

Trabajo Final de Grado

Ingeniería en tecnologías Industriales

Diseño y Simulación de un Modelo predictivo para la Velocidad y Dirección del viento

MEMORIA

Autor: Tornike Gongadze Gogvadze
Director/es: M^a Isabel García Planas, José Luís Domínguez
Convocatoria: Mayo 2016



Escuela Técnica Superior de Ingeniería
Industrial de Barcelona



A quien ha estado siempre a mi lado/ მისთვის ვინც ჩემს გვერდით ყოფილა

"Iterar es humano, 'recursivar' es divino"

-- L. Peter Deutsch

Resumen

El trabajo realizado está englobado en el área de Álgebra lineal (Cadenas de Markov) y de programación informática (Matlab) aplicada al estudio de las condiciones del viento para la posible implementación de un molino eólico.

El objetivo principal es diseñar y simular un modelo predictivo para la velocidad y dirección del viento mediante Cadenas de Markov. Una cadena de Markov es un proceso aleatorio sin memoria en el que básicamente conocido el estado 'presente' de un sistema, el estado 'futuro' no depende del estado 'pasado'.

Esto se lleva a cabo mediante la creación de dos programas (uno para velocidades y otro para direcciones) en Matlab que consiguen analizar datos de velocidad y dirección del viento obtenidos de una estación meteorológica cualquiera. La meta es demostrar que, si los datos analizado son modelables como una Cadena de Markov (también se validan los datos estudiados), la dirección y la velocidad a la que tenderá el viento en el futuro no depende de los estados pasados del mismo, y basta con conocer un estado inicial actual para saber la tendencia futura.

Además, se aplica el programa creado a una estación meteorológica, concretamente la de Trinidad de Iturgoyen, Navarra. Se estudia el comportamiento del viento a lo largo del periodo que va desde Marzo de 2015 hasta Marzo de 2016 y se analiza la tendencia a la que habitualmente se aproximará tanto la velocidad como la dirección del viento. Para ello se realizan varias simulaciones. Primero con todos los datos descargados para ver la tendencia global, y más adelante dos simulaciones de dos días aleatorios del año para ver la predicción a corto plazo de las condiciones del viento.

La idea es poder utilizar el diseño realizado para poder determinar si una zona concreta tiene las condiciones óptimas como para colocar un molino o parque eólico. Por ello todos los resultados analizados en la estación estudiada se comparan a las condiciones necesarias para el correcto funcionamiento y una generación eléctrica notable de un molino eólico.

En conclusión, se ha conseguido realizar un diseño que determina si las condiciones del viento en una zona concreta son favorables para la posterior implementación de un molino eólico, una fuente de energía renovable cuyo uso está en crecimiento.

Sumario

Resumen	5
Glosario	13
1. Prefacio	15
1.1. Motivación del proyecto	15
1.2. Requisitos previos	15
2. Introducción	17
2.1. Objetivos del trabajo	17
2.2. Alcance del trabajo	17
3. Bases	19
3.1. Conceptos básicos de Álgebra Lineal.....	19
3.1.1. Propiedades Espectrales.....	19
3.1.2. Propiedades de las matrices no negativas	22
3.2. Introducción a las Cadenas de Markov.....	23
3.3. Qué es Matlab	25
3.4. Interés del Estudio del Viento	26
3.5. Obtención de datos de una estación Meteorológica	27
4. Diseño del Modelo. Creación de un programa con Matlab	29
4.1. Lectura de Datos	29
4.2. Tratamiento de la Información	30
4.3. Generación de la matriz de Transición	38
4.4. Comprobación de Datos Markovianos.....	41
4.4.1. Fichero ‘condicionesProbabilidad.m’	42
4.4.2. Fichero ‘EsMarkovianoVelocidad.m’.....	43
4.4.3. Fichero ‘EsMarkovianoDireccion.m’	46

4.4.4.	Gráficos de comprobación.....	47
4.4.5.	Gráficos extra para la comprobación del carácter Markoviano.....	48
5.	Estudio mediante Cadenas de Markov y Predicción	51
5.1.	Cálculo de Valores propios y Vectores propios	51
5.2.	Búsqueda de la distribución estacionaria.....	55
5.2.1.	Distribución estacionaria para Velocidades del viento.....	55
5.2.2.	Distribución estacionaria para Direcciones del viento.....	57
5.3.	Análisis de resultados	58
5.3.1.	Análisis de resultados para Velocidad	59
5.3.2.	Análisis de resultados para Direcciones	60
5.3.3.	Conclusión global	61
5.4.	Ejemplos específicos	61
5.4.1.	Día 14 de Abril de 2015.....	61
5.4.2.	Día 22 de Noviembre de 2015.....	63
6.	Planificación.....	65
7.	Presupuestos	67
8.	Impacto Ambiental.....	69
	Conclusiones.....	71
	Agradecimientos	73
	Bibliografía.....	75
	Anexos	77

Sumario de figuras

Figura 1: Estación de Trinidad de Iturgoyen	27
Figura 2: Rosa de los Vientos.....	31
Figura 3: Clasificación de estados del viento.....	32
Figura 4: Energía eléctrica generada	32
Figura 5: Energía eléctrica generada a bajas velocidades	33
Figura 6: Gráfico de datos de dirección*.....	36
Figura 7: Gráfico de datos de velocidad*	36
Figura 8: Gráfico de los Datos de Velocidad de 1 día aleatorio*.....	37
Figura 9: Gráfico de los Datos de Dirección de 1 día aleatorio*	37
Figura 10: Comprobación de datos markovianos para las velocidades del viento	47
Figura 11: Comprobación de datos markovianos para las direcciones del viento	48
Figura 12: Caso de 1 probabilidad velocidades.....	49
Figura 13: Caso de 1 probabilidad direcciones.....	49
Figura 14: Diagrama de Gantt del proyecto.....	65
Figura 15: mundo sostenible	69

Sumario de tablas

Tabla 1: Datos importados a Excel (desde la estación de Iturgoyen)	29
Tabla 2: Estados asignados a las direcciones.....	31
Tabla 3: Estados asignados a los datos de velocidad	33
Tabla 4: Resultados globales velocidad	59
Tabla 5: Resultados globales dirección	60
Tabla 6: Resultados Dirección 14 de Abril.....	62
Tabla 7: Resultados Velocidad 14 de Abril	62
Tabla 8: Resultados Dirección 22 de Noviembre.....	63
Tabla 9: Resultados Velocidad 22 de Noviembre.....	63
Tabla 10: Duración de las tareas realizadas.....	65
Tabla 11: Presupuestos.....	67

Glosario

Software: Conjunto de programas, instrucciones y reglas informáticas para ejecutar ciertas tareas en una computadora.

Programación informática: proceso de diseñar, codificar, depurar y mantener el código fuente de programas computacionales.

VAP: Valor Propio

VEP: Vector Propio

Matriz cuadrada: Matriz que tiene el mismo número de filas que de columnas.

Matriz diagonal: Matriz cuadrada en la que todas sus entradas son nulas excepto las de la diagonal principal.

Matriz Identidad: es el elemento neutro del producto de las matrices. Consiste en una matriz diagonal cuyas entradas en la diagonal principal son iguales a 1.

Matriz invertible: Se dice de una matriz cuadrada A que es invertible, cuando existe otra matriz cuadrada denotada como A^{-1} tal que $A \cdot A^{-1} = Id$, donde Id es la matriz identidad.

$M_{n \times m}$: Conjunto de matrices con número de filas n y número de columnas m . Por ejemplo $M_{n \times n}$ es el conjunto de matrices cuadradas.

\mathbb{R} : Conjunto de números reales

\mathbb{C} : Conjunto de números complejos

Matlab: *Matrix laboratory*, laboratorio de matrices / actualmente *Mathematics Laboratory*, Laboratorio de matemáticas

Energía eólica: Energía obtenida a partir del viento. Conversión de la energía cinética generada por las corrientes del aire a energía útil para la actividad humana.

1. Prefacio

1.1. Motivación del proyecto

La motivación que he tenido para realizar este proyecto es particularmente diferente a lo habitual. Si lo habitual es la gran destreza o afición por el tema escogido, en mi caso ha sido diferente. Si bien es cierto que las matemáticas me apasionan desde pequeño, siempre he tenido más habilidad en la rama de Cálculo Integral que en el de Álgebra Lineal, en la que he tenido más complicaciones. De la misma manera que este último, tuve ciertas dificultades con la asignatura Informática, en la que principalmente se aprende a programar con Python. Mis cualificaciones en el primer curso de esta carrera así lo demuestran.

Precisamente, tuve la motivación de realizar este proyecto para 'quitarme la espina' que me queda en estos dos ámbitos desde el inicio de la carrera. Tanto el Álgebra Lineal como la Informática son probablemente los temas que más problemas me han traído, y el hecho de que este trabajo una a la perfección (con Matlab) estos dos temas me motivó para demostrar que sí que estoy preparado y cualificado para abordarlos. Además, el hecho de poder hacer uso del programa Matlab, que hoy en día tiene infinidad de aplicaciones y utilidades tanto en lo profesional como en lo académico, también supuso un aliciente.

Esto, unido a las posibles aplicaciones futuras del proyecto me decantó claramente para realizar este proyecto en vez de otros que quizás se ajustaban más a mis mejores habilidades académicas.

1.2. Requisitos previos

Evidentemente, este proyecto requiere unos conocimientos mínimos de Informática y Álgebra Lineal. Muchos de los conceptos teóricos que se aplican en este proyecto se han adquirido a lo largo de toda la carrera, en especial con asignaturas como Álgebra Lineal, fundamentos de Informática e Informática.

De todas formas, también ha sido necesaria una gran carga de aprendizaje autónomo. En primer lugar, la programación empleada en Matlab no se realiza con el mismo código que el empleado en Python, que es el que se aprende en ETSEIB, sino que se realiza con Código C, que es muy similar conceptualmente pero tiene diferencias que hay que estudiar. En segundo lugar, la teoría de Cadenas de Markov, que es la que se aplica en el proyecto, en sí misma no se imparte a lo largo de la carrera, por lo que también es necesario dedicarle su tiempo, si bien los conceptos básicos necesarios para una buena comprensión, sí que forman parte de su programa docente.

2. Introducción

2.1. Objetivos del trabajo

El objetivo de este trabajo es el diseño y simulación de un modelo predictivo para la dirección y velocidad del viento mediante Cadenas de Markov. Esto se llevará a cabo con la creación de un programa utilizando Matlab, que consiga leer datos meteorológicos que incluyan tanto las velocidades como las direcciones, y a partir de ello prediga unas condiciones futuras o una tendencia futura de las condiciones, en la medida que sea posible. El soporte teórico aplicado para realizar el modelo predictivo es el de Cadenas de Markov, que básicamente se refiere a un proceso aleatorio sin memoria.

La meta principal es conseguir un programa lo suficientemente bueno como para que se puedan analizar tantos datos meteorológicos como se deseen y aplicar el mismo a los datos de una estación meteorológica concreta para demostrar que funciona. De esta manera se llevará a cabo el estudio y predicción de condiciones futuras en una estación concreta, verificando si es o no un lugar con condiciones estables del viento y si es un buen sitio para colocar, por ejemplo, unos molinos eólicos para generar electricidad.

2.2. Alcance del trabajo

Evidentemente, como todo trabajo, tiene sus limitaciones. La principal limitación de este proyecto es que se trata de un TFG (trabajo final de grado) y por tanto sólo supone 12 créditos. Este hecho, limita a su vez el tiempo que se dedica al proyecto (que apenas llega a 300 horas en total), por lo que la profundidad del proyecto es limitada. Aun así se consigue realizar un trabajo completo que se puede aplicar a cualquier estación meteorológica.

El trabajo realizado aborda los siguientes puntos principales:

- Bases: conceptos fundamentales de Álgebra lineal y Cadenas de Markov,
- Qué es Matlab, interés del estudio y elección de una estación meteorológica
- Diseño del modelo mediante la creación de un programa con Matlab
- Simulación del modelo con la predicción de condiciones en un estación meteorológica

3. Bases

3.1. Conceptos básicos de Álgebra Lineal

Antes de iniciar la realización del programa para la predicción de los estados del viento se explican brevemente aspectos de Álgebra Lineal que se deben conocer para poder entender ciertas partes del trabajo. Cabe destacar que el objetivo de este apartado es mencionar los conceptos estrictamente necesarios, sin profundizar en la explicación teórica de los mismos y suponiendo como sabidos muchos otros conceptos matemáticos previos. Hay mucha bibliografía relativa al tema de Algebra Lineal, y este proyecto se ha basado en diversos textos que tratan los conceptos explicados ([5], [6]). Sí se habla en relación a matrices no negativas, la bibliografía es menos extensa y más específica, y el trabajo se ha basado en textos más concisos ([1], [4]).

Para el estudio mediante Cadenas de Markov que se realiza en este proyecto, es imprescindible conocer las Propiedades espectrales, y más concretamente la teoría de valores propios y vectores propios para matrices cuadradas, y algunas propiedades de las matrices no negativas.

3.1.1. Propiedades Espectrales

El objetivo del problema espectral es encontrar aquellos valores $\lambda \in \mathbb{C}$ y vectores no nulos $v \in \mathbb{C}^n$ de manera que se cumpla:

$$AV = \lambda V$$

Siendo:

- A una matriz cuadrada compleja dada, de orden n
- V es la matriz de orden $n \times 1$ cuyos elementos son las coordenadas del vector v
- λ es un escalar que recibe el nombre de Valor Propio (que para abreviar notaremos por VAP) de la matriz A
- v es el Vector Propio (que para abreviar notaremos por VEP) asociado al VAP λ de A

Es decir, los vectores propios de una matriz son aquellos vectores no nulos que, cuando son transformados por la matriz, dan lugar a un múltiplo escalar (valor propio) de sí mismos, por lo que no cambian su dirección.

Pasamos a dar a continuación, las definiciones formales de valor y vector propio.

