SIAM/ASA J. Uncertainty Quantification
Vol. 6, No. 2, pp. 596–626

Scenario Generation Methods that Replicate Crossing Times in Spatially Distributed Stochastic Systems*

Joseph Durante†, Raj Patel‡, and Warren B. Powell‡

Abstract. The purpose of this paper is to bring to light the importance of time series simulation methods which accurately replicate the *crossing times* of stochastic processes. A crossing time is a contiguous block of time for which a stochastic process is above or below some benchmark such as a forecast. In addition to bringing attention to the issue, we present a family of models, which we call *crossing state models* (both univariate and multivariate models are introduced), that outperform standard time series modeling techniques in their ability to replicate these crossing times. This is verified using a weighted quadratic empirical distribution function statistic. In addition, in multivariate processes (which may be spatially distributed) we address the problem of replicating crossing times at both the disaggregate and aggregate levels. Proper modeling of crossing times is especially significant in applications in the realm of energy systems. For example, a robust control policy for an energy system with high penetrations of renewables must account for the possibility that energy from wind falls below forecasts for a long period of time. A policy that performs well on a set of renewable power scenarios in which the crossing times are accurately modeled will likely be robust in practice as well. Modeling crossing time behavior is pertinent in other problems involving stochastic optimization as well, such as portfolio management and inventory management problems.

Key words. crossing times, scenario generation, time series models, multivariate processes

AMS subject classifications. 62M30, 62M10, 62P30, 60K30

DOI. 10.1137/17M1120555

1. Introduction.

When developing policies for controlling a system under uncertainty, it is essential they perform well across a realistic population of scenarios. In the realm of energy systems, uncertainty in power generation from renewable power sources, such as wind, creates a need for simulation methods which accurately replicate the characteristics of power outputs from renewable sources. For example, the stochastic programming formulation of the day-ahead unit commitment problem with high penetrations of wind energy will require a set of wind power scenarios [39, 11].

Often available to the system operator is a forecast of wind power outputs (or a wind speed forecast combined with a method for producing a corresponding power forecast) over the planning horizon. In general, the power output series tend to follow their forecasted outputs, plus or minus some error. For energy systems modeling, it is particularly important to replicate not just the distribution of errors when the actual power output is below the

*Received by the editors March 13, 2017; accepted for publication (in revised form) March 12, 2018; published electronically May 1, 2018.
http://www.siam.org/journals/juq/6-2/M112055.html
Funding: The research of the authors was supported in part by NSF grant CCF-1521675.
†Department of Electrical Engineering, Princeton University, Princeton, NJ 08544-5263 (jdurante@princeton.edu).
‡Department of Operations Research and Financial Engineering, Princeton University, Princeton, NJ 08544 (rmpatel@princeton.edu, powell@princeton.edu).

forecast, but how long it stays below (and by how much) so we can properly plan backup storage or generation.

Other time series models can produce scenarios that replicate certain characteristics of observed forecast error series, such as the autocorrelation, partial autocorrelation, cross correlations (if applicable), and the distribution of errors. However, we have observed that most time series models often fail to capture one important, yet overlooked, characteristic of stochastic processes involved in sequential decision making problems—the *crossing times* of the process. A crossing time is a consecutive period of time for which the observed value of a stochastic process is above or below a benchmark reference series such as a forecast. It is related to a zero crossing interval as described in [6] except that it is relative to the reference series. Alternatively, a crossing time can be viewed as a zero crossing interval of the series of errors-from-benchmark.

There is a growing literature on scenario generation, with special interest in modeling wind for the stochastic unit commitment problem [30, 15, 20]. Methods have been developed for ensuring that sampled scenarios are representative (in particular, not too similar), but this work has not attempted to model crossing time behavior.

In this problem, to schedule generation a day in advance we can utilize algorithms which rely on the generation of wind power scenarios such as stochastic dual decomposition procedure (SDDP) [32, 36, 33, 35] or scenario trees [17, 25, 23, 5]. Consider using a wind power model that generates scenarios in which crossing times, and specifically the down-crossing times, are too short compared to reality. This is the case if, for example, we assume intertemporal independence of the forecast errors (a fairly standard assumption in the SDDP literature). Without having to plan for the possibility of wind power dropping below its forecast for extended periods of time, the policy chosen might be less costly in expectation, but the risk of a load-shedding event will be higher. Thus, we see that modeling crossing time behavior correctly is instrumental in the development of robust policies.

Applications involving a single stochastic process, such as a simple energy storage problem in which there is one exogenous renewable power source, will only require univariate models that reproduce crossing time and error distributions in one dimension. In spatially distributed applications (and multivariate processes in general) it is desirable to produce scenarios in which crossing time and error distributions are replicated at both the disaggregate and the aggregate level. This is, by itself, a new challenge addressed in this paper that has not been addressed in the statistics literature.

To see why crossing time behavior is important at both the aggregate and the disaggregate level, consider modeling a power grid with high penetrations of renewables. The aggregate power produced in the system must match the total load; however, the transmission of power across long distances will incur resistive losses and introduce transmission constraints, such as maximum power constraints on certain transmission lines, that limit the amount of power that can be sent from one area of the grid to another. Thus, our simulation method must produce scenarios with realistic surpluses and deficits of locally available renewable energy compared to the forecasted outputs as well as in aggregate.

This paper presents a family of time series simulation methods, which we call *crossing state models*, that focus on replicating forecast error distributions, as well as up- and down- crossing distributions. In addition, the models for multivariate time series replicate these distributions

not only at the disaggregate (individual) level but on the aggregate level as well. To be clear, this is not a forecasting paper and is not an attempt to create improved forecasting models to reduce forecast errors. Rather, we present simulation methods which, given any point forecast of an exogenous stochastic process that is available to the controller of a system, whether it is based on classical time series models or state-of-the-art methods, will, without the need for additional explanatory variables, generate sample paths in which the crossing times are consistent with training data.

Though the main application considered in this paper is wind power scenario generation, the models presented here can be used in many applications in which crossing times are important, such as the following:

- Modeling electricity prices, including the replication of price spikes, so that utilities can prepare financially for periods of time when prices exceed the rates they are being paid for the cost of electricity. A useful benchmark series here may be a time-dependent mean price that exhibits daily seasonality.
- Modeling the demand for units of blood, where weather can create bursts of accidents for which we have to have sufficient inventories.
- Internet retailers have to plan inventories to anticipate bursts of demand for a product that suddenly becomes popular.
- Developing a portfolio management policy for buying and selling stocks and derivatives. Risk averse policies must account for situations in which correlated stocks underperform expectations.

Crossing state models share a common form in that they all employ two-level strategies. The first level controls the evolution of what we call the *crossing state* in this paper which is designed to control the crossing times of our sample paths. The second level is an error generation model conditioned on the crossing state that is chosen such that it is appropriate for the stochastic process we are modeling. One indicator that we have chosen a good error generation model is that the error distribution of the simulations closely matches the empirical distribution from the training data. The crossing state models can be supplemented with additional explanatory variables (in the above examples, time of day or weather conditions may be relevant explanatory variables), but it is not a necessity. The multivariate models can be used to model spatially distributed stochastic processes, as is done in this paper, though they can model general multivariate processes without a spatial dimension as well.

This paper makes the following contributions: (1) We develop a hidden semi-Markov model for general univariate processes. (2) We extend this method to multivariate stochastic processes with the goal of replicating crossing time distributions at both the disaggregate and aggregate levels. (3) We show empirically that our methods closely replicate crossing time behaviors for both univariate and multivariate settings and that our methods outperform standard time series modeling techniques.

The rest of the paper is organized as follows. Section 2 reviews standard time series methods for modeling forecast errors as well as recent efforts in modeling spatially distributed wind power forecast errors. In section 3 we formally describe crossing times, crossing time distributions, and the crossing state of the process. Section 4 presents a univariate nonparametric hidden semi-Markov crossing state model. Sections 5 and 6 both present multivariate crossing state models. The low dimensional model in section 5 is a regime switching vector

autoregressive model that performs well in low dimensions but will not scale to more than four dimensions. The high dimensional model presented in section 6 is an approximation of the low dimensional model that will scale to many dimensions, but it will perform slightly worse in low dimensions. Finally, numerical results are presented in section 7 and the paper is concluded in section 8.

2. Review of scenario generation methods and modeling techniques. For a stochastic process with J dimensions (for example, the energy generated from J locations), let the forecast for subprocess j at lead time t be $f_{t,j}^P$. The error, given the actual realization of $P_{t,j}$, is given by $X_{t,j} = P_{t,j} - f_{t,j}^P$. In addition, by summing the J forecasts together we obtain an aggregate forecast at time t, $f_t^{P,agg} = \sum_{j=1}^J f_{t,j}^P$. Similarly, given a realization of vector P_t, we obtain an aggregate error $X_t^{agg} = \sum_{j=1}^J P_{t,j} - f_t^{P,agg} = \sum_{j=1}^J X_{t,j}$. Note that in the univariate case we drop the subprocess j subscript as the error vector at time t becomes the scalar X_t. In this section we review methods to generate scenarios of these forecast errors (or actual outputs directly) and modeling techniques.

2.1. Z-variate transforms. From training data we can form marginal error distributions for each subprocess j, represented by F_j^X. These empirical marginal error distributions may be nonnormal, heavy tailed, or even skewed or shifted in one direction depending on the properties of the error process and/or quality of the provided forecast. Likewise, the empirical aggregate error distribution is represented by $F^{X,agg}$ and may exhibit many of the same traits as the marginal distributions.

One technique to produce scenarios with arbitrary marginal distributions is to first perform a transformation of the time series to Z-variates such that the transformed variables have standard normal marginal distributions $\mathcal{N}(0,1)$s. This is done utilizing empirical marginal cumulative distribution functions (CDFs) and the standard normal quantile function. Then, a model can be fit to the transformed time series of Z-variates. Following the simulation of a Z-variate time series, the inverse transform is performed to recover the actual sample paths. This Z-transform technique is described in [10]. Formally, letting Φ be the standard normal CDF, $Z_{t,j}$, the Z-transform of $X_{t,j}$, is given by

$$Z_{t,j} = \Phi^{-1}\left(F_j^X\left(X_{t,j}\right)\right).$$

Note that $Z_j \sim \mathcal{N}(0,1)$. The inverse transform is then given by

$$X_{t,j} = F_j^{X,-1}\left(\Phi\left(Z_{t,j}\right)\right).$$

This transformation is not necessary when forming a scenario generation model, but it is widely used in wind power scenario generation methods including many referenced later in this section and some of the methods presented in this paper. To avoid repeating this throughout the paper, *if a model utilizes Z-transforms, it is assumed that the transforms took place prior to modeling, and, following simulation, the appropriate inverse transform is performed to recover the actual sample paths.*

We can also perform the same procedure using conditional distributions. Consider an explanatory variable Y and conditional marginal CDF's $F_j^{X|Y}$. If, for example, Y is discrete

we can perform a similar transform with the above formulas by simply replacing F_j^X with $F_j^{X|Y=y}$ for different values that Y may take on. We will denote a conditional Z-transformed time series with $Z_{t,j}^Y$. Scenario generation in [27] uses Z-transforms conditioned on the forecast bin to which the point forecast for subprocess j at lead time t belongs. Thus, the explanatory variable Y here would represent the forecast bin to which $f_{t,j}^P$ belongs. The multivariate crossing state models in sections 5 and 6 also utilize this conditional form of the Z-transform.

2.2. Classical time series models: (V)ARIMA and GARCH. For a univariate time series, whether it is a Z-variate series or not, a common approach is to model the series as an autoregressive integrated moving average process of order p,d,q (an ARIMA(p,d,q) process). As an initial modeling step, differencing can be used to form a stationary time series out of a nonstationary one. A first order differencing forms the series $X_t' = X_t - X_{t-1}$, while a second order differencing forms the series $X_t'' = X_t' - X_{t-1}'$. This pattern is repeated up until order d. Consider $\{X_t\}$ the series we are working with, differenced or not. It can be modeled as the ARMA(p,q) process,

$$X_t = c + \sum_{i=1}^{p} A_i X_{t-i} + W_t + \sum_{k=1}^{q} B_k W_{t-k},$$

where c, the $A_i's$, and the $B_k's$, are scalar coefficients and W_t is a zero mean white noise (such as $W_t \sim \mathcal{N}(0, \sigma^2)$).

Determining the order of the model (the p, d, and q) as well as the parameters associated with the chosen model can be done with the Box and Jenkins method of model identification, parameter estimation, and model checking [8]. To assist the Box and Jenkins method, or as an alternative, minimizing the Akaike information criteria (AIC) can be used to determine the order of the model. The AIC method selects the model M with the minimum AIC value which is given by $\text{AIC} = 2k_f - 2ln(p(x|\hat{\theta}, M))$, where x is the data, k_f denotes the number of free parameters in the model, and $\hat{\theta}$ is set of parameter values for model M which maximizes the likelihood function [1]. As a general rule, if the AIC values of two models are close, it is better to choose the lower order model as it is more parsimonious.

