COMBINING OF RENEWABLE ENERGY PLANTS TO IMPROVE ENERGY PRODUCTION STABILITY

by

Adam C. Broders

A Thesis
Submitted to the Faculty
of the
WORCESTER POLYTECHNIC INSTITUTE
in partial fulfillment of the requirements for the
Degree of Master of Science
in
Electrical and Computer Engineering
by

____

May 2008

APPROVED:

____

Professor Alexander E. Emanuel, Major Advisor

____

Professor Alexander M. Wyglinski

____

Professor Yiming Rong

Abstract

This thesis details potential design improvements by exploiting a new general grid model utilizing multiple wind and solar energy plants. A single renewable energy plant which relies on wind speed or solar insolation is unreliable because of the stochastic nature of weather patterns. To allow such a plant to match the requirements of a variable load some form of energy storage must be incorporated. To ensure a low loss of load expectation (LOLE) the size of this energy storage must be large to cope with the strong fluctuations in energy production. It is theorized that by using multiple renewable energy plants in separate areas of a region, the different weather conditions might approach a probabilistically independent relationship. The probability of energy generated from combined plants will then approach a Gaussian distribution by the central limit theorem. While maintaining the same LOLE as a single renewable plant this geographic separation model theoretically stabilizes the energy production and reduces the system variables: energy storage size, energy storage efficiency, and cumulative plant capacity. New generic weather models that incorporate levels of independence are created for wind speeds and solar insolations at different locations to support the geographic separation model. As the number of geographically separated plants increases and the weather approaches independence the system variables are reduced.

Acknowledgements

There are many people that helped me with this masters thesis. Most notably there are a few people that stand out in this regard. My parents, Chris and Bonnie Broders, helped me in so many ways that I cannot even describe the gratitude due. Professor Alexander Emanuel also deserves an amazing thank you as my thesis adviser. He does not know now and probably will never know how much he helped me on my thesis and my outlook on life. In addition I extend my gratitude towards the Professors Alexander Wyglinski (ECE) and Yiming Rong (ME) for being on my thesis committee. For help with editing and always keeping me in a positive attitude I would like to thank Jessica Rosewitz. Finally the institution of WPI allowed for me to attend graduate school and to do this thesis and as such I think every employee and past employee of WPI is in need of a large all encompassing thank you.

Contents

List of Figures

List of Tables

List of Terms

Auto-regressive and moving average (ARMA) - A mathematical method used to characterize some random processes such as wind speed and solar insolation.

Conventional power plant - A controllable energy generation plant such as a coal plant, nuclear plant, or natural gas powered plant.

Dependence - The probability that all geographic locations will have the same wind speed or solar insolation for a given day.

Diesel generator (DG) - Electric generator which converts diesel fuel to electric energy.

Forced outage rate (FOR) - The probability that a particular power plant will fail to operate and electric energy will not be produced.

Load Requirement - The energy required to serve the model's load.

Loss of load (LOL) - Number of hours per year the load requirement was not met.

Loss of load expectation (LOLE) - Expected number of hours per year that a load requirement will not be met.

Penetration level - Percent renewable power generation capacity as compared to total power generation capacity on a grid.

Photovoltaic (PV) - Silicon component design intended to convert sunlight to electricity.

Probability distribution function (PDF) - A function which describes the probability of various numerical occurances.

Renewable system - A grid that utilizes a renewable source or renewable sources such as wind and solar.

Solar insolation - Power available per meter squared from sunlight.

Solar radiation - Energy available per meter squared from sunlight.

Wind turbine generator (WTG) - Electric generator and associated equipment which converts wind energy to electricity.

List of Symbols

D_c		Day of interest in window method
D_m		Probability of D between two geographic locations, See application
D_w		Day PDF is being generated for.
D		Probability of having identical weather for all geographic locations
$E_s(T)$	Wh	Energy storage charge at time T
E_{gt}	Wh	Energy generated during fifteen minute increment t
E_{pwt}	$\frac{Wh}{W}$	Energy per watt of production capacity during time t
E_s		Overall energy storage efficiency
E		Efficiency
$F_c()$		PDF for ith plant's weather
H_i	h	ith loss of load in hours
I_j		jth data element acquired from the window method
i	$\frac{Wh}{m^2}$	Random variable indicating solar radiation
L_{ath}		Average load during hour th of the day
L_{nt}		Normalized load during time interval t
L_t	Wh	Load energy requirement during fifteen minute increment t
LM_t	Wh	Load mismatch during fifteen minute increment t
LOL_i	h	Loss of load from ith Monte Carlo simulation
LOL_{sT}	h	Hours of LOL condition during time interval T
M_{PV}	W	Solar array size in watts.

M		Number of Monte Carlo simulations
n_x		Number of plants in situation X
n		Number of plants
N		Number of data samples
PC_x	W	Production capacity of each X plant
PV_r	$\frac{W}{m^2}$	Photovoltaic constant representing production per unit of panel surface area
S_{5i}	Wh	ith 5-minute increment for solar radiation on a given day
S_c		Solar radiation for a given day
S_m	Wh	Maximum solar radiation for a given day
S_{rt}	$\frac{Wh}{m^2}$	Average solar radiation during time t
S_t		Incident insolation on solar arrays
$S1$, $S2$, $S3$		Different random weather conditions
t_s	h	Time separation of time samples in hours
T_t		Temperature
T		End of time span t, measured in t units
W_j	$\frac{m}{s}$	jth wind speed attained from window method
W	d	Window size in days
$\delta()$		Dirac delta function

Chapter 1

Introduction

The intent of this thesis is to recognize a new model of geographic separation which improves the stability of wind and solar plants. These renewable energy plants produce electricity with a sporadic behavior that has made them untrustworthy on the grid as a major contributor. The inclusion of energy storage is a solution to this problem, however the amount of energy storage required is typically enormous representing an immense implementation expense. Lacking enough energy storage a renewable plant will often not match its load resulting in an undesired grid condition.

The geographic separation model improves the stability of the renewable energy plants by physically placing a large distance between many plant sites to attain different weather conditions. If it is assumed that the weather at the various locations is probabilistically independent, then the central limit theorem states that as the number of weather-independent geographic locations increases the probability distribution function (PDF) for the generated energy of all the renewable energy plants combined should approach a normal distribution. It is expected that this will increase the grid stability of the system without incurring a significantly larger equipment cost.

Independent weather patterns are unlikely so an additional weather model is presented to vary the probability of having identical or independent weather. As the probability of independent weather is increased the likelihood of matching the energy generation to the load shows what levels of geographic separation are required to achieve worthwhile results given the number of geographic locations.

Although the overall grid model is not a proof that geographic separation will always stabilize renewable energy production this does show that it is likely that one can stabilize the energy production. In addition the same stability should be achievable with a smaller storage size, lower storage efficiency, and smaller cumulative production capacity.

Within this thesis the the geographic separation model and the tiered implementation method are novel contributions to the field of renewable energy generation.

Chapter 2

The Geographic Separation Model

In the process of designing a new renewable energy production system model, the geographic separation model, that improves energy production stability a few steps were taken. A new model that included multiple renewable energy plants had to be designed. Past models were reviewed to understand their strengths and weaknesses. These were then used to advance to a new model design that better describe a generic renewable system with multiple renewable plants. In order to better evaluate the stability of the renewable energy and test the effect of this new model on energy storage all conventional generators were removed to allow for complete renewable production penetration.

2.1 Typical Models

In order to mathematically evaluate a renewable system a particular model must be used as an accurate representation. When renewable energy is considered two different forms of models are typically chosen depending on what kind of grid the system is to be connected to. These two types of models are often used based on the isolation from a main grid.

When electric energy is supplied far from a local grid renewable energy is a viable option that is often considered. The connectivity of such a system includes one or more forms of renewable energy generation and sometimes a level loading unit to provide electricity when the renewable energy is not available. A diagram of this model can be seen in figure 2.1 which includes a photovoltaic source (PV), wind turbine generator (WTG), and a diesel generator (DG). The level loading unit in the figure represents a diesel generator however other forms of generators and energy storage can be used instead.

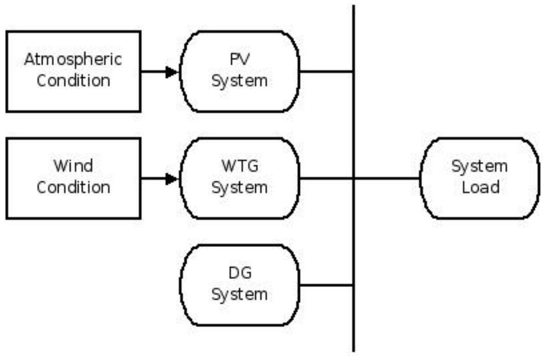

Figure 2.1: Typical model of small isolated grid utilizing renewable energy sources [1].

The other typical model was used when a connection to a large grid was available. Most simulations associated with these models were used to test the stability of the grid as stochastic renewable energy penetration levels increase [7]. These studies also have proposed methods of increasing the useful production capacity by performing basic modifications to the grid's model. Some studies in particular [8] have determined how effective a certain plant would be if none of the excess energy was used and in this case how high of a production capacity is required to reach certain penetration levels. A basic model representing such a grid connected configuration can be seen in figure 2.2.

Both of these models assume a perfect grid. The main focus is on energy production and demand on a generalized grid so propagation and conversion losses are ignored as well as any phasor analysis. This simplifies the study so that approximate stability limits can be determined for a region's weather patterns without concerns for the specific grids that share that weather pattern. If a specific transmission design had been used in the model then other layouts may not apply. Even moving a renewable plant to another point in the network could possibly invalidate the study.

It can be seen from figure 2.2 that the classical model is described by a few basic interacting components: What does the load require, what can the renewable plants generate based on the wind, and what can the traditional power plants provide. When using Monte

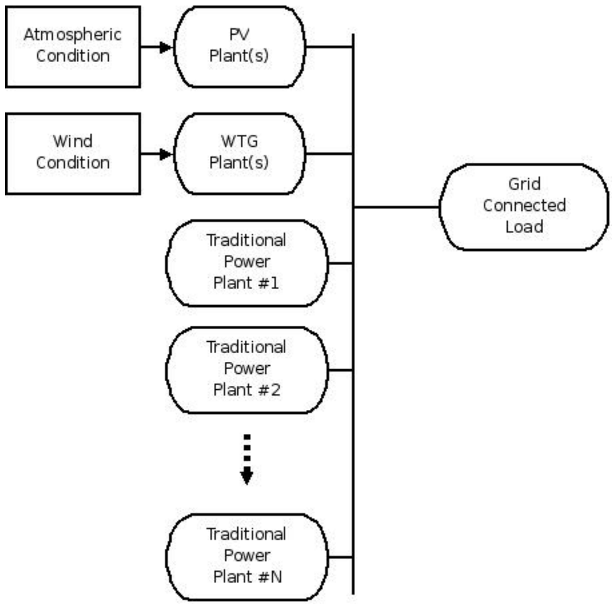

Figure 2.2: Typical model of local grid utilizing renewable energy sources [1].

Carlo simulations the load will sometimes be higher than what all the generating units can provide for a certain number of hours per year. The hours per year expectation over multiple simulations is the LOLE. This is one of the standard methods to measure the stability of the grid and is explained in more detail later. If two sets of simulations are compared the system with the lower LOLE is said to be more stable than the other.

The two most common ways to determine a LOLE when wind plants are included in the model is to use a multistate wind farm representation or a Monte Carlo simulation [3]. Solar power is typically modeled using a very similar Monte Carlo simulation such as in [2] or using a monthly averaging method [8]. Traditional power plants will often use another method similar to the multistate wind farm assuming two states defined by the forced outage rate (FOR).

The FOR is used as a probability of failure indicating either full production capacity or no production capacity [3]. The FOR model has been extended to include wind power [3] and is referred to as the multistate WTG model. This model allows for multiple states of power production based on typical weather patterns and WTG failure rates. Depending on how forced outage rate is represented and the intention of the model there are various methods of finding a LOLE. Aleksandar Dimitrovski and Kevin Tomsovic's work [7] discusses many of these methods. FOR begins to deteriorate when the power output from one point in time is not independent of all others as with systems dependent on weather. Systems that involve a time series model or a model which depends on previous events are not accurately described by a FOR but can be used as a rough estimate.

When wind or solar power is added to a system the more advanced methods of modeling often involve some form of time or event based power production estimate. Auto-regressive and moving average (ARMA) models are the typical method by which the weather is modeled. These models are very specific to a particular location and use a large amount of data gathered over many years to generate. Once the model has created chronological weather for multiple Monte Carlo simulations, the average of the results is one of the most accurate outcomes one can attain. For certain locations the wind model can be estimated using a 6-step FOR model which has been shown to compare to these more advanced simulation methods [3]. When solar weather is not generated using an ARMA model an averaged monthly or yearly expectation is often used. An example of these averages can be seen in the results of [8].

Energy storage poses other interesting problems as well. Although multistate models are used in this thesis for wind and solar weather predictions the normal mathematical procedures to determine LOLE based on FOR cannot be used. Even though the generated energy and daily load is independent of other day's daily power output in this model the current charge of the energy storage is not. For this reason a Monte Carlo simulation is used in order to preserve the energy storage's state over time.

From any one of these forms of simulation there must be a method of determining the LOLE. The equation used to determine the Loss of Load (LOL) from a discrete Monte Carlo simulation can be seen in equation 2.1,

$$LOL = \frac{8760 * \sum_{i=1}^{N} H_i}{N * t_s} \tag{2.1}$$

where 8760 is the number of hours in one non-leap year, N is the number of data samples, H_i is the loss of load in hours for the ith data sample, and t_s is the time separation between time samples in hours. This equation adds up the number of hours the available energy could not match the load and normalizes outcome over one year.

A Monte Carlo simulation must be run numerous times to determine a probability distribution function (PDF) of the likely LOL conditions. Various aspects of the grid can then be inspected such as the expected value, also known as the LOLE, and the standard deviation. The standard deviation is often ignored if it is small compared to the LOLE. Equation 2.2 is used to determine the LOLE from numerous Monte Carlo simulation LOL's,

$$LOLE = \frac{\sum_{i=1}^{M} LOL_i}{M} \tag{2.2}$$

where M is the number of Monte Carlo simulations and LOL_i is the LOL from the ith Monte Carlo simulation. This equation is derived from the LOLE equation found in [1].

A lower LOLE is typically viewed as better however there is sometimes more to consider. One example is a grid tolerable to short duration blackouts. Two simulations showing the same LOLE could have drastically different real world results. If one simulation often fails but for very short durations of time this will tend to not affect this particular load. The other simulation may have the same LOLE however will have a very unlikely case with a loss of load condition that lasts for a long duration of time. This would have a significant effect on this particular load as compared to the many short blackouts. With specific loads this is something important to consider and may warrant a modification of the LOLE or the model. Healthy state probability and loss of health expectation from [1], expected unserved energy from [7], and ramp rate from [9] are all examples of various evaluation methods specific to a type of study. In this paper generalized loads are used so LOLE is used unchanged as the measure of stability.

Improving on renewable energy in order to attain a better LOLE has been attempted through various means. One of the methods stated as the ultimate solution is energy storage [10]. The issue with energy storage is the sheer Wh needed to stabilize any large portion of the grid when penetration levels begin to affect the overall grid stability. In order to decrease the need for energy storage with large renewable generating capacities the renewable energy plants must stabilize their energy production first.

One typical method to improve on a random signals predictability is to use the central limit theorem and use multiple independent sources. Unfortunately weather conditions do not allow a guarantee of independence. Regardless there is a possibility for improvement over one power plant by increasing the number of plants and distributing them geographically in an attempt to have somewhat independent weather characteristics. At this point the question becomes how many plants are needed and how unrelated do those plants need to be in order to see an improvement in the stability of the system.

2.2 Improvement Through Number of Plants

By distributing power generation to multiple independent renewable energy plants the power output may eventually become normalized and somewhat predictable depending on how independent the weather is between the locations. The number of plants that requires this is highly dependent on the shape of the PDF that represents the power output from the renewable source. Additionally every day of the year the weather characteristics would be different so this also depends on all the various distributions over the entire year. To ensure that the power output is reliable through the entire year each day's power generation PDF would have to be sampled and summed repeatedly in order to generate an adequately normalized power production expectation. The maximum number of times a distribution was summed to become normalized within certain decided limits would imply the number of geographic locations required for such stability. This will most likely reveal an extreme situation in which every day must be highly predictable requiring an unrealistic number of independent renewable power plants. Instead of inspecting every day's PDF a Monte Carlo simulation should indicate when the addition of plants no longer has a significant effect on the system's overall operation regardless of each day's independence.

2.3 Improvement Through Geographic Separation

The reality of weather systems dictates that no matter how far a renewable energy plant is separated from another they will have some form of dependence. It is rather obvious that a distant enough separation will be highly independent, such as an installation in New England compared to an installation in Australia, however a global wide event such as a large volcanic eruption will no doubt affect solar power output in both regions.

If the number of power plants is limited the dependence now dictates the improvement in the renewable system. One distinct method of analyzing such a set of weather patterns would be the correlation. The issue with correlation is reversing it so that random weather can be generated. Based on a PDF and a value of correlation there can be infinite solutions. In contrast if the correct specific joint PDF is determined this possibly prevents the simulation from being generalized to other locations.

For this reason a notion of dependence (D) has been created. Dependence in this thesis refers to a number between 0 and 1 symbolizing the probability that all the plants received the same weather on any given day. This implies that the probability of the plants not receiving the same weather is simply $1 - D$.

Determining dependence for multiple locations is not a matter of correlation. This has to do with how likely it will be for two or more plants to be the same. By observing weather fronts it can be seen that one plant's weather may be vastly different or the same as another depending on the shape and movement of a front. In an attempt to equalize the system for a given dependence each plant was given a probability of being equal to the next. This allows D to define all the probabilities associated with the individual sites.

2.4 Execution Methods

The design of a mathematically sound method to determine the dependencies amongst the various plants is critical to how the full model will behave. Two methods were devised both with desired and undesired characteristics. The first method involved total dependence or independence by a probable chance every day while the second method used a tiered system to independently decide each plant's relative dependence. Two tiered systems were devised in order to meet desired system characteristics.

This first model that relies on either complete independence or dependence for a given day is rather simple. The dependence given as D determined exactly how likely the plants were to be identical to one another. This provides two outcomes for the system. For any given day S_1 would be the case when the weather conditions were identical while S_2 would imply that all the individual site's power production should be calculated independently. Mathematically this can be represented by Equations 2.3 and 2.4.

$$P[S_1] = D \tag{2.3}$$

$$P[S_2] = 1 - D \tag{2.4}$$

When the weather for this mathematical representation is observed two possible situations exist. Either all of the geographic sites would receive the same weather or their weather would be decided independently. Unfortunately this model does not support more than one set of identical weather conditions spread between many plants. This is assumed to occur as weather fronts pass over a region that the plants occupy.

The second method was then developed to improve on the first method. The tiered scenario allows the plants to generate electricity quantized by the number of weather patterns, the weather condition for each weather pattern, and the number of plants affected by each weather pattern. The purpose is to keep the probability of two plants having the same weather the same for any two plants attached to the simulated grid. From a perspective of two plants the joint PDF will have a high likelihood of remaining on the Y=X line if D is high. An example of a two plant joint PDF assuming a uniform distribution for $D = 1$ can be seen in figure 2.3.

Increasing the number of plants to three and preserving the same principles seen with two plants generates a joint PDF as in figure 2.4. The darker the shading on the various surfaces inside the volume the more likely it is that that particular event will occur. This is an approximated graph and the shading is not exact in order to clarify the method used when moving from two plants to three.

Although this model is relatively basic, issues arise with its software implementation. For up to three locations it may be practical to generate a joint PDF however as more locations are added the computational complexity increases a great deal. For this reason a process was devised to imitate the PDF representations shown in figures 2.3 and 2.4.

For simplicity it was decided that some method of determining one plant's weather at a time would be necessary. Initially a method was devised that exactly modeled figure 2.4. Assuming $D = .25$ a diagram of the implementation model can be seen in figure 2.5. Each

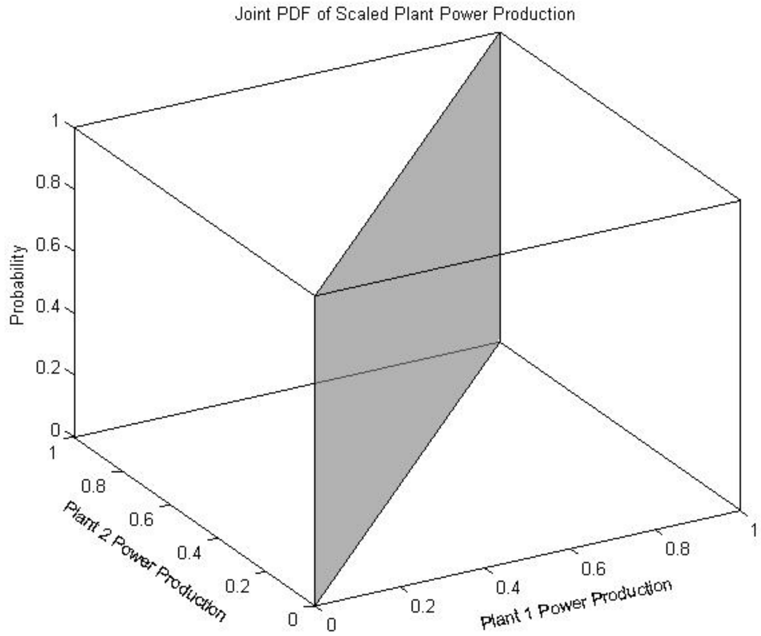

Figure 2.3: Joint PDF of two plants assuming $D = 1$ and assuming both uniformly distributed.

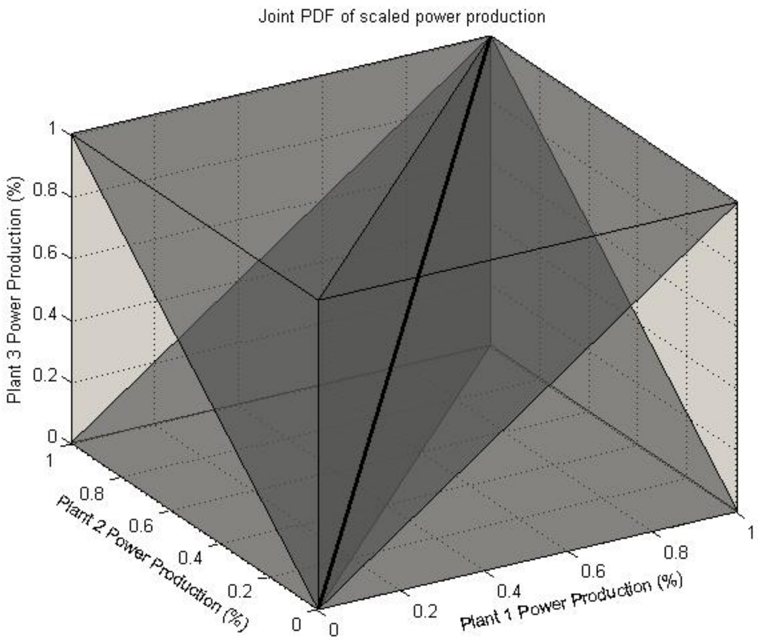

Figure 2.4: Joint PDF of three plants assuming $D \approx .5$ and assuming all uniformly distributed.

tier to the model represents the possible weather conditions for a particular plant and S1-S3 are the weather conditions for a given day that the renewable plants are experiencing. The first plant is decided based on its pure independent weather PDF. The second plant has a probability D_m of having the same conditions as the first and a probability of $1 - D_m$ of having different weather. Following this, the third plant has a D_m probability of having similar weather to the first plant, D_m probability of being like the second, or $1 - 2 * D_m$ probability of having a different weather condition. D_m in this case is set at .3536 to force $D = .25$.