Definición de Valor Propio: Un escalar $\lambda \in \mathbb{C}$ recibe el nombre de *Valor propio (VAP)*

de A si existe un vector columna $v \in M_{nx1}(\mathbb{C})$ tal que:

$$v \neq 0 \text{ y } Av = \lambda v$$

Definición de Vector Propio: Un vector columna $v \in M_{nx1}(\mathbb{C})$ recibe el nombre de *Vector Propio (VEP)* de A si $v \neq 0$ y si existe $\lambda \in \mathbb{C}$ tal que $Av = \lambda v$.

Se puede observar que cada VAP va directamente relacionado con un VEP, por lo que se dice que λ es un VAP asociado al VEP v y que v es un VEP asociado al VAP λ .

Llamaremos espectro de A al conjunto de sus valores propios.

Radio espectral

El radio espectral de una matriz es igual al mayor módulo de entre los módulos de los valores propios. Si A es una matriz cuadrada de orden n , entonces el radio espectral es igual a:

$$\rho(A) = \max(|\lambda_1|, \dots, |\lambda_n|)$$

Cálculo de los valores propios. Polinomio Característico de una Matriz

Cabe destacar que los métodos para encontrar los valores propios no son imprescindibles para este trabajo ya que el programa Matlab los encuentra directamente. Con esta introducción teórica solo se ven las partes más básicas para poder entender el proyecto, pero se dejan de explicar muchas propiedades y fundamentos teóricos importantes.

Para encontrar los valores explicados, es de gran utilidad utilizar el *Polinomio Característico de la Matriz* ($Q_A(t)$). cuyas raíces son precisamente los valores propios de la matriz. Si se considera una matriz A cuadrada de orden n , entonces dicho polinomio se define de la siguiente manera:

$$Q_A(t) = \det(A - tI_n)$$

Dicho en palabras, el polinomio característico es el determinante de la matriz $(A - tI_n)$.

Observamos que si $AV=tV$, se tiene que $(A-tI_n)V=0$, entonces $\text{rango}(A-tI_n) < n$ y por lo tanto $\det(A-tI_n)=0$, y recíprocamente. Tenemos pues que es equivalente buscar valores propios que buscar las raíces del polinomio característico.

Cálculo de los vectores propios

Como hemos visto si λ es un valor propio entonces $\text{rango}(A-\lambda I) < n$ por lo que existe un vector v tal que $(A-\lambda I)v=0$. Por lo que

$$v \in \text{Ker}(A-\lambda I).$$

Diagonalización de una matriz

Las matrices diagonales son muy útiles ya que es mucho más sencillo trabajar con ellas a nivel operativo. Por ejemplo, nos permiten resolver de forma fácil sistemas de la forma

$$x(k+1)=Ax(k), x(0)=x_0$$

ya que su solución viene dada por

$$x(k)=A^k x_0$$

y si la matriz es diagonal, sus potencias son las potencias de los elementos de la diagonal:

$$D = \begin{pmatrix} \lambda_1 & 0 & \dots & 0 \\ 0 & \lambda_2 & 0 & \vdots \\ \vdots & 0 & \ddots & 0 \\ 0 & \dots & 0 & \lambda_n \end{pmatrix} \quad D^k = \begin{pmatrix} \lambda_1^k & 0 & \dots & 0 \\ 0 & \lambda_2^k & 0 & \vdots \\ \vdots & 0 & \ddots & 0 \\ 0 & \dots & 0 & \lambda_n^k \end{pmatrix}$$

Nos preguntamos si es posible reducir una matriz no diagonal a una diagonal conservando las propiedades espectrales de dicha matriz.

Una matriz A de orden n es diagonalizable si existe una matriz S invertible tal que $SAS^{-1} = D$, siendo D una matriz diagonal como la vista anteriormente.

Con esta definición es fácil observar que las matrices A y D tienen el mismo espectro.

Para obtener esta matriz D diagonal se deben empezar buscando los VAPS y los VEPS asociados a A que en caso de existir nos proporcionarán la matriz D y la matriz S invertible mencionada.

Una vez obtenidos los VAPS y los VEPS, ya se tienen tanto la matriz D (que tiene en la diagonal los VAPS), como la matriz S , que tiene por columnas los VEPS hallados, en el mismo orden que los VAPS aparecen en la diagonal de D , de la siguiente forma:

$$S = \begin{pmatrix} \vdots & \vdots & \vdots & \vdots \\ v_1 & v_2 & \cdots & v_n \\ \vdots & \vdots & \vdots & \vdots \end{pmatrix}$$

3.1.2. Propiedades de las matrices no negativas

En el caso particular que nos ocupa las matrices que utilizamos nos indican la probabilidad de que en un momento determinado el viento tenga una u otra dirección y una u otra velocidad, por lo que los datos de la matriz son no negativos, por lo que vamos a continuación a introducir alguna información sobre este tipo de matrices. Empezamos definiendo lo que es una matriz no negativa. Una matriz no negativa es aquella en la que todos los coeficientes son mayores o iguales a 0.

De manera más formal, una matriz $A \in M_{n \times m}(\mathbb{R})$ es no negativa si todos sus elementos son mayores o iguales a 0 ($A \geq 0$).

Dentro de las propiedades que verifican en lo que se refiere a matrices no negativas, hay algunas que son de más relevancia para este proyecto. Concretamente, es importante conocer los siguientes tipos de matrices y sus propiedades y que se detallan a continuación:

- Matriz Irreducible
- Matriz Primitiva
- Matriz Estocástica

Matriz Irreducible

La definición de esta matriz es meramente conceptual, ya que a lo largo del trabajo las matrices con las que se trabaja son irreducibles.

Definición de matriz reducible: Una matriz $A \in M_n(\mathbb{R})$ ($n \geq 2$) es reducible si existe una matriz permutación P , cuyas columnas son una reordenación de las columnas de la matriz identidad, tal que:

$$PAP^{-1} = \begin{pmatrix} B & 0 \\ C & D \end{pmatrix}$$

Donde B y D son matrices cuadradas.

Si esta matriz P no existe, se dice que la matriz es **irreducible**.

Matriz Primitiva

Dentro de las diversas definiciones de matriz primitiva, la que más interesa para este trabajo es la que viene a continuación.

Definición de matriz primitiva: Sea A una matriz no negativa e irreducible. Se dice que A es una matriz primitiva si y sólo si tiene un único valor propio de módulo igual a su radio espectral. En caso contrario, se dice que la matriz es imprimitiva.

Matriz Estocástica

Se verá, más adelante, que las matrices que se utilizarán serán estocásticas por columnas. Por tanto es importante conocer la definición y algunas de sus propiedades.

Definición de matriz estocástica (por columnas): Si $A \in M_n(\mathbb{R})$ es una matriz donde a_{ij} son los elementos de cada fila i y columna j , entonces se dice que la matriz es estocástica (por columnas) si todos los elementos de cada columna suman 1. Es decir:

$$a_{ij} \geq 0, \quad \sum_{i=1}^n a_{ij} = 1 \text{ para cada } j = 1, \dots, n$$

Es importante tener en cuenta que, si una matriz no negativa es estocástica, entonces su radio espectral es 1, por lo que el VAP con mayor módulo es 1 y este es único.

3.2. Introducción a las Cadenas de Markov

Las cadenas de Markov (ver [1], [5], [16]) constituyen una herramienta para estudiar el comportamiento de procesos que evolucionan de forma no determinista, a lo largo del tiempo sobre un conjunto de estados posibles. Esta herramienta es considerada esencial en disciplinas como la ingeniería, la economía y la investigación operativa entre otros.

De manera más formal, una *cadena de Markov* es un proceso aleatorio sin memoria. Esto es, un sistema que evoluciona en el tiempo ($k = 0, 1, \dots$), está en el estado i_k en el instante k , si en el instante $k - 1$ está en el estado i_{k-1} :

$$P(X_k = i_k | X_0 = i_0, \dots, X_{k-1} = i_{k-1}) = P(X_k = i_k | X_{k-1} = i_{k-1}) = p_{i_{k-1}, i_k}$$

Intuitivamente, esta ecuación implica que, conocido el estado “presente” de un sistema, el estado “futuro” es independiente de su estado “pasado”.

Las cadenas de Markov se pueden describir matricialmente de la siguiente manera. Si $p_1(k), \dots, p_n(k)$ son las probabilidades de los n estados en el tiempo k y $p_{ij}(k)$ la probabilidad de la evolución del estado i al j en el tiempo k , $1 \leq i, j \leq n$. Entonces,

$$\left. \begin{aligned} p_1(k+1) &= p_{11}(k)p_1(k) + \dots + p_{1n}(k)p_n(k) \\ &\vdots \\ p_n(k+1) &= p_{n1}(k)p_1(k) + \dots + p_{nn}(k)p_n(k) \end{aligned} \right\}$$

lo que matricialmente tiene la siguiente forma:

$$\begin{pmatrix} p_1(k+1) \\ \vdots \\ p_n(k+1) \end{pmatrix} = \begin{pmatrix} p_{11}(k) & \cdots & p_{1n}(k) \\ \vdots & \ddots & \vdots \\ p_{n1}(k) & \cdots & p_{nn}(k) \end{pmatrix} \begin{pmatrix} p_1(k) \\ \vdots \\ p_n(k) \end{pmatrix}$$

Para el caso estudiado, interesa conocer las cadenas de Markov homogéneas. En primer lugar, se denomina Matriz de transición de estados, que incluye las probabilidades de evolucionar de un estado i a un estado j en el tiempo k ($1 < i, j < n$), a la siguiente matriz:

$$A(k) = \begin{pmatrix} p_{11}(k) & \cdots & p_{1n}(k) \\ \vdots & \ddots & \vdots \\ p_{n1}(k) & \cdots & p_{nn}(k) \end{pmatrix}$$

Entonces, se dice que una cadena es homogénea si ninguna de los términos $p_{ij}(k)$ depende de k , que será el caso a lo largo de todo el proyecto.

En el caso de cadenas homogéneas, se cumple:

$$p(k) = A^k p(0)$$

Es decir, la probabilidad en el tiempo k , es igual a la Matriz de transición de estado elevado al tiempo k multiplicado por un estado inicial $p(0) = (p_1(0), \dots, p_n(0))$

Por último, es necesario definir el concepto de distribución estacionaria de una cadena de Markov. Se denomina distribución estacionaria de una cadena en caso de existir el siguiente:

$$\pi = \lim_{k \rightarrow \infty} p(k)$$

Y si se trata de una cadena homogénea, entonces:

$$\lim_{k \rightarrow \infty} p(k) = \lim_{k \rightarrow \infty} A^k p(0) = (\lim_{k \rightarrow \infty} A^k) p(0)$$

En caso de que la matriz A sea irreducible, existe el límite mencionado. Además, si la matriz de transición es primitiva, que es el caso estudiado, entonces el límite π es independiente de la distribución inicial de probabilidad $p(0)$ e igual al VEP de VAP 1 de la matriz:

$$\pi = \frac{1}{\sum v_i} \begin{pmatrix} v_1 \\ \vdots \\ v_n \end{pmatrix} = \text{VAP de VEP 1, normalizado}$$

3.3. Qué es Matlab

Esta herramienta de software matemático llamada MATLAB fue creada por el matemático y programador de computadoras Cleve Moler en 1984, con la idea de poder ser usado en el campo del Álgebra lineal y análisis matricial.

El nombre de dicha herramienta viene de la abreviatura de 'MATrix LABoratory', es decir, laboratorio de matrices.

Inicialmente, MATLAB se creó para proporcionar fácil acceso a las librerías LINPACK [8] y EISPACK [9], dos de las más importantes en computación y cálculo matricial. Con el tiempo, sus usos se han expandido. Actualmente, se usa para llevar a cabo proyectos que impliquen un elevado cálculo matemático y de visualización de resultados. Además, hoy en día, MATLAB se ha implementado en numerosas escuelas y universidades, convirtiéndose en una herramienta básica para los docentes e investigadores. Se utiliza en muchas asignaturas de grado y master de ciencias e ingenierías a todos los niveles, desde Álgebra lineal hasta sistemas de control.

Cabe destacar que MATLAB contiene diferentes módulos que amplían aún más su uso. Al paquete básico, utilizado para programación de funciones matemáticas y resolución de problemas, se suman diferentes cajas de tareas que se utilizan para cálculos estadísticos o de optimización, sistemas de control, procesamiento de señales y comunicaciones, procesamiento de imágenes y visión artificial, pruebas y medición, biología computacional, finanzas computacionales, bases de datos... Algunos ejemplos respectivamente son: Neural Network Toolbox , Robust control Toolbox, Signal processing Toolbox, Image Processing Toolbox, Data Acquisition Toolbox, Bioinformatics Toolbox, Financial Toolbox y Database Toolbox.

Además, hoy en día MATLAB dispone unido el programa Simulink, de un entorno de diagrama de bloques utilizado para la simulación multidominio y el diseño basado en modelos, entre otras muchas aplicaciones.

De esta forma, y sin mencionar muchos otros productos MATLAB, se observa que este programa ha crecido de tal forma que su uso no se reduce a la manipulación de matrices, como se originó, sino que a una gran cantidad de campos, no solo matemático-físicos.

Por último, existe mucha información de aprendizaje de MATLAB, y a lo largo de todo el trabajo se ha usado el paquete oficial de la web ([3]).

3.4. Interés del Estudio del Viento

La realización de este trabajo cobra un sentido práctico con el interés que supone estudiar y predecir la dirección y la velocidad del viento.

En primer lugar, este proyecto puede ayudar a la implementación de parques eólicos en lugares adecuados. El hecho de conocer la dirección y velocidad del tiempo a corto plazo en un lugar concreto permite saber si en este lugar sería conveniente o no, instalar una granja eólica. Si se observa una clara estabilidad tanto en la dirección y la velocidad de las corrientes de aire, entonces es una zona idónea para colocar un parque, ya que se puede garantizar que los molinos estarán en funcionamiento la mayor parte del tiempo y por tanto se aumentará la eficiencia de los mismos. Además, se reducirán gastos evitando que los molinos tengan un giro de 360°, lo que resulta más caro de construir y mantener.

