

E.T.S. de Ingeniería Industrial,  
Informática y de Telecomunicación

# Optimización de la potencia eólica ofertada en el mercado diario considerando escenarios de generación y precio estocásticos



Máster Universitario en  
Energías Renovables: Generación Eléctrica

Trabajo Fin de Máster

Autor: Ander Cilveti Álvarez

Director: Alberto Berrueta Irigoyen  
Pamplona, noviembre 2020

upna  
Universidad Pública de Navarra  
Nafarroako Unibertsitate Publikoa

---

## RESUMEN

---

En el presente documento se analiza y simula mediante herramientas de software los efectos de considerar las variables involucradas en los mercados eléctricos estocásticas para la optimización de la venta de energía eléctrica de origen eólico.

Para ello se realizan la abstracción de modelos ARIMA de las variables consideradas como son el precio de la electricidad en el mercado diario, precio del mercado de desvíos y viento, y mediante el uso de la técnica de simulaciones mediante el método Montecarlo usada con los modelos ARIMA, se generan los escenarios. A continuación, se realiza la reducción de escenarios, para finalmente hacer la resolución del problema de optimización estocástico.

---

## LISTA DE PALABRAS CLAVE

---

Mercados eléctricos, programación estocástica, generación escenarios, modelos ARIMA, series temporales.

---

## ABSTRACT

---

At the present document it will be analyzed and simulated the effect of taking into consideration the stochastic nature of the main variables involving the electrical market to achieve an optimization of the electric eolic energy selling.

In order to achieve the objective, there will be created ARIMA models of the variables, namely day ahead market, balancing market and wind. Using the Montecarlo simulation method there will be generated the scenarios using the created ARIMA models. Afterwards it will be reduced the number of scenarios to have into consideration, to finish with the resolution of the stochastic optimization problem.

---

## KEYWORDS

---

electrical markets, stochastic programming, scenario generation, ARIMA models, temporal data series.

## ÍNDICE

RESUMEN .....	2
LISTA DE PALABRAS CLAVE .....	2
ABSTRACT .....	3
KEYWORDS .....	3
ÍNDICE.....	4
ÍNDICE DE FIGURAS.....	5
1. INTRODUCCIÓN.....	7
1.1. SITUACIÓN A ANALIZAR .....	7
1.2. RETOS AFRONTADOS EN LA RESOLUCIÓN DEL PROBLEMA.....	9
2. ESTADO DEL ARTE .....	11
2.1. ESTRUCTURA TÉCNICA DEL TRABAJO. ....	12
3. HERRAMIENTAS MATEMÁTICAS UTILIZADAS .....	13
3.1. GENERACIÓN DE ESCENARIOS.....	13
3.1.1. MODELO ARIMA. ....	13
3.1.2. SIMULACIÓN MONTECARLO. ....	14
3.2. REDUCCIÓN DE ESCENARIOS.....	16
3.3. RESOLUCIÓN PROBLEMA ESTOCÁSTICO.....	18
4. DATOS DE PARTIDA.....	19
4.1. SERIE TEMPORAL DE VIENTO .....	19
4.2. NÚMERO Y TECNOLOGÍA DE LOS AEROGENERADORES .....	22
4.3. SERIE TEMPORAL DE PRECIO EN EL MERCADO DIARIO.....	22
4.4. SERIE TEMPORAL DE RATIO DE PRECIO ENTRE PRECIO DE BALANCE Y PRECIO DIARIO. ..	23
4.5. FORMATO DE MERCADO ASUMIDO.....	24
4.6. ASUNCIONES SOBRE VARIABLES A ANALIZAR. ....	26
5. IMPLEMENTACIÓN DEL ALGORITMO EN MATLAB .....	27
5.1. GENERACIÓN Y REDUCCIÓN DE ESCENARIOS DE VIENTO. ....	27
5.2. GENERACIÓN Y REDUCCIÓN ESCENARIOS PRECIO MERCADO DIARIO. ....	40
5.3. GENERACIÓN Y REDUCCIÓN ESCENARIOS PRECIO MERCADO DE BALANCE.....	41
5.4. RESOLUCIÓN DEL PROBLEMA ESTOCÁSTICO. ....	44
5.4.1. CREACIÓN DE VECTORES DE ESCENARIOS A PARTIR DE VALORES CONSEGUIDOS DE CADA VARIABLE ESTOCÁSTICA. ....	44
5.4.2. SOLUCIONAR PROBLEMA BASE.....	45
6. RESULTADOS OBTENIDOS. DISCUSIÓN.....	48
6.1. RESOLUCIÓN AL PROBLEMA ESTOCÁSTICO BASE. ....	48
6.2. EJEMPLO DE APLICACIÓN PARA TESTEAR CONTROL DE BATERÍAS. ....	48
7. CONCLUSIÓN.....	51
8. LÍNEAS FUTURAS .....	52
9. BIBLIOGRAFÍA .....	53

## ÍNDICE DE FIGURAS

Figura 1 Generación vs Consumo eléctrico [4]	8
Figura 2 Cruce curva de generación y demanda	8
Figura 3 Solución esquemática del problema	12
Figura 4 Aplicación método Montecarlo $t = 5$ y 20 trayectorias	15
Figura 5 Aplicación método Montecarlo $t = 5$ y 2 trayectorias	16
Figura 6 Árbol de escenarios original	17
Figura 7 Árbol de escenarios de después de la reducción de escenarios	17
Figura 8 Procesamiento Inputs	20
Figura 9 Distribución de Weibull	21
Figura 10 Serie de datos de viento	21
Figura 11 Curva de potencia aerogenerador	22
Figura 12 Precio mercado spot diario España	23
Figura 13 Precio del desvío medido entre precio marginal diario a subir y precio del desvío medido entre precio marginal diario a bajar	24
Figura 14 Esquema programación estocástica a realizar	25
Figura 15 Código para la generación del perfil de viento	27
Figura 16 Serie de datos generada	28
Figura 17 Transformación de la velocidad de viento a una función estándar normal	29
Figura 18 Código para realizar transformación de Weibull a una función normal	29
Figura 19 Valores conseguidos de vientos	30
Figura 20 Valores normalizados	30
Figura 21 Código para el modelo ARIMA	31
Figura 22 Resultado ajuste ARMA_mdl	32
Figura 23 Resultado ARMA_est	33
Figura 24 Código modelo Montecarlo	33
Figura 25 Representación de 100 escenarios de viento diferentes con 23 pasos	34
Figura 26 Código para la transformación de distribución normal a Weibull	34
Figura 27 Representación $v\_gen\_scenari$	35
Figura 28 Código para la reducción de escenarios	36
Figura 29 Matriz valores S y P	36
Figura 30 Representación gráfica valores S y P	37
Figura 31 Valores medios de energía eólica observados en el mercado diario	38
Figura 32 Código escenario día en adelante	39
Figura 33 Código para el cálculo de potencia generada	39
Figura 34 Código generación de precios del mercado diario	40
Figura 35 Escenarios después de la reducción	41
Figura 36 formulación de la variable $r_t$	42
Figura 37 Precio del desvío medio entre precio marginal diario a subir y a bajar	43
Figura 38 Código escenarios de ratios de precio	44
Figura 39 Código para la creación de vectores de escenarios	45
Figura 40 Expresión del beneficio económico por venta de energía	46
Figura 41 Código para el cálculo de beneficios	46
Figura 42 Beneficios obtenidos para cada Pd	47
Figura 43 Código para optimizar y crear controles de almacenamiento para unidades generadoras	49
Figura 44 Beneficios obtenidos para cada Pd con y sin baterías	49



## 1. INTRODUCCIÓN

### 1.1. Situación a analizar

Dentro de los diferentes desarrollos tecnológicos energéticos usados en la actualidad, las denominadas energías renovables, tienen como aspecto más característico que las define el recurso renovable que usan como fuente de energía. El recurso renovable es usado por soluciones tecnológicas como son los paneles fotovoltaicos, aerogeneradores, etc. que transforman la energía contenida en los diferentes recursos renovables (irradiación solar, viento, etc.) para transformarla en otro tipo de energía, en la mayoría de los casos, energía eléctrica.

A su vez, el recurso renovable tiene diferentes aspectos que lo definen. Uno de los más característicos, y que para el desarrollo de este trabajo supone una de las razones de ser, es su naturaleza no determinística. El recurso renovable, a día de hoy, no es un recurso controlable por el ser humano. No hay soluciones tecnológicas que consigan controlar la cantidad de irradiación solar o velocidad del viento que se puede conseguir para un emplazamiento concreto. De esta manera, el recurso renovable está a merced de la naturaleza [1].

Sin embargo, no es correcta la afirmación de que el recurso renovable es algo impredecible, ya que es posible predecir, aceptado márgenes de error definidos, diferentes posibles valores. Es decir, el recurso renovable tiene naturaleza estocástica, al contrario que los recursos deterministas de las industrias energéticas tradicionales.

Este aspecto estocástico es uno de los factores más limitantes para la integración de las energías renovables en el paradigma energético eléctrico actual, el cual está diseñado casi en su totalidad por procesos pensados para energías deterministas.

Sin embargo, con el aumento de rentabilidad asociado al uso de energías renovables, hace que la variabilidad del recurso sea un problema que cobra cada vez más importancia a medida que se instalan más megavatios “verdes”, siendo ya para organizaciones o empresas del sector una ventaja competitiva el hecho de gestionar mejor la aleatoriedad del recurso [2].

Por otro lado, se debe considerar el otro lado de la moneda. Si bien ya se explica la “estocasticidad” del recurso renovable, también es cierto que la demanda del comúnmente denominado *pool* eléctrico no es determinista, y al igual que la generación tiene una naturaleza estocástica (Figura 1).

Si bien se puede definir de manera bastante precisa el consumo que va a tener una región basándose en comportamientos típicos y datos históricos [3], el comportamiento correspondiente al consumo eléctrico es intrínsecamente no determinista debido a que también está constituido por elementos aleatorios, volviendo de esta manera a la idea anterior, el consumo eléctrico tiene naturaleza estocástica.

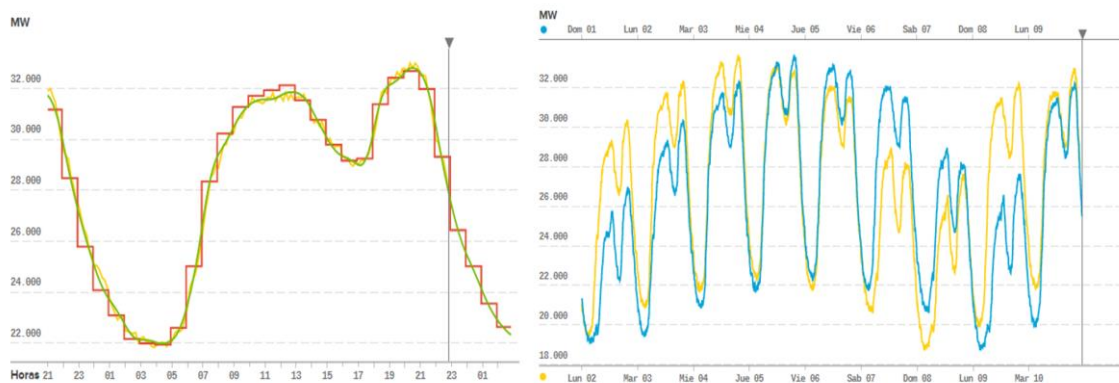


Figura 1 Generación vs Consumo eléctrico [4]

Teniendo en cuenta el carácter estocástico de la generación y consumo (Figura 1), es entendible que en aquellas variables que sean obtenidas del cruce de las dos variables, también va a tener un carácter estocástico [2].

Para el caso del mercado español [5], el cual es el caso analizado en este trabajo, la determinación del precio de la energía se hace mediante el cruce de las curvas de generación y demanda (Figura 2) del pool, con lo que como es de entender, también tiene un componente estocástico.

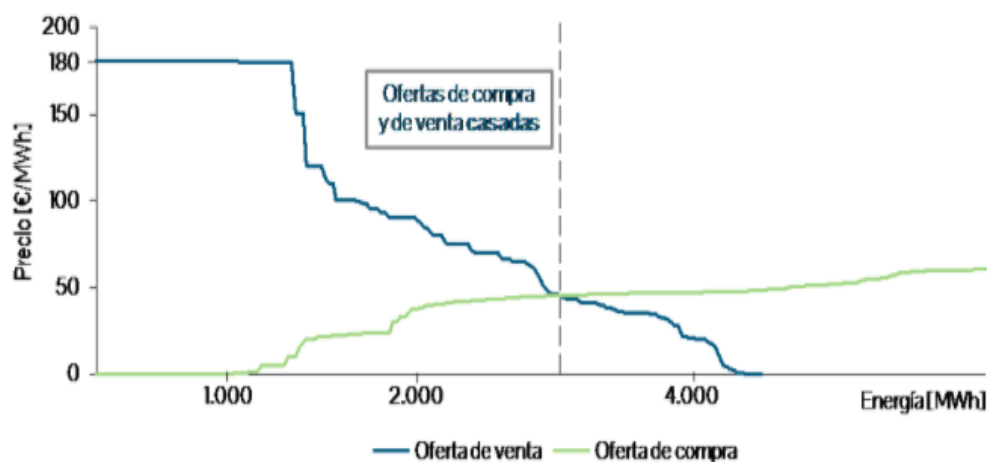


Figura 2 Cruce curva de generación y demanda



De esta manera, resumiéndolo desde el punto de vista de un generador eólico, [6] el cual es el caso analizado en el presente trabajo, las dos variables fundamentales que tiene que gestionar, es decir, la generación eléctrica que va a conseguir, y al precio al cual va a vender dicha energía, son variables estocásticas. Es evidente que se va a necesitar gestionar estas variables de la manera más eficiente posible para ser competitivo. En el presente documento, se va a desarrollar un modelo en el cual poder simular y poder mejorar la toma de decisiones en este entorno de incertidumbre.

### **1.2. Retos afrontados en la resolución del problema**

Para abordar la problemática de las variables futuras estocásticas, se va a usar un enfoque basado en análisis de escenarios. Si bien esta estrategia va a ser explicada en siguientes puntos en más profundidad, es interesante, desde el punto de vista del lector de este trabajo, saber de antemano cuáles son las partes que tienen una resolución más compleja.

En primer lugar, tal y como está desarrollado en el complemento del presente trabajo [7] y en [8] [9] está el hecho de que lo que se intenta hacer es, en cierta manera, predecir acontecimientos futuros. El hecho de predecir el futuro, por definición, es imposible, ya que es futuro, y se desconoce. Se puede argumentar que basándonos en datos históricos se puede llegar a hacer una predicción suficientemente precisa de eventos futuros. Sin embargo, hay que tener cuidado con esta afirmación, debido a que estamos simplificando la problemática; ya que se debe asumir no solo que el modelo que se use para realizar la predicción es correcto, sino que el escenario futuro va a tener las mismas características que el escenario pasado.

En este sentido, lo que hay que hacer no es intentar confirmar si los datos históricos son aplicables para hacer predicciones a futuro, porque como ya se ha dicho, eso es imposible de saber. Al contrario, lo que se debe hacer es evaluar si los errores que se puedan cometer en ese sentido son asumibles o no. En este sentido, como se verá en los siguientes puntos, la variable de salida es una meramente económica, y una posible consecuencia del error del modelo podría ser pérdida económica, y al ser la venta de electricidad un proceso repetitivo de en unidades horarias durante todo el año, cualquier error podría acotarse a la hora en la que ocurre, y ser subsanado de manera rápida. El ejemplo contrario podría ser aplicaciones críticas, por ejemplo, control de un reactor nuclear. Un error en un modelo de predicción en este ámbito no es asumible.

La segunda dificultad destacable en este problema es la complejidad y volumen de procesado desde el punto de vista técnico. Las variables de distribución estocástica se traducen en infinitos

escenarios debido al carácter continuo que tienen. Es por ello que se deberán usar técnicas para poder procesar los escenarios que se vayan a generar reduciendo la cantidad de variables a tratar.

En tercer lugar, está el hecho de que las variables a usar en el problema tienen naturaleza estocástica. Esto hace que el tratamiento matemático que deben tener sea diferente al que, al menos desde la experiencia del autor, se está habituado a usar en las ramas de la ingeniería eléctrica. De esta manera, se deberá recurrir al marco de programación estocástica, el cual muestra la ruta en cómo tratar dichas variables.

Por último, cabe mencionar en cuarto lugar que se debe tener cierto entendimiento del sector del mercado eléctrico para entender y plasmar correctamente la información en la simulación. En este sentido, cada mercado que se intenta evaluar será diferente, sin embargo, como se ha llegado a apuntar anteriormente, para la simplificación de este punto se usará el mercado español para la realización de estas simulaciones.

## 2. ESTADO DEL ARTE

Como bien se ha explicado en la introducción, los generadores eólicos están sujetos a cierta aleatoriedad que afectan a su toma de decisiones [10]. En este sentido, la generación de escenarios funciona como una herramienta para caracterizar dicha aleatoriedad. Cada escenario generado es una posible realización futura, con lo que generando infinitos escenarios, se estaría representado todas las posibilidades futuras.

Para modelizar las diferentes realizaciones de las variables estocásticas se usan los llamados árbol de decisión. Para la construcción de estos árboles de decisiones hay diferentes métodos en la literatura.