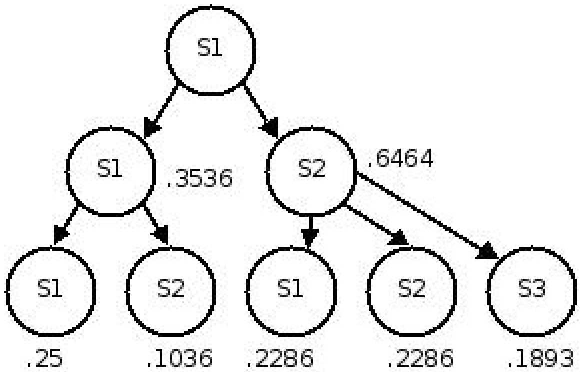

Figure 2.5: First probabilistic implementation model with D=0.25.

As much as the model proposed was designed directly from the diagram in figure 2.4 there was a caveat: it was determined that when $D = 1$ the probability of every plant's weather being the same was not 1. For this reason a model slightly different from figures 2.3 and 2.4 was used that still follows the same principles. This new model which was used for both solar and wind weather generation can be seen with a three-tier configuration representing the separate power plant's weather in figure 2.6.

D_m takes on a new role as the probability that a second or later tier will be modeled after a previous tier. Using equation 2.5 where n is the number of plants $D_m = 0.5$ when $D = 0.25$ and $n = 3$. The following paragraph describes the process in figure 2.6 with more detail.

$$D_m = \sqrt[n-1]{D} \tag{2.5}$$

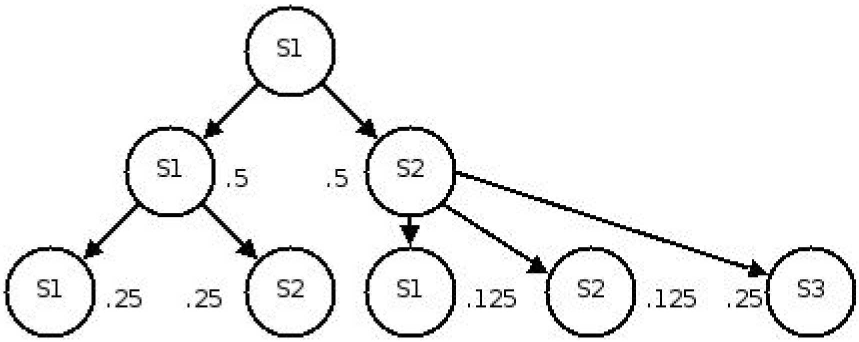

Figure 2.6: Probabilistic implementation model used with D=0.25.

The first tier S1 is always set based on a sampled random variable c_1 from an independent PDF representing the possible weather patterns. The second tier has a 50% chance of having the same weather and a 50% chance of not having the same weather. This repeats when a new weather pattern has not been created. If a new weather pattern has been identified there is still a 50% chance of creating a third weather pattern on the third tier but the other 50% is split evenly at 25% for condition S1 and 25% for condition S2. Mathematically a PDF is modified to follow this process. The second PDF is based on the first PDF and is generated using equation 2.6.

$$F_{C_2}(c_2|C_1) = F_{C_1}(c_2) * (1 - D_m) + \delta(c_2 - C_1) * D_m \tag{2.6}$$

where F_{C_1} is the independent weather distribution, $\delta(x)$ is the Dirac delta function, and D_m is the $\sqrt[n-1]{D}$ where n is the number of plants and D is the probability of all the plants producing the same power. The nth PDF is then adjusted as in equation 2.7.

Unfortunately thorough analysis of this method shows that the probability of the first plant is 50% likely to be the same as the second plant however it is only 37.5% that the first plant will be the same as the third plant. Although the number of plants with the same weather is represented as desired the relationship of any given plant to any other plant is incorrect. If the power plants are of equal production capacity then this caveat can be ignored.

$$
\begin{aligned}
&F_{C_n}(c_n|C_1 \cap C_2 \cap \cdots \cap C_{n-1}) = \\
&\quad F_{C_1}(c_n) * (1 - D_m) + \\
&\quad \delta(c_n - C_1) * \frac{D_m}{n-1} + \\
&\quad \delta(c_n - C_2) * \frac{D_m}{n-1} + \\
&\quad \cdots + \\
&\quad \delta(c_n - C_{n-1}) * \frac{D_m}{n-1}
\end{aligned} \tag{2.7}
$$

2.5 Proposed Model

In order to incorporate multiple plants a new model for the grid simulations was designed. The most significant change was the addition of multiple solar plants and wind plants. To better utilize the renewable sources the system also uses energy storage instead of a generator to provide electricity when renewable energy is not available. This model provides two great features. First the energy storage can be evaluated given various dependence levels and numbers of plants. More importantly when the renewable system stabilizes the grid with no outside influence it shows what is required for 100% solar, wind, or combined penetration. Figure 2.7 shows the layout for this new model.

Each of either the solar pants or wind plants are assumed to be the same type and design while all have the same production capacity. Wind data is generated separately from solar data as well which implies that they are completely independent. Other models have used a similar method of independent wind and solar such as [1].

The energy storage can be anything from batteries to artificial reservoirs to designs such as superconductive inductors embedded in the bedrock [11]. All of these real world designs have operational nuances which are ignored in this thesis for a more general model. The design decided upon assumes an overall efficiency and storage capacity. One to three days worth of storage were always required creating a maximum of 1/6 C discharge or charge rate based on a peak load of four times the average load. Most storage elements can meet this requirement so these constraints were ignored. If the discharge or charge rate of an

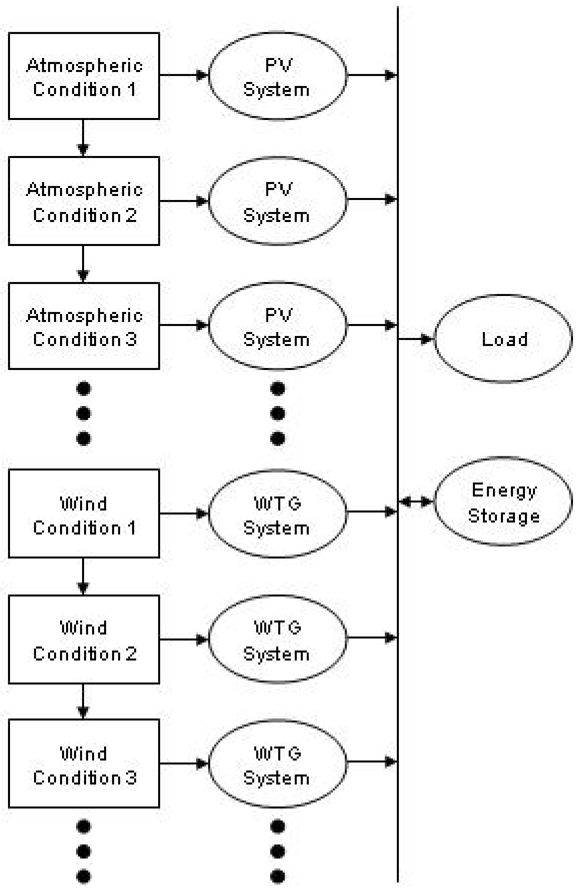

Figure 2.7: New model of large scale grid utilizing renewable energy and energy storage.

actual design is limited to a slower rate then the energy storage will be too small to handle the load properly.

Chapter 3

Generation of Simulated Solar Radiation Data

The generation of solar radiation data is done in multiple steps. In this case historical power output from a solar array was available through [12]. This power output data is characterized through some mathematical method in order to describe its behavior so that solar power could be estimated for multiple partially dependent solar power plants. The process involved changing to solar insolation, analyzing the insolation data, creating randomized solar insolation for multiple plants, averaging that solar insolation over the numerous plants, and then determining the solar plant output power based on the insolation, cumulative plant size of all the plants, and solar conversion efficiency.

3.1 Typical Methods of Data Generation

There are two typical methods used in the design and analysis of solar plants. One method uses expected power output averaged over months or years of time in order to estimate the energy going to the grid. This method is crude and does not consider weather patterns, diurnal energy output, or variations in weather from day to day. This method is typically used in grid-connected units designed to create electricity for income assuming either no need for load matching or a certain percentage of lost power that did not match the load.

The other method used to determine a solar plant's effectiveness uses time-based Monte Carlo simulations. This method has numerous variations but every one is designed to inspect the ability of the solar plants to match the load. This is often used to either determine the effectiveness of the plant by determining the plant's capacity factor or determine the stability of the grid that the solar plants are connected to using LOLE.

This more advanced model is implemented in three steps as described in [2] in reference to the design of the WATGEN program. These first three steps are used in order to generate hourly solar radiation using ARMA models based on available data. Figure 3.1 shows a basic block diagram showing the three steps used in order to generate hourly solar radiation.

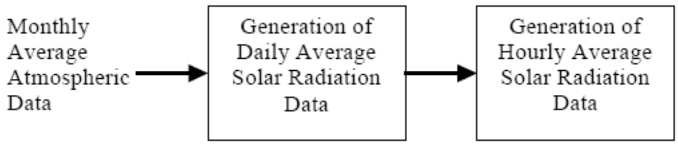

Figure 3.1: Basic steps WATGEN uses to generate solar insolation [2].

The incident solar insolation on the solar panels must be determined from the solar insolation. This is largely dependent on the physical characteristics of the solar plant such as which direction the solar collector is aimed, the tilt angle of the collector, array tracking capabilities, and geographic location.

The last step takes the incident solar insolation and converts it to energy output from the plants that utilize solar energy production. The energy that is generated from a photovoltaic solar panel can be modeled by equation 3.1 obtained from [9] where M_{PV} is the solar array size, T_t is the temperature, and S_t is the incident solar insolation on the solar arrays. It should be noted that this equation is specific to solar panels and that there are various other forms of energy production that use solar radiation. All of these have separate equations and specific application differences in determining the electric output so this is only intended as an example.

$$E_t = 3.24 * M_{PV}(1 - 0.0041 * (T_t - 8)) * S_t \tag{3.1}$$

3.2 Adjustments for Number of Plants and Geographic Separation

Unfortunately the more advanced ARMA model to determine solar insolation is not appropriate. This is because the simulation requires a generic set of weather to New England and a dependence between multiple plants. ARMA models cannot currently meet either of these requirements. Additionally the average expectation method is not appropriate. This is because a simulation indicating a time-based insolation is required while this more basic method uses an average over a long period of time. For this reason a new method was devised that is based on a wind prediction method as seen in [3]. A PDF is generated for each day of the year indicating the expected solar radiation for the entire day based on past data. These PDFs are then used to create a time-based solar radiation generation model.

The generation of these daily solar radiation PDFs are based off of collected power production data in Worcester, MA. This data was acquired from Matt Arner at Heliotronics [13], the sponsor for "*Solar energy and photovoltaics education in Worcester*" [12]. Using the daily solar radiation a table was generated showing the weather conditions that were present in the past for each day of the year. This allows the use of a window method to generate the PDFs.

The window method is used multiple times in this report as a tool to acquire pertinent data from a set of gathered data. This method takes all the data within a time window over multiple years when the data is available and recognizes it for analysis. As an example if the window was set for 2 days and the day of interest was January 15th then all of the data gathered on the 13th, 14th, 15th, 16th, and 17th of January for all years would be recognized. Data gathered for the 15th of January with varying window sizes can be seen in table 3.1. It should be noted that the data in table 3.1 is just an example using fake data with no particular meaning required.

Once the window size is determined selecting what data to use when creating the PDF is rather simple. The data is first collected from the given day of interest, in this case the 15th day of the year, January 15th. The window then dictates how many days before and after this day data should also be collected. If the window size is 0 then the only data used

Table 3.1: Example data gathering of individual PDF set up for the window method.

Day of year	...	12	13	14	15	16	17	18	...
1999		544	395	619	83	405		673	
2000		852	442		265	437	856	545	
2001		143	536	915	81			870	
2002		597					708	49	

Table 3.2: Data gathered for January 15th.

Window Size	Data Gathered
0 Days	83 265 81
1 Days	83 265 81 619 915 405 437
2 Days	83 265 81 619 915 405 437 395 442 536 856 708

is from the 15th. If the window size is 1 then the 14th, 15th, and 16th should all have their data collected. Equation 3.2 dictates the range of days that data will be collected. D_w is the day of the year that the PDF is being generated for, W is the window size in days, and D_c is the day of interest. If this evaluation is true then D_c's data over the various years should be included. This is a wrap-around style algorithm so $D_c = -1$ implies the 365th day of the year.

$$D_w - W \leq D_c \leq D_w + W \tag{3.2}$$

Using equation 3.2 and table 3.1 example sets of gathered data were generated for the 15th of January. The example sets can be seen in table 3.2. When data is not available the blank data is ignored in the window method.

From the resultant data a PDF can be generated for that particular day. Each collected piece of data is represented as a delta function with an area corresponding to the probability of that event and the position based on the radiation defined by the collected data value. For a window size of 0 days equation 3.3 represents the PDF for January 15th where i is the random variable for possible outcomes according to the window method.

$$F_I(i) = \frac{\delta(i-83)}{3} + \frac{\delta(i-265)}{3} + \frac{\delta(i-81)}{3} \tag{3.3}$$

Equation 3.3 is generated using the general form of the PDF function from equation 3.4 where I_j is the jth data element acquired from the window method and N is the number of data elements acquired using the window method. In order to represent these PDFs digitally the delta functions are spread out over 100 equally distributed areas over the range of the corresponding PDF.

$$F_I(i) = \frac{\delta(i-I_1)}{N} + \frac{\delta(i-I_2)}{N} + \cdots + \frac{\delta(i-I_N)}{N} \tag{3.4}$$

Once the various PDF's have been generated for each day of the year simulated solar radiation data may be generated for one location that has similar weather characteristics to the gathered data. This single plant's weather data can then be used with the generation of more PDF's as described in the execution methods section of the *The Geographic Separation Model* chapter. Besides generation of solar radiation the rest of the solar simulation work just as a normal Monte Carlo simulation does.

3.3 Data Generation Methodology

The generation of random solar radiation is done using the PDF's that were described in the previous section, *Adjustments for Number of Plants and Geographic Separation*. At the top level a few major functions are performed. First the data is attained and placed in the proper format for the window method. It should be mentioned that the data in this simulation was determined from solar array power output and as such this information is transformed into estimated radiation based on the solar arrays efficiency of 12.35% and size of 9.72 m^2 which were determined from [12]. All other efficiency losses such as DC to AC conversion losses are ignored. This data is then passed to a maximum solar radiation generator, a PDF generator, and a perfect day generator. These pass PDF's, daily maximum output energy, and an array of perfect days for every day of the year to the simulation data generator. Figure 3.2 shows the major functions just described.

The generation of daily maximum solar radiation is done for one major reason: it was

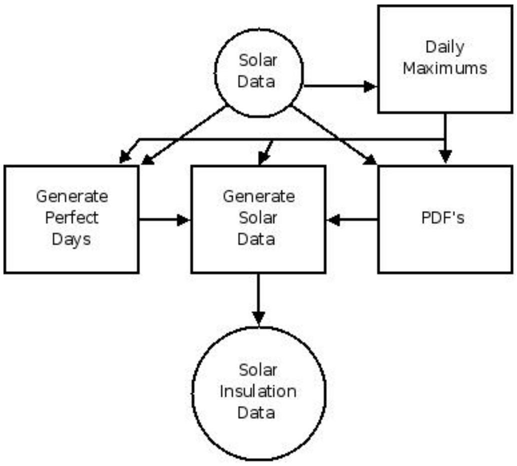

Figure 3.2: Basic flow diagram representing the generation of random solar radiation data.

desired that the system could be analyzed as compared to maximum energy output for that given day. A perfect day would be represented by 1 and a day with no energy production would be characterized by a 0. If more advanced methods were used then cloud cover could be utilized to determine the effective percentage of the daily maximum radiation.

Creating the maximum data set is a rather simple process. This uses the window method as described for the solar radiation PDFs. For the particular day D_c the maximum energy output within the window for all the years specified is chosen as the maximum for that day. Specifying a window of $W = 2$ days for January 15th and using the data present in table 3.2 the maximum for $W = 2$ days is 915. Most important to describing the daily maximum solar radiation is the window size used to determine those maximums. A window size that is too large will provide results that are higher than is to be expected. Using a window size that is too small can often result in a maximum that does not characterize the maximum for that day of the year because of a chance set of low value conditions. The window size must also be equal to or larger than the window used to create the PDFs in order to ensure that any simulated day does not produce more than 100% of the maximum. For these reasons the window was increased from 0 days until a relatively smooth output was generated that had no significant dips in the maximum energy production for the entire year. Even though it was found that the maximum energy could be generated with a window range of $W = 8$ days the window range used was $W = 20$ days to match the PDF's window size.

The daily maximum solar radiation levels are passed to the PDF generator, the solar radiation generator, and the perfect days generator. The PDF generator scales the data according to the daily maximums and then using the window method creates a set of PDF's for every day of the year. For simulation the PDF's must be represented by some form of probability mass function. For this reason they must also be split up into some number of discrete probabilities, also referred to as steps in [3]. Karki, Hu, and Billinton have described the minimum number of steps for the type of simulation that they are running [3]. Unfortunately their simulations were for wind power and included no form of energy storage. For both of these reasons 100 steps were used instead of their proposed six as a large safety margin.

In order for the solar data to be realistic a set of days assuming perfect weather for solar generation is created and passed to the solar data generator. The model used for a perfect day is the positive half of a sine wave stretched between the start of the day's sunlight and end of the day's sunlight. The magnitude in W is scaled such that the integral of the power over the entire day is equal to the maximum data set's element in Wh corresponding to the same day. The beginning and end of each day is determined based on a linear interpolation between the longest and shortest day of the year. Figure 3.3 shows the length of each day of the year assuming that the shortest day lasts from 7:30 AM to 4:00 PM and the longest day lasts from 6:00 AM to 8:00 PM. Daylight savings time is not incorporated into this model.

Generating the solar data was the more complex block of figure 3.2. This block uses a loop to iteratively generate solar data for multiple plants. Up to this point all of the information generated and gathered is generic to all solar plants and all simulation scenarios used in this thesis. This block takes this information and creates unique data sets for each set of simulation conditions. The number of plants and the dependence of the plants determines the unique characteristics of the data set being created. A flow chart indicating the process used to generate the sets of radiation data is presented in figure 3.4.

For each data set there are a specified number of plants that data will be generated for. The flow chart starts by generating the first plant's solar radiation for four years worth of data. Following this if more plants have to have their radiation data generated then a

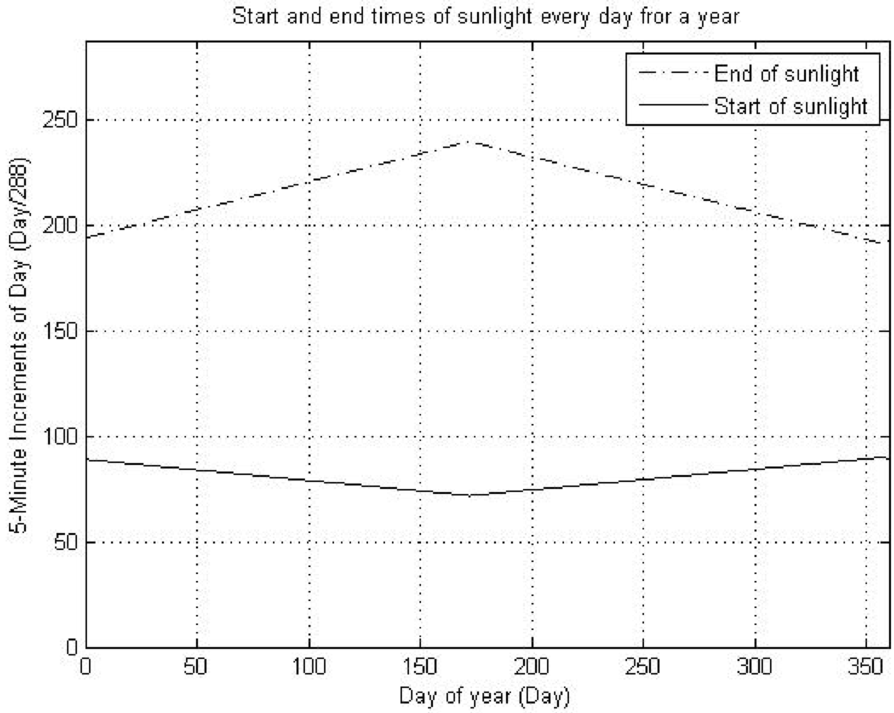

Figure 3.3: Length of each day of the year used in solar radiation data generation.

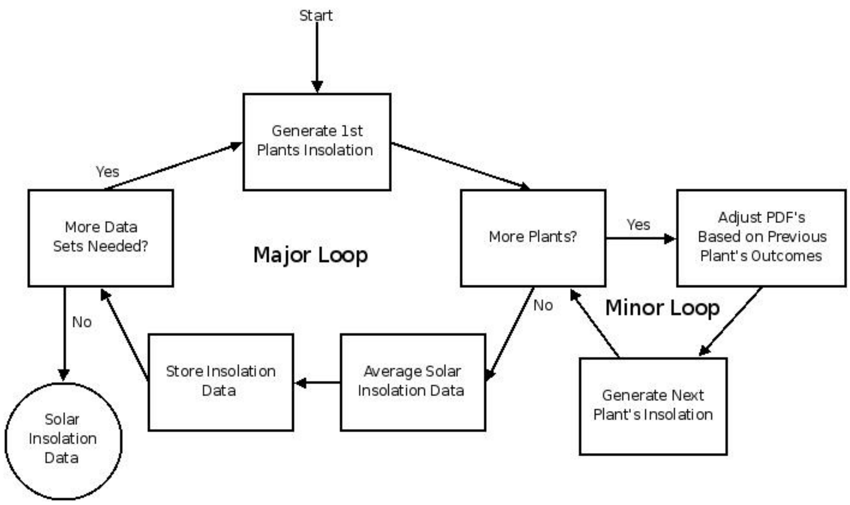

Figure 3.4: Flow chart indicating method of solar weather generation.

minor loop, where each of the other power plants have their solar radiation generated, is entered. First the PDF's are modified as described by equation 2.7 for every day of the four years. The modified PDFs are then used to create the plant's output and if no more power plants need solar radiation data then the minor loop is terminated and the process returns to the main loop. Directly following this the solar radiation output for all of the sites is averaged together with the assumption that the sum of all the identical plants will have the same production capacity as a data set assuming one plant. This data is saved into a table and the loop continues until one hundred sets of random solar insolation data is generated and is passed to the output. Equation 3.5 shows the relationship between the production capacity of n_1 plants as compared to n_2 plants where PC_1 is the production capacity for n_1 plants and PC_2 is the production capacity for n_2 plants.

$$n_1 * PC_1 = n_2 * PC_2 \tag{3.5}$$

As can be seen a single plant has the same production capacity as numerous plants. For this reason the average solar radiation can be used assuming a linear relationship between the solar insolation and power production capacity which is true in this model.