ARMA (autoregressive moving average) models are often used to generate wind power scenarios as in [37], for example. Another common method is to model wind speed forecast error time series with an ARMA model [38]. To transform wind speed scenarios to a wind power series, a wind speed-to-power curve can be used as in [4], [31], and [3]. In general, when wind speed scenarios are generated instead of wind power, some type of power curve is used to transform the series to a wind power series. However, the methods in this paper directly generate wind power sample paths.

Extending ARMA models to produce spatially distributed scenarios can be accomplished by incorporating correlation matrices. For example, in [3], the sample vector of errors at each time t is drawn from a multivariate normal distribution $\mathcal{N}(Z_t, \Sigma)$. The time t mean for each wind farm's Z-transformed forecast error, $Z_{t,j}$, is determined by an ARMA model for farm j and the spatial correlation coefficients in the covariance matrix Σ are based on an exponential function of distance between pairs of wind farms. Other methods will also cross-correlate the random error components between multiple ARMA models with a covariance matrix but will not attempt to relate correlation to distance between wind farms. Instead, they rely

directly on a covariance matrix estimated directly from historical data. This is done in [28], for example.

Another common way to extend ARMA modeling to multivariate settings is through vector autoregressive moving average (VARMA) models. A model of order p, q (a VARMA(p,q) model) can be written in the same format as an ARMA(p,q) model, noting the following differences: X_t, c, and the W_t's are all length-J column vectors, the A_i's and the B_k's are $J \times J$ matrices, and $W_t \sim \mathcal{N}(0, \Sigma)$, where Σ is a $J \times J$ covariance matrix. As there are more parameters to fit than in the univariate case, a more efficient method than maximizing likelihood for finding the best fit set of parameters for a model M, $\hat{\theta} = \{c, A_1, \ldots, A_p, B_1, \ldots, B_q, \Sigma\}$, is ordinary least squares regression. The order p and q of the model can then be determined by minimizing the AIC where $\hat{\theta}$ is determined for each model M by ordinary least squares regression rather than by maximizing likelihood.

If we do not incorporate any of the $B_k W_{t-k}$ terms for $j > 0$ the VARMA(p,$q = 0$) model simplifies to a vector autoregressive model of order p (a VAR(p) model), given by

$$X_t = c + \sum_{i=1}^{p} A_i X_{t-i} + W_t,$$

where c is a length-J vector of constants, each A_i is a $J \times J$ dimension coefficient matrix for the lag-i vector X_{t-i}, and $W_t \sim \mathcal{N}(0, \Sigma)$, where Σ is a $J \times J$ covariance matrix. These tend to be much simpler to identify and work with than full VARMA(p,q) models. For multivariate series in which the current value of each component has an approximately linear dependence on its previous values and the previous values of other components, a VAR model is an appropriate choice. VAR models have been used for modeling spatially correlated wind speeds in both [12] and [24].

Note that in the ARMA model, the volatility σ is constant throughout the process as we assume we have a stationary time series. However, nonstationarity is common, and volatility may tend to change over time. To allow for this, a generalized autoregressive conditional heteroscedastic (GARCH) process of order p', q' described in [7], an extension of the ARCH(q') model presented in [16], can be used to model the conditional volatility at time t as $\sigma_t^2 = c + \sum_{n=1}^{p'} \phi_n W_{t-n}^2 + \sum_{m=1}^{q'} \zeta_m \sigma_{t-m}^2$. This can then be merged with the ARMA model by letting σ_t replace σ to produce a hybrid ARMA-GARCH model. An information criterion, such as the AIC, can then be used to guide model order. After the order is determined, parameters are again chosen to maximize likelihood.

In renewable power applications, allowing volatility to evolve over time based on recent volatility history would seem to make sense. To see why, consider, for example, a time period in which there are violent storms with intermittent calmer periods. The wind power generated by this weather pattern may be difficult to predict in a forecast. Thus, during these times forecast errors will likely be much more volatile than during calm periods. Having a general model to handle these volatility changes without needing additional explanatory variables can be useful, and indeed ARMA-GARCH models for generating wind speed scenarios have been used as in [26].

2.3. Other models and approaches to scenario generation. If, instead, we want to allow more parameters of our nonstationary model to evolve over time, we can use a state space

method. A state space model is used for wind speed scenarios in [14]. Another approach is to assume there exist distinct changes in regime over the course of the time series that cause the parameters to change suddenly; this idea is described in [22]. Using this method, one could use a discrete state Markov chain to model changes in regime, while fitting a unique ARIMA, GARCH, or other type of model to the data occurring in each regime. This is done in [40], for example, where wind power scenarios are generated using wind direction based regime switching linear models. Regime switching models are also the basis for the univariate and low dimensional multivariate crossing state models.

Other efforts in wind power scenario generation include using an artificial neural network to generate a forecast error time series [41]. In [34] scenarios are generated from probabilistic forecasts. This method is then extended in [29] to produce spatially dependent scenarios through the use of unique spatial correlation matrices for each forecast lead time. Copula techniques have been employed in wind power scenario generation techniques as well, such as in [43], where a copula is used to relate the joint distribution of actual and forecast power outputs to the marginals. A conditional forecast error can then be generated from this copula. This method can be extended to multiple wind farms. The spatial stochastic dependence of wind speed at different wind farm sites is captured using a copula in [21]. The copula is then used for generating spatially distributed wind speed scenarios.

It is important to note that none of the above models consider crossing time behavior. We have observed that without directly modeling this, crossing times at both the individual and aggregate levels will not match historical data. To our knowledge, this is the first time crossing time behavior has been directly modeled in a scenario generation method.

3. Crossing time distributions and crossing states. Unlike forecast error distributions, crossing time distributions are often difficult to replicate and will require specialized models. The empirical distributions are formed from data using the procedure below in which we start by defining crossing points.

For subprocess j, let the set of all indices such that errors crossed over from the negative to the positive regime (up-crossing points) be $\mathcal{C}_j^U = \{t | X_{t-1,j} \leq 0 \wedge X_{t,j} > 0\}$. Likewise, the set of down-crossing points is defined as $\mathcal{C}_j^D = \{t | X_{t-1,j} > 0 \wedge X_{t,j} \leq 0\}$. An up-crossing time of duration d starting at time $t' \in \mathcal{C}_j^U$ is then defined as

$$T_{t',j}^U = d \text{ if } \begin{cases} X_{t'+i,j} > 0 & \forall i \in \{0, 1, \ldots, d-1\}, \\ X_{t'+d,j} \leq 0. \end{cases}$$

Similarly, a down-crossing time of duration d starting at time $t' \in \mathcal{C}_j^D$ is defined as

$$T_{t',j}^D = d \text{ if } \begin{cases} X_{t'+i,j} \leq 0 & \forall i \in \{0, 1, \ldots, d-1\}, \\ X_{t'+d,j} > 0. \end{cases}$$

Examples of points in time belonging to $\mathcal{C}_j^U$ and $\mathcal{C}_j^D$, as well as an up- and a down-crossing time, are shown in Figure 1. For both up-crossing and down-crossing times, for each subprocess j there exists empirical disaggregate distributions F_j^U and F_j^D, respectively. A similar procedure can be carried out for the aggregate series as well to form empirical distributions $F^{U,agg}$ and $F^{D,agg}$.

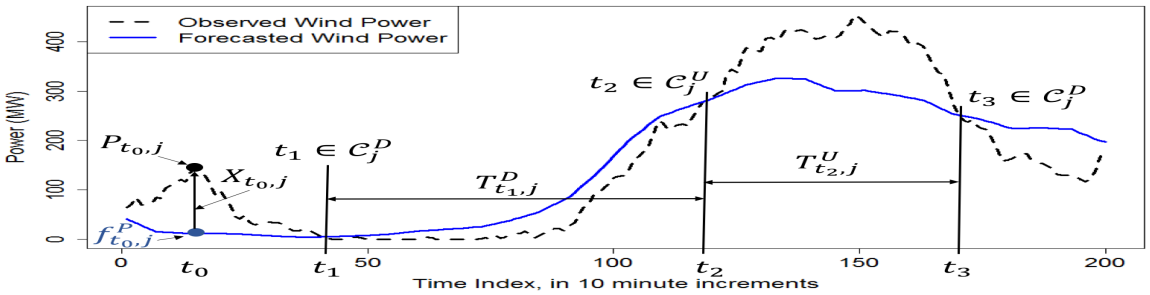

Figure 1. *Points in time belonging to $\mathcal{C}_j^U$ and $\mathcal{C}_j^D$. A forecast error and both an up- and a down-crossing time are also shown.*

In order to describe the models presented in this paper, it is necessary to introduce the concept of the *crossing state* of a process. Assume we have a vector of point forecasts (or, in general, a vector of benchmark reference points) at lead time t, f_t^P for J stochastic subprocesses. For each subprocess $j \in \{1, 2, \dots J\}$, let element j of the crossing state be the indicator variable $S^C_{t,j} = \mathbf{1}_{\{P_{t,j} > f^P_{t,j}\}}$, or equivalently $S^C_{t,j} = \mathbf{1}_{\{X_{t,j} > 0\}}$, representing whether or not the subprocess is above or below its forecast at time t. The variable $S^C_{t,j}$ is called the subprocess crossing state for subprocess j. The full time t crossing state can then be defined as the length J column vector

$$S_t^C = \left(S^C_{t,1}, S^C_{t,2}, \dots, S^C_{t,J}\right)^T = \left(\mathbf{1}_{\{X_{t,1}>0\}}, \mathbf{1}_{\{X_{t,2}>0\}}, \dots, \mathbf{1}_{\{X_{t,J}>0\}}\right)^T.$$

We incorporate the crossing state into our models in order to control the crossing times of our sample paths. Note that sample paths in which crossing state transitions are similar to those seen in training data will also have similar crossing time distributions (at least at the disaggregate level) because this is precisely what the crossing state models—the consecutive periods of time for which each subprocess is above or below its forecast. If we simply rely on the natural transitions in the crossing state produced by a standard time series model when simulating a forecast error time series, we obtain poor replications of crossing time distributions. However, if we can accurately model the dynamics of the crossing state, we can guide, or possibly force, the time series into consistency with the crossing state (consistency would be ensuring $X_{t,j} > 0$ when $S^C_{t,j} = 1$ or $X_{t,j} \leq 0$ when $S^C_{t,j} = 0$) in hopes of correcting the discrepancies between actual and simulated crossing time distributions. The methods for ensuring consistency are described later.

Finally, note that this is not a strict definition of the crossing state, and there are some variations in the following models. However, all the models utilize the crossing state for the same two purposes—to control the crossing times of the simulations and to influence the error generation process (to be described later as well).

4. A univariate hidden semi-Markov crossing state model. In this section we develop a nonparametric univariate crossing state hidden semi-Markov model (HSMM). Note that the low dimensional model presented later in section 5 can also be applied to univariate settings. However, the HSMM can be more useful in sequential decision making problems

under uncertainty as there are a relatively small number of information states which the exogenous process can be in at any time t. This would, for example, keep the run time of finding an optimal control policy for the system by solving the full backward Markov decision process low relative to a model which can take on many states. Furthermore, this model lends itself to any policy which relies on fitting value functions to system states.

4.1. Crossing state transition model: A semi-Markov model. Recall that for both the up-crossing and down-crossing times, there exist distributions F^U and F^D, respectively. Up-crossing time distributions are quantized by partitioning into Q bins, splitting at the quantile points such that q_i^U is the ith quantile for $i \in \{1, 2, \ldots, Q\}$ (where q_Q^U represents the maximum value in the distribution). Note that this implies the bins will likely not be equally sized. If we also let $q_0^U = 0$, an up-crossing time $T_{t'}^U$ starting at t' would then belong to bin b_i^U if $q_{i-1}^U \geq T_{t'}^U < q_i^U$, with the exception being that the maximum value in the distribution belongs to bin b_Q^U. If, for example, Q is chosen to be 3, the lower third of up-crossing times belonging to bin b_1^U could be interpreted as "short runs above the forecast," while the middle third belonging to bin b_2^U would be interpreted as "medium length runs above the forecast." The same procedure is carried out for down-crossing times.

We then define our crossing state variable S_t^C as the pair of variables describing whether the path is above the forecast and what crossing time duration bin the system is currently in. We can express this as the pair $S_t^C \equiv (I_t, B_t)$, where

$$I_t = \begin{cases} U \text{ if } X_t > 0, \\ D \text{ if } X_t \leq 0, \end{cases}$$

and, after identifying $\text{argmax}_{t' \in \mathcal{C}^{I_t}}(t' - t)$ such that $t' - t \leq 0$ (finding the most recent crossing point), $B_t = i$ if $T_{t'} \in b_i^{I_t}$. The set of all possible crossing states is given by $\mathcal{S}^C$, which is the Cartesian product of the possible I_t's and B_t's. When training is finished, there exists a distribution of crossing times F_s^T for each crossing state $s \in \mathcal{S}^C$. This will be important for simulation.

