterioration level of a component of a system. The most common way is to dispatch a maintenance crew to conduct an on-site inspection. With emerging of new sensor technology, more advanced sensors (e.g., high-speed imaging or mobile acoustic sensors) can be invoked to carry out a follow-up investigation upon the initial sounding of alerts or alarms. Regardless of which mechanism is used, the follow-up investigation is usually much more expensive (and more accurate). Collectively, we refer to this observation mode as the ‘Smart Sensor’. We assume that once a ‘Smart Sensor’ is invoked, the system state can be revealed with certainty. In our model, we set the cost to invoke smart sensors to be about 10% of the corrective maintenance costs according to the suggestions of our industry partners.

Maintenance model. Two different maintenance approaches are implemented in the simulation model: *scheduled maintenance* (SM) and *condition-based maintenance* (CBM). We consider an SM model that reflects current maintenance practices. Wind farm operators usually carry out SM twice a year in low wind speed seasons.³⁰ Following the industry practice, SM is set to

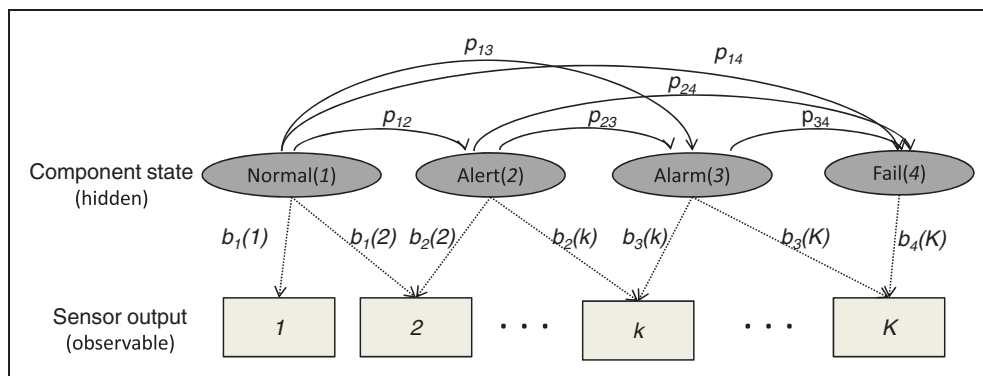


Figure 3. Illustration of the hidden Markov model.

be performed in the Spring and Fall seasons in our simulation model. The second model is a CBM model, which implements the condition-based preventive maintenance strategy. Modern wind turbines are equipped with automated alarm call-out systems inside condition monitoring equipment so that when a sensor signal exceeds a certain threshold an alarm is sent to a wind farm operator by fax, pager, simple message service (SMS), or email.³⁰ In our simulation model, when the estimated state of a component is the *alarm* state, the alarm message is sent to the operation center. Then, *smart sensors* are invoked to evaluate the system conditions exactly. When the actual system state that the smart sensors report is equal to the *alarm* state, then preventive maintenance is carried out. In both maintenance strategies, unplanned failures are not completely avoidable. When a wind turbine fails before SM or preventive maintenance, wind farm operators carry out corrective maintenance. In this case, one should consider a substantial lead time for organizing maintenance crews and spare parts. It can take several weeks for a component such as a gearbox to be delivered.³⁵ This considerable lead time after a failure significantly affects the downtime and, accordingly, revenue losses, because a failed turbine cannot generate power. Furthermore, the cost for corrective maintenance is much higher than that for preventive maintenance.²

Sometimes maintenance actions are constrained by adverse weather conditions.^{10,30} For example, during wind speeds higher than 10 m/s, lifting the roof of the nacelle is not allowed.¹⁰ Even climbing up the turbines is not allowed under high wind speeds of more than 20 m/s.¹⁰ Therefore, repair actions cannot be carried out under these harsh weather conditions. Also, when the weather becomes adverse after the crew has started repairing actions, the crew must stop work and wait until weather conditions become favorable. These delays cause revenue losses because wind turbines cannot be operated until the repairs are completed. In our simulation model, we assume that maintenance actions are carried out only when wind speeds are less than 10 m/s.

3.2. Performance measures

There are four performance measures commonly considered in the literature for evaluating wind farm O&M strategies. The first performance measure is the *capacity factor*. The capacity factor quantifies the productivity of a wind turbine.³¹ It compares a wind turbine's (or a wind farm's) actual production over a given period of time with the amount of power the turbine (or wind farm) would have produced if it had run at full capacity for the same amount of time. Let P_{actual} denote the actual amount of power produced over a given time period and let P_{ideal} denote the amount of power that would have been produced at full capacity. Then capacity factor can be expressed as the ratio $P_{\text{actual}}/P_{\text{ideal}}$. Capacity factors are typically in the range 0.25 to 0.40, although higher capacity factors may be achieved during windy seasons.³¹

The second performance measure is *availability* of wind turbines. Availability refers to the percentage of time a wind turbine is available to generate power and is not out of service or under repair. The third performance measure is *number of failures* in a given time period. This is an important factor for measuring the reliability of a wind turbine. The fourth performance measure is *maintenance cost*. In the simulation average maintenance cost over a given time period is computed to quantify the effectiveness of the two maintenance strategies, SM and CBM. Maintenance cost accounts for a significant portion of the total wind farm operational cost.² The four performance measures are summarized in Table 1.

4. DEVS atomic and coupled models

Let us now turn to deriving DEVS atomic and coupled models for the DEVS wind farm simulation. These models provide the basic building blocks for the overall wind farm simulation model. We chose the DEVS formalism to build models that can be tailored to any wind farm due to DEVS's well-defined concepts for coupling components, hierarchical and modular

Table 1. Performance measures for the wind farm simulation model

Performance measures	Description
1. Capacity factor	Ratio of the amount power generated by the wind farm and the power generated when it is at full capacity over a given period
2. Availability	Percentage of time the wind farm is available for generating power over a given period
3. Number of failures	Number of times a turbine fails over a given period
4. Maintenance cost	Cost of performing maintenance over a given period

model construction, and its object-oriented approach supporting repository reuse. We follow a bottom-up approach by first deriving atomic models, coupling them to create coupled models, and then coupling the coupled models to create an overall simulation model. Before presenting the models, we first provide some preliminaries on DEVS. The reader familiar with DEVS can skip Section 4.1 without loss of continuity in the discussion.

4.1. DEVS preliminaries

Parallel DEVS¹ follows a hierarchical approach to building complex models from the basic models called *atomic* models. DEVS has a well-defined concept of component coupling. Atomic models are coupled to form *coupled* (composite) models. In DEVS coupled models are treated as components via the property of *closure under coupling*, which enables the hierarchical model composition construct. An atomic model possesses *input* and *output ports* through which all interactions with the environment are mediated and it has to be in a defined state at any given time. The input ports allow the atomic model to receive external *events* arising outside the model. Coupling is done by joining the output of one atomic model to the input of another to enable communication between models. The description of the internal workings of the model determines how the model responds to external events. The internal events arising within the model can change its state and manifest themselves as events on the output ports to be transmitted to other models. Parallel DEVS allows for all imminent atomic models to be activated so that they can send their outputs to other components of the system. We now provide the mathematical description of a parallel DEVS model.

Let us denote by M an atomic model with a set of input ports $IPorts$, a set of input values (events) X_p , a set of output ports $OPorts$, and a set of output values (events) Y_p . Let (p, v) denote the port-value pair. Then a basic *parallel DEVS* is a structure defined as follows:

DEFINITION 4.1.

$$DEVS = (X_M, Y_M, S, \delta_{ext}, \delta_{int}, \delta_{con}, \lambda, ta)$$

where

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

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

S is the set of sequential states;

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

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

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

$\lambda : S \rightarrow Y_M^b$ is the *output* function;

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

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

Parallel DEVS has the ability to handle multiple inputs and uses a *bag* to store the inputs. A bag is a set with possible multiple occurrences of its elements.

According to Definition 4.1, at any time the system is in some state s and if no external events occur the system will remain in its current state for a time $ta(s) \in [0, \infty]$. When this time expires the system outputs the value $\lambda(s)$, and transitions to a state $s' = \delta_{int}(s)$. In DEVS an output is only possible after an internal transition. If an external event $x \in X_M$ occurs when the system is total state (s, e) with $e \leq ta(s)$, the system changes 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 the new state when no events occur since the last transition. The confluent function decides the next state in cases when there is an external event exactly when an internal transition has to occur.

To construct models from components (DEVS models), the DEVS specification includes the external interface, the components, and the coupling relations. Let EIC , EOC , and IC denote the external input coupling, external output coupling, and internal coupling, respectively. Then a coupled model N can be defined mathematically as follows:

DEFINITION 4.2.

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

where

$$X = \{(p, v) \mid p \in IPorts, v \in X_p\}$$

is the set of input ports and values and

$$Y = \{(p, v) \mid p \in OPorts, v \in Y_p\}$$

is the set of output ports and values. 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

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

and

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

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

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

The external output coupling, *EOC*, connects external outputs to component outputs:

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

Lastly, the internal coupling, *IC*, connects component outputs to component inputs:

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

Finally, 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.

4.2. Atomic models

The DEVS wind farm simulation model we propose comprises the models abstracted in Section 3.1. Specifically, we derive the following atomic models: Power Generator (PWRGEN), Component Degradation (CMPDEG), Wind Generator (WGENR), Sensor (SENSR), Smart Sensor (SMSNSR), State Evaluation (STEVAL), Maintenance Scheduler (MSCHEDR), Maintenance Generator (MGENR), and Transducer (TRANSD). PWRGEN models electrical power generation based on wind speeds from the WGENR atomic model. The amount of power generated at any given time is calculated using a power curve of the form given in Equation (1) as described in Section 3.1. The WGENR atomic model generates wind speeds calculated using the spatio-temporal wind model described in Equations (5)–(9). CMPDEG models the degradation or deterioration process of wind turbine components.

SENSR and SMSNSR models unreliable sensor and smart sensor behavior, respectively, as described in

Section 3.1. The STEVAL atomic model is responsible for estimating the actual state of a wind turbine gearbox based on the sensor information from SENSR. This is done based on the hidden Markov model described in Section 3.1. MSCHEDR models the two types of maintenance strategies, SM and CBM, described in Section 3.1. The MGENR atomic model is responsible for generating the actual maintenance actions (e.g. dispatching a maintenance crew) based on the maintenance schedules created by MSCHEDR. Finally, TRANSD is in charge of computing the performance measures presented in Section 3.1 and statistical parameters of interest (e.g. number of wind turbine gearbox failures) during the simulation run.