The generation of solar radiation for a particular plant is a little more detailed than the rest of the flow chart in figure 3.4. Initially the daily solar radiation is determined on a scale of zero to one based on the PDF's provided from the weather data. This value indicates how close the weather is to the maximum solar radiation of that day. This is also the solar radiation that is used in the adjustment of the other power plant's PDFs for solar radiation.

In order to generate solar radiation data over an entire day a few more steps must be taken. A perfect day is used as a starting point with five minute time increments. A repetitive process is employed until equation 3.6 is true where S_c is the scaled solar radiation, S_m is the maximum solar radiation for the entire day, and S_{5i} is the ith five minute increment of weather for that day.

$$\Sigma_{i=0}^{288} S_{5i} \leq S_m * S_c \tag{3.6}$$

The first step in this repeated process is to pick a five minute time span during the day determined by a PDF that is uniformly distributed between 1 and 288. This five minute time span's solar radiation is then multiplied by a random number whose PDF is uniformly distributed between 0 and 1. This repeats until the total solar radiation is correct for that given day. When the scaled solar radiation is less than .1 the perfect day is automatically scaled to 1/10th it's original size before the loop. If the scaled solar radiation is less than .01 then the perfect day is scaled to the scaled solar radiation and the repetitive process is avoided. This is done to speed up the generation of solar radiation data.

Although the solar radiation was generated in five minute increments it is desired that the final output be in fifteen minute increments to speed up the Monte Carlo simulations. For this reason the discrete solar radiation data is added together in groups of three in order to form fifteen minute increments. If multiple plants are to have their output combined then the fifteen minute increments amongst the plants are then averaged. This is the final form of solar radiation data generated. The size of the plants adjusts how much energy is produced by simply multiplying the cumulative plant size by all the fifteen minute increments.

In order to combine solar and wind plants there had to be a method to convert the solar radiation to $\frac{Wh}{W}$ for each of the fifteen minute time increments, a measure of energy produced in Wh per unit of production capacity in W. Using constants for panel efficiency and panel $\frac{W}{m^2}$ the needed $\frac{Wh}{W}$ measurements can be determined. For example equation 3.7 can be used to perform this conversion where E_{pwt} is the energy produced per watt of production capacity during the fifteen minute time interval t in $\frac{Wh}{W}$, S_{rt} is the solar radiation during the time interval t in $\frac{Wh}{m^2}$, E is the efficiency of the solar panel array, and PV_r is the $\frac{W}{m^2}$ constant for the solar panels. Multiplying E_{pwt} by the production capacity in W yields the energy in Wh produced during the time interval t.

$$E_{pwt} = \frac{S_{rt} * E}{PV_r} \tag{3.7}$$

Chapter 4

Generation of Simulated Wind Speed Data

Generating wind speeds characteristic of a geographic location can be done using many different methods. Unfortunately like the generation of solar radiation the typical methods used to generate random wind speeds are not entirely appropriate for a system in which multiple locations are partially dependent on one another. Because of this a modified version of the existing models is presented for data generation that can vary dependence over multiple locations.

4.1 Typical Methods of Data Generation

Wind models are often similar to solar radiation models. ARMA models are used most often for sites with a large amount of historical data as seen in [1, 3, 14, 15]. Sites with less than adequate data for an ARMA model or without need for such detail will often use a different method involving the use of a weibull distribution [3] or a direct distribution modeled identically to the sample data through PDFs such as in [5].

For site specific models the ARMA model is used extensively. For one geographic location this model can provide an hour by hour wind prediction which when compared to the actual wind can be very accurate preserving not only sample by sample probabilities but also autocorrelation as seen in [15]. For an in depth description of how an ARMA model works [14] is a good source. Unfortunately wind speed ARMA models do not allow for the dependence between two sites to be modeled. They must be either completely dependent

Table 4.1: Six step common wind speed model [3].

Wind Speed (By μ and σ)	Wind Speed (km/hour)	Probability
$\mu - 2 * (\frac{5\sigma}{3})$	0	0.0051
$\mu - (\frac{5\sigma}{3})$	2.7633	0.1920
μ	19.5300	0.6120
$\mu + (\frac{5\sigma}{3})$	36.2967	0.1796
$\mu + 2 * (\frac{5\sigma}{3})$	53.0633	0.0109
$\mu + 3 * (\frac{5\sigma}{3})$	69.8300	0.0003

or independent.

Weibull distributions can be used as an alternative to ARMA models. Because this method relies on a PDF to determine wind speed multidimensional PDF's can be generated identically to the process used in the generation of random solar radiation data. As was mentioned earlier an alternative to the weibull distribution involves the use of a distribution created directly from the collected wind speed data.

The creation of a direct model is very simple. Wind data is gathered over an entire year and the results of wind speed are used in the creation of a PDF in the form of equation 4.1, much like equation 3.4. W_j is the jth wind speed measured and N is the number of data samples gathered. This effectively builds a common model for any geographic area comparable to where the data was gathered [3].

$$F_W(w) = \frac{\delta(w - W_1)}{N} + \frac{\delta(w - W_2)}{N} + \cdots + \frac{\delta(w - W_N)}{N} \tag{4.1}$$

The best feature of a common model is the way it can be modify in order to generate another site's weather when little or no historical data is available. By scaling and shifting the distribution according to the wind's mean and standard deviation at the particular site a new relatively accurate model can be attained. The common model devised by [3] can be seen in figure 4.1 and the specific values used can be seen in table 4.1.

Once wind speeds are predicted the actual power production must be estimated based on the plant's power generation characteristics. A typical wind turbine will have three speed ratings; Cut in, rated, and cut out speed. Cut in speed is the speed at which the wind speed is high enough to generate electricity. The power output rises with wind speed until the

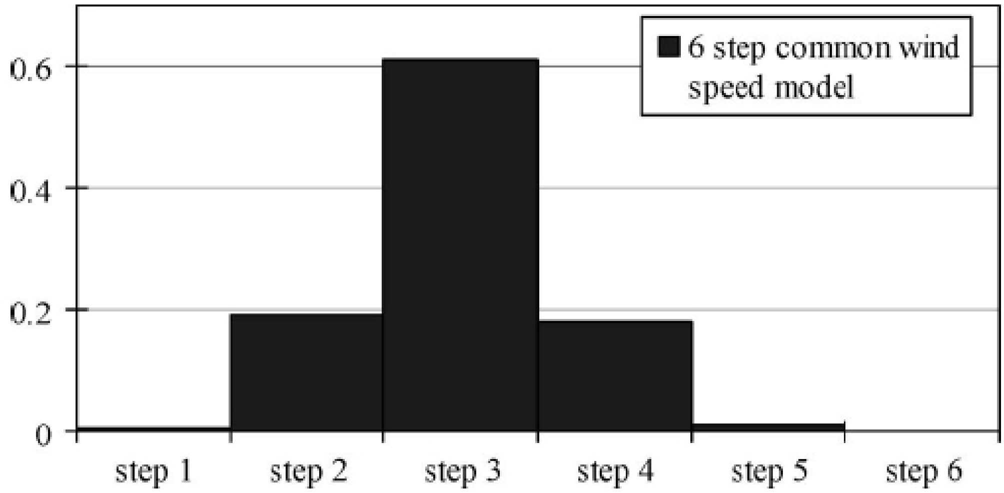

Figure 4.1: Six step common wind speed model [3].

rated speed is reached. The rated speed is the point at which the generator is producing a maximum specified power and cannot produce any more. The power output from the wind turbine does not change until the wind speed reaches the cut-out speed. The amount of energy generated at these wind speeds is the rated output. At the cut-out speed it is unsafe for the wind turbine to operate so it is shut down and no energy is generated. An example power output to wind speed curve is shown in figure 4.2 for a Clipper Liberty wind turbine.

Once random wind speeds are generated the electricity that a particular wind speed produces depends on the particular model and design of wind turbine. The wind speed to power output curve describes how a particular turbine reacts so that a designer can easily translate the system's power output given the wind speed. In order to determine a PDF of wind turbine energy production the wind speed to power curve can be used as a function of the random variable for wind speed. A typical power curve will have a high probability of no power production and rated power output since both cover a rather wide range of wind speeds. As an example the six step common wind speed model was also translated into a PDF representing power production in table 4.2. It can be seen that the slowest two wind speed steps and the fastest two wind speed steps have been combined because of the cut in speed and rated speed.

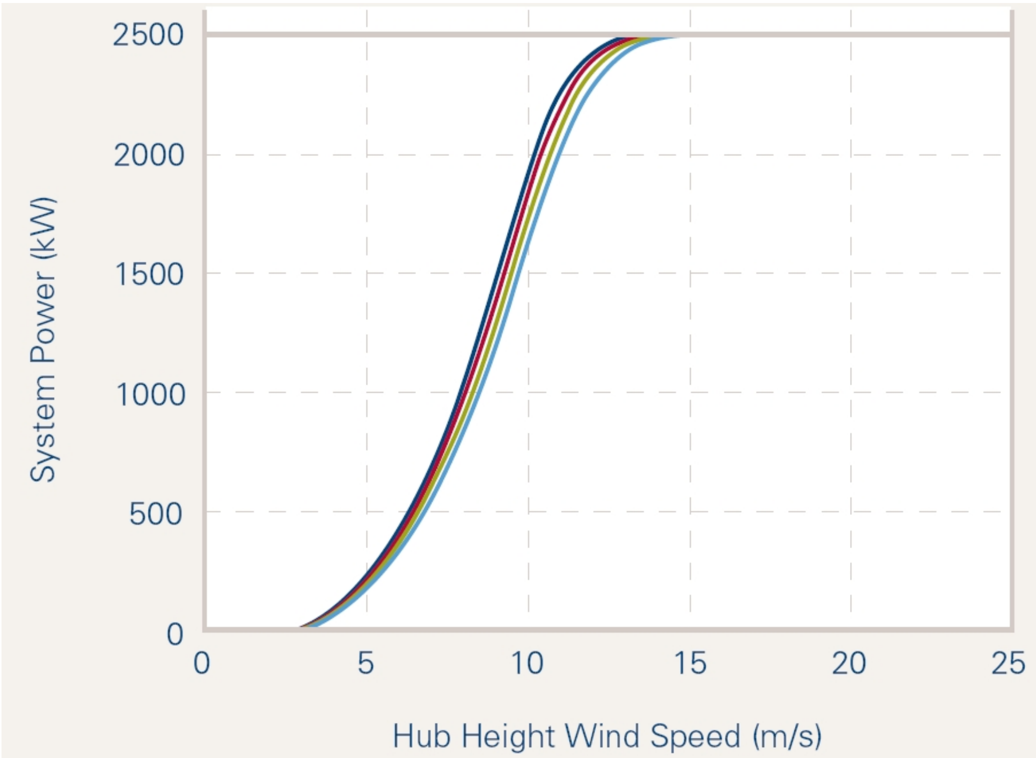

Figure 4.2: Example wind speed to power output curve set for an actual family of four wind turbines [4].

Table 4.2: Six step common wind speed model translated into PDF of power production [3].

Power Output (MW)	Probability
0.000	0.1971
1.325	0.6120
14.650	0.1796
27.000	0.0112

4.2 Simulation Concepts

For a more accurate representation of the LOLE it was desired that a yearly trend and a daily trend be used. A PDF based wind generation system similar to the solar method was also desired since the dependence methods could still be used. Unfortunately all of the classical wind modeling techniques described did not meet both of these needs. As a compromise a twenty five step model was employed from a set of distribution data local to Ipswitch, MA [5] assuming a hub height of thirty nine meters. This model was then normalized such that it had a mean wind speed of 1 m/s. In order to meet the yearly trend each monthly mean wind speed was used to scale the PDF to an appropriate size. The normalized PDF and the monthly average wind speeds can be seen in figures 4.3 and 4.4.

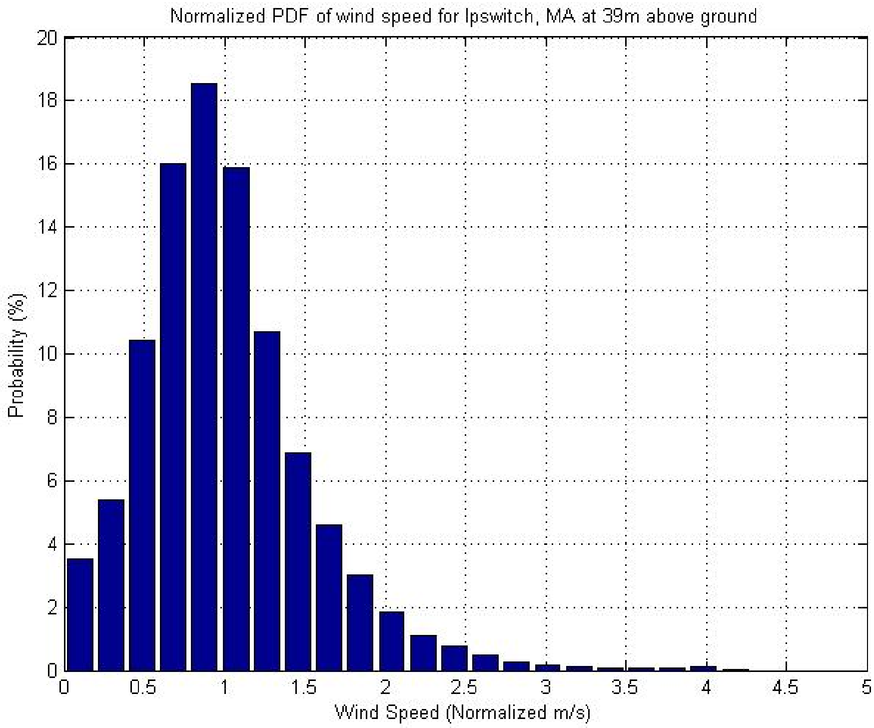

Figure 4.3: Normalized wind PDF from one year's worth of data gathered in Ipswitch, MA [5].

The next step to generate random wind data required fifteen minute time increments of wind. Preliminary testing showed that solar power required multiple days of energy storage

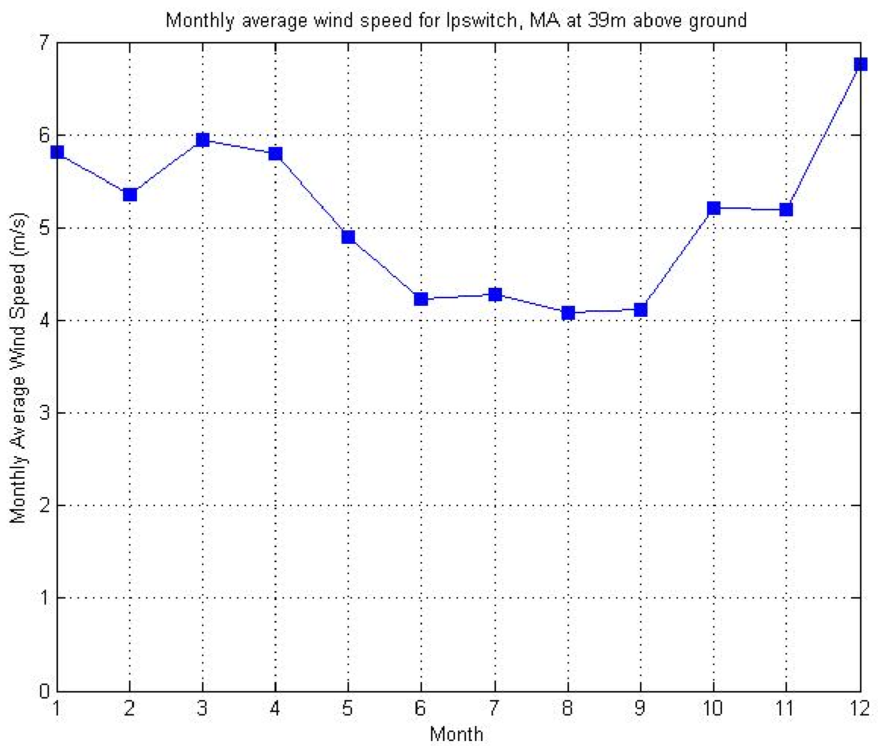

Figure 4.4: Average monthly wind speeds for Ipswitch, MA [5].

in order to stabilize the grid without outside aid. Wind power can be very sporadic much like solar energy so it was assumed that the need for energy storage would also be in the range of multiple days worth of storage. It was assumed that small fluctuations in wind power would be averaged and could be ignored because of the size of the storage. These assumptions allowed the use of the diurnal average output power according to daily wind characteristics from [5] to specify the fifteen minute energy production levels. This diurnal data was then normalized so that it's average was 1 m/s. By multiplying this normalized diurnal wind speed by the random daily average the fifteen minute increments were then attained. The normalized diurnal wind speed can be seen in figure 4.5.

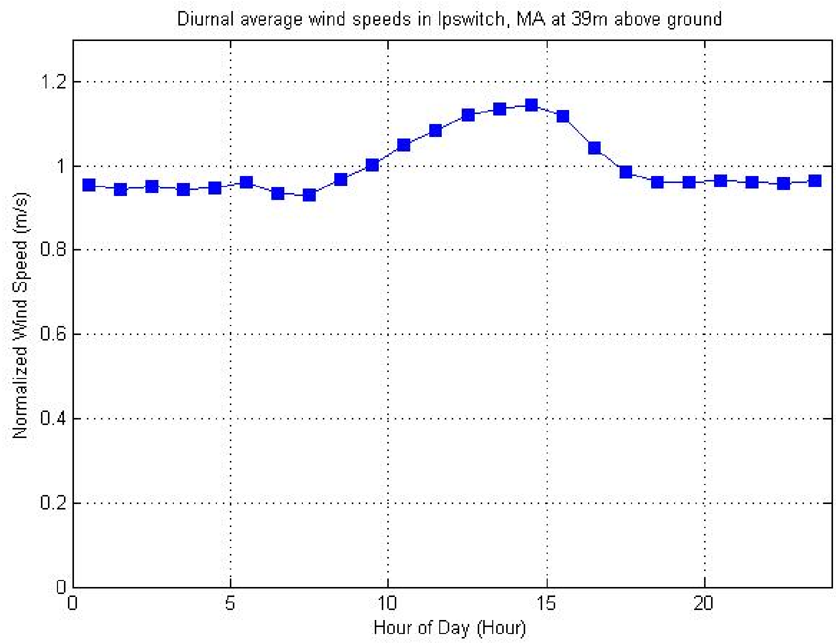

Figure 4.5: Normalized diurnal average from one year's worth of data gathered in Ipswitch, MA [5].

4.3 Data Generation Methodology

The methods used to generate random wind speed data was very similar to the basic steps involved while making the random solar radiation data. Daily average winds are determined the same way that daily solar radiation is determined. Multiple wind plants have their PDFs modified just like with multiple solar plant's. Most of the differences appear when one wants to generate the random wind on an hour by hour basis and then translate that to electric energy. Figure 4.6 shows the basic steps used to generate the wind energy data.

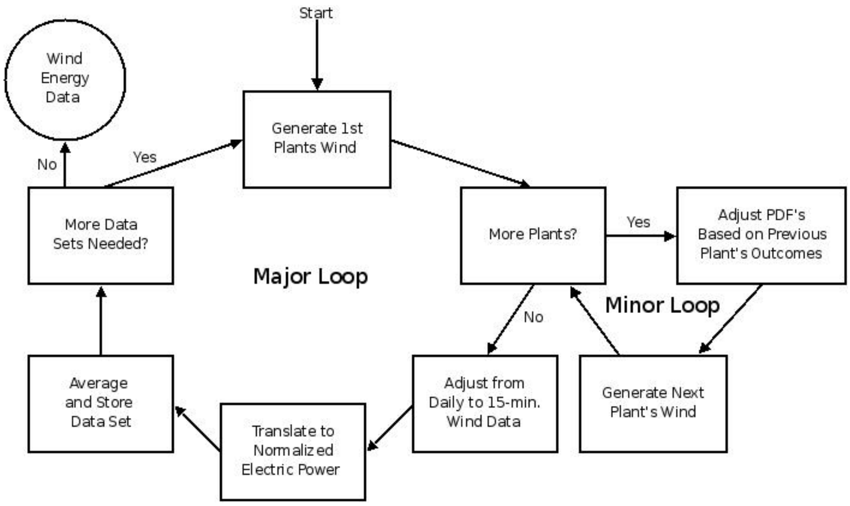

Figure 4.6: Flow chart indicating method of random wind energy generation.

The first step in the process is to generate four years worth of randomized daily average wind speeds. The random wind speeds are created from the PDFs determined for every day of the year. As described previously there are 12 PDFs split between the months of the year all of which are scaled versions of a normalized PDF to meet the average wind speed of the given month.

If there are multiple plant locations then the minor loop is entered. This minor loop works identically to the minor loop for solar radiation generation by repeatedly adjusting the daily PDF's for every day in the four year data set and then generating random wind data based on the new PDFs. Once all of the wind plants have wind data then the system returns to the major loop.

The next two blocks in the major loop deviate from the solar radiation generation methods. First for each day of random wind the information is spread out into a full days worth of fifteen minute increments. The values placed in each of these increments is based on the normalized diurnal average wind speed multiplied by that day's average wind speed.

Next the fifteen minute increments of wind are translated to electric power based on a specialized distribution shown in figure 4.7. This curve is an estimate for the curve of a wind generator recommended for the Ipswitch MA site in [5] which is also included as a comparison. It was desired that the wind turbine be estimated from the cut in, rated, and cut out wind speeds rather than a detailed set of points in case many different wind turbines were used in experimentation. No other wind turbine models were used in this thesis despite this feature.

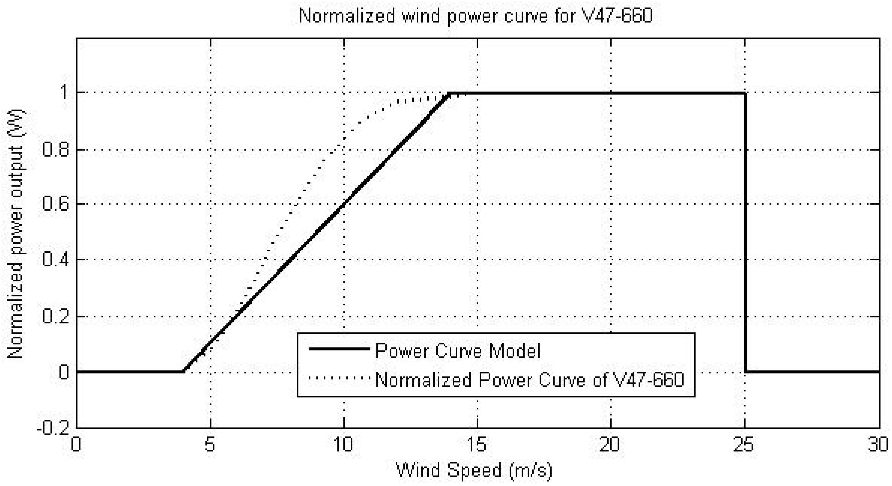

Figure 4.7: Normalized estimated power curve for wind turbine appropriate for winds in Ipswitch, MA[6].