Note if $t + 1 \notin (\mathcal{C}^U \cup \mathcal{C}^D)$, $P(S_{t+1}^C = S_t^C | S_t^C) = 1$ as the crossing state remains constant until errors switch signs. If, however, $t+1 \in (\mathcal{C}^U \cup \mathcal{C}^D)$, the crossing state transition matrix has a nontrivial $P(S_{t+1}^C = s' | S_t^C)$ for $s' \in \mathcal{S}^C$. This is computed from data as follows. Considering only points in time such that $t + 1 \in (\mathcal{C}^U \cup \mathcal{C}^D)$, let $n(S_{t+1}^C = s' | S_t^C = s)$ be the count of the transitions from state s to state s' occurring for each pair of crossing states (s', s) and let $n(S_t^C = s)$ be the number of times $S_t^C = s$ for each crossing state s. The transition probability from crossing state s to s' is then given by

$$P(S_{t+1}^C = s' | S_t^C = s) = \frac{n(S_{t+1}^C = s' | S_t^C = s)}{n(S_t^C = s)}. \tag{1}$$

During simulation, we insist that the crossing state remains constant for a crossing time drawn from F_s^T before being allowed to transition according to $P(S_{t+1}^C = s' | S_t^C = s)$. This transition model is a semi-Markov model.

Note that half of the entries in each row of the transition probability matrix must be equal to zero as down-crossing states ($I_t = D$) cannot transition to another down-crossing

state, and likewise for up-crossing states ($I_t = U$). The remaining half of the row forms a multinomial distribution. Individual transition probabilities are estimated using (1), which gives the maximum likelihood estimator $P(S^C_{t+1} = s'|S^C_t = s)$ with corresponding standard deviation

$$\sqrt{\frac{P(S^C_{t+1} = s'|S^C_t = s)(1 - P(S^C_{t+1} = s'|S^C_t = s))}{n(S^C_t = s)}}.$$

This is useful in guiding the choice of Q as larger values of Q will partition the crossing time data into finer bins and result in fewer observations of transitions to form transition probability estimates. Since duration bins are divided at points corresponding to evenly spaced quantiles of the crossing time distributions, the number of observations per bin will be approximately equal. Thus, if one has a training data set with N total crossings, one can expect $n(S^C_t = s) = \frac{N}{2Q}$ for each $s \in \mathcal{S}^C$, where the factor of 2 stems from the fact that there are both up- and down-crossings.

4.2. Error generation model: A discrete state Markov chain. The crossing state-conditioned error distributions F^X_s are not identical across all possible crossing states $s \in \mathcal{S}^C$. In fact, they are likely to be quite different, such as in the case where the error distribution is asymmetric. Furthermore, the error variance likely to increase with run length as well as longer deviations from the forecast tend to produce larger error magnitudes. This behavior is seen in Figure 2, which shows error densities for each type of run length bin with $Q = 3$. Thus, to better capture the behavior of the error process, the error generation model is conditioned on the crossing state.

In addition to errors being crossing state-dependent, they are dependent on error history as well; a first order discrete state Markov chain is used to model this behavior. As errors may be continuous or take on many discrete values, we must first quantize the distribution to form a manageable number of error states. Similar to how the crossing time distributions are partitioned, each error distribution F^X_s for $s \in \mathcal{S}^C$ is split into R bins: $b^s_1, b^s_2, \ldots, b^s_R$, splitting

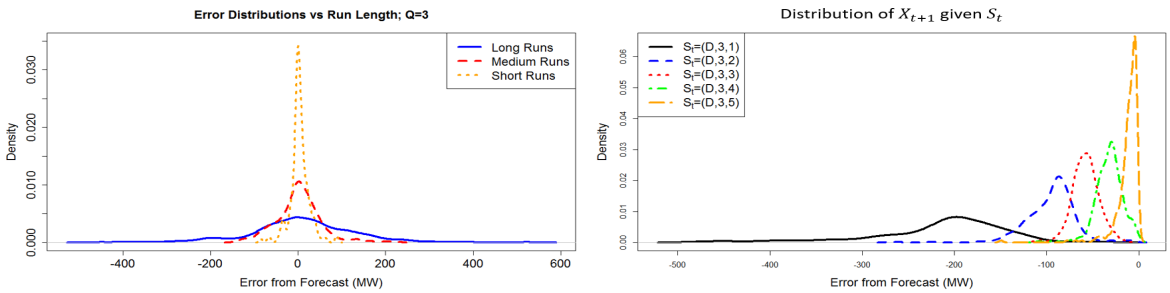

Figure 2. *Left: Error distributions conditioned on B_t, the crossing time bin, with $Q = 3$. Both positive and negative errors for equal values of B_t are combined to form the distributions. The variance of the errors tends to increase with run length. Right: Example of conditional distributions for X_{t+1} given a fixed crossing state, $S^C_t = (D, 3)$, but varying which error bin, X^q_t, that X_t belongs to. The magnitude of the next error is largely dependent on the magnitude of the current error.*

at the quantile points such that q_i^s is the ith quantile for $i \in \{1, 2, \ldots, R\}$ (where q_R^s represents the maximum value in the distribution).

Letting q_0^s be the minimum value in the distribution, $X_t \in b_i^s$ if $q_{i-1}^s \leq X_t < q_i^s$ (with the exception that the maximum value in the distribution belongs to bin b_R^s) and the crossing state at time t is s. Then, given $X_t \in b_i^s$, we have a conditional distribution for the error at time $t+1$ giving $P(X_{t+1}|s, X_t \in b_i^s)$. Therefore, the complete information state at time t is defined as $S_t \equiv (S_t^C, X_t^g) \equiv (I_t, B_t, X_t^g)$, where $X_t^g = i$ if $X_t \in b_i^{(I_t, B_t)}$.

The dependence of X_{t+1} on X_t^g, the state of the current error, is illustrated in Figure 2 in which conditional distributions for X_{t+1} are plotted for a fixed crossing state, but varying error states X_t^g. In the example $Q = 3$ and $R = 5$. Note that the number of possible information states this stochastic process may be in at any time is $(Q \times 2) \times R$, where an information state contains all necessary information about the process at time t to determine its distribution at time $t+1$. Let the set of all the possible information states be given by $\mathcal{S}$ and the distribution of the error at time $t+1$ given we are in information state $i \in \mathcal{S}$ at time t be F_i^X.

Overall, this model fits under the category of hidden semi-Markov models as the crossing state is partially unobservable. The HSMM was first introduced in [18], and since then has been used for a wide variety of applications including some time series modeling applications (see [42] for more examples). However, the concept of using an HSMM to model crossing times for generating sample paths is new.

There are a few practicalities to consider when using the model to generate a sample path. To produce a sample path of errors from forecast of length L, assuming that Q and R have been chosen, use Algorithm 1, which is written in pseudocode in which vectors are indexed starting at 1.

Choosing the parameters Q and R is not a trivial task as one risks overfitting by choosing values that are too large for the available training data. If an extensive set of training data

Algorithm 1. Univariate crossing state HSMM: Simulation procedure.

From training data, for $s', s \in \mathcal{S}^C$ form and store transition matrix with entries giving $P(s'|s)$ and empirical distributions F_s^T, F_s^X. Also for $i \in \mathcal{S}$, form F_i^X.
Initialize $E = [\,]$
Initialize $t \leftarrow 1$
Sample an initial crossing state $s \in \mathcal{S}^C$.
while $t < L$ **do**
  Sample a run length (crossing time) r from F_s^T.
  $V = zeros(r)$
  for $t' = 0, 1, 2, \ldots, r-1$ **do**
    if $t' = 0$ **then**
      $V_{t'+1}$ is sampled from F_s^X.
    else
      Determine $X_{t'}^g$ from the error $V_{t'}$ and s, and the $i \in \mathcal{S}$ corresponding to $(s, X_{t'}^g)$.
      $V_{t'+1}$ is sampled from F_i^X.
    end if
  end for
  $E \leftarrow concat(E, V)$
  Change crossing states by sampling according to the probabilities found transition matrix row $P(s'|s)$ for $s' \in \mathcal{S}^C$.
  Let $s \leftarrow s'$ for the state s' that was sampled.
  $t \leftarrow t + r$
end while
return $E_{1:L}$

is available, larger values of Q and R may be chosen as this will allow the model to capture more complex dynamics in the transition of the crossing state as well as the error generation process. However, as Q and R increase, the data is partitioned into an increasing number of bins. If there are only a few observed crossing times belonging to each bin, we are likely overfitting to data by choosing Q too large (a discussion of the precision of crossing state transition probability estimates is included in subsection 4.1). Similarly, if there are relatively few observed errors belonging to each error bin, R should be decreased. As any choice of Q and R will allow the model to capture the crossing time distributions (as it does so by construction), one should err on the side of caution when choosing Q and R to avoid overfitting.

5. Low dimensional multivariate crossing state model.

A motivating application for a low dimensional crossing state model is the control of a microgrid in which we have access to three wind farms located far enough apart that they cannot be accurately modeled as one source but close enough to exhibit correlation. In addition to fossil fuel generation plants and other sources of power, there are energy storage devices available. The goal is to find a robust control policy—one that is unlikely to fail even in scenarios in which renewable power falls below its forecast for extended periods of time. To develop or test candidate policies, a multidimensional model for wind power outputs in which crossing times are preserved at the individual level and the aggregate level would be useful.

Depending on the application and available training data, the maximum number of dimensions J for which this model is usable varies. A recommendation is to use this model for $J \leq 3$ and the high dimensional model when $J \geq 5$. If $J = 4$, one may want to experiment with both models to determine which is more appropriate.

5.1. Forecast bin-conditioned Z-transforms.

We first discuss the forecast bin-conditioned Z-transform in greater detail as it is utilized in both multivariate models. There are situations (such as modeling wind) where the stochastic process must fall within specified bounds. In this case, it is useful to condition on the forecast itself. This is because when forecasts are near the minimum, errors tend to have a heavy positive skew; the opposite is true for forecasts near the maximum.

Figure 3, in which empirical conditional distributions are shown for one wind farm's forecast errors conditioned on five different forecast bins, displays how the distribution of forecast errors is heavily dependent on the forecast level. Thus, before forming our error generation model, we perform this type of conditional Z-transform. As mentioned in section 2, [27] also employs a similar transform. More formally, for subprocess j, we first divide the interval $[P_j^{min}, P_j^{max}]$ into m subintervals to form forecast bins: $b_{j,1} = [P_j^{min}, p_{j,1}), b_{j,2} = [p_{j,1}, p_{j,2}), \ldots, b_m = [p_{j,m-1}, P_j^{max}]$, where $P_j^{min} < p_{j,1} < p_{j,2} < \cdots < P_j^{max}$. Then, the random variable $Y = y$ if $f_{t,j}^P \in b_{j,y}$. For the multivariate models, $Z_{t,j}^Y$ results from this forecast bin-conditioned transform.

5.2. Crossing state transition model: A semi-Markov model.

The crossing state transition semi-Markov model used here is similar to the one presented in subsection 4.1, but with some variations. First, we make a slight change to the basic crossing state format from section 3 by adding the indicator variable for whether or not the aggregated output is above or below its forecast. So our crossing states now have the form

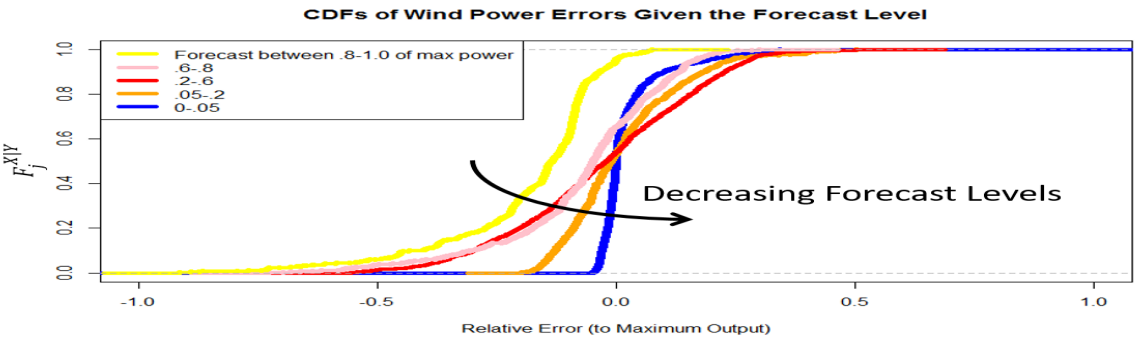

Figure 3. *Empirical conditional CDFs of wind power forecast errors from one wind farm where conditioning is done on the forecast bin to which the forecast at lead time t belongs. The positive skew of the errors for forecasts near the minimum value (0 MW) and the negative skew for forecasts near the maximum value (maximum power output of the wind farm) can be clearly seen here.*

$$S_t^C = \left(S_{t,1}^C, \ldots, S_{t,J}^C, S_t^{C,agg}\right)^T = \left(\mathbf{1}_{\{X_{t,1}>0\}}, \ldots, \mathbf{1}_{\{X_{t,J}>0\}}, \mathbf{1}_{\{X_t^{agg}>0\}}\right)^T.$$

The set of possible crossing states, $\mathcal{S}^C$, has cardinality $2^{J+1}-2$ as the crossing states $(0,\ldots,0,1)^T$ and $(1,\ldots,1,0)^T$ cannot exist. From the training data we can then form empirical run length distributions, F_s^T, for each crossing state $s \in \mathcal{S}^C$, where a run length, T^s, is a consecutive period of time for which the crossing state remains unchanged. Note these are not exactly the same as crossing times, but observe that if we can model the transitions between crossing states accurately, as well as the lengths of time we remain in each crossing state (run lengths or sojourn times), we will be able to replicate crossing times at both the aggregate and disaggregate levels.