Power Generator (PWRGEN) atomic model. The PWRGEN atomic model has several basic states; *off_normal*, *off_normal_waiting*, *on_normal*, *off_alert*, *off_alert_waiting*, *on_alert*, *off_alarm*, *off_alarm_waiting*, *on_alarm*, *failed*, and *report_status*. We consider a PWRGEN with the input and output ports as shown in Figure 4. The model has seven basic input ports, namely, ‘turb_on_off’, ‘wind_in’, ‘deg_in’, ‘corr_mnt’, ‘prev_mnt’, ‘obsv’, and ‘req_status’.

The operation of the PWRGEN atomic model is depicted in Figure 5. The model is initialized in the *off_normal* state. When an input is received on the ‘turb_on_off’ input port, the wind speed at the current location is verified. A transition to the *on_normal* state occurs if the current wind speed is within the thresholds specified for the proper operation of the turbine. If the current wind speed is outside the thresholds, the model transitions to the *off_normal_waiting* state. Messages received on the ‘wind_in’ input port will notify changes in the wind speed. A transition from the *off_normal_waiting* state to the *on_normal* state occurs when the wind speed satisfies the thresholds required for wind turbine operation. If an input is received on the ‘prev_mnt’ input port or on the ‘turb_on_off’ input port when the model is in the *on_normal* state, a transition to *off_normal* state occurs.

A transition from the *on_normal* state to the *on_alert* state occurs if a message is received on the ‘deg_in’ input port containing a component degradation information of *alert* (or type 1). When in the *on_alert* state,

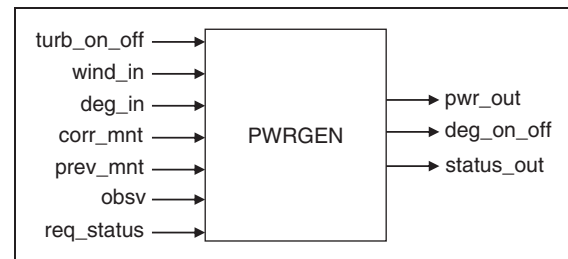


Figure 4. The PWRGEN atomic model.

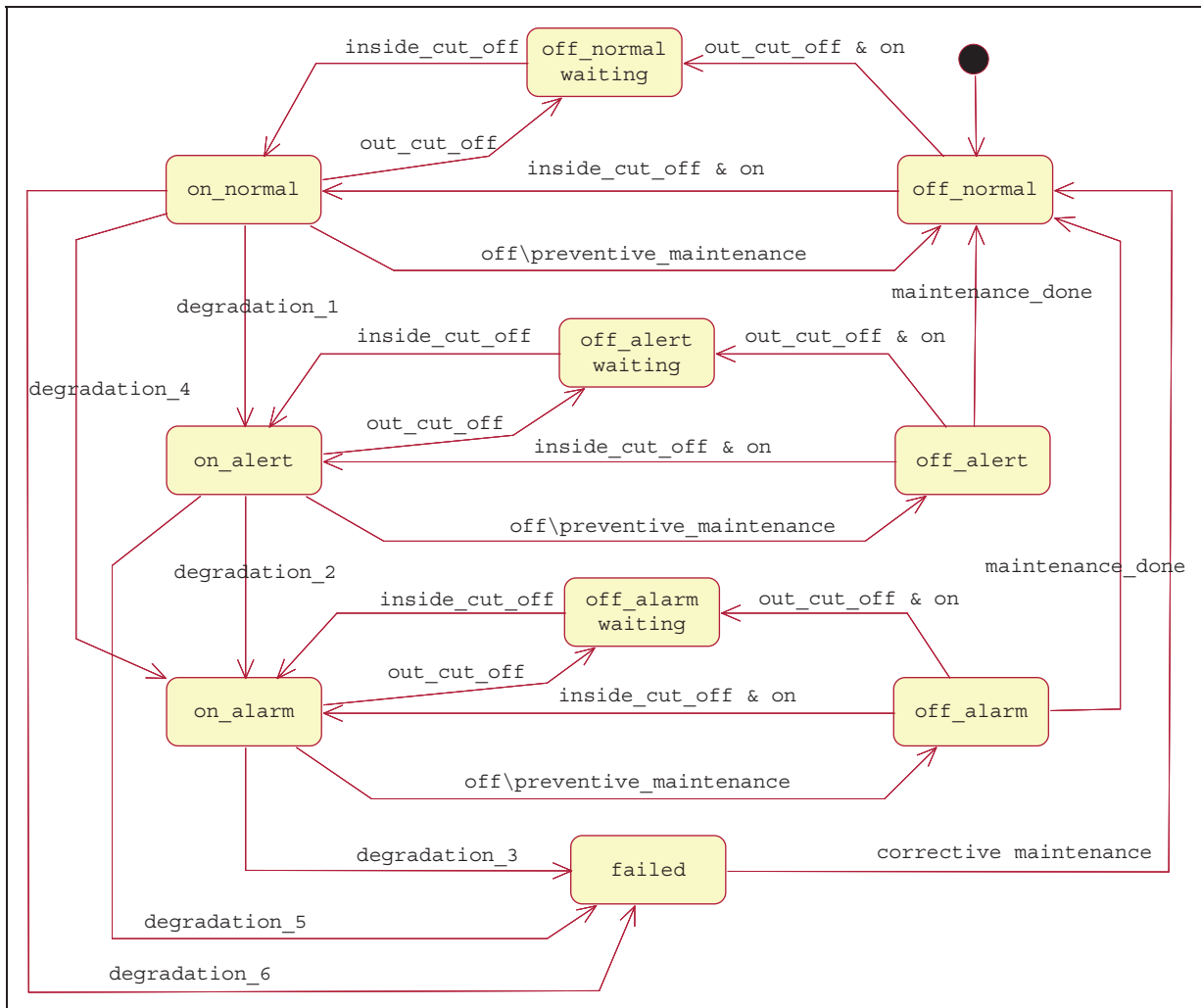


Figure 5. The PWRGEN state transition diagram.

the model transitions to the *off_alert_waiting* state if the wind speed goes out of the specified thresholds, or to the *off_alert* state if the component is turned off or if preventive maintenance has to be performed. In the case when preventive maintenance is performed, the model transitions from the *off_alert* to the *off_normal* state upon completion of maintenance.

A transition to the *on_alarm* state occurs if the model is in the *on_normal* or *on_alert* state and a message is received on the 'deg_in' input port containing a component degradation information of *alarm* (or type 2 or type 4). When in the *on_alarm* state, the model transitions to the *off_alarm_waiting* state if the wind speed goes out of the specified thresholds, or it transitions to the *off_alarm* state if the component is turned off or if preventive maintenance has to be performed. In the case when preventive maintenance is performed, the model transitions from the *off_alarm* state to the *off_normal* state upon completion of maintenance.

A transition to the *failed* state occurs if the model is in one of the following states: *on_normal*, *on_alert*, or *on_alarm*; a message is received on the 'deg_in' input port containing a component degradation information of *failed* (or type 3, type 5, or type 6).

The model transitions from the *failed* state to the *off_normal* when a message is received on the 'corr_mnt' input port indicating that the turbine has been fixed because corrective maintenance has been performed. When a message is received on the 'req_status' input port the model transitions to the *report_status* state and remains in this state for a short time interval (STI). Once this STI has elapsed, the model transitions to the state it was in before receiving the message.

The PWRGEN atomic model has three output ports, namely, 'pwr_out', 'deg_on_off', and 'status_out'. The 'pwr_out' output port is used to notify the amount of power generated by the turbine during a period of time. A message is sent using the 'deg_on_off' output port

when the turbine is turned off. This message will turn off other components that are linked to this model. The ‘status_out’ output port is used to report the status of the turbine when requested by the smart sensor. A mathematical expression of the PWRGEN atomic model in parallel DEVS is given in Appendix A.

Component Degradation (CMPDEG) atomic model. CMPDEG is a model of a critical component of a wind turbine and in our case represents the degradation of a wind turbine gearbox. This atomic model has six basic states: *passive*, *active*, *passive_wind*, *report_status*, *report_deg*, and *passive_service*. CMPDEG is always coupled to a PWRGEN atomic model to create the wind turbine (WTURBINE) coupled model. The couplings between these two models are discussed next under the WTURBINE coupled model. CMPDEG has three input ports, namely, ‘wind_on_off’,

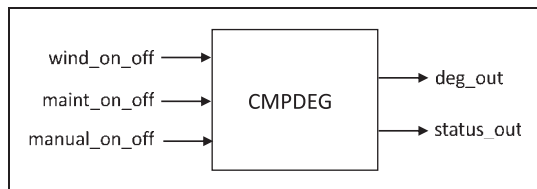


Figure 6. The CMPDEG atomic model.

‘main_on_off’, and ‘manual_on_off’. Figure 6 shows the input and output ports for the model, while its operation is depicted in Figure 7.

CMPDEG is initialized in the *passive* state. If an input is received on the ‘manual_on_off’ input port, the model transitions to the *active* state. Once in the *active* state, three things can happen. First, a message received at the ‘wind_on_off’ input port will indicate that the component has to be turned off due to the current wind speed being above the threshold. In this case the model transitions to the *passive_wind* state. Second, a transition to the *report_status* state occurs when a predetermined large time interval (LTI) has elapsed. The *report_status* state is used to report the current component status to the sensors. Third, a change in the status of the model due to degradation will take the model to the *report_deg* state. The model will transition to the *passive_service* state if it is in the *active* or *passive_wind* state and a message is received on the ‘maint_on_off’ port. When maintenance is completed the model returns to its initial *passive* state.