El hecho de mejorar la situación de los parques resulta más relevante a día de hoy ya que las energías renovables están cobrando importancia. De facto, en España, en 2015, la energía eólica ha supuesto un 19,4% de toda la demanda energética convirtiéndose en la 3ra tecnología eléctrica ([10]). Por ello, mejorar la ubicación y la eficiencia de estos parques resultará de gran ayuda para el futuro, y predecir la dirección y velocidad del viento ayudará a conseguirlo. Consecuentemente, si se ayuda al rendimiento de los parques eólicos, se favorece al medioambiente, proporcionando la posibilidad de incrementar una fuente de energía renovable y por tanto evitando otras fuentes contaminantes o que dejan residuos.

En segundo lugar, conocer información futura del viento, ayudará a la predicción del tiempo en lo que se refiere a las rachas de viento. De esta manera, se pueden prevenir acontecimientos a distintos niveles. En muchas ocasiones, el hecho de que pueda haber fuertes rachas de viento o que este se muy variable perjudica y pone en riesgo la realización de distintas actividades y eventos que se realizan al aire libre. Por tanto, este programa permitirá prevenir y corregir estas situaciones.

En conclusión, estudiar el viento mediante la realización de un programa que determine la dirección del viento utilizando Cadenas de Markov es de gran interés ya que ayudará a mejorar el rendimiento de los parques eólicos y ayudará a prevenir situaciones de posible riesgo o inadecuadas.

3.5. Obtención de datos de una estación Meteorológica

Para la realización del estudio del viento, en este proyecto se escoge como fuente de información las mediciones registradas por una estación meteorológica concreta para poder validar el modelo y garantizar que este funciona de manera correcta.

La estación meteorológica escogida es la “Estación Meteorológica de Trinidad de Iturgoyen”[13] y se halla ubicada en Navarra, a una altura de 1200 metros, con unas coordenadas de (42°46'34"N, 1°57'15"O).



Figura 1: Estación de Trinidad de Iturgoyen

Se ha escogido esta estación ya que registra velocidades suficientemente elevadas a lo largo de todo el año. Esto quiere decir que las velocidades son las deseadas para que un molino eólico genere electricidad, como se verá posteriormente en el trabajo. De esta manera, si se demuestra una cierta estabilidad en las direcciones del viento en esta estación, se puede tomar como un buen lugar donde colocar molinos eólicos.

Los datos que se estudian, escogidos para el análisis durante todo el documento, son concretamente, desde el 23 de Marzo de 2015 hasta el 9 de Marzo de 2016, con datos en intervalos de 10 minutos.

4. Diseño del Modelo. Creación de un programa con Matlab

Para poder realizar un análisis preciso de los datos obtenidos de la estación meteorológica de Trinidad de Iturgoyen, se crea un programa que trata los datos descargados como se desea, genera la matriz de transición y posteriormente analiza los resultados.

Los datos se descargan desde el Servicio Meteorológico de Navarra, [meteo.navarra](http://meteo.navarra.es) ([13]). Este permite obtener los datos de las direcciones y las velocidades del viento a 10 metros de altura en periodos de cada 10 minutos. Permite seleccionar sólo los parámetros que se desean, que en este caso son los mencionados, y tiene la opción de descargar a fichero Excel. Una vez seleccionados los parámetros indicados, se importa a un archivo Excel y se guarda. Para normalizar el fichero Excel que se utiliza, independientemente de la fuente de importación, se colocan los datos de la siguiente manera. Como ya se ha mencionado, los datos importados y analizados van desde el 23 de Marzo de 2015 hasta el 9 de Marzo de 2016, cada 10 minutos, tal y como se observa a continuación.

Fecha	Dirección viento 10 m (°)	Velocidad viento 10 m (m/s)	Velocidad viento 10 m (km/h)
01/01/2016 0:00	177	3,4	12,3
01/01/2016 0:10	165	3	10,7
01/01/2016 0:20	187	2,3	8,4
01/01/2016 0:30	178	2,9	10,5
01/01/2016 0:40	157	3,6	12,9
01/01/2016 0:50	160	3,8	13,8
01/01/2016 1:00	155	4,1	14,9
01/01/2016 1:10	174	4,6	16,7
01/01/2016 1:20	184	3,8	13,5
.....

Tabla 1: Datos importados a Excel (desde la estación de Iturgoyen)[13]

Una vez se han importado los datos y se guardan en el formato deseado, que será siempre el mismo para que el programa funcione correctamente, se procede a la creación del programa Matlab para analizar estos datos.

Cabe destacar que se han realizado dos programas, uno para predecir la dirección del viento y otro para la velocidad del viento, aunque el procedimiento es muy similar y las características de los mismos también.

4.1. Lectura de Datos

El primer paso para poder realizar el estudio es conseguir leer los datos y poder

manipularlos utilizando MATLAB. Para ello, creamos los ficheros 'leerDatosExcelVelocidad.m' y 'leerDatosExcelDireccion.m' que tienen como objetivo principal leer los datos importados de velocidad y dirección respectivamente y asignarle a cada dato el estado correspondiente. A continuación, se muestra la parte del archivo 'leerDatosExcelVelocidad.m' que lee los datos importados y toma como datos a modificar los de la columna de datos de velocidad del viento.

```
%leer datos excel y modificarlos como yo desee
a = readtable ('UlldeconaElsValentins2015.xls'); %me lee la tabla entera
exportada
b = table2cell (a); %paso a celda, más fácil de trabajar
VViento = b(:,3); %cojo la columna del viento (m/s)
```

Se observa que el comando 'readtable', lee los datos importados a Excel guardándolos en una variable. Además, después de modificar los datos para que sea fácil trabajar con ellos, se guarda una variable que sólo incluye la columna de la velocidad del viento (VViento).

En el archivo 'leerDatosExcelDireccion.m', análogamente al caso de velocidades, se leen los datos de dirección guardándose los datos de la columna de direcciones en grados en una variable (DViento).

4.2. Tratamiento de la Información

Una vez se han leído los datos, se procede a asignar estados a estos. En los mismos ficheros 'leerDatosExcelVelocidad.m' y 'leerDatosExcelDireccion.m' se realiza esta función, como ya se ha mencionado antes. Tanto los datos de velocidad del viento como dirección se dividen en 8 estados. En ambos casos, se observará que la programación empleada es similar.

En primer lugar, se procede a definir los estados tanto para el caso de direcciones como el caso de velocidades.

Para el caso de las **direcciones** se definen los estados utilizando como referencia la rosa de los vientos que se muestra a continuación.

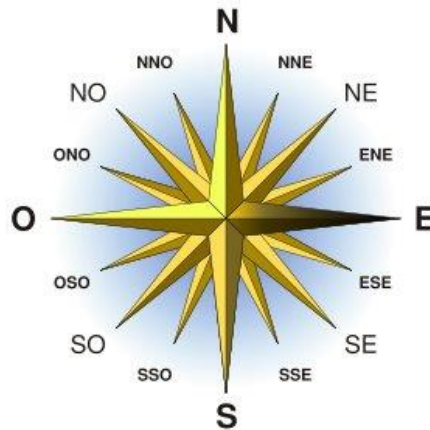


Figura 2: Rosa de los Vientos [12]

Se observa que existen 4 direcciones principales, que son: Norte, Este, Sur y Oeste. A partir de estos 4 estados, se puede subdividir en 8 estados, 16 estados, 36 estados, y así sucesivamente. En la imagen aparece la rosa dividida en 16 estados, donde por ejemplo, ESE se refiere a Este-Sur-Este. En el caso estudiado, se utilizará la subdivisión de 8 estados, ya que está ya es suficientemente precisa para el estudio que se pretende realizar, que incluirá:

- Norte = Estado 1
- Noreste = Estado 2
- Este = Estado 3
- Sureste = Estado 4
- Sur = Estado 5
- Suroeste = Estado 6
- Oeste = Estado 7
- Noroeste = Estado 8

A continuación se muestran los estados definidos asignados a los diferentes grados, escritos e ilustrados con una imagen:

Estado	Grados	Dirección
1	$337,5 < \text{Valor} \leq 22,5$	Norte
2	$22,5 < \text{Valor} \leq 67,5$	Noreste
3	$67,5 < \text{Valor} \leq 112,5$	Este
4	$112,5 < \text{Valor} \leq 157,5$	Sureste
5	$157,5 < \text{Valor} \leq 202,5$	Sur
6	$202,5 < \text{Valor} \leq 247,5$	Suroeste
7	$247,5 < \text{Valor} \leq 292,5$	Oeste
8	$292,5 < \text{Valor} \leq 337,5$	Noroeste

Tabla 2: Estados asignados a las direcciones

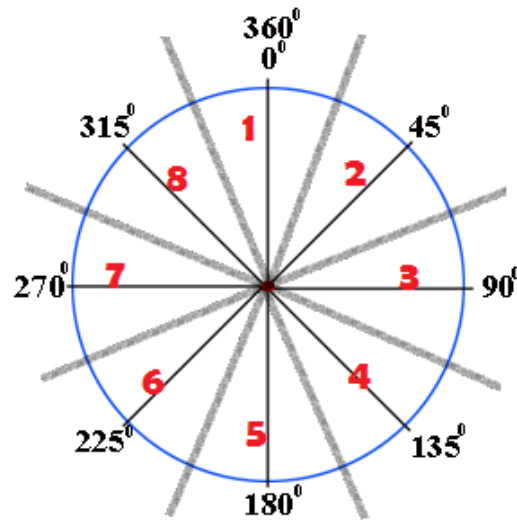


Figura 3: Clasificación de estados del viento [11]

Para el caso de las **velocidades** se toman como referencia las velocidades más habituales en las estaciones meteorológicas en general (y también concretamente en la elegida de Trinidad de Iturgoyen, Navarra) además de la cantidad de energía eléctrica generada en función de la velocidad del viento y de la velocidad de rotación de las aspas del molino. A continuación se muestran 2 gráficos que muestran esta dependencia.

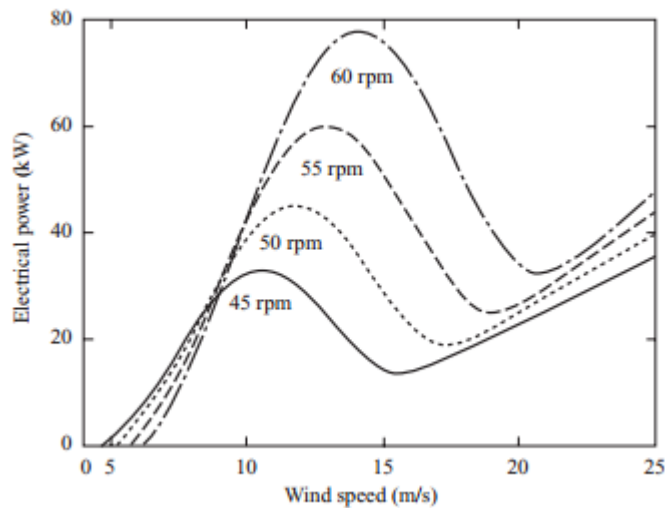


Figura 4: Energía eléctrica generada [2]

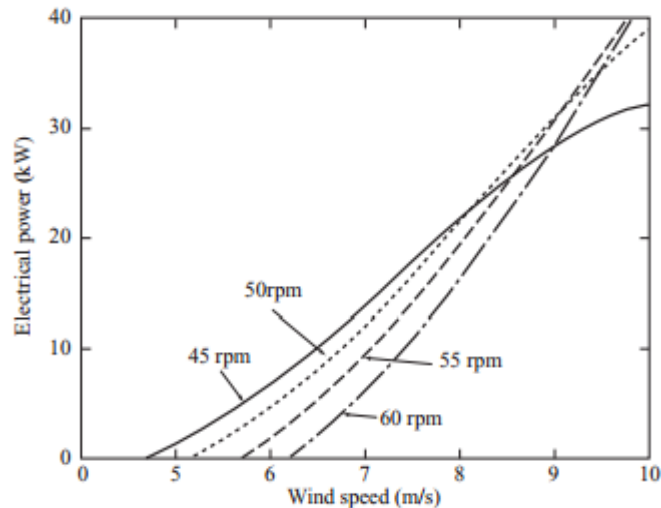


Figura 5: Energía eléctrica generada a bajas velocidades[2]

Se observa que las velocidades a las que el molino eólico empieza a entrar en funcionamiento son a partir de los 4m/s. Se verá en el caso estudiado que la mayoría de velocidades a lo largo del tiempo (a 10 metros de altura obtenidos los datos) superan este valor mínimo. También se observa en los gráficos, que sea cual sea la velocidad de rotación, a las velocidades a las que se genera más energía eléctrica es desde los 7m/s hasta los 18m/s aproximadamente. Por ello, el rango de velocidades incluye todas estas velocidades posibles a las que el molino estaría en funcionamiento. La división de estados se hace desde los 0m/s hasta los valores superiores a 16,66m/s (que equivale a 60km/h). Se definen 8 estados divididos de manera que el primero y el último son de referencia y los 6 del medio avanzan de 10km/h. Por ejemplo, el estado 3 incluye las velocidades de 2,77 m/s a 5,55 m/s, es decir de 10 a 20 km/h.