This curve assumes no power production below the cut in speed and a linear interpolation to one watt at the rated wind speed which stays constant to the cut out wind speed. It is assumed that the number of wind turbines will be numerous so the production capacity of an individual wind turbine is unimportant. Instead if the normalized power curve is used to translate wind speed into normalized power, then $\frac{Wh}{W}$ energy approximations per watt of production capacity are attained.

Once the estimated power has been determined for all of the locations the fifteen minute energy approximations can be averaged and then stored. This repeats in order to create multiple sets of random weather based energy production estimates. Multiplying all the data by a production capacity yields energy production estimates for the system. This data is then passed to the Monte Carlo simulations for analysis.

Chapter 5

Load Data Generation

In order to characterize a simplified load a few assumptions were made. One of the most prevalent assumptions was that the load would be modeled after the existing energy usage in Worcester County, Massachusetts, USA based on a scaled New England load. It was also assumed that the daily load probability would be normally distributed and the hourly load would follow from an average diurnal load pattern generated from historical data. Using these constraints a quick and effective load model was generated using a typical diurnal load, daily load mean values, and daily load standard deviations. With the exception of the assumed normally distributed load this method closely resembles that of [7].

If there are numerous independent but identically distributed loads on the grid then the addition of all the identical load's power requirements become normally distributed. It is safe to assume that there are similar types of loads on the grid because of residential as well as industrial applications that use similar equipment. Assuming that the loads operate independently on a daily basis they should form many normally distributed loads. By adding all the normal load distributions together a total distribution is attained which is also normally distributed. This allows for the normally distributed daily load assumption as an approximation of the actual grid.

The method used to generate the fifteen minute load data is most similar to the wind power data generation methods assuming a single site location. Daily means and standard deviations are determined for each day of the year. These are used to create normally distributed PDFs for each day of the year which are then used to create the random daily load requirements. These daily approximations are then converted to fifteen minute increments using the typical normalized diurnal load. A flow chart showing the generation steps to make the random load data is shown in figure 5.1.

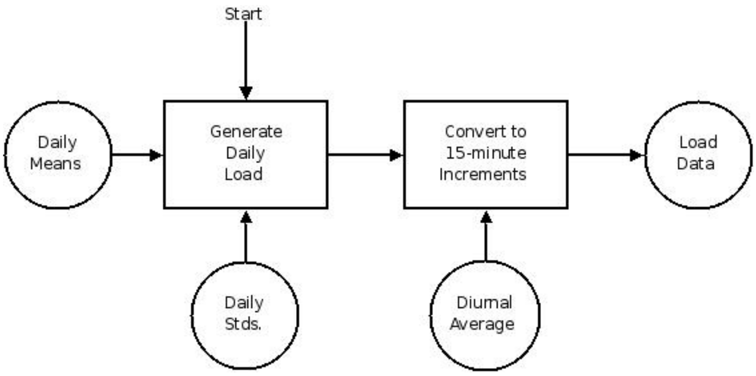

Figure 5.1: Basic flow diagram indicating method used to generate random load data.

To get the characteristics of the load there had to be collected data or an available model already created. Eight years of the New England load was attained through [16]. The typical diurnal load was determined by averaging the power required during every hour of the day over the entire set of data. Every element of the typical diurnal load was divided by the sum of all the elements to create the normalized diurnal load. A conversion from hour increments to fifteen minute increments was also required. Equation 5.1 shows a single step method to attain this fifteen minute normalized load from a diurnal load where L_{nt} is the normalized load during the fifteen minute increment t represented as a percentage of the whole day's load and L_{ath} is the average load during the hour th in Wh.

$$L_{nt} = \frac{L_{a\frac{t-(t(mod4))}{4}}}{4 * \sum_{h=1}^{24} L_{ah}} \tag{5.1}$$

By taking the daily New England load measurements and organizing it according to day of the year a mean and standard deviation was determined for every day. These were the means and standard deviations used to generate the distributions for the daily load requirements. When each element of the fifteen minute normalized load was multiplied by the full daily estimated load requirement the fifteen minute load requirements for that particular day were created.

Generating the daily load is done for all the data sets all at once. There is no dependence issues such as with wind or solar data generation so no looping is required. These daily loads were then each stretched out over a full day using the fifteen minute normalized load and cascaded together. The first day is always January first to match the random wind and solar data sets.

The last step to convert the fifteen minute load requirements was to scale it to an approximate Worcester County load. It was assumed that each person in New England uses the same amount of electricity so the final step was to scale the power requirement by the percent population of Worcester County as compared to the rest of New England. The scale factor used was an approximation of 5.5% of the total New England population [17].

Chapter 6

Simulation Methodology

The Monte Carlo simulations used to evaluate the system were rather basic from a top level view. This system took in sets of fifteen minute energy production and fifteen minute load requirements as well as storage characteristics. The result was the mean and standard deviation of the LOL for the circumstances that the sets of data and storage specifications created. The fifteen minute energy production and fifteen minute load requirement sets were determined as according to the previous few chapters while the storage characteristics were constants used in the estimation of grid sized energy storage. Figure 6.1 shows this very basic top level diagram.

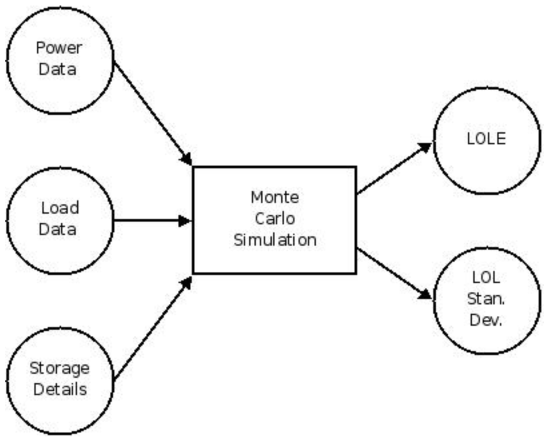

Figure 6.1: Basic Monte Carlo Simulation.

For each fifteen minute time increment the load was balanced using the energy storage and the available energy. Excess energy was used to charge the storage element while the storage provides energy when the power generation facilities could not. If the energy storage was depleted this represented a loss of load condition determined by the ratio of the fifteen minutes that were not provided power. At the end of the simulation the numerous total possible LOL conditions from numerous simulations were averaged over one years time to attain the LOLE. The standard deviation was also recorded as a measure of accuracy.

The load data is taken from the load data generation methods directly with no modification. This particular load data set is used in every Monte Carlo simulation. It should be noted that the size of the load was rather unimportant, but rather the comparison of the load to the equipment used to power it was of more importance. This load was similar to approximately 784,000 people's energy usage according to the approximate population of Worcester County, Massachusetts as compared to the entire New England load.

Generated energy data is not as simple as the load data. Energy data represents the generated energy provided to the grid for every fifteen minute increment of time over four years. By taking the $\frac{Wh}{W}$ measurements for every fifteen minute time increment and multiplying it by the watts of production capacity a Wh measurement was determined. This allowed control of the production capacity of a given power source for every set of simulations run. When wind and solar were to be combined they were equally represented by averaging the $\frac{Wh}{W}$ measurements for every set of fifteen minute increments. This allowed a fair comparison to a system that uses just wind or solar plants.

The storage specifications included the storage size and the average overall storage efficiency. For simplicity it was chosen that the storage element would have a basic efficiency that determined how much energy would be lost between the time the system puts energy into the storage element until the energy comes out of the storage element. The storage size itself represents how much energy can be taken out of the storage element and delivered to the load with no efficiency loss when it is at full capacity. During the Monte Carlo simulations it was theorized that a properly sized storage element would be on average half full thus the system should be started with the energy storage at half of it's full capacity. This was preferred over no charge because the first few days would represent an uncharacteristic event in a well designed renewable system which should always have some level of charge. Full charge was also not desired because it would represent a possible rare circumstance which would skew the results towards a lower LOLE. Maximum charge and discharge rates were ignored because of the large size of the energy storage elements.

The method of actually performing the Monte Carlo simulation is a little more complicated than the top level diagram implies. The details of this simulation methodology include finding the difference between the load and the power generated, on a step by step basis determining the state of charge in the storage element, and determining the LOL for every fifteen minute interval. This process is done for each of the one hundred sets of simulation data in each set of simulations. A flow chart showing the process of the Monte Carlo simulation can be seen in figure 6.2.

Initially a load mismatch is determined. This is simply the difference between the load and the energy production for every fifteen minute increment. Equation 6.1 is used to determine the load mismatch where LM_t is the load mismatch during the fifteen minute increment t, E_{gt} is the energy generated during the fifteen minute interval t, and L_t is the load requirement during the fifteen minute increment t. A purely stable grid will have a very small variation in LM_t however with renewable energy the initial load mismatch at any given time can have a very large amplitude compared to the load requirement.

$$LM_t = E_{gt} - L_t \tag{6.1}$$

The energy storage was used to compensate for the large amplitude in the load mismatch. For every fifteen minute increment in the sequence the current state of the energy storage will change. The given storage condition can be determined from equation 6.2 where $E_s(X)$ is the energy storage condition at time X, LM_t is the load mismatch during time interval t, E_s is the average overall energy efficiency, and T is related to t as the end of the time span represented by t measured in time increments equal to t.

$$E_s(T) = E_s(T-1) + LM_t * E_s \tag{6.2}$$

The energy storage is limited by the overall storage size so before the next storage state can be determined from a current state the energy storage must be limited to its maximum and minimum capacity. The load mismatch was then adjusted according to how much the

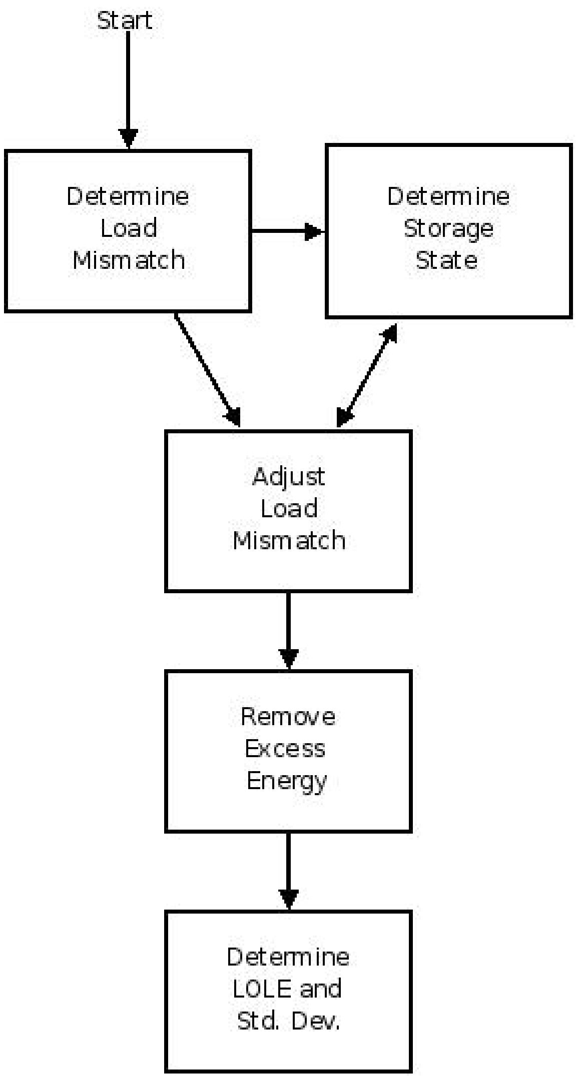

Figure 6.2: Details of the Monte Carlo simulation.

energy storage could balance the load's requirements. If excess power is generated beyond what the energy storage can absorb then the load mismatch still has a positive energy mismatch and the energy storage state is capped at full capacity. If the energy storage was completely depleted in the effort to power the grid and more energy was needed then there will still be a negative energy mismatch and the energy storage is capped at 0 Wh of storage. Any positive mismatch can be dumped through some means or simply not produced in the case of solar power and is thus removed from the load mismatch. Negative energy mismatch is of more importance and represents a loss of load condition. Assuming fifteen minute increments are used equation 6.3 provides an estimate for the number of hours the LOL condition lasted during a fifteen minute interval that a LOL condition has occurred where LOL_t is the number of hours of LOL condition present during the time span t. This occurs in the last step of the flow chart where the LOL conditions are determined.

$$LOL_t = \frac{LM_t - E_s(T-1)}{4 * LM_t} \tag{6.3}$$

Because the separate simulations have no effect on each other they are all run at the same time in the form of a giant set of arrays. When the final array containing all LOL_t conditions was available the hours LOL were summed for each simulation. The resulting set of summed LOL conditions creates a mean and standard deviation providing the LOLE originally sought using equation 2.2.

Chapter 7

Results and Discussion

It was theorized that by designing a power generating plant that is geographically separated into multiple smaller plants a grid could operate with a smaller storage size. Also of interest was how certain changes would affect the need for high energy storage efficiency as well as large cumulative plant size. The LOLE variation showed how the storage capacity, storage efficiency, and cumulative production capacity requirements were affected. Each of the simulations run could vary with plant size, dependence between the plants, number of plants, load size, storage size, and storage efficiency.

In order to understand the effects these variables would have on a system, three sets of simulations were run. One set was run with only wind power, the second set was run with only solar power, and the last with an equal amount of wind and solar power. Each one of these simulation sets had three variations; storage capacity, storage efficiency, and cumulative plant production capacity. Each of the three simulation types were run with 1, 3, 6, and 10 separate plant locations with 11 dependence levels ranging from completely independent to completely dependent. This made 396 simulation configurations in all. Figure 7.1 shows this layout just described.

The most important simulations resulted in a LOLE approaching 0 hours per year. For this reason the worst condition must approach 0 hours per year in the simulation to properly evaluate the results. It was expected that the worst case scenario would be using a single geographic location regardless of dependence level. To ensure this, various storage size and cumulative plant size pairs were found through repetitive simulations. These pairs would approach 0 hours per year LOLE to be used as maximum points during simulation. It

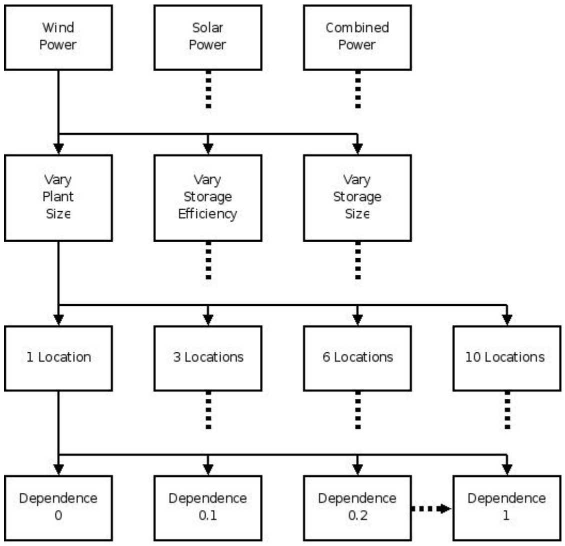

Figure 7.1: Simulation layout.

was found that even when the storage efficiency was set at 100% there had to be either an enormous storage size or cumulative production capacity to meet a LOLE of 0 hours per year. For this reason a 10 hours per year LOLE intercept line was used to find a single pair for solar, wind, and both combined.

These maximums were used as both limiting test points and constants during simulation. As an example, when simulating with variation in storage capacity the size was varied from 0 KWh to the maximum KWh while the cumulative power production capacity was held constant at the maximum limit chosen. The storage was also held at 100% efficient.

Determining these maximums was the next step before running the simulations. An algorithm was used to approximate the cumulative plant size where $LOLE = 10$ hours per year for various energy storage sizes assuming a 100% storage efficiency. This provided a graph as seen in figure 7.2 for wind, 7.3 for solar, and 7.4 for both.

As is shown in the plots of maximum pairs, a point has been chosen which is the maximum cumulative plant size and maximum energy storage capacity pair used in their respective simulations. Any point on the line was appropriate as the maximum pairs reference point however it was assumed that a power facility would be designed with cost in mind. These points would perhaps represent a low energy storage size as well as low cumulative

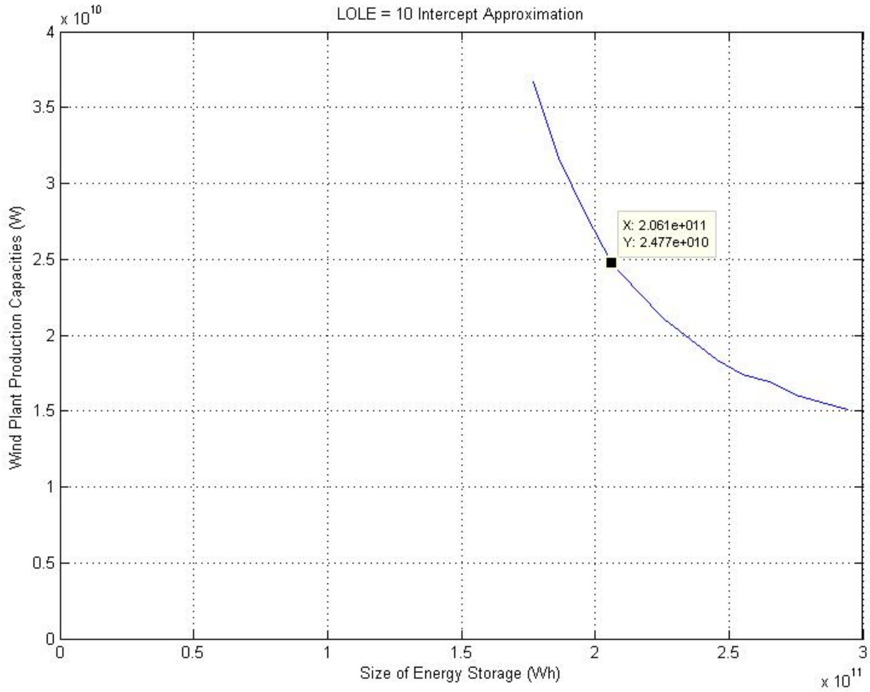

Figure 7.2: $LOLE = 10$ hours per year intercept approximation for wind.

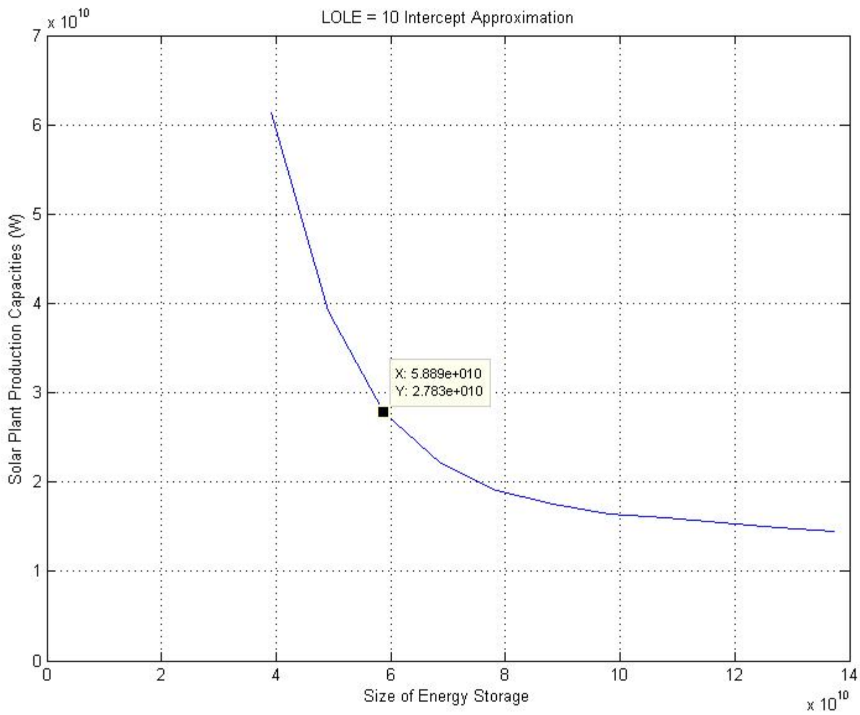

Figure 7.3: $LOLE = 10$ hours per year intercept approximation for solar.

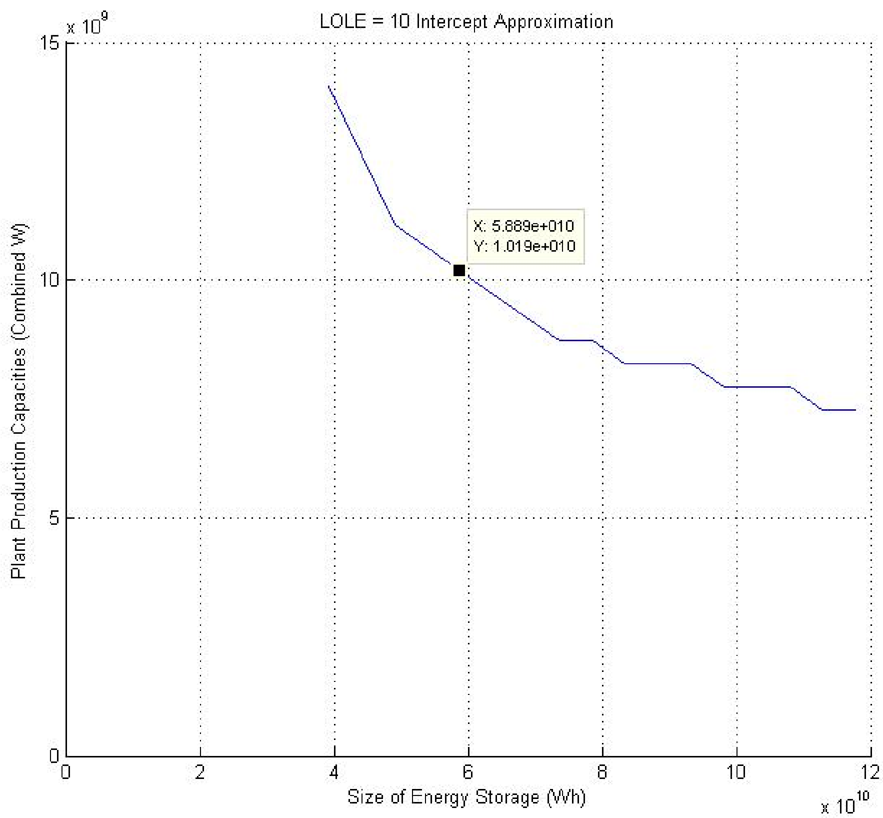

Figure 7.4: $LOLE$ = 10 hours per year intercept approximation for both solar and wind combined.

production capacity. These points should also prevent the storage or production elements from overwhelming each other in terms of stabilizing effects.