The CMPDEG model has two output ports, namely, ‘deg_out’ and ‘status_out’. The first output is used to send messages to the PWRGEN atomic model when a status change occurs. The second output port is used to report the current state of the component to the

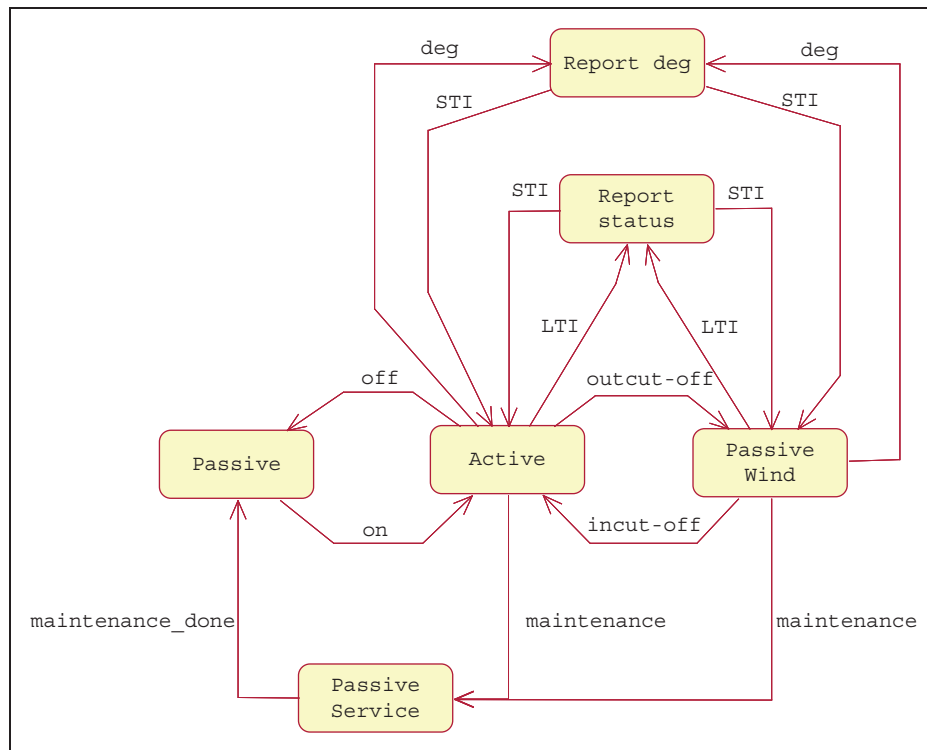


Figure 7. The CMPDEG state transition diagram.

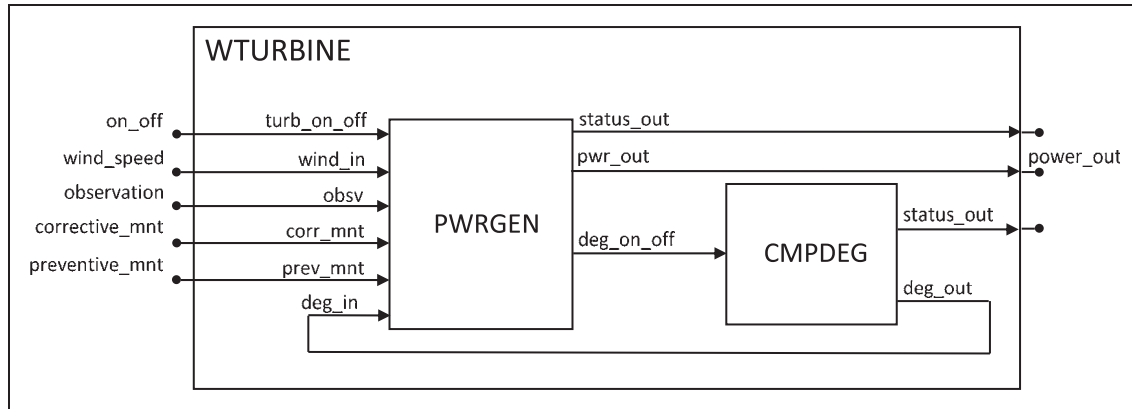


Figure 8. Wind turbine block diagram with input and output ports.

sensors. A mathematical expression of the CMPDEG atomic model in parallel DEVS is given in Appendix A.

4.3. Coupled models

We couple the four atomic models, PWRGEN, CMPDEG, SENSR, and STEVAL, to create a Wind Turbine (WTURBINE) coupled model. A collection of several WTURBINE coupled models forms the Wind Farm (WF) coupled model. We couple MSCHEDR and MGENR to create an Operation and Maintenance (OPMNT) coupled model. Similarly, we couple WGENR and TRANSD to create an Experimental Frame (EF). Finally, we couple the three coupled models, WF, OPMNT, and EF, to form the DEVS wind farm simulation model. Due to space limitations, we now provide details on two atomic models, PWRGEN and CMPDEG, to illustrate the derivation of the atomic models. We present the mathematical expressions of the two atomic models in parallel DEVS in Appendix A for the interested reader for testing and verification purposes. We omit detailed descriptions of the rest of the atomic models and instead provide descriptions of the three coupled models, WF, OPMNT, and EF, using block diagrams. We end this section with a system entity structure (SES) to provide a summary of the hierarchical structure and possible structures of the DEVS wind farm simulation model.

Wind Turbine (WTURBINE) coupled model. WTURBINE is a coupled model comprising the PWRGEN and CMPDEG atomic models. Figure 8 gives the block diagram for the WTURBINE coupled model showing the input and output ports. Five input ports are defined for this coupled model, namely, 'on_off', 'wind_speed', 'observation', 'corrective_maintenance', and 'preventive_maintenance'. The atomic models communicate using two internal couplings (ICs). Information is passed to the CMPDEG model when the PWRGEN is turned off (on). Every time the

PWRGEN atomic model is turned off, the CMPDEG atomic model is turned off as well. The CMPDEG atomic model notifies the PWRGEN atomic model about changes in the component's degradation status. Recall that if degradation occurs PWRGEN transitions from the *normal* to the *alert* state. The three output ports in Figure 8, namely, 'pwrgen_status_out', 'comp_status_out', and 'power_out', are used to report the status and performance of WTURBINE.

Operation and Maintenance (OPMNT) coupled model. OPMNT is a coupled model formed by coupling the MSCHEDR and MGENR atomic models. Figure 9 shows the block diagram for a OPMNT coupled model. One input port is defined for this coupled model, namely, 'status_in'. The MSCHEDR atomic model communicates with the MGENR atomic model by using one IC. Information is passed to the MGENR atomic model when a maintenance procedure is scheduled. MGENR uses the information to generate maintenance jobs at the scheduled times.

Experimental Frame (EF). The experimental frame is one of the most important components of the simulation model because it is used to define the experiment parameters and to collect the information of interest from the simulation runs.

In the presented wind farm model, EF is a coupled model that is formed by coupling the WGENR and TRANSD atomic models. Figure 10 shows the atomic models that are part of the EF coupled model and the way they are connected. Recall that the WGENR atomic model is in charge of generating wind speed information for each one of the turbines used in the simulation. This model allows the computation of the wind speed for each WTURBINE atomic model based on its height and location in the wind farm. The TRANSD coupled model collects the information of interest and computes the performance measures specified by the user.

Overall simulation model The overall wind farm simulation model is depicted in Figure 11. Owing to space

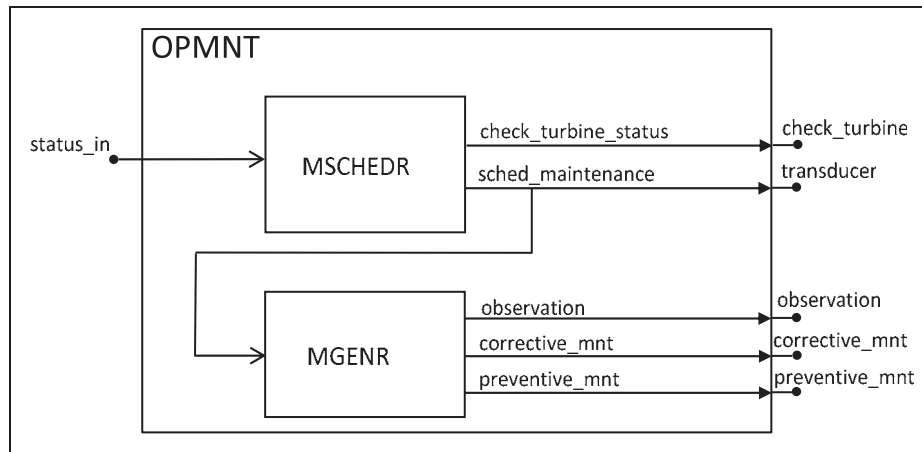


Figure 9. Operation and maintenance block diagram with input and output ports.

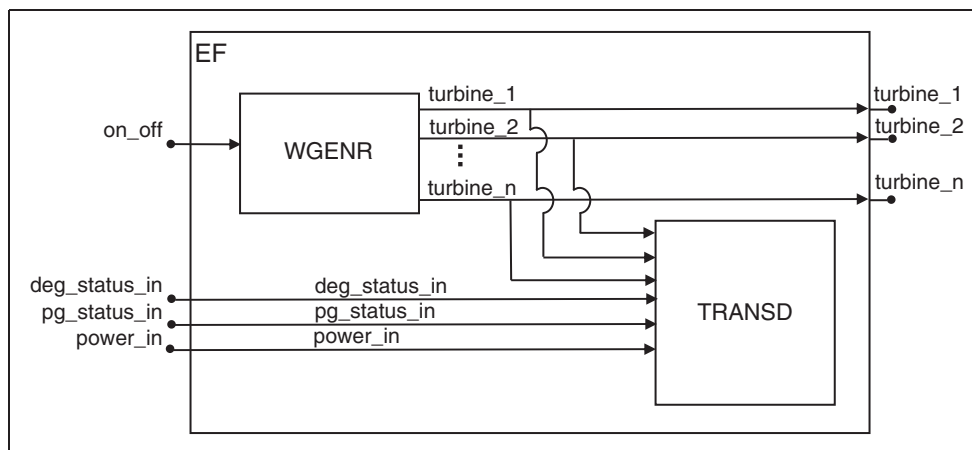


Figure 10. The EF coupled model.

restrictions, the figure shows only one wind turbine. In general, the WF would have several wind turbines coupled to the EF and OPMNT coupled model. Observe that the EF and OPMNT are designed to be separate from WF. This allows for changes in the experimenter's goals or changes in the O&M policies to be done independently without concern for making changes to the WF coupled model.

4.4. System entity structure

The SES is utilized to plan, generate, and evaluate the design of simulation-based systems.¹ This is a scheme that organizes a set of possible structures of a system. A library of models is generated when all the components abstracted from the real system are implemented. The SES is used to classify these components by their characteristics and to organize them in hierarchical composition. This representation allows the modeler

to visualize the system as a whole. The goal of the SES is to synthesize a simulation model by traversing a model hierarchical structure. A SES represents not a single model structure, but a family of structures from which a candidate entity structure can be selected.