Los 8 estados definidos asignados a los valores se dividen de la siguiente manera:

Estado	m/s	km/h (aprox)
1	$0 < \text{Valor} \leq 0,0833$	$0 < \text{Valor} \leq 0,3$
2	$0,0833 < \text{Valor} \leq 2,77$	$0,3 < \text{Valor} \leq 10$
3	$2,77 < \text{Valor} \leq 5,55$	$10 < \text{Valor} \leq 20$
4	$5,55 < \text{Valor} \leq 8,33$	$20 < \text{Valor} \leq 30$
5	$8,33 < \text{Valor} \leq 11,11$	$30 < \text{Valor} \leq 40$
6	$11,11 < \text{Valor} \leq 13,88$	$40 < \text{Valor} \leq 50$
7	$13,88 < \text{Valor} \leq 16,66$	$50 < \text{Valor} \leq 60$
8	$16,66 < \text{Valor}$	$60 < \text{Valor}$

Tabla 3: Estados asignados a los datos de velocidad

Asignación de datos de Dirección del viento

En el caso de los datos obtenidos de la Dirección, se importan datos desde los 0 a los 360°. Se muestra la programación empleada en el fichero 'leerDatosExcelDireccion.m' para asignar estados a los distintos grados y se citan los mismos posteriormente.

```
%leer datos excel dirección viento, datos pegados en excel descargado de
meteo.navarra
a = readtable ('TrinidadDeIturgoein2015.xls'); %me lee la tabla entera
exportada
b = table2cell (a); %paso a celda, más fácil de trabajar
DViento = b(:,2); %cojo la columna del viento
vectorDViento = []; %vector Vacio inicial donde ire asignando estados
for i = (1:size(b,1)) %para i igual a las filas de la tabla
    if DViento{i,1}<=22.5 && DViento{i,1}>0 || DViento{i,1}>337.5 &&
DViento{i,1}<=360
        vectorDViento(i) = 1;
    elseif 22.5<DViento{i,1} && DViento{i,1}<=67.5 %condic en grados
        vectorDViento(i) = 2;
    elseif 67.5<DViento{i,1} && DViento{i,1}<=112.5
        vectorDViento(i) = 3;
    elseif 112.5<DViento{i,1} && DViento{i,1}<=157.5
        vectorDViento(i) = 4;
    elseif 157.5<DViento{i,1} && DViento{i,1}<=202.5
        vectorDViento(i) = 5;
    elseif 202.5<DViento{i,1} && DViento{i,1}<=247.5
        vectorDViento(i) = 6;
    elseif 247.5<DViento{i,1} && DViento{i,1}<=292.5
        vectorDViento(i) = 7;
    else
        vectorDViento(i) = 8;
    end
end
vectorDViento;
plot (vectorDViento, '.')
```

Se observa que hay 8 condiciones, cada uno referido a un estado concreto. Cada vez que asigna un estado a un dato, este se va guardando en el vector 'vectorDViento', por lo que una vez se han leído todos los datos, se tiene un vector que contiene enumerados todos los estados de cada valor importado. De esta manera también se muestra en pantalla un gráfico con todos los datos, utilizando la función 'plot (vectorDViento, 'b--o')'.

Asignación de datos de Velocidad del viento

Como en el caso de direcciones del viento, en el fichero 'leerDatosExcelVelocidad.m' se realiza la misma tarea. A continuación se muestra la programación de las 8 condiciones, ya que todo lo demás se mantiene igual que en el caso de direcciones (salvo el nombre de ciertas variables).

```
vectorViento = []; %vector Vacio inicial donde ire asignando estados
for i = (1:size(b,1)) %para i igual a las filas de la tabla
    if VViento{i,1}<=0.0833 %condiciones en m/s para el viento
        vectorViento(i) = 1;
    elseif 0.0833<VViento{i,1} && VViento{i,1}<=2.77
        vectorViento(i) = 2;
    elseif 2.77<VViento{i,1} && VViento{i,1}<=5.55
        vectorViento(i) = 3;
    elseif 5.55<VViento{i,1} && VViento{i,1}<=8.33
        vectorViento(i) = 4;
    elseif 8.33<VViento{i,1} && VViento{i,1}<=11.11
        vectorViento(i) = 5;
    elseif 11.11<VViento{i,1} && VViento{i,1}<=13.88
        vectorViento(i) = 6;
    elseif 13.88<VViento{i,1} && VViento{i,1}<=16.66
        vectorViento(i) = 7;
    else
        vectorViento(i) = 8;
    end
end
end
```

De nuevo en este caso, cada vez que se asigna un estado a un valor, este se va guardando en un vector que en este caso se llama 'vectorViento'. Finalmente, también se muestra en pantalla un gráfico que enseña los datos de velocidad asignados, utilizando el mismo comando 'plot' que en el fichero de direcciones.

A continuación, se muestran los dos gráficos obtenidos para los datos de velocidades del viento y direcciones del viento que muestra Matlab. Cabe destacar, que al haber muchos datos, desde el 23 de Marzo de 2015 hasta el 9 de Marzo de 2016 con información cada 10 minutos, los gráficos carecen de calidad y precisión visual.

* Eje x: número de datos, Eje y: Estados

Gráfico de datos de Dirección

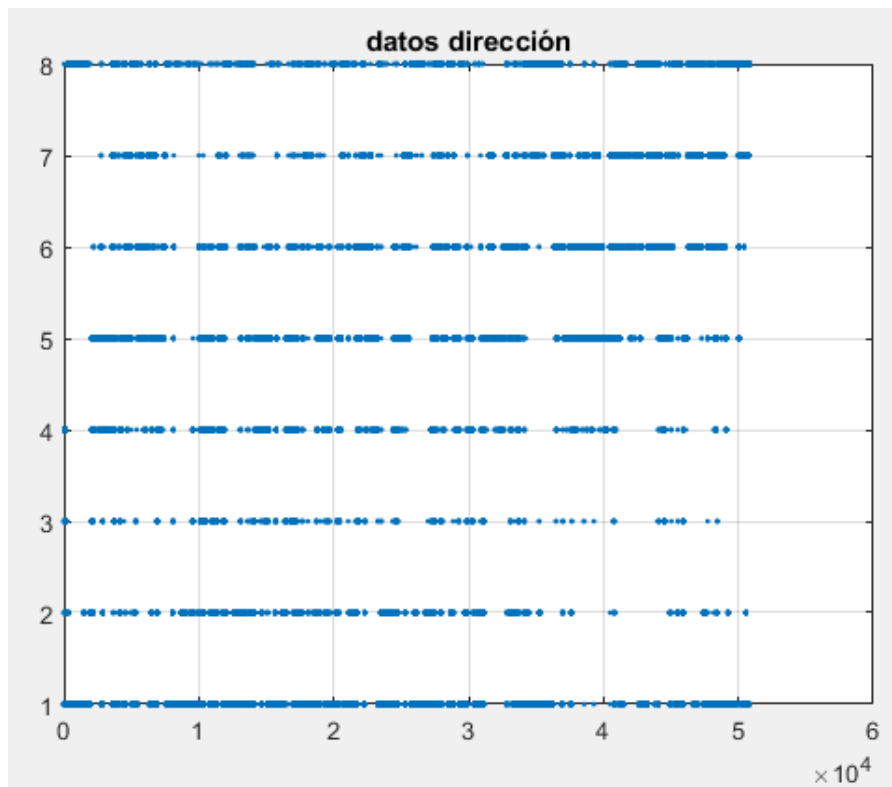


Figura 6: Gráfico de datos de dirección*

Gráfico de datos de Velocidad

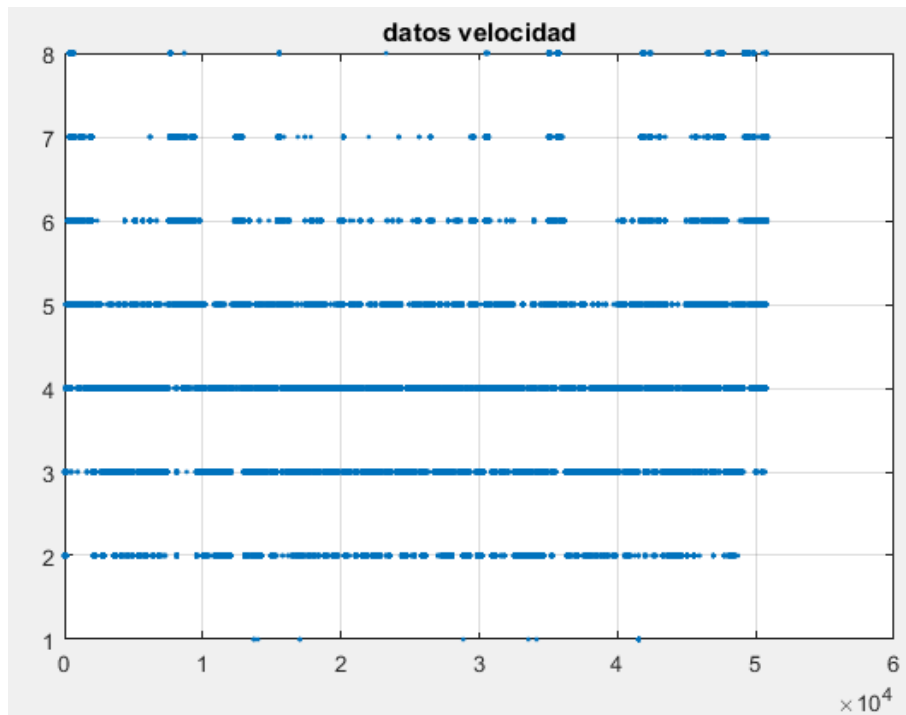


Figura 7: Gráfico de datos de velocidad*

Dado que, como se ha mencionado y se observa en los gráficos, la calidad es muy baja y la gran cantidad de datos impide ver con claridad las situaciones, se ha decidido escoger 1 día aleatorio del año para las velocidades y las direcciones del viento y mostrar los gráficos. A continuación se muestran los mismos, y se observa que evidentemente las tendencias en un día son mucho más constantes en cuanto a estados se refiere.

Gráfico de datos de 1 día de velocidad: 29 de Agosto de 2015

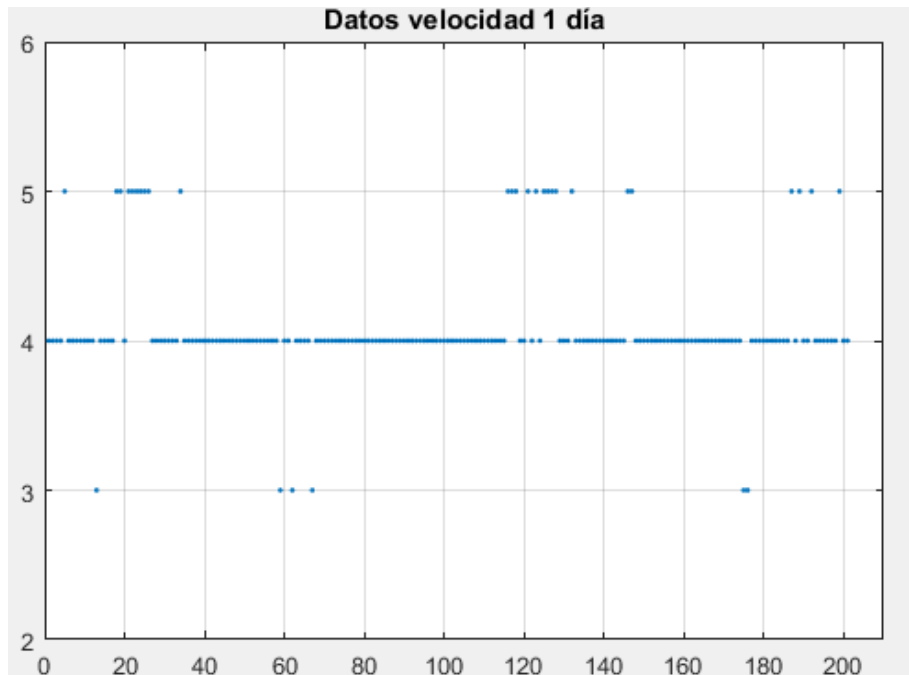


Figura 8: Gráfico de los Datos de Velocidad de 1 día aleatorio*

Gráfico de datos de 1 día de Dirección del viento: 4 de Junio de 2015

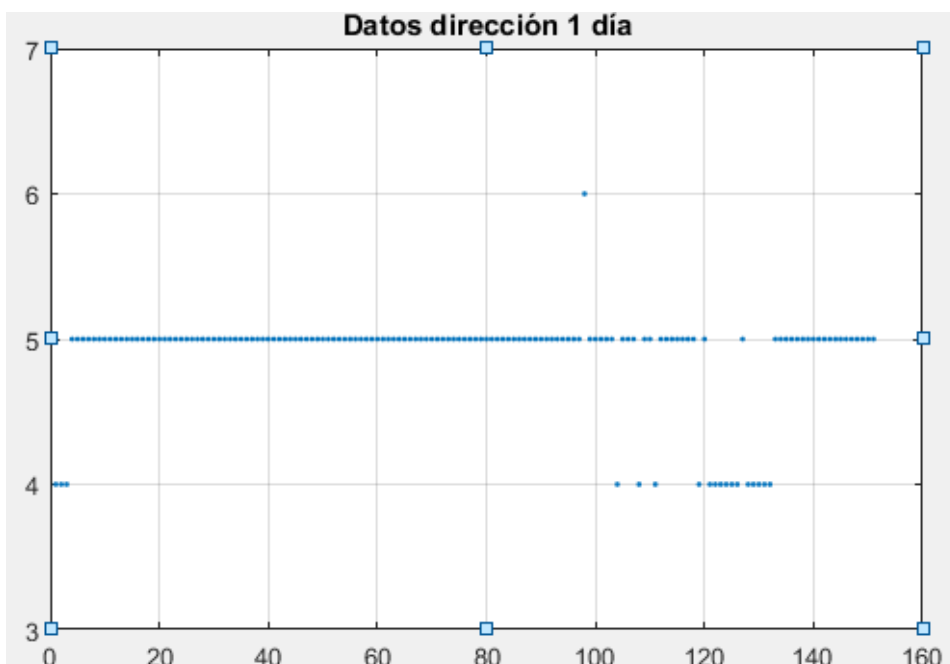


Figura 9: Gráfico de los Datos de Dirección de 1 día aleatorio*

Una vez se han asignado todos los valores a estados, de la forma explicada, y se ha guardado el vector que contiene los mismos en el orden que se asignan, se puede proceder a estudiar los estados para poder obtener la matriz de transición y de probabilidad que luego se analizará.