Unfortunately no pertinent results can be attained from using the same constants for wind, solar, and combined simulations so comparisons between these simulation sets can sometimes be subjective. What can clearly be seen is that the combined wind and solar plants often require a lower storage size and plant size as compared to the other simulation sets to attain an equal LOLE.

7.1 Variation of Storage Size

It has been confirmed that the energy storage required to stabilize a grid is very large. One of the main objectives of geographic separation was to partially stabilize the renewable power that was being generated in order to reduce the need for such large amounts of energy storage. In order to evaluate any improvements, a set containing cumulative plant size, storage efficiency, and storage size was determined for each of the three kinds of plants that would provide a low LOLE with one geographic location. Cumulative plant size and storage efficiency were held constant while the storage size was decreased in increments of 2% of the determined storage size until it reached 0 KWh of storage. These steps were performed with solar, wind, and both combined using varying numbers of locations and varying levels of dependence. To compare them fairly all simulations pertaining to a given plant type used the same cumulative plant size, storage efficiency, and maximum storage size.

7.1.1 Wind Simulations

The first simulation run assumed only wind power. Vast improvements could be seen assuming complete independence, $D = 0$, as seen in figure 7.5. This shows that it is possible to remove approximately 75% the energy storage for three locations and perhaps even 87% for more locations.

As much as the energy storage was improved for a completely independent set of conditions, it is also desirable to know how the system will react with weather that is more

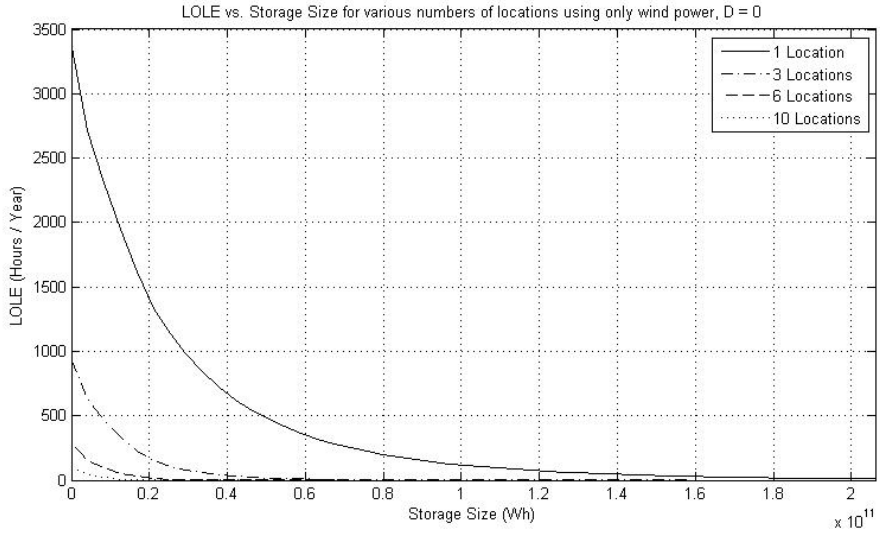

Figure 7.5: LOLE for $D = 0$ and 1, 3, 6, and 10 geographic locations with varying storage size using wind plants.

dependent. It is hard to see the difference in results for 6 and 10 locations so the complete set of 11 dependence levels for 3 locations is compared to a single location in figure 7.6. Each of the black solid lines represents a different dependence level with three locations. The lower the LOLE as compared to the other lines, the lower the dependence level. Complete dependence acts almost identically to the single location's performance, which should be expected. There are immediate improvements to LOLE as soon as some independence is seen, however more independence beyond $D = 0.8$ does not lower the LOLE much in comparison. A large improvement from $D = 1.0$ to $D = 0.8$ is a desirable result because very little independence leads to a huge improvement in the LOLE. For six and ten locations the same pattern is seen.

In order to better see the LOLE for various geographic locations and dependence, levels they are all graphed in figure 7.7. This figure is also zoomed in over a small range compared to the previous graphs for clarity.

As useful as figure 7.7 is, a better understanding of the 10 hours per year LOLE intercepts was desired. These intercepts were interpolated and placed in table A.1. Additionally this table includes the percent comparison to one geographic location. Figure 7.8 shows the percent comparisons on a graph for the various numbers of locations for a visual interpre-

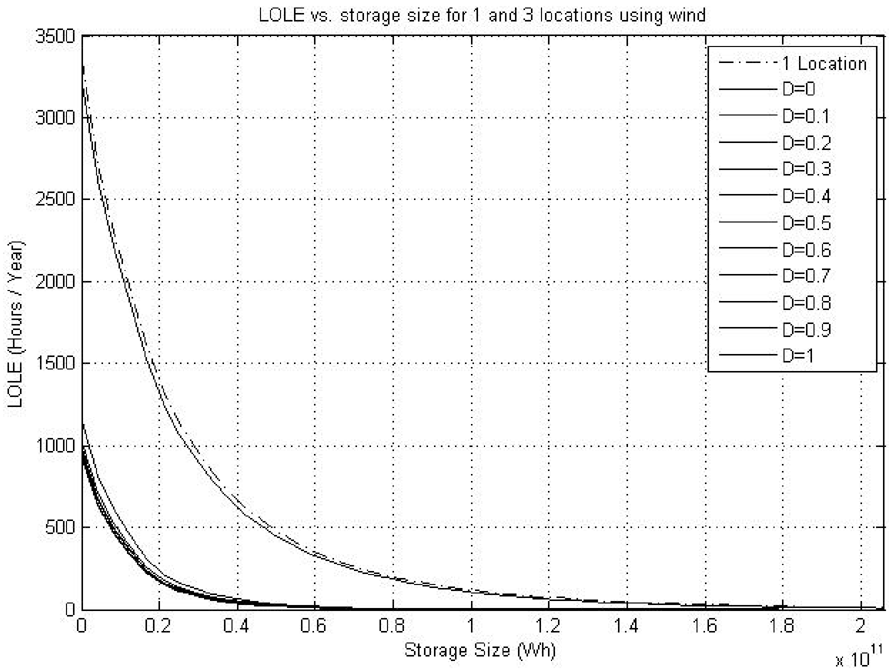

Figure 7.6: LOLE for various dependence levels for 3 geographic locations compared to 1 geographic location with varying storage size.

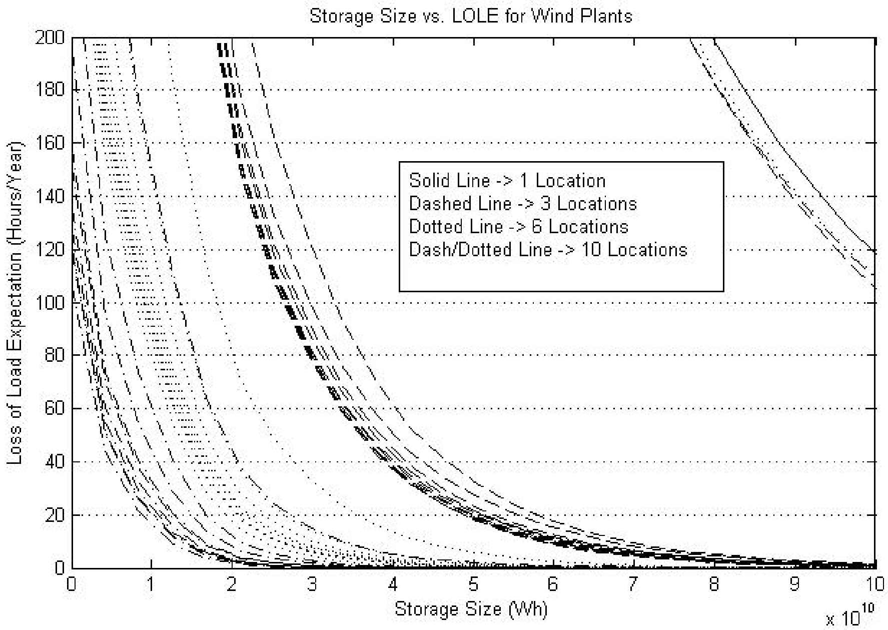

Figure 7.7: LOLE for various dependence levels and geographic locations with varying storage size.

tation.

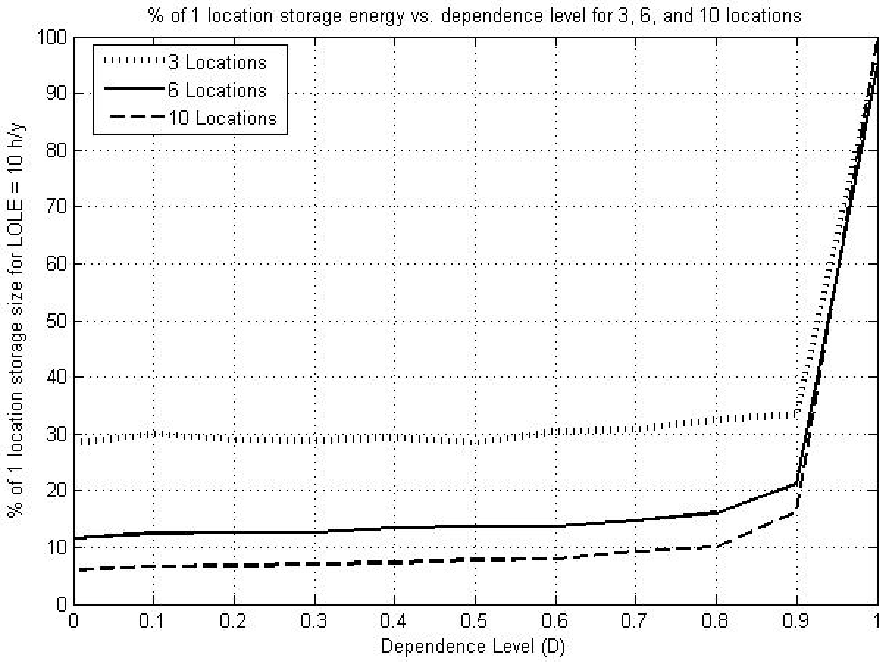

Figure 7.8: Percent of one location storage size for $LOLE = 10$ hours per year vs. dependence level for 3, 6, and 10 locations using wind power.

7.1.2 Solar Simulations

Although wind power seems to improve with an increase in the number of locations, the simulations with exclusively solar plants did not show an obvious improvement. Instead the probability of independence seemed to dictate the difference between the various LOLE curves. Given a certain D the plots with 3, 6, and 10 locations were virtually identical. Using measured standard deviations of each Monte Carlo simulation it was found that virtually every set of 3, 6, and 10 locations for a given dependence level were within one standard deviation of each other except for when $D = 0$, complete independence. This showed that when given a level of dependence the number of locations would typically have little affect on the LOLE. A graph showing the various dependence levels for 10 locations is shown in figure 7.9.

Analysis of the results as dependence changes shows that the system does not react the

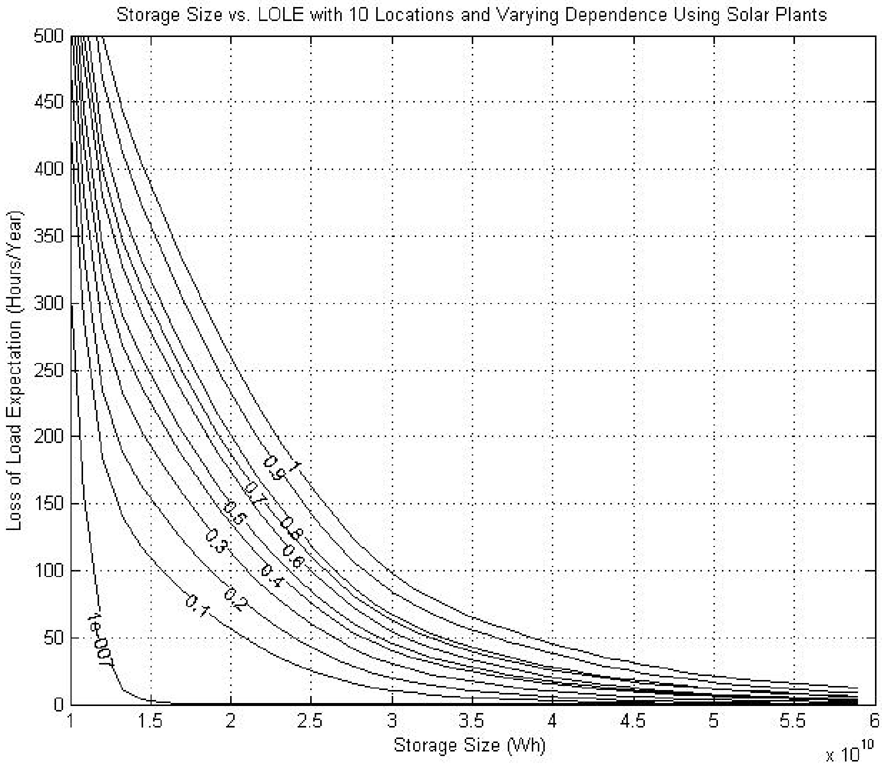

Figure 7.9: LOLE for varying storage sizes with 10 geographic locations and varying levels of dependence using solar plants.

same as it would with wind power. Instead of having a significant immediate change as the locations became more independent such as with wind simulations, the system reacted slower as seen from the data in table A.2 and the graph of that data in figure 7.10. This may imply that a farther geographic separation may be necessary to have improvements comparable to what wind power plants achieve and this may imply that there is not much improvement available with solar generation because the geographic separation required may be impractical.

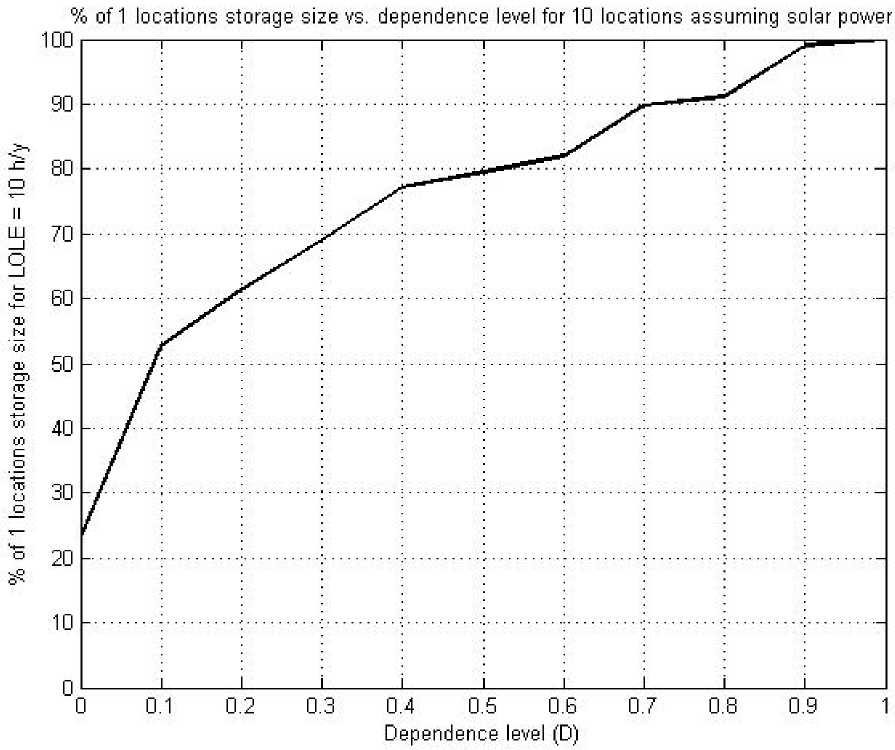

Figure 7.10: Percent of 1 location storage size for $LOLE = 10$ hours per year vs. dependence level for 10 locations using solar power.

7.1.3 Combined Wind and Solar Simulations

By using both solar and wind power it was theorized that the LOLE would decrease as compared to just solar or wind because the generated diurnal energy would more closely match the diurnal load. Although this will neither be proved nor disproved, the LOLE is lower for a given storage size than just solar or wind power. It should also be noted that the cumulative plant size maximum was much smaller than for just wind or solar power plant simulations. Figure 7.11 shows LOLE's for various storage sizes and numbers of geographic locations assuming complete independence. The lower dependence levels are closely matched much like what was observed with the wind power curves in figure 7.6.

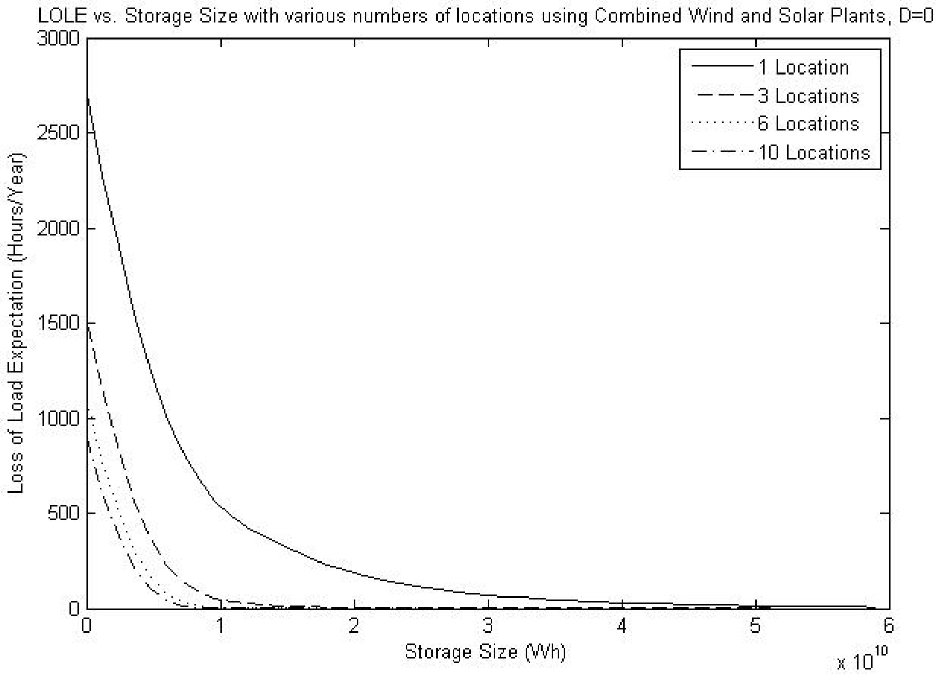

Figure 7.11: LOLE for $D = 0$ and 1, 3, 6, and 10 geographic locations with varying storage size using wind and solar power plants.

It is again important to inspect the decrease in storage size required for 10 hours LOLE as the geographic locations become more independent. Table A.3 displays the storage size required to reach a LOLE of 10 hours per year for each dependence level and number of plants pair. Figure 7.12 shows the percent decrease from the maximum storage size for 3, 6, and 10 locations for all dependence levels. The decrease in LOLE appears very similar to the characteristics shown with wind plants however the system does not improve as much. This was because of the difference in maximum storage sizes between the combined simulations and the exclusive wind simulations.

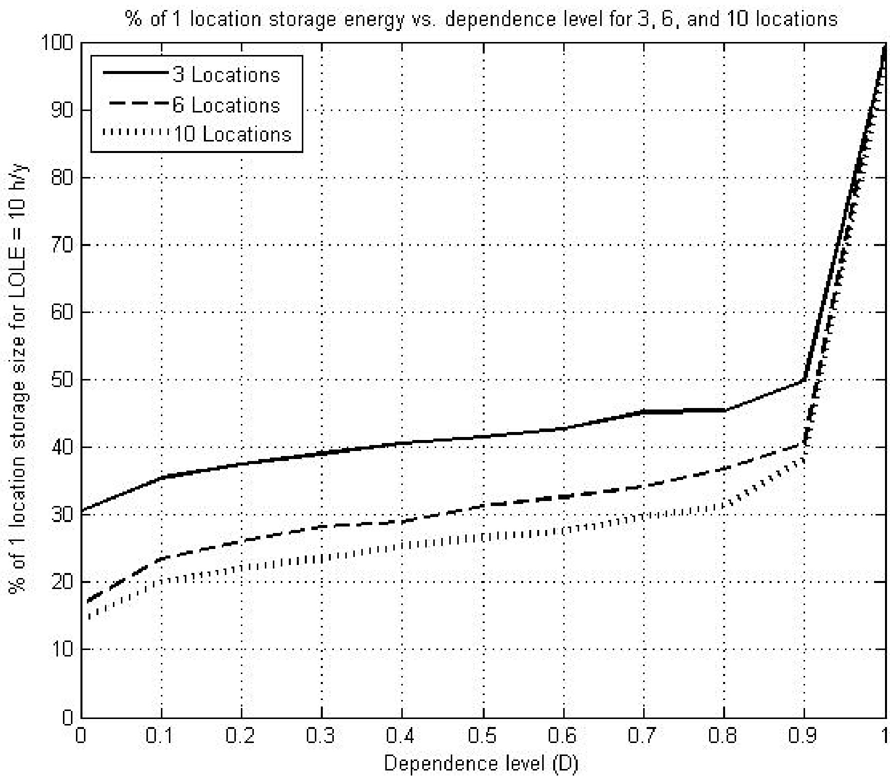

Figure 7.12: Percent of 1 location storage size for $LOLE = 10$ hours per year vs. dependence level for 3, 6, and 10 locations using wind and solar power.

7.2 Variation of Storage Efficiency

Different forms of energy storage have a different efficiency associated with them. Also different forms of energy storage will have a cost and possibly a size limit associated with them. Sometimes forms of storage with lower efficiency will cost less and if lowering the storage efficiency does not lead to a large change in LOLE then a less expensive form of energy storage may be used. For these simulations the storage size and the plant size are chosen as mentioned before, for the wind, solar, and combined simulations individually. The storage efficiency is now varied instead from 100% to 0% efficient in intervals of 2%.

It should be noted that although results may indicate that the system can stabilize with lower storage efficiency as the number of locations and independence improve this simulation also keeps the storage size constant. There is likely a balance between the size of the storage and the storage efficiency such that as the storage size decreases a higher efficiency is still needed. This however was not investigated directly.

7.2.1 Wind Simulations

Wind power again lead to significant levels of improvement. This time the improvement was in storage efficiency instead of storage size. Figure 7.13 shows this improvement with complete independence for all numbers of locations. Table A.4 and figure 7.14 shows the percent of one locations required efficiency as the independence increases for 3, 6, and 10 geographic locations. Yet again the independence for wind power shows little improvement beyond $D = 0.8$ as the independence improves.

7.2.2 Solar Simulations

Solar storage efficiency also seems to follow a similar pattern to the solar storage size simulations. Although there is improvement with the increase in number of geographic locations it is not obvious until 0 hours per year LOLE is approached. Figure 7.15 shows the two extremes of the simulations which indicates how similar they are. Additionally the standard deviation indicates that the two extremes are within approximately two to three standard deviations of each other until low LOLE levels are attained.