Uno los métodos más usados es el de “moment matching” [11] [12], en el cual se definen los escenarios a partir de condiciones estadísticas predefinidas. Otra técnica usada para la generación de escenarios es la llamada “path-based”, en la cual se generan los escenarios usando modelos de series de datos. [13]. También es usado la reducción de escenarios [14, 15] usando el concepto de distancia de probabilidad, el cual sirve para como de parecidos son dos grupos de escenarios representando la misma variable estocástica. De esta manera, permite simplificar un set de escenarios con muchos escenarios a otro set con menos escenarios manteniendo la información. Por último, también está presente de manera habitual la técnica de “internal sampling” [16].

Para realizar este trabajo se va a usar de las técnicas de generación de trayectorias y reducción de escenarios, con lo que se va a realizar el estado del arte sobre estos puntos específicos.

Para la generación de escenarios se va a usar la modelización mediante ARIMA. En la cita [17] recoge toda la información técnica referente al uso matemático de esta herramienta, y en [18] [19] [20] se muestran varias aplicaciones para energéticas, como son energía solar, eólica y petróleo. Sobre la reducción de escenarios, en [21, 22].

Para la programación estocástica, en las citas correspondientes al capítulo 2 del libro citado en la referencia [23] de la bibliografía.

### 2.1. Estructura técnica del trabajo.

Para la resolución de la problemática explicada anteriormente, lo que va a plantear el presente trabajo de investigación es la solución descrita a en el esquema de la Figura 3.

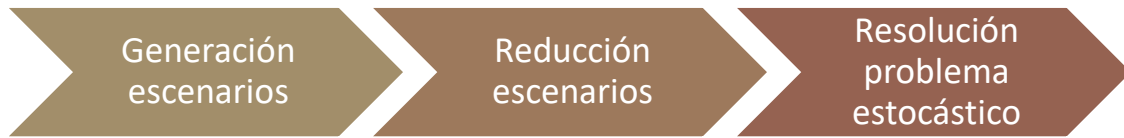


Figura 3 Solución esquemática del problema

En primer lugar, para la realización del árbol de decisiones mediante la generación de trayectorias, utilizando el modelo ARIMA y aplicando el modelo de simulación Montecarlo. La generación de escenarios permitirá caracterizar la aleatoriedad del recurso y de los precios del mercado eléctrico. Cada escenario representa una posible realización de las variables estocásticas. Si se generasen infinitos escenarios, se conseguiría teóricamente definir dicha aleatoriedad [24].

En segundo lugar, se van a reducir los escenarios generados sin comprometer la información contenida en ellos, para esto se utiliza la técnica matemática denominada distancia de probabilidad de Kantorovich [25]. El algoritmo usado se desarrolla y explica en los siguientes puntos del presente documento.

### 3. HERRAMIENTAS MATEMÁTICAS UTILIZADAS

#### 3.1. Generación de escenarios

Para la generación de escenarios se parte realizando la abstracción matemática a partir de series de datos temporales de los modelos ARIMA (“autoregressive integrated moving average”) de las variables estocásticas, para a continuación realizar simulaciones mediante el método Montecarlo partiendo de los modelos generados, donde se crearán las diferentes realizaciones de escenarios posibles [26].

##### 3.1.1. Modelo ARIMA.

El modelo de series temporales ARIMA se puede interpretar con un modelo ARMA con componente integral. El modelo ARMA (Ecuación 1) cuenta con una componente autorregresiva y de media móvil, el cual tiene la siguiente expresión.

$$y_t = \sum_{j=1}^p \phi_j y_{t-j} + \varepsilon_t - \sum_{j=1}^q \theta_j \varepsilon_{t-j}$$

Ecuación 1 Modelo ARMA

El sumando relativo al término  $p$ , hace referencia al componente autorregresivo y  $q$  hace referencia al componente de la media móvil. Para la generación del modelo, se comienza asignando valores de  $p$  y  $q$  [17] (En el presente trabajo no se realiza este paso con motivo de acotar el alcance y sencillez del mismo, cogiendo como referencia valores típicos encontrados en [23]). A continuación, teniendo  $p$  y  $q$  determinadas, se pasa a estimar los valores de  $\theta$  y  $\phi$  siendo estos parámetros los necesarios para ajustar el modelo a la serie de datos presentada [23] [17]. En este sentido, de manera general los modelos ARMA se representan con la siguiente notación:  $ARMA(p, q)$ . Para finalizar los  $y_t$  son los valores de la serie temporal, y  $\varepsilon$  representa el término de innovación, el cual es el término que durante la simulación Montecarlo va a coger valores aleatorios para generar distintos escenarios.

Como evolución de ARMA, el modelo ARIMA incorpora el componente “integral”, el cual, en la práctica permite incorporar a la función información sobre la “estacionalidad” de una serie de datos. Esto, como se explica en [23] en el punto 3.2.2.2. se realiza como manera de solventar la asunción de estacionalidad que se hace para generar el modelo. Dicho de otra manera, el modelo ARMA se asume que es estacionario, sin embargo, los fenómenos que generan la serie de

datos no son estacionarios. Añadiendo el término integral se permite modelizar series de datos no estacionarias. Por ejemplo, para una serie de datos que contempla valores que se repiten en cadencia diaria, como puede ser el precio de la electricidad diario, el cual sigue un patrón similar en bloques de 24 horas, esa información se introduce al modelo mediante el término integral. Para una explicación más detallada del tema se recomienda [17].

La generación del modelo ARIMA es una de las secciones más críticas para el problema planteado. De manera general, para la generación del modelo hay que determinar los siguientes puntos

1. Orden de los términos autorregresivo y de media móvil ( $p$  y  $q$ ).
2. Valor de los parámetros de los términos autorregresivo y de media móvil.

La combinación de los valores correctos hace que los fenómenos que suceden en la realidad, puedan ser replicados por las series de datos que se consigan. En este sentido, siempre se puede generar series de datos, aun teniendo valores de  $p$  y  $q$  o parámetros mal ajustados, pero las series de datos teóricamente no replicarán los valores que se conseguirían en la realidad.

En los puntos anteriormente descritos, como se ha explicado, se ajustan en base a valores pasados de series de datos temporales, y de ahí viene precisamente una de las problemáticas principales de las predicciones basados en modelos ARMA, ya que se tiene que presuponer que en el futuro el modelo ARMA se sigue manteniendo, lo que es una afirmación que no es posible confirmar, como se explica en las citas [8] [9] y se trata más en profundidad en el trabajo asociado [7]. Sin embargo, para el desarrollo de este trabajo, se obvia esta problemática, y se asumirá que el modelo ARIMA que se calcule es válido en el futuro. La generación del ARIMA se determina mediante el algoritmo desarrollado.

### **3.1.2. Simulación Montecarlo.**

La generación de los escenarios, propiamente dicha, se hace aplicando el método de simulación Montecarlo sobre los modelos ARIMA, los cuales están previamente definidos. La aplicación de Montecarlo es simplemente asignar al valor llamado "término de innovación"  $\varepsilon$  en el modelo ARIMA un valor aleatorio siguiendo una distribución normal estándar. Este término de innovación es el que hace que cada escenario difiera entre sí.

Cada serie de datos temporal, que comprende de  $y_0$  hasta  $y_t$  se le denomina trayectoria. Se realizan la cantidad de trayectorias necesarias en función de los requerimientos que imponga

el gestor de riesgos. (cuantas más trayectorias se generen, mejor definida estará la aleatoriedad de la variable) En la Figura 4 y Figura 5 se presentan dos simulaciones ejemplo del método Montecarlo [23] [27]. Nótese que, para las series de datos, no se requiere conocer la naturaleza de las variables. En las figuras siguientes los ejes no son más que valores que se dan a las series de datos, y lo único que tratan de ilustrar las figuras son como se puede realizar simulaciones Montecarlo con diferente cantidad de trayectorias.

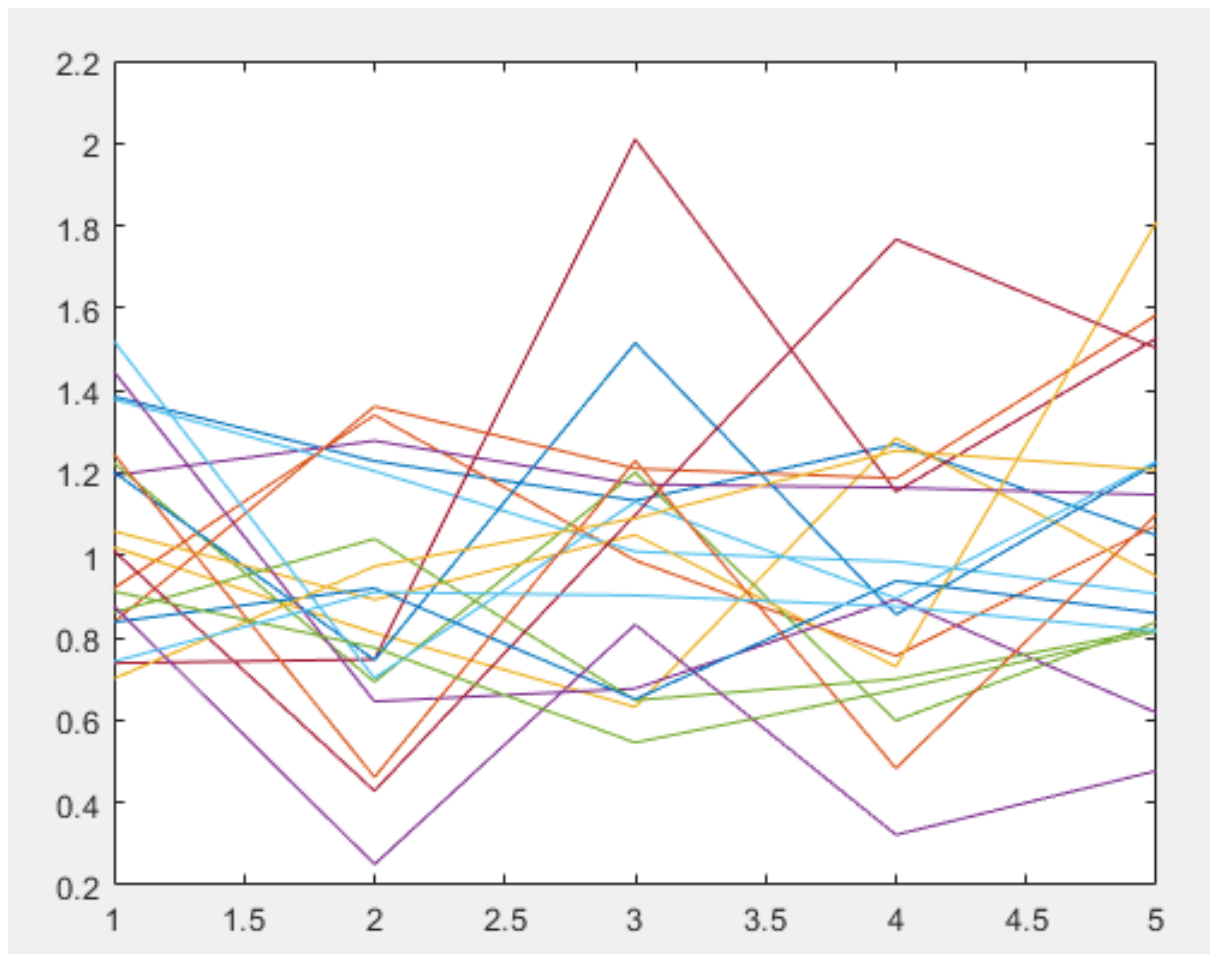


Figura 4 Aplicación método Montecarlo  $t = 5$  y 20 trayectorias

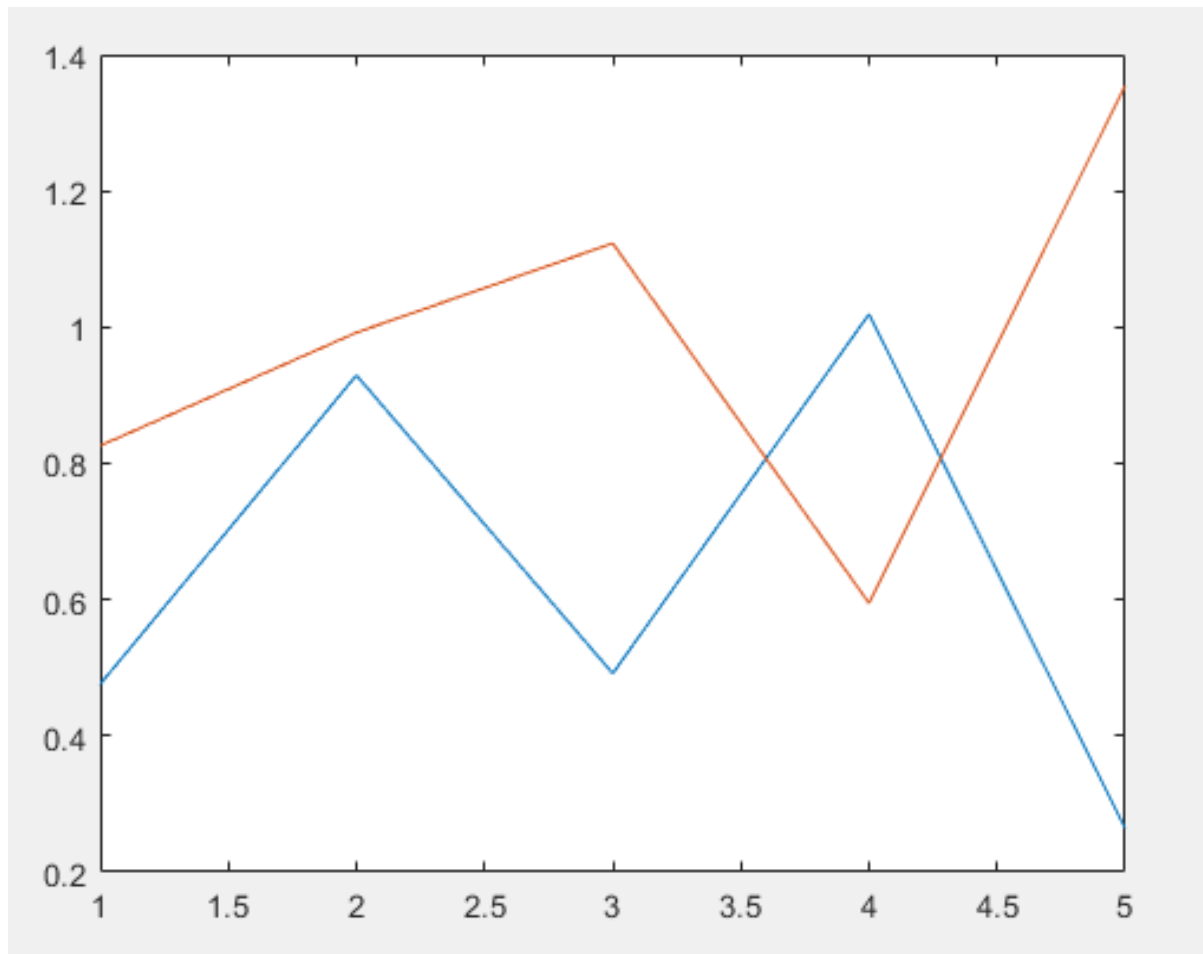


Figura 5 Aplicación método Montecarlo t = 5 y 2 trayectorias

### 3.2.Reducción de escenarios

Como se ha explicado en el apartado de Generación de escenarios, la técnica que se usa es identificar un modelo ARIMA de series temporales que represente las variables estocásticas que se requieran. Con estos ARIMA se realizan los escenarios, lo cual sirve como una aproximación a la variable a estudiar. Para que esta aproximación resulte correcta, se requiere de un alto número de escenarios con el fin de representar la mayor cantidad de posibilidades [28].

Desde un punto de vista computacional, el número de escenarios se puede llegar a disparar por la cantidad de variables involucradas, lo cual imposibilita su procesamiento. Como ejemplo, en un caso que se quiera analizarán escenarios con 5 variables (potencia, precio diario, precio de ajuste, precio de balance, disponibilidad de unidades generadores), con 100 escenarios posibles cada uno, el número total de escenarios a generar sería de:

$$n_{scen}^o = 100 \times 100 \times 100 \times 100 \times 100 = 10^{10}$$



En este sentido, se requiere reducir el número de escenarios sin comprometer la información embebida en ellos.

La técnica matemática en la que se basa el código desarrollado en el presente estudio es la distancia de probabilidad de Kantorovich. [29] El algoritmo usado para la aplicación está basado [30] En las Figura 6 y Figura 7 se presentan ejemplos de los algoritmos.

En el caso de las Figuras, se puede visualizar el efecto de reducir un árbol de escenarios. La información se mantiene, pero las trayectorias se reducen de manera significativa.