4.3. Generación de la matriz de Transición

Como ya se ha dicho, con el vector de estados obtenidos, se procede a analizar los mismos para poder obtener una matriz de transición normalizada que permita analizar el modelo. La matriz de transición consiste en una matriz que incluya en cada fila la cantidad de veces que se llega a un estado viniendo del resto de estados. Para normalizar estos valores se divide cada valor obtenido por la suma de toda la columna, de esta manera se consigue tanto la matriz sin normalizar como la matriz normalizada. Esta última tiene la característica de ser una matriz de probabilidad ya que cada una de las columnas sumará 1, como se verá. Se debe conseguir una matriz 8x8 ya que en ambos parámetros estudiados se tienen 8 estados diferentes.

Por tanto:

$$P_{ji} = Pr \{X_{n+1} = j \mid X_n = i\} \text{ para cada } i, j \in \mathbb{E} \text{ y } n \in \mathbb{N}$$

y se obtiene en nuestro caso una matriz de tamaño 8x8 como la siguiente:

$$\begin{pmatrix} p_{11} & p_{12} & p_{13} & p_{14} & p_{15} & p_{16} & p_{17} & p_{18} \\ p_{21} & p_{22} & p_{23} & p_{24} & p_{25} & p_{26} & p_{27} & p_{28} \\ p_{31} & p_{32} & p_{33} & p_{34} & p_{35} & p_{36} & p_{37} & p_{38} \\ p_{41} & p_{42} & p_{43} & p_{44} & p_{45} & p_{46} & p_{47} & p_{48} \\ p_{51} & p_{52} & p_{53} & p_{54} & p_{55} & p_{56} & p_{57} & p_{58} \\ p_{61} & p_{62} & p_{63} & p_{64} & p_{65} & p_{66} & p_{67} & p_{68} \\ p_{71} & p_{72} & p_{73} & p_{74} & p_{75} & p_{76} & p_{77} & p_{78} \\ p_{81} & p_{82} & p_{83} & p_{84} & p_{85} & p_{86} & p_{87} & p_{88} \end{pmatrix}$$

Donde, p_{ij} indica el número de veces que se llega al estado i a partir del estado j dividido por la suma de todos los valores de la columna j . Así por ejemplo, p_{24} indica el número de veces que se llega al estado 2 a partir del estado 4 dividido por la suma de todos los valores de la columna 4.

Para ello, se crean y guardan los ficheros creados 'matrizDetransicionVelocidad.m' y 'matrizDetransicionDireccion.m'. La programación en Matlab empleada para ambos casos ha sido muy similar de nuevo. Se explican ambos programas a continuación.

Fichero 'matrizDeTransicionDirección.m'

En este fichero, se utiliza el vector de estados de direcciones de viento obtenidos y a partir de ello se obtiene la matriz de transición deseada.

```
%generar matriz de transición direcciones de viento
%con este programa se recibe el plot de Datos y además la matriz
normalizada
%de transicion para el caso de direcciones de viento
matrizTransicion = zeros(8,8);
leerDatosExcelDireccion
v = vectorDViento;%vector importado de leerDatosExcelDireccion
```

En primer lugar, como se observa en las líneas de programación anteriores, se crea una matriz 8x8 de ceros y se inicializa el programa creado anteriormente 'leerDatosExcelDireccion.m' para tener a disposición todas las variables del mismo y sobre todo el vector 'vectorDViento' que contiene la lista de estados. Precisamente a este vector se aplican las 64 condiciones que se tienen ya que hay 64 casillas en la matriz de transición. A continuación se observan las primeras condiciones, y el final del programa.

```
for i = (2:size (v,2))
    if v(i-1) == 1 && v(i) == 1 %p11
        matrizTransicion(1,1)= (matrizTransicion(1,1)+1);
    elseif v(i-1) == 2 && v(i) == 1 %p12
        matrizTransicion (1,2) = (matrizTransicion(1,2)+1);
    elseif v(i-1) == 3 && v(i) == 1 %p13
        matrizTransicion (1,3) = (matrizTransicion(1,3)+1);
    elseif v(i-1) == 4 && v(i) == 1 %p14
        matrizTransicion (1,4) = (matrizTransicion(1,4)+1);
    elseif v(i-1) == 5 && v(i) == 1 %p15
        matrizTransicion (1,5) = (matrizTransicion(1,5)+1);
    elseif v(i-1) == 6 && v(i) == 1 %p16
        matrizTransicion (1,6) = (matrizTransicion(1,6)+1);
    elseif v(i-1) == 7 && v(i) == 1 %p17
        matrizTransicion (1,7) = (matrizTransicion(1,7)+1);
    elseif v(i-1) == 8 && v(i) == 1 %p18
        matrizTransicion (1,8) = (matrizTransicion(1,8)+1);
        %buscare otra forma de hacerlo más corto, así debo hacer 64
        condiciones, que son los estados de la matriz
        %primera fila end
    .....
    .....
    .....

%He obtenido la matriz con los valores exactos, ahora normalizo estos
valores para tener una matriz de probabilidad donde las columnas sumen 1
sumaColumnas = sum(matrizTransicion);
mTransNormalDire = bsxfun (@rdivide, matrizTransicion,
sum(matrizTransicion)); %me divide cada valor por la suma de toda la
columna
mTransNormalDire(isnan(mTransNormalDire))=0;%paso los valores de la
matriz divididos por 0 y con valor NaN a 0
```

Después de obtener la matriz 'matrizTransicion' que contiene las cantidades sin normalizar que se mostrará más adelante, se procede a normalizarla dividiendo cada valor de la matriz por la suma de la columna. Esto se consigue con las últimas líneas de programación, usando la función 'bsxfun' para obtener la matriz normalizada. Al inicializar el fichero 'leerDatosExcelDirección.m' al inicio del este, todas las funciones del programa importan.

Por tanto, con el fichero 'matrizDeTransicionDireccion.m' se obtienen para el posterior uso las siguientes matrices:

Matriz de transición sin normalizar, con cantidades reales:

14542	382	8	5	5	14	29	958
435	2816	66	9	3	4	2	9
7	107	539	78	13	6	2	2
3	6	115	1781	475	21	2	1
5	6	20	506	10764	594	6	1
10	11	5	21	626	5078	383	24
15	4	1	3	9	413	3403	232
927	12	0	1	7	28	253	5018

Matriz de transición normalizada, para el posterior estudio:

0,91206724	0,11423445	0,01061008	0,00207987	0,0004201	0,00227347	0,00710784	0,15340272
0,02728299	0,84210526	0,08753316	0,00374376	0,00025206	0,00064956	0,0004902	0,00144115
0,00043904	0,03199761	0,71485411	0,03244592	0,00109225	0,00097434	0,0004902	0,00032026
0,00018816	0,00179426	0,15251989	0,74084859	0,03990926	0,0034102	0,0004902	0,00016013
0,0003136	0,00179426	0,0265252	0,21048253	0,90438582	0,09645989	0,00147059	0,00016013
0,0006272	0,00328947	0,0066313	0,00873544	0,0525962	0,82461838	0,09387255	0,00384307
0,00094079	0,00119617	0,00132626	0,00124792	0,00075618	0,06706723	0,83406863	0,03714972
0,05814099	0,00358852	0	0,00041597	0,00058814	0,00454693	0,0620098	0,80352282

Fichero 'matrizDeTransicionVelocidad.m'

Igual que el caso de Direcciones de viento, en este caso se utiliza la misma programación para obtener las matrices de transición para las Velocidades del viento. Por ello, los programas completos se anexan al final y a continuación se muestran las dos matrices que se usarán para el posterior análisis.

Matriz de transición sin normalizar, con cantidades reales:

16	8	0	0	0	0	0	0
7	3990	825	3	0	0	0	0
0	820	11869	1426	11	0	0	0
1	6	1424	13074	1196	4	1	0
0	0	9	1196	7748	724	5	1
0	1	0	6	719	3101	334	3
0	0	0	1	9	328	1012	152
0	0	0	0	0	7	149	645

Matriz de transición normalizada, para el posterior estudio:

0,66666667	0,00165803	0	0	0	0	0	0
0,29166667	0,82694301	0,05839881	0,00019101	0	0	0	0
0	0,16994819	0,84016422	0,09079333	0,00113601	0	0	0
0,04166667	0,00124352	0,10079989	0,83242073	0,12351544	0,00096061	0,00066622	0
0	0	0,00063708	0,07614924	0,80016524	0,17387128	0,00333111	0,00124844
0	0,00020725	0	0,00038202	0,07425385	0,74471662	0,22251832	0,00374532
0	0	0	6,37E-05	0,00092946	0,07877041	0,67421719	0,1897628
0	0	0	0	0	0,00168108	0,09926716	0,80524345

4.4. Comprobación de Datos Markovianos

Una vez se han obtenido los datos, se han asignado los estados y se ha generado la matriz de transición, el siguiente paso es comprobar que los datos estudiados en la estación meteorológica se pueden modelar como una Cadena de Markov.

Para ello, se debe comprobar el cumplimiento de la siguiente igualdad relacionada con las propiedades de Markov ([7]).

$$Pr \{X_2 = i_2 | X_1 = i_1\} = Pr \{X_2 = i_2 | X_0 = i_0 ; X_1 = i_1\}$$

De esta forma, se compara la probabilidad que un estado dado X_2 , se dé después de que se haya dado otro estado X_1 , con la probabilidad de que se de este estado X_2 después de que se haya dado el mismo estado X_1 a la vez que este se haya dado después de otro estado X_0 .

A continuación se muestran las fórmulas empleadas para calcular ambas probabilidades:

1. $Pr \{X_2 = i_2 | X_1 = i_1\} =$

$$\frac{\text{Numero de veces que se pasa del estado } X_1 = i_1 \text{ al estado } X_2 = i_2}{\text{Número de veces que el estado sale de } X_1 = i_1}$$


```

while i<=n
    j = 1;
    vectorFila = [];%fila vacia
    x = 1;
    cont = 1;
    while j<=length(l)
        valor = (length(l)^(n-i))*cont;
        while x<=valor
            vectorFila(x) = l(j);
            x = x+1;
        end
        j = j+1;
        if length(vectorFila)~=length(l)^(n) && j == (length(l)+1)
            j=1;
        end
        cont = cont + 1;
    end
    M(i,:) = vectorFila;%añado fila a la matriz 3x512
    i = i+1;
end

```

Se puede observar que la matriz explicada se guarda en una matriz llamada 'M', que en cada iteración va añadiendo una condición nueva. El valor 'n' indica el orden al que se eleva el número de estados, es decir en este caso es 8^3 , y la variable 'i' contiene los 8 estados existentes. Por último, la variable 'valor', indica el número de veces que se repite un valor por fila, lo que quiere decir que para generar todas las combinaciones, en la primera fila hay 64 de estado 1, 64 de estado 2, en la segunda fila hay 8 de estado 1, 8 de estado 2 y así sucesivamente (esto se puede ver claramente en el trozo de matriz enseñado anteriormente). Cuando se llega al estado 8, el valor de 'j' se reinicia a 1 y vuelve a empezar. De esta forma, y explicado con mucha brevedad, se consigue la matriz deseada.

4.4.2. Fichero 'EsMarkovianoVelocidad.m'

Este fichero contiene un programa que analiza los datos de velocidades del viento para obtener las dos matrices de probabilidad y graficarlas. Concretamente, se conseguirá una matriz 8x8, que incluirá todas las probabilidades de la condición 1 ($Pr\{X_2 = i_2 | X_1 = i_1\}$) y otra matriz de tamaño 64x3 que contiene todas las probabilidades de la condición 2, ($Pr\{X_2 = i_2 | X_0 = i_0; X_1 = i_1\}$). Una vez conseguidas las matrices, se grafican y se determina si se pueden validar los datos como una cadena Markov. A continuación se muestra y se explica el fichero.

```

%comprobación de carácter markoviano
matrizDeTransicionVelocidad
saleDe1 = size(find(vectorViento == 1),2);
saleDe2 = size(find(vectorViento == 2),2);
saleDe3 = size(find(vectorViento == 3),2);
saleDe4 = size(find(vectorViento == 4),2);
saleDe5 = size(find(vectorViento == 5),2);

```

```

saleDe6 = size(find(vectorViento == 6),2);
saleDe7 = size(find(vectorViento == 7),2);
saleDe8 = size(find(vectorViento == 8),2);
MatrizProbabilidad1 = zeros(8,8);
MatrizProbabilidad1(:,1) = matrizTransicionV(:,1)/saleDe1;
MatrizProbabilidad1(:,2) = matrizTransicionV(:,2)/saleDe2;
MatrizProbabilidad1(:,3) = matrizTransicionV(:,3)/saleDe3;
MatrizProbabilidad1(:,4) = matrizTransicionV(:,4)/saleDe4;
MatrizProbabilidad1(:,5) = matrizTransicionV(:,5)/saleDe5;
MatrizProbabilidad1(:,6) = matrizTransicionV(:,6)/saleDe6;
MatrizProbabilidad1(:,7) = matrizTransicionV(:,7)/saleDe7;
MatrizProbabilidad1(:,8) = matrizTransicionV(:,8)/saleDe8;
MatrizProbabilidad1(isnan(MatrizProbabilidad1))=0;
%Ya tengo una matriz con las probabilidades de pasar de un estado X a un
estado Y dividido por las veces que se sale de X
MatrizProbabilidad1Grafico = reshape (MatrizProbabilidad1, 64,1);
%Me he generado la columna deseada para hacer el gráfico

condicionesProbabilidad
%hago una búsqueda de todas las ocurrencias/condiciones
i=1;
VectorProbabilidad2V = [];
MatrizCantidades2V = [];
while i<=size(M,2)
    buscado = M(:,i)';
    numeroRepeticiones = length(strfind(vectorViento, buscado));
    MatrizCantidades2V(i) = numeroRepeticiones;
    VectorProbabilidad2V(i)=
numeroRepeticiones/matrizTransicionV(buscado(2),buscado(1));
    i = i+1;
end
MatrizCantidades2V = reshape (MatrizCantidades2V,64,8);
VectorProbabilidad2V(isnan(VectorProbabilidad2V))=0;
matrizFinalPr2V = reshape (VectorProbabilidad2V,64,8);
%al final consigo una matriz 64x8 donde se se tienen las 512
probabilidades
%calculadas y divididas, 111,112,113.....121,122 (columna 1),
%211,212,...221 (columna 2).....

%Finalmente dibujo el gráfico que enseña si se puede modelar como una
%cadena de markov
plot (MatrizProbabilidad1Grafico, matrizFinalPr2V, '.', 'markersize',
12)
title ('Comprobación de datos Markovianos para las velocidades')
grid on;
refline(1,0)

