Applications

Simulation

Simulation: Transactions of the Society for Modeling and Simulation International
2015, Vol. 91(4) 360–382

DOI: 10.1177/0037549715572490
sim.sagepub.com

Multi-component wind turbine modeling and simulation for wind farm operations and maintenance

Eduardo Pérez[1], Lewis Ntaimo[2] and Yu Ding[2]

Abstract

Wind farms produce electricity and provide a source of renewable energy. The growth in wind farm installations in the past 10 years has led to an increase in the number of wind turbines reaching the end of their manufacturing warranties. As a consequence, the wind industry is now facing a rising cost of unscheduled maintenance, which is pushing up the operation and maintenance expenditures. Wind turbines experience stochastic loading due to seasonal variations in wind speed and direction. These harsh operational conditions lead to failures of wind turbine components, which are difficult to predict. Consequently, it is challenging to schedule maintenance actions that will avoid failures. In this paper, we derive algorithms for scheduling wind farm maintenance for wind turbines modeled explicitly as comprising multiple components: the gearbox, power generator, blades and control system. We perform simulations based on a real wind farm with 100 turbines and report on several wind farm performance measures. The results we obtain provide insights regarding how to efficiently manage limited maintenance resources in wind farms. For example, the results show that maintenance policies that consider performing maintenance on multiple components of a wind turbine on the same maintenance scheduled trip provides significant cost savings while reducing the number of turbine failures.

Keywords

Wind energy, wind turbine, simulation, maintenance, scheduling

1. Introduction

Wind farms use a collection of wind turbines to generate electricity using wind. The number of planned wind farms in the USA is growing rapidly and wind power capacity is expected to constitute 20% of the total power capacity by 2030.[1] Wind farms are usually located in parts of the world with high-speed winds throughout the year. Due to harsh weather conditions and seasonal variations in wind speed and direction, wind turbines experience large stochastic forces that often lead to component failures. Failures result in costly repairs, but also revenue losses due to unavailability of the wind turbine to generate electricity. Operation and maintenance (O&M) costs are a significant part of the overall power generation cost and account for roughly 20% of the total cost of energy.[2] Consequently, scheduling maintenance to avoid wind turbine component failures is critical. Since many wind farms are located in remote areas or offshore, they are less accessible. Furthermore, the uncertain nature of the forces experienced by wind turbines makes it difficult to predict the actual condition of wind turbine components, such as the gearbox, power generator, control system and blades, for maintenance purposes.

Current maintenance practices are based mostly on industry experience. *Scheduled maintenance* (*SM*) is the typical maintenance practice for wind turbines. The frequency of maintenance largely depends on the manufacturer's recommendation. In general, *preventive maintenance* (PM) is carried out twice a year on each wind turbine. In addition to regular PM, wind farm operators still have to respond to unanticipated breakdowns, which require *corrective maintenance* (CM). Considering today's trend toward large-scale wind farms and the long distances from the operation and monitoring centers, wind farm operators can save visits to the remote sites by monitoring, detecting and fixing problems before failures occur. Manufacturers have recognized the benefits of condition-based

[1]Ingram School of Engineering, Texas State University, USA
[2]Department of Industrial and Systems Engineering, Texas A&M University, USA

Corresponding author:
Eduardo Pérez, Ingram School of Engineering, Texas State University, 601 University Drive, San Marcos, TX 78666, USA.
Email: eduardopr@txstate.edu

monitoring and most modern wind turbines are equipped with many sensors to help in detecting catastrophic component failures. Because sensors may not provide accurate information on the actual condition of each wind turbine component, deciding *when* and *what* type of maintenance action to undertake is still a challenging problem in wind farm O&M.

To schedule PM and CM for wind turbine component failures it is important to know what components are about to fail, have failed and the type of failures. This enables the acquisition of the right type and number of replacement parts/components, as well as mobilizing limited maintenance crews. Acquiring parts/components often takes several days or weeks (lead time). Since it is better to plan for maintenance before failures happen (PM), it becomes necessary to know the failure behavior of each turbine component. This is useful in devising scheduling algorithms for both PM and CM for unforeseen failures, which is very challenging. In fact, limited progress has been made in the literature to date on multi-component turbine O&M scheduling. The few available scheduling strategies are devised under simplifying assumptions. These assumptions include modeling the wind turbine as having a single component instead of multiple components; ignoring important factors such as lead times for repairing turbines and weather conditions; assuming the availability of unlimited maintenance crews; and assuming perfect sensor information for condition-based monitoring. Also, most work in the literature lack models that enable real-time wind farm O&M decision-making. This work considers a discrete event modeling and simulation (M&S)[3] approach for wind farm O&M planning and *relaxes* such assumptions.

Discrete event M&S is a powerful technique for studying complex systems because it allows for capturing reality at different granularities of time and space, and enables the user to observe the eventual effects of alternative actions on the system. A detailed representation of wind farm O&M is simply too complex to be adequately captured in closed form, thus ruling out analytical solutions. Our work in Byon et al.[4] and Pérez et al.[5] introduced a discrete event system specification (DEVS)[3] model for a wind turbine. This work focused on studying the deterioration and failures of a single component, the *gearbox*, while ignoring the degradation of other components. Consequently, the impact of having a limited number of maintenance crews on maintenance decisions, for example, was not fully studied. In this paper, we build on the work in Byon et al.[4] and Pérez et al.[5] by explicitly modeling multiple components of a wind turbine and studying the impact of the failure of the components on maintenance actions over time. In addition to the *gearbox*, we model the *power generator, blades* and *control system*. Modeling multiple turbine components requires capturing their deterioration behavior simultaneously, as well as their impact on the overall performance of the wind turbine. This leads to building a new simulation model to enable exploring maintenance strategies that take into account each turbine component.

Specifically, this work makes two major extensions from previous work: (1) builds a new multi-component wind turbine simulation model using DEVS; and (2) derives new algorithms for maintenance scheduling that consider the deterioration and failure of multiple components of a wind turbine. The new DEVS simulation model involves a logical design that interconnects multiple components in a wind turbine to mimic the actual operation of a wind turbine in a more realistic manner. It characterizes the turbine's dynamic responses, deterioration, maintenance scheduling and local wind prediction using DEVS. The DEVS formalism is a viable approach for modeling complex systems, such as a wind farm with several wind turbines. It enables building a wind farm simulation platform by developing models at different levels of the wind power system: wind turbine components, wind turbine, wind farm, power grid and network. In terms of algorithms for maintenance scheduling, the key issue is when to schedule maintenance given that each wind turbine has multiple components that degrade and fail at different rates. Furthermore, there are limitations in terms of the availability of maintenance crews and wind turbine components, which impose lead times on maintenance schedules. This requires new maintenance strategies that would, for example, enable scheduling a single maintenance trip to attend to multiple components before they fail so that the wind farm can maintain the desired amount of power generation.

The contributions of this work include the following: (a) a discrete event-based simulation for commercial size wind farms; (b) new maintenance strategies for commercial size wind farms; (c) computational results to assess the performance of a wind farm under different scheduling maintenance strategies; and (d) insights into the implications of maintenance capacity limitations and component replacement lead times on wind farm performance. The rest of the paper unfolds as follows. Section 2 reviews related work on wind farm O&M simulation. Section 3 provides details of our simulation model emphasizing wind turbine model abstraction and overall operation of the wind farm simulation. Maintenance scheduling strategies and practices are presented in Section 4. Section 5 presents a computational study with results and discussion. Finally, the paper ends with concluding remarks and future directions for research in Section 6.

2. Literature review

There are several research papers that use M&S to study wind farm O&M scheduling. For instance, Rademakers et al.[6] use Monte Carlo simulation for O&M of offshore wind farms. The authors consider a 100 MW wind farm and model the O&M aspects by considering several challenging factors, such as the weather, for repairing wind turbines. In their paper, turbine component failures are stochastic and are generated using reliability distributions and mean-time-to-failure (MTTF). Their model does not consider PM, so maintenance actions are only performed after a component failure. Results show that revenue losses account for 55% of the total maintenance cost due to long lead times for component replacements and waiting times for favorable weather conditions. Other similar studies include Ribrant,[7] Vittal and Teboul,[8] Van Bussel[9] and Hendriks et al.[10]

These works do not consider the status and degradation pattern of wind turbine components. In contrast, McMillan and Ault[11] employ Markov models to represent the degradation of wind turbine components (electronic related, blades, gearbox and generator). The authors use simulation to quantify the cost-effectiveness of condition monitoring equipment and compare the performance of two maintenance policies: *SM* and condition-based maintenance (*CBM*). Their models assume that condition monitoring equipment reveals the degradation status of each component perfectly. Maintenance activities are subject to weather conditions and component replacement lead times. The results show the economic benefit of *CBM* versus *SM* under different wind profiles, downtime durations, replacement lead times and costs.

Andrawus et al.[12] develop an optimal wind turbine component replacement policy for a wind farm with 26 turbines. The authors use the Weibull distribution to model the failure pattern of different wind turbine components (main shaft, main bearing, gearbox and generator) and test their replacement strategy using Monte Carlo simulation. The results show that to minimize maintenance costs, the gearbox and the generator has to be replaced every six and three years, respectively. Likewise, Hall and Strutt[13] use probabilistic failure models to predict wind turbine component failures and use Monte Carlo simulation to assess replacement strategies.

Byon et al.[4] develop a DEVS[3]model for a wind turbine. Their wind turbine model considers the deterioration and failure of a single component, the gearbox. They implement both *SM* and *CBM* algorithms and report on results involving a wind farm with 100 turbines located in West Texas. The results demonstrate the advantages of using *CBM* over *SM* in terms of cost savings and power generation. Pérez et al.[5] extends the simulation model using stochastic DEVS[14] to model the gearbox degradation process based on a Markov model. The results show that *CBM* provides about 10% extra power generation on average compared to *SM* over a 20-year period.

In recent years, a group of analytical models[15–19] have been proposed to study O&M of wind farm systems. Most of these studies focus on taking advantage of *CBM*-type maintenance for wind farms. Although these studies do not use simulation, they still reveal good insights on how condition monitoring information from wind turbines can aid in improving current maintenance practices. The *CBM* strategy considers economic dependencies among wind turbine components (blades, main bearing, gearbox and generator) and determines optimal maintenance actions based on failure probability thresholds.

Despite the benefits of discrete event M&S, O&M studies using this technique are limited in the wind farm literature. The limitations of current simulation models can be summarized as follows. (a) Current wind turbine model abstraction is oversimplified. Several studies are based on a single wind turbine or a single wind farm, but with the assumption of identical turbines. In addition weather uncertainty, wind turbine types and operational strategies are generally not considered. (b) PM actions are considered to be replacement, which returns components to the as-good-as-new state.[20] According to Spinato et al.,[21] repair actions must include new parts, adjustments and preventive actions, such as lubrication and cleaning. Table 1 gives a summary of the most closely related work to this paper and highlights the differences in terms of the key factors considered in each study.

3. Wind turbine simulation model

Wind farm M&S development using DEVS[3] involves the following tasks: (a) developing wind farm basic models; (b) coupling the basic models to create more complex models; (c) building an experimental frame to allow for a suite of simulation experimental choices; (d) implementing the models on a computer; and (d) testing and validating the simulation model. We design the simulation model to handle multiple random events, including wind turbine component failures, weather disruptions, parts replacement lead times and maintenance duration. The outputs from the simulation include the amount of *power generated, availability factor* and *number of component failures* under different maintenance strategies and maintenance capacities.