We use a semi-Markov model for the evolution of the crossing state in which the crossing state is held constant for a period of time sampled from F_s^T. After this period, the crossing state must transition. Much like in the one-dimensional crossing state transition model, we can estimate the transition probability between each pair of crossing states by first isolating consecutive pairs of points between which crossing states changed in the training data. Then, using (1), we compute the transition probability between each pair of crossing states $P(S_{t+1}^C = s' | S_t^C = s)$ for $(s', s) \in \mathcal{S}^C \times \mathcal{S}^C$. The process of holding the crossing state constant for a sampled run length and then changing crossing states by sampling according to the probabilities found in transition matrix row $P(S_{t+1}^C | S_t^C)$ is repeated until the end of the simulation.

5.3. Error generation model: A regime switching VAR model. Overall, this crossing state model is a regime switching VAR model that uses a semi-Markov model for regime transitions. The regime transition model is in subsection 5.2, and there are $2^{J+1}-2$ different VAR models used for error generation, one for each crossing state. These VAR models are fit to forecast bin-conditioned Z-transformed errors rather than the errors themselves. The rationale behind using separate VAR models for the error generation in each regime is that the stochastic dependencies of the subprocesses can be fundamentally different during periods

when many or all of the farms are above their forecast than during periods when many or all of the farms are below their forecast.

5.3.1. Fitting a VAR model for each crossing state. Recall that an order p VAR model has parameters $\hat{\theta} = \{c, A_1, \ldots, A_p, \Sigma\}$. Thus, to form a unique VAR model for each crossing state $s \in \mathcal{S}^C$, we must first determine the proper order of the model for the crossing state s, p^s, and then find the corresponding best fit parameter set $\hat{\theta}^s = \{c^s, A_1^s, \ldots, A_p^s, \Sigma^s\}$. This procedure is described next, in which we fit a VAR(p^s) model to the training data occurring in the crossing state s by finding p^s and subsequently $\hat{\theta}^s$ for a single crossing state s.

Given the time series, $\{X_t\}_{t=0}^T$, we first find the crossing state, S_t^C, at each time period. A forecast bin-conditioned series of Z-variates is also formed from the data, $Z_{t,j}^Y = \Phi^{-1}(F_j^{X|Y}(X_{t,j}))$, where $Y = y$ if $f_{t,j}^P \in b_{j,y}$ for $t \in \{0, 1, \ldots, T\}$. We then isolate all the data Z_t^Y such that $S_t^C = s$ and concatenate it together in order of increasing t to construct a new time series, $\{Z_{t'}^{Y,s}\}_{t'=0}^{T'}$, where T' is the length of the new series, which has gaps in the original series for which $S_t^C \neq s$ removed. Then, using the newly constructed series of Z-variates $\{Z_{t'}^{Y,s}\}_{t'=0}^{T'}$, we use ordinary least squares regression to find the parameter set $\hat{\theta}^s = \{c^s, A_1^s, \ldots, A_p^s, \Sigma^s\}$ for different values of p^s and choose the $(p^s, \hat{\theta}^s)$ pair such that the AIC is minimized. The same procedure is then carried out for all other possible crossing states $s' \in \mathcal{S}^C$, $s' \neq s$ as well.

5.3.2. Generating errors with crossing state-dependent VAR models. If the process is in crossing state $s = S_t^C$ at time t, the VAR(p^s) model corresponding to s is used to generate the Z-variate errors at time t,

$$Z_t^Y = c^s + \sum_{i=1}^{p^s} A_i^s Z_{t-i}^Y + W_t,$$

where $W_t \sim \mathcal{N}(0, \Sigma^s)$. Algorithm 2 (in which vectors are indexed starting at 1) describes the simulation procedure for producing a sample path of length L, given all the model parameters are already determined from the training data. However, one aspect of simulation that should be highlighted is the rejection and resampling of error vectors that are inconsistent with the crossing state. Inconsistency occurs when $X_{t,j} \leq 0$ and $S_{t,j} = 1$ or vice versa. Additionally, even if the individual errors are consistent with their corresponding crossing state components, we can still obtain an inconsistency if the summation of the errors is inconsistent with $S_t^{C,agg}$. The following is an example of this scenario in $J = 2$ dimensions, accompanied by Figure 4.

Assume at time $t+1$ in our simulation our current crossing state is $S_{t+1}^C = (0, 1, 1)^T$, where the last element represents the aggregate level indicator variable. Imagine the simulation produces the time $t + 1$ error vector (following the inverse Z-transform) $X_{t+1} = (-3, 1)^T$. Though each individual subprocess is consistent with S_{t+1}^C, this results in an inconsistency as we obtain an aggregate error of $X_{t+1}^{agg} = -2$ which corresponds to a crossing state of $S_{t+1}^C = (0, 1, 0)^T$.

If we do encounter an inconsistency, the sample point is rejected and resampled until consistency is achieved (in Figure 4, consistency is achieved when $X_{t+1} = (-5, 8)^T$) or a stopping condition (maximum number of resamples) is reached. If the stopping condition is reached, we retain the most recent sample and proceed.

Algorithm 2. Low dimensional crossing state model: Simulation procedure.

Initialize $Z^Y = [\,]$
Initialize $t \leftarrow p^{max}$, where p^{max} is the largest order of all the VAR models.
Retain initial t Z-transformed errors and initialize $s = S_t^C$.
while $t < L$ **do**
  Sample a run length r from empirical distribution F_s^T.
  for $t' = 1, 2, \ldots, r$ **do**
    $Z_{t+t'}^Y = c^s + \sum_{i=1}^{p^s} A_i^s Z_{t+t'-i}^Y + W_{t+t'}$ where $W_{t+t'} \sim \mathcal{N}(0, \Sigma^s)$
    Set $Resamples = 0$
    while (The crossing state resulting from the inverse Z-transforms of the vector $Z_{t+t'}^Y \neq s$) **and** ($Resamples <$ $MaxResamples$) **do**
      Resample $W_{t+t'} \sim \mathcal{N}(0, \Sigma^s)$ in the VAR(p^s) model.
      $Resamples \leftarrow Resamples + 1$
    end while
  end for
  Change crossing states by sampling according to the transition matrix row giving $P(s'|s)$ for each crossing state $s' \in \mathcal{S}^C$. Let $s \leftarrow s'$ for the s' that was sampled.
  $t \leftarrow t + r$
end while
return $Z_{1:L,1:J}^Y$

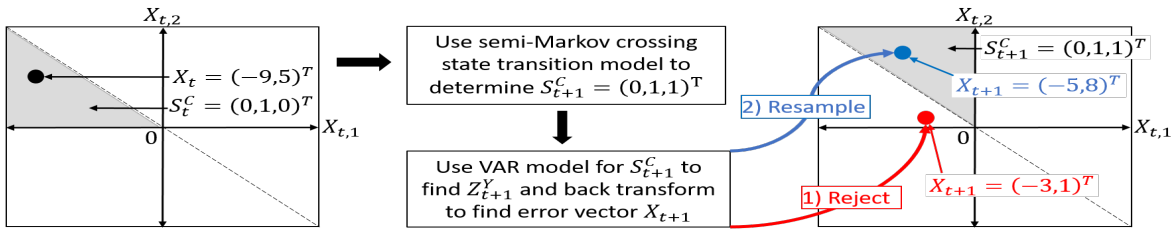

Figure 4. *Graphical representation of rejection and resampling of inconsistent error vectors in $J = 2$ dimensions.*

The stopping condition is crucial to the efficiency of the algorithm, as the simulation may encounter cases where the probability of sampling a point consistent with the crossing state is very low, resulting in an excessive number of rejections. This is a common problem suffered by standard rejection-based sampling methods in multiple dimensions. Depending on the dimensionality and characteristics of the stochastic process, the rejection rate will vary. For reference, in the spatially distributed wind power application with $J = 3$ dimensions the rate at which we reach this stopping condition is approximately 7 percent. With an additional wind farm ($J = 4$ dimensions) this rises to approximately 9 percent. In this application, this relatively low rate is acceptable. However, if one finds that the rejection rate is too high and is affecting the quality of the simulations (by reaching the stopping condition too frequently), one can consider integrating methods that sample from constrained densities to improve efficiency (see [9, 19]) into the algorithm.

Finally, before considering how to extend the crossing state model to higher dimensions, we discuss a weakness shared by both the univariate and low dimensional models as they are presented and suggest a method for circumventing the problem. By simulating the crossing

state transitions and crossing times separately from the errors, error magnitude is not directly related to the elapsed time since the last crossing or the remaining time before the next crossing. This may result in unusually large jumps between the positive and negative regimes when the crossing state transitions. One way to curb this effect is to fit both an initial and a terminal error density for each crossing state. Upon entering a new state, one would sample an error (or error vector) from the initial distribution. Then, prior to exiting the state, the final error is sampled from the terminal density. These error distributions would likely exhibit means closer to zero as well as smaller variances, resulting in smoother transitions between regimes. This alteration to the model is relatively easy to include for both the univariate and low dimensional models, but not for the high dimensional model presented in the subsequent section. Thus, we have not included it in the general framework of the models to maintain consistency.

6. High dimensional multivariate crossing state model. In higher dimensional settings, such as a large power grid with many spatially distributed wind farms, the previous crossing state model does not scale for several reasons. With J subprocesses there are, at minimum, 2^J possible crossing states. A semi-Markov crossing state transition model would certainly not be scalable as, for one, we cannot maintain a look-up table representation of the transition function. In addition, we are unable to fit a unique error generation model to each crossing state using the same methods. A high dimensional model is thus developed to circumvent these issues.

6.1. Crossing state transition model: A logistic model. Here we use the definition of the crossing state found in section 3. Instead of a look-up representation of the transition function for the full crossing state, we model the evolution of each subprocess crossing state with a logistic model that incorporates the previous p crossing states in the transition function. This scalable approximation of the transition probability for each element of the crossing state is used to control the crossing times of the individual subprocesses. The model is described below.

For each element j we estimate

$$P(S^C_{t+1,j} = 1 | S^C_t, \ldots, S^C_{t-p+1}) = \frac{1}{1 + exp\left[c_j + \beta_{0,j} S^C_t + \cdots + \beta_{p-1,j} S^C_{t-p+1}\right]}$$

and

$$P(S^C_{t+1,j} = 0 | S^C_t, \ldots, S^C_{t-p+1}) = 1 - P(S^C_{t+1,j} = 1 | S^C_t, \ldots, S^C_{t-p+1}),$$

where the scalar parameter c_j and the length J parameter row vectors $\beta_{0,j}, \ldots, \beta_{p-1,j}$ are chosen to minimize the probability of incorrect selection from the training data. Element j' of $\beta_{p',j}$ is the coefficient corresponding to the j'th element of the lag-p' crossing state in the logistic transition model for the jth crossing state component. During simulation, a crossing state transition for component j is made by first finding $P(S^C_{t+1,j} = 1 | S^C_t, S^C_{t-1}, \ldots, S^C_{t-p+1})$ using the previous p crossing states. Then, a uniform random variable $U \sim Unif[0,1]$ is sampled. If $U < P(S^C_{t+1,j} = 1 | S^C_t, S^C_{t-1}, \ldots, S^C_{t-p+1})$, then $S^C_{t+1,j} = 1$; otherwise, $S^C_{t+1,j} = 0$.