```

Se puede observar que hay tres partes diferenciadas en el programa. La primera parte es la que genera la matriz 8x8 con las probabilidades de la condición 1. Se aprecia que para obtener esta matriz se utiliza la matriz de transición obtenida en el fichero 'matrizDeTransicionVelocidad.m'. Básicamente, se utiliza la matriz de transición sin normalizar obtenida en ese fichero y se divide cada término por el número de veces que sale del estado primero. Por ejemplo, todos los valores de la columna 1 se dividen por el número de veces que se sale del estado 1, y así con todas las columnas. Para ello primero se busca el número de veces que aparece cada estado en el vector de estados

'vectorViento' con la función 'size(find(vectorViento == estado),2)'. Una vez se tienen estos valores sólo se debe añadir a 'MatrizProbabilidad1' todos los valores deseados en cada fila y columna, que con mucho sentido es igual que la matriz de transición normalizada obtenida con anterioridad (ya que el número de veces que se sale de 1 es lo mismo que la suma de todos los valores de la columna 1 por ejemplo). Aun así, se realiza la programación para asegurarse que la matriz de igual. Por último, en esta parte del programa se realiza un reajuste del tamaño de la matriz 8x8, convirtiéndola en una 64x1. Esta matriz se guarda en la variable 'MatrizProbabilidad1Grafico' y se cambia el tamaño de la matriz utilizando la función 'reshape', como se puede ver. Este reajuste se realiza sólo para facilitar la posterior creación de los gráficos. No es necesario enseñar las matrices de esta condición ya que como se ha dicho son iguales que las matrices de transición. En el anexo [A] se adjuntan tanto la matriz de probabilidad 1 de dirección como de velocidad.

La segunda parte del programa empieza a partir del llamamiento del fichero 'condicionesProbabilidad.m' explicado anteriormente. Se tiene como objetivo obtener la matriz de probabilidad de la condición 2. En este caso se realiza un programa que recorre la matriz 'M' obtenida anteriormente en el fichero 'condicionesProbabilidad.m'. Como ya se ha explicado, esta matriz 'M' contiene todas las combinaciones posibles, por lo que en cada iteración seguida (cada vez que se entra al 'while'), se va cambiando de valores, añadiendo la nueva probabilidad a un vector creado, 'VectorProbabilidad2V'. De esta forma se guardan en este vector todas las probabilidades $Pr\{X_2 = i_2 | X_0 = i_0; X_1 = i_1\}$, como se observa en la programación mostrada. Una vez obtenido este vector, se reajusta el tamaño del mismo con la función 'reshape', obteniendo una matriz de tamaño 64x8, que facilitará la creación del gráfico. Finalmente, cabe destacar que todos los valores divididos por 0 se convierten a 0 con la función 'isnan'.

A continuación se muestra una parte de la matriz 64x8, 'matrizFinalPr2V' obtenida junto con una tabla que indica la combinación de estados seguida en cada casilla.

0,66666667	0,875	0,25	0	0	0	0	0	0
0,29166667	0,0625	0,75	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0,04166667	0,0625	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0,00165803	0	0,00200501	0	0	0	0	0	0
0,82694301	1	0,8679198	0,62909091	0,33333333	0	0	0	0
0,16994819	0	0,12957393	0,36484848	0,66666667	0	0	0	0
0,00124352	0	0,00050125	0,00484848	0	0	0	0	0
.....

111	211	311	411	511	611	711	811
112	212	312	412	512	612	712	812
113	213	313	413	513	613	713	813
114	214	314	414	514	614	714	814
115	215	315	415	515	615	715	815
116	216	316	416	516	616	716	816
117	217	317	417	517	617	717	817
118	218	318	418	518	618	718	818
121	221	321	421	521	621	721	821
.....

Finalmente, hay una tercera parte del programa que básicamente se ocupa de realizar el gráfico que es el que determina si los datos son modelables como una cadena de Markov. Esto se realiza con un simple 'plot' que contiene en el eje x las probabilidades de la condición 1 y en el eje y las probabilidades de la condición 2. Junto con esto, se grafica una bisectriz que se dibuja con el comando 'refline', y que sirve para comparar ambas condiciones. Más adelante se verán los gráficos.

4.4.3. Fichero 'EsMarkovianoDireccion.m'

El fichero 'EsMarkovianoDireccion.m' es completamente análogo al de velocidades pero utilizando los datos de direcciones del viento, por lo que no es necesario reexplicarlo, ya que las variaciones son mínimas. El programa de este fichero se adjunta en los anexos[B] del trabajo, al final del mismo.

A continuación se muestra, al igual que en las velocidades, parte de la matriz 'matrizFinalPr2D' obtenida.

0,912067	0,926488	0,70680	0,625	0,4	0,4	0,42857	0,55172	0,80062
0,027282	0,021523	0,25916	0,25	0	0,6	0,28571	0,20689	0,00835
0,000439	0,000275	0	0	0	0	0	0	0,00313
0,0001881	6,88E-05	0	0	0,2	0	0	0,034482	0
0,0003135	0,0002062	0,0026178	0	0	0	0	0	0,00104
0,0006271	0,0003438	0,0052356	0	0	0	0,07142857	0	0,00208
0,000940	0,0006876	0,0052356	0	0,2	0	0	0,034482	0,00104
0,0581409	0,0504057	0,02094241	0,125	0,2	0	0,21428571	0,172413	0,183716
0,114234	0,3011494	0,08487216	0,09090909	0,33333	0	0,5	0,5	0
0,842105	0,655172	0,87571	0,75757	0,66666	0,333333	0,25	0	0,77777
.....

4.4.4. Gráficos de comprobación

Como se ha mencionado anteriormente, hay una tercera parte del programa que tiene como objetivo graficar las dos probabilidades, devolviéndolas por pantalla, para poder ver si los datos son válidos para ser modelados como una cadena de Markov.

Para ello, se realizan los dos gráficos, en los que el eje x está compuesto por las probabilidades de la condición 1 ($Pr\{X_2 = i_2 | X_1 = i_1\}$) y el eje y con las probabilidades de la condición 2 ($Pr\{X_2 = i_2 | X_0 = i_0 ; X_1 = i_1\}$). Para que los datos sean modelables como una cadena de Markov, se ha de cumplir, aproximadamente, que los datos coincidan. Esto quiere decir que si se cumple la relación explicada anteriormente, la nube de puntos en el gráfico debe estar alrededor de la bisectriz (recta $x = y$). A continuación se muestran las dos gráficas obtenidas, cada una con su fichero respectivamente.

Gráfico de Velocidades obtenida en 'EsMarkovianoVelocidad.m'

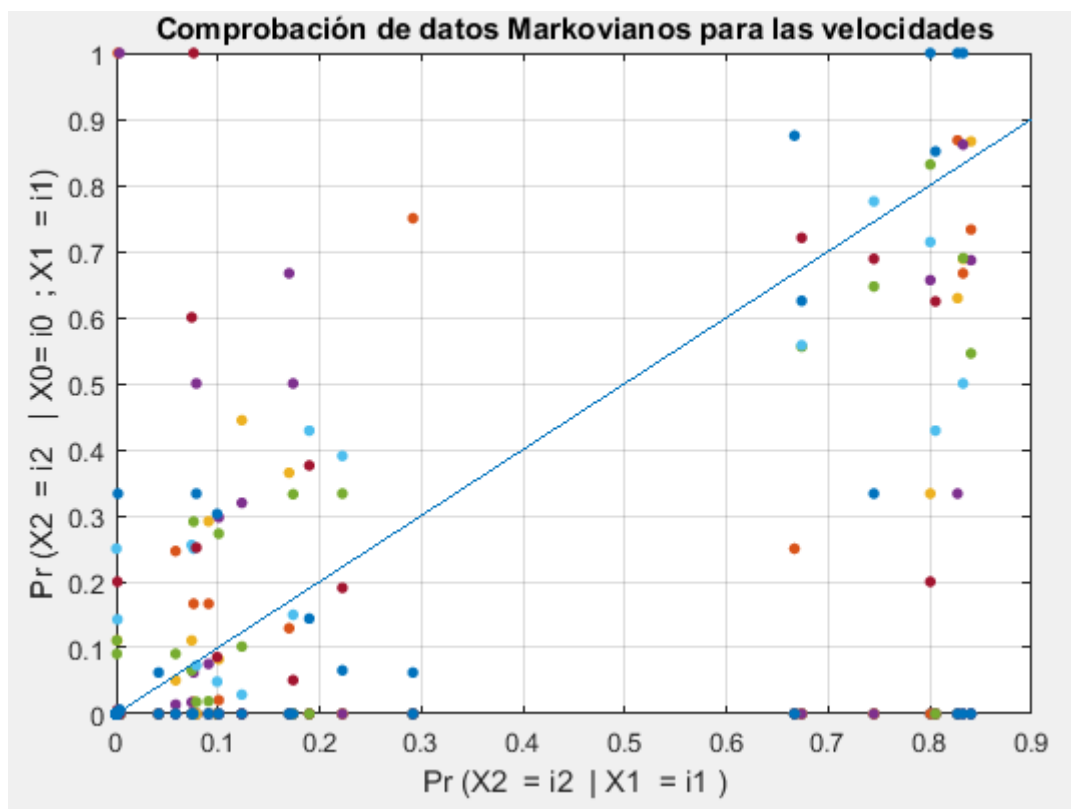


Figura 10: Comprobación de datos markovianos para las velocidades del viento

Se puede observar que la nube de puntos no forma estrictamente una línea recta en la bisectriz, ya que las probabilidades no coinciden con exactitud. De todas formas, la mayoría de puntos observados en el gráfico se encuentran alrededor de la bisectriz. Por esto, se considera que los datos escogidos de Marzo de 2015 a Marzo de 2016 de la estación de Iturgoyen, Navarra se pueden modelar como una Cadena de Markov. de los

casos de probabilidades:

Gráfico de Direcciones 'EsMarkovianoDireccion.m'

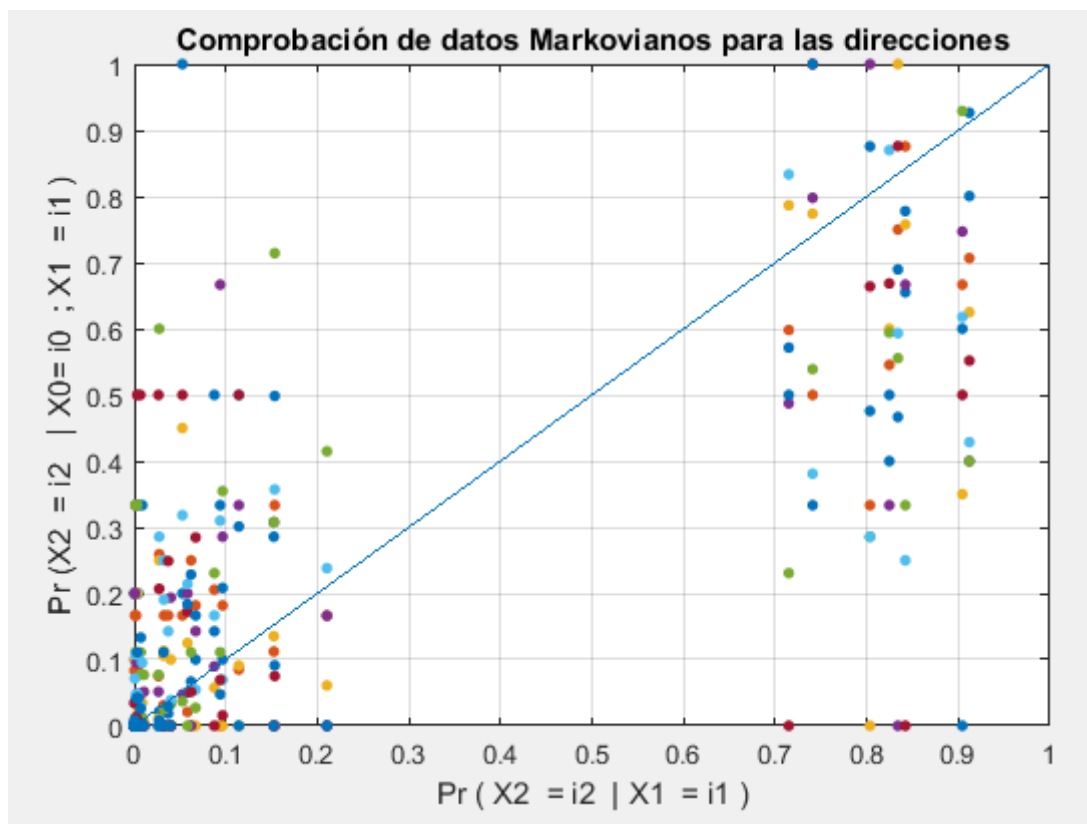


Figura 11: Comprobación de datos markovianos para las direcciones del viento

En el caso de direcciones, como se puede observar, la nube de puntos no está sobre la bisectriz. De todas formas, como en el caso de velocidades, los puntos observados sí que se acercan lo suficiente a la bisectriz como para poder considerar los datos estudiados modelables como una cadena de Markov.

4.4.5. Gráficos extra para la comprobación del carácter Markoviano

Con las gráficas anteriores ya se ve en gran medida que tanto los datos de velocidad como de dirección de corrientes de aire se puede modelar como una cadena de Markov. De todas formas, para corroborar aún más que los datos obtenidos de la estación meteorológica de trinidad de Iturgoyen son modelables como una cadena de Markov, se muestran a continuación 2 gráficos hechos al margen del programa(que ya es suficiente), con Excel, que certifican la validez de los datos.