DEVS is a M&S approach that enables building dynamical models in a hierarchical manner, starting with simple models and coupling them to create more complex models. DEVS basic models are called *atomic* models, and these can be linked to create *coupled* models. An atomic model is the lowest level model and contains structural dynamics. A coupled model comprises one or more atomic and/or coupled models, and is constructed in a hierarchical manner and allows for model modularity. Each DEVS component (atomic or coupled model) can be viewed as a system with *inputs, states* and *outputs*. It has an internal structure that dictates how inputs and states are transformed to outputs. The components are coupled together to create systems, which themselves can be components in

Table 1. Comparison of wind farm simulation studies.

Key Factor	Andrawus et al.[12]	McMillan and Ault[11]	Tian et al.[19]	Byon et al.[4]	This work
Number of turbines considered in analysis	26	1	5	100	100
Power generation considers turbine type	×	×		×	×
Power generation considers weather				×	×
Power generation considers turbine location				×	×
Power generation considers turbine height				×	×
Wind turbine components	4	4	4	1	4
Component degradation	Weibull distribution	Markov model	Weibull distribution	Markov model	Markov model
Corrective maintenance time duration (days)	×			×	×
Preventive maintenance duration (days)	×			×	×
Maintenance capacity limitations					×
Component replacement lead time				×	×
Maintenance re-scheduling due to harsh weather conditions					×

larger systems. Thus, components are coupled together to create a DEVS simulation model that captures the dynamical behavior of a system of interest. An atomic model has *input* and *output ports* through which the couplings with other atomic or coupled models are done. These couplings enable interaction via message passing between the linked models. The model receives external input through the inputs ports, processes the input and generates output through the output ports. The dynamic behavior of an atomic model is captured using mathematical *functions* that define how the model transitions between its *states*. Atomic and coupled models allow for processing multiple inputs simultaneously. We omit the mathematical definitions of DEVS atomic and coupled models and refer the interested reader to Zeigler et al.[3]

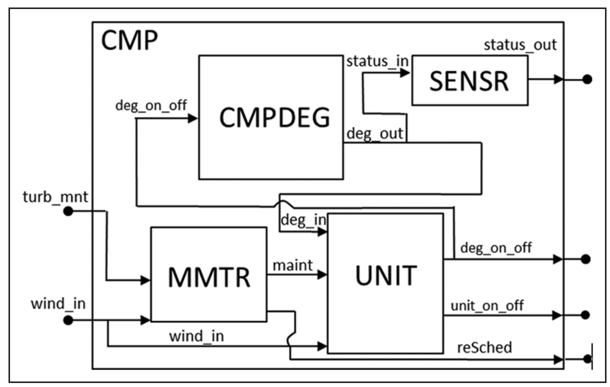

Figure 1. Wind turbine component coupled model.

3.1 Multi-component wind turbine model

We abstract wind turbine components as dynamical subsystems that together represent a wind turbine and use DEVS atomic and coupled models to characterize their structure and behavior. Our simulation model of a wind turbine component (CMP) is depicted in Figure 1 using a block diagram. Due to space limitation, the DEVS mathematical expressions of these atomic and coupled models are given in the Appendix. The CMP coupled model comprises four atomic models: component degradation (CMPDEG), maintenance monitor (MMTR), unit (UNIT) and sensor (SENSR). CMPDEG and SENSR were introduced by Byon et al.[4] in the context of having only one component, the gearbox. CMPDEG captures the degradation behavior of the turbine component CMP. SENSR provides current status information of the turbine component. Fault diagnosis based on sensor measurements is non-trivial due to the wind turbine's non-steady operating conditions. Therefore, it is often very difficult to obtain the actual state of a component based only on sensor measurements. In this work, a hidden Markov model (HMM) is adopted to model the uncertainty in the sensor measurements. UNIT captures the status of the turbine component and is specified as gearbox, power generator, blades or control system. Each UNIT atomic model has the following *four* states: ''normal'', the as-good-as-new operation state; ''alert'', a deteriorated but still safe to operate state; ''alarm'', a deteriorated state that could fail soon; and ''fail'', the state in which the turbine component is no longer functioning. UNIT is always initialized in the ''normal'' state, and its stochastic deterioration is characterized by a *probability transition matrix P*. Each type of component will have its own probability transition matrix, which is typically developed based on historical reliability data of the component.

An example of a transition matrix P for a turbine blade is given as follows:

$$P = \begin{pmatrix} 0.990 & 0.009 & 0.001 & 0.000 \\ 0.000 & 0.985 & 0.010 & 0.005 \\ 0.000 & 0.000 & 0.985 & 0.015 \\ 0.000 & 0.000 & 0.000 & 1.000 \end{pmatrix}.$$

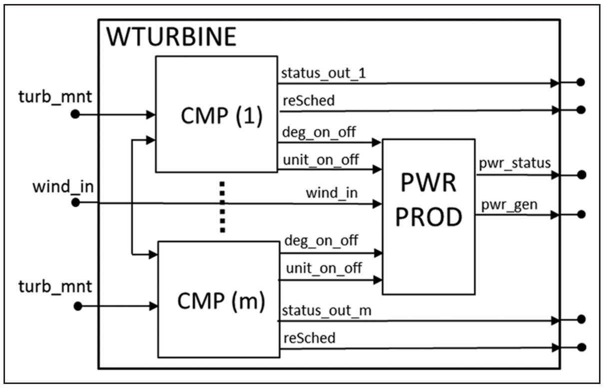

Figure 2. Multi-component wind turbine coupled model.

The columns of the matrix correspond to the component states, ''normal'', ''alert'', ''alarm'' and ''fail'', in that order. If no maintenance is performed after some stochastic time duration, the transition matrix indicates that the wind turbine blade will stay in ''normal'' state with probability 0.99, or transitions from ''normal'' to ''alert'' with probability 0.009, or to ''alarm'' with probability 0.001. When the blade is in ''alert'' state, the model remains in this state with probability 0.985, or transitions to ''alarm'' or ''fail'', with probability 0.01 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 the model reaches the ''fail'' state it stays in this state with probability one. Only after CM or component replacement will the model be re-initialized to ''normal'' state.

Finally, the MMTR atomic model is responsible for rescheduling maintenance actions suspended or canceled due to bad weather conditions. MMTR keeps track of maintenance actions already scheduled for the component and based on current weather conditions decides if maintenance actions need to be rescheduled. The UNIT model keeps track of the component state based on the degradation information provided by the CMPDEG and the maintenance information supplied by the MMTR. The multi-component wind turbine (WTURBINE) coupled model is shown in Figure 2 and contains *m* CMP models (*m* number of components) and a power production (PWRPROD) atomic model. The PWRPROD model has two main states: ''active'' and ''passive''. When in ''passive'' state the wind turbine generates no power and this occurs when the wind turbine is under maintenance or when weather conditions force the turbine to shut down. In contrast, when ''active'', the PWRPROD model uses the current wind speed at the turbine to compute the amount of power generated for a specific period of time. The amount of power generated by the wind turbine is calculated using a power curve model, as described by Byon et al.[4]

3.2 Wind farm simulation

The DEVS wind farm simulation model is depicted in Figure 3. The model has three main coupled models: wind farm (WF), experimental frame (EF) and operations and maintenance (OPMNT). The wind farm (WF) coupled model represents the wind farm. This coupled model can represent any type of wind farm, since it can be configured according to user specifications. For example, the amount, type, height and location of the wind turbines are parameters provided by the user. The wind farm coupled model contains several WTURBINE models, which represent the turbines at the location under study. EF and OPMNT were introduced by Byon et al.[4] EF is a coupled model and is linked to both WF and OPMNT and is used for specifying and running experiments of interest by the user. For example, EF allows for specifying parameters and computing performance measures. It collects information from the components while the simulation is running and computes performance measures at the end of each simulation run. The performance measurements for this study are discussed later in Section 5.2. EF has two atomic models, namely, wind generator (WGENR) and transducer (TRANSD). WGENR generates wind speed using a spatio-temporal model in a hierarchical manner to generate sequences of wind speed at turbine locations and heights, as explained by Byon et al.[4] Finally, the TRANSD atomic model collects information of interest from WF and computes performance measures specified by the user.

Wind farm maintenance and operation activities are managed by the OPMNT coupled model in the simulation. This model comprises two atomic models: the maintenance scheduler (MSCHEDR) and maintenance generator (MGENR). MSCHEDR implements different management strategies or algorithms for scheduling wind turbine maintenance. In this work, we consider several strategies for maintenance scheduling involving multiple components: the gearbox, power generator, blades and control system, and these are discussed in Section 4. The MSCHEDR atomic model communicates with the MGENR atomic model when a maintenance procedure is scheduled. MGENR generates the maintenance activity at the time scheduled by the MSCHEDR.

4. Wind turbine maintenance strategies

Since each wind turbine model comprises multiple components, scheduling maintenance becomes more complicated than for the case where only a single component is considered. In the single component setting in Byon et al.,[4] wind farm operations were simulated under two maintenance strategies, *SM* and *CBM*. *SM* reflects the common maintenance practice and in the implementation in Byon et al.[4] PM actions were performed twice a year in low windy conditions regardless of the deterioration status of the

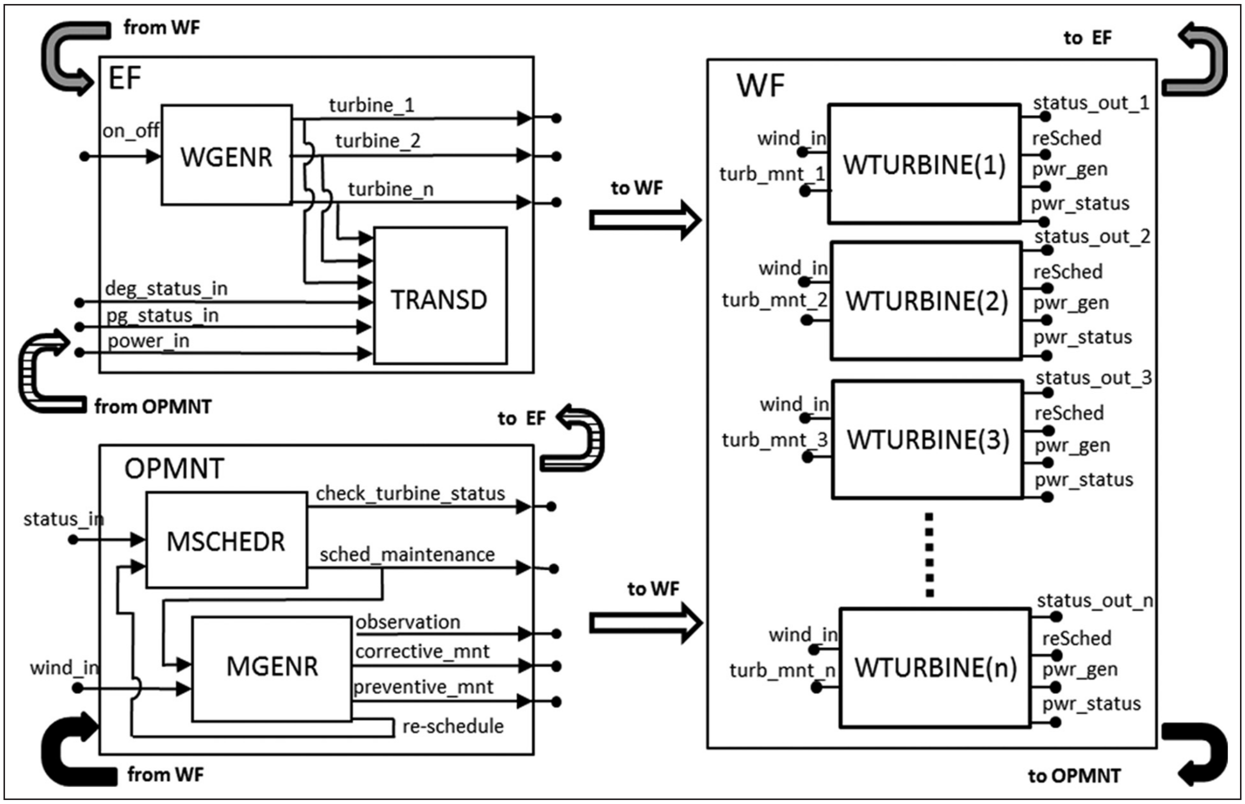

Figure 3. Discrete event system specification wind farm simulation model.

gearbox. Under *CBM*, PM actions were carried out only when sensors produced ''alarm'' signals. Furthermore, for both *SM* and *CBM*, CM was performed when there was an unplanned failure with a lead time of six weeks before repairing the turbine. In this work we extend *SM* and *CBM* to the multi-component setting and introduce a new scheduling algorithm, termed *concurrent condition-based maintenance* (*CCBM*). *CBM* and *CCBM* maintenance strategies use sensor information from the wind turbine to decide *when* and *what* type of maintenance to perform based on the status of *all* turbine components. The goal of both maintenance strategies is to monitor, detect and service or replace wind turbine components before a failure occurs. Both *CBM* and *CCBM* algorithms use information from turbine sensors to decide when to schedule PM and CM. However, they differ in the way maintenance is scheduled. *CCBM* plans maintenance of multiple components within the same turbine on a single maintenance trip, whereas *CBM* plans maintenance of a single component per maintenance trip.