Figure 12 shows the SES for the wind farm discrete event simulation. At the top level, the scheme shows the two major coupled models that define the system structure. The *Experimental Frame* (EF) branch can be decomposed into two branches that are assigned to the *Transducer* (TRANSD) and *Generator* (GENR) atomic models. The double line under the GENR branch means specialization. The GENR model can be categorized into a specialized entity called *Wind Generator* (WGENR). The *Wind Farm* (WF) branch can be decomposed into three branches: *System Evaluation* (EVAL), *Wind Turbine* (WTURBINE), and *Operation and Maintenance* (OPMNT). The EVAL branch can be decomposed into two

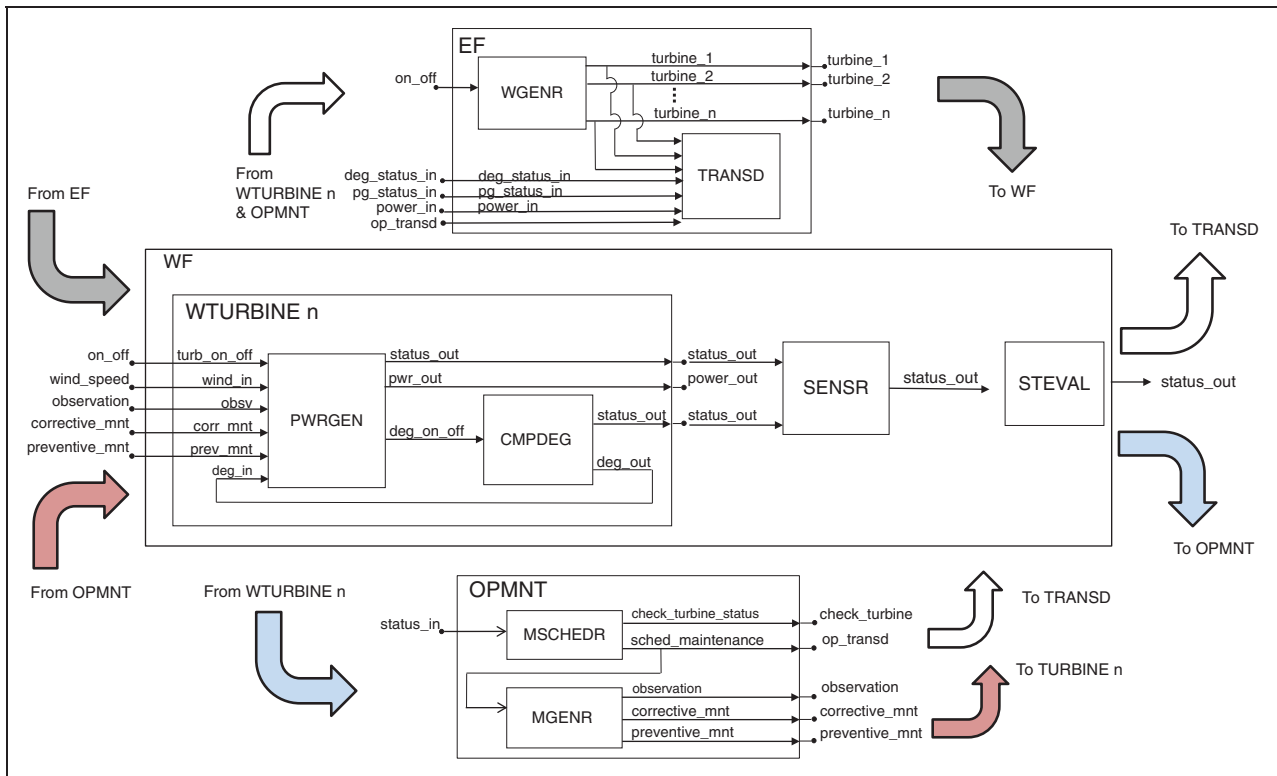


Figure 11. The DEVS wind farm system.

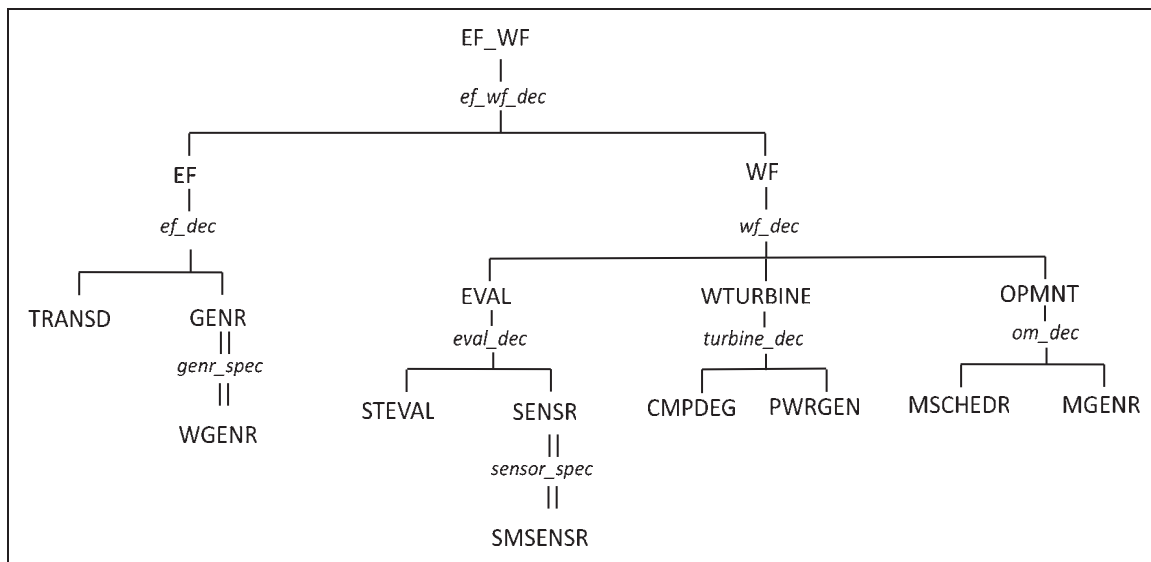


Figure 12. SES for the DEVS wind farm simulation.

branches, *State Evaluation* (STEVAL) and *Sensor* (SENSR). The SENSR model can be categorized into a specialized entity called *Smart Sensor* (SMSENSR). The WTURBINE and the OPMNT branches are decomposed into the models that are used to form their respective coupled models.

5. Application

The wind farm simulation model was implemented in DEVSJAVA³⁶ which is a Java-based software implementation of the DEVS formalism. The simulation model was verified and tested using DEVSJAVA

SimView version 1.0.4.³⁶ This visual interface allows the modeler to inspect the behavior of each atomic and coupled model created in DEVJSJAVA. Atomic models were inspected first because they need to perform as expected to achieve the proper functioning of the coupled models. SimView has several convenient functionalities such as allowing the user to control the simulation run (start and stop), simulation fast-forwarding or slow motion, and being able to insert user-defined parameters created for model verification and testing by using the models' input ports. The SimView user interface provides a top menu that allows the user to select the appropriate model and run it with the click of a button. A simulation clock is always displayed on the interface during the simulation run as well as statistics of the active models by simply positioning the mouse cursor on top of the model block. All computational experiments were conducted on a DELL Optiplex GX620 with a Pentium D processor running at 3.0 GHz with 3.5 GB of RAM.

5.1. Computational experiments design

We applied our simulation model to a 100-unit wind farm located in West Texas. We also assumed that the wind farm operates 24 hours a day for 365 days a year. Several computational experiments were performed to validate the simulation model and gain management insights into the impact of maintenance scheduling policies (SM and CBM) on system performance. Under each maintenance strategy, ten replications for the scheduling time horizon of 20 years were made, considering that the average lifespan of a wind turbine is typically 20 years. The number of replications was set such that the variance of each performance measure was sufficiently small enough to show stable performance among replications. The Monte Carlo method was used to allow for independence among the replications.

Among the several components in a wind turbine, McMillan and Ault¹⁰ show that the most critical failures are associated with the gearbox because of high capital cost, long lead time for repairs, difficulty in replacing it, and lengthy downtime compounded by adverse weather conditions. Therefore, we chose the *gearbox* among several components of a wind turbine to illustrate our simulation model. The proposed simulation model can be extended in the future to include other critical components such as drive train, generator, and the electrical system.

A preliminary validation of the simulation model was achieved by configuring the simulation based on a real setting and using the performance measures presented in Section 3.2.

Table 2. GE 1.5 sle turbine specifications

Turbine specifications	
Cut-in speed	3.5 m/s
Rated speed	14 m/s
Cut-out speed	25 m/s
Rated power	1.5 MW

Wind turbine configuration. We assumed all the wind turbines to be *General Electric (GE) sle*'s with 1.5 MW rated power with a hub height of 100 m. The parameters corresponding to the specifications of a (GE) 1.5 sle wind turbine are listed in Table 2.

Gearbox degradation. In this study, the transition matrix P , which reflects the degradation behavior of a gearbox, is devised to be consistent with the historical failure frequency of a gearbox. According to Ribrant,³⁵ the failure frequency of a gearbox ranges from 0.05 to 2.29 times per year. Since most wind farm operators perform the SM, we believe that this failure frequency is a result of the SM practice. Based on this understanding, we constructed the following transition matrix P :

$$P = \begin{bmatrix} 0.95 & 0.04 & 0.01 & 0.00 \\ 0.00 & 0.97 & 0.02 & 0.01 \\ 0.00 & 0.00 & 0.94 & 0.06 \\ 0.00 & 0.00 & 0.00 & 1.00 \end{bmatrix} \quad (11)$$

We consider one week as a transition period and therefore P represents a weekly-based deterioration process.

Weather data. The spatio-temporal model discussed in Section 3.1 was developed using historical wind data from the West Texas Mesonet.³⁷ The West Texas Mesonet is a network of meteorological monitoring instruments, dispersed across West Texas. Based on the wind characteristics at this wind farm location, a value of 0.31 for α in Equation (9) was used, which is a typical value for suburban areas.²⁹

O&M strategies. We simulated wind farm operations under the two maintenance strategies: SM and CBM. As mentioned in Section 3.1, we assumed that, under SM, maintenance actions are performed twice a year in low windy conditions regardless of the deterioration status of the turbine. Under the CBM strategy, preventive maintenance actions are carried out only when sensors in the condition monitoring equipment produce *alarm* signals. We assumed that SM as well as preventive maintenance for each turbine takes two days at a cost of €6,360 based on a study by Rademakers et al.² Under both SM and CBM, corrective maintenance is performed upon an unplanned

failure. In this case, we consider a lead time of six weeks before repairing the turbine. The cost of performing corrective maintenance on a single gearbox was assumed to be €12,720.² The number of available repair crews was fixed at five throughout the simulation.