Gráfico extra de velocidad

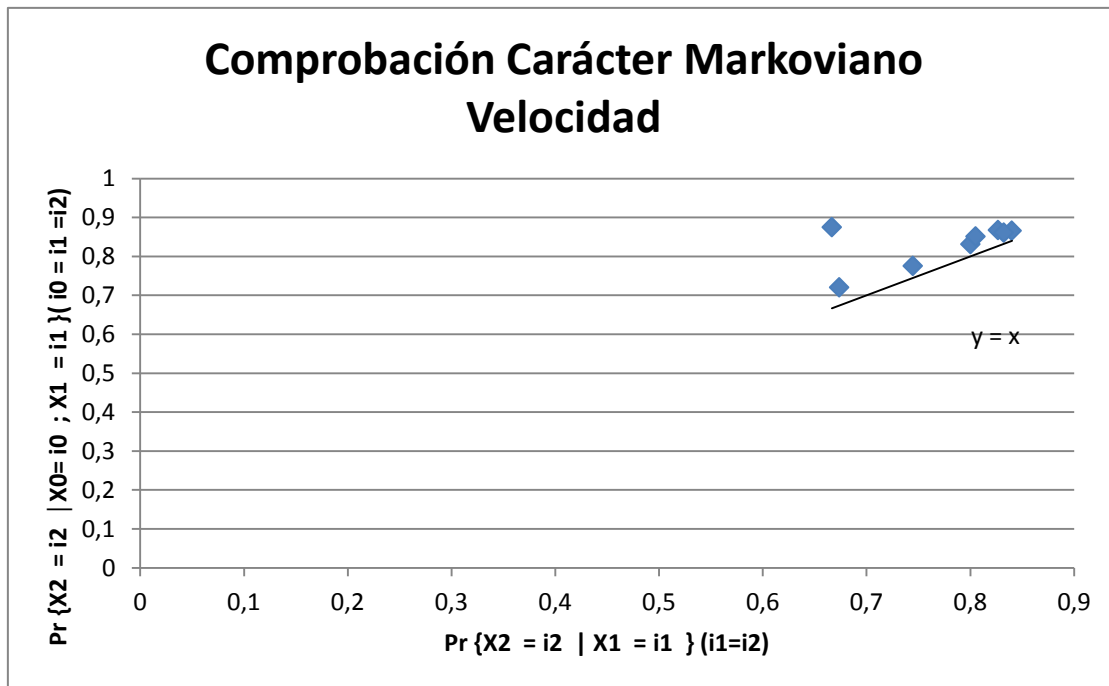


Figura 12: Caso de 1 probabilidad velocidades

En este gráfico se puede observar los 8 casos en los que los estados de probabilidad son iguales. Por ejemplo, se compara la probabilidad de que se vaya del estado 3 al estado 3, con la probabilidad de que se vaya del estado 3, al estado 3, y de este último al 3 también. Se observa, que los valores se acercan mucho a la bisectriz, que sería el caso ideal. Además, se han puesto estos casos ya que son los más repetidos a lo largo de la serie de datos.

Gráfico extra de direcciones

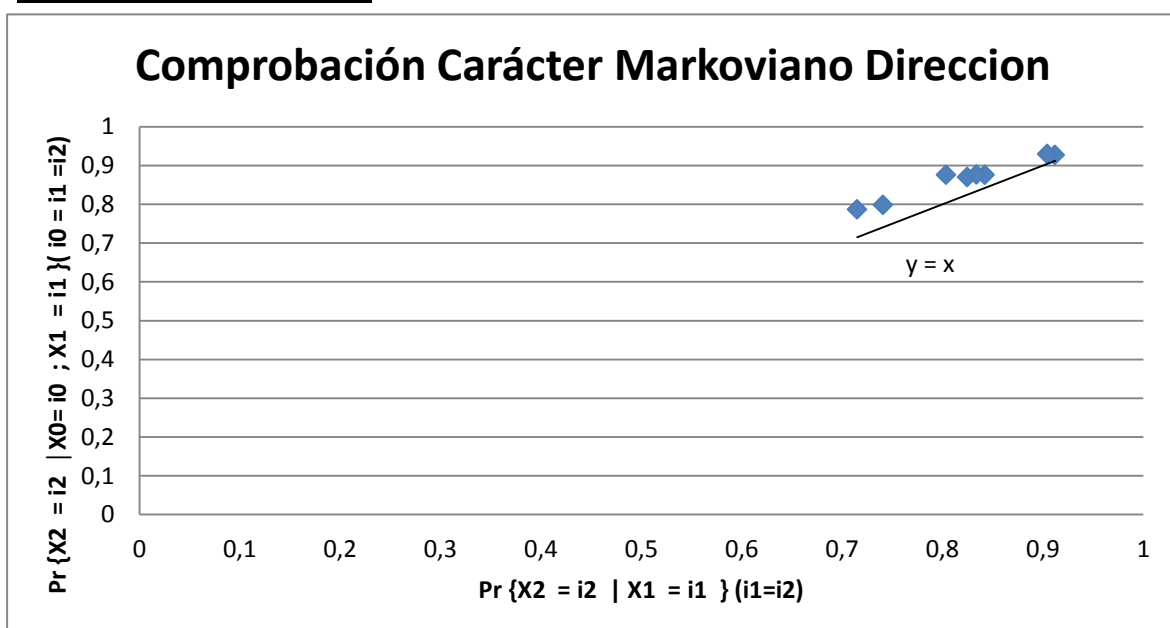


Figura 13: Caso de 1 probabilidad direcciones

Este gráfico muestra lo mismo que la figura 12 pero en el caso de direcciones del viento. Se ve claramente que las probabilidades son prácticamente iguales ya que se acercan casi a la perfección a la bisectriz dibujada, por lo que se reafirma el hecho de que los datos de dirección se ajustan a los deseados para analizarlos como una Cadena de Markov.

En resumen, se considera que los datos escogidos para el estudio se pueden modelar como una cadena de Markov, ya que como se demuestra, aproximadamente, con los gráficos que al menos hasta dos estados anteriores no importa demasiado el camino seguido hasta llegar al estado actual.

5. Estudio mediante Cadenas de Markov y Predicción

Como se ha visto con anterioridad, se han obtenido las dos matrices de transición normalizadas y se ha corroborado que los datos se pueden modelar como una cadena de Markov, tanto para el caso de velocidades como de direcciones del viento. Por tanto, se puede proceder al análisis de estas matrices para poder predecir el estado del viento en el futuro para el caso concreto de la estación meteorológica de Iturgoyen. Para ello, se seguirán los siguientes pasos:

- Cálculo de VAPS y VEPS de matrices de transición normalizadas
- Búsqueda de distribución estacionaria
- Análisis de resultados
- Ejemplos específicos

5.1. Cálculo de Valores propios y Vectores propios

En primer lugar, se crean dos ficheros nuevos para realizar este nuevo cálculo con Matlab. Estos ficheros se llaman 'PrediccionDireccionViento.m' y 'PrediccionVelocidadViento.m'. La idea de estos ficheros es calcular los VAPS, los VEPS, la matriz D diagonal y la matriz S de las matrices de transición normalizadas. Además, los ficheros normalizan todos los VEPS. A continuación se explican los dos ficheros.

Fichero 'PrediccionDireccionViento.m'

En este fichero se programa de manera sencilla el cálculo de VAPS y VEPS de la matriz de transición normalizada para las direcciones del viento. Se muestra la programación empleada seguidamente.

```
%calculo de Vaps y Veps, prediccion de estado de direccion viento
matrizDeTransicionDireccion;
%calculo los valores propios y los vectores propios de la matriz de
transicion normalizada
VapsDireccion = eig (mTransNormalDire); %vector columna de vaps
[VepsDireccion, MatrizDiagonalVapsD] = eig (mTransNormalDire); %vectores
propios en columnas de valores propios

VepsDireccionNormalizado = bsxfun (@rdivide, VepsDireccion,
sum(VepsDireccion)); %me divide cada valor por la suma de toda la
columna
VepsDireccionNormalizado(isnan(VepsDireccionNormalizado))=0;%paso los
valores de la matriz divididos por 0 y con valor NaN a 0
VepsDireccionNormalizado(abs(VepsDireccionNormalizado)>1)= 0; %Me cambia
los valores infinitos a 0

syms x
polCaracXD = sym(charpoly(mTransNormalDire, x));
coefPolCaractD = poly (mTransNormalDire);
```

Se puede observar que en primer lugar el programa importa todo el fichero 'matrizDeTransicionDireccion.m'. De esta manera se pueden utilizar todas las variables y datos guardados en ese fichero. En este caso, la matriz necesitada es la matriz de transición normalizada con las probabilidades de direcciones del viento, llamada 'mTransNormalDire'.

Una vez se tiene la matriz necesaria, simplemente se calcula con el uso de la función 'eig' de Matlab los valores de los VAPS, la matriz diagonal D y la matriz S con los VEPS, guardados en las variables 'VapsDireccion', 'MatrizDiagonalVapsD' y 'VepsDireccion', respectivamente. Lo siguiente que se ve en el programa es la normalización de la matriz de los VEPS, dividiendo por la suma de cada una de las columnas cada valor. Además, con el uso del programa adicional a Matlab, Symbolic toolbox, se consigue mostrar, en función de x, el polinomio característico de la matriz de transición llamado 'polCaracXD', usando 'sym(charpoly())', tal y como se ve. Finalmente también se obtiene un vector columna con los coeficientes del polinomio característico, guardados como 'coefPolCaractD' en la última línea del programa. A continuación se muestran todos los resultados obtenidos.

VAPS de la matriz de transición de dirección ('VapsDireccion'):

1
0,97766307
0,63569891
0,8957267
0,84188432
0,77788762
0,71597739
0,73163283

Matriz Diagonal D para direcciones del viento (VAPS en la diagonal, 'MatrizDiagonalVapsD'):

1**	0	0	0	0	0	0	0
0	0,97766307	0	0	0	0	0	0
0	0	0,63569891	0	0	0	0	0
0	0	0	0,8957267	0	0	0	0
0	0	0	0	0,84188432	0	0	0
0	0	0	0	0	0,77788762	0	0
0	0	0	0	0	0	0,71597739	0
0	0	0	0	0	0	0	0,73163283

Matriz S para direcciones, con los VEPS en columnas ('VepsDireccion')

0,707909517	0,691835784	0,072403311	0,304071439	-0,494314819	0,496077088	-0,520172609	0,580160513
0,14848304	0,13683836	-0,140092434	0,285703411	0,572641113	0,027798761	0,084120138	-0,22567993
0,033486578	0,001403549	0,337705067	0,081658787	0,152865232	-0,155552279	0,034047534	0,100304903
0,106788447	-0,105735607	-0,701616879	0,179992283	0,050390333	-0,350174383	-0,082650056	0,280727338
0,528727884	-0,621259868	0,577802905	0,418833532	-0,495986566	0,22790977	-0,110930864	-0,454554409
0,27357587	-0,245819821	-0,169904935	-0,449588236	0,156390284	0,511546501	0,39304789	0,176005943
0,181279098	-0,060688576	0,065777882	-0,600791653	0,283648743	-0,325368467	-0,412078305	0,072854364
0,277547719	0,20342618	-0,042074917	-0,219879563	-0,22563432	-0,432236991	0,614616273	-0,52981872

***Nótese que el VEP en la columna 1 de la matriz va asociado al VAP de la columna 1 de la Matriz diagonal, y así sucesivamente con el resto de VAPS y VEPS.*

Matriz de VEPS, S, normalizada:

0,31353977	0	0	0	0	0	0	0
0,06576453	0	0	0	0	0	0	0
0,01483152	0	0	0	0	0	0	0
0,04729761	0	0	0	0	0	0	0
0,23417854	0	0	0	0	0	0	0
0,12116932	0	0	0	0	0	0	0
0,08029021	0	0	0	0	0	0	0
0,12292849	0	0	0	0	0	0	0

En esta matriz se puede observar que salvo la primera columna, todos los valores son iguales a 0. Esto se debe a que la suma de todas las columnas es igual a 0, exceptuando la primera columna. Por ello, al realizar la división por 0, el programa da error, y consecuentemente se han igualado estos valores a 0. La primera columna, coincide además que es el VEP asociado al VAP de valor 1, tal y como se ve en la matriz diagonal anterior. Esta columna es la que interesa estudiar, por lo que ya son resultados deseados.

Polinomio característico en función de λ (el programa lo da en función de x pero se ha cambiado a λ para utilizar las mismas notaciones que en el capítulo 3):

$$Q_D(\lambda) = \lambda^8 - 6,57\lambda^7 + 18,86\lambda^6 - 30,82\lambda^5 + 31,37\lambda^4 - 20,37\lambda^3 + 8,24\lambda^2 - 1,9\lambda + 0,19$$

Fichero 'PrediccionVelocidadViento.m'

De manera equivalente al caso de direcciones del viento, en este caso lo único que cambia en la programación empleada es la importación del fichero 'matrizDeTransicionViento.m' y la matriz de transición normalizada utilizada, como se observa a continuación.

```
%calculo de Vaps y Veps, prediccion de estado del viento
matrizDeTransicionVelocidad;
%calculo los valores propios y los vectores propios de la matriz de
```



```

transicion normalizada
VapsVelocidad = eig (mTransNormalVeloc); %vector columna de vaps
[VepsVelocidad, MatrizDiagonalVapsV]= eig (mTransNormalVeloc); %vectores
proprios en columnas de valores propios

syms x
polCaracXV = sym(charpoly(mTransNormalVeloc, x));
coefPolCaractV = poly (mTransNormalVeloc);
    
```

Se muestran a continuación los resultados obtenidos para el caso de velocidades.