To mathematically describe the new *SM, CBM* and *CCBM* maintenance scheduling algorithms, we begin with some notation in Table 2. For convenience, we use the following symbols: $\leftarrow$ denotes assignment and == denotes (equality) comparison. We define the set of day and turbine pairs (d, j) for maintenance crew r as $U_r = \{(d, j) | 1 \leqslant d \leqslant h_0, 1 \leqslant j \leqslant |J|\}$, where h_0 is the last day of the scheduling horizon and $|J|$ is the total number of turbines in the wind farm. The set U_r includes all the days that are already occupied in the schedule for maintenance crew r. For ease of exposition, we first describe two basic *functions* implemented by all the scheduling algorithms, named CheckSchedule() and ScheduleOnline().

Figure 4 describes the CheckSchedule() function, which checks the availability of a maintenance crew r during a given time interval $[d, d + \alpha_{mi}]$. The function returns a Boolean indicating whether (*true*) or not (*false*) the time interval is available. The function simply checks whether or not any of the days from day d to $d + \alpha_{mi}$ are already booked in the schedule for maintenance crew r. The maintenance schedule set G is initialized in line 1. A counter named *day* is used to keep track of the current day in the schedule. Line 2 uses a *for-loop* to define the number of days in a row needed to schedule the maintenance requested. If one of the days in the time period is already occupied, maintenance cannot be scheduled (lines 3–5). If the current *day* is available, it is added to G and the day counter is incremented by one day. When the number of consecutive days required for maintenance is found, maintenance is scheduled.

The ScheduleOnline() function, detailed in Figure 5, finds the first maintenance crew r available for performing

Table 2. Maintenance scheduling algorithm sets and parameters.

Sets

J : Set of turbines, indexed j.
I : Set of components in a wind turbine, indexed i (i = 1 gearbox, i = 2 power generator, i = 3 blades, and i = 4 control system).
D: Set of days in the scheduling horizon, indexed d.
S: Set of states, indexed s (s = 1 for "normal"; s = 2 for "alert"; s = 3 for "alarm"; and s = 4 for "fail").
S_{ij}: Set of states of component i of turbine j.
R: Set of maintenance crews, indexed r.
T: Set of turbines assignments under SM.
T_r: Set of turbines assigned to maintenance crew r under SM.
M: Set of maintenance types, indexed m (m = 1 for corrective and m = 2 for preventive maintenance).
U_r: Set of day and turbines pairs (d, j) for maintenance crew r.
G: Set of day and turbines pairs (d, j) used to check the availability of a maintenance crew.

Parameters

d_j: Day of failure for turbine j.
$\bar{d}$: First day of spring season.
$\underline{d}$: First day of fall season.
k: Number of maintenance crews, $k = |R|$.
h : Number of days planned for system operation.
ℓ_{mi}: Lead time (days) required for performing maintenance type m on component i.
α_{mi}: Time (days) required to perform maintenance type m on component i.
β_{ij} : Current state of component i of turbine j.
μ : Number of wind turbines assigned to each maintenance crew under SM, $\mu = |J|/k$.

```
CheckSchedule (U_r, d, α_mi, j ).
1-  G ← {∅}
2-  for day = d to d + α_mi do
3-    if day ∈ U_r then
4-      G ← {∅}
5-      return false
6-    else
7-      G ← G ∪ {(day, j)}
8-      day = day + 1
9-    end
10- end
11- U_r ← U_r ∪ G
12- return true
```

Figure 4. Pseudocode for CheckSchedule().

```
ScheduleOnline(U_r, day, α_mi, j)
1-  R* ← R
2-  while R* ≠ {∅} do
3-    θ ← CheckSchedule (U_r, day, α_mi, j )
4-    if θ == false then
5-      R* ← R*\{r} and go to step 2
6-    else
7-      End
8-    end
9-  end
10- day = day + 1 and go to step 1
11- end
```

Figure 5. Pseudocode for ScheduleOnline().

maintenance. Line 1 assigns the set of maintenance crews to set R^*. Line 2 keeps track of the maintenance crew under search. Each maintenance crew is searched until the one with the earliest availability for scheduling the maintenance requested is found. Line 3 invokes the CheckSchedule() function (Figure 4). The algorithm terminates if maintenance can be scheduled (line 7), unless the algorithm reaches the end of the scheduling horizon. If the maintenance cannot be scheduled within the fixed scheduling horizon, the algorithm terminates. If maintenance cannot be scheduled, the algorithm removes the current maintenance crew from the set R^* and returns to line 2 (line 5). If no maintenance crew is available on the time period defined in line 2, the algorithm increases the *day* counter by one unit and repeats the scheduling process until the maintenance is scheduled (line 10). A rolling scheduling horizon can also be considered to make sure that that a feasible schedule can be found.

4.1. Scheduled maintenance

The objective of *SM* is to prevent wind turbine component failures by performing PM during the year. The time between two consecutive PM actions is based on the manufacturer's recommended maintenance schedule. Under *SM*, PM actions are performed on each turbine at least twice a year during low wind speed seasons. We assume that PM under *SM* is performed at the beginning of the spring and fall seasons, since we study wind farms located in Texas, USA. In addition, CM actions are performed to respond to

```
TurbineTeamAssignment ( j*, μ )
1-  μ* ← μ,  T ← {∅},  T_r ← {∅}, r = 1
2-  for j = j* to |J|
3-      if j ≤ μ* then
4-          T_r ← T_r ∪ {j}
5-      else
6-          μ* = μ* + μ
7-          T ← T ∪ {T_r}
8-          r = r + 1
9-          T_r ← T_r ∪ {j}
10-     end
11- end
12- return T
```

Figure 6. Pseudocode for TurbineTeamAssignment().

```
SchedulePM(d*, T ).
1-  Let day = d*, m = 1
2-  while R ≠ {∅} do
3-  |  U_r ← {∅}
4-  |  while T_r ≠ {∅} do
5-  |  |  θ ← CheckSchedule (U_r, day, α_mi, j )
6-  |  |  if θ == true then
7-  |  |  |  T_r ← T_r\{j}
8-  |  |  else
9-  |  |  |  day = day + 1 and go to step 4
10- |  |  end
11- |  end
12- |  day = d*
13- end
14- end
```

Figure 7. Pseudocode for SchedulePM().

unanticipated breakdowns. The steps of the *SM* algorithm are listed next. We use a Boolean variable θ to store the output returned by the CheckSchedule() function. Recall that the variable θ takes the value *true* if maintenance crew r is available and *false* otherwise. For ease of exposition, we first describe two additional functions implemented by the *SM* algorithm, named TurbineTeamAssignment() and SchedulePM(). Both of these functions are used in the initialization of the *SM* algorithm.

The TurbineTeamAssignment() function is described in Figure 6 and assigns a group of turbines to each maintenance team. Line 1 initializes the parameters required for the function. Line 2 begins the assignment process turbine by turbine. A group of turbines is assigned to the same maintenance team until the limit μ^* is reached (line 4). When the limit is reached the algorithm moves to the next team (line 8) and restarts the process until all turbines are assigned.

The SchedulePM() function is described in Figure 7. This function uses the turbine-teams assignments to schedule PM. Line 1 initializes the day and the maintenance type. This function uses the assignment sets T_r to schedule PM to each turbine using the corresponding maintenance team (lines 3–10). After PM is scheduled to all turbines assigned to team r, the process is repeated for the rest of the teams (line 12). The *SM* algorithm steps can be given as follows.

***SM* algorithm:**

Step 0 **Initialization:** Set $\beta_{ij} = 1, \forall i \in I, \forall j \in J$, $h \leftarrow h_0$, $d \leftarrow d_0$, $\mu \leftarrow \mu_0, j \leftarrow 1$, and $U_r \leftarrow \{\emptyset\}$
  $T \leftarrow$TurbineTeamAssignment (j, μ)
  for $year = 0$ **to** $h/365$ **do**
    SchedulePM($\bar{d} + year * 365, T$)
    SchedulePM($d + year * 365, T$)
  end
Step 1 **Online Requests:**
  for $d = 1$ **to** h **do**
    if $\beta_{ij} == 4$ **then** $m = 1$, $day = d + \ell_{mi}$
      ScheduleOnline $(U_r, day, \alpha_{mi}, j)$
    end
  end
Step 2 **Termination:**
  if $day == h$ **then**
    End
  end

The algorithm has three major steps: *initialization, online requests* and *termination*. The initialization process assigns initial parameter values and schedules PM twice a year for all the turbines. In Step 1 the algorithm tracks the condition of the wind turbines. An online oracle will provide the status information from the turbines as time progresses. Turbine status reports arrive one at a time in an online fashion. If turbine one (j = 1) reports the *failure* of the gearbox (i = 1), then $\beta_{11} = 4$ and CM will be scheduled for the turbine using the ScheduleOnline() function. Then the algorithm begins the search to schedule maintenance on $day = d_j + \ell_{mi}$. The algorithm takes into account the day the report is received and the lead time required to assemble the team and acquire the parts and equipment needed to perform the maintenance. Performing CM after a component failure involves replacing the component, which may not be readily available and will have to be ordered. Lastly, in Step 2 the algorithm terminates if the current day equals the scheduling horizon.

4.2. *Condition-based maintenance*

As in *SM*, the objective of *CBM* is to prevent wind turbine component failures by performing PM as needed based on the condition of the components. *CBM* uses information from wind turbine sensors to decide when to schedule PM. Therefore, the *CBM* algorithm uses information about the current state of the wind turbine components to decide when to schedule PM. The four degradation states (*normal, alert, alarm* and *fail*) for a wind turbine component, explained in Section 3.2, are used as input to the *CBM* algorithm. The *CBM* algorithm is a simplified version of the maintenance strategy proposed by Byon et al.[15] The steps of the *CBM* algorithm are listed next.