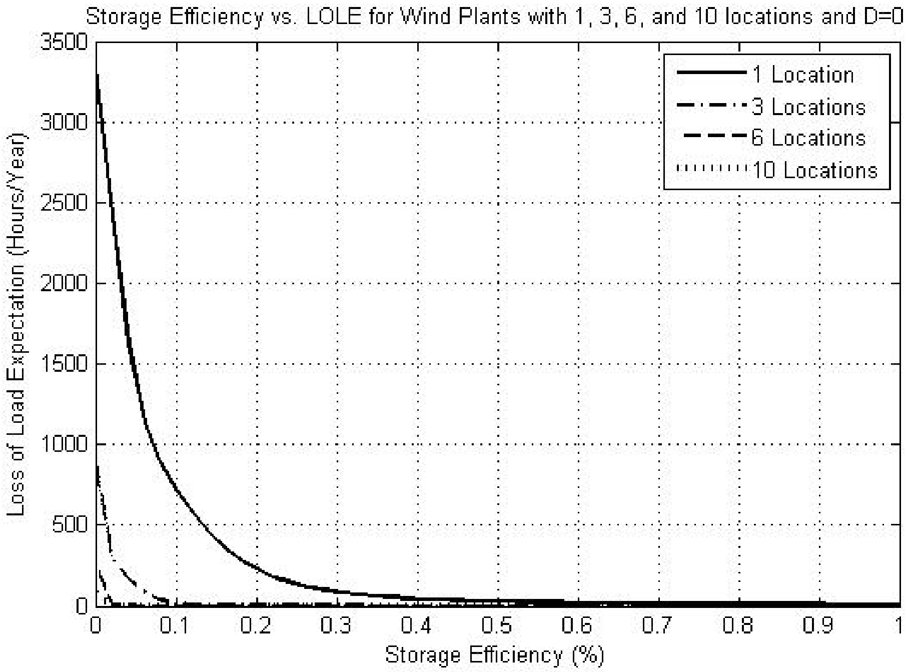

Figure 7.13: LOLE for $D = 0$ and 1, 3, 6, and 10 geographic locations with varying storage efficiency using wind plants.

Inspecting the storage efficiency required to reach 10 hours per year LOLE reveals an interesting pattern similar to the one seen with storage size. Table A.5 and figure 7.16 show this data and it is rather clear that the system seems to improve with an increase in dependence levels. This data also shows that there is little change when the number of locations is changed.

7.2.3 Combined Wind and Solar Simulations

The case with solar and wind appears similar to the case with just wind because of the immediate improvement as complete dependence changes to a slight amount of independence. Unfortunately the improvements are not as drastic. This may be because the wind plants were configured with approximately four times the amount of energy storage as when both wind and solar plants are combined. Figure 7.17 shows the LOLE vs. storage efficiency curves which also resemble the wind simulations.

The intercepts for 10 hours per year LOLE show results that yet again have little im-

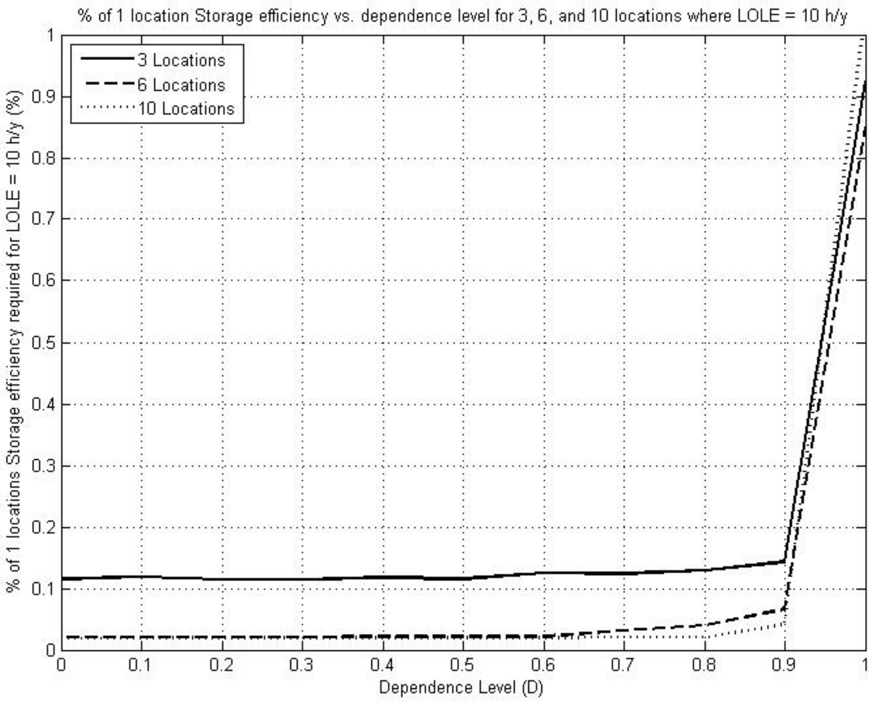

Figure 7.14: Percent of 1 location storage efficiency for $LOLE$ = 10 hours per year vs. dependence level for 3, 6, and 10 locations using wind power.

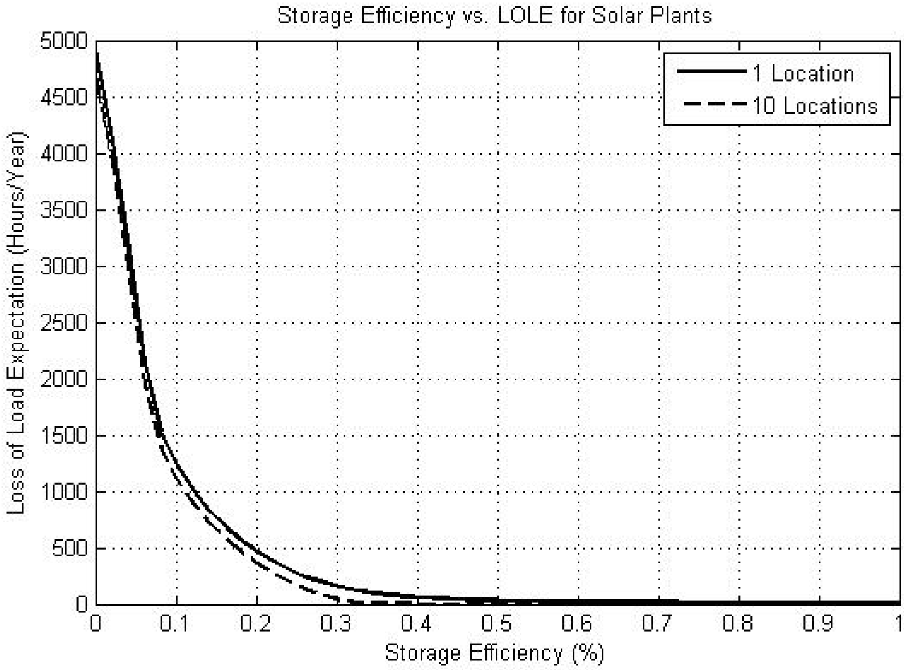

Figure 7.15: LOLE for $D = 0$ with 1 and 10 geographic locations with varying storage efficiency using solar plants.

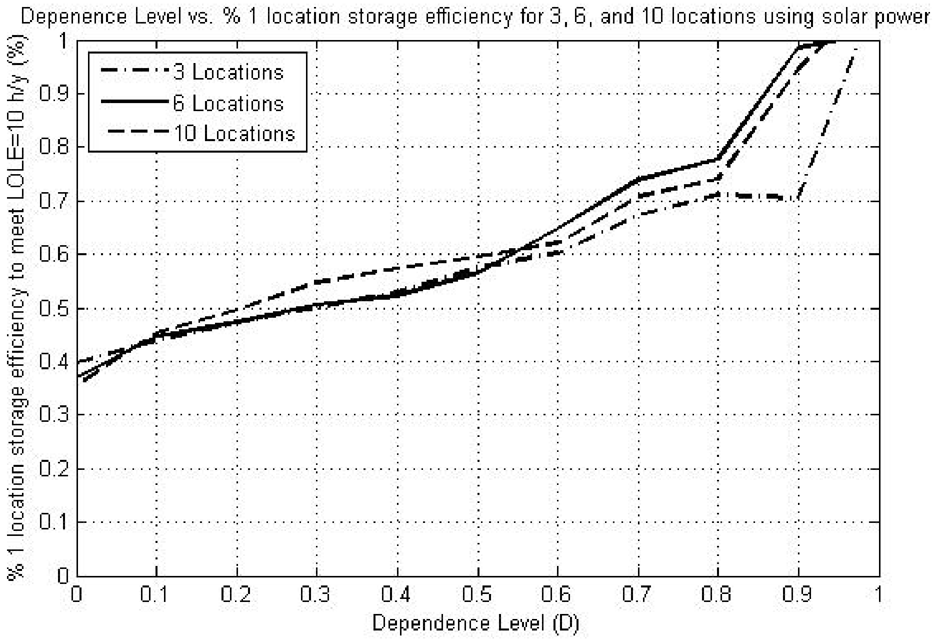

Figure 7.16: Percent of 1 location storage efficiency for $LOLE = 10$ hours per year vs. dependence level for 3, 6, and 10 locations using solar power.

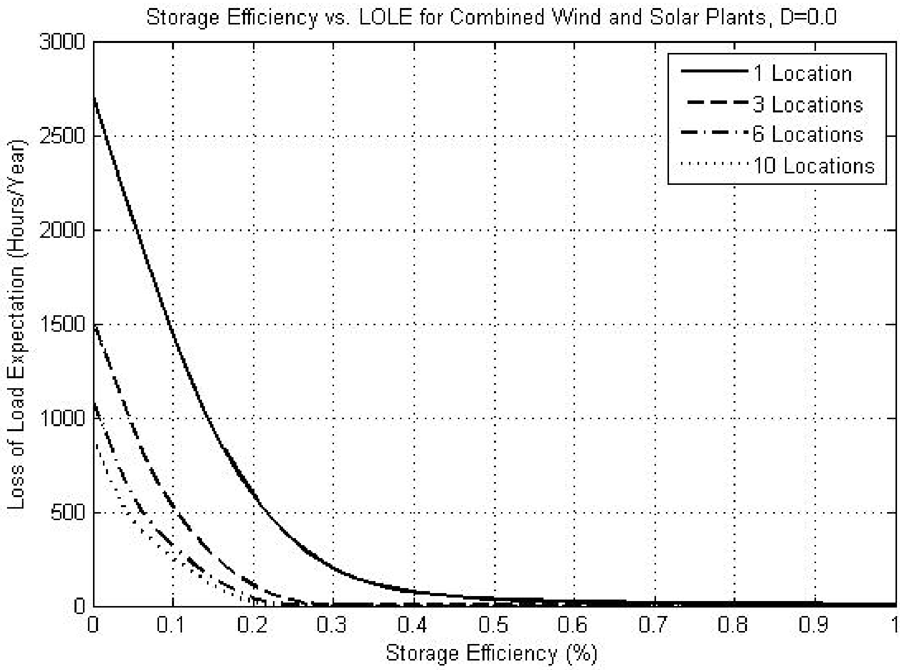

Figure 7.17: LOLE for $D = 0$ with 1, 3, 6, and 10 geographic locations with varying storage efficiency using both solar and wind plants.

provement once a level of dependence of $D = 0.8$ is achieved. As mentioned before this is good because the geographic locations can be rather close to one another while still stabilizing the power production.

7.3 Variation of Plant Size

Lowering the cumulative plant size was the last experiment. As the dependence decreases and the number of plants increases it is theorized that the power output will lend itself to lower production extremes where energy will be wasted, thus allowing a smaller cumulative plant size for the same LOLE. This applies to all three of wind solar and combined simulations although improvements amongst these vary. It should be noted that any set of wind, solar, or wind and solar combined simulations were very similar with little difference until low LOLE conditions are met.

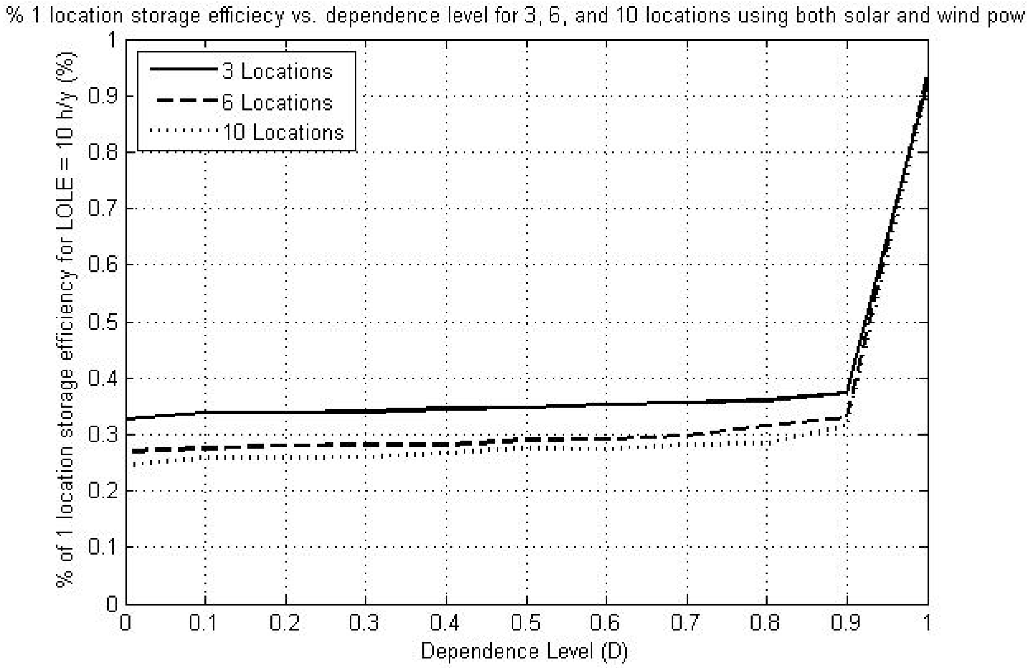

Figure 7.18: Percent of 1 location storage efficiency for $LOLE$ = 10 hours per year vs. dependence level for 3, 6, and 10 locations using solar and wind power.

7.3.1 Wind Simulations

To get a feel for how the LOLE varies with plant size only 10 and 1 locations were graphed with complete independence in figure 7.19. This forms the limits that all the other simulations within this set reside in. The intercepts with 10 hours per year LOLE do vary a great deal even though these two simulation sets are rather close.

Determining the 10 hours per year intercepts for all the graphs was performed and the results are presented in table A.7. The comparison in % of 1 location's LOLE can be seen in figure 7.20. Yet again there is a large improvement with very little independence. Unlike with previous simulations the improvement is approximately one third the original plant size which is not as good as with storage size or storage efficiency simulations but still represent a significant improvement.

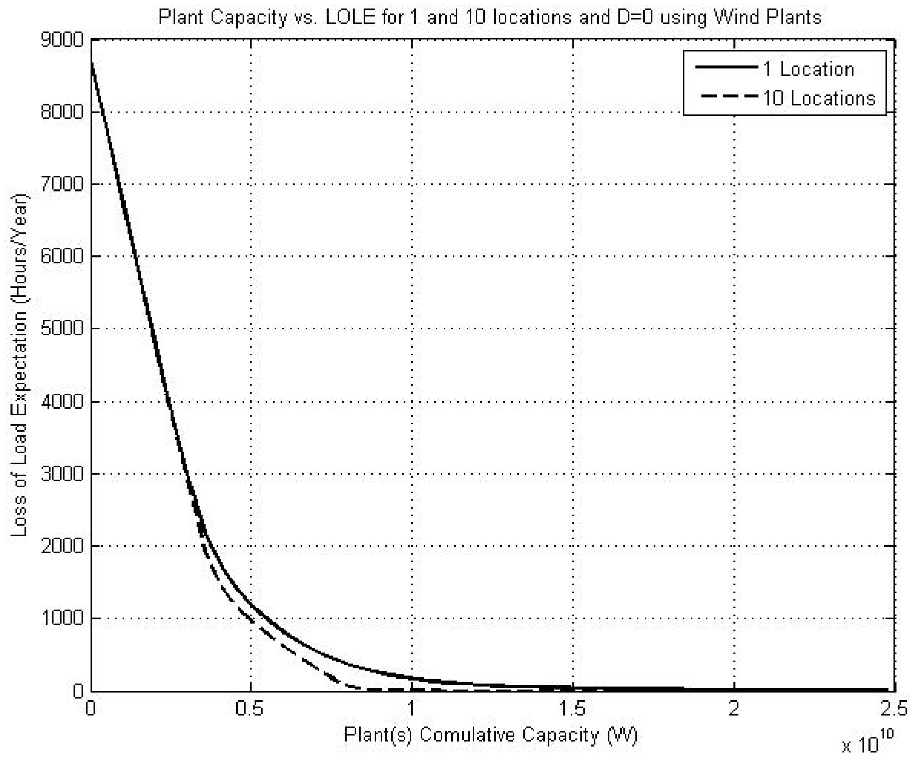

Figure 7.19: LOLE vs. cumulative wind plant size for $D = 0$ with 1 and 10 geographic locations.

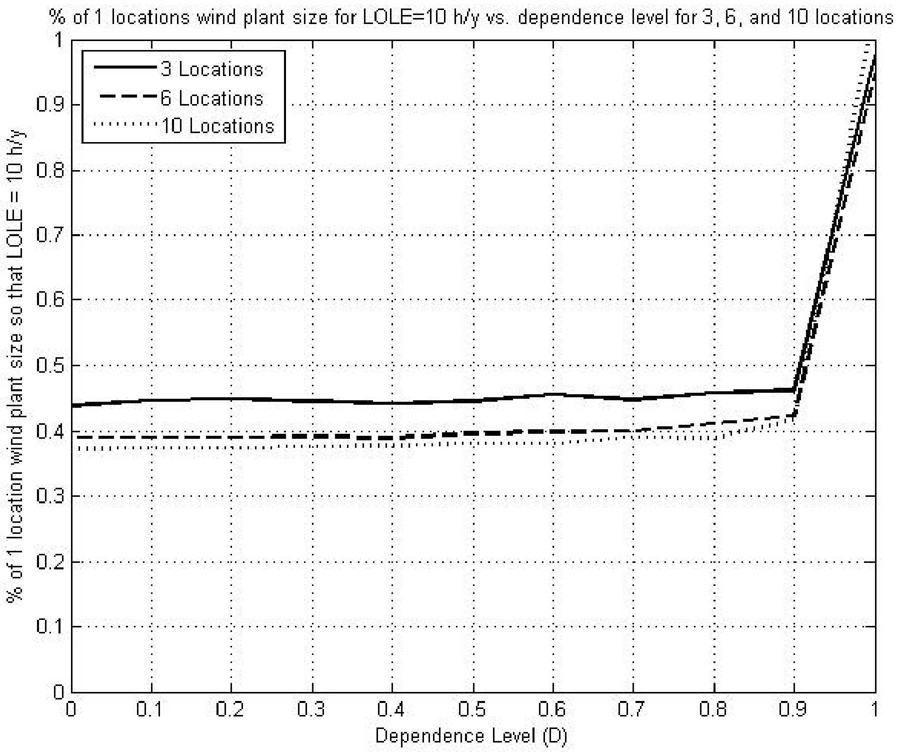

Figure 7.20: Cumulative wind plant size needed for $LOLE = 10$ hours per year vs. dependence level for 3, 6, and 10 locations.

7.3.2 Solar Simulations

As mentioned before the simulations have a very small difference in LOLE over all cumulative plant sizes and thus only the independent 1 location and 10 location LOLE curves are shown in figure 7.21. The solar plant simulations were the closest results of all the cumulative plant size simulations, however the slope as the simulations approached 10 hours per year LOLE was so low that the improvement was still more than the combined wind and solar simulations. As mentioned before, because of the constants used in these simulations the improvement may be worse than the combined wind and solar simulations if a different maximum plant size and storage size pair had been chosen.

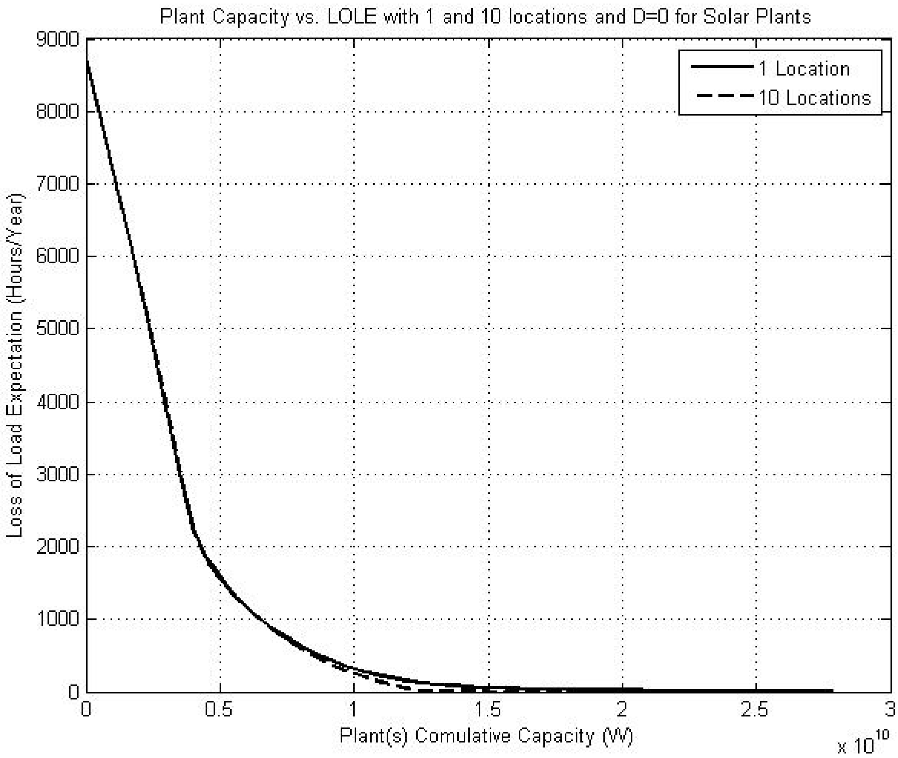

Figure 7.21: LOLE for $D = 0$ with 1 and 10 geographic locations with varying cumulative solar plant size.

The system seemed to vary with dependence level and not number of locations as can be seen again in figure 7.22. Table A.8 shows the solar plant size required for 10 hours per year LOLE assuming 10 geographic locations for all levels of dependencies. Just like previous simulation sets using solar plants, the improvement seems to be semi-linear over the dependence range and not offering an immediate significant improvement as the dependence level decreases.