6.2. Error generation model: Individual lag-p linear models. Just as we allow each element of the crossing state to evolve separately, rather than modeling the dynamics of the entire Z-variate vector with a single model such as a VAR model, each subprocess j will have its own $S^C_{t,j}$-dependent transition model. This allows the sample point $Z^Y_{t,j}$ to be generated independently of $Z^Y_{t,j'}$ for $j' \neq j$ at time t. This turns out to be a reasonable simplification here because in many cases, if we examine the standard VAR(p) model more closely, we notice the off-diagonal terms of the covariance matrix Σ for the multivariate normal random vector W_t are much smaller in magnitude than the main diagonal entries and often relatively close to 0. If we do assume the covariance matrix Σ is diagonal, the VAR(p) model,

$$Z^Y_t = c + \sum_{i=1}^{p} A_i Z^Y_{t-i} + W_t,$$

where $W_t \sim \mathcal{N}(0, \Sigma)$, is separable, and we can rewrite it as J decoupled equations, one for each component j of Z^Y_t,

$$Z^Y_{t,j} = c_j + \sum_{i=1}^{p} A_{i,j} Z^Y_{t-i} + W_{t,j},$$

where c_j is the jth element of the constant vector c, $A_{i,j}$ is the jth row of the lag-i coefficient matrix A_i, and $W_{t,j} \sim \mathcal{N}(0, \sigma^2_j)$, where $\sigma^2_j = \Sigma_{j,j}$.

We can then incorporate crossing state-dependence by forming two models of this form for each component j, one for when subprocess j is above its forecast ($S^C_{t,j} = 1$), and one for when it is below its forecast ($S^C_{t,j} = 0$). We perform the training procedure described in subsection 5.3.1 with the following modifications. For *each* subprocess j we first construct two multivariate time series from the training data—one from times at which the subprocess crossing state $S^C_{t,j} = 1$, and another when $S^C_{t,j} = 0$. For these two series, we fit linear models of the form

$$Z^Y_{t,j} = c^s_j + \sum_{i=1}^{p^s} A^s_{i,j} Z^Y_{t-i} + W_{t,j},$$

where $W_{t,j} \sim \mathcal{N}(0, \sigma^{2,s}_j)$ for $s \in \{0, 1\}$. The order of subprocess j's linear model for subprocess crossing state s, p^s, can be chosen by minimizing the AIC. The remaining parameters for each model—c^s_j, the $A^s_{i,j}$ coefficient row vectors, and σ^s—are those corresponding to p^s which were determined using ordinary least squares regression.

The change from the low dimensional multivariate method to the high dimensional multivariate method is necessary as we cannot form $2^{J+1} - 2$ separate VAR models for each crossing state as in section 5 for larger values of J from a limited amount of training data. Additionally, even if infinite training data was available, since it is unlikely a high dimensional crossing state will remain unchanged for several consecutive time periods, the concatenated error series would contain large time gaps between each point. Thus, the VAR models fit to these series would not be a good representation of how the error vector evolves from one time period to the next.

A rejection method is used here to ensure sampled errors are consistent with the crossing state and is very similar to the one explained in subsection 5.3.2. At each time t, for each component j, $W_{t,j}$ is sampled until either (1) $X_{t,j} > 0$ and $S^C_{t,j} = 1$ or $X_{t,j} \leq 0$ and $S^C_{t,j} = 0$ or (2) a maximum number of resampling iterations has been reached. If we reach the second case, one may chose to leave the generated error as is or reject one additional time and sample an error from the observed distribution of forecast bin-conditioned Z-transformed errors that occur in the current crossing state of subprocess j, $S^C_{t,j}$. Though the latter strategy occasionally samples from a secondary distribution (the observed error distribution) instead of the intended distribution, the method presented in this paper employs this strategy as it produces more accurate crossing time distributions.

6.3. Postsimulation sample path smoothing. If the high dimensional method as described until this point is utilized for simulation, the resulting sample paths will have error distributions closely matching observed distributions at both the aggregate and disaggregate levels. Additionally, the simulated crossing time distributions will be consistent with observed crossing time distributions at the disaggregate level. However, at the aggregate level, simulated crossing times will tend to be shorter than observed crossing times. We now search for a method to correct this.

The inability to replicate aggregate crossing time distributions stems from the fact that, unlike the low dimensional model, the high dimensional model does not have an indicator variable to control the crossing times of the aggregate sum. One might consider including this in the logistic model to simulate the evolution of the aggregate crossing state as well. We could then follow the procedure of generating and rejecting samples until agreement with the crossing state at both the aggregate and disaggregate levels is reached. However, in high dimensions we would often encounter cases where the probability of acceptance is extremely low and, unlike in the low dimensional model, resampling is unlikely to produce a point consistent with the crossing state at both the individual and aggregate levels. Thus, it is not useful to simulate the aggregate crossing state as we cannot enforce agreement with it.

It is, however, simple to guarantee that each individual component is consistent with its individual subprocess crossing state. Once the crossing state is simulated using the logistic model (subsection 6.1), we sample errors as explained in subsection 6.2, rejecting if necessary. However, the rejection probability is often significantly lower in a single dimension. Resampling can thus successfully correct inconsistencies with the crossing state at the disaggregate level.

To achieve better performance at the aggregate level, a smoothing step is employed to reduce some of the variability in the sample paths at the aggregate level without a major change to the model that would disrupt its success in replicating the majority of the desired distributions. The smoothing step is simply an adjustment to the simulated aggregate paths such that their crossing times are more consistent with the observed data. The adjustment may come at a small expense (if any) to the quality of the simulated error distributions, but capturing the aggregate crossing time behavior is of high priority for the application we consider.

If we view the sample paths more carefully, we see that at the aggregate level some of the longer runs above or below the forecast are interrupted by very short periods of time for which

the error switches signs. Furthermore, this pattern is not observed often in the historical data. Thus, a small integer a is chosen for which we attempt to eliminate any crossing times at the aggregate level less than a through resampling. The aggregate sample paths are, in effect, smoothed slightly.

A short example of this process is explained here to clarify the procedure. The full smoothing step is also described in Algorithm 3. Suppose the high dimensional method is used to produce simulations that result in the following aggregate crossing states for times $t = 0, 1, \ldots, 99$: $S_t^{C,agg} = 0$ for $t \in [0, 57]$, $S_t^{C,agg} = 1$ for $t \in [58, 59]$, and $S_t^{C,agg} = 0$ for $t \in [60, 99]$. Also suppose $a = 3$. In this case, the error vectors X_{58} and X_{59} would be resampled using the appropriate linear model for each component of the vector until $S_{58}^{C,agg} = 0$ and subsequently $S_{59}^{C,agg} = 0$ as well (or a maximum number of resampling iterations is reached). Instead of 2 medium-length crossing times of length 58 and 40 and a short one of length 2 in between, we are left with a smoothed sample path and one longer crossing time of 100 time steps.

Picking an appropriate a requires some tuning as different stochastic processes (or wind data from different seasons) will likely exhibit different behaviors. a should be chosen to produce the best fit between simulated aggregate crossing time distributions and observed distributions. The result of the smoothing step is that, without affecting the ability of the model to fit the other desired distributions, we are able to significantly improve the simulated aggregate crossing time distributions.

7. Numerical results.

The day ahead unit commitment problem in PJM Interconnections, in which steam generation must be scheduled a day in advance, is one important application of the crossing state-based scenario generation methods. Thus, the forecast series of interest in this section are the 12–36 hour ahead wind power forecasts. PJM uses an external forecasting vendor using a proprietary method, but all we need is a history of forecasts and the corresponding actual power production from the different wind farms. Both the forecast and output series are available in 10-minute intervals and a month's worth of data is collected from wind farms across the United States' Great Plains region from January 2013.

In this section, we first use the data to fit univariate models for the individual farms. We then address the problem of replicating output at both the disaggregate and aggregate levels with multivariate models.

In each setting (univariate, low dimensional multivariate, high dimensional multivariate), the models simulate 20 forecast error scenarios over the course of one month. For each wind farm involved, the forecast output is then added to each scenario to produce wind power output scenarios. These must be nonnegative and are clipped if they fall below 0 MW. They are also capped at the maximum power capacity of each wind farm. The resulting simulated crossing time and error distributions are formed from the combination of all scenarios produced by a single model combined.

We will be evaluating models on their ability to produce scenarios that replicate forecast error and crossing time distributions. A main tool we use as an indicator of performance is a weighted two-sample quadratic empirical distribution function statistic as a distance measure between two probability distributions. For empirical CDF's F^X of a size N sample and F^Y of a size M sample, the general weighted Q^2 statistic is given by

Algorithm 3. High dimensional crossing state model: Simulation procedure.

Initialize $Z^Y = [\,]$
Retain initial forecast conditioned Z-transformed errors, $Z^Y_{t,j}$, up to the greatest order p^{max} in all the linear models.
Find initial crossing states S^C_t for $t \in \{1, 2, \ldots, p^{max}\}$.
for $t = p^{max}, p^{max}+1, \ldots, L-1$ **do**
  for $j = 1, 2, \ldots, J$ **do**
    $P(S^C_{t+1,j} = 1 | S^C_t, S^C_{t-1}, \ldots, S^C_{t-p+1}) = \frac{1}{1+exp[c_j + \beta_{0,j} S^C_t + \beta_{1,j} S^C_{t-1} + \cdots + \beta_{p-1,j} S^C_{t-p+1}]}$
    Sample $U \sim Unif[0,1]$
    If $U < P(S^C_{t+1,j} = 1 | S^C_t, S^C_{t-1}, \ldots, S^C_{t-p+1})$ then $S_{t+1,j} = 1$, otherwise $S_{t+1,j} = 0$.
  end for
  $Z^Y_{t+1} \leftarrow$ GenerateZVector(S^C_{t+1})
end for

Postsimulation smoothing step:
$t \leftarrow p^{max}$
while $t < L$ **do**
  $S^{C,agg}_t = 1$ if $\sum_{j=1}^{J} F_j^{X|Y,-1}(\Phi(Z^Y_{t,j})) > 0$, $S^{C,agg}_t = 0$ otherwise.
  $r \leftarrow 0$
  while $S^{C,agg}_t == S^{C,agg}_{t+r}$ **do**
    $r \leftarrow r + 1$
    $S^{C,agg}_{t+r} = 1$ if $\sum_{j=1}^{J} F_j^{X|Y,-1}(\Phi(Z^Y_{t+r,j})) > 0$, $S^{C,agg}_{t+r} = 0$ otherwise.
  end while
  if $r < a$ **then**
    for $r' = 0, 1, \ldots, r-1$ **do**
      Repeat $Z^Y_{t+r'} \leftarrow$ GenerateZVector$(S^C_{t+r'})$ using the crossing states generated in the first pass of the simulation (before the smoothing step) to generate new errors until $S^{C,agg}_{t+r'} == S^{C,agg}_{t+r}$ or a maximum number of resampling iterations has been reached.
    end for
  end if
  $t \leftarrow t + r$
end while
return $Z^Y_{1:L,1:J}$

function GenerateZVector(S^C_{t+1})
  for $j = 1, 2, \ldots, J$ **do**
    $s \leftarrow S^C_{t+1,j}$
    $Z^Y_{t+1,j} = c^s_j + \sum_{i=1}^{p^s} A^s_{i,j} Z^Y_{t+1-i} + W_{t+1,j}$ where $W_{t+1,j} \sim \mathcal{N}(0, \sigma^{2,s}_j)$
    Set $Resamples = 0$
    while $(F_j^{X|Y,-1}(\Phi(Z^Y_{t+1,j})) > 0$ inconsistent with $s)$ **and** $(Resamples < MaxResamples)$ **do**
      Resample $W_{t+1,j} \sim \mathcal{N}(0, \sigma^{2,s}_j)$ in the linear model.
      $Resamples \leftarrow Resamples + 1$
    end while
    if $Resamples == MaxResamples$ **then**
      Sample $Z^Y_{t+1,j}$ from $F^{Z,s}_j$, the empirical distribution of forecast bin-conditioned Z-transformed errors occurring in subprocess j's current subprocess crossing state s.
    end if
  end for
  return Z^Y_{t+1}
end function

$$Q^2 = \frac{NM}{N+M} \int_{-\infty}^{\infty} \left[F^X(x) - F^Y(x)\right]^2 w(x) dF^{XY}(x),$$

where F^{XY} is the joint empirical CDF of the two samples combined. Note that two well-known statistics can be formulated in this manner. If $w(x) = 1$ this is the two-sample Cramer–von Mises statistic, and if $w(x) = [F^{XY}(x)(1 - F^{XY}(x))]^{-1}$ this is the two-sample Anderson–Darling statistic [2, 13].

We choose the weighting function $w(x) = |x|$. In the energy systems context, the characteristics of forecast error series most likely to induce blackouts are long (down-) crossing times combined with large (negative) forecast errors. In other applications these will result in shortages or unnecessary surpluses of inventory or resources if not accounted for. Thus, if a model produces simulated distributions that are slightly inaccurate at small errors and short crossing times, yet accurate in the tails of the distributions, we would favor it over a model that does the opposite. Therefore, $w(x) = |x|$ is an appropriate weight for our purposes as it places more emphasis on a model's ability to replicate the tails of forecast error and crossing time distributions.