***CBM* algorithm:**
Step 0 **Initialization** Set $\beta_{ij} = 1,\ \forall i \in I,\ \forall j \in J,\ h \leftarrow h_0$, $d \leftarrow d_0$, $\mu \leftarrow \mu_0$, and $U_r \leftarrow \{\emptyset\}$
Step 1 **Online Requests:**
  for $d = 1$ **to** h **do**
    if $\beta_{ij} == 3$ **then** $m = 2$, $day = d + \ell_{mi}$
      ScheduleOnline $(U_r, day, \alpha_{mi}, j)$
    else $\beta_{ij} == 4$ **then** $m = 1$, $day = d + \ell_{mi}$
      ScheduleOnline $(U_r, day, \alpha_{mi}, j)$
    end
  end
Step 2 **Termination:**
  if $d == h$ **then**
    End
  end

The *CBM* algorithm has three major steps: *initialization, online requests* and *termination*. In Step 0 the algorithm parameters are initialized. Step 1 is named online requests. In this step, PM is scheduled every time a turbine component reports an alarm state. For instance, if turbine one ($j = 1$) reports that the gearbox ($i = 1$) is in *alarm* state, then $\beta_{11} = 3$, and PM maintenance is scheduled using the ScheduleOnline() function. In addition, every time a turbine component fails CM is scheduled for the turbine using the ScheduleOnline() function. As in the *SM* algorithm, it is assumed that the online oracle provides the status information from the turbines as time progresses. Lastly, in Step 2 the algorithm terminates if the scheduling horizon comes to an end.

4.3. Concurrent condition-based maintenance

As in *CBM*, the *CCBM* algorithm uses information from turbine sensors to decide when to schedule preventive and CM. However, *CCBM* plans maintenance of multiple components within the same turbine on a single maintenance trip. Cost savings can be achieved by decreasing the number of visits to the same wind turbine. Instead of performing maintenance to one component per maintenance visit, *CCBM* performs maintenance to multiple components (as needed) every time maintenance is requested from a turbine. The four degradation states (*normal, alert, alarm* and *fail*) for a wind turbine component, explained in Section 3.2, are used as input in the *CCBM* algorithm. The steps of the *CCBM* algorithm are listed next.

***CCBM* algorithm:**
Step 0 **Initialization** Set $\beta_{ij} = 1,\ \forall i \in I,\ \forall j \in J,\ h \leftarrow h_0$, $d \leftarrow d_0$, $\mu \leftarrow \mu_0$, and $U_r \leftarrow \{\emptyset\}$
Step 1 **Online Requests:**
  for $d = 1$ **to** h **do**
    if $\beta_{ij} == 3$ **then**
      Let $i' = i$ and define set G as the set of components with $i \in I \backslash \{i'\}$
      Let P be the set of components with $\beta_{ij} == 2$ where $i \in G$
      $m = 2$
      $day = d + \max_{i \in P \cup \{i'\}} \ell_{mi}$
      $\alpha = \sum_{i \in P \cup \{i'\}} \alpha_{mi}$
      ScheduleOnline (U_r, day, α, j)
    else if $\beta_{ij} == 4$ **then** $m = 1$, $day = d_j + \ell_{mi}$,
      Let $i' = i$ and define set G as the set of components with $i \in I \backslash \{i'\}$
      Let P be the set of components with $\beta_{ij} == 2$ where $i \in G$
      Let Q be the set of components in G with current state $\beta_{ij} == 3$
      $day = d_j + \ell_{1i}$(CM lead time is always larger)
      $\alpha = \sum_{i \in P \cup Q} \alpha_{2i} + \alpha_{1i'}$
      ScheduleOnline (U_r, day, α, j)
    end
  end
Step 2 **Termination:**
  if $d == h$ **then**
    End
  end

The *CCBM* algorithm also has three major steps: *initialization, online requests* and *termination*. In Step 0 initial values are assigned to the algorithm parameters. Step 1 processes online requests. If PM is needed by one turbine component ($\beta_{ij} = 3$), the algorithm checks the status of the other components within the same turbine. If one or more components are found to be in *alert* ($\beta_{ij} = 2$), PM is scheduled for those components, meaning that they will be maintained on the same visit. The first day that the algorithm uses to begin the search for maintenance is $d_j + \max_{i \in P \cup \{i'\}} \ell_{mi}$. The algorithm considers maintenance lead time of each turbine component requiring service and uses the maximum lead time to determine the first day to begin the search for maintenance. If CM is required for one of the turbine components ($\beta_{ij} = 4$), the algorithm considers the lead times required by the other components requiring PM at the time of the failure and selects the maximum lead time to determine the first day to begin the search for maintenance. Lastly, in Step 2 the algorithm is terminated if the scheduling horizon comes to an end.

5. Application

We implemented our DEVS simulation model in Java using DEVSJAVA.[22] The model represents a 100-unit wind farm located in West Texas. This wind farm operates

Table 3. Corrective maintenance cost.[25]

Component	Material ($)	Labor ($)	Production loss ($)	Total maintenance cost ($)
Gearbox	102,401.25	5976.00	3155.33	111,532.58
Power generator	38,912.06	3984.00	2760.91	45,656.97
Blade	59,004.70	3984.00	2760.91	65,749.61
Control system	6629.50	3984.00	2760.91	13,374.40

Table 4. Preventive maintenance cost.[11,25]

Component	Material ($)	Labor ($)	Production loss ($)	Total maintenance cost ($)
Gearbox	10,240.13	597.60	315.53	11,153.26
Power generator	3891.21	398.40	276.09	4565.70
Blade	5900.47	398.40	276.09	6574.96
Control system	3314.75	398.40	276.09	3989.24

365 days a year, 24 hours a day. The simulation model uses wind speed measurements reported by the West Texas Mesonet.[23] Each wind turbine model is represented by four components, the gearbox, power generator, blades and control system. A probability transition matrix P was used to model the degradation and failure rate of each wind turbine component, as explained in Section 3.2. We assumed a variable number of maintenance teams and that each maintenance team performed the same tasks and was capable of performing maintenance on any wind turbine. Lead times were required for assembling a maintenance team, request part/component replacements and to rent equipment (e.g. a crane) to access the wind turbines.

5.1 Experimental setup

We set up experiments based on the 100-turbine wind farm operating for a period of 20 years, which is the average lifetime of a wind turbine.[24] The number of replications was arbitrarily set to 20. Computational experiments were conducted using a DELL X5355 with 2 Intel Xeon X processors at 2.66 GHz each with 12.0 GB of RAM. As stated in Section 1, this computational study had three major goals: (1) demonstrate that theoretical maintenance strategies based on simplified assumptions are not always optimal for commercial size wind farms; (2) improve the performance of current scheduling maintenance strategies used in practice; and (3) study the implications of maintenance capacity limitations and component replacement lead times.

The experiments considered three key factors, namely *maintenance strategy, maintenance capacity* and component replacement *lead times*. Four performance measures (responses) were considered: *power generated* (PG), *systems availability* (SA), *number of failures* (NF) and *maintenance cost* (MC). To test the first factor, the wind farm simulation model was run using the three maintenance strategies discussed in Section 4: *SM, CBM* and *CCBM*. Recall that *CBM* is based on the maintenance strategy developed by Byon et al.[4] and we use it as a point of comparison or benchmark in our simulations. The second factor examines the effect of maintenance capacity limitations. Previous research on wind farm O&M assumes the availability of an unlimited number of maintenance teams. We relaxed this assumption by running the wind farm simulation model with varying numbers of maintenance teams: 5, 10, 20 and 50. The third factor looks at the effect of lead time duration on O&M. In case of maintenance, one should consider the lead time for organizing maintenance teams and parts/components. For example, it can take several weeks for a component such as a gearbox to be delivered. Previous research on wind farm O&M using simulation[11,12,19] has seldom considered this factor. We considered lead times of one, three, six and eight weeks for each wind turbine component. Tables 3–5 show the cost figures reported by McMillan and Ault[11] and Andrawus et al.,[25] which we used to estimate the maintenance cost of O&M strategies. The performance measures PG, SA, NF and MC were then determined from the simulation runs.

Table 5. Cost of access.[25]

Component	Access cost ($)
Gearbox	18,724.80
Power generator	14,043.60
Blade	14,043.60
Control system	14,043.60

Table 6. Computational results for scheduled maintenance (*SM*).

Maintenance capacity (# teams)	Replacement lead times (days)				Performance measures			
	Gearbox	Power generator	Blades	Control system	Avg. power generated (MW)	Avg. system availability (%)	Avg. number of failures	Avg. number of preventive actions
5	1	4	2	1	11,286,321.71	0.813	885.60	3215.00
5	3	4	2	1	11,258,940.10	0.811	879.60	3210.20
5	4	4	2	1	11,245,998.32	0.810	880.00	3194.80
5	8	4	2	1	11,014,127.66	0.794	825.40	3193.20
10	1	4	2	1	11,725,669.59	0.845	990.20	3359.00
10	3	4	2	1	11,711,169.10	0.844	985.40	3355.80
10	4	4	2	1	11,704,376.95	0.844	985.40	3348.80
10	8	4	2	1	11,432,692.71	0.824	915.80	3278.80
20	1	4	2	1	12,165,017.47	0.877	1094.80	3503.00
20	3	4	2	1	12,163,398.11	0.877	1091.20	3501.40
20	4	4	2	1	12,162,755.59	0.877	1090.80	3502.80
20	8	4	2	1	11,851,257.76	0.854	1006.20	3364.40
50	1	4	2	1	12,346,006.79	0.890	1153.20	3638.60
50	3	4	2	1	12,345,835.59	0.890	1149.00	3638.20
50	4	4	2	1	12,345,816.90	0.890	1150.20	3640.60
50	8	4	2	1	12,003,894.96	0.865	1044.80	3487.80

5.2 Computational results

The simulation results for the three maintenance scheduling algorithms, *SM, CBM* and *CCBM*, are reported in Tables 6–8, respectively. Each table reports the algorithm performance (response) at different levels of maintenance capacity and component replacement lead times. The observed variability in the performance measures is relatively small and, thus, only the average values are reported. The maximum 95% confidence interval limits for the performance measures are $\pm 2\%$ for power generated, $\pm 1.8\%$for system availability and $\pm 1\%$ for number of failures. In terms of lead time levels, only the results for the gearbox component are reported. This is because the results obtained for the other components are similar and are thus omitted.

SM is the traditional industry practice for O&M of wind farms. The best performance in terms of power generation for the *SM* algorithm (Table 6) is observed when the number of maintenance teams is at the maximum and the lead time for the gearbox is at the minimum. This indicates that *maintenance capacity* has a significant impact on the performance of *SM*. The results show a percentage increase of about 10% in *system availability* and *power generated* when the number of maintenance teams is increased from five to 50. A rise in the average *number of failures* is also observed when the number of maintenance teams is increased using *SM*. Since *SM* schedules PM in advance twice a year for each turbine, this results in busy schedules for maintenance teams during certain periods of the year. When the maintenance capacity is low, system availability decreases because the number of maintenance teams to fix failed turbines is inadequate. In this case, failed turbines are not operational for a longer period of time under *SM*. Also, a higher average number of failures is observed when maintenance capacity is high. Since the system is now operating for a longer period of time, the turbines are expected to fail more. Remember that under *SM* the state of the components is not taken into account at the time of scheduling PM. Thus, the chance of component failures is not necessarily reduced. *Lead time* appears to have relatively less significant impact when compared to *maintenance capacity*. However, a 2.8% average percentage decrease in the power generation is observed when, for example, the lead time for the gearbox goes from one week to eight weeks. The pattern is repeated for all levels of maintenance capacities.

Table 7 shows the results for the condition-based monitoring (*CBM*) algorithm. *CBM* schedules maintenance actions based on the condition of the wind turbine components. The best performance of *CBM* is observed when the lead time of the gearbox is at the two lower levels (one and three weeks) and the number of maintenance teams is greater or equal to 10. The results show a percentage increase in system availability and power generated of about 3% when the number of maintenance teams is increased from five to 10. Also, a decrease in the average number of failures is observed when the number of maintenance teams is increased. Since a higher number of maintenance crews are available, the waiting time for PM decreases and wind turbines can be served sooner. When compared to *SM, CBM* provides an average 9% percentage increase in power generation and a percentage decrease in the number of failures of about 52%. In terms of cost,

Table 7. Computational results condition-based monitoring (*CBM*).