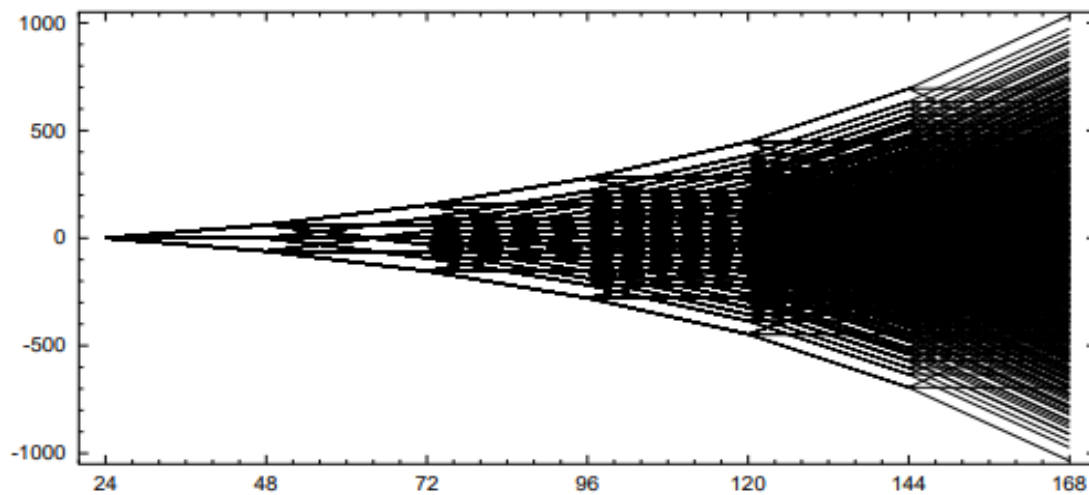


Figura 6 Árbol de escenarios original

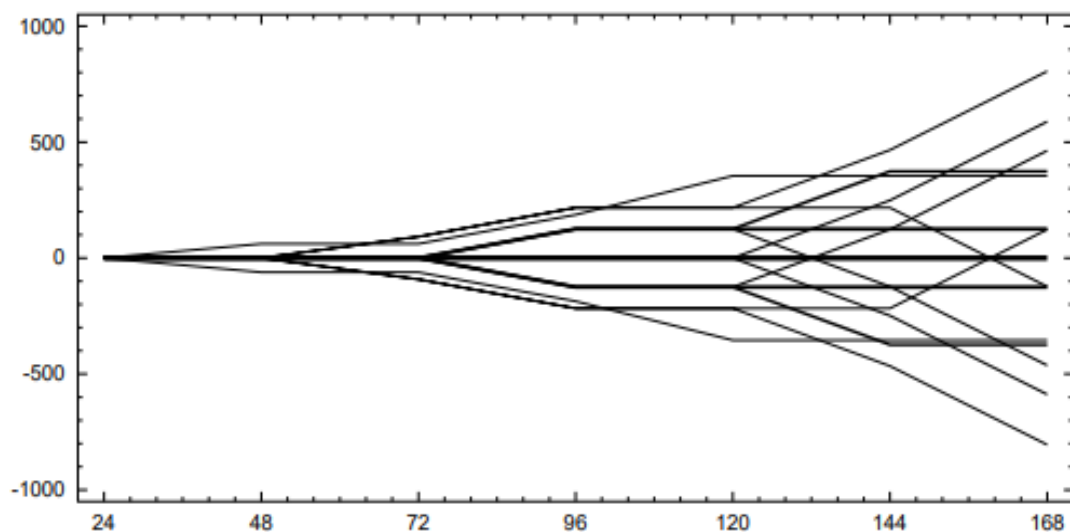


Figura 7 Árbol de escenarios de después de la reducción de escenarios

### **3.3. Resolución problema estocástico**

Respecto a la resolución del problema estocástico no se usa ninguna técnica matemática especial. La particularidad de esta sección es que las variables que se usan son desconocidas pero descriptibles, es decir, son variables estocásticas [31].

La resolución del problema estocástico, se va a realizar mediante la optimización de una de las características de las variables estocásticas, en este caso, al querer conseguir la maximización del beneficio conseguido mediante el mercado eléctrico, la medida a optimizar va a ser la esperanza del beneficio.

Para ello, se va a partir del árbol de escenarios, y se va a asignar a una variable controlable por el agente del mercado, en este caso la potencia que se vende en el mercado diario  $P_d$ , todos los valores posibles. Se evaluará para cada valor la esperanza de beneficio, y se escoge finalmente aquel valor más alto. Para más información respecto al marco de la programación estocástica para variables renovables acudir a [23].

## 4. DATOS DE PARTIDA

---

Para el problema estocástico se requieren los siguientes datos como input. Estos inputs no son las propias variables estocásticas, sino una realización real de las variables estocástica.

- Serie temporal de viento de la unidad generadora
- Número y tecnología de los aerogeneradores de la unidad generadora
- Serie temporal de precio en el mercado diario
- Serie temporal de ratio de precio en el mercado de balance respecto a precio diario

### 4.1. Serie temporal de viento

Para el input de viento, en un problema en el que se quiera evaluar escenarios de una unidad generadora concreta, evidentemente lo que se necesitaría son las mediciones de viento durante un periodo establecido. En este trabajo, como no se busca la evaluación de una instalación en concreto, se ha buscado la manera de hacerlo de la manera más flexible posible para poder evaluar el modelo de resolución generado para diferentes velocidades de viento. Por lo tanto, se ha optado por generar las series de viento en la programación mediante un algoritmo.

La manera de conseguir la serie de tiempo en el algoritmo se describe en el punto 5.2 del presente documento. Lo que se va a conseguir es una distribución Weibull partiendo de unos parámetros introducidos por el usuario de "shape" y "scale" como se presenta en la Figura 9, y a partir de esta Weibull se van a generar series de datos aleatorias mediante funciones de matlab(Figura 10).

En la siguiente Figura 8 se presenta todo el procesamiento de los diferentes inputs, descritos a lo largo del documento.

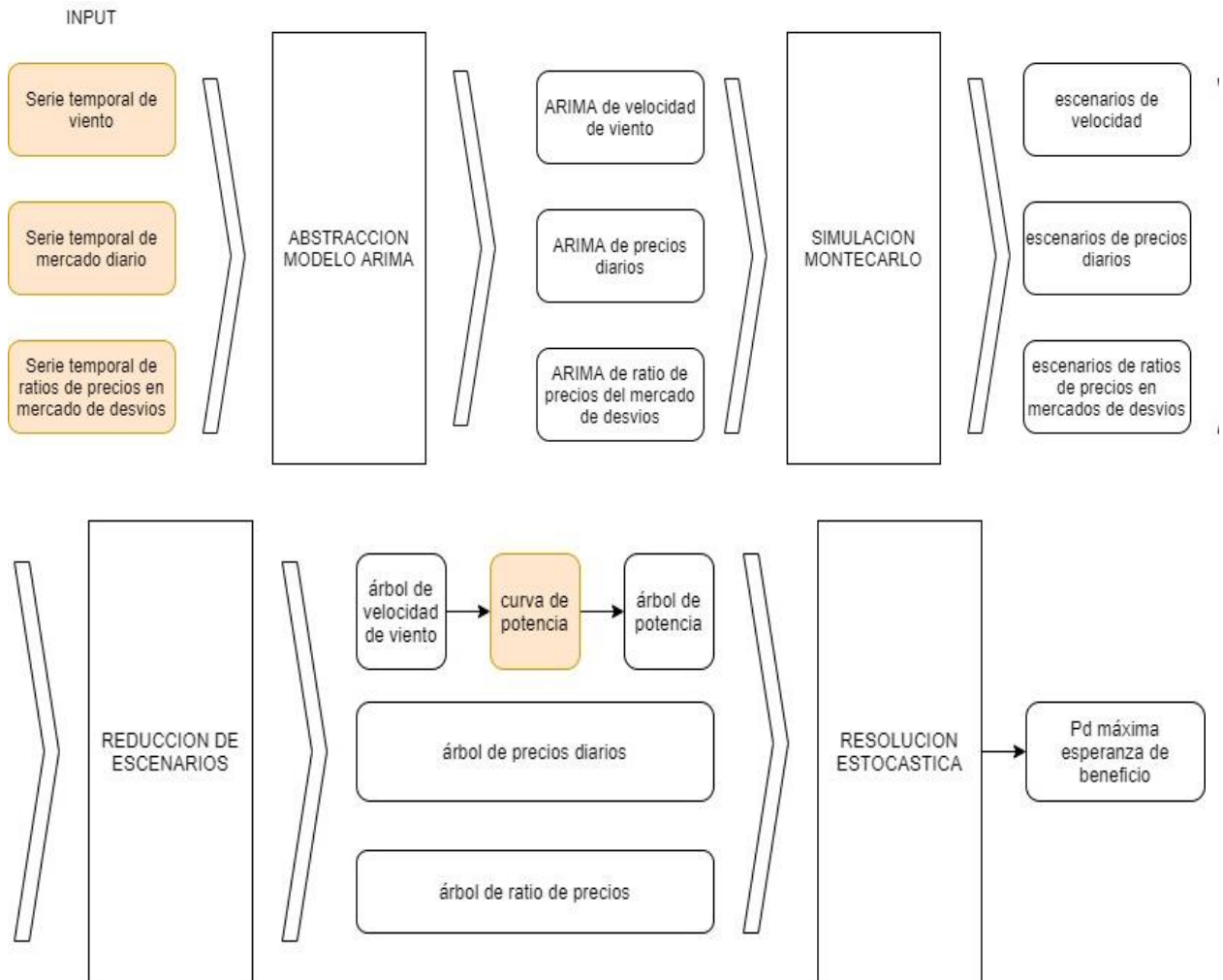


Figura 8 Procesamiento Inputs

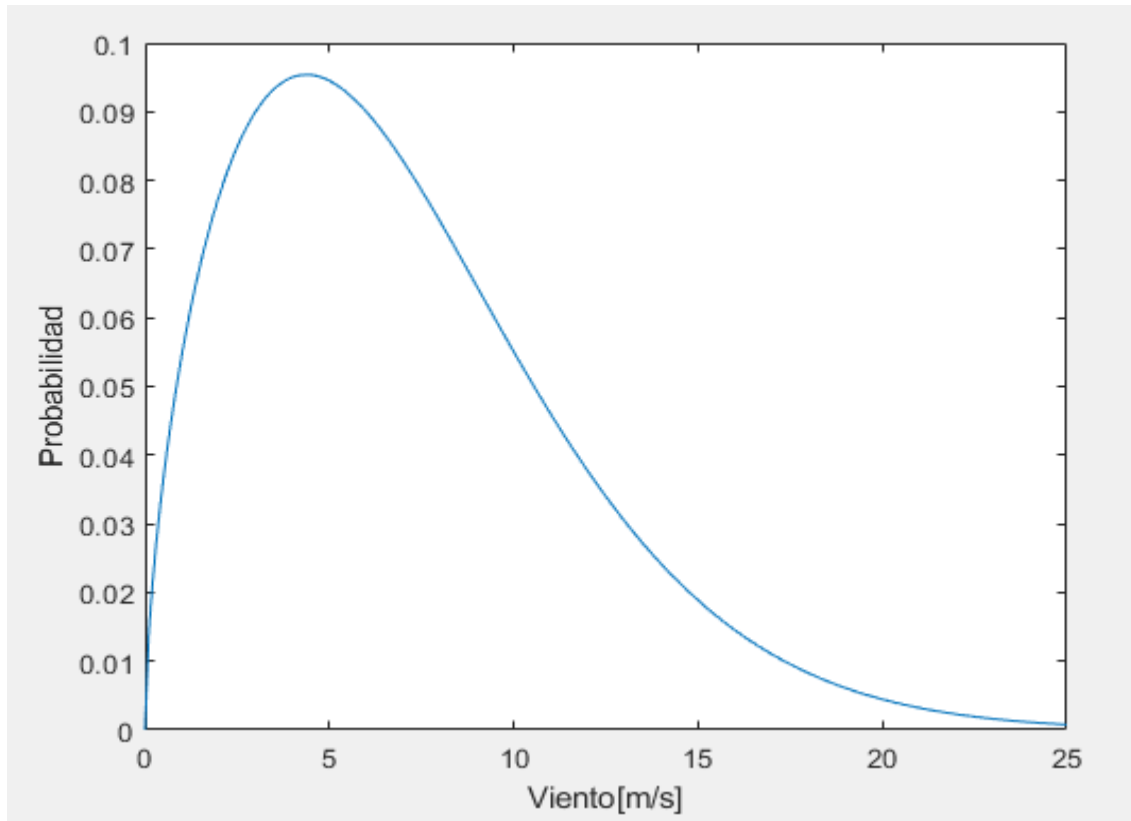


Figura 9 Distribución de Weibull

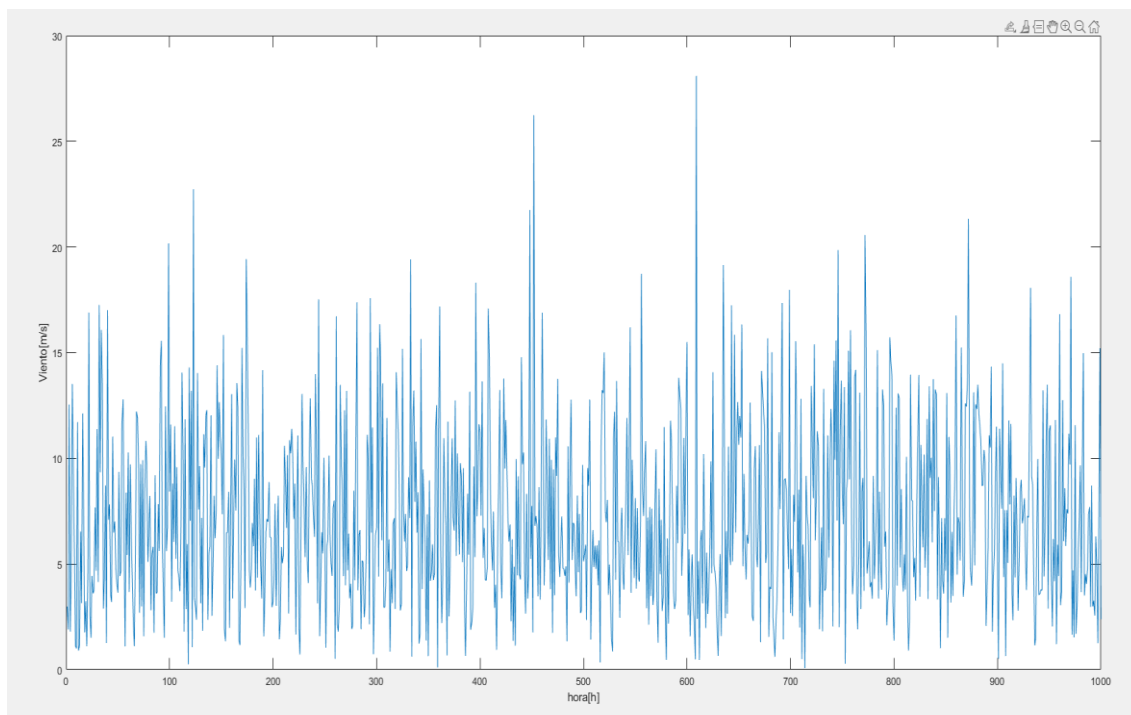


Figura 10 Serie de datos de viento

#### 4.2. Número y tecnología de los aerogeneradores

Para las simulaciones, el input necesario es el número de aerogeneradores que va a tener la unidad generadora, y la tecnología que se va a usar. En la práctica, lo que se quiere cuando se refiere a la tecnología es la curva de potencia o modelo que permita transformar la variable viento a la variable potencia [32].

Para este caso, se ha supuesto una unidad con 10 aerogeneradores con la curva de potencia mostrada en la Figura 11

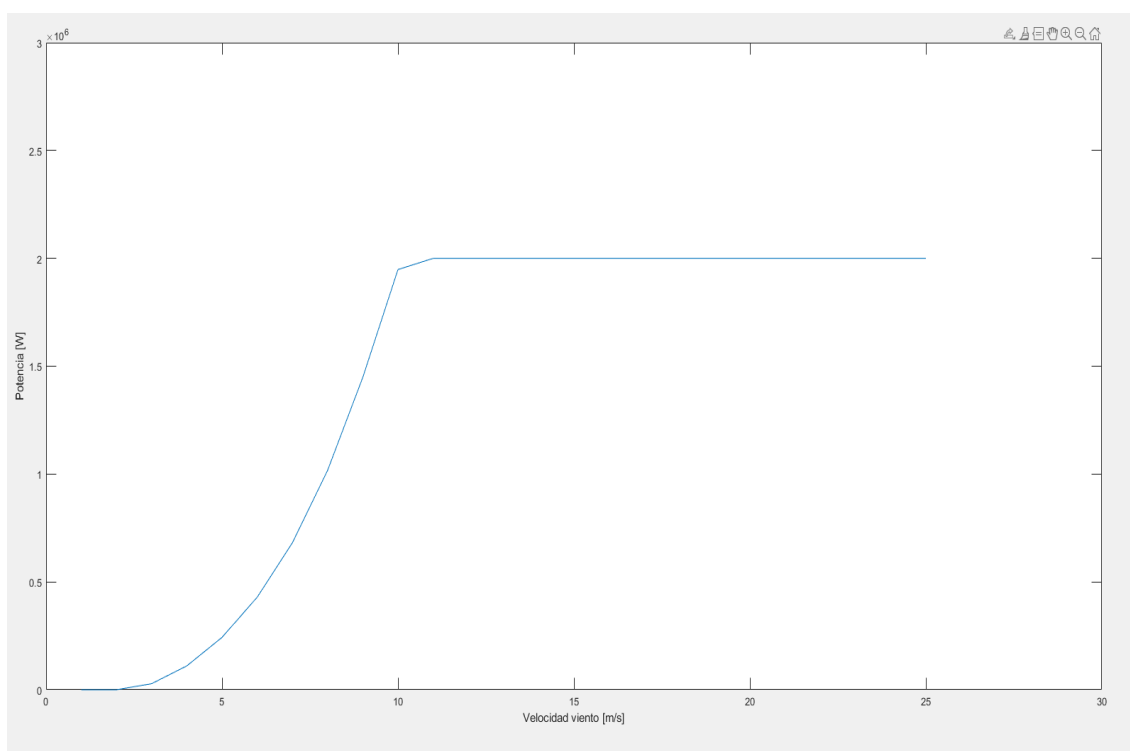


Figura 11 Curva de potencia aerogenerador

#### 4.3. Serie temporal de precio en el mercado diario.

Para la serie temporal del precio diario, al haber elegido el mercado eléctrico español se acude a la página web de Red Eléctrica España (REE), el cual actúa como operador del sistema. La web en cuestión es [33].