For the remainder of the paper, when the Q^2 statistic is used, F^Y will be the observed empirical distribution from training data and F^X will be the corresponding simulated distribution generated by a model. Smaller Q^2 statistics indicate the model is doing a better job replicating a distribution. However, this statistic does not tell the complete story, and we will be analyzing the shapes of the distributions as well.

Finally, note that we present additional numerical results in the appendix, where we repeat all tests in this section on different groupings of PJM wind farms during a different month of the year (July 2013 rather than January 2013).

7.1. Univariate setting. Here we select a single wind farm in the data set for modeling and testing. Models involving an ARIMA model are chosen and fit by cross validating the results of the Box and Jenkins method and AIC minimization. ARIMA-GARCH models are selected via AIC minimization. Six models are tested:

1. an ARIMA(3,0,0) model, referred to as an AR(3) model,
2. an AR(3)-GARCH(1,1) model,
3. an AR(2) model for forecast bin-conditioned Z-transformed time series Z_t^Y,
4. an ARIMA-GARCH model for the Z_t^Y series,
5. the $J = 1$ dimensional version of the low dimensional multivariate model presented in section 5, referred to as the CS-AR (crossing state AR) model,
6. the crossing state HSMM from section 4 with parameters $Q = 3$ and $R = 5$.

To keep the plots uncluttered, the results for models 3 and 4 are only given in Table 1, in which we compare Q^2 distances between observed and simulated distributions. However, these Q^2 distances result from a different test case in which the same procedure was carried out for a different Great Plains wind farm in April 2013. Both the plots and Q^2 distances in different test cases support the fact that standard time series models consistently either overestimate or underestimate crossing times and do not model them as well as the crossing state models designed for that purpose.

Figure 5 shows the resulting error distributions for each model versus observed, while Figure 6 displays crossing time distributions. From these, we see that the AR(3)-GARCH(1,1)

Table 1

Q^2 distances between observed and simulated distributions for the models tested on a single wind farm power output. In general, the crossing state models are best at replicating the distributions.

Dist'n \ Model	AR	AR-GARCH	ARIMA-Z_t^Y	ARIMA-GARCH-Z_t^Y	CS-AR	HSMM
F^X	16.39	341.2	1.533	252.9	0.745	0.059
F^U	16.21	27.09	9.381	23.14	0.087	0.016
F^D	10.72	18.67	7.276	12.63	0.197	0.028
$F^{A,U}$	4485.3	24251	3209.2	20575	3409.2	278.66
$F^{A,D}$	10104	11535	6034.4	14943	1943.4	2134.4

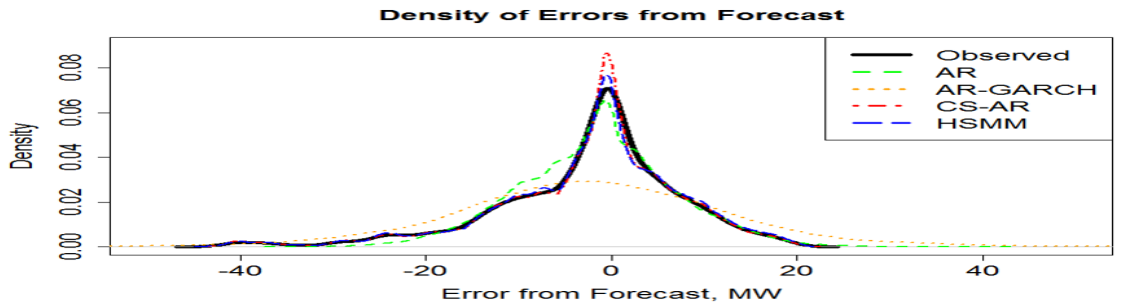

Figure 5. *Simulated univariate error distributions versus the observed distributions. The lines for the crossing state models may be difficult to see as they often overlap the observed distribution.*

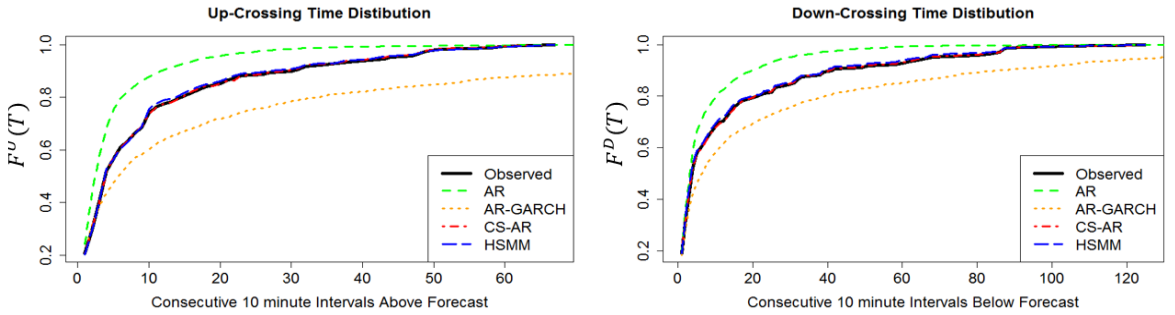

Figure 6. *Simulated univariate up- (left) and down- (right) crossing time CDFs versus the observed distributions. The lines for the crossing state models may be difficult to see as they often overlap the observed distribution.*

model does not replicate any distribution well and is not a strong model choice for our purposes. The AR(3) model does match the error distribution decently well, but note without the crossing state variable to guide transitions between error regimes, crossing times are generally shorter than in the observed data. Both crossing state models reproduce all three distributions well. This is seen in Table 1 as well.

In Table 1 we consider an additional measure of model effectiveness—the distributions of areas, above and below, between the actual output path and the forecast series. Reproducing

these distributions—$F^{A,U}$ for the areas above and $F^{A,D}$ for the areas below—well would indicate not only are we replicating how long the paths deviate from expected power output but also by how much during those periods, giving the total error in energy produced during these periods.

The models best matching both observed area distributions incorporate the crossing state into the model. However, none of these models has a technique devoted to directly controlling the distribution of areas above and below the forecast similar to the crossing state transition model for crossing time distributions. Thus, the quality of simulated area distributions is more variable than that of simulated crossing time distributions, and in other test cases area distributions do not match as well. This is a problem for future research to address.

7.2. Low dimensional multivariate setting. The training data and testing methods are similar to those performed in subsection 7.1, except that model wind power forecast errors from three randomly chosen wind farms from the Great Plains and view aggregate distributions as well as individual distributions. Additionally, a different set of models is tested, listed below:

1. a standard VAR model fit to a forecast bin-conditioned Z-variate time series,
2. the low dimensional crossing state (LD CS) model presented in section 5,
3. the high dimensional crossing state model (HD CS) with smoothing presented in section 6 with smoothing parameter $a = 3$.

Note only one standard time series model is tested for comparison. As many multivariate time series models are extensions of a univariate model to multiple dimensions, since we have observed in subsection 7.1 that standard univariate models do not preserve crossing times during simulation, we should not expect that the multivariate versions will do so at the disaggregate level. Effectiveness at the aggregate level is generally poor as well; however, this is not readily inferred through comparison with the univariate versions.

Here, the VAR model is the extension of a univariate AR model to more than one dimension (and conversely, a VAR model with $J = 1$ dimension is identical to an AR model). Of the standard univariate time series models, the AR model for the forecast bin-conditioned Z-transformed errors performed best according to Table 1, and thus its corresponding multivariate version is used for a benchmark in multivariate settings.

The VAR model is slightly better at replicating error distributions (see Table 2 and Figure 7). However, at a slight performance drop in replicating error distributions, the crossing state models realize significant improvements over the standard time series model in their ability to replicate crossing time distributions (see Table 2 and Figures 8 and 9). Furthermore, note the Q^2 distances for the distributions of areas above and below the forecast for the crossing state models (see Table 2) are small relative to the VAR model, especially at the aggregate level. This implies that the crossing state models are more accurate in modeling the total surpluses and deficits of available wind energy in the system compared to the expected amount.

Though the performance of the two crossing state models is very similar, the low dimensional model is still presented in this paper for the following reasons. First, it is a stepping stone to the high dimensional model as the high dimensional model is formed via approximations of components of the low dimensional model. It also generates sample paths much faster than the high dimensional model, requiring one-third of the CPU time. Finally, it is simpler

Table 2

Comparing Q^2 distances between observed and simulated distributions for the models tested on three wind farms at the individual and aggregate level. Due to variations in farm size and distribution type, only relative performance of models within each (distribution, farm) cell should be considered.

Wind farm	Model	F_j^X	F_j^U	F_j^D	$F_j^{A,U}$	$F_j^{A,D}$
1	VAR	15.0	11.5	23.9	10400	30200
	HD CS	23.1	1.14	2.07	12800	5200
	LD CS	17.0	0.78	1.32	3400	3600
2	VAR	24.5	13.2	13.0	11200	18700
	HD CS	45.0	0.33	8.86	4100	8700
	LD CS	30.7	0.21	1.89	1200	3200
3	VAR	38.3	9.53	12.9	18300	42500
	HD CS	36.6	3.13	8.84	50000	10700
	LD CS	14.1	1.30	5.19	29400	6700
Agg	VAR	71.6	7.01	10.9	16900	36900
	HD CS	98.8	0.66	2.93	26700	13900
	LD CS	88.9	1.91	0.79	23000	8700

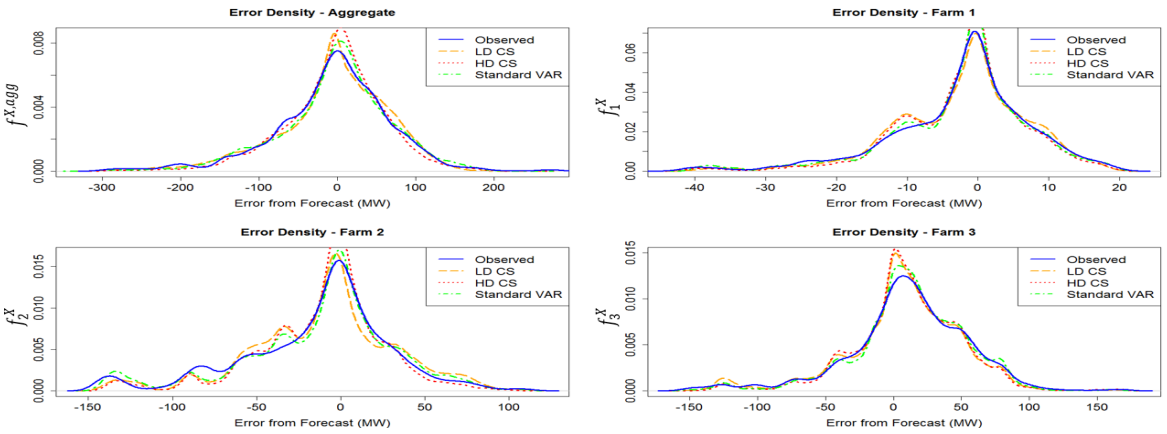

Figure 7. *Simulated error distributions versus the observed distributions at the aggregate level (top left) and the individual farms in the low dimensional setting.*

to use as it does not require the manual tuning of any parameters (as the high dimensional model does).

7.3. High dimensional multivariate setting. We generate 20 wind power scenarios from each of the following models, which have been appropriately fitted to the forecast error time series over the course of a month from 10 randomly chosen wind farms in the Great Plains region:

1. a standard VAR(p) model for the forecast bin-conditioned Z-variates,
2. the high dimensional crossing state model with no smoothing step (HD-NS),
3. the high dimensional crossing state model with the smoothing step (HD-S) and smoothing parameter $a = 4$.