Maintenance capacity (# teams)	Replacement lead times (days)				Performance measures			
	Gearbox	Power generator	Blades	Control system	Avg. power generated (MW)	Avg. system availability (%)	Avg. number of failures	Avg. number of preventive actions
5	1	4	2	1	12,663,429.51	0.913	520.80	962.80
5	3	4	2	1	12,642,855.10	0.911	505.80	1007.60
5	4	4	2	1	12,636,514.63	0.911	504.00	1018.00
5	8	4	2	1	12,473,949.97	0.899	501.60	1015.20
10	1	4	2	1	12,994,759.62	0.937	470.80	1076.20
10	3	4	2	1	12,992,814.32	0.937	463.00	1149.80
10	4	4	2	1	12,988,165.28	0.936	464.80	1136.20
10	8	4	2	1	12,765,720.99	0.920	460.60	1141.00
20	1	4	2	1	12,995,478.47	0.937	469.60	1077.00
20	3	4	2	1	12,993,616.16	0.937	464.00	1150.80
20	4	4	2	1	12,989,100.31	0.936	463.40	1136.20
20	8	4	2	1	12,766,754.85	0.920	460.40	1141.60
50	1	4	2	1	12,995,478.47	0.937	469.80	1077.00
50	3	4	2	1	12,993,616.16	0.937	464.00	1150.20
50	4	4	2	1	12,989,237.15	0.936	463.40	1136.20
50	8	4	2	1	12,766,754.85	0.920	461.40	1141.60

CBM provides savings of about 60% when compared to *SM*.

Table 8 provides the results for the *CCBM* algorithm. *CCBM* schedules maintenance actions based on the condition of the wind turbine components and allows for maintenance to multiple components on the same maintenance trip. The best performance of *CCBM* is observed when the number of maintenance teams is greater than or equal to 10. *Lead time* appears to have a relatively limited effect on the performance of *CCBM.* The performance remains steady when lead time is greater than or equal to three weeks and the number of maintenance teams available is greater than or equal to 10. However, a percentage decrease of about 37% in the average number of failures is observed when lead time is increased from one week to three weeks. A shorter lead time allows for wind turbines to be repaired and be available sooner. However, with shorter lead times turbines will be working for longer periods of time and are expected to fail more. Consequently, under *CCBM*, the average number of failures is larger under short lead times as compared to longer lead times.

The results show a percentage increase in *system availability* and *power generated* of about 3% when the number of maintenance teams is increased from five to 10. When compared to *SM, CCBM* provides an average of about 9.3% increase in power generation and a reduction in the number of failures of about 88%. In terms of cost, *CCBM* provides a decrease of about 65% when compared to *SM*. When compared to *CBM, CCBM* provides about 1% increase on average in power generation and a decrease of about 78% in the average *number of failures*. The results show that even though the number of failures is reduced, the power generation stays about the same compared to *CBM*. This result can be explained by the fact that both algorithms, *CBM* and *CCBM*, provide about the same level of system availability. *CCBM* reduces the number of failures by increasing the number of PMs by about 40% compared to *CBM.* The amount of time required to perform PM actions under *CCBM* balances the amount of time required to perform CM actions under *CBM*. In addition, *CCBM* provide savings of about 20% compared to *CBM*. Since PM actions are less expensive than CM actions, *CCBM* provides some cost savings.

Figures 8 and 9 compare the performance of the maintenance scheduling algorithms. Figure 8 compares the average wind farm power generated (columns) and average maintenance cost per turbine (line) when the gearbox replacement lead times are one, three, six and eight weeks, respectively. Figure 9 compares the average number of failures (columns) and average wind farm availability (line) for the same gearbox replacement lead times.

Figure 8 shows a similar pattern for the three scheduling maintenance strategies when one week, three weeks and six weeks are considered as replacement lead times for the gearbox component. *CBM* and *CCBM* perform similarly in terms of power generation and a rise in power generation is noticed when the maintenance capacity is increased from five teams to 10 teams. *CBM* and *CCBM* produce about 10% more power than *SM* across all experiments. However, it is important to notice that the performance of *SM* improves as the maintenance capacity increases. *CCBM* performs better than both *SM* and *CBM*

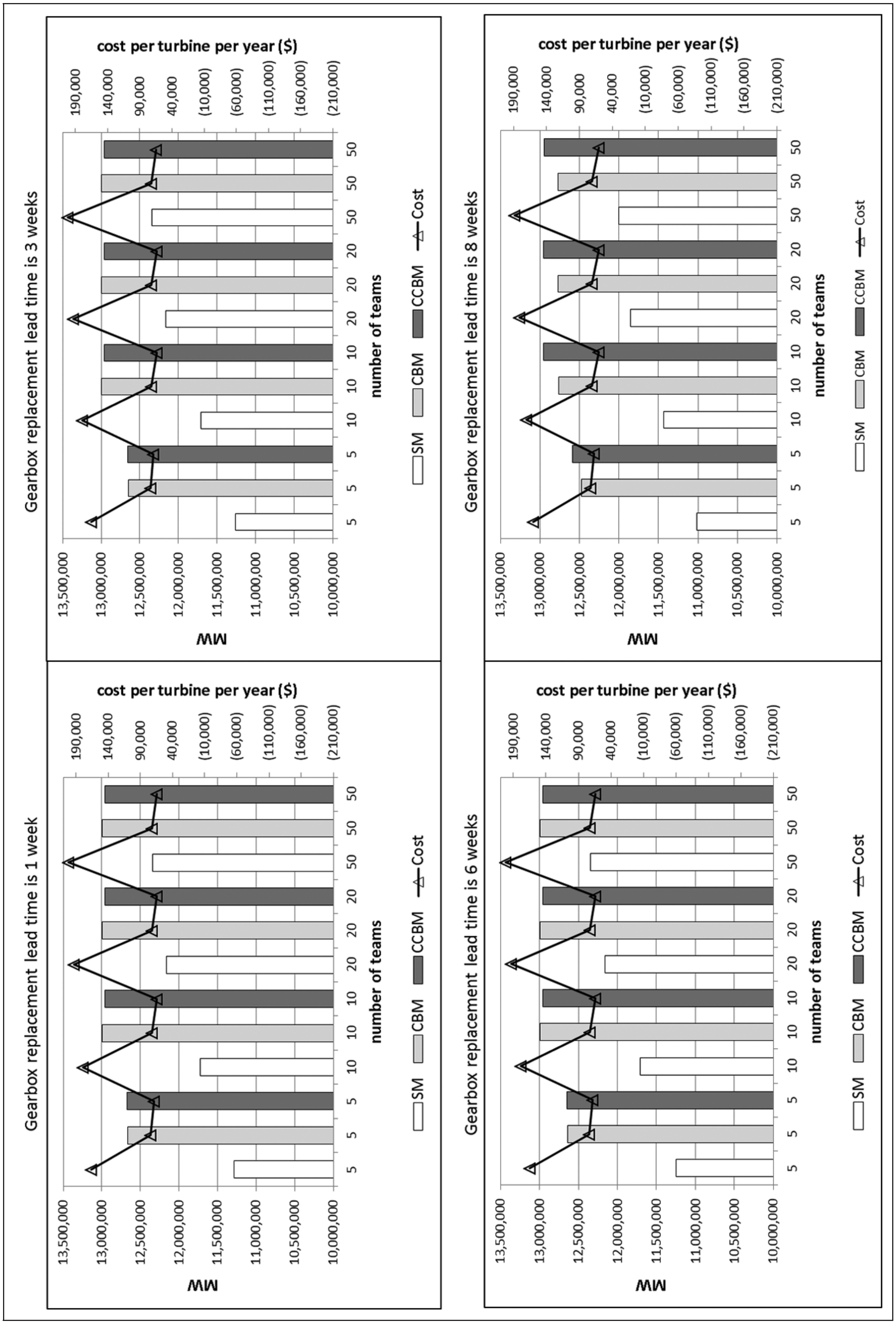

Figure 8. Average power generated and maintenance cost per turbine.

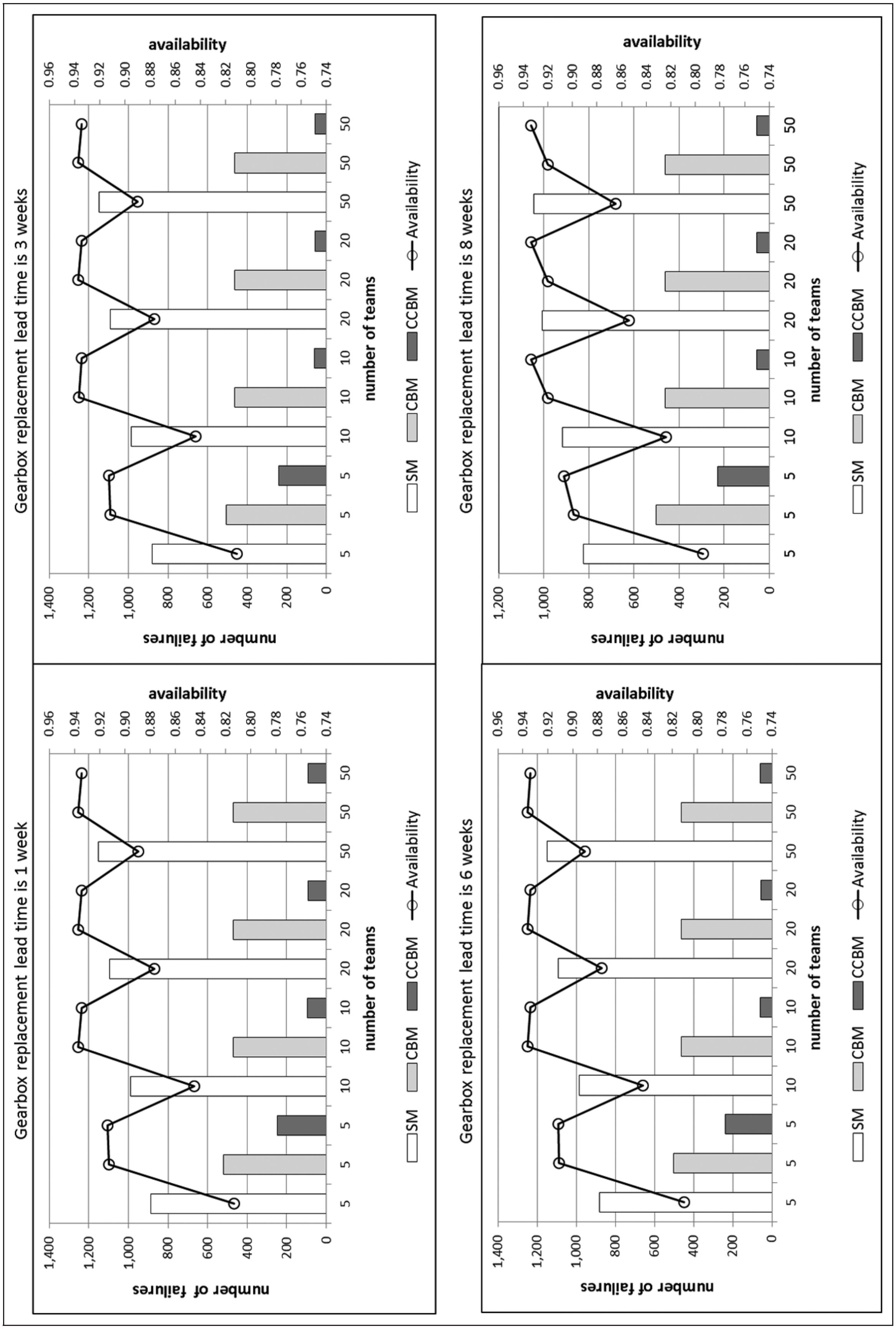

Figure 9. Average numbers of failures and wind farm availability.