Para conseguir los precios diarios se busca la variable “precio mercado spot diario España” (Figura 12). Como se está usando el método de generación por trayectorias, eso implica que no se deben introducir ningún dato estadístico (media, máximo, mínimo). Sin embargo, si es importante mencionar que el periodo temporal que se recoja debe respetar lo marcada en la parte de

estacionalidad del ARIMA generado. En este sentido, como se va realizar la simulación para que cumpla con cualquier día del año, la serie de datos debería ser anual.

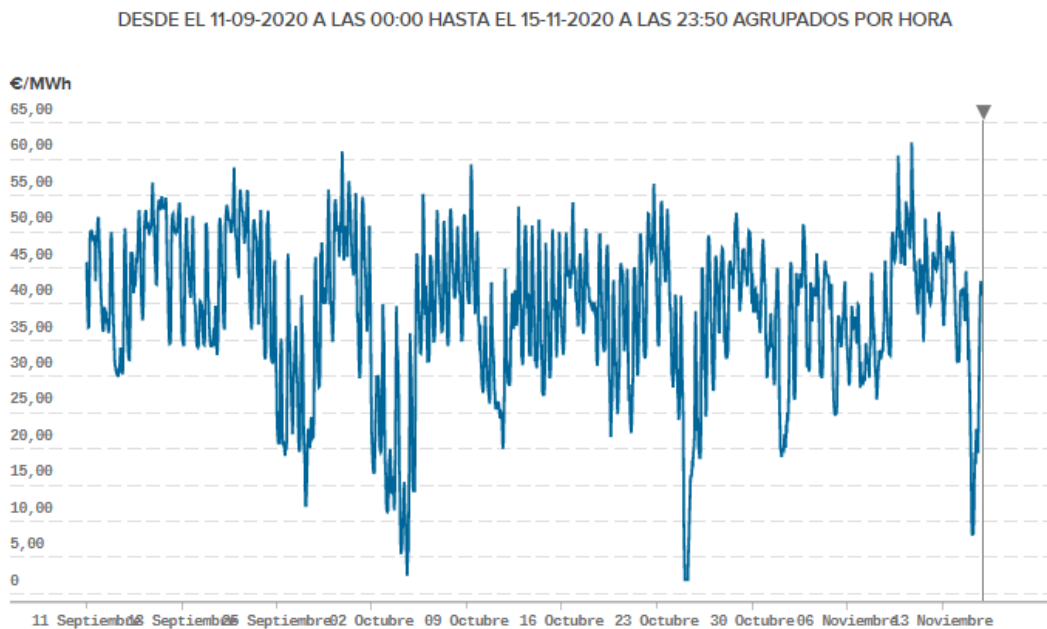


Figura 12 Precio mercado spot diario España

#### 4.4. Serie temporal de ratio de precio entre precio de balance y precio diario.

Esta variable representa el precio que se ha de pagar cuando se vende o compra en el mercado de desvíos frente al precio del mercado diario. Se usa la variable referenciada frente al mercado diario para reducir efectos que puedan tener la propia correlación entre el precio de mercado diario frente al mercado diario. Por ejemplo, que un precio del mercado de desvíos a subir sea relativamente alto, por ejemplo 200€, no significa necesariamente que el efecto del desvío ha sido grande, ya que puede ser que el precio en el mercado diario fuese alto también, por ejemplo 190€.

En este caso, tal y como para el precio diario, se acude a la página de REE, [4] pero en este caso se deben buscar dos variables, la ratio de precio cuando el mercado es a bajar y de cuando es a subir.

Para ello, se buscan las variables de “precio del desvío medido entre precio marginal diario a subir” y “precio del desvío medido entre precio marginal diario a bajar” como se presenta en la Figura 13.

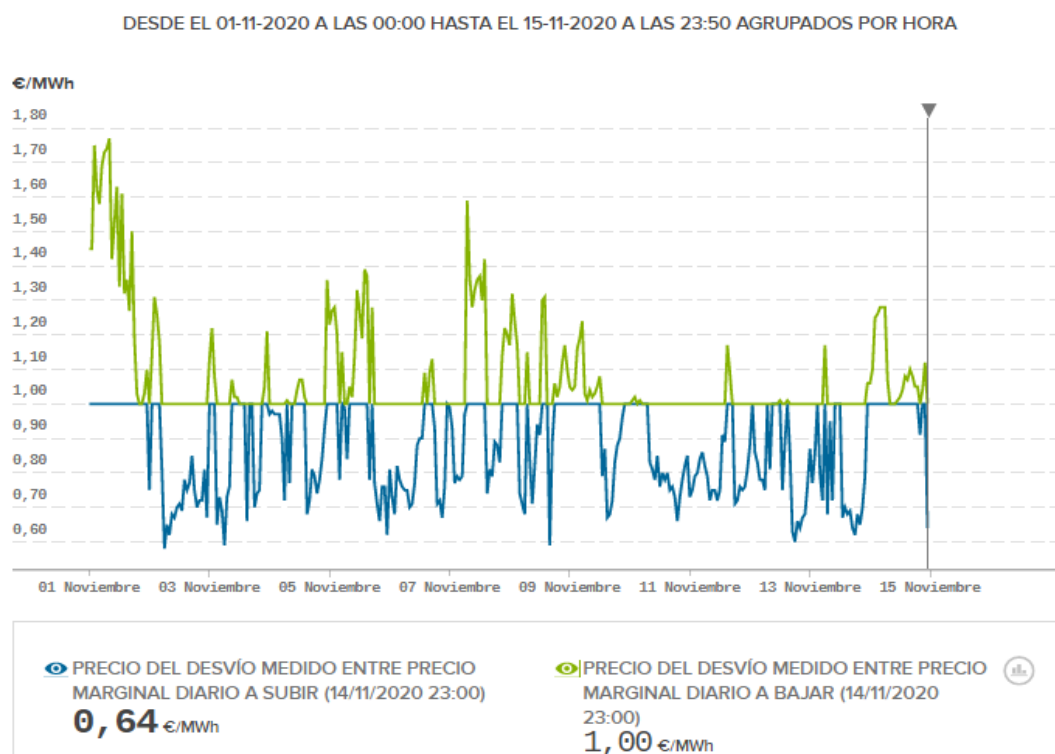


Figura 13 Precio del desvío medido entre precio marginal diario a subir y precio del desvío medido entre precio marginal diario a bajar

Como se puede apreciar en la figura, es evidente que las 2 variables son complementarias. Como se explica en el punto relativo al algoritmo en la generación de escenarios, para modelizar la variable se requiere de generar una variable auxiliar que represente las 2 variables a la vez.

#### 4.5. Formato de mercado asumido.

Para modelar el funcionamiento del problema a plantear se debe saber cómo es el mercado en el cual se subasta la electricidad [34]. Para este caso, se hacen ciertas simplificaciones de cara a acotar el problema a resolver, entendiendo en que si se requiere de cara a mejorar el modelo se deberán incluir los diferentes mecanismos de mercados simplificados en las líneas futuras. La simplificación es considerar solo los mercados de mercado diario y mercado de balance de energía (desvíos), eliminando el resto de mercados disponibles, entre ellos el de ajuste. De esta manera, después de simplificar el funcionamiento del mercado se hace que la toma de decisiones sea más sencilla al reducir el número de variables estocásticas.

Resumiéndose, plantean dos mercados en el que se puede acudir:

- Mercado diario (D+1)
- Mercado de balance de energía



Una ilustración esquemática extraída de [23] , donde también se incluyen las variables estocásticas que se deben usar es la presentada en la Figura 14. En ella se ve como se consideran 2 fases. La 1 es en el momento donde se debe realizar el envío del mercado diario, en el cual se toma una decisión sin saber los valores que alcanzarán las variables estocásticas consideradas.

En la fase 2, las variables estocásticas han resultado en variables ya determinadas, y se tendrá el output determinado.

Además, también se ilustra el vector de escenario que se debe considerar, con las variables de precio diario, potencia generada, ratio de precio de desvíos a subir, ratio de precio de desvíos a bajar

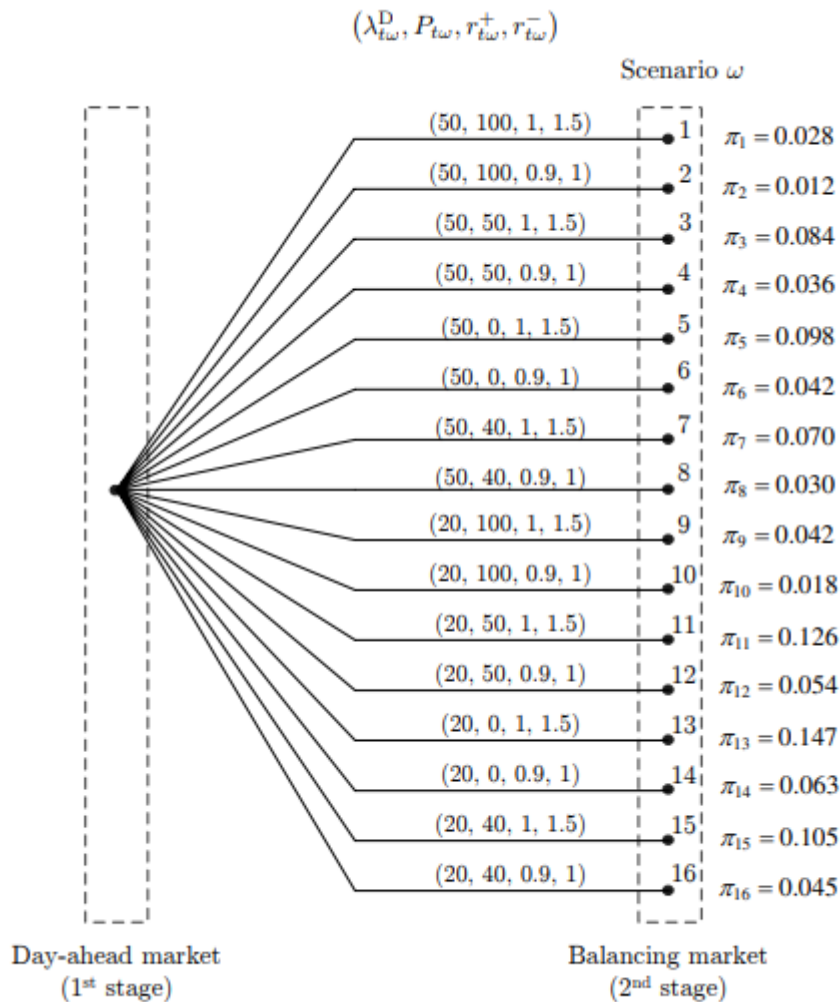


Figura 14 Esquema programación estocástica a realizar

#### **4.6. Asunciones sobre variables a analizar.**

Para plantear el problema, respecto a las variables a usar también se ha realizado una asunción que debe ser mencionadas. Como se repite varias veces durante esta memoria, se debe tratar gran parte de estos puntos como futuras mejoras que ayudan a mejorar la precisión de los modelos.

Se consideran todas las variables estocásticas independientes entre sí. Esto es incorrecto en la realidad, ya que existen relaciones conocidas entre parte de las variables que se usan. Por ejemplo, el viento durante una hora está inversamente correlacionado con el precio de la electricidad para esa hora. Esto es porque generalmente cuando hay mucho viento en una localización concreta, es probable que sea un efecto más zonal, incluso a nivel de toda España, y al aumentar la oferta eléctrica, el cruce que se realiza para las subastas diarias ocasiona que el precio sea menor.

Aunque como se explica en puntos anteriores, no se ha contemplado el mercado de ajuste, también es mencionable que existe correlación entre los precios de mercado diario y el de mercado intradiario.

Estas relaciones, en el caso de considerarlas dependientes entre sí, requiere de un grado de complejidad mayor, que también deberá incluirse en el algoritmo desarrollado.

## 5. IMPLEMENTACIÓN DEL ALGORITMO EN MATLAB

En el presente punto se va a explicar cómo se ha ido desarrollando el algoritmo en el software Matlab [35], para la resolución del problema estocástico planteado, así como explicaciones técnicas de puntos específicos.

### 5.1. Generación y reducción de escenarios de viento.

En la primera sección del código (Figura 15) se introducen los valores manualmente de “shape” y “scale” que definen la distribución de probabilidad Weibull.

Con la función `wblrnd` se genera una serie temporal de valores de viento en función de la distribución Weibull especificada anteriormente. En este caso se genera una serie de datos, con mil valores de vientos horarios. En la variable `wind_h_data`, si se quisiera usar el modelo generado en este trabajo para un emplazamiento concreto, se debería introducir los valores conseguidos en mediciones.

```
shape = 8;  
scale =1.61;  
  
%wind_h_data corresponde a datos de entrada horarios de viento  
wind_h_data = wblrnd(shape, scale,1000,1);
```

Figura 15 Código para la generación del perfil de viento

Un posible valor de `wind_h_data` puede ser el presentado en la Figura 16 (hay que tener en cuenta que, tal y como está escrito, el código genera una serie de datos distinta cada vez que se ejecuta, ya que se basa en una generación de datos aleatoria).

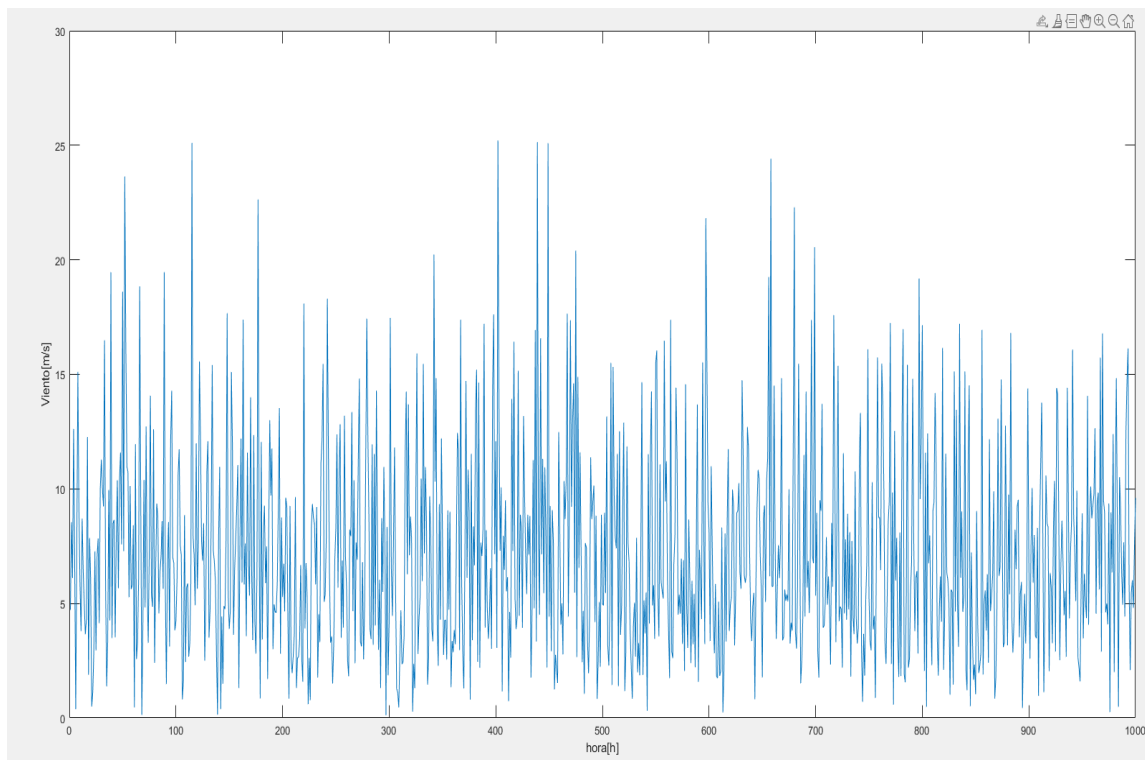


Figura 16 Serie de datos generada

Los valores conseguidos por la distribución Weibull no llevan una distribución normal. Sin embargo, los valores que se deben usar para la determinación de los modelos ARIMA deben ser normales por definición. Es por ello que en la siguiente parte del código se va a realizar la normalización de los valores conseguidos en la Weibull siguiendo el procedimiento recogido en la sección 3.2.2.3 en [23].

Para ello, lo que se hace es el siguiente procedimiento. Primero se calcula partiendo de la función de probabilidad de Weibull la función de acumulación. Después, para cada valor conseguido de viento, se le asigna su probabilidad en la función de acumulación. Por último, se calcula con esa probabilidad el valor correspondiente en una función estándar normal. De esta manera, para cada valor de viento se ha conseguido su equivalente en una función estándar normal con la que se puede trabajar con los modelos ARIMA. Esta serie de transformaciones se representan gráficamente en la Figura 17.