5.2. Simulation results and discussion

The simulation results for the average total power generation and the capacity factor for 20 years under SM and CBM are reported in Table 3. We also report the simulation computational time. The columns of the table show the mean and standard deviation (StDev) for each scheduling strategy. The results show that CBM performs better than SM on both power generation and capacity factor. There is an increase of about 5.85% (703,969 MW) in power generation and an increase of about 5.92% in capacity factor under CBM. Each simulation run took about 1.2 hours due to the large number of wind turbines (100) and the lengthy planning horizon (20 years) used.

Annual power generation for each maintenance policy is reported in Figure 13. The results show that CBM outperforms SM for all the years except year 12, where SM has a slightly higher power output. Also,

observe that the amount of power generation decreases annually under SM starting in year 14. This is the time when the wind turbines are near their lifespan and are more prone to failure. The results indicate that there is relatively more corrective maintenance under SM than CBM.

We also report on the ‘accumulated’ capacity factor, that is, the average capacity factor based on the number of years from the start of the simulation. The results are plotted in Figure 14 and show steady capacity factors for both SM and CBM. However, CBM has relatively larger values than SM. This means that, on average, relatively more power is generated under CBM throughout the wind farm operational years. Recall that the capacity factor in actual wind farms is between 0.25 and 0.4, which is typically reported for SM since it is the most widely used maintenance policy in the wind power industry. In our case we obtain slightly higher values for capacity factor. We believe this is due to the fact that we only consider gearbox failures. If we incorporate failures of other wind turbine components such as blades and generator, the capacity factor under SM would fall within the range reported in the literature. Furthermore, the capacity factor under CBM is expected to be higher since CBM outperforms SM.

Table 3. Simulation results for power generation and capacity factor

	SM		CBM	
	Mean	StDev	Mean	StDev
Generated power (MW)	12,025,091.98	18,851.97	12,729,060.05	10,488.85
Capacity factor	0.422	0.001	0.447	0.001
CPU time (s)	5102.67	84.23	5023.02	7.79

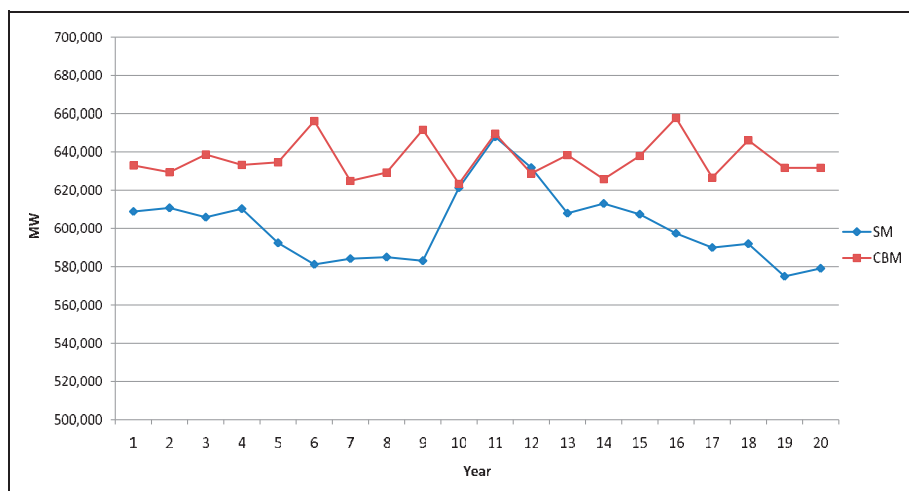


Figure 13. Annual average power generation.

The rest of the wind farm simulation performance measures are reported in Table 4. The table shows average wind turbine availability, average number of failures per turbine, and average maintenance cost per turbine over the 20-year period. The results show that CBM has about 1.0% higher wind turbine availability over SM. CBM also has about 11.7% less gearbox failures than SM. In terms of average total maintenance cost, SM has a significantly higher cost (about 31.5% higher) than CBM. Recall that under SM preventive maintenance is always performed twice a year on each turbine even when it may not be necessary, thus contributing to the higher maintenance cost. Under CBM preventive maintenance is only performed when necessary based on sensor information. Even though CBM has much lower maintenance costs, one has to also factor in the cost of managing and maintaining the sensors. Since we were unable to find cost figures related to sensor management in the literature, we did not factor in such a cost into the total average maintenance cost. We should point out that availability for onshore wind turbines reported in the literature is about 0.98, which is slightly higher than what we obtained (about 0.90). The low availability we obtained is due to the fact that we used a fixed and

relatively long lead time of six weeks for corrective maintenance of unplanned gearbox failures. According to a study by Ribrant,³⁵ the downtime upon a gearbox failure widely varies from 1 hour to 2,067 hours (almost 12 weeks). So in our simulation we used the mid-range value of six weeks for lead time. However, in practice this time can significantly vary. We should also mention that the average number of failures per turbine in a year under the SM policy, 0.932, falls within the range of actual gearbox failures, which is 0.05–2.29 times per year.

Figure 15 compares the average number of failures per year during the 20-year period. The results show that the average number of failures is always higher for SM. Also, notice that there is a cyclic failure pattern similar to a sinusoid curve in both maintenance policies, which indicate the general Markovian degradation pattern followed by gearbox failures.

Figure 16 depicts the ‘accumulated’ average number of failures at a given time period, that is, the average number of failures from year one up to a given year. The results show that SM has higher average number of failures at each time period. It is interesting to notice that there is an increasing number of failures in the last five years under the SM policy. This can be attributed

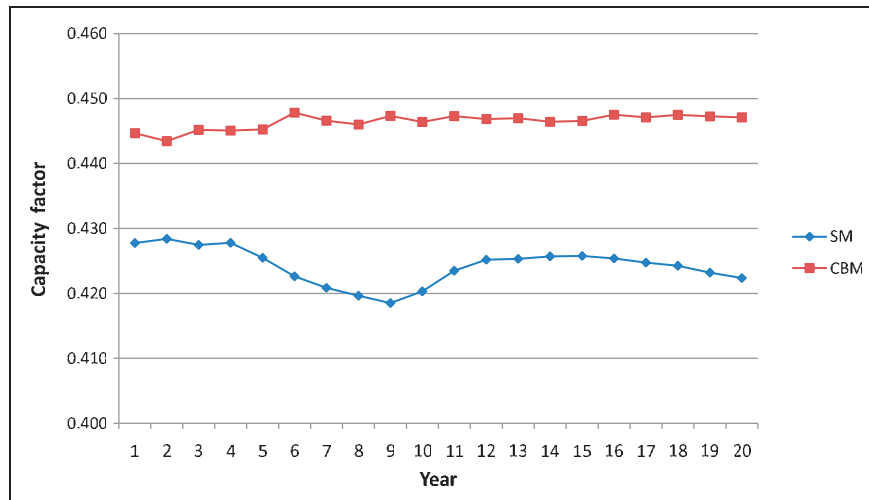


Figure 14. Accumulated average capacity factor.

Table 4. System performance measurements over a 20-year period

Performance measure	SM		CBM	
	Mean	StDev	Mean	StDev
Availability	0.898	0.008	0.907	0.006
Number of failures per wind turbine per year	0.932	0.014	0.823	0.011
Maintenance cost per wind turbine (€) per year	24,583.31	172.00	16,828.56	137.55

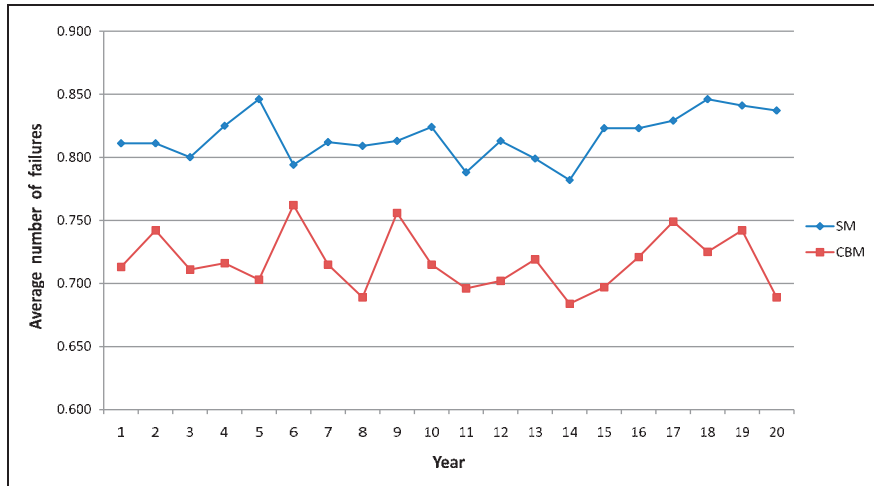


Figure 15. Average number of failures per wind turbine.

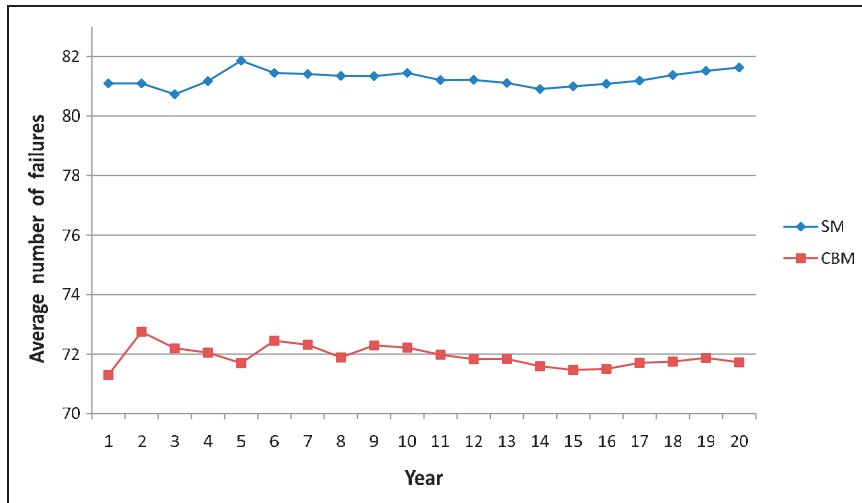


Figure 16. Accumulated average number of failures per turbine.

to the fact that we observed more failures than the available maintenance crews could handle towards the end of the wind turbine lifespan. Recall that the number of repair crews was fixed at five throughout the simulation. Therefore, wind turbines that were close to failure during those years had to wait for repair crews to finish preventive maintenance already scheduled for other turbines. The limited maintenance resources thus resulted in long downtime leading to increased revenue losses. In contrast, the accumulated average number of failures under CBM remained steady because the CBM strategy utilized maintenance resources only when repairs were needed to avoid failures. This result indicates that CBM is a beneficial maintenance strategy in wind farms with limited repair resources.