Table 8. Computational results for concurrent condition-based monitoring (*CCBM*).

Maintenance capacity (# teams)	Replacement lead times (days)				Performance measures			
	Gearbox	Power generator	Blades	Control system	Avg. power generated (MW)	Avg. system availability (%)	Avg. number of failures	Avg. number of preventive actions
5	1	4	2	1	12,675,769.32	0.914	249.20	1763.60
5	3	4	2	1	12,654,316.88	0.912	240.80	1785.40
5	4	4	2	1	12,643,320.87	0.911	238.00	1799.80
5	8	4	2	1	12,587,059.60	0.907	228.80	1821.40
10	1	4	2	1	12,962,324.32	0.934	96.40	2199.00
10	3	4	2	1	12,961,324.67	0.934	59.80	2305.20
10	4	4	2	1	12,957,865.80	0.934	59.00	2293.00
10	8	4	2	1	12,950,755.32	0.933	56.00	2300.20
20	1	4	2	1	12,962,324.32	0.934	93.40	2203.00
20	3	4	2	1	12,961,324.67	0.934	59.00	2310.40
20	4	4	2	1	12,957,865.80	0.934	58.20	2295.20
20	8	4	2	1	12,950,755.32	0.933	54.80	2302.40
50	1	4	2	1	12,961,651.55	0.934	93.60	2203.00
50	3	4	2	1	12,960,773.53	0.934	58.60	2310.60
50	4	4	2	1	12,957,949.26	0.934	58.60	2302.40
50	8	4	2	1	12,949,933.84	0.933	55.40	2295.20

in terms of maintenance cost. *CCBM* provides a 230% and 65% percentage decrease in cost compared to *SM* and *CBM*, respectively. These savings were computed using the average maintenance cost per turbine per year under each maintenance strategy. The cost reduction can be explained by the fact that *CCBM* decreases the number of CM actions or component replacements required by performing preventive actions sooner. In addition, for the case in which a maintenance lead time of eight weeks is considered, *CCBM* performs better across all the experiments. In this particular case, there is a longer wait period for an important component and turbines requiring such a component will remain inactive for a longer period of time. Since *CCBM* decreases the number of failures when compared to *SM* and *CBM*, it is expected to show a higher power throughput because the system remains active for a relatively longer period of time.

Figure 9 also shows a similar pattern in terms of number of failures and availability for the three scheduling maintenance strategies when one, three and six weeks are considered for the replacement lead time for the gearbox component. *CBM* and *CCBM* perform similarly in terms of availability. However, *CCBM* shows an 88% and 78% reduction in terms of the number of failures when compared to *SM* and *CBM*, respectively. Although *CCBM* is able to decrease the average number of failures for the system, it is important to notice that system availability is about the same compared to *CBM*. This result can be explained by understanding that *CCBM* performs more PM actions than *CBM*, which balances the amount of time required to perform CM actions under *CBM*. For the case in which the replacement lead time is eight weeks, *CCBM* performs the best across all the experiments. Again, turbines will have to wait for a longer period for component replacement, and turbines requiring such components will remain inactive for a longer period of time. Since *CCBM* reduces the number of failures compared to *SM* and *CBM*, it is expected to yield a higher power throughput because the system remains active for a longer period of time.

6. Discussion and conclusions

One of the main factors for enhancing wind energy marketability is reducing O&M costs. Most of the O&M costs can be attributed to dispatching maintenance crews with heavy-duty equipment to remote wind farm sites. In this paper we derive new algorithms to improve scheduling strategies for O&M of wind farms. The algorithms are *SM, CBM*, and *CCBM*. The *CBM* and *CCBM* algorithms consider the degradation process of multiple components of a wind turbine to decide *when* and *what* type of maintenance to perform. We devise a DEVS simulation model of a wind farm with multi-component wind turbines to test and study the benefits of the proposed algorithms. The DEVS simulation model mimics the actual operation and degradation of multiple components of a wind turbine and allows for modeling maintenance scheduling as well as rescheduling if the weather is adverse. Our implementation of the wind farm simulation allows the user to select the factors, responses, and the scheduling algorithm to use in each simulation run.

The computational study demonstrates the importance of avoiding over-simplistic assumptions when making

O&M decisions for wind farms. For instance, in the existing literature lead times are usually modeled as fixed, while maintenance capacities are considered unlimited. The simulation results show that different levels of maintenance capacity and component replacement lead time have an impact on scheduling wind turbine maintenance, which in turn affects wind farm operations performance in terms of the number of *turbine failures, availability capacity* and *power generation*. For example, the best performance in terms of wind turbine availability and power generation for *SM* is observed when the maintenance capacity is at the maximum level (50 teams). SM performs poorly when maintenance capacity is limited. This is because with inadequate maintenance capacity many failed turbines do not receive maintenance for long periods of time.

The results also show that using condition monitoring maintenance strategies such as *CBM* and *CCBM* results in a reduction in O&M costs compared to *SM. CCBM* provides the lowest cost and minimum number of turbine failures. This can be explained by the fact the *CCBM* reduces the number of corrective actions (component replacements) by performing PM earlier than both *SM* and *CBM*. Recall that *CCBM* allows for maintenance to be performed on multiple components on the same maintenance visit to each turbine. However, the results show that *CBM* provides slightly higher availability and average power generation for most of the experiments compared to *CCBM*. This result is counterintuitive but can be explained by first understanding that PM takes less time than CM. Secondly, *CCBM* almost doubles the number of PM actions compared to *CBM* and, thus, more time is spent on performing maintenance.

As future work, we would like to consider the integration of optimization models in the maintenance decision process. We envision the development of an online framework that will allow for maintenance scheduling using optimization algorithms that take into account possible future turbine component failures and weather conditions. In addition, it would be interesting to incorporate the wake effect into the simulation model to study its impact on the scheduling algorithms. Finally, this study can be extended to a broad array of potential applications related to wind farm systems. For example, the algorithms can be extended to study offshore wind farms, which consider additional factors such as transportation of replacement components and maintenance crews using helicopters. Finally, this work can be extended and used in the evaluation of future wind farm construction sites and their integration into existing power systems.

Funding

This research received no specific grant from any funding agency in the public, commercial or not-for-profit sectors.

References

1. Lindenberg S, Smith B, O'Dell K, et al. *20% Wind Energy by 2030: increasing wind energy's contribution to U.S. electricity supply*. Springfield, VA: DIANE Publishing, 2009.
2. Walford CA. *Wind turbine reliability: understanding and minimizing wind turbine operation and maintenance costs*. Albuquerque, NM/Livermore, CA: United States Department of Energy, Sandia National Laboratories, 2006.
3. Zeigler BP, Praehofer H and Kim TG. *Theory of modeling and simulation: integrating discrete event and continuous complex dynamic systems*. Orlando, FL: Academic Press, Inc., 2000.
4. Byon E, Pérez E, Ding Y, et al. Simulation of wind farm operations and maintenance using discrete event system specification. *Simulation* 2011; 87: 1093–1117.
5. Pérez E, Ntaimo L, Byon E, et al. A stochastic DEVS wind turbine component model for wind farm simulation. In: *proceedings of the 2010 spring simulation multiconference*, Orlando, FL, 2010, pp.1–8.
6. Rademakers LWMM, Verbruggen TW and Braam H. Condition monitoring for lowering maintenance costs of offshore wind turbines. In: *proceedings of ISMA 2004: international conference on noise and vibration engineering*, Vols 1-8, 2005, pp.3943–3952.
7. Ribrant J. *Reliability performance and maintenance, a survey of failures in wind power systems*. Master's thesis. Stockholm, Sweden: KTH School of Electrical Engineering, 2006.
8. Vittal S and Teboul M. Performance and reliability analysis of wind turbines using Monte Carlo methods based on system transport theory. In: *46th AIAA structures, structural dynamics and materials conference*, 2005.
9. Van Bussel G. The development of an expert system for the determination of availability and O&M costs for offshore wind farms. In: *European wind energy conference and exhibition*, 1999, pp.402–405.
10. Hendriks H, Bulder B, Heijdra J, et al. DOWEC concept study; evaluation of wind turbine concepts for large scale offshore application. In: *proceedings of the European seminar offshore wind energy in Mediteranean and other European seas*, 2000, pp.211–219.
11. McMillan D and Ault GW. Condition monitoring benefit for onshore wind turbines: sensitivity to operational parameters. *Iet Renewable Power Generation* 2008; 2: 60–72.
12. Andrawus JA, Watson J and Kishk M. Modelling system failures to optimise wind turbine maintenance. *Wind Eng* 2007; 31: 503–522.
13. Hall P and Strutt J. Probabilistic physics-of-failure models for component reliabilities using Monte Carlo simulation and Weibull analysis: a parametric study. *Reliab Eng Syst Saf* 2003; 80: 233–242.
14. Castro R, Kofman E and Wainer G. A formal framework for stochastic discrete event system specification modeling and simulation. *Simulation* 2010; 86: 587–611.
15. Byon E, Ntaimo L and Ding Y. Optimal maintenance strategies for wind turbine systems under stochastic weather conditions. *IEEE Trans Reliab* 2010; 59: 393–404.
16. 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; 25: 1823–1834.

17. Byon E. Wind turbine operations and maintenance: a tractable approximation of dynamic decision-making. *IIE Trans* 2013; 45: 1188–1201.
18. 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 Convers* 2007; 22: 223–229.
19. Tian ZG, Jin TD, Wu BR, et al. Condition based maintenance optimization for wind power generation systems under continuous monitoring. *Renewable Energy* 2011; 36: 1502–1509.
20. Ding F and Tian Z. Opportunistic maintenance optimization for wind turbine systems considering imperfect maintenance actions. *Int J Reliab Qual Saf Eng* 2011; 18: 463.
21. Spinato F, Tavner P, Van Bussel G, et al. Reliability of wind turbine subassemblies. *IET Renew Power Gener* 2009; 3: 387–401.
22. Zeigler BP and Sarjoughian H. *Introduction to DEVS modeling & simulation with JAVATM: developing component-based simulation models*. Temple, AZ: Arizona State University, 2003.
23. Mesonet WT. http://www.mesonet.ttu.edu (2008). Accessed October, 9, 2013.
24. Weinzettel J, Reenaas M, Solli C, et al. Life cycle assessment of a floating offshore wind turbine. *Renew Energ* 2009; 34: 742–747.
25. Andrawus JA, Watson J, Kishk M, et al. The selection of a suitable maintenance strategy for wind turbines. *Wind Eng* 2006; 30: 471–486.

Author biographies

Dr Eduardo Pérez is an assistant professor at Texas State University, Ingram School of Engineering, San Marcos, Texas, USA. He obtained his PhD in systems and industrial engineering from the Texas A&M University and his BS in industrial engineering from the University of Puerto Rico, Mayaguez Campus. He is the director of the Integrated Modeling and Optimization for Service Systems (iMOSS) research laboratory. His research interests include stochastic optimization and discrete event simulation with application to healthcare systems, parallel machines and renewable energy.

Dr Lewis Ntaimo is an associate professor of industrial and systems engineering at Texas A&M University, College Station, Texas, USA, and was also a visiting professor at the University of Milan-Bicocca, Milan, Italy.

Dr Yu Ding is the Mike and Sugar Barnes Professor and associate department head for graduate affairs of industrial and systems engineering, a professor of electrical and computer engineering, and a faculty affiliate with Texas A&M Engineering Institute and the Institute of Applied Mathematics and Computational Sciences (IAMCS), all at Texas A&M University.