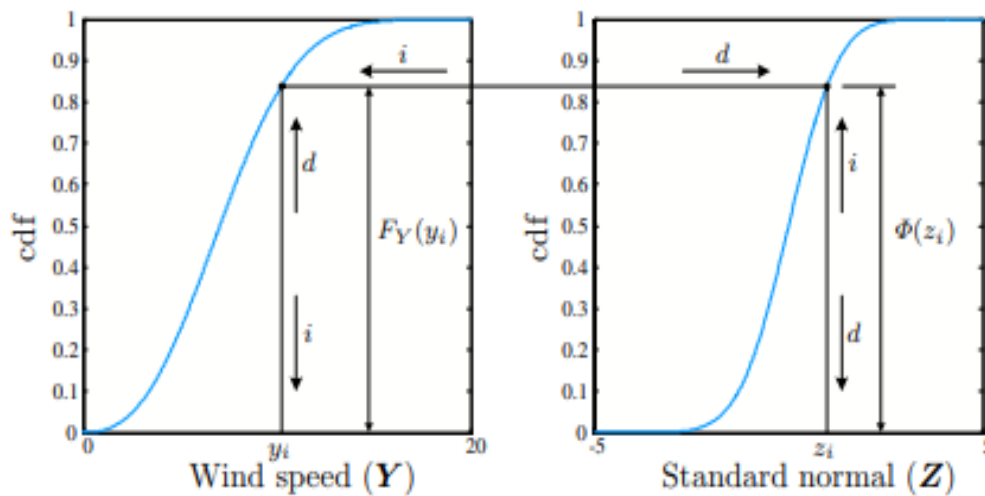


Figura 17 Transformación de la velocidad de viento a una función estándar normal

El código correspondiente a estas transformaciones es el siguiente

```
%normality of weibull
v = wind_h_data;

%sacar datos desde los valores wind_h_data
p = wblcdf (v, shape, scale);
normalized_wind_speed = norminv(p);
```

Figura 18 Código para realizar transformación de Weibull a una función normal

En esta sección, la función wblcdf devuelve el valor de probabilidad que tiene cada valor  $v$  en una función de acumulación Weibull definida por los valores de shape y scale definidos. Nótese en este punto que en el código planteado en este trabajo los parámetros de scale y shape son datos de partida. Sin embargo, si lo que se hace es analizar para un caso que el dato de partida es una serie temporal de viento, se debe conseguir mediante funciones correspondientes los valores de scale y shape. Estos valores se podrán estimar para ciertos márgenes de confianza, y por lo tanto es una fuente de error en los cálculos. En la Figura 19 y Figura 20 se puede comparar los valores conseguidos de vientos y su transformación posterior en valores normalizados.

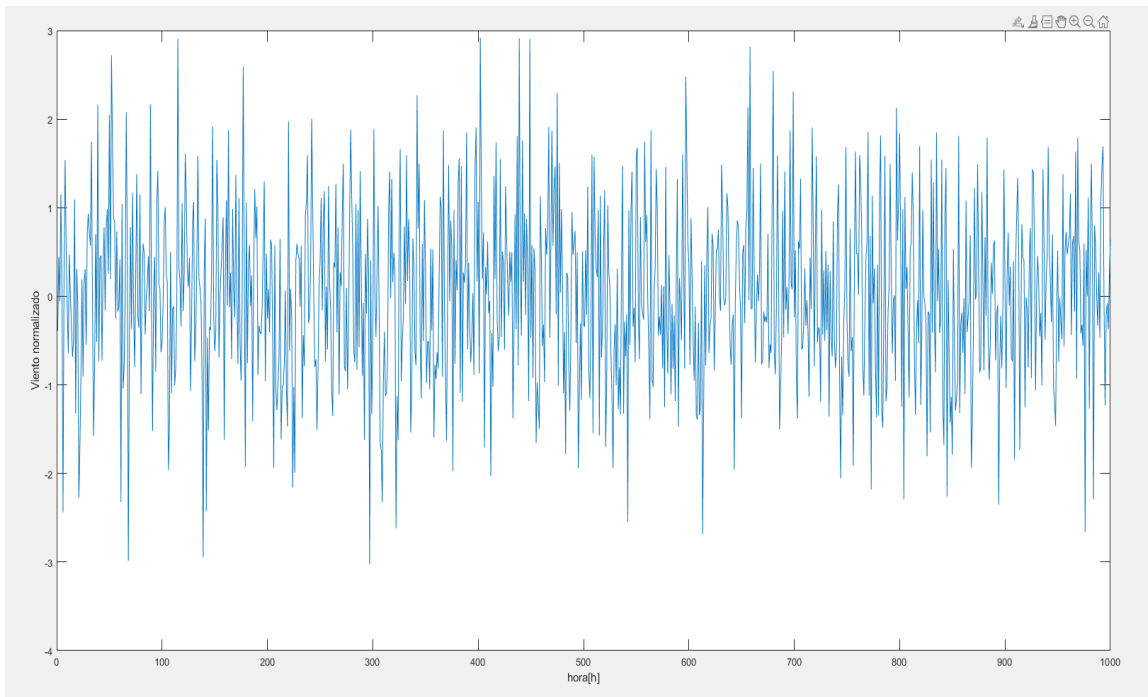


Figura 19 Valores conseguidos de vientos

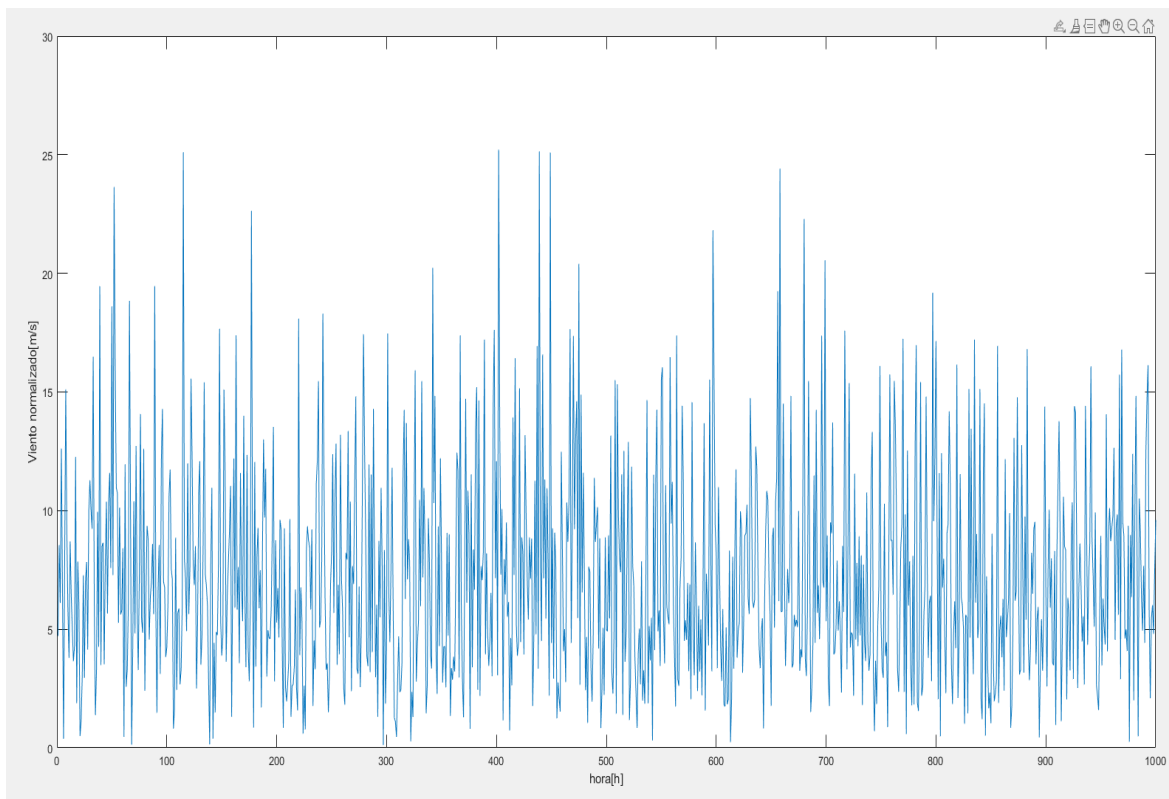


Figura 20 Valores normalizados

En la siguiente parte del código (Figura 21) se pasa a definir y ajustar los valores para el modelo de ARIMA para los valores de viento. Esto se representa en la siguiente sección del código:

```
%arma fit
ARMA mdl = arima('ARLags',1:2,'Constant',0);
ARMA_est = estimate(ARMA mdl, normalized_wind_speed );
```

Figura 21 Código para el modelo ARIMA

En la primera línea de código lo que se están definiendo son los valores de las órdenes en las componentes autorregresiva y de media móvil que se han de usar. En este caso se ha usado un ARMA (2,0,0), es decir, no se contempla media móvil ni parte integral, y se considera que el componente AR tiene orden 2. La implicación que tiene esto es que la fidelidad que se va a tener con el modelo ARIMA respecto a la serie de datos original va a ser peor, y de que no se considera estacionalidad para el modelo.

Nótese que en este punto se ha hecho una consideración muy importante, ya que para el cálculo del orden de los términos se requiere de unos cálculos más complejos. Sin embargo, se ha considerado para simplificar el alcance de este trabajo que el orden de los términos es correcto de partida. Este punto se añadirá como una de las líneas futuras a trabajar. Los valores de los que se parte como referencia están presentes en [23]

En la segunda línea de la Figura 22 se define completamente el modelo ARMA\_est. Para ello se ajustan los valores de los parámetros correspondientes de ARMA explicados anteriormente haciendo el ajuste de ARMA mdl para los valores de “normalized\_wind\_speed”

Los valores que devuelve Matlab por la consola para la primera línea de código se presentan en la Figura 22 y para la segunda línea de código se presenta en la Figura 23.

```
>> ARMA_mdl = arima('ARLags',1:2,'Constant',0)
```

```
ARMA_mdl =
```

[arima](#) with properties:

```
Description: "ARIMA(2,0,0) Model (Gaussian Distribution)"
Distribution: Name = "Gaussian"
          P: 2
          D: 0
          Q: 0
Constant: 0
          AR: {NaN NaN} at lags [1 2]
          SAR: {}
          MA: {}
          SMA: {}
Seasonality: 0
          Beta: [1×0]
          Variance: NaN
```

Figura 22 Resultado ajuste ARMA\_mdl

En la figure 21 se puede apreciar como el modelo generado en matlab solo cuenta con los valores de  $p$  y  $q$  definidos, pero al no hacer el ajuste del modelo con la serie de datos todavía no cuenta con valores de  $\theta$  y  $\Phi$  definidos.

En la figura 22, entre otros datos se muestra como después del ajuste los valores para de  $\theta$  y  $\Phi$  están ya definidos.



```
>> ARMA_est = estimate(ARMA mdl, normalized_wind_speed )

ARIMA(2,0,0) Model (Gaussian Distribution):
```

	Value	StandardError	TStatistic	PValue
Constant	0	0	NaN	NaN
AR{1}	0.02775	0.035034	0.7921	0.4283
AR{2}	0.0079808	0.031134	0.25634	0.79769
Variance	0.98069	0.043652	22.466	8.9388e-112

```
ARMA_est =

arima with properties:

    Description: "ARIMA(2,0,0) Model (Gaussian Distribution)"
    Distribution: Name = "Gaussian"
                P: 2
                D: 0
                Q: 0
    Constant: 0
                AR: {0.0277504 0.00798076} at lags [1 2]
                SAR: {}
                MA: {}
                SMA: {}
    Seasonality: 0
                Beta: [1x0]
                Variance: 0.980689
```

Figura 23 Resultado ARMA\_est

En este punto del programa ya está el modelo ARMA para el viento generado, con lo que ya se puede proceder a realizar las simulaciones Montecarlo en base a este modelo para generar los diferentes escenarios de viento. Esta funcionalidad está recogida en la siguiente parte de código (Figura 24).

```
%%DAY AHEAD WIND SCENARIO GENERATION

%montecarlo_simulations

scen_number = 100

[gen_scenario, innovations] = simulate(ARMA_est, 23, 'NumPaths', scen_number)
```

Figura 24 Código modelo Montecarlo

En este caso, se van a generar 100 escenarios diferentes con 23 pasos (en este caso horas) cada uno, teniendo en cuenta que se considera  $y_0=0\text{m/s}$ .

La representación gráfica se presenta en la Figura 25. Nótese que en este momento del programa los valores de viento siguen normalizados.

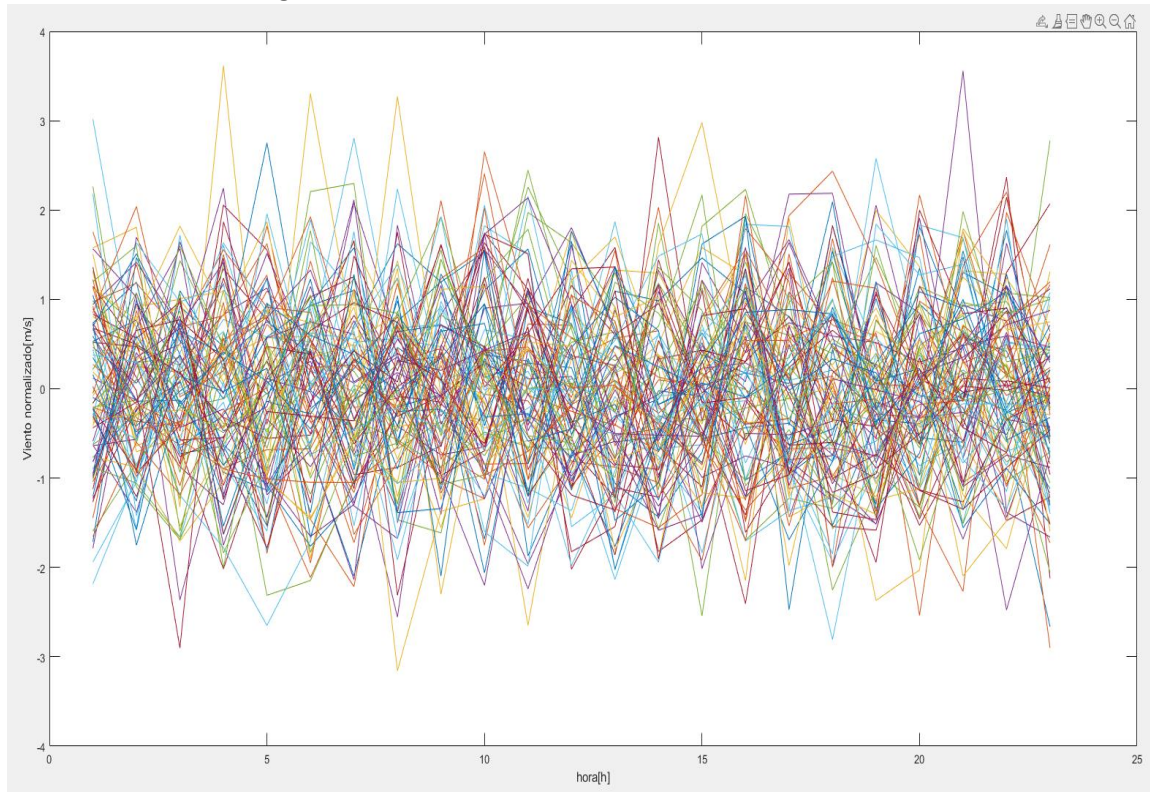


Figura 25 Representación de 100 escenarios de viento diferentes con 23 pasos

Para hacer la transformación inversa, es decir, pasar de la distribución normal a valores de vientos comprendidos en la distribución Weibull, se ejecuta la siguiente porción de código (Figura 26).

```
%normal ->weibull|  
  
p_gen_scenariio = normcdf(gen_scenariio);  
v_gen_scenariio =wblinv(p_gen_scenariio, shape, scale);
```

Figura 26 Código para la transformación de distribución normal a Weibull

La representación correspondiente de  $v\_gen\_scenariio$  se presenta a continuación en la Figura 27.

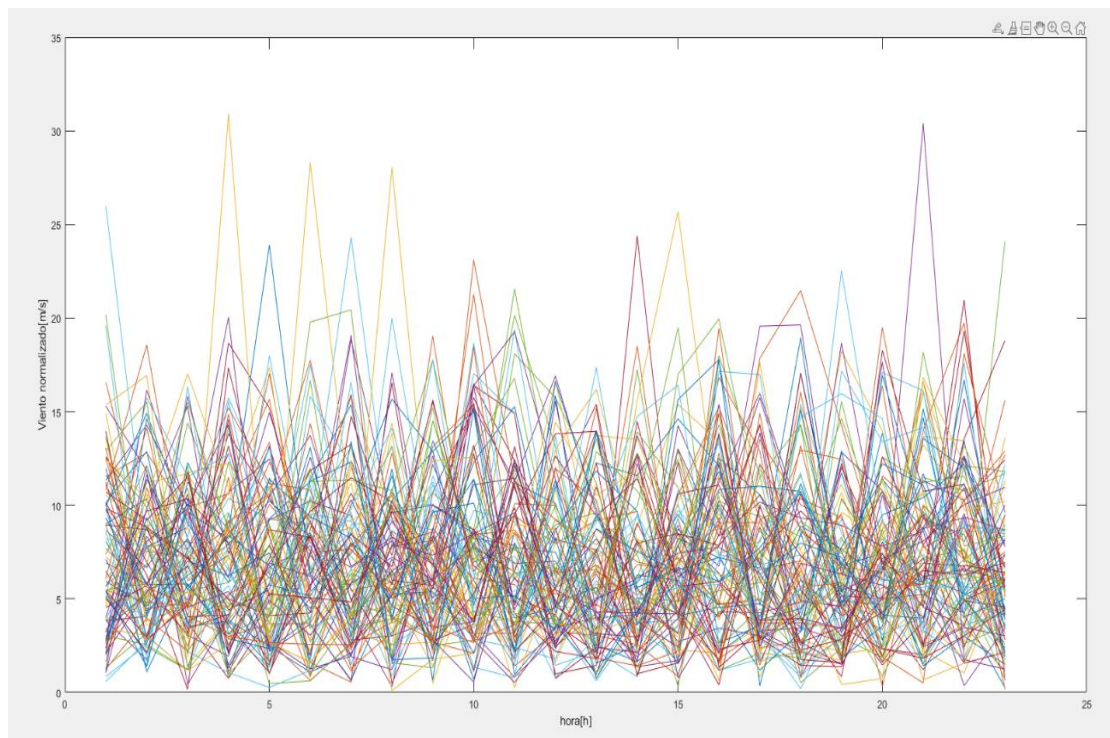


Figura 27 Representación v\_gen\_escenario

En este punto de ejecución del código, se cuentan con 100 escenarios de viento, pero por limitaciones computacionales explicadas en puntos anteriores se debe realizar una reducción de escenarios. Recapitulando, una variable estocástica desconocida está mejor definida mediante el método de generación de escenarios cuantos más escenarios se generen. Sin embargo, la cantidad de escenarios, teniendo en cuenta que se tienen varias variables, puede ser una cantidad que sature la capacidad de procesamiento de los ordenadores. Para ello, se reduce la cantidad de escenarios, sin embargo, no se quiere perder la información sobre la aleatoriedad de la variable que está embebida los escenarios, con lo que se aplica métodos de reducción de escenarios basado en distancia de probabilidad específicos.