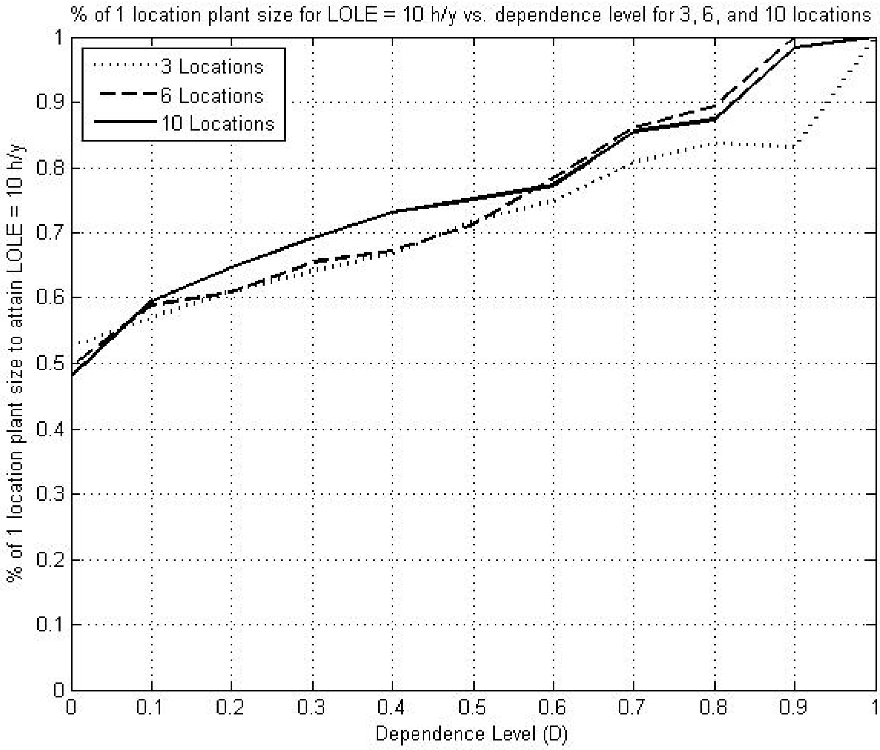

Figure 7.22: Cumulative solar plant size needed for $LOLE = 10$ hours per year vs. dependence level for 3, 6, and 10 locations.

7.3.3 Combined Wind and Solar Simulations

While using wind and solar plants the 1 location simulation arguably acts like two locations already because the solar and wind power are completely independent in these experiments. This is one of the innate benefits that combined solar and wind simulations posses. It is theorized that this causes the extremely close limits of independent 1 and 10 location graphs as seen in figure 7.23 because the single location acts more like two locations.

A smaller improvement compared to just solar or just wind is very obvious from table A.9 and figure 7.24 as the best the system improves is 60 % of the original 1 location power

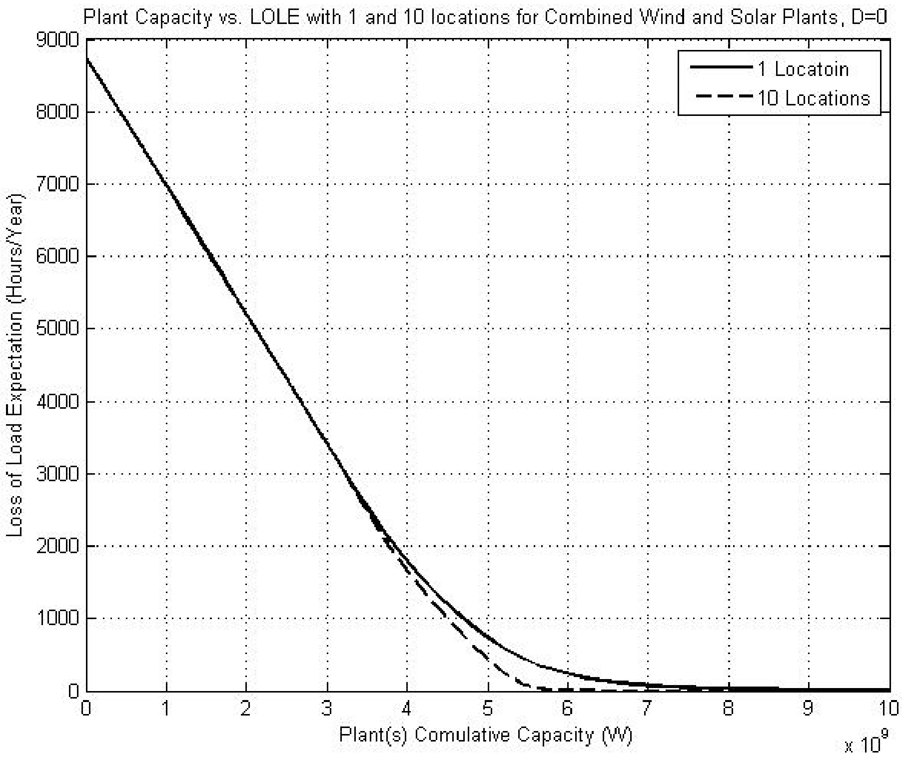

Figure 7.23: LOLE for $D = 0$ with 1 and 10 geographic locations with varying cumulative solar and wind plant size.

production capacity. What should be noted is that the physical production capacity that corresponds to a reduction of 60 % of the original level is significantly lower than either wind or solar power alone by a significant amount. This is also attained with a smaller storage size at the same storage efficiency indicating that the combination of independent wind and solar power will probably result in a lower power plant equipment cost.

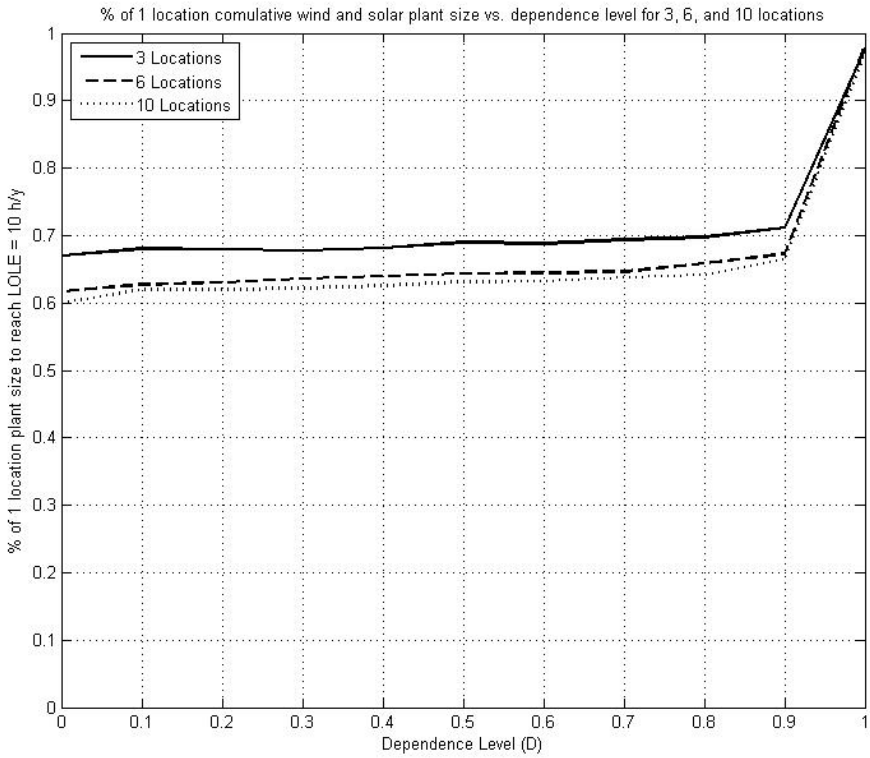

Figure 7.24: Cumulative wind and solar plant size needed for $LOLE = 10$ hours per year vs. dependence level for 3, 6, and 10 locations.

Chapter 8

Conclusions

There are many results indicating improvement of the overall system stability as more geographic locations are added and as these geographic locations' weather become more independent of one another. It is also particularly good to see that combining wind and solar power improves the system's stability as compared to using just wind or just solar power. The real world will most likely use both forms of energy generation because of geographic conditions and availability of proper weather conditions forcing a combination of solar and wind energy generation.

Unfortunately these results do not show that there will always be improvement. What they do reveal are improvements that can be made with geographic separation of renewable power plants. They do not prove that there will be improvements to any given set of geographic locations. Although complete independence allows a mathematical proof of increased stability, complete independence is highly unlikely in a real world circumstance and virtually impossible to prove. Every possible real world condition would have to be evaluated.

For further research a design related to an ARMA model should be developed that relates multiple wind and solar locations together. This would allow a site specific model to evaluate the actual improvement available. This includes time based as well as location based relationships for solar vs. solar, wind vs. wind, and solar vs. wind power plants. This model should allow a proof that some sets of locations can benefit from the geographic separation model and will do so with much more accuracy for a specific set of locations.

Incorporating a power generation facility that has innate energy storage should also be used as an extension of this model. Hydroelectric power is an example of this where energy is stored in the form of elevated water. Theoretically the energy storage size requirement could be lowered for the same LOLE possibly allowing a less expensive renewable solution.

Appendix A

Simulation Output Data and Results

Table A.1: Storage Size for LOLE = 10 hours per year using wind power.

D	1 Loc.	3 Locations		6 Locations		10 Locations	
	GWh	GWh	%	GWh	%	GWh	%
0.0		58.19	28.2	23.90	11.6	12.34	5.99
0.1		61.69	29.9	25.83	12.5	13.65	6.63
0.2		59.53	28.9	26.07	12.7	13.88	6.74
0.3		59.37	28.8	26.00	12.6	14.50	7.04
0.4		60.37	29.3	27.62	13.4	15.09	7.32
0.5	206.1	58.57	28.4	28.29	13.7	16.19	7.86
0.6		62.49	30.3	28.16	13.7	16.53	8.02
0.7		63.42	30.8	30.37	14.7	18.95	9.20
0.8		66.98	32.5	33.13	16.1	20.80	10.1
0.9		68.94	33.5	43.72	21.2	33.65	16.3
1.0		198.2	96.2	195.2	94.7	204.9	99.4

Table A.2: Storage Size for LOLE = 10 hours per year using solar power assuming 10 locations.

D	GWh	%
0.0	13.35	23.23
0.1	30.26	52.63
0.2	35.22	61.25
0.3	39.69	69.03
0.4	44.39	77.21
0.5	45.70	79.48
0.6	47.12	81.96
0.7	51.62	89.78
0.8	52.42	91.17
0.9	56.94	99.04
1.0	57.49	100

Table A.3: Storage Size for LOLE = 10 hours per year using wind and solar power.

D	1 Loc.	3 Locations		6 Locations		10 Locations	
	GWh	GWh	%	GWh	%	GWh	%
0.0		16.35	30.7	8.824	16.5	7.666	14.4
0.1		18.91	35.4	12.52	23.5	10.71	20.1
0.2		19.97	37.4	13.91	26.1	11.77	22.1
0.3		20.84	39.1	15.03	28.2	12.58	23.6
0.4		21.66	40.6	15.40	28.9	13.56	25.4
0.5	53.34	22.15	41.5	16.68	31.3	14.24	26.7
0.6		22.77	42.7	17.40	32.6	14.67	27.5
0.7		24.13	45.2	18.21	34.1	15.83	29.7
0.8		24.21	45.4	19.59	36.7	16.75	31.4
0.9		26.58	49.8	21.65	40.6	20.36	38.2
1.0		52.59	98.6	53.24	99.8	51.70	96.9

Table A.4: Storage Efficiency for LOLE = 10 hours per year using wind power.

D	1 Location % Efficiency	3 Locations % Efficiency	6 Locations % Efficiency	10 Locations % Efficiency
0.0		10.8	2.0	1.8
0.1		11.2	2.0	1.8
0.2		10.7	2.0	1.8
0.3		10.8	2.0	1.8
0.4		11.0	2.0	1.9
0.5	94.2	10.9	2.0	1.9
0.6		11.8	2.0	1.9
0.7		11.7	3.0	1.9
0.8		12.2	3.8	2.0
0.9		13.5	6.3	3.8
1.0		87.2	80.1	97.8

Table A.5: Storage Efficiency for LOLE = 10 hours per year using solar power.

D	1 Location % Efficiency	3 Locations % Efficiency	6 Locations % Efficiency	10 Locations % Efficiency
0.0		35.4	33.9	32.5
0.1		40.2	41.0	41.5
0.2		43.2	43.4	45.5
0.3		45.9	46.4	50.1
0.4		48.4	47.9	52.6
0.5	91.7	52.7	51.9	54.5
0.6		55.2	59.3	56.9
0.7		61.6	67.7	64.8
0.8		65.1	71.2	67.8
0.9		64.6	90.3	86.7
1.0		100	93.0	100

Table A.6: Storage Efficiency for LOLE = 10 hours per year using both wind and solar power.

D	1 Location % Efficiency	3 Locations % Efficiency	6 Locations % Efficiency	10 Locations % Efficiency
0.0		28.1	23.2	21.1
0.1		29.2	23.7	22.1
0.2		29.1	24.1	22.2
0.3		29.3	24.3	22.3
0.4		29.7	24.3	23.0
0.5	91.7	29.8	24.9	23.6
0.6		30.3	25.1	23.6
0.7		30.6	25.6	24.2
0.8		31.0	27.1	24.5
0.9		32.2	28.3	27.1
1.0		79.3	80.9	78.3

Table A.7: Comulative wind plant size for LOLE = 10 hours per year.

D	1 Loc.	3 Locations		6 Locations		10 Locations	
	GW	GW	%	GW	%	GW	%
0.0		10.39	43.9	9.227	39.0	8.810	37.2
0.1		10.59	44.7	9.226	38.9	8.859	37.4
0.2		10.64	44.9	9.231	39.0	8.843	37.3
0.3		10.54	44.5	9.256	39.1	8.900	37.6
0.4		10.48	44.2	9.212	38.9	8.945	37.8
0.5	23.68	10.56	44.6	9.381	39.6	9.020	38.1
0.6		10.79	45.6	9.428	39.8	9.016	38.1
0.7		10.60	44.7	9.457	39.9	9.239	39.0
0.8		10.87	45.9	9.727	41.1	9.205	38.9
0.9		10.93	46.1	10.02	42.3	9.873	41.7
1.0		23.13	97.6	22.46	94.8	24.52	103.5

Table A.8: Comulative solar plant size for LOLE = 10 hours per year assuming 10 locations.

D	GW	%
0.0	12.81	48.0
0.1	15.89	59.5
0.2	17.29	64.7
0.3	18.48	69.2
0.4	19.53	73.1
0.5	20.06	75.1
0.6	20.64	77.2
0.7	22.82	85.4
0.8	23.34	87.3
0.9	26.26	98.3
1.0	26.72	100.0

Table A.9: Comulative wind and solar plant size for LOLE = 10 hours per year.

D	1 Loc.	3 Locations		6 Locations		10 Locations	
	GW	GW	%	GW	%	GW	%
0.0		6.479	67.0	5.965	61.7	5.802	60.0
0.1		6.575	68.0	6.067	62.8	5.992	62.0
0.2		6.568	67.9	6.087	63.0	5.995	62.0
0.3		6.554	67.8	6.151	63.6	6.006	62.1
0.4		6.585	68.1	6.181	63.9	6.046	62.5
0.5	9.667	6.667	69.0	6.216	64.3	6.103	63.1
0.6		6.655	68.8	6.228	64.4	6.106	63.2
0.7		6.702	69.3	6.248	64.6	6.163	63.8
0.8		6.739	69.7	6.365	65.8	6.198	64.1
0.9		6.876	71.1	6.503	67.3	6.421	66.4
1.0		9.441	97.7	9.501	98.3	9.381	97.0

Appendix B

Matlab Code for Solar Data Generation

```
%By combining the PDF's into a multi-dimmensional PDF you can attain the
%result of multiple sites. It is assumed that all sites follow the same
%PDF.
%
% pdfs - (365xgranularity) Output PDF probabilities for any given site.
% yearly_maxs - (1x365) Daily maximums for any given day in a year (Wh/m^2).
% ratio - (1x1) How dependent the two sources should be. (between 0 and
% 1). 0 implies no dependence, 1 complete dependence.
% Sample_Count - (1x2) [Number of sample sets (M), # of points per set (N)].
% Time_Sep - (1x1) Time in hours between time samples.
% Location_cout - (1x1) Number of solar array plants there are.

% power_output - (MxN) Solar insolation going down to each meter squared of
% solar panel at the plants. (Wh/m^2).

function power_output = solar_power_prediction( pdfs,
                                                bins,
                                                yearly_maxs,
                                                ratio,
                                                Sample_Count,
                                                Time_Sep,
                                                Location_count)

prompt = 'Creating yearly perfects'
```

```
%Generate perfect days of the year (ASSUMES NE ENVIRONMENT SIMILAR TO
%WORCESTER)
Max = [72 240]; %Longest day of year
Min = [90 192]; %Shortest day of year
yearly_perfect = yearly_perfects(Max, Min, yearly_maxs);

prompt = 'Pre-generating cdfs'
%Pre-generate the cdfs to speed things up in the loop. These cdfs can only
%be used for the first set of solar array data.
pdfs_first = pdfs;
cdfs_first = pdfstocdfs(pdfs_first);

prompt = 'Defining variables before the loop.'
%Instantiate data variables before loop in order to speed up the process.
days = ceil(Sample_Count(2) * Time_Sep / 24);
data = zeros(Location_count, days * 288);
raw = zeros(Location_count, days);
power_output = zeros(Sample_Count(1), Sample_Count(2));
yearly_maxs_total = yearly_maxs;
pdfs_new = pdfs_first;
if Location_count == 1
    ratio = 1;
else
    ratio = ratio ^ (1/(Location_count - 1));
end
while size(yearly_maxs_total, 2) < days
    yearly_maxs_total = [yearly_maxs_total yearly_maxs];
end
yearly_maxs_total = yearly_maxs_total(1:days);

%Loop anything after this
for i = 1:Sample_Count(1)
    %Generate first set of elements
    raw(1, 1:days) = gen_solar_from_cdfs(days, cdfs_first);
    %Generate second and further sets of dependent elements
    %Will require generation of various pdfs/cdfs
    if(Location_count > 1)
```

```
        for j = 2:Location_count
            pdfs_new = pdf_adjuster(pdfs_new, pdfs_first, bins, ratio, raw(1:j-1, 1:days));
            cdfs_new = pdfstocdfs(pdfs_new);
            raw(j, 1:days) = gen_solar_from_cdfs(days, cdfs_new);
        end
    end
    raw = raw .* (ones(Location_count, 1) * yearly_maxs_total);

    %Generate daily 5-minute data from daily power
    data = gen_five_minute_insolation(data, raw, yearly_perfect, yearly_maxs);
    %Generate properly split "hourly specified" data and add up for all the
    %power locations.
    power_output(i, 1:Sample_Count(2)) = ...
      (1/Location_count) .* ones(1, Location_count) * ...
      conv_time_sep(Time_Sep, data, Sample_Count(2));
end;
```

```
%The point of this function is to attain an approximated daily PDF for a
%whole year based off of previous data over multiple years.
%
% yearly: yearly data in the format of a matrix 366 days "tall" and as wide
% as necesary. Last coloumn is reserved for validity counts as inevitably
% the data has holes in it from year to year. Everything should be
% compacted to the left and the validity count should represent how many
% accurate values there are in that row.
%
%days: Number of days to include in the PDF. Recommendation is to get at
%least as many as are in the smallest recurring frequency.
%
%granularity: Number of bins to use in the PDF's.
%
% pdfs - (365xgranularity) Output pdf for any given day corrsponding to the
% data given.
% bins - (1xgranularity+1) Bins that the pdf lies in.

function [pdfs, bins] = dailypdf(yearly, days, granularity, Array_Size, Array_Eff)
```

```
%Seperate the number of valid values per day and the data itself.
temp1 = size(yearly);
count = yearly(1:temp1(1), temp1(2));
temp1 = yearly(1:temp1(1), 1:temp1(2) - 1);

%attain maximums and regularize the data.
maxs = yearly_maxs(yearly, Array_Size, Array_Eff, days);
A = size(temp1);
maxs = maxs * ones(1, A(2));
temp1 = temp1 ./ maxs;

%create the pdf variable.
pdfs = zeros(1,granularity);

%Prepare data for loop.
temp2 = size(temp1);
temp1 = [temp1(size(temp1, 1) - days:size(temp1, 1), ...
  1:temp2(2)) ; temp1 ; temp1(1:days, 1:temp2(2))];
count = [count(size(count, 1) - days:size(count, 1)) ; count ; count(1:days)];

%bins creation
bins = 0:1/(granularity):1;
bins(granularity + 1) = inf;

holder = temp2(1);
%Loop per day
for rep1 = 1:holder;
    %initialize temp2
    temp2 = 0;
    %loop for grouping
    for rep2 = rep1:rep1 + 2*days + 1;
        if(count(rep2) > 0)
            temp2 = [temp2 temp1(rep2, 1:count(rep2))];
        end
    end
    %remove 0 from temp2
```

```
    temp2 = temp2(2:size(temp2, 2));

    %create pdf for that day!
    temp3 = histc(temp2, bins);
    temp4 = sum(temp3);
    pdfs = [pdfs ; (temp3(1:granularity) ./ temp4)];
end
pdfs = pdfs(2:size(pdfs, 1), 1:granularity);
```

```
% yearly_maxs is a function to determine a maximum output trend so that
% PDF's and CDF's can be created using a uniform 1 maximum.
%
% yearly - (MxN) Data in the form of a tall array for the whole year. All
% valid data is compacted "to the left" for a given day and the final
% column is given a validity count.
% days - (1x1) Number of days to check forward and backwards to find
% maximum.
% Array_Size - (1x1) Size of array that gave "yearly" (m^2)
% Array_Eff - (1x1) Efficiency of array that gave "yearly" (Unitless,
% Efficiency).
%
% maxs - (Mx1) - Maximums for any particular day of the year (based on
% input). (Wh/M^2)

function maxs = yearly_maxs(yearly, Array_Size, Array_Eff, days)

%Creating a maximum size vector for individual days.
temp = size(yearly);
%Strip off the pertinent data count at the end
temp = yearly(1:temp(1), 1:(temp(2)-1));
%Find the maximums for all the years.
temp = max(temp')';

%Finding max over days represented by left/right.
maxs = zeros(1, length(temp));
temp = [temp((length(temp)-days):(length(temp))) ; temp ; temp(1:days)];
```

```
for rep1 = 1:length(maxs);
    maxs(rep1) = max(temp((rep1):(rep1+2*days+1)));
end
maxs = maxs';
maxs = maxs / Array_Size;
maxs = maxs / Array_Eff;
```

```
%This function is intended to produce a set of perfect yearly solar
%insolation graphs. The time seperation per unit is 5 minutes.

% Max - (1x2) Maximum day spread of time.
%       Start time first. (Time of day in 5-minute increments)
% Min - (1x2) Minimum day spread of time.
%       Start time first.(Time of day in 5-minute increments)
% Daily_Maxs - (1x365) Maximum power for a perfect non-cloudy day. (Wh/m^2)

%yearly_perfects - (288x365) Power output in 5-minute increments. (Wh/m^2)

function yearly_perfects = yearly_perfects(Max, Min, Daily_Maxs)

%Step 1: Create ideal power for a whole year.
ideal_power = [1:1:288]' * ones(1,365); %5 minute increments by each day of year.

%ratios = [min distance from longest day ; min distance from shortest day];
ratios = [min([abs([1:1:365] - 172) ; abs([1:1:365] - 537)]) ; ...
  min([abs([1:1:365] - 356) ; abs([1:1:365] + 9)])];
%ratios = [ratio of shortest day to use ; ratio of longest day to use];
ratios = [ratios(2, 1:365) ./ (ratios(1, 1:365) + ratios(2, 1:365)) ; ...
  ratios(1, 1:365) ./ (ratios(1, 1:365) + ratios(2, 1:365))];

%hours = [start of sunlight (365 days) ; end of sunlight (365 days)];
hours = [ratios(1, 1:365) .* Max(1) + ratios(2, 1:365) .* ...
  Min(1) ; ratios(1, 1:365) .* Max(2) + ratios(2, 1:365) .* Min(2)];

%difference = [number of 5-minute increments sunlight is available]
difference = (hours(2,1:365) - hours(1, 1:365));
```

```
%difference is propogated through all 5-minute increments for later math.
difference = ones(288, 1) * difference;
%minimum = starting point of sunlight.
minimum = ones(288, 1) * hours(1, 1:365);

%ideal_power = sin wave with amplitude according to daily_maxs, frequency
%dependent on difference, and offset based on minimum.
ideal_power = sin((2 * pi ./ (2 * difference)) .* (ideal_power - minimum));

%Clear out upper and lower sine wave repeats near midnight.
hours(1, 1:365) = floor(hours(1, 1:365));
hours(2, 1:365) = ceil(hours(2, 1:365));
blockout = (1:288)' * ones(1, 365);
for i = 1:365
    blockout(1:288, i) = (blockout(1:288, i) > hours(1, i)) .* ...
      (blockout(1:288, i) < hours(2, i));
end
yearly_perfects = ideal_power .* (ideal_power > 0) .* blockout;
for i = 1:365
    yearly_perfects(1:288, i) = yearly_perfects(1:288, i) .* ...
      Daily_Maxs(i) ./ sum(yearly_perfects(1:288, i));
end
%([zeros(Max(1),365) ; ones(Max(2) - Max(1), 365) ; zeros(288 - Max(2), 365)]);
```

```
%The purpose of this function is to create a random set of solar insolation
%data based upon the cdfs provided. Always starts from day 0 of the year
%(January 1st). Assumes every year is not a leap year.

% predictions - (1x1) Number of days to predict solar insulation for.
% (days)
% cdfs - (NxM) cdfs of the solar insolation (typically 365xgranularity).

% raw_data = (1xpredictions) Random solar data (before daily_maxs is
% applied) (%)

function raw_data = gen_solar_from_cdfs(predictions, cdfs)
```

```
raw_data = zeros(1, predictions);
random_day = rand(1, predictions);

A = size(cdfs);
A = A(2);
B = size(cdfs);
k = 0;
for g = 1:predictions
    k = k + 1;
    raw_data(g) = interp1(cdfs(k, 1:A), 0:1/(A - 1):1, random_day(g));
    if k == B(1)
        k = 0;
    end
end;
```

```
%The purpose of this function is to convert any given pdf of a single
%dimmension into a cdf.

%pdfs - (MxN) pdf of something (N seperate bins, M cases).

%cdfs - (MxN+1) cdf corresponding to pdf (N seperate bins, M cases).

function cdfs = pdfstocdfs(pdfs)

A = size(pdfs);
multiply = ones(A(2));
multiply = triu(multiply, 1);
cdfs = [pdfs * multiply pdfs * ones(A(2), 1)];

for i = 1:size(cdfs, 1)
    for j = 2:size(cdfs, 2)
        if cdfs(i, j - 1) >= cdfs(i, j)
            cdfs(i, j) = cdfs(i, j - 1) + 1/10000000;
        end
    end
    cdfs(i, 1:size(cdfs, 2)) = cdfs(i, 1:size(cdfs, 2)) ./ cdfs(i, size(cdfs, 2));
```

```
end
```

```
%This function is intended to generate a set of pdfs for a year given the
%raw data from other somewhat-dependent variables.

% new_pdfs - (GxM) Pass in the output variable so a new copy in memory is
% not instantiated every time.
% original_pdfs - (365XM) pdf's that each solar array follows.
% bins - (1xM) Bins that was used to create the pdf.
% ratio - (1x1) Ratio of dependence (1 is full dependence, 0 is independent).
% givens - (NxG) Raw 0-1 ratio of maximum power out. [givens per day, day]

% new_pdfs - (GxM) new PDF with givens applied.

function new_pdfs = pdf_adjuster(new_pdfs, original_pdfs, bins, ratio, givens)

A = size(givens);
B = size(original_pdfs);

for g = 1:A(2)
    temp = histc(givens(1:A(1), g), bins) * ratio / A(1);
    new_pdfs(g, 1:B(2)) = temp(1:B(2));
end

while size(original_pdfs, 1) < size(new_pdfs, 1)
   original_pdfs = [original_pdfs ; original_pdfs];
end

A = size(new_pdfs);
new_pdfs = original_pdfs(1:A(1), 1:A(2)) .* (1-ratio) + new_pdfs;
```

```
%This function is intended to take the power for a day, the yearly perfect
%representations (in 5 minute increments), and generate solar data that has
%the proper Wh/m^2 for the given day.
```

```
% solar_insolation - (Mx288*N) Pass in the output variable so a new copy in memory is
% not instantiated every time.
% daily - (MxN) Daily Solar Insolation (Wh/m^2)
% perfects - (288x365) Perfect solar insolation for any given 5 minute
% increment of the year. (Wh/m^2)
% Daily_Maxs - (365x1) Daily maximum power output (Wh/m^2)

% solar_insolation - (Mx288*N) Solar insolation data output in 5-minute
% increments (Wh/m^2)

function solar_insolation = gen_five_minute_insolation( solar_insolation,
                                                         daily,
                                                         perfects,
                                                         Daily_Maxs)
C = size(daily);
J = 0;

for I = 1:C(2)
    J = J + 1;
    if J == 366
        J = 1;
    end
    this_day = ones(C(1), 1) * perfects(1:288, J)';
    for B = 1:C(1)
        if(daily(B, I) <= .01 * Daily_Maxs(J))
            this_day(B, 1:288) = this_day(B, 1:288) * (daily(B)/Daily_Maxs(J));
        else
            if(daily(B, I) <= .1 * Daily_Maxs(J))
                this_day(B, 1:288) = this_day(B, 1:288) * .1;
            end
            while sum(this_day(B, 1:288)) > daily(B, I)
                deduct = randint(1,1,[1,288]);
                this_day(B, deduct) = this_day(B, deduct) .* rand;
            end
            %Scale lost watts to equal prediction
            if(sum(this_day(B, 1:288)) < daily(B, I))
                this_day(B, 1:288) = ...
```

```
                this_day(B, 1:288) + (perfects(1:288, J)' - ...
                this_day(B, 1:288)) .* ((daily(B, I) − sum(this_day(B, 1:288))) ...
                / (Daily_Maxs(J) − sum(this_day(B, 1:288))));
            end
        end
    end
    solar_insolation(1:C(1), 288 * (I − 1) + 1:288 * I) = this_day;
end
```

```
% The purpose of this function is to convert a time series 5-minute data
% into another time series data. (MUST BE IN INCREMENTS OF 5 MINUTES).

% Time_Sep - (1x1) Time seperation between data points desired.
% solar_insolation_in - (MxN) 5-minute solar insolation input.
% length - (1x1) Number of data points that are being kept.

% solar_insolation_out - (Mxlength) Solar insolation output in Time_Sep
% increments.

function solar_insolation_out = conv_time_sep(Time_Sep, solar_insolation_in, length)

A = size(solar_insolation_in);
solar_insolation_out = solar_insolation_in(1:A(1), 1:length);
seperation = 12 * Time_Sep;
if seperation ~= 1
    for B = 1:length
        solar_insolation_out(1:A(1), B) = ...
        sum(solar_insolation_in(1:A(1), ((B−1) * seperation + 1) : B * seperation)')';
    end
end
```