Appendix. Mathematical expressions of the discrete event system specification models

In this section we provide mathematical expressions of the atomic and coupled models that comprise the WTURBINE model using parallel DEVS.[3,22] We use set theory in the mathematical expressions that follow. The symbol $\wedge$ will denote logic AND, $\times$ will denote the Cartesian product and $R^{+}_{0,\infty}$ will denote the extended non-negative real line that includes infinity.

Atomic models

We define the following atomic models: UNIT, CMPDEG, MMNTOR, SENSOR and PWRPROD. In the UNIT atomic model a "*windOK*" Boolean variable is used to notify whether the wind speed is within a specified threshold (*true*) or not (*false*). A set of Boolean variables is used to describe the state of the wind turbine: "*normalstate*", "*alertstate*", "*alarmstate*" and "*failstate*", where a *true* value indicates the status of the turbine according the state name. A second set of Boolean variables describes the state transitions of the wind turbine: "*degType*1", "*degType*2", and "*degType*3". In this case a *true* value for "*degType*1" indicates that the component will transition to "*alertstate*"; a *true* value for "*degType*2" indicates that the component transition to the "*alarmstate*"; and a *true* for "*degType*3" indicates that the component will transition to "*failstate*". Finally, two Boolean variables, "*mainType*1" and "*mainType*2", are used to respectively indicate the type of maintenance performed to the turbine, preventive (PM) or corrective (CM).

The UNIT atomic model has a set of input ports $IPorts = \{\text{wind\_in}, \text{maint\_in}, \text{deg\_in}\}$, where $X_{wind\_in} = V_1, X_{maint\_in} = V_2$, and $X_{deg\_in} = V_3$ are arbitrary sets. The set of output ports are $OPorts = \{\text{unit\_on\_off}, \text{deg\_on\_off}\}$, where $Y_{unit\_on\_off}$ and $Y_{deg\_on\_off}$ are arbitrary sets. The UNIT atomic model can be defined using parallel DEVS as follows:

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

where

$X_M = \{(p, v)|p \in IPorts, v \in X_p\}$ is the set of input ports and values;
$Y_M = \{(p, v)|p \in OPorts, v \in Y_p\}$ is the set of output ports and values; and
$S = \{\text{off, normal, alert, alarm, fail, endOff, PM, CM, endPM, endCM}\} \times R^{+}_{0,\infty} \times V_1 \times V_2 \times V_3$ is the set of sequential states.

External transition function:

$$\delta_{ext}((phase, \sigma), e, (p, v))$$

$$= (\text{off}, \infty), \text{ if } \begin{cases} phase = \text{``off''} \wedge p = \text{``wind\_in''} \\ phase = \text{``normal''} \wedge p = \text{``wind\_in''} \wedge \text{``windOK''} = false \\ phase = \text{``alert''} \wedge p = \text{``wind\_in''} \wedge \text{``windOK''} = false \\ phase = \text{``alarm''} \wedge p = \text{``wind\_in''} \wedge \text{``windOK''} = false \\ phase = \text{``PM''} \wedge p = \text{``wind\_in''} \wedge \text{``windOK''} = false \\ phase = \text{``CM''} \wedge p = \text{``wind\_in''} \wedge \text{``windOK''} = false \end{cases}$$

$$= (\text{normal}, \infty), \text{ if } phase = \text{``off''} \wedge p = \text{``wind\_in''} \wedge \ normalstate = true \wedge \ windOK = true$$

$$= (\text{alert}, \infty), \text{ if } \begin{cases} phase = \text{``off''} \wedge p = \text{``wind\_in''} \wedge alertstate = true \wedge windOK = true \\ phase = \text{``normal''} \wedge p = \text{``deg\_in''} \wedge degType1 = true \end{cases}$$

$$= (\text{alarm}, \infty), \text{ if } \begin{cases} phase = \text{``off''} \wedge p = \text{``wind\_in''} \wedge alertstate = true \wedge windOK = true \\ phase = \text{``normal''} \wedge p = \text{``deg\_in''} \wedge degType2 = true \\ phase = \text{``alert''} \wedge p = \text{``deg\_in''} \wedge degType2 = true \end{cases}$$

$$= (\text{fail}, \infty), \text{ if } \begin{cases} phase = \text{``off''} \wedge p = \text{``wind\_in''} \wedge failstate = true \wedge windOK = true \\ phase = \text{``normal''} \wedge p = \text{``deg\_in''} \wedge degType3 = true \\ phase = \text{``alert''} \wedge p = \text{``deg\_in''} \wedge degType3 = true \\ phase = \text{``alarm''} \wedge p = \text{``deg\_in''} \wedge degType3 = true \end{cases}$$

$$= (\text{PM}, \infty), \text{ if } \begin{cases} phase = \text{``off''} \wedge p = \text{``maint\_in''} \wedge maintType1 = true \\ phase = \text{``normal''} \wedge p = \text{``maint\_in''} \wedge maintType1 = true \\ phase = \text{``alert''} \wedge p = \text{``maint\_in''} \wedge maintType1 = true \\ phase = \text{``alarm''} \wedge p = \text{``maint\_in''} \wedge maintType1 = true \end{cases}$$

$$= (\text{CM}, \infty), \text{ if } \begin{cases} phase = \text{``off''} \wedge p = \text{``maint\_in''} \wedge maintType2 = true \\ phase = \text{``normal''} \wedge p = \text{``maint\_in''} \wedge maintType2 = true \\ phase = \text{``alert''} \wedge p = \text{``maint\_in''} \wedge maintType2 = true \\ phase = \text{``alarm''} \wedge p = \text{``maint\_in''} \wedge maintType2 = true \\ phase = \text{``fail''} \wedge p = \text{``maint\_in''} \wedge maintType2 = true \end{cases}$$

$$= (\text{endPM}, \infty), \text{ if } phase = \text{``PM''} \wedge p = \text{``maint\_in''} \wedge maintType0 = true$$
$$= (\text{endCM}, \infty), \text{ if } phase = \text{``CM''} \wedge p = \text{``maint\_in''} \wedge maintType0 = true.$$

Internal transition function:

$$\delta_{int}(phase, \sigma)$$
$$= (\text{normal}, \infty), \text{ if } \begin{cases} phase = \text{``endPM''} \ \wedge \ maintType1 = true \ \wedge \ windOK = true \\ phase = \text{``endCM''} \ \wedge \ maintType2 = true \ \wedge \ windOK = true \end{cases}$$

$$= (\text{alert}, \infty), \text{ if } phase = \text{``endPM''} \ \wedge \ maintType1 = true \ \wedge \ windOK = true$$

$$= (\text{off}, \infty), \text{ if } \begin{cases} phase = \text{``endPM''} \ \wedge \ maintType1 = true \ \wedge \ windOK = false \\ phase = \text{``endCM''} \ \wedge \ maintType2 = true \ \wedge \ windOK = false \end{cases}$$

***Confluence function*:**

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

***Output function*:**

$$\lambda(phase, \sigma) = (\text{unit\_on\_off}, onoffPWRGEN)\text{if} \begin{cases} phase = \text{``endCM''} \\ phase = \text{``endPM''} \end{cases}$$

$$= (\text{deg\_on\_off}, onoffCMPDEG)\text{if} \begin{cases} phase = \text{``endCM''} \\ phase = \text{``endPM''}. \end{cases}$$

***Time advance function*:**

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

In the CMPDEG atomic model the Boolean variables are used to describe the actions that will affect the degradation of a wind turbine component and are ''*operation*1'', ''*operation*2'', ''*operation*3'' and ''*operation*4''. A value of *true* for each of the variables indicates the following: PM is completed, PM is completed but wind speed is out of the threshold, CM is completed, CM is completed but wind speed is out of the threshold, respectively. A parameter STI (short time interval) is used in the internal transition function. The level of *degradation* is output as a message from the model and describes the current state of the wind turbine component. The CMPDEG atomic models has the input set $IPorts = \{\text{cmpdeg\_on\_off}\}$, where $X_{cmpdeg\_on\_off} = V_1$ is an arbitrary set. The output set $OPorts = \{\text{deg\_out}\}$, where Y_{deg_out} is an arbitrary set. This atomic model can be expressed in parallel DEVS as follows:

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

where

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

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

$S = \{\text{passive, active}\} \times R^+_{0,\infty} \times V_1$ is the set of sequential states.

***External transition function*:**

$$\delta_{ext}((phase, \sigma), e, (p, v))$$

$$= (\text{active}, \infty), \text{ if} \begin{cases} phase = \text{``passive''} \wedge p = \text{``cmpdeg\_on\_off''} \wedge operation1 = true \\ phase = \text{``active''} \wedge p = \text{``cmpdeg\_on\_off''} \wedge operation1 = true \end{cases}$$

$$= (\text{passive}, \infty), \text{ if} \begin{cases} phase = \text{``passive''} \wedge p = \text{``cmpdeg\_on\_off''} \wedge operation2 = true \\ phase = \text{``active''} \wedge p = \text{``cmpdeg\_on\_off''} \wedge operation2 = true \end{cases}$$

$$= (\text{restart\_active}, \infty), \text{ if} \begin{cases} phase = \text{``passive''} \wedge p = \text{``cmpdeg\_on\_off''} \wedge operation3 = true \\ phase = \text{``active''} \wedge p = \text{``cmpdeg\_on\_off''} \wedge operation3 = true \end{cases}$$

$$= (\text{restart\_passive}, \infty), \text{ if} \begin{cases} phase = \text{``passive''} \wedge p = \text{``cmpdeg\_on\_off''} \wedge operation4 = true \\ phase = \text{``active''} \wedge p = \text{``cmpdeg\_on\_off''} \wedge operation4 = true. \end{cases}$$

***Internal transition function*:**

$$\delta_{int}(phase, \sigma) = (\text{active}, STI), \text{ if} \begin{cases} phase = \text{``active''} \\ phase = \text{``restart\_active''} \end{cases}$$

$$= (\text{passive}, \infty), \text{ if} \begin{cases} phase = \text{``passive''} \\ phase = \text{``restart\_active''} \end{cases}$$

***Confluence function*:**

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

***Output function*:**

$$\lambda(phase, \sigma)$$

$$= (\text{deg\_out}, degradation)\text{if } phase = \text{``active''}$$

$$= (\text{deg\_out}, degradation)\text{if } phase = \text{``restart\_active''}.$$

Time advance function:

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

The MMNTR atomic model has a Boolean variable "*windOK*", which is used to indicate whether the wind speed is within a specified threshold (*true*) or not (*false*). Two types of output messages are created by this model: *timeleft* and *maintType*. In case maintenance needs to be rescheduled, the message *timeleft* contains the amount of time needed to complete the maintenance. The message *maintType* contains the type of maintenance to be performed in the turbine, PM or CM. MMNTR has the input set $IPorts = \{\mathrm{wind\_in}, \mathrm{turb\_mnt}\}$, where $X_{wind\_in} = V_1$ and $X_{turb\_mnt} = V_2$ are arbitrary sets. The output set $OPorts = \{reSched, \mathrm{maint}\}$, where $Y_{reSched}$ and Y_{maint} are arbitrary sets. This atomic model can be expressed in parallel DEVS as follows:

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

where
$X_M = \{(p, v)|p \in IPorts, v \in X_p\}$ is the set of input ports and values;
$Y_M = \{(p, v)|p \in OPorts, v \in Y_p\}$ is the set of output ports and values; and
$S = \{\mathrm{mtOff}, \mathrm{mtOn}, \mathrm{mtOver}\} \times R^{+}_{0,\infty} \times V_1 \times V_2$ is the set of sequential states.

External transition function:

$\delta_{ext}((phase, \sigma), e, (p, v))$
$= (\mathrm{mtOn}, \infty)$, if $phase =$ "mtOff" $\wedge\, p =$ "wind_in" $\wedge\, windOK = true$
$= (\mathrm{mtOff}, \infty)$, if $phase =$ "mtOn" $\wedge\, p =$ "wind_in" $\wedge\, windOK = false$
$= (\mathrm{mtOver}, \infty)$, if $phase =$ "mtOver" $\wedge\, p =$ "turb_mnt".