La siguiente sección del código realiza esta reducción de escenarios (Figura 28)

```

data = {v_gen_scenario}

%[S_tol,P_tol,J_tol] = scenred(data, 'cityblock','tol',0.2);

%scenario reduction

T = size(data{1},1); % timesteps
N = 5; % number of scenarios at t==T
[S,P,J,L] = scenred(data, 'cityblock','nodes',round(linspace(1,N,T)));
    
```

Figura 28 Código para la reducción de escenarios

En el código mostrado, es un valor a introducir por el usuario del código la cantidad de escenarios que desea mantener. El número escogido será un compromiso entre el tiempo que se dispone para realizar el procesamiento de los datos y de la calidad de los datos que se quiera mantener. En este caso se ha escogido el valor de 5.

La función de scenred devuelve mediante S el árbol de escenarios resultante, mientras que P devuelve la probabilidad de cada nodo del árbol de escenarios.

Para el valor de S y P se consigue la siguiente matriz (Figura 29) y representación gráfica (Figura 30).

	1	2	3	4	5	t
1	13.5643	13.5643	13.5643	13.5643	13.5643	1
2	3.8067	3.8067	3.8067	3.8067	3.8067	2
3	4.6099	4.6099	4.6099	4.6099	4.6099	3
4	8.0370	8.3728	8.0370	8.3728	8.0370	4
5	4.3615	6.7613	4.3615	6.7613	4.3615	5
6	10.2915	4.4924	10.2915	4.4924	10.2915	6
7	8.7126	7.9932	8.7126	7.9932	8.7126	7
8	1.9766	6.2525	1.9766	6.2525	1.9766	8
9	7.8662	8.0152	7.8662	8.0152	7.8662	9
10	4.0334	11.3890	8.6083	11.3890	4.0334	10
11	7.8777	2.1586	7.6157	2.1586	7.8777	11
12	2.3323	16.6065	6.4419	16.6065	2.3323	12
13	5.7935	9.8118	4.6775	9.8118	5.7935	13
14	4.8253	3.6280	3.5783	3.6280	4.8253	14
15	2.9430	5.9527	10.5368	5.9527	5.4620	15
16	8.8263	10.8331	15.2544	10.8331	9.8013	16
17	6.7936	11.0154	9.2997	11.0154	3.1364	17
18	2.8006	10.7281	6.6302	10.7281	5.5986	18
19	2.1563	5.3482	2.0252	5.3482	9.2397	19
20	4.8433	9.4800	9.2254	9.4800	6.4860	20
21	5.0934	4.9393	5.0528	10.4060	3.1818	21
22	7.7011	1.5661	8.8559	12.1887	5.3136	22
23	3.5263	2.1435	4.0555	7.9183	3.7058	23
24						24

Figura 29 Matriz valores S y P

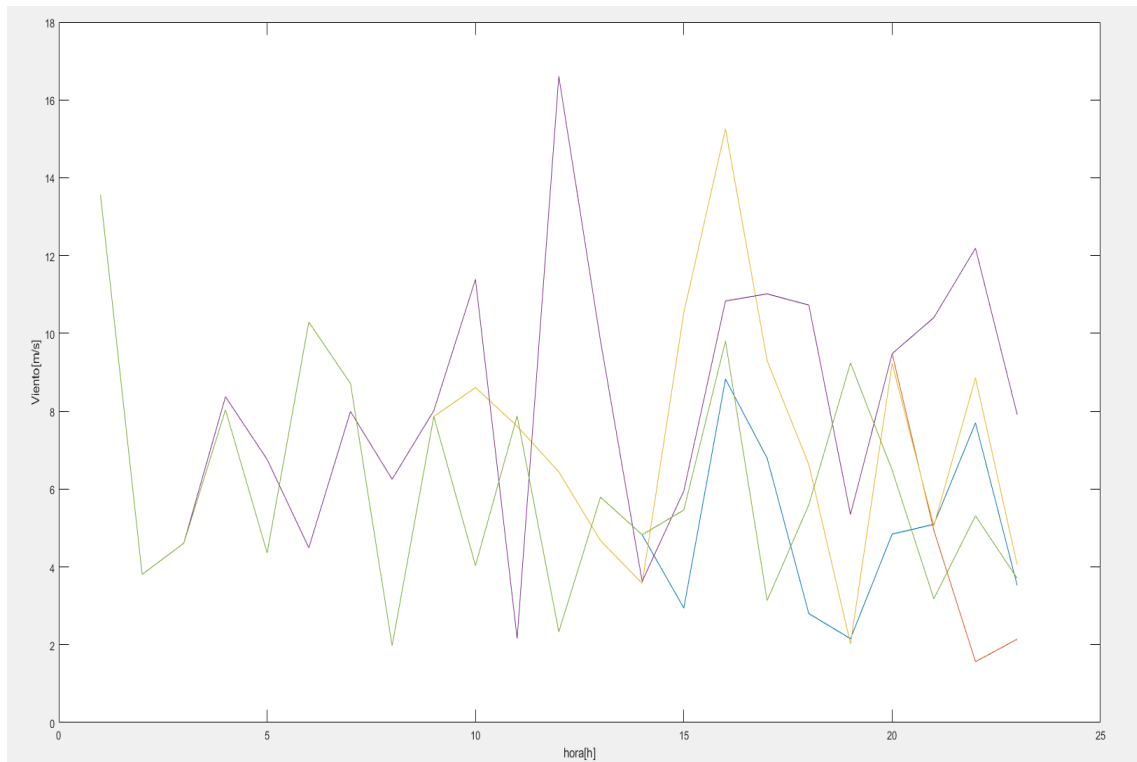


Figura 30 Representación gráfica valores S y P

En la siguiente sección de código lo que se hace es partir de unos valores de viento iniciales, los cuales son los últimos dos valores de cada escenario de viento, volver a generar y reducir escenarios de viento. Esto se hace para modelar el funcionamiento del mercado de ajuste, que, aunque para simplificar y limitar el alcance del problema no se ha incluido en las simulaciones de este trabajo, si se incluye la generación de escenarios de viento partiendo de otros escenarios de viento.

Lo que se consigue haciendo estos dos escenarios es simular como se aumenta la certeza a medida que se acerca la hora de ejecución del mercado, es decir, se tiene más certeza sobre la velocidad de viento que se va a tener cuanto más cerca temporalmente en el tiempo del momento que se quiere predecir.

Gráficamente, tal y como está en [23] se representa de la Figura 31.

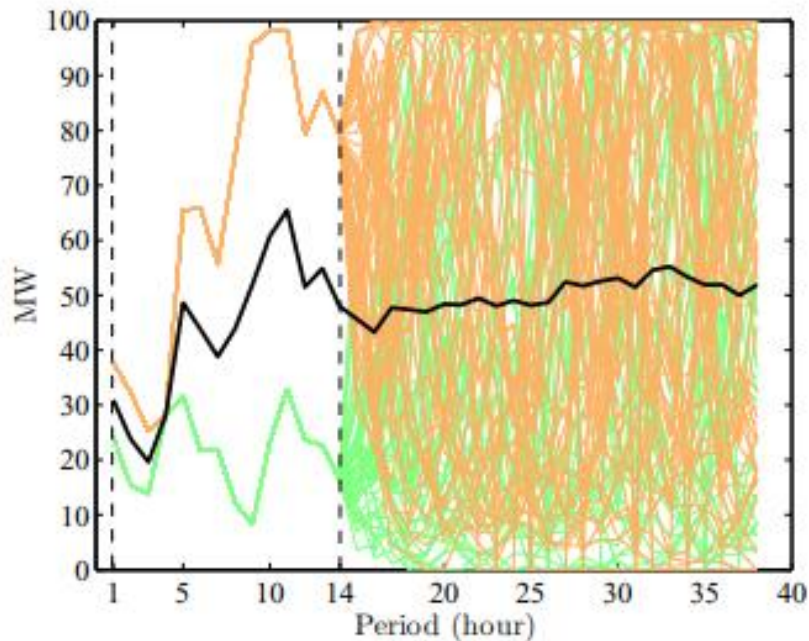


Figura 31 Valores medios de energía eólica observados en el mercado diario

Sin embargo, tal y como se menciona en el presente estudio, no se incluye un mercado intradiario, con lo que el valor que se usa en este trabajo será el valor conseguido tras la segunda generación de escenarios. Es decir, el único valor que se analiza en este trabajo es el último valor de viento que se obtiene tras la generación del viento.

Como se explica en puntos posteriores por motivos de sencillez se ha analizado solo en una unidad horaria, por lo que de toda la matriz de escenarios y probabilidades generado solo se usa el último valor.

El código que representa esta segunda parte de escenarios generados es el representado en la Figura 32.

```

%%DAY WIND SCENARIO GENERATION ( FROM DAY AHEAD WIND DATA)

load('curva.mat');

p = wblcdf(S_red, shape, scale);
YO_VALUES = norminv(p)
YO = zeros ( 2, scen_number)

M = 5;
v2_scen_array = zeros ( N,M)
p_scenari2_matrix = zeros ( N,M)
P_scen_2 = zeros (N*M, 1)
wind_scen_2 = zeros (N*M, 1)

k = 1
for i = 1:N
    YO(1,:) = YO_VALUES(1,i)
    YO(2,:) = YO_VALUES(2,i)
    [gen_scenariol, innovatiol] = simulate(ARMA_est, 7, 'NumPaths', scen_number, 'YO', YO) %montecarlo with initial data of YO

    data = {gen_scenariol}
    T = size(data{1},1); % timesteps
    % number of scenarios at t==T

    [S,P,U,U] = scenred(data, 'cityblock', 'nodes', round(linspace(1,M,T)));

    %get only useful data
    [nrow, ncol] = size(S)
    S_red_2 = S(nrow,:)
    P_red_2 = P(nrow,:)

    for ii=1:M
        P_v_scen_2(k) = P_red_2(1, ii)*P_red(1,i)
        wind_scen_2(k) = S_red_2(1, ii)
        k = k+1;
    end

    %v2_scen_array(i,:) = S_red_2
    %p_scenari2_matrix(i, :) = P_scen_2
end
    
```

Figura 32 Código escenario día en adelante

Para terminar con la generación de escenarios de viento lo que se hace es utilizando una curva de potencia de aerogenerador conseguir la potencia generada, la cual, es realmente el valor que se quiere conseguir, a partir de cada valor de viento, como se observa en la Figura 33

```

p_gen_scenariol = normcdf(wind_scen_2);
wind = wblinv(p_gen_scenariol, shape, scale)
val_gen_wind2 = interp1(curva(:,1), curva(:,2)/1000000*10, wind) %%Pt=val_gen_wind2
v_gen_scenariol = transpose(v_scen_step1)
    
```

Figura 33 Código para el cálculo de potencia generada

Los valores de potencia conseguidos es el vector “val\_gen\_wind2”

## 5.2. Generación y reducción escenarios precio mercado diario.

La generación de precios del mercado diario (Figura 34) sigue el mismo patrón general que para la generación de escenarios de viento, siendo el código correspondiente el siguiente

```

#####
% model day ahead price
➔ load('value.mat')

day_ahead_Mdl = arima('ARLags', 2, 'D', 0)
v=value.^2
ARMA_est = estimate(day_ahead_Mdl, v)

[gen_scenari, innovations] = simulate(ARMA_est, 30, 'NumPaths', 100);
day_gen_scen = real(gen_scenari.^(1/2));

%figure
%plot(gen_scenari(:,1).^2)
%figure
%plot(value)

R = 25
data = {day_gen_scen}
T = size(data{1},1); % timesteps
                    % number of scenarios at t==T

[S,P,U,L] = scenred(data, 'cityblock', 'nodes', round(linspace(1,R,T)));

%get only useful data
[nrow, ncol] = size(S)
S_red_2 = S(nrow,:)
P_red_2 = P(nrow,:)
P_scen_2 = P_red_2*P_red(1,i)

val_d_price = S_red_2;
P_d_price = P_red_2;
    
```

Figura 34 Código generación de precios del mercado diario

Como se puede apreciar en el código correspondiente (Figura 34), para la generación de precios no se ha necesitado normalizar los valores iniciales, ya que de inicio se ha considerado el precio diario como una variable estocástica normalizada (a diferencia del viento el cual sigue una distribución Weibull).



Un aspecto que es diferente es que se elevan los valores iniciales de precio  $\text{value}$  al cuadrado. A continuación, estos valores se usan para ajustar el modelo ARMA, para después realizar las simulaciones Montecarlo y la posterior reducción de escenarios. A continuación, se hace la raíz cuadrada de los valores. Esta parte de elevar el valor al cuadrado para después hacer la raíz cuadrada, es una transformación usada para poder aplicar el principio de normalidad, ya que en este caso los valores de precio no pueden ser negativos con lo que haciendo estas transformaciones solo se consiguen valores positivos sin llegar a alterar estadísticamente el valor del estocástico.

El valor de los escenarios después de la reducción es la siguiente (Figura 35):

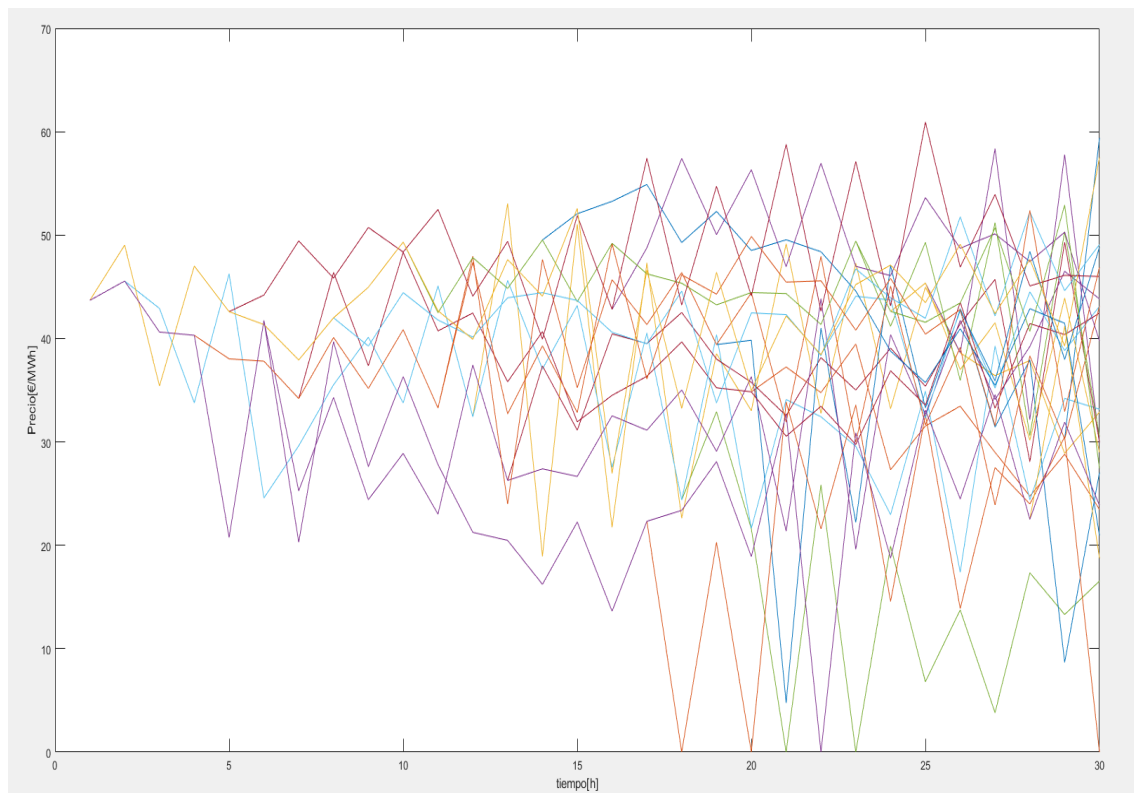


Figura 35 Escenarios después de la reducción

### 5.3. Generación y reducción escenarios precio mercado de balance.

En el mercado de balance lo que se trata de generar son valores de las ratios para el desvío a subir y a bajar. Las ratios representan el precio de venta/compra en el mercado de balance y el precio diario. Sin embargo, para caracterizar estas variables existe una problemática específica.