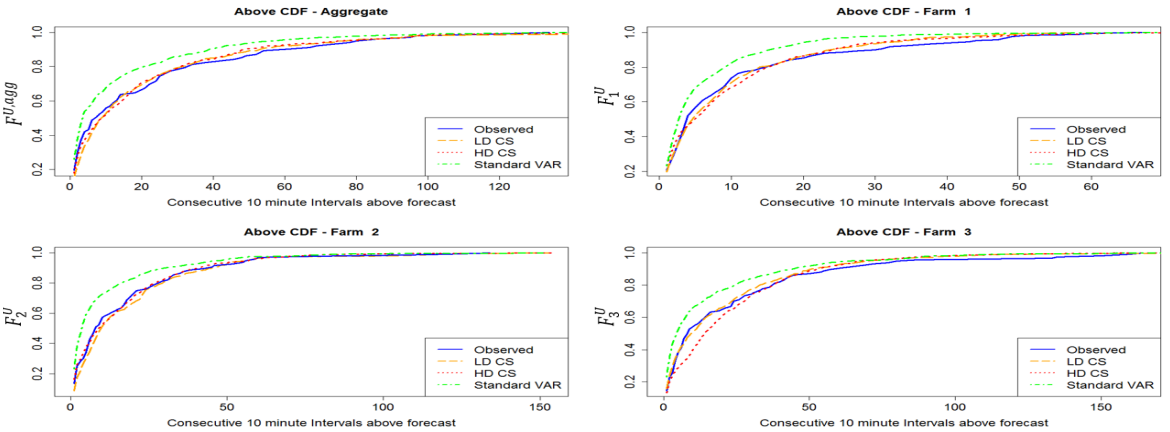

Figure 8. *Up-crossing simulated distributions versus the observed distributions at the aggregate level (top left) and the individual farms in the low dimensional setting.*

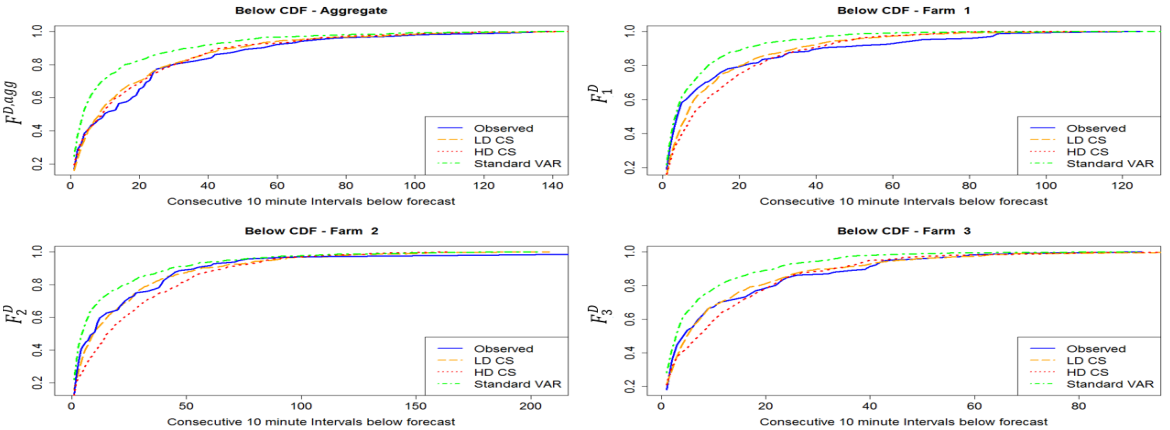

Figure 9. *Down-crossing simulated distributions versus the observed distributions at the aggregate level (top left) and the individual farms in the low dimensional setting.*

From Table 3 we see the error distribution is best replicated by the standard VAR model; however, the crossing state models perform better at replicating the remaining distributions most often. This is also seen in Figures 10, 11, and 12. The figures also display the effect of the smoothing step on aggregate crossing time distributions as the HD-S method corrects discrepancies between observed and simulated crossing time distributions compared to the HD-NS method. As a result, the distributions of areas above and below the forecast at the aggregate level are replicated better by the HD-S method as well.

8. Conclusion. This paper introduces crossing times as an important characteristic of stochastic processes involved in certain stochastic optimization problems. The two-level

Table 3

Comparing Q^2 distances between observed and simulated distributions for the models tested on 10 wind farms at the individual and aggregate levels. Due to variations in farm size and distribution type, only relative performance of models within each (distribution, farm) cell should be considered.

Dist'n	Model	Agg	1	2	3	4	5	6	7	8	9	10
F_j^X	VAR	625	162	20.9	10.3	23.4	3.00	3.98	10.7	3.62	53.5	32.3
	HD-NS	720	237	47.6	149	25.8	91.2	51.0	15.3	19.0	137	79.9
	HD-S	713	236	47.4	147	25.7	87.7	51.4	15.3	18.6	137	79.3
F_j^U	VAR	15.0	2.51	9.14	8.16	13.5	5.66	11.5	1.69	14.2	10.4	11.4
	HD-NS	11.7	2.62	2.37	1.22	2.66	6.16	2.07	1.98	1.85	3.04	2.20
	HD-S	.905	2.51	2.44	1.26	2.75	5.36	2.22	2.09	1.87	3.11	2.27
F_j^D	VAR	18.8	6.22	2.50	1.87	10.1	10.3	2.78	.695	7.63	6.04	8.75
	HD-NS	22.8	1.91	2.48	4.25	1.37	2.06	1.00	3.98	4.08	2.68	1.01
	HD-S	6.17	1.79	2.43	4.21	1.32	1.92	1.03	4.05	4.00	2.78	1.00
$F_j^{A,U}$ ($\times 10^3$)	VAR	196	7.10	5.65	17.3	59.8	6.68	8.95	.429	5.67	53.3	22.4
	HD-NS	135	16.3	2.21	4.58	1.90	32.9	2.55	1.28	1.47	38.6	11.8
	HD-S	39.8	16.3	2.24	4.62	1.96	29.7	2.75	1.29	1.44	36.3	11.8
$F_j^{A,D}$ ($\times 10^3$)	VAR	173	8.45	1.59	2.25	12.0	28.6	2.58	.177	8.59	33.7	13.2
	HD-NS	210	4.53	1.30	30.2	.969	8.46	1.23	1.28	1.44	19.2	2.56
	HD-S	33.2	4.40	1.35	29.1	.967	8.57	1.21	1.32	1.47	19.3	2.49

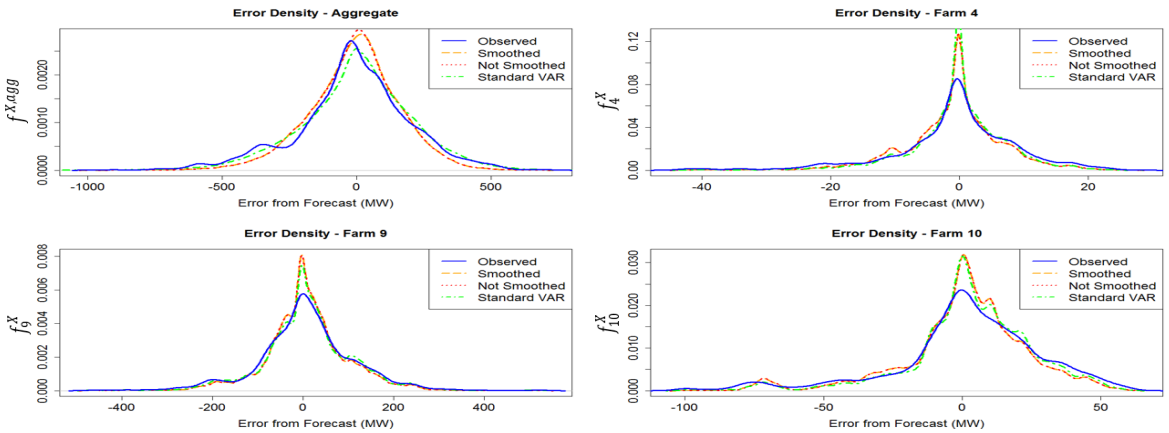

Figure 10. *Simulated error distributions versus the observed distributions at the aggregate level (top left) and three randomly chosen individual farms in the high dimensional setting.*

crossing state method of modeling and simulating is presented as a general approach to producing sample paths which outperform other time series methods in replicating crossing times. The first level crossing state transition model is a mechanism to control crossing times, while the second level error generation model conditioned on the crossing state should be chosen such that it is appropriate for the stochastic process of interest and reproduces error distributions accurately.

There are three different crossing state models we introduce in this paper. The hidden semi-Markov model in section 4 is tailored for producing sample paths of wind power forecast errors at individual wind farms. Using this method, areas between forecast and actual power,

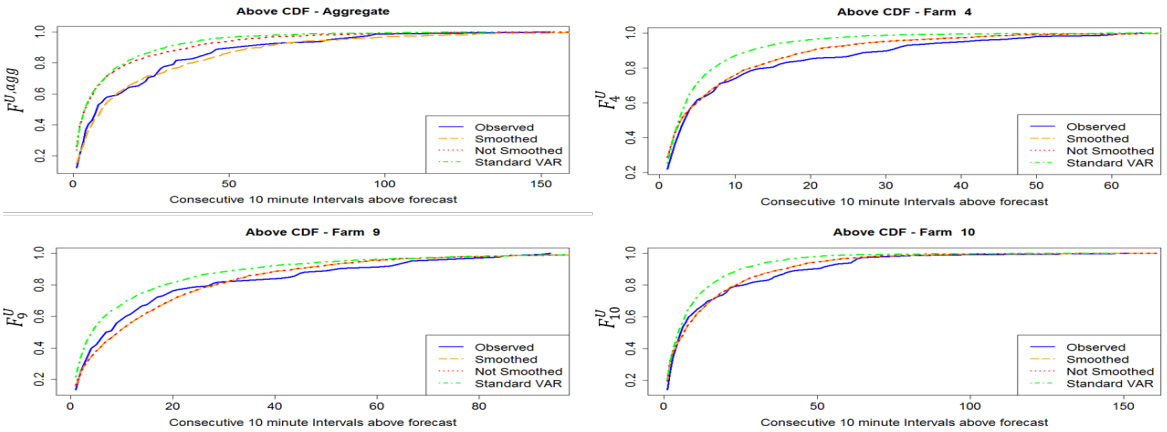

Figure 11. *Up-crossing simulated distributions versus the observed distributions at the aggregate level (top left) and three randomly chosen individual farms in the high dimensional setting.*

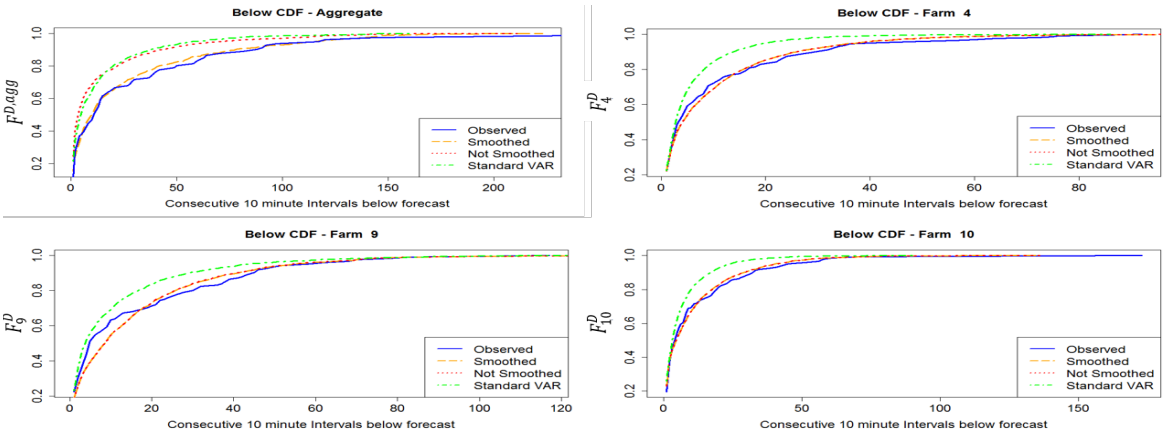

Figure 12. *Down-crossing simulated distributions versus the observed distributions at the aggregate level (top left) and three randomly chosen individual farms in the high dimensional setting.*

and thus the surpluses or deficits of energy produced versus expected, are also replicated well. Therefore, the intertemporal behaviors of wind power outputs which may lead to power outages if unaccounted for, such as their tendency to underperform expectations for extended periods of time, are reflected in the sample paths. Two multivariate methods are then introduced which replicate crossing time, forecast error, and, to a lesser extent, area distributions at both the aggregate and disaggregate levels. These are characteristics of power outputs from spatially distributed renewable power sources that are especially important in power system control. A low dimensional multivariate crossing state model is developed in section 5, while a high dimensional multivariate model is presented in section 6.

Finally, we note that it may be important to model the crossing time behavior of other linear functionals of a multidimensional stochastic process besides the aggregate sum. For

example, the crossing times of power outputs from groupings of wind farms in similar geographical locations may be just as important as the total aggregate wind power in a power grid with security and capacity constraints. Altering the models such that they also capture the crossing time behavior and error distribution of additional linear functionals, such as these sums at intermediate levels of aggregation, would thus be useful for certain applications. This seems like a natural extension of the low dimensional model from section 5, but doing so in high dimensions appears to be a more difficult challenge.

Appendix A. Additional numerical results. We repeat in Tables 4, 5, and 6 the numerical results from section 7 on different groupings of PJM wind farms during a different month of the year (July 2013 instead of January 2013). We observe similar results in terms of the relative ability of each model to replicate the desired distributions.

Table 4

Univariate setting. Q^2 distances between observed and simulated distributions for the models tested on a single wind farm power output. This table repeats the tests from Table 1 on a second wind farm using July data instead of January data.