To assess the performance of each individual wind turbine, we plotted the average availability (left axis) and the number of failures (right axis) for each turbine for the 20-year period for SM and CBM in Figures 17 and 18, respectively. The average availability and number of failures for SM are between 0.87 and 0.92 and 15 and 22, respectively. For CBM, the average availability and number of failures are between 0.89 and 0.93 and 13 and 20, respectively. Thus CBM has relatively better values for both availability and number of failures for each turbine. Finally, Figure 19 plots the availability for each turbine for both SM and CBM. As can be seen in the graph, CBM gives higher availability for most of the wind turbines. Also, CBM has lesser variability in average availability among the wind turbines.

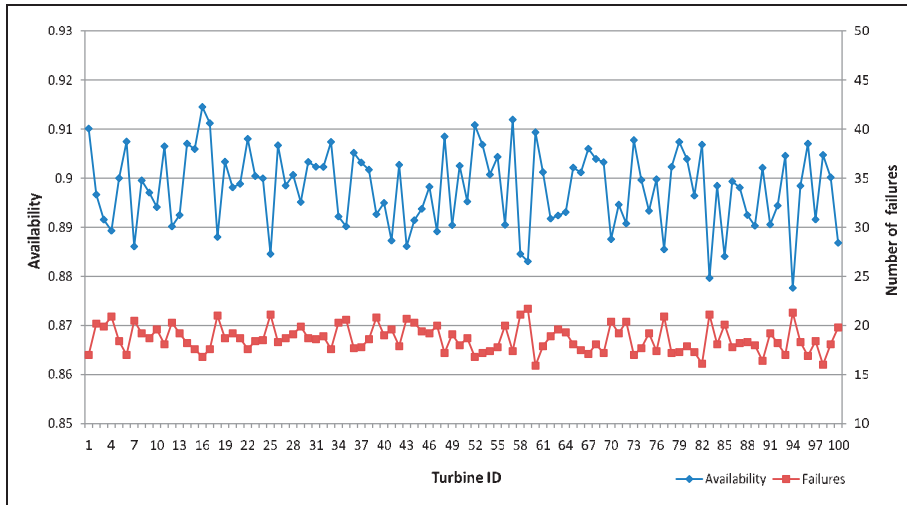


Figure 17. Average availability and number of failures per turbine under SM.

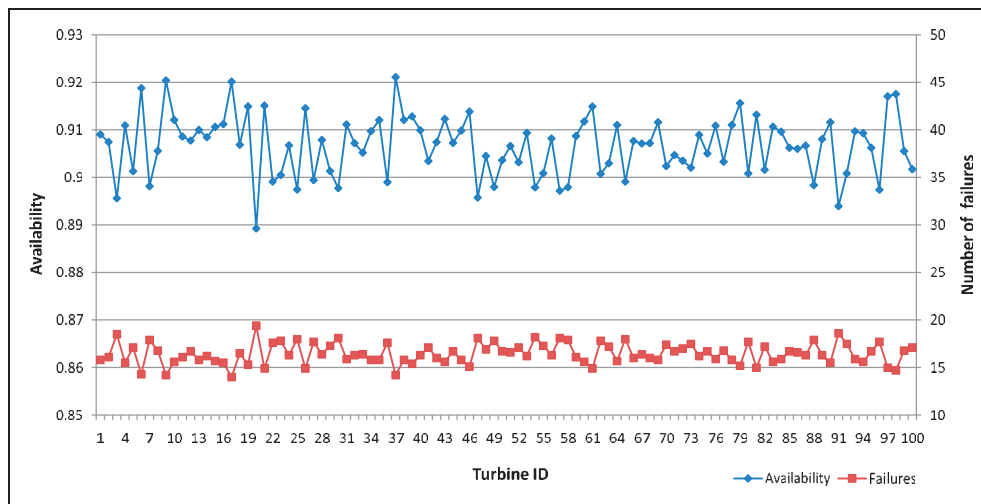


Figure 18. Average availability and number of failures per turbine under CBM.

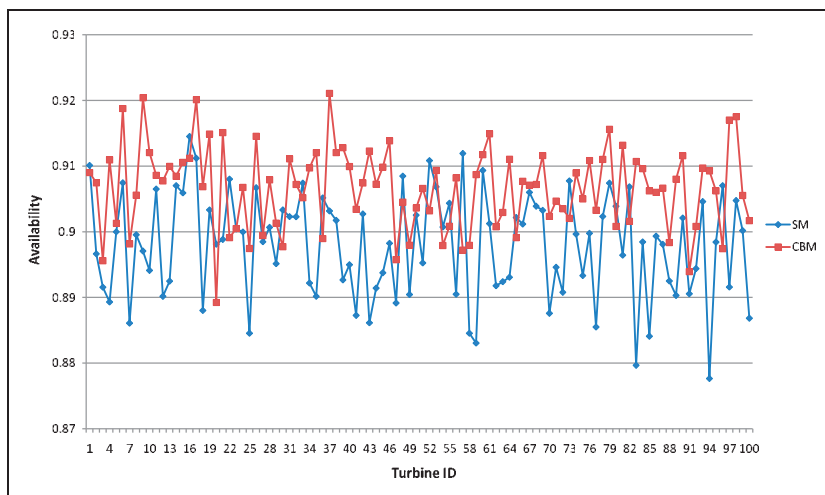


Figure 19. Average availability per turbine under both scheduling algorithms.

6. Conclusion

We have used the discrete event M&S approach to build a generic simulation model for wind farm operations. The study involves the following subtasks: (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 experiment choices; (d) computer implementation of the models; (e) testing, verification, and validation of the simulation models. The simulation platform represents actual wind farm operations with sufficient details, enabling insight to be gained into the operation of large-scale wind power systems. It provides a tool for wind farm operators to select the most cost-effective O&M strategy. In particular, two maintenance policies are studied, scheduled maintenance (SM) and condition based maintenance (CBM). SM performs preventive repairs regularly and represents the standard practice in the industry, whereas CBM performs preventive repairs based on several sensor output alarm signals. The implementation results demonstrate that CBM provides better benefits than the SM under all the performance measures considered such as capacity factor, availability, number of component failures, and maintenance cost. For example, the failure frequency and the overall O&M costs are considerably reduced by 11.7% and 31.5%, respectively, when CBM is used instead of SM.

This work opens up several future research directions. For example, the current simulation model can be extended to incorporate degradation of multiple wind turbine components such as gearbox, generator, and blades. The current study only considers gearbox maintenance. It would be interesting to study how robust the recommended CBM policy would perform when multiple components are considered. Another direction for future research is to incorporate an optimization model into the simulation model for computing maintenance actions at a given time period.^{17,38} Such an optimization model would consider several critical factors, such as a component's degradation status, weather, and revenue losses, in making operations and maintenance decisions.

We also believe that the current simulation model can provide a platform for a broad array of potential applications related to wind power systems. For example, the simulation model can be extended to allow for the evaluation of wind farm sites,³⁹ correlation studies for multiple wind farms,³ wind power system reliability analysis,^{5,6} and evaluation of generation adequacy of power systems^{4,40} with minimal changes to the platform. Also, the model can be extended to enable assessment of generation capacity of hybrid models with different power generators such as conventional

fuel-fired power, battery and wind energy.⁴¹ Finally, the authors will consider the use of stochastic DEVS⁴² to add stochastic state transitions to the critical components such as CMPDEG.

Eunshin Byon received her BS and MS (Honors) in Industrial and Systems Engineering from Korea Advanced Institute of Science and Technology (KAIST), South Korea, and her PhD in Industrial Engineering from Texas A&M University, USA, in 2010. She is currently a postdoctoral research associate in the Department of Industrial and Systems Engineering at Texas A&M University. Her research interests include operations and management of wind power systems, statistical modeling and analysis for complex systems, discrete event simulation, and simulation-based optimization. She is a member of IIE, INFORMS, and IEEE.

Eduardo Pérez received his BS degree in Industrial Engineering at the University of Puerto Rico, Mayaguez, Puerto Rico in 2004, and his PhD degree in Industrial Engineering at Texas A&M University, College Station, Texas in 2010. He is currently a postdoctoral research associate in the Department of Industrial and Systems Engineering at Texas A&M University. His research interests include systems modeling and systems engineering processes, discrete event simulation, stochastic programming and scheduling. He is a member of INFORMS, IIE, and SCS.

Yu Ding received his BS degree in Precision Engineering from University of Science and Technology of China in 1993, a MS degree Precision Instruments from Tsinghua University, China in 1996, a MS degree in Mechanical Engineering from the Pennsylvania State University in 1998, and his PhD in Mechanical Engineering from the University of Michigan in 2001. He is currently an associate professor and holder of the Centerpoint Energy Career Development Professorship in the Department of Industrial and Systems Engineering at Texas A&M University. His research interests are in the area of quality and reliability engineering and systems informatics. He currently serves as a department editor of IIE Transactions and is a member of IEEE, INFORMS, IIE, and ASME.

Lewis Ntaimo is an associate professor at Texas A&M University, Department of Industrial and Systems Engineering, College Station, Texas, USA. He obtained his BS degree in Mining Engineering in 1998, his MS in Mining and Geological Engineering in 2000, and his PhD in Systems and Industrial Engineering in 2004, all from the University of Arizona. He has been with Texas A&M University

since 2004. His research interests are in the area of systems modeling and simulation, and stochastic programming. He currently serves as vice chair for the INFORMS Optimization Society, research advisor for operations research for the Arizona Center for Integrative Modeling and Simulation (ACIMS). He is a member of INFORMS, IIE, and SCS.

Acknowledgments

The authors would like to thank Dr Carsten Hein Westergaard of Vestas, USA, for insightful conversations on wind farms that partly motivated this work.