Como tal, las variables  $r^+$  y  $r^-$  por sus limitaciones por el funcionamiento del mercado no se pueden abstraer en modelos ARIMA, ya que por sus límites en valores superiores e inferiores (la

ratio del desvío a bajar puede ser máximo 1 y el desvío para bajar es mínimo 1) no puede aplicarse la asunción de normalidad. Sin embargo, aunque cada una de ellas representan variables estocásticas independientes, las dos están interrelacionadas por el funcionamiento del mercado. De esta manera, se puede representar la variable  $r$  definida por la función de la figura 35, la cual sí se puede abstraer en un modelo ARMA para luego generar escenarios a partir de ese modelo. La Figura 36 extraída de [23] muestra lo anteriormente descrito.

$$r_t = r_t^+ + r_t^- - 1$$

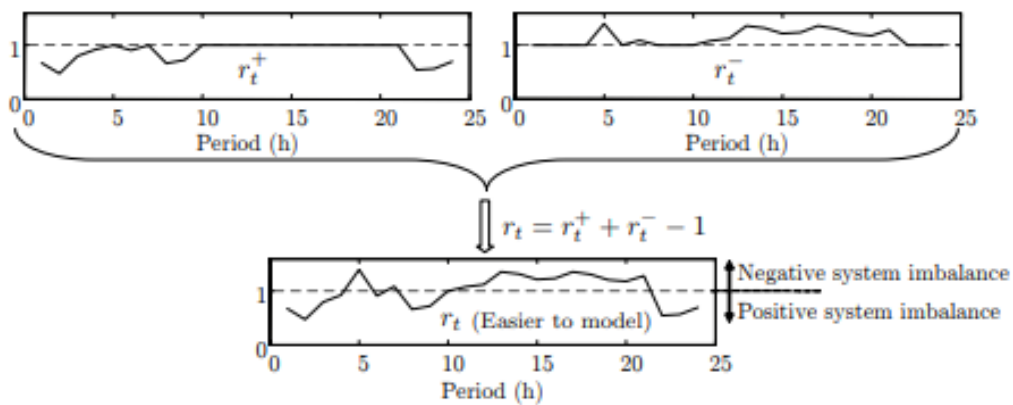


Figura 36 formulación de la variable  $r_t$

DESDE EL 01-11-2020 A LAS 00:00 HASTA EL 15-11-2020 A LAS 23:50 AGRUPADOS POR HORA

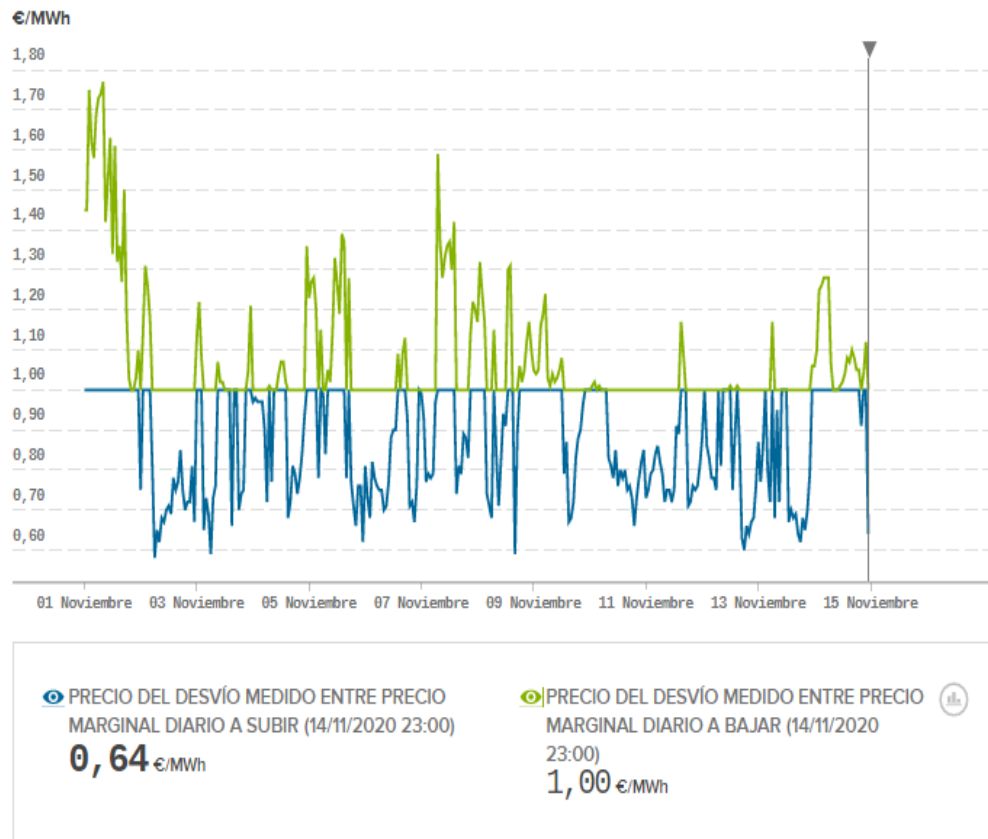


Figura 37 Precio del desvío medio entre precio marginal diario a subir y a bajar

El código que genera los escenarios de ratios de precio es el presentado en la Figura 38.

```

#####
%balance price

% model day ahead price
load('dev_ratio_subir.mat')
load('dev_ratio_bajar.mat')

r == dev_ratio_subir + dev_ratio_bajar -1

r_ahead_Mdl == arima('ARLags', 2, 'D', 0 )

ARMA_est == estimate(r_ahead_Mdl, r)

[gen_scenario, innovations] = simulate(ARMA_est, 30, 'NumPaths', 100);

%figure
%plot(gen_scenario(:,1).^2)
%figure
%plot(value)

R == 25
data == {gen_scenario}
T = size(data{1},1); % timesteps
% number of scenarios at t==T

[S,P,J,L] = scenred(data, 'cityblock','nodes',round(linspace(1,R,T)));

%get only useful data
[nrow, ncol] == size(S)
val_bal_price_ratio ==S(nrow,:)
P_bal_price_ratio ==P(nrow,:)

```

Figura 38 Código escenarios de ratios de precio

## 5.4. Resolución del problema estocástico.

### 5.4.1. Creación de vectores de escenarios a partir de valores conseguidos de cada variable estocástica.

En las secciones anteriores se han generado escenarios individuales para cada variable estocástica. Sin embargo, la cantidad de escenarios del problema final contiene todas las combinaciones posibles para las todas las variables en conjunto. En esta parte del algoritmo se va a realizar la combinación para tener en vectores cada escenario posible.

La sección del código que reordena todos los valores se presenta en la Figura 39.

```
#####  
%generate scenario vector  
n=1  
vector = [0, 0, 0, 0, 0]  
for i = 1:length(val_gen_wind2)  
    for j = 1:length(val_d_price)  
        for k = 1:length(val_bal_price_ratio)  
  
            if val_bal_price_ratio(k)>1  
                val_bal_price_ratio_pos=val_bal_price_ratio(k);  
                val_bal_price_ratio_neg=1;  
            elseif val_bal_price_ratio(k)<1  
                val_bal_price_ratio_pos=1;  
                val_bal_price_ratio_neg=val_bal_price_ratio(k);  
            else  
                val_bal_price_ratio_pos=1;  
                val_bal_price_ratio_neg=1;  
            end  
  
            vector(n, :) = [val_gen_wind2(i), val_d_price(j), val_bal_price_ratio_pos, val_bal_price_ratio_neg, P_v_scen_2(i)*P_d_price(j)*P_bal_price_ratio(k)];  
            n = n+1;  
  
        end  
  
    end  
  
end  
  
Pd = 0:0.1:20;  
|  
scenario_probability = vector(:,5);  
scenario_d_price= vector(:,2);  
bal_ratio_pos= vector(:,3);  
bal_ratio_neg= vector(:,4);  
Pt= vector(:,1);
```

Figura 39 Código para la creación de vectores de escenarios

#### 5.4.2. Solucionar problema base.

En esta sección del código se va a ejecutar el objetivo final que se quiere alcanzar con el algoritmo, es decir, optimizar la venta de energía eléctrica en el mercado.

Para la configuración del mercado que se ha configurado, tal y como señala [23] en sección 6.5.1. la expresión presentada en la Figura 40, del beneficio económico que se va a tener por la venta de energía es la siguiente.

$$\text{Maximize}_{P_t^D, \forall t; \Delta_{t\omega}^+, \forall t, \forall \omega; \Delta_{t\omega}^-, \forall t, \forall \omega}$$

$$\xi\{R\} = \sum_{\omega=1}^{N_{\Omega}} \sum_{t=1}^{N_T} \pi_{\omega} (\lambda_{t\omega}^D P_t^D d_t + \lambda_{t\omega}^D r_{t\omega}^+ \Delta_{t\omega}^+ - \lambda_{t\omega}^D r_{t\omega}^- \Delta_{t\omega}^-)$$

subject to

$$0 \leq P_t^D \leq P^{\max}, \forall t$$

$$\Delta_{t\omega} = d_t (P_{t\omega} - P_t^D), \forall t, \forall \omega$$

$$\Delta_{t\omega} = \Delta_{t\omega}^+ - \Delta_{t\omega}^-, \forall t, \forall \omega$$

$$0 \leq \Delta_{t\omega}^+ \leq P_{t\omega} d_t, \forall t, \forall \omega$$

$$0 \leq \Delta_{t\omega}^- \leq P^{\max} d_t, \forall t, \forall \omega.$$

Figura 40 Expresión del beneficio económico por venta de energía

La fórmula principal es básicamente el sumatoria de las esperanzas de beneficios obtenidos para cada escenario, siendo el beneficio en cada escenario la suma de ingresos por energía en el mercado diario con la suma de los ingresos por venta en el mercado de balance.

El objetivo es ajustar la Pd, la cual es la energía que se oferta para el mercado diario para conseguir maximizar el beneficio económico.

La resolución de esta parte del problema en el código es el de la Figura 41.

```
E = scenario_probability.*(scenario_d_price.*Pd + scenario_d_price.*bal_ratio_neg.*subplus(Pt-Pd)-scenario_d_price.*bal_ratio_pos.*subplus(Pd-Pt));
Esum = sum(E);
```

Figura 41 Código para el cálculo de beneficios

Haciendo un gráfico de E frente a Pd, se consigue la función de beneficios obtenidos para cada Pd, el cual facilita la comprensión del resultado obtenido, representado en la Figura 42. La discusión del resultado obtenido se hace en punto 6 de este trabajo.

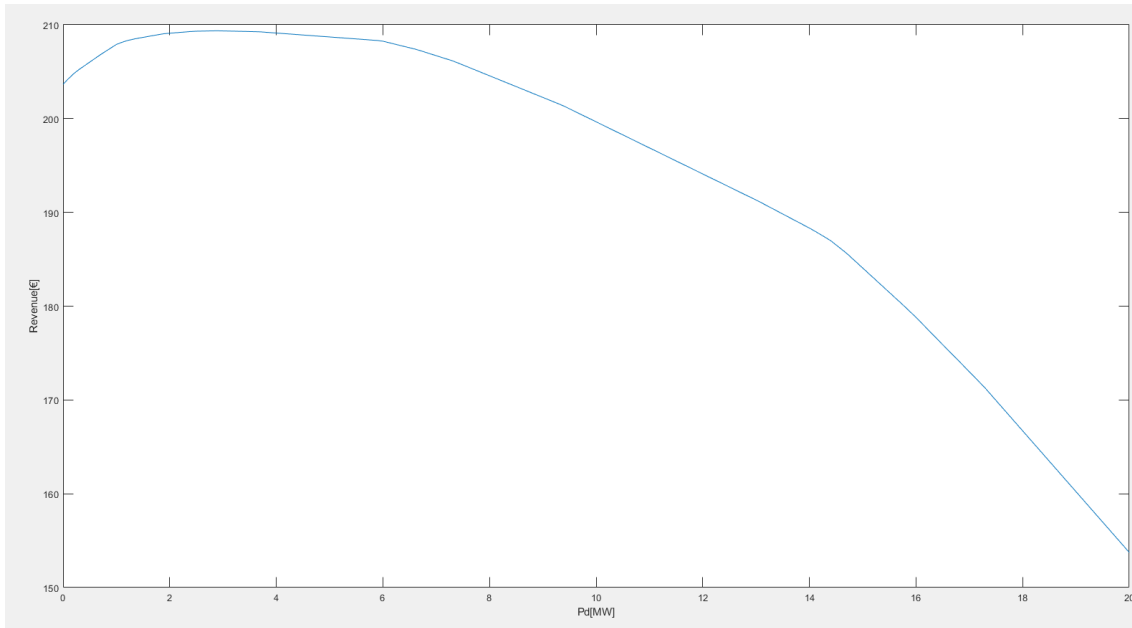


Figura 42 Beneficios obtenidos para cada Pd

De la Figura 42 se extrae que el Pd que se debería ofertar para el mercado diario con el fin de maximizar el beneficio económico debería ser un valor cercano a 3.5MW.

La discusión de los resultados obtenidos se hará en el punto 6 del presente documento

## 6. RESULTADOS OBTENIDOS. DISCUSIÓN.

### 6.1. Resolución al problema estocástico base.

En la Figura 42 se ha visto la solución gráfica al problema base planteado. El resultado en el que la esperanza de los escenarios devuelve un mayor beneficio económico para este caso estará cercano a 3. Este punto se puede intuir que será el punto que menor desvíos de energías negativos se espera conseguir. Por otro lado, se ve que si se desplaza la  $P_d$  (potencia ofertada en el mercado diario) hasta el 0, se sigue teniendo beneficio económico, aunque es menor que para el máximo. Esto es porque siendo 0 MW la oferta para el mercado diario, significa que toda la generación que se tenga se venderá en el mercado de balance, y el precio al que se va a poder vender esta energía va a ser igual o menor a la marcada en el mercado diario. Por otro lado, si se desplaza por el eje horizontal hasta el límite de 20 MW ofertados, se entiende que se deberá acudir al mercado de balance para comprar toda la energía que no se sea capaz de generar, de ahí que sea progresivamente menor.

A continuación, en el punto 6.2 se describe un ejemplo de aplicación para testear el control de baterías con el fin de anticipar la utilidad del modelo desarrollado para usarlo a la hora de testear controles de baterías diseñados

### 6.2. Ejemplo de aplicación para testear control de baterías.

En esta sección se implementa un ejemplo de cómo se podría usar el código desarrollado para optimizar y crear controles de almacenamiento para unidades generadoras (Figura 43).

Para ello, se va a hacer un ejemplo donde se puede ver el efecto que tendría un sistema de almacenamiento en un parque eólico.

Para ello, se usa un sistema de baterías y control con las siguientes características:

- 1 MW de potencia
- Capacidad de entregar y absorber energía infinita
- El control entrega o absorbe energía siempre en contra del sentido del desvío

Estas especificaciones en el código se traducen en la siguiente línea



```
Ebat = 1;
```

```
Ebat = scenario_probability.*(scenario_d_price.*Pd + scenario_d_price.*bal_ratio_neg.*subplus(Pt-Ebat-Pd)-scenario_d_price.*bal_ratio_pos.*subplus(Pd-Pt-Ebat));
```

```
Ebatsum = sum(Ebat);
```

Figura 43 Código para optimizar y crear controles de almacenamiento para unidades generadoras

Se realiza el gráfico de la aplicación con control de baterías junto con sin baterías, y el resultado es el siguiente

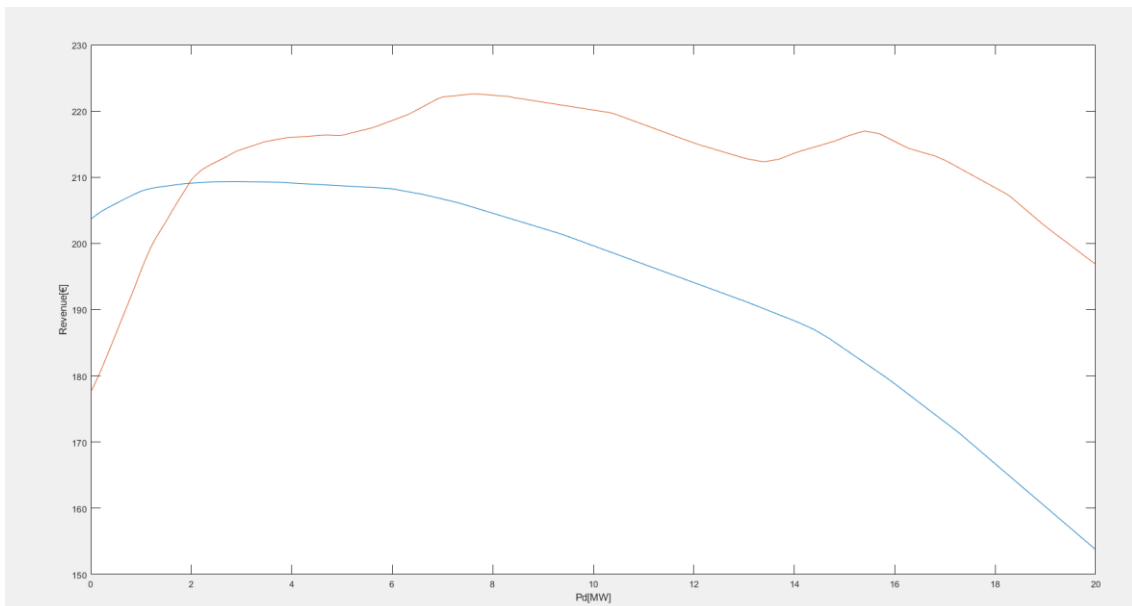


Figura 44 Beneficios obtenidos para cada Pd con y sin baterías

En la gráfica obtenida Figura 42 se puede apreciar con la línea roja el resultado del control de las baterías que se ha obtenido en el punto 6.2. En ella se aprecia que a Pd pequeños el beneficio económico se reduce, mientras que para Pd mayores el beneficio se incrementa.