Appendix C

Matlab Code for Wind Data Generation

```
% This function is meant to produce random wind generated power based off
% of a cutin wind, cutout wind, rated wind, and the weather charactaristics
% of that area.

% pdf_bins_edges - (365xM) edges of the bins that the pdf's are based off
% of for each day of the year. (m/s)
%
% year_wind_pdf - (1xM-1) probability for any given day that the wind
% will be within certain bins. A certain distribution is assumed and scaled
% accordng to the bins. (probability, unitless)
%
% diurnal_average_normalized - (1x24) the average daily wind distributed
% over each hour so that the average value over the day is 1 m/s.
%
% Time_Sep - (1x1) Number of hours for each data point. (hours)
%
% data_size - (1x2) [M, N] Size of output data [# of samples, Data set from one
% sample]
%
% locations - (1x1) Number of locations to simulate for each sample. (#
% locations)
%
% dependence - (1x1) Probability of having all locations being the same
% (probability, unitless)
```

```
%
% cutin_wind - (1x1) Wind speed at which the wind turbine will begin
% generating electricity (m/s)
%
% rated_wind - (1x1) Wind speed that the wind turbine is rated for (m/s)
%
% cutout_wind - (1x1) Wind speed that the generator shuts down for safety
% reasons (m/s)

% power_output - (M, N) Power output for each Time_Sep time increment
% in each sample. ((Wh/(W of wind power))

function power_output = gen_wind( pdf_bins_edges,
                                  year_wind_pdf,
                                  diurnal_average_normalized,
                                  Time_Sep,
                                  data_size,
                                  locations,
                                  dependence,
                                  cutin_wind,
                                  rated_wind,
                                  cutout_wind)

days = ceil(data_size(2) * Time_Sep / 24);
year_wind_pdf = ones(365, 1) * year_wind_pdf;
year_wind_cdf = pdfstocdfs(year_wind_pdf);

if locations == 1
    ratio = 1;
else
    ratio = dependence ^ (1/(locations - 1));
end

wind_power_profile = [-1, cutin_wind, rated_wind, ...
  cutout_wind, cutout_wind + 10^-9, 1000 ; 0, 0, 1, 1, 0, 0];

power_output = zeros(data_size);
```

```
random_day_wind = zeros(locations, days);
year_wind_pdf_new = 0;
pdf_bins_new = 0;
for i = 1:data_size(1)

    random_day_wind(1, 1:days) = ...
      gen_wind_from_cdfs(days, year_wind_cdf, pdf_bins_edges);
    if(locations > 1)
        for j = 2:locations
            [year_wind_pdf_new, pdf_bins_new] = ...
              pdf_adjuster(year_wind_pdf_new, year_wind_pdf, ...
              pdf_bins_new, pdf_bins_edges, ratio, ...
              random_day_wind(1:(j−1), 1:days));
            random_day_wind(j, 1:days) = ...
              gen_wind_from_cdfs(days, pdfstocdfs(year_wind_pdf_new), pdf_bins_new);
        end
    end
    random_time_sep_wind = days_to_time_sep(diurnal_average_normalized, ...
      Time_Sep, random_day_wind);

    for j = 1:locations
        random_time_sep_wind(j, 1:data_size(2)) = ...
          interp1(wind_power_profile(1, 1:6), wind_power_profile(2, 1:6), ...
          random_time_sep_wind(j, 1:data_size(2)));
    end
    power_output(i, 1:data_size(2)) = ones(1, locations) * ...
      random_time_sep_wind(1:locations, 1:data_size(2)) .* 1/locations .* Time_Sep;
end
```

```
%This function is intended to generate a set of pdfs for a year given the
%raw data from other somewhat-dependent variables.M̂
% new_pdfs - (GxM) Pass in the output variable so a new copy in memory is
% not instantiated every time.
% original_pdfs - (365XM) pdf's that each solar array follows.
% bins - (1xM) Bins that was used to create the pdf.
% ratio - (1x1) Ratio of dependence (1 is full dependence, 0 is independent).
```

```
% givens - (NxG) Raw wind speed output. [givens per day, day]

% new_pdfs - (GxM) new PDF with givens applied.

function [new_pdfs, bins_new] = pdf_adjuster(new_pdfs, original_pdfs, bins_new, bins, ratio, givens)

A = size(givens);
B = size(original_pdfs);
C = size(bins);
k = 0;
for g = 1:A(2)
    k = k + 1;
    temp = histc(givens(1:A(1), g), bins(k, 1:C(2))) * ratio / A(1);
    new_pdfs(g, 1:B(2)) = temp(1:B(2));
    if(k == C(1))
        k = 0;
    end
end

while size(original_pdfs, 1) < size(new_pdfs, 1)
    original_pdfs = [original_pdfs ; original_pdfs];
end

bins_new(1:C(1), 1:C(2)) = bins;
while size(bins_new, 1) < size(new_pdfs, 1)
    bins_new = [bins_new ; bins];
end

A = size(new_pdfs);
new_pdfs = original_pdfs(1:A(1), 1:A(2)) .* (1-ratio) + new_pdfs;
bins_new = bins_new(1:A(1), 1:C(2));
```

```
%The purpose of this function is to convert any given pdf of a single
%dimmension into a cdf.

%pdfs - (MxN) pdf of something (N seperate bins, M cases).
```

```
%cdfs - (MxN+1) cdf corresponding to pdf (N seperate bins, M cases).

function cdfs = pdfstocdfs(pdfs)

A = size(pdfs);
multiply = ones(A(2));
multiply = triu(multiply, 1);
cdfs = [pdfs * multiply pdfs * ones(A(2), 1)];

for i = 1:size(cdfs, 1)
    for j = 2:size(cdfs, 2)
        if cdfs(i, j - 1) >= cdfs(i, j)
            cdfs(i, j) = cdfs(i, j - 1) + 1/10000000;
        end
    end
    cdfs(i, 1:size(cdfs, 2)) = cdfs(i, 1:size(cdfs, 2)) ./ cdfs(i, size(cdfs, 2));
end
```

```
%The purpose of this function is to create a random set of wind speed
%data based upon the cdfs provided. Always starts from day 0 of the year
%(January 1st). Assumes every year is not a leap year.

% predictions - (1x1) Number of days to predict wind speed for.
% (days)
% cdfs - (365xM) cdfs of the wind speed (typically 365xgranularity).
% bin_edges - (365xM)

% raw_data = (1xpredictions) Random wind data as average wind speed over
% that day. (m/s average for a day)

function raw_data = gen_wind_from_cdfs(predictions, cdfs, bin_edges)
raw_data = zeros(1, predictions);
random_day = rand(1, predictions);

A = size(bin_edges);
```

```
k = 0;
for g = 1:predictions
    k = k + 1;
    raw_data(g) = interp1(cdfs(k, 1:A(2)), bin_edges(k, 1:A(2)), random_day(g));
    if k == A(1)
        k = 0;
    end
end;
```

```
%The purpose of this function is to take the wind predictions and create a
%typical average day with that average wind speed.

% day_pattern - (1x24) Average wind during the day for each hour
% normalized. (m/s average per hour normalized)
% Time_Sep - (1x1) amount of time for each average placement (hours)
% predictions - (1xM) Average wind for all days (m/s average over a day
% each)

% raw_data = (1xM * 24 / Time_Sep) Time_Sep seperated average wind speed
% data.

function raw_data = days_to_time_sep(day_pattern, Time_Sep, predictions)

day_sep = 24 / Time_Sep;

raw_data = zeros(size(predictions, 1), size(predictions, 2) * day_sep);

day_pattern_adjust = zeros(1, day_sep);
for j = 0:23
    for i = 1:(1/Time_Sep)
        day_pattern_adjust(j / Time_Sep + i) = day_pattern(j + 1);
    end
end
for j = 1:size(predictions, 1)
    for i = 1:size(predictions, 2)
        raw_data(j, ((i - 1) * day_sep + 1):(i * day_sep)) = ...
```

```
        day_pattern_adjust * predictions(j, i);
    end
end
```

Appendix D

Matlab Code for Load Data Generation

```
%means, standard_deviations, typical_day

Days = 365*4;
Time_Sep = .25;
Points = Days * 24 / Time_Sep;
cases = 1100;
Worcester_Percent = 784992/14239724;

%Create load for all of NE
random_daily_load = create_daily_load(means .* 10^6, standard_deviations .* 10^6, [cases, Days]);
%Then divide it by approximate Worcester load
random_daily_load = random_daily_load .* Worcester_Percent;

%Pre-allocate output for speed.
load_data = zeros(cases, Points);
for i = 1:Days
    for j = 1:cases
        load_data(j, ((i - 1) * 24 / Time_Sep + 1):(i * 24 / Time_Sep)) = ...
          typical_day .* random_daily_load(j, i);
    end
end
```

```
%function create_daily_load assumes normally distributed load data and generates
```

```
%daily loads specified by the input arguments.

% means - (1x365) means of various days. (Wh)
% stds - (1x365) Standard deviation of various days. (Wh)
%Size - (1x2) Size of final array [M, N] (unitless)

%daily_load_data - (MxN) Random load based on the means and standard
%deviations. (Wh)

function daily_load_data = create_daily_load(means, stds, size)
daily_load_data = zeros(size);
J = 0;
for i = 1:size(2)
    J = J + 1;
    daily_load_data(1:size(1), i) = normrnd(means(J), stds(J), [size(1), 1]);
    if(J >= 365)
        J = 0;
    end
end
```

```
%function Convert_hours_quarter_hours

% in - (1x24) Daily expected load. (% daily output)

% out - (1x96) Daily expected load (% daily output)

function out = Convert_hours_quarter_hours(in)

out = 0;

for i = 1:size(in, 2)
    out = [out, in(i) .* ones(1, 4) .* .25];
end
out = out(2:size(out, 2));
```

Appendix E

Matlab Code for Monte Carlo Simulations

```
%Currently the point of this function is to take in solar data,
% solar array information, energy storage information, and load
% information and return the load mismatch for every data point.
%
%
% System_Power_In - (MxN) System power in (Wh)
% Time_Sep - (1x1) Time Seperation in Solar_Insolation and Load Data
% Measurements (Hours)
% Storage_Max - (1x1) Storage Maximum Charge (Wh)
% Storage_Eff - (1x1) Energy Storage Efficiency (Unitless)
% Load - (MxN) Load Data (Wh)
%
% LOL - (Mx1) Loss of Load Results (Hours total)

function [LOL] = balance_predict( System_Power_In,
                                  Time_Sep,
                                  Storage_Max,
                                  Storage_Eff,
                                  Load)


[M,N] = size(System_Power_In);

%Determine load mismatch
```

```
Load_Mismatch = System_Power_In - Load;

%Filter through storage element

Current_Charge = zeros([M (N+1)]);
Current_Charge(1:M, 1) = ones(M,1) * (Storage_Max / 2);
Load_Mismatch = Load_Mismatch .* ((Load_Mismatch < 0) + ...
  (Load_Mismatch >=0) .* Storage_Eff);

for i = 1:N
    Current_Charge(1:M,i+1) = Current_Charge(1:M,i) + Load_Mismatch(1:M,i);
    undercharge = ((Current_Charge(1:M,i+1) < 0).*Current_Charge(1:M,i+1));
    overcharge = ((Current_Charge(1:M,i+1) > Storage_Max).* ...
      (Current_Charge(1:M,i+1)-Storage_Max));
    Load_Mismatch(1:M,i) = undercharge+overcharge;
    Current_Charge(1:M,i+1) = Current_Charge(1:M,i+1) - Load_Mismatch(1:M,i);
end

if(Storage_Eff > 0)
    Load_Mismatch = Load_Mismatch .* ((Load_Mismatch < 0) + ...
      (Load_Mismatch >= 0) ./ Storage_Eff);
end



Load_Zeros = Load == 0;
Load = Load + Load_Zeros;
LOL = (Load_Mismatch .* (Load_Mismatch < 0)) ./ Load;
LOL = (Load_Zeros == 0) .* LOL;

LOL = LOL .* Time_Sep;

length = size(LOL);
length = length(2);

LOL = LOL * ones(length, 1) * -1;
LOL = 8760 * LOL / (N*Time_Sep);
```

Bibliography

[1] R. Billinton and R. Karki, "Maintaining supply reliability of small isolated power systems using renewable energy," *IEE Proc.-Gener. Transm. Distrib.*, vol. 148, no. 6, pp. 530–534, 2001.

[2] Y. Gao, "Adequacy assessment of electric power systems incorporating wind and solar energy," Master's thesis, University of Saskatchewan, Saskatoon, Canada, January 2006.

[3] R. Karki, P. Hu, and R. Billinton, "A simplified wind power generation model for reliability evaluation," *IEEE Transactions on Energy Conversion*, vol. 21, no. 2, pp. 533–540, 2006.

[4] (2006) Liberty. Product Brochure. Carpinteria, California. [Online]. Available: http://www.clipperwind.com/pdf/liberty_brochure.pdf

[5] J. F. Manwell, A. F. Ellis, and M. Dua. (2004, October) Wind data report draft copy ipswich. Internet draft. Renewable Energy Research Laboratory, University of Massachusetts, Amherst. [Online]. Available: http://74.94.169.139/utilities/wind/WIND%20DATA%20REPORT%20Yearly.pdf

[6] ——. (2004, December) Wind resource data interpretation report ipswich. Renewable Energy Research Laboratory, University of Massachusetts, Amherst. Amherst, MA. [Online]. Available: http://74.94.169.139/utilities/wind/Ipswich%20Report.pdf

[7] A. Dimitrovski and K. Tomsovic, "Impact of wind generation uncertainty on generating capacity adequacy," in *9th International Conference on Probabilistic Methods Applied to Power Systems KTH*, Stockholm, Sweden, Jun.11–15, 2006, pp. 1–6.

[8] R. Perez, R. Margolisa, M. Kmiecik, M. Schwab, and M. Perez, "Update: Effective load-carrying capability of photovoltaics in the united states preprint," in *Solar 2006 Conference*, Golden Colorado, USA, Jul.8–13, 2006, pp. 1–6.

[9] H. Asano and S. Bando, "Load fluctuation analysis of commercial and residential customers for operation planning of a hybrid photovoltaic and cogeneration system," in *Power Engineering Society General Meeting, 2006. IEEE*, Los Alamitos, CA, June 2006, pp. 1–6.

[10] P. Denholm and R. Margolis, "Very large-scale deployment of grid-connected solar photovoltaics in the united states: Challenges and opportunities preprint," in *Solar 2006 Conference*, Golden Colorado, USA, Jul.8–13, 2006, pp. 1–5.

[11] R. W. Boom, B. C. Haimson, G. E. McIntosh, H. A. Peterson, and W. C. Young, "Superconductive energy storage for large systems," *IEEE Transactions on Magnetics*, vol. MAG-11, no. 2, pp. 475–481, 1975.

[12] Y. Wang, S. G. Rupani, M. LaBossiere, D. C. Brande, and R. F. Vaz, "Solar energy and photovoltaics education in worcester," unpublished.

[13] M. Arner, private communication, 2007.

[14] R. Billinton, H. Chen, and R. Ghajar, "Time-series models for reliability evaluation of power systems including wind energy," *Microelectronics and Reliability*, vol. 36, no. 9, pp. 1253–1261, 1996.

[15] R. Karki and P. Hu, "Wind power simulation model for reliability evaluation," in *Electrical and Computer Engineering, 2005. Canadian Conference on*, Los Alamitos, CA, 2005, pp. 541–544.

[16] (2007, December) System loads in eei format 2000s. [Online]. Available: http://www.iso-ne.com/markets/hstdata/hourly/syslds_eei/index.html

[17] (2007, December) U.S. Census Bureau Fact Finder. [Online]. Available: http://factfinder.census.gov/home/saff/main.html?_lang=en