References

- Zeigler BP, Kim TG and Praehofer H. *Theory of modeling and simulation*. Orlando, FL: Academic Press, Inc., 2000.
- Rademakers LWMM, Braam H and Verbruggen TW. R&D needs for O&M of wind turbines. *ECN Wind Energy* 2003; ECN-RX-03-045.
- Wan Y, Milligan M and Parsons B. Output power correlation between adjacent wind power plants. *J Solar Energy Eng* 2003; 125(4): 551–555.
- Karki R and Billinton R. Cost-effective wind energy utilization for reliable power supply. *IEEE Trans Energy Conversion* 2004; 19(2): 435–440.
- Karki R and Patel J. Reliability assessment of a wind power delivery system. *Proc Instit Mech Eng, Part O: J Risk Reliabil* 2009; 223(1): 51–58.
- Wen J, Zheng Y and Donghan F. A review on reliability assessment for wind power. *Renew Sustain Energy Rev* 2009; 13(9): 2485–2494.
- Rademakers LWMM, Braam H, Zaaier MB and van Bussel GJW. *Assessment and optimisation of operation and maintenance of offshore wind turbines* 2003; Technical Report, 5 pages. Available at <http://www.ecn.nl/library/reports/main.html>.
- Van Bussel GJW. The development of an expert system for the determination of availability and O&M costs for offshore wind farms. In *1999 European Wind Energy Conference and Exhibition, Conference Proceedings*. Nice, France, 1999, p.402–405.
- Hendriks HB, Bulder BH, Heijdra JJ, et al. DOWEC concept study; evaluation of wind turbine concepts for large scale offshore application. Siracusa, Italy: Offshore Wind Energy in Mediterranean and other European Seas (OWEMES conference), 2000.
- McMillan D and Ault GW. Condition monitoring benefit for onshore wind turbines: sensitivity to operational parameters. *IET Renew Power Gener* 2008; 2(1): 60–72.
- Andrawus JA, Watson J and Kishk M. Modeling system failures to optimise wind turbine maintenance. *Wind Eng* 2007; 31(6): 503–522.
- ReliaSoft BlocSim-7 software <http://www.reliasoft.com/products.htm/> (2007)
- Hall PL and Strutt JE. Probabilistic physics-of-failure models for component reliabilities using Monte Carlo simulation and Weibull analysis: a parametric study. *Reliab Eng Syst Safety* 2007; 80: 233–242.
- Ravindra MM and Prakash SK. Generator system reliability analysis including wind generators using hourly mean wind speed. *Electr Power Compon Syst* 2008; 36: 1–16.
- Di Fazio AR and Russo M. Wind farm modelling for reliability assessment. *IET Renew Power Gener* 2008; 2: 239–248.
- Leite AP, Borges CLT and Falcao DM. Probabilistic wind farms generation model for reliability studies applied to brazilian sites. *IEEE Trans Power Syst* 2006; 21: 1493–1501.
- Byon E, Ntaimo L and Ding Y. Optimal maintenance strategies for wind power systems under stochastic weather conditions. *IEEE Trans Reliab* 2010; 59(2): 393–404.
- Nilsson J and Bertling L. Maintenance management of wind power systems using condition monitoring systems-life cycle cost analysis for two case studies. *IEEE Trans Energy Conversion* 2007; 22(1): 223–229.
- Law AM and Kelton WD. *Simulation modeling and analysis*. London: McGraw-Hill Higher Education, 1997.
- Gneiting T, Larson K, Westrick K, Genton MG and Aldrich E. Calibrated probabilistic forecasting at the stateline wind energy center: the regime-switching space-time method. *J Amer Stat Assoc* 2007; 101: 968–979.
- de Luna X and Genton MG. Predictive spatio-temporal models for spatially sparse environmental data. *Statist Sinica* 2005; 15: 547–568.
- Soares LJ and Medeiros MC. Modeling and forecasting short-term electricity load: A comparison of methods with an application with Brazilian data. *Int J Forecast* 2008; 24: 630–644.
- Magnano L and Boland JW. Generation of synthetic sequences of electricity demand: Application in South Australia. *Energy* 2007; 32: 2230–2243.
- Im H, Rathouz PJ and Frederick JE. Space-time modeling of 20 years of daily air temperature in the Chicago Metropolitan region. *Environmetrics* 2008; 20(5): 494–511.
- Bowerman BL, O'Connell RT and Koehler AB. *Forecasting, time series, and regression*. Belmont, CA: Duxbury, 2005.
- Haining R. *Spatial data analysis in the social and environmental sciences*. Cambridge: Cambridge University Press, 2000.
- Zhou W, Yang H and Fang Z. Wind power potential and characteristic analysis of the Pearl River Delta region, China. *Renew Sustain Energy Rev* 2006; 31(6): 739–753.
- Negra NB, Holmstrom O, Bak-Jensen B and Sørensen P. Wind farm generation assessment for reliability analysis of power systems. *Wind Eng* 2007; 31(6): 383–400.
- Gipe P. *Wind power, revised edition: renewable energy for home, farm, and business*. Vermont: Chelsea Green Publishing Company, 2000.
- Pacot C, Hasting D and Baker N. Wind farm operation and maintenance management. In *Proceedings of the PowerGen Conference Asia*, 25–27.
- American Wind Energy Association <http://awea.org/> (2008)
- Qian S, Jiao W, Hu H and Yan G. Transformer power fault diagnosis system design based on the HMM

- method. In *IEEE International Conference on Automation and Logistics 2007*; 1077–1082, IEEE.
33. Khan MM, Iqbal MT and Khan F. Reliability and condition monitoring of a wind turbine. In *Canadian Conference on Electrical and Computer Engineering*. 2005, p.1978–1981, IEEE.
 34. Ding Y, Byon E, Park C, Tang J, Lu Y and Wang X. Dynamic data-driven fault diagnosis of wind turbine systems. In *International Conference on Computational Science*. Beijing, China, 2007, p.1197–1204.
 35. Ribrant J. Reliability performance and maintenance – a survey of failures in wind power systems. *Master's thesis*. 2006; 81 pages.
 36. Zeigler BP and Sarjoughian H. *Introduction to DEVS modeling & simulation with JAVA™: Developing component-based simulation models*. Arizona: Tucson/Tempe, 2003.
 37. West Texas Mesonet, [Http://www.mesonet.ttu.edu/](http://www.mesonet.ttu.edu/), (2008).
 38. Byon E and Ding Y. Season-dependent condition-based maintenance for a wind turbine using a partially observed Markov decision process. *IEEE Trans Power Syst* 2010 In press.
 39. Acker TL, Williams SK, Duque EPN, Brummels G and Buechler J. Wind resource assessment in the state of Arizona: Inventory, capacity factor, and cost. *Renewable Energy* 2007; 32(9): 1453–1466.
 40. Kaviani AK, Baghaee HR and Riahy GH. Optimal sizing of a stand-alone wind/photovoltaic generation unit using particle swarm optimization. *Simulation* 2009; 85(2): 89–99.
 41. Nelson DB, Nehrir MH and Wang C. Unit sizing and cost analysis of stand-alone hybrid wind/PV/fuel cell power generation systems. *Renewable Energy* 2006; 31(10): 1641–1656.
 42. Castro R, Kofman E and Wainer G. A formal framework for stochastic discrete event system specification modeling and simulation. *Simulation*; OnlineFirst, published on June 29, 2009 as doi:10.1177/0037549709104482.

Appendix A

In this section a mathematical definition of the PWRGEN atomic model is provided. A ‘*cut_off*’ boolean variable is used to notify when the wind speed is within a specified threshold (true) or not (false). Another boolean variable called ‘*degradation*’ is used to notify when degradation has occurred (true) in the component or not (false). *STI* is used to denote a short time interval. An entity called *msg* is used to carry out the output information of the model.

$$DEVSPWRGEN = (X_M, Y_M, S, \delta_{ext}, \delta_{int}, \delta_{con}, \lambda, ta) \quad (12)$$

where

$IPorts = \{\text{“turb_on_off”}, \text{“wind_in”}, \text{“deg_in”}, \text{“corr_mnt”}, \text{“prev_mnt”}, \text{“obsv”}, \text{“req_status”}\}$, where $X_{turb_on_off} = V_1$, $X_{wind_in} = V_2$, $X_{deg_in} = V_3$, $X_{corr_mnt} = V_4$, $X_{prev_mnt} = V_5$, $X_{obsv} = V_6$, $X_{req_status} = V_7$ are arbitrary sets;
 $X_M = \{(p, v) \mid p \in IPorts, v \in X_p\}$ is the set of input ports and values;
 $OPorts = \{\text{“pwr_out”}, \text{“deg_on_off”}, \text{“status_out”}\}$, where Y_{pwr_out} , $Y_{deg_on_off}$, and Y_{status_out} are arbitrary sets;
 $Y_M = \{(p, v) \mid p \in OPorts, v \in Y_p\}$ is the set of output ports and values; and
 $S = \{\text{“off_normal”}, \text{“off_normal_waiting”}, \text{“on_normal”}, \text{“off_alert”}, \text{“off_alert_waiting”}, \text{“on_alert”}, \text{“off_alarm”}, \text{“off_alarm_waiting”}, \text{“on_alarm”}, \text{“failed”}, \text{“report_status_n”}, \text{“report_status_t”}, \text{“report_status_m”}\} \mathfrak{R}_0 \times V_1 \times V_2 \times V_3 \times V_4 \times V_5 \times V_6 \times V_7$ is the set of sequential states.