Esto es debido al objetivo que se le ha fijado al algoritmo, que funciona siempre en contra del desvío de energía existente en el mercado en ese momento, sin tener en cuenta el beneficio económico esperado, por ello, cuando se ha ofertado 0 MW, el control de baterías entiende que esa energía es mejor absorberla, y por lo tanto no se podrá vender en el mercado de balance.

Al contrario, a potencias ofertadas mayor, el control de las baterías entiende en gran parte de los casos que se requerirá de más energía para alcanzar el compromiso marcado en el mercado diario, con lo cual aportará más energía.

Es evidente que el control de estas baterías está lejos de ser algo realista, e incluso el control es bastante “inocente”, sin embargo, sirve para poder anticipar la utilidad del modelo desarrollado para usarlo a la hora de testear controles de baterías diseñados.

## 7. CONCLUSIÓN

---

Para valorar lo que se ha conseguido en este Trabajo Fin de Máster, se debe entender que, más allá de los resultados discutidos en el punto 6, el desarrollo más importante realizado es del propio algoritmo y modelo realizado, el cual se debe interpretar como una herramienta que se puede usar para conseguir validar o desarrollar soluciones para diferentes áreas que tengan relación con la venta de electricidad.

En este sentido, es habitual que durante el desarrollo de una solución en ingeniería se optimice pensando solo en lecturas de medidas ya obtenidas, pero no se explora más allá, teniendo en cuenta todas las posibilidades de escenarios que hubiesen sido posibles.

Por ejemplo, poniendo un caso como podría ser el analizado, el control de baterías, si se hace el análisis del resultado de usar un determinado control con los valores de viento medidos durante un año, no se está teniendo en cuenta todos los valores de viento que hubiesen podido ser, con lo que el resultado que se obtenga estará siempre a merced de los resultados obtenidos.

De esta manera si se usa la generación de escenarios, se podrá conseguir una imagen más completa del problema.

Respecto a la optimización de la venta de energía, a opinión del autor es imprescindible tratar las variables con sus propiedades estocásticas, ya que de no hacerlo se estará tratando un problema sin toda la información disponible, lo que conlleva llegar a soluciones subóptimas.

## 8. LÍNEAS FUTURAS

---

Como líneas futuras, en caso de querer mejorar el algoritmo desarrollado para aumentar su precisión el autor recomienda comenzar analizando aquellas simplificaciones que se han realizado durante el desarrollo de este trabajo, siendo algunas de ellas las siguientes:

- Consideración de variables estocásticas independientes
- Exclusión del mercado de ajuste
- Uso de valores para los órdenes de los términos prefijados para el modelo ARIMA

En el caso de querer aumentar y extender las capacidades que se pueden obtener a través del modelo el autor recomienda explorar los siguientes puntos:

- Inclusión de gestión de riesgo en los resultados obtenidos.
- Incluir gestión multihoraria.

## 9. BIBLIOGRAFÍA

---

- [1] P. T. Lam, «Financing for renewable energy projects: A decision guide by developmental stages with case studies,» *Renewable and Sustainable Energy Reviews*, 2018.
- [2] A. Zakaria, «Uncertainty models for stochastic optimization in renewable energy applications,» *Renewable Energy*, 2019.
- [3] N. Pizarro, «RECONOCIMIENTO DE PATRONES Y PRONÓSTICO,» 2017.
- [4] R. Electrica, «Red Electrica España,» 2020. [En línea]. Available: <https://www.ree.es>.
- [5] E. Renovables, «Energías Renovables,» 2019. [En línea]. Available: <https://www.energias-renovables.com/panorama/asi-funciona-el-mercado-electrico-espanol-20191007>.
- [6] S. Watson, «Quantifying the variability of wind energy,» *WIREs Energy and Environment*, 2014.
- [7] A. Cilveti, *Indicadores y gestión del riesgo de unidades generadoras renovables en mercados eléctricos*, 2020.
- [8] N. Taleb, *Black Swan*, New York: Random House and Penguin Books, 2007.
- [9] N. Taleb, *Antifragile: Things That Gain from Disorder*, New York: Random House, 2012.
- [10] C. Moehrlen, «Uncertainty in wind energy forecasting,» *Cork Open Research Archive*, 2003.
- [11] K. Hoyland y W. Stein, «Generating Scenario Trees for Multistage Decision Problems,» *MANAGEMENT SCIENCE*, vol. 47, nº 2, pp. 295-307, 2001.
- [12] K. Hoyland, M. Kaut y S. Wallace, «A Heuristic for Moment-Matching Scenario Generation,» *Computational Optimization and Applications*, vol. 24, pp. 169-185, 2003.
- [13] J. Dupacova, G. Consigli y S. Wallace, «Scenarios for Multistage Stochastic Programs,» *Annals of Operations Research*, vol. 100, pp. 25-53, 2000.
- [14] J. Dupacova, N. Gröwe-Kuska y W. Römisch, «Scenario reduction in stochastic programming,» *Math. Program*, vol. 95, pp. 493-511, 2003.
- [15] H. Heitsch y W. Römisch, «Scenario reduction algorithms in stochastic programming,» *Computational Optimization and Applications*, vol. 24, pp. 187-206, 2003.

- [16] J. Higle y S. Sen, «Stochastic decomposition: An algorithm for two-stage linear programs with recourse,» *mathematics of operations research*, vol. 16, nº 3, pp. 650-669, 1991.
- [17] G. Box, G. Jenkins, G. Reinsel y G. Ljung, *Time Series Analysis*, wiley.
- [18] G. Marulanda, A. Bello y J. ., C. Cifuentes, «Wind Power Long-Term Scenario Generation considering Spatial-temporal dependencies in coupled electricity markets,» *Energies*, 2020.
- [19] S. Atique, S. Nouree y V. R. Roy, «Forecasting of total daily solar energy generation using ARIMA: a case study,» de *2019 IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC)*, Las Vegas, NV, 2019.
- [20] V. Ediger y S. Akar, «ARIMA forecasting of primary energy demand by fuel in Turkey,» *ELSEVIER*, nº 35, pp. 1701-1708, 2007.
- [21] C. Yildiz, M. Tekin y G. Ahmet, «A Day-Ahead Wind Power Scenario Generation, Reduction and Quality Test Tool,» *Sustainability*, vol. 9, 2017.
- [22] K. Sharma, P. Jain y R. Bhakar, «Wind Power Scenario Generation and Reduction in Stochastic Programming Framework,» *Electric Power Components and Systems*, vol. 41, 2013.
- [23] A. J. Conejo, M. Carrión y J. M. Morales, *Decision Making Under Uncertainty in Electricity Markets*, Springer US, 2010.
- [24] A. I. o. o. p. Domenica, «Stochastic programming and scenario generation within a simulation framework: An information systems perspective,» *Decision Support Systems*, 2007.
- [25] S. T. Rachev, *Probability metrics and the stability of stochastic models*, New York: Wiley, 1991.
- [26] H. F. L. Dani Gamerman, *Markov Chain Monte Carlo: Stochastic Simulation*, Chaoman and Hall/CRC, 2006.
- [27] K. B. Abdusamad, «Wind Energy Reliability Analysis based on Monte Carlo Simulation Method,» de *The First Conference for Engineering Sciences and Technology*, 2018.
- [28] B.-M. S. H. Zeiler, «Improved wind power forecasting with ARIMA models,» 2011.
- [29] L. C. M. William T Ziemba, *Stochastic Optimization Methods in Finance and Energy*, Spriner, 2011.

- [30] H. H. W. R. N. Growe-Kuska, «Scenario reduction and scenario tree construction for power management problem,» de *Power Tech Conference Proceedings, 2003 IEEE Bologna*, 2003.
- [31] U. P. Athanasio Papoulis, *Probability, Random Variables and Stochastic Processes*, McGraw-Hill, 2002.
- [32] J. T. Robert Gasch, *Wind Power Plants: Fundamentals, Design, Construction and Operation*, Springer, 2012.
- [33] R. E. España. [En línea]. Available: <https://www.esios.ree.es/es>.
- [34] M. R. Darryl Biggar, *The Economics of Electricity Markets*, Wiley, 2014.
- [35] S. Chapra, *Applied Numerical Methods with Marlab for engineers and scientist*, MacGraw-Hill, 2017.

## ANEXO CÓDIGO MATLAB

```
clear all

shape = 8;
scale = 1.61;

%wind_h_data corresponde a datos de entrada horarios de viento
wind_h_data = wblrnd(shape, scale, 1000, 1);

%figure;
%plot(wind_h_data)
%xlabel('Time')
%ylabel('Wind speed')

%%pdf
% v_range = 0:0.1:25;
% pdf = wblpdf(v_range, shape, scale);
%
% figure;
% plot(v_range, pdf)
% xlabel('Observatikon')
% ylabel('Probability Density')

%%cdf
% cdf = wblcdf(v_range, shape, scale);
%
% figure;
% plot(v_range, cdf)
% xlabel('Observation')
% ylabel('Cumulative Probability')

%normality of weibull
v = wind_h_data;

%sacar datos desde los valores wind_h_data
p = wblcdf(v, shape, scale);
normalized_wind_speed = norminv(p);

%figure;
%plot(normalized_wind_speed)
%xlabel('Time')
%ylabel('normalized Wind speed')

%arma fit
ARMA_mdl = arima('ARLags', 1:2, 'Constant', 0);
ARMA_est = estimate(ARMA_mdl, normalized_wind_speed);

%%DAY AHEAD WIND SCENARIO GENERATION
```



```
%montecarlo_simulations

scen_number = 100

[gen_scenariio, innovations] = simulate(ARMA_est, 23, 'NumPaths',
scen_number)

%figure
%plot(gen_scenariio)

%normal ->weibull

p_gen_scenariio = normcdf(gen_scenariio);
v_gen_scenariio =wblinv(p_gen_scenariio, shape, scale);

%figure
%plot(v_gen_scenariio)

data = {v_gen_scenariio}

%[S_tol,P_tol,J_tol] = scenred(data, 'cityblock','tol',0.2);

%scenario reduction

T = size(data{1},1); % timesteps
N = 5; % number of scenarios at t==T
[S,P,J,L] = scenred(data, 'cityblock','nodes',round(linspace(1,N,T)));

%get only useful data
[nrow, ncol] = size(S)
S_red =S((nrow-1):nrow,:)
P_red =P(nrow,:)

v_scen_step1 =S(nrow,:)

%%DAY WIND SCENARIO GENERATION ( FROM DAY AHEAD WIND DATA)

load('curva.mat');

p = wblcdf (S_red, shape, scale);
Y0_VALUES = norminv(p)
Y0 = zeros ( 2, scen_number)

M = 5;
v2_scen_array = zeros ( N,M)
p_scenariio2_matrix = zeros ( N,M)
P_scen_2 = zeros (N*M, 1)
wind_scen_2= zeros (N*M, 1)

k = 1
for i = 1:N
```

```

Y0(1,:) = Y0_VALUES(1,i)
Y0(2,:) = Y0_VALUES(2,i)
[gen_scenariol, innovations1] = simulate(ARMA_est, 7, 'NumPaths',
scen_number, 'Y0', Y0) %montecarlo with initial data of Y0

data = {gen_scenariol}
T = size(data{1},1); % timesteps
% number of scenarios at t==T

[S,P,J,L] = scenred(data, 'cityblock','nodes',round(linspace(1,M,T)));

%get only useful data
[nrow, ncol] = size(S)
S_red_2 =S(nrow,:)
P_red_2 =P(nrow,:)

for ii=1:M
    P_v_scen_2 (k) = P_red_2(1, ii)*P_red(1,i)
    wind_scen_2(k) = S_red_2(1, ii)
    k = k+1;
end

%v2_scen_array(i,:) = S_red_2
%p_scenariol_matrix(i, :) = P_scen_2

end

p_gen_scenariol = normcdf(wind_scen_2);
wind =wblinv(p_gen_scenariol, shape, scale)
val_gen_wind2 = interp1(curva(:,1),curva(:,2)/1000000*10,wind)
%%Pt=val_gen_wind2
v_gen_scenariol = transpose(v_scen_step1)

%%%%%%%%%
% model day ahead price
load('value.mat')

day_ahead_Mdl = arima('ARLags', 2, 'D', 0 )
v=value.^2
ARMA_est = estimate(day_ahead_Mdl, v)

[gen_scenariol, innovations] = simulate(ARMA_est, 30, 'NumPaths', 100);
day_gen_scen = real(gen_scenariol.^(1/2));

%figure
%plot(gen_scenariol(:,1).^2)
%figure
%plot(value)

R = 25
data = {day_gen_scen}
T = size(data{1},1); % timesteps

```

```
% number of scenarios at t==T

[S,P,J,L] = scenred(data, 'cityblock', 'nodes', round(linspace(1,R,T)));

%get only useful data
[nrow, ncol] = size(S)
S_red_2 =S(nrow,:)
P_red_2 =P(nrow,:)
P_scen_2 = P_red_2*P_red(1,i)

val_d_price = S_red_2;
P_d_price = P_red_2;

%%%%%%%%%%
%balance price

% model day ahead price
load('dev_ratio_subir.mat')
load('dev_ratio_bajar.mat')

r = dev_ratio_subir + dev_ratio_bajar -1

r_ahead_Mdl = arima('ARLags', 2, 'D', 0 )

ARMA_est = estimate(r_ahead_Mdl, r)

[gen_scenari, innovations] = simulate(ARMA_est, 30, 'NumPaths', 100);

%figure
%plot(gen_scenari(:,1).^2)
%figure
%plot(value)

R = 25
data = {gen_scenari}
T = size(data{1},1); % timesteps
                    % number of scenarios at t==T

[S,P,J,L] = scenred(data, 'cityblock', 'nodes', round(linspace(1,R,T)));

%get only useful data
[nrow, ncol] = size(S)
val_bal_price_ratio =S(nrow,:)
P_bal_price_ratio =P(nrow,:)

%%%%%%%%%%
%generate scenario vector
n=1
vector = [0, 0 ,0 , 0, 0]
```

```

for i = 1:length(val_gen_wind2)
    for j = 1:length(val_d_price)
        for k = 1:length(val_bal_price_ratio)

            if val_bal_price_ratio(k)>1
                val_bal_price_ratio_pos=val_bal_price_ratio(k);
                val_bal_price_ratio_neg=1;
            elseif val_bal_price_ratio(k)<1
                val_bal_price_ratio_pos=1;
                val_bal_price_ratio_neg=val_bal_price_ratio(k);
            else
                val_bal_price_ratio_pos=1;
                val_bal_price_ratio_neg=1;
            end

            vector(n, :) = [val_gen_wind2(i), val_d_price(j),
                val_bal_price_ratio_pos, val_bal_price_ratio_neg,
                P_v_scen_2(i)*P_d_price(j)*P_bal_price_ratio(k)];
            n = n+1;

        end

    end

end

Pd = 0:0.1:20;

scenario_probability = vector(:,5);
scenario_d_price= vector(:,2);
bal_ratio_pos= vector(:,3);
bal_ratio_neg= vector(:,4);
Pt= vector(:,1);

E = scenario_probability.*(scenario_d_price.*Pd + scenario_d_price.*bal_ratio_neg.*subplus(Pt-Pd)-scenario_d_price.*bal_ratio_pos.*subplus(Pd-Pt));
%Escen_pdopt = (scenario_d_price.*Pd + scenario_d_price.*bal_ratio_neg.*subplus(Pt-Pd)-scenario_d_price.*bal_ratio_pos.*subplus(Pd-Pt))
Esum = sum(E);
plot(Pd, Esum);
xlabel('Pd[MW]')
ylabel('Revenue[€]')
hold on;

Pbat = 1;

Ebat = scenario_probability.*(scenario_d_price.*Pd + scenario_d_price.*bal_ratio_neg.*subplus(Pt-Pbat-Pd)-scenario_d_price.*bal_ratio_pos.*subplus(Pd-Pt-Pbat));
Ebatsum = sum(Ebat);

```

```
plot(Pd, Ebatsum);
xlabel('Pd[MW]')
ylabel('Revenue[€]')
hold off;

%E = revenue_risk_free(scenario_probability,Pt, scenario_d_price,bal_ratio_neg, bal_ratio_pos, Pd)

x = fmincon(@(Pd, omega, phi) revenue_risk_free(scenario_probability,Pt, scenario_d_price,bal_ratio_neg, bal_ratio_pos,Pd), [0,0,0], [[1,0,0];[-1,0,0]];[sum(Pt.*(scenario_d_price+scenario_d_price.*bal_ratio_pos + scenario_d_price.*bal_ratio_neg)),1,1],[20;0;sum(Pt.*scenario_d_price.*(bal_ratio_pos-bal_ratio_neg))])

%%%%%%%%%%%%%
%function

function Esum_f = revenue_risk_free(scenario_probability,Pt, scenario_d_price,bal_ratio_neg, bal_ratio_pos, variables)
    Pd = variables(1);
    omega = variables(2);
    phi = variables(3);

    E = scenario_probability.*(scenario_d_price.*Pd + scenario_d_price.*bal_ratio_neg.*subplus(Pt-Pd)-scenario_d_price.*bal_ratio_pos.*subplus(Pd-Pt));
    Esum_f = -(sum(E) + 1*(omega-sum(scenario_probability.*phi)/(1-0.095)));
end
```