Dist'n \ Model	AR	AR-GARCH	ARIMA-Z_t^Y	ARIMA-GARCH-Z_t^Y	CS-AR	HSMM
F^X	483.7	273.9	97.95	134.5	483.8	1.774
F^U	8.335	7.377	5.303	16.02	1.914	0.012
F^D	0.558	7.700	3.301	20.97	2.446	0.191
$F^{A,U}$	9409	5681	12470	24026	4896	1905
$F^{A,D}$	21377	3315	3221	19369	19329	2782

Table 5

Low dimensional multivariate setting. Comparing Q^2 distances between observed and simulated distributions for the models tested on three wind farms at the individual and aggregate levels. Due to variations in farm size and distribution type, only relative performance of models within each (distribution, farm) cell should be considered. This table repeats the tests from Table 2 on a second grouping of wind farms using July data instead of January data.

Wind farm	Model	F_j^X	F_j^U	F_j^D	$F_j^{A,U}$	$F_j^{A,D}$
1	VAR	1.79	8.17	5.63	3360	4400
	HD CS	7.09	0.69	3.90	720	1800
	LD CS	17.4	1.17	4.86	920	3100
2	VAR	5.65	11.4	9.19	11800	6700
	HD CS	36.0	0.70	4.88	2950	24500
	LD CS	112	2.52	1.90	2510	26500
3	VAR	46.1	22.2	9.57	55300	23900
	HD CS	48.5	0.89	11.0	6200	32800
	LD CS	192	6.73	1.45	3330	6800
Agg	VAR	61.3	8.90	15.4	40100	53600
	HD CS	114	1.07	1.43	6900	8600
	LD CS	195	3.29	1.08	20500	4800

Table 6

High dimensional multivariate setting. Comparing Q^2 distances between observed and simulated distributions for the models tested on 10 wind farms at the individual and aggregate levels. Due to variations in farm size and distribution type, only relative performance of models within each (distribution, farm) cell should be considered. This table repeats the tests from Table 3 on a second grouping of wind farms using July data instead of January data.

Dist'n	Model	Agg	1	2	3	4	5	6	7	8	9	10
F_j^X	VAR	131	3.79	12.2	4.04	3.12	28.5	3.34	0.25	4.26	37.1	3.02
	HD-NS	106	14.1	26.8	11.3	6.51	12.7	6.64	1.35	11.6	34.5	14.2
	HD-S	107	14.0	27.2	11.2	6.50	12.6	6.67	1.33	11.5	33.8	14.8
F_j^U	VAR	7.57	11.7	2.87	5.52	1.61	11.7	22.3	5.56	8.53	6.67	13.3
	HD-NS	10.8	1.04	2.13	0.60	2.16	1.33	2.93	4.21	1.02	3.18	0.90
	HD-S	1.06	1.05	2.21	0.61	2.14	1.37	2.99	4.06	0.93	3.01	0.89
F_j^D	VAR	13.7	28.2	7.70	14.6	1.97	17.5	15.6	4.51	15.0	12.4	13.9
	HD-NS	11.7	0.95	10.6	6.54	6.34	8.48	5.44	5.67	3.81	14.3	1.69
	HD-S	2.09	0.94	10.5	6.10	6.06	8.60	5.35	5.43	3.26	13.6	1.68
$F_j^{A,U}$ ($\times 10^3$)	VAR	28.8	9.25	0.69	1.09	0.13	12.8	9.67	0.23	2.74	12.4	9.09
	HD-NS	36.3	7.34	4.35	1.68	0.56	2.90	0.30	0.63	1.49	69.7	1.34
	HD-S	6.22	7.28	4.38	1.57	0.54	2.66	0.29	0.63	1.56	64.9	1.38
$F_j^{A,D}$ ($\times 10^3$)	VAR	38.9	28.7	5.69	11.7	0.51	33.6	7.02	0.30	7.26	37.0	7.36
	HD-NS	34.8	4.22	12.1	0.99	0.51	10.3	1.84	0.50	1.11	28.9	1.97
	HD-S	16.9	4.16	12.2	0.93	0.49	10.4	1.82	0.51	1.15	28.3	1.97

REFERENCES

[1] H. Akaike, *A new look at the statistical model identification*, IEEE Trans. Automat. Control, 19 (1974), pp. 716–723.

[2] T. W. Anderson, *On the distribution of the two-sample Cramer-Von Mises criterion*, Ann. of Math. Stat., 33 (1962), pp. 1148–1159.

[3] C. Archer, H. Simão, W. Kempton, W. Powell, and M. Dvorak, *The challenge of integrating offshore wind power in the US electric grid. Part* I: *Wind forecast error*, Renewable Energy, 103 (2017), pp. 346–360.

[4] R. Barth, L. Söder, C. Weber, H. Brand, and D. J. Swider, *Methodology of the Scenario Tree Tool*, Tech. report, Wilmar International, 2006.

[5] G. Bayraksan and D. P. Morton, *Assessing solution quality in stochastic programs via sampling*, in Decision Technologies and Applications: INFORMS, 2009, pp. 102–122.

[6] I. Blake and W. Lindsey, *Level-crossing problems for random processes*, IEEE Trans. Inform. Theory, 19 (1973), pp. 295–315.

[7] T. Bollerslev, *Generalized autoregressive conditional heteroskedasticity*, J. Econometrics, 31 (1986), pp. 307–327.

[8] G. E. Box and G. M. Jenkins, *Time Series Models for Forecasting and Control*, Holden-Day, San Francisco, 1970.

[9] M. Brubaker, M. Salzmann, and R. Urtasun, *A family of MCMC methods on implicitly defined manifolds*, in Proceedings of the Artificial Intelligence and Statistics Conference, 2012, pp. 161–172.

[10] M. C. Cario and B. L. Nelson, *Modeling and Generating Random Vectors with Arbitrary Marginal Distributions and Correlation Matrix*, Tech. report, 1997.

[11] E. M. Constantinescu, V. M. Zavala, M. Rocklin, S. Lee, and M. Anitescu, *A computational framework for uncertainty quantification and stochastic optimization in unit commitment with wind power generation*, IEEE Trans. Power Systems, 26 (2011), pp. 431–441.

[12] P. Correia and J. F. de Jesus, *Simulation of correlated wind speed and power variates in wind parks*, Electric Power Systems Research, 80 (2010), pp. 592–598, https://doi.org/10.1016/j.epsr.2009.10.031.
[13] D. A. Darling, *The Kolmogorov-Smirnov, Cramer-Von Mises tests*, Ann. Math. Stat., 28 (1957), pp. 823–838.
[14] G. Díaz, J. Gómez-Aleixandre, and J. Coto, *Wind power scenario generation through state-space specifications for uncertainty analysis of wind power plants*, Appl. Energy, 162 (2016), pp. 21–30.
[15] J. Dupačová, N. Gröwe-Kuska, and W. Römisch, *Scenario reduction in stochastic programming*, Math. Program., 95 (2003), pp. 493–511.
[16] R. F. Engle, *Autoregressive conditional heteroscedasticity with estimates of the variance of United Kingdom inflation*, Econometrica, 50 (1982), pp. 987–1007.
[17] Y. Feng, D. Gade, S. M. Ryan, J.-P. Watson, R. J.-B. Wets, and D. L. Woodruff, *A new approximation method for generating day-ahead load scenarios*, in Proceedings of the Power and Energy Society General Meeting, IEEE, 2013, pp. 1–5.
[18] J. D. Ferguson, *Variable duration models for speech*, in Proceedings of the Symposium on the Application of HMMs to Text and Speech, 1980, pp. 143–179.
[19] S. Golchi and D. A. Campbell, *Sequentially constrained Monte Carlo*, Comput. Statist. Data Anal., 97 (2016), pp. 98–113.
[20] N. Growe-Kuska, H. Heitsch, and W. Romisch, *Scenario reduction and scenario tree construction for power management problems*, in Power Tech Conference Proceedings, Bologna, Vol. 3, IEEE, 2003.
[21] S. Hagspiel, A. Papaemannouil, M. Schmid, and G. Andersson, *Copula-based modeling of stochastic wind power in Europe and implications for the swiss power grid*, Appl. Energy, 96 (2012), pp. 33–44.
[22] J. D. Hamilton, *A new approach to the economic analysis of nonstationary time series and the business cycle*, Econometrica, 57 (1989), pp. 357–384.
[23] H. Heitsch and W. Römisch, *Scenario tree modeling for multistage stochastic programs*, Math. Program., 118 (2009), pp. 371–406.
[24] D. C. Hill, D. McMillan, K. R. W. Bell, and D. Infield, *Application of auto-regressive models to UK wind speed data for power system impact studies*, IEEE Trans. Sustainable Energy, 3 (2012), pp. 134–141, https://doi.org/10.1109/TSTE.2011.2163324.
[25] S. Jin, S. M. Ryan, J.-P. Watson, and D. L. Woodruff, *Modeling and solving a large-scale generation expansion planning problem under uncertainty*, Energy Systems, 2 (2011), pp. 209–242.
[26] A. Lojowska, D. Kurowicka, G. Papaefthymiou, and L. van der Sluis, *Advantages of Arma-Garch wind speed time series modeling*, in 11th International Conference on Probabilistic Methods Applied to Power Systems, IEEE, 2010, pp. 83–88.
[27] X.-Y. Ma, Y.-Z. Sun, and H.-L. Fang, *Scenario generation of wind power based on statistical uncertainty and variability*, IEEE Trans. Sustainable Energy, 4 (2013), pp. 894–904.
[28] J. M. Morales, R. Minguez, and A. J. Conejo, *A methodology to generate statistically dependent wind speed scenarios*, Appl. Energy, 87 (2010), pp. 843–855.
[29] G. Papaefthymiou and P. Pinson, *Modeling of spatial dependence in wind power forecast uncertainty*, in Proceedings of the 10th International Conference on Probabilistic Methods Applied to Power Systems, IEEE, 2008, pp. 1–9.
[30] A. Papavasiliou and S. S. Oren, *Multiarea stochastic unit commitment for high wind penetration in a transmission constrained network*, Oper. Res., 61 (2013), pp. 578–592.
[31] A. Papavasiliou, S. S. Oren, and R. P. O'Neill, *Reserve requirements for wind power integration: A scenario-based stochastic programming framework*, IEEE Trans. Power Systems, 26 (2011), pp. 2197–2206.
[32] M. V. Pereira and L. M. Pinto, *Multi-stage stochastic optimization applied to energy planning*, Math. Program., 52 (1991), pp. 359–375.
[33] A. B. Philpott and V. L. De Matos, *Dynamic sampling algorithms for multi-stage stochastic programs with risk aversion*, European J. Oper. Res., 218 (2012), pp. 470–483.
[34] P. Pinson, H. Madsen, H. A. Nielsen, G. Papaefthymiou, and B. Klöckl, *From probabilistic forecasts to statistical scenarios of short-term wind power production*, Wind Energy, 12 (2009), pp. 51–62.
[35] A. Shapiro, D. Dentcheva, et al., *Lectures on Stochastic Programming: Modeling and Theory*, MOS/SIAM Ser. Optim. 16, SIAM, Philadelphia, 2014.

[36] A. Shapiro, W. Tekaya, J. P. da Costa, and M. P. Soares, *Risk neutral and risk averse stochastic dual dynamic programming method*, European J. Oper. Res., 224 (2013), pp. 375–391.
[37] K. C. Sharma, P. Jain, and R. Bhakar, *Wind power scenario generation and reduction in stochastic programming framework*, Electric Power Components Systems, 41 (2013), pp. 271–285.
[38] L. Soder, *Simulation of wind speed forecast errors for operation planning of multiarea power systems*, in Proceedings of the International Conference on Probabilistic Methods Applied to Power Systems, IEEE, 2004, pp. 723–728.
[39] S. Takriti, J. R. Birge, and E. Long, *A stochastic model for the unit commitment problem*, IEEE Trans. Power Systems, 11 (1996), pp. 1497–1508.
[40] J. Tastu, P. Pinson, E. Kotwa, H. Madsen, and H. A. Nielsen, *Spatio-temporal analysis and modeling of short-term wind power forecast errors*, Wind Energy, 14 (2011), pp. 43–60.
[41] S. I. Vagropoulos, E. G. Kardakos, C. K. Simoglou, A. G. Bakirtzis, and J. P. Catalão, *ANN-based scenario generation methodology for stochastic variables of electric power systems*, Electric Power Systems Research, 134 (2016), pp. 9–18.
[42] S.-Z. Yu, *Hidden semi-Markov models*, Artificial Intelligence, 174 (2010), pp. 215–243.
[43] N. Zhang, C. Kang, Q. Xia, and J. Liang, *Modeling conditional forecast error for wind power in generation scheduling*, IEEE Trans. Power Systems, 29 (2014), pp. 1316–1324.