Internal transition function:

$\delta_{int}(phase, \sigma).$
$= (\mathrm{mtOver}, \infty)$, if $phase =$ "mtOff"
$= (\mathrm{mtOn}, \infty)$, if $phase =$ "mtOver"
$= (\mathrm{mtOver}, \infty)$, if $phase =$ "mtOn".

Confluence function:

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

Output function:

$\lambda(phase, \sigma)$
$= (\mathrm{reSched}, timeleft)$if $phase =$ "mtOn"
$= (\mathrm{maint}, mainType)$if $phase =$ "mtOver".

Time advance function:

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

The SENSOR atomic model generates a message of type *cmpstate*, which indicates the current state of the component. The input set for this model $IPorts = \{\mathrm{status\_in}\}$, where $X_{status\_in} = V_1$ are arbitrary sets. The output set $OPorts = \{\mathrm{status\_out}\}$, where Y_{status_out} is an arbitrary set. The SENSOR atomic model can be defined as follows:

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

where
$X_M = \{(p, v)|p \in IPorts, v \in X_p\}$ is the set of input ports and values;
$Y_M = \{(p, v)|p \in OPorts, v \in Y_p\}$ is the set of output ports and values; and
$S = \{\mathrm{passive}, \mathrm{active}\} \times R^{+}_{0,\infty} \times V_1$ is the set of sequential states.

External transition function:

$\delta_{ext}((phase, \sigma), e, (p, v))$
$= (\mathrm{passive}, \infty)$, if $phase =$ "active" $\wedge\, p =$ "status_in".

Internal transition function:

$\delta_{int}(phase, \sigma).$
$= (\mathrm{active}, \infty)$, if $phase =$ "passive".

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

***Output function*:**
$\lambda(phase, \sigma)$
$= (\text{status\_out}, cmpstate)\text{if } phase = \text{``active''}$.

***Time advance function*:**

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

The PWRPROD model uses two Boolean variables: ''*windOK*'' indicates whether the wind speed is within a specified threshold (*true*) or not (*false*) and ''*unitOK*'' indicates the current status of the UNIT atomic model, whether it is operating (*true*) or not (*false*). Three types of output messages are created by this model: *statuson, statusoff* and *power*. The first two messages report the status of the turbine and the third reports the power generated by the wind turbine in a given time period. The PWRPROD atomic model has input set $IPorts = \{\text{cmp\_on\_off}, \text{wind\_in}, \text{turb\_on\_off}\}$, where $X_{cmp\_on\_off} = V_1, X_{wind\_in} = V_2$ and $X_{turb\_on\_off} = V_3$ are arbitrary sets. The output set $OPorts = \{\text{pwr\_status}, \text{pwr\_gen}\}$, where Y_{pwr_status} and Y_{pwr_gen} are arbitrary sets. This atomic model can be defined as followed:

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

where
$X_M = \{(p, v) | p \in IPorts, v \in X_p\}$ is the set of input ports and values;
$Y_M = \{(p, v) | p \in OPorts, v \in Y_p\}$ is the set of output ports and values; and
$S = \{\text{active}, \text{cmpfail}, \text{passive}\} \times R^+_{0,\infty} \times V_1 \times V_2 \times V_3$ is the set of sequential states.

***External transition function*:**
$\delta_{ext}((phase, \sigma), e, (p, v))$
$= (\text{cmpfail}, \infty), \text{ if } phase = \text{``active''} \wedge p = \text{``cmp\_on\_off''} \wedge\ unitOK = false$
$= (\text{active}, \infty), \text{ if } phase = \text{``active''} \wedge p = \text{``wind\_in''} \wedge windOK = true$
$= (\text{passive}, \infty), \text{ if } phase = \text{``active''} \wedge p = \text{``turb\_on\_off''}\ windOK = true$.

***Internal transition function*:**
$\delta_{int}(phase, \sigma)$
$= (\text{off}, \infty), \text{ if } phase = \text{``cmpfail''}$
$= (\text{active}, \infty), \text{ if } phase = \text{``passive''}$

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

***Output function*:**
$\lambda(phase, \sigma)$
$= (\text{``pwr\_status''}, statusoff), \text{ if } phase = \text{``cmpfail''}$
$= (\text{``pwr\_status''}, statuson), \text{ if } phase = \text{``passive''}$
$= (\text{``pwr\_gen''}, power), \text{ if } phase = \text{``cmpfail''}$.

***Time advance function*:**
$ta(phase, \sigma) = \sigma$.

The DEVS mathematical expressions of rest of the atomic models WGENR, TRANSD, MSCHEDR and MGENR can be given in a similar manner but are omitted due to space restrictions. Next we provide mathematical expressions for the coupled models that make up WTURBINE.

Coupled models

Given the above specifications of the atomic DEVS models (components), we are now in a position to mathematically define the DEVS coupled models CMP and WTURBINE. This will involve, for each coupled model, specifying the set of components D where for each $d \in D$,

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

is a DEVS atomic model. The coupling requirements for a coupled model N are as follows.

- External input couplings connect external inputs to components inputs: $EIC \subseteq \{((N, ip_N), (d, ip_d)) | ip_N \in \text{IPorts}, d \in D, ip_d \in IPorts_d\}$.
- External output couplings connect components outputs to external outputs: $EOC \subseteq \{((d, op_d), (N, op_N)) | op_N \in \text{OPorts}, d \in D, op_d \in OPorts_d\}$.
- Internal coupling couplings connect outputs to components inputs: $IC \subseteq \{((a, op_a), (b, ip_b)) | a, b \in D, op_a \in OPorts_a, ip_b \in IPorts_b\}$.

Finally, the coupled DEVS model specification is as follows:

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

where X is the set of input ports and values, and Y is the set of output ports and values for the coupled model.

The coupled model CMP represents a wind turbine component and comprises four atomic models: CMPDEG, SENSR, MMTR and UNIT. Its couplings are shown in Figure 1. This coupled model can be defined expressed in DEVS as follows:

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

where
$X = \{(p, v) | p \in IPorts, v \in X_p\}$ is the set of input ports and values;
$Y = \{(p, v) | p \in OPorts, v \in Y_p\}$ is the set of output ports and values;
$IPorts = \{\text{turb\_mnt}, \text{wind\_in}\}$, where X_{turb_mnt} and X_{wind_in} are arbitrary sets;
$OPorts = \{\text{status\_out}, \ \text{deg\_on\_off}, \ \text{unit\_on\_off}, \ \text{reSched}\}$, where Y_{status_out}, $Y_{\deg\_on\_off}$, $Y_{unit\_on\_off}$, $Y_{reSched}$ are arbitrary sets; and
$D = \{d_1, d_2, d_3, d_4\}$, where for each $d \in D$ the atomic DEVS models M_{d1}, M_{d2}, M_{d3} and M_{d4} are CMPDEG, SENSR, MMTR and UNIT, respectively.

The couplings for CMP are defined as follows:

$$EIC = \{((\text{CMP}, \text{turb\_mnt}), (\text{MMTR}, \text{turb\_mnt})), ((\text{CMP}, \text{wind\_in}), (\text{MMTR}, \text{wind\_in})), ((\text{CMP}, \text{wind\_in}), (\text{UNIT}, \text{wind\_in}))\}.$$

$$EOC = \{((\text{SENSR}, \text{status\_out}), (\text{CMP}, \text{status\_out})), ((\text{UNIT}, \text{deg\_on\_off}), (\text{CMP}, \text{deg\_on\_off})), ((\text{UNIT}, \text{unit\_on\_off}), (\text{CMP}, \text{unit\_on\_off})), ((\text{MMTR}, \text{reSched}), (\text{CMP}, \text{reSched}))\}.$$

$$IC = \{((\text{CMPDEG}, \text{deg\_out}), (\text{SENSR}, \text{status\_in})), ((\text{CMPDEG}, \text{deg\_out}), (\text{UNIT}, \text{deg\_in})), ((\text{MMTR}, \text{maint}), (\text{UNIT}, \text{maint})), ((\text{UNIT}, \text{deg\_on\_off}), (\text{CMPDEG}, \text{deg\_on\_off}))\}.$$

The wind turbine model WTURBINE comprises m CMP coupled DEVS models and one PWRPROD atomic model, as shown in Figure 2. This coupled model can expressed mathematically as follows:

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

where
$X = \{(p, v) | p \in IPorts, v \in X_p\}$ is the set of input ports and values;
$Y = \{(p, v) | p \in OPorts, v \in Y_p\}$ is the set of output ports and values;
$IPorts = \{\text{turb\_mnt}, \ldots, \text{wind\_in}\}$, where X_{turb_mnt} and X_{wind_in} are arbitrary sets (the same as in CMP);
$OPorts = \{\text{status\_out\_1}, \ldots, \text{status\_out\_m}, \text{reSched}, \text{pwr\_status}, \text{pwr\_gen}\}$,
where$Y_{status_out_1}, \ldots, Y_{status_out_m}, Y_{reSched}, Y_{pwr_status}, Y_{pwr_gen}$ are arbitrary sets; and
$D = \{d_1, \ldots, d_m, d_{m+1}\}$, where for each $d \in D$ the DEVS models $M_{d1}, \ldots, M_{dm}$ are the CMP coupled models labeled CMP(1), …, CMP(m) and $M_{dm+1}$is the PWRPROD atomic model.

The couplings for WTURBINE are defined as follows:

$$
\begin{aligned}
EIC = \{&((\text{WTURBINE}, \text{turb\_mnt}), (\text{CMP}(1), \text{turb\_mnt})),\\
&\ldots,\\
&((\text{WTURBINE}, \text{turb\_mnt}), (\text{CMP}(m), \text{turb\_mnt})),\\
&((\text{WTURBINE}, \text{wind\_in}), (\text{CMP}(1), \text{wind\_in})),\\
&\ldots,\\
&((\text{WTURBINE}, \text{wind\_in}), (\text{CMP}(m), \text{wind\_in})),\\
&((\text{WTURBINE}, \text{wind\_in}), (\text{PWRPROD}, \text{wind\_in}))\}.\\
EOC = \{&((\text{CMP}(1), \text{status\_out}), (\text{WTURBINE}, \text{status\_out\_1})),\\
&\ldots,\\
&((\text{CMP}(m), \text{status\_out}), (\text{WTURBINE}, \text{status\_out\_m})),\\
&((\text{CMP}(1), \text{reSched}), (\text{WTURBINE}, \text{reSched})),\\
&\ldots,\\
&((\text{CMP}(m), \text{reSched}), (\text{WTURBINE}, \text{reSched})),\\
&((\text{PWRPROD}, \text{pwr\_status}), (\text{WTURBINE}, \text{pwr\_status})),\\
&((\text{PWRPROD}, \text{pwr\_gen}), (\text{WTURBINE}, \text{pwr\_gen}))\}.\\
IC = \{&((\text{CMP}(1), \text{deg\_on\_off}), (\text{PWRPROD}, \text{deg\_on\_off})),\\
&\ldots,\\
&((\text{CMP}(m), \text{deg\_on\_off}), (\text{PWRPROD}, \text{deg\_on\_off})),\\
&\ldots,,\\
&((\text{CMP}(1), \text{turb\_on\_off}), (\text{PWRPROD}, \text{unit\_on\_off}))\\
&\ldots,\\
&((\text{CMP}(m), \text{turb\_on\_off}), (\text{PWRPROD}, \text{unit\_on\_off}))\}.
\end{aligned}
$$

The DEVS mathematical expressions of rest of the coupled models WF, OPMNT and EF are omitted due to limited space.