External transition function:

$$\delta_{ext}((phase, \sigma), e, (p, v)) = (\text{“off_normal”}, \infty), \quad \text{if } \begin{cases} p = \text{“turb_on_off”} \\ phase = \text{“off_alert”} \wedge p = \text{“prev_mnt”}; \\ phase = \text{“off_alarm”} \wedge p = \text{“prev_mnt”}; \\ phase = \text{“failed”} \wedge p = \text{“corr_mnt”}; \\ phase = \text{“on_normal”} \wedge p = \text{“turb_on_off”}; \\ phase = \text{“on_normal”} \wedge p = \text{“prev_mnt”}. \end{cases}$$

$$= (\text{“off_normal_waiting”}, \infty), \quad \text{if } \begin{cases} phase = \text{“off_normal”} \wedge p = \text{“turb_on_off”} \\ \wedge cut_off = false; \\ phase = \text{“on_normal”} \wedge p = \text{“wind_in”} \\ \wedge cut_off = false. \end{cases}$$

$$\begin{aligned}
&= (\text{"on_normal"}, \infty), \text{ if } \left\{ \begin{array}{l} \text{phase} = \text{"off_normal"} \wedge p = \text{"turb_on_off"} \\ \wedge \text{cut_of } f = \text{true}; \\ \text{phase} = \text{"off_normal_waiting"} \\ \wedge p = \text{"wind_in"} \wedge \text{cut_of } f = \text{true}. \end{array} \right. \\
&= (\text{"off_alert"}, \infty), \text{ if } \left\{ \begin{array}{l} \text{phase} = \text{"on_alert"} \wedge p = \text{"turb_on_off"}; \\ \text{phase} = \text{"on_alert"} \wedge p = \text{"prev_mnt"} \end{array} \right. \\
&= (\text{"off_alert_waiting"}, \infty), \text{ if } \left\{ \begin{array}{l} \text{phase} = \text{"off_alert"} \wedge p = \text{"turb_on_off"} \\ \wedge \text{cut_of } f = \text{false}; \\ \text{phase} = \text{"on_alert"} \wedge p = \text{"wind_in"} \\ \wedge \text{cut_of } f = \text{false}. \end{array} \right. \\
&= (\text{"on_alert"}, \infty), \text{ if } \left\{ \begin{array}{l} \text{phase} = \text{"off_alert"} \wedge p = \text{"turb_on_off"} \\ \wedge \text{cut_of } f = \text{true}; \\ \text{phase} = \text{"off_alert_waiting"} \wedge p = \text{"wind_in"} \\ \wedge \text{cut_of } f = \text{true}; \\ \text{phase} = \text{"on_normal"} \wedge p = \text{"deg_in"} \\ \wedge \text{degradation} = \text{true}. \end{array} \right. \\
&= (\text{"off_alarm"}, \infty), \text{ if } \left\{ \begin{array}{l} \text{phase} = \text{"on_alarm"} \wedge p = \text{"turb_on_off"}; \\ \text{phase} = \text{"on_alarm"} \wedge p = \text{"prev_mnt"} \end{array} \right. \\
&= (\text{"off_alarm_waiting"}, \infty), \text{ if } \left\{ \begin{array}{l} \text{phase} = \text{"off_alarm"} \wedge p = \text{"turb_on_off"} \\ \wedge \text{cut_of } f = \text{false}; \\ \text{phase} = \text{"on_alarm"} \wedge p = \text{"wind_in"} \\ \wedge \text{cut_of } f = \text{false}. \end{array} \right. \\
&= (\text{"on_alarm"}, \infty), \text{ if } \left\{ \begin{array}{l} \text{phase} = \text{"off_alarm"} \wedge p = \text{"turb_on_off"} \\ \wedge \text{cut_of } f = \text{true}; \\ \text{phase} = \text{"off_alarm_waiting"} \wedge p = \text{"wind_in"} \\ \wedge \text{cut_of } f = \text{true}; \\ \text{phase} = \text{"on_alert"} \wedge p = \text{"deg_in"} \\ \wedge \text{degradation} = \text{true}; \\ \text{phase} = \text{"on_normal"} \wedge p = \text{"deg_in"} \\ \wedge \text{degradation} = \text{true}. \end{array} \right. \\
&= (\text{"failed"}, \infty), \text{ if } \left\{ \begin{array}{l} \text{phase} = \text{"on_normal"} \wedge p = \text{"deg_in"} \\ \wedge \text{degradation} = \text{true}; \\ \text{phase} = \text{"on_alert"} \wedge p = \text{"deg_in"} \\ \wedge \text{degradation} = \text{true}; \\ \text{phase} = \text{"on_alarm"} \wedge p = \text{"deg_in"} \\ \wedge \text{degradation} = \text{true}. \end{array} \right. \\
&= (\text{"report_status_n"}, STI), \text{ if } \{ \text{phase} = \text{"on_normal"} \wedge p = \text{"req_status"}; \\
&= (\text{"report_status_t"}, STI), \text{ if } \{ \text{phase} = \text{"on_alert"} \wedge p = \text{"req_status"}; \\
&= (\text{"report_status_m"}, STI), \text{ if } \{ \text{phase} = \text{"on_alarm"} \wedge p = \text{"req_status"}; \\
&= (\text{phase}, \sigma - e), \text{ otherwise}
\end{aligned}$$

Internal transition function:

$$\delta_{\text{int}}(\text{phase}, \sigma)$$

$$\begin{aligned}
&= (\text{"on_normal"}, STI), \text{ if } \text{phase} = \text{"report_status_n"} \\
&= (\text{"on_alert"}, STI), \text{ if } \text{phase} = \text{"report_status_t"}
\end{aligned}$$

$$= (\text{"on_alarm"}, STI), \text{ if } phase = \text{"report_status_m"}$$

Confluence function:

$$\delta_{con}(s, ta(s), x) = \delta_{ext}(\delta_{int}(s), 0, x).$$

Output function:

$$\begin{aligned} \lambda(phase, \sigma) &= (\text{"pwr_out"}, msg) \text{ if } \begin{cases} phase = \text{"off_normal"}; \\ phase = \text{"off_alert"}; \\ phase = \text{"off_alarm"}; \\ phase = \text{"failed"} \end{cases} \\ &= (\text{"deg_on_off"}, msg) \text{ if } \begin{cases} phase = \text{"off_normal"}; \\ phase = \text{"off_normal_waiting"}; \\ phase = \text{"off_alert"}; \\ phase = \text{"off_alert_waiting"}; \\ phase = \text{"off_alarm"}; \\ phase = \text{"off_alarm_waiting"}; \\ phase = \text{"failed"} \end{cases} \\ &= (\text{"status_out"}, msg) \text{ if } \begin{cases} phase = \text{"report_status_n"}; \\ phase = \text{"report_status_t"}; \\ phase = \text{"report_status_m"} \end{cases} \end{aligned}$$

Time advance function:

$$ta(phase, \sigma) = \sigma$$

Similarly the CMPDEG atomic model is defined in parallel DEVS. The boolean variable *deg* is used to denote a change in degradation (true) or no change (false). Another boolean is defined named *LTI*, which assumes the value of true when a large time interval is elapsed.

$$DEVS_{CMPDEG} = (X_M, Y_M, S, \delta_{ext}, \delta_{int}, \delta_{con}, \lambda, ta) \quad (13)$$

where

$$\begin{aligned} IPorts &= \{\text{"wind_on_off"}, \text{"maint_on_off"}, \text{"manual_on_off"}\}, \text{ where } X_{wind_on_of\ f} = V_1, X_{maint_on_of\ f} = V_2, \\ &\text{ and } X_{manual_on_of\ f} = V_3 \text{ are arbitrary sets;} \\ X_M &= \{(p, v) \mid p \in IPorts, v \in X_p\} \text{ is the set of input ports and values;} \\ OPorts &= \{\text{"deg_out"}, \text{"status_out"}\}, \text{ where } Y_{deg_out}, \text{ and } Y_{status_out} \text{ are arbitrary sets;} \\ Y_M &= \{(p, v) \mid p \in OPorts, v \in Y_p\} \text{ is the set of output ports and values; and} \\ S &= \{\text{"passive"}, \text{"active"}, \text{"passive_wind"}, \text{"report_status"}, \text{"report_deg"}, \text{"passive_service"}\} \times \\ &\mathfrak{N}_0^+ \times V_1 \times V_2 \times V_3 \text{ is the set of sequential states.} \end{aligned}$$

External transition function:

$$\begin{aligned} \delta_{ext}((phase, \sigma, status, deg), e, (p, v)) &= (\text{"passive"}, \infty, status, deg), \text{ if } \begin{cases} p = \text{"manual_on_off"}; \\ phase = \text{"active"} \wedge p = \text{"manual_on_off"}; \\ phase = \text{"passive_service"} \wedge p = \text{"maint_on_off"} \end{cases} \\ &= (\text{"active"}, \infty, status, deg), \text{ if } \begin{cases} phase = \text{"passive"} \wedge p = \text{"manual_on_off"}; \\ phase = \text{"passive_wind"} \wedge p = \text{"wind_on_off"} \end{cases} \end{aligned}$$

$$\begin{aligned}
&= (“passive_wind”, \infty, \text{status}, \text{deg}), \text{ if } \text{phase} = “active” \wedge p = “wind_on_off” \\
&= (“passive_service”, \infty, \text{status}, \text{deg}), \text{ if } \begin{cases} \text{phase} = “active” \wedge p = “maint_on_off”; \\ \text{phase} = “passive_wind” \wedge p = “maint_on_off”. \end{cases} \\
&= (\text{phase}, \sigma - e), \text{ otherwise.}
\end{aligned}$$

Internal transition function:

$$\begin{aligned}
&\delta_{\text{int}}(\text{phase}, \sigma, \text{status}, \text{deg}), e, (p, v) \\
&= (“report_status”, STI, \text{status}, \text{deg}), \text{ if } \begin{cases} \text{phase} = “active” \wedge LTI = \text{true}; \\ \text{phase} = “passive_wind” \wedge LTI = \text{true}. \end{cases} \\
&= (“report_deg”, STI, \text{status}, \text{deg}), \text{ if } \begin{cases} \text{phase} = “active” \wedge \text{deg} = \text{true}; \\ \text{phase} = “passive_wind” \wedge \text{deg} = \text{true}. \end{cases} \\
&= (“active”, \infty, \text{status}, \text{deg}), \text{ if } \begin{cases} \text{phase} = “report_status” \wedge \text{status} = \text{true}; \\ \text{phase} = “report_deg” \wedge \text{status} = \text{true}. \end{cases} \\
&= (“passive_wind”, \infty, \text{status}, \text{deg}), \text{ if } \begin{cases} \text{phase} = “report_status” \wedge \text{status} = \text{false}; \\ \text{phase} = “report_deg” \wedge \text{status} = \text{false}. \end{cases}
\end{aligned}$$

Confluence function:

$$\delta_{\text{con}}(s, \text{ta}(s), x) = \delta_{\text{ext}}(\delta_{\text{int}}(s), 0, x).$$

Output function:

$$\begin{aligned}
&\lambda(\text{phase}, \sigma, \text{status}, \text{deg}) \\
&= (“status_out”, \text{msg}) \text{ if } \text{phase} = “report_status” \\
&= (“deg_out”, \text{msg}) \text{ if } \text{phase} = “report_deg”
\end{aligned}$$

Time advance function:

$$\text{ta}(\text{phase}, \sigma, \text{status}, \text{deg}) = \sigma$$