VAPS de la matriz de transición de dirección ('VapsVelocidad'):

0,514618063
0,653470593
0,665582626
0,718793016
1
0,805422398
0,947220023
0,8854304

Matriz Diagonal D para velocidades del viento (VAPS en la diagonal, 'MatrizDiagonalVapsV'):

0,514618063	0	0	0	0	0	0	0	0	0
0	0,653470593	0	0	0	0	0	0	0	0
0	0	0,665582626	0	0	0	0	0	0	0
0	0	0	0,71879302	0	0	0	0	0	0
0	0	0	0	1**	0	0	0	0	0
0	0	0	0	0	0,8054224	0	0	0	0
0	0	0	0	0	0	0,94722002	0	0	0
0	0	0	0	0	0	0	0,8854304	0	0

***De nuevo en este caso, nótese que el VEP en la columna 1 de la matriz S va asociado al VAP de la columna 1 de la Matriz diagonal D, el de la columna 3 con el de la columna 3, y así sucesivamente con todos los valores*

Matriz S para velocidades, con los VEPS en columnas ('VepsVelocidad'):

0,000117062	-0,026696974	0,083961766	0,011868684	0,000992893	-0,005169862	-0,001976964	0,003052525
-0,010735114	0,212478071	-0,054895236	0,37313605	0,199612883	-0,432650509	-0,334519619	0,402755936
0,057454652	-0,499908479	-0,269717835	-0,751282292	0,584440241	0,182824418	-0,67851156	0,389928338
-0,191521725	0,636587953	0,629646488	0,301728024	0,650211373	0,743675863	-0,178730211	-0,552461487
0,448205286	-0,509212938	-0,660717777	0,330413399	0,401191035	-0,303069871	0,387246535	-0,560962229
-0,661425637	0,151102598	0,234730859	-0,283196407	0,172753345	-0,336692476	0,402570097	-0,039037565
0,537758478	0,107886997	0,137667538	-0,079765375	0,062451098	0,005963453	0,234910445	0,159763825
-0,179853002	-0,072237228	-0,100675803	0,097097915	0,033322392	0,145118984	0,169011277	0,196960657

Polinomio característico (en función de λ (el programa lo da en función de x pero se ha cambiado ya que se utiliza la misma notación que la empleada en el capítulo 3):

$$Q_V(\lambda) = \lambda^8 - 6,19\lambda^7 + 16,67\lambda^6 - 25,50\lambda^5 + 24,24\lambda^4 - 14,65\lambda^3 + 5,50\lambda^2 - 1,17\lambda + 0,11$$

Matriz de VEPS, S, normalizada, para velocidades:

0	0	0	0	0,00047169	0	0	0
0	0	0	0	0,09482909	0	0	0
0	0	0	0	0,27764708	0	0	0
0	0	0	0	0,30889264	0	0	0
0	0	0	0	0,19059181	0	0	0
0	0	0	0	0,08206906	0	0	0
0	0	0	0	0,02966833	0	0	0
0	0	0	0	0,0158303	0	0	0

Esta matriz tiene la misma circunstancia que la matriz de VEPS de direcciones, los valores que se dividían por 0 se han igualado a 0. La única columna que no suma 0 es la del VEP asociado al VAP de valor 1, de nuevo, que como se verá más adelante es el que interesa.

5.2. Búsqueda de la distribución estacionaria

El siguiente objetivo es encontrar una distribución estacionaria y por tanto la evolución a lo largo del tiempo tanto de las velocidades como de las direcciones. Para ello, usando los resultados mostrados se debe calcular la distribución estacionaria de la Cadena de Markov.

Observaciones de los resultados de VAPS y VEPS

Lo más importante es recalcar el hecho de que el radio espectral de las matrices de transición normalizadas es 1. Esto quiere decir que el valor propio de valor absoluto más alto es 1, y único además en ambos casos. De esta forma se puede asegurar que todos los VAPS son estrictamente más pequeños que 1, excepto 1, que es igual a 1. Se verá que este hecho resulta clave a la hora de calcular los límites de las matrices.

5.2.1. Distribución estacionaria para Velocidades del viento

Se quiere ver la tendencia de las **velocidades** del viento a lo largo del tiempo. A continuación se demuestra que la Cadena de Markov estudiada es regular y se encuentra esta tendencia.

En primer lugar, se comprueba la siguiente igualdad:

$$\lim_{k \rightarrow \infty} A^k = \lim_{k \rightarrow \infty} S D^k S^{-1}$$

Dicho en palabras, si se eleva la matriz de transición a un número muy elevado (tendiendo a infinito), el resultado obtenido es el mismo que el de elevar la matriz diagonal de VAPS multiplicada por la matriz de VEPS en columnas y su inversa.

A continuación, se muestra este límite elevando las matrices primero a 50, luego a 150 y luego a 500.

$$A^{50} = SD^{50}S^{-1} =$$

0,000528533	0,00052442	0,000507145	0,000478275	0,000440932	0,000402627	0,00036516	0,000331703
0,104270734	0,103613962	0,100785237	0,096002661	0,089715038	0,083153593	0,07663526	0,070757674
0,296150132	0,294956116	0,28956375	0,280245099	0,267615666	0,254011164	0,24011287	0,227363622
0,312796503	0,31268227	0,311777827	0,309904701	0,306778404	0,302747327	0,29803642	0,293384104
0,179544089	0,180322431	0,183658495	0,189274546	0,196592334	0,204129684	0,21151295	0,218106349
0,071376168	0,072025628	0,075073226	0,080431267	0,087865896	0,096069173	0,1046231	0,112566535
0,02368935	0,024017201	0,025657507	0,028623999	0,032902014	0,037810688	0,04309934	0,048105429
0,01164449	0,011857972	0,012976813	0,015039452	0,018089715	0,021675746	0,0256149	0,029384584

$$A^{150} = SD^{150}S^{-1} =$$

0,00047192	0,000471906	0,00047184	0,000471725	0,000471564	0,000471385	0,0004712	0,000471024
0,094868155	0,094865798	0,094854705	0,094835159	0,094807933	0,09477774	0,0947461	0,094716637
0,277726322	0,277721542	0,277699043	0,2776594	0,277604177	0,277542935	0,27747876	0,277418991
0,308913511	0,308912253	0,308906329	0,30889589	0,308881344	0,30886521	0,3088483	0,308832549
0,190546579	0,190549307	0,190562149	0,190584776	0,190616294	0,190651246	0,19068787	0,190721978
0,08202205	0,082024886	0,082038234	0,082061754	0,082094518	0,082130854	0,08216893	0,082204396
0,029640897	0,029642552	0,02965034	0,029664064	0,029683182	0,029704386	0,02972661	0,029747303
0,015810566	0,015811756	0,015817359	0,015827232	0,015840988	0,015856243	0,01587223	0,015887122

$$A^{500} = SD^{500}S^{-1} =$$

0,000471689	0,000471689	0,000471689	0,000471689	0,000471689	0,000471689	0,00047169	0,000471689
0,094829087	0,094829087	0,094829087	0,094829087	0,094829087	0,094829087	0,09482909	0,094829087
0,277647083	0,277647083	0,277647083	0,277647083	0,277647083	0,277647083	0,27764708	0,277647083
0,308892644	0,308892644	0,308892644	0,308892644	0,308892644	0,308892644	0,30889264	0,308892644
0,190591805	0,190591805	0,190591805	0,190591805	0,190591805	0,190591805	0,19059181	0,190591805
0,082069062	0,082069062	0,082069062	0,082069062	0,082069062	0,082069062	0,08206906	0,082069062
0,029668329	0,029668329	0,029668329	0,029668329	0,029668329	0,029668329	0,02966833	0,029668329
0,015830301	0,015830301	0,015830301	0,015830301	0,015830301	0,015830301	0,0158303	0,015830301

Como se puede observar, los resultados convergen en un valor común a medida que se aumenta el valor de la potencia k. Además, no solo se llega a una matriz a la que se llamará ‘matriz estacionaria’ a partir de ahora, sino que esta además tiene todas las columnas iguales y equivalentes al VEP de VAP 1. Por tanto, se ve se cumple la siguiente igualdad:

$$\lim_{k \rightarrow \infty} A^k p(0) =$$

0,00047169
0,09482909
0,27764708
0,30889264
0,19059181
0,08206906
0,02966833
0,0158303

Siendo el resultado igual al VEP de VAP con valor 1 y p(0) un estado inicial cualquiera. De esta forma se demuestra que al modelar los datos como una cadena de Markov, el resultado del límite es independiente del estado inicial.

5.2.2. Distribución estacionaria para Direcciones del viento

Para el caso de las direcciones del viento, el proceso para encontrar la distribución estacionaria es el mismo que el de las velocidades del viento. Por ello, básicamente se muestra a continuación el resultado obtenido, observando de nuevo la convergencia de la matriz de transición hasta llegar a la ‘matriz estacionaria de direcciones’ y también la distribución estacionaria independiente del estado inicial. Evidentemente, en este caso la matriz A es la matriz de transición normalizada de **direcciones**, y la matriz D la diagonal con los VAPS de esta matriz.

$$A^{50} = SD^{50}S^{-1} =$$

0,422808715	0,388228624	0,253179057	0,185952279	0,18033634	0,223538237	0,29708824	0,40446657
0,087406867	0,081054989	0,054425575	0,040901364	0,039609202	0,047664665	0,06187929	0,08352244
0,015066458	0,015152777	0,014915994	0,014708041	0,01463232	0,014537937	0,01456686	0,014939457
0,030648057	0,036272864	0,057052643	0,067181612	0,067866761	0,060724281	0,04914148	0,033211235
0,13623617	0,168143068	0,289904158	0,349933565	0,354463169	0,31396196	0,24685642	0,151991747
0,082270219	0,094032337	0,14181046	0,165923681	0,16818155	0,153641949	0,12802264	0,089150359
0,070571715	0,072780667	0,084264164	0,090507712	0,091434424	0,0890239	0,08344337	0,072786973
0,154991799	0,144334674	0,104447949	0,084891746	0,083476235	0,096907072	0,11900169	0,149931219

$$A^{150} = SD^{150}S^{-1} =$$

0,324944729	0,321295335	0,307165992	0,300158498	0,299595304	0,304186741	0,31191801	0,323061515
0,068020323	0,067298514	0,06450387	0,063117854	0,063006457	0,063914589	0,06544375	0,067647837
0,014854658	0,014847256	0,014818592	0,014804375	0,014803232	0,014812543	0,01482823	0,014850836
0,045554548	0,046112303	0,048271741	0,049342717	0,049428789	0,048727056	0,04754545	0,045842362
0,223937038	0,227214165	0,239902145	0,246194779	0,246700511	0,242577432	0,23563484	0,225628128
0,117116964	0,118413641	0,123434009	0,125923883	0,126123998	0,124492604	0,12174558	0,117786105
0,079289757	0,079609872	0,080849307	0,081464018	0,081513429	0,081110686	0,08043251	0,079454965
0,126281983	0,125208914	0,121054344	0,118993876	0,118828279	0,120178349	0,12245164	0,125728252

$$A^{500} = SD^{500}S^{-1} =$$

0,313543972	0,313542628	0,313537424	0,313534843	0,313534635	0,313536327	0,31353917	0,313543278
0,065765363	0,065765098	0,065764068	0,065763558	0,065763517	0,065763851	0,06576441	0,065765226
0,014831528	0,014831526	0,014831515	0,01483151	0,014831509	0,014831513	0,01483152	0,014831527
0,047296964	0,047297169	0,047297964	0,047298359	0,047298391	0,047298132	0,0472977	0,04729707
0,234174772	0,234175979	0,234180652	0,23418297	0,234183156	0,234181638	0,23417908	0,234175395
0,121167829	0,121168306	0,121170155	0,121171072	0,121171146	0,121170545	0,12116953	0,121168075
0,080289846	0,080289964	0,08029042	0,080290647	0,080290665	0,080290517	0,08029027	0,080289907
0,122929726	0,122929331	0,1229278	0,122927041	0,12292698	0,122927478	0,12292831	0,122929522

Vista la convergencia de la matriz estacionaria, de nuevo se puede comprobar que se cumple el hecho de que la tendencia a lo largo del tiempo es independiente del estado inicial de direcciones sabiendo que:

$$\lim_{k \rightarrow \infty} A^k p(0) =$$

0,31353977
0,06576453
0,01483152
0,04729761
0,23417854
0,12116932
0,08029021
0,12292849

Siendo este el VEP de VAP 1 encontrado con anterioridad y a la vez la distribución estacionaria de esta cadena de Markov.

5.3. Análisis de resultados

Habiendo encontrado las 2 distribuciones estacionarias, tanto para las velocidades como para las direcciones del viento, se analizan a continuación para saber cómo se traduce este resultado en el caso concreto de la Estación meteorológica de Iturgoyen.

5.3.1. Análisis de resultados para Velocidad

Básicamente, se observa el resultado final de distribución estacionaria obtenida, que se recuerda es igual a:

Distribución estacionaria de velocidades =

(0.000471, 0.094829, 0.277647, 0.308892, 0.190591, 0.082069, 0.029668, 0.01583)

Si se cambia el resultado a un porcentaje y se asocia a los estados pertinentes de velocidad, queda una distribución a lo largo del tiempo de las velocidades de la siguiente forma:

Estado de Velocidad	%
1	0,04716887
2	9,48290873
3	27,7647083
4	30,8892644
5	19,0591805
6	8,20690622
7	2,96683287
8	1,58303012

Tabla 4: Resultados globales velocidad

Por tanto, se concluye que en la Estación meteorológica de Iturgoyen, la tendencia de las velocidades del viento oscila, en un 80% del tiempo aproximadamente entre los 10 km/h y los 40 km/h, ya que si se suma el porcentaje de los estados 3, 4 y 5 la suma es de 77,7%. Además cabe destacar que prácticamente nunca (no llega al 10%) el viento se encuentra en el estado 2 o 1, lo que significaría una velocidad menor a 10km/h y por tanto insuficiente para la generación de energía eólica con un molino.

Por tanto, se puede decir que la estación de Iturgoyen, en lo que a velocidades se refiere, es un lugar muy bueno para colocar un parque eólico. Se recuerda que el rango de velocidades en las cuales se genera más energía eólica va desde los 7m/s hasta los 18m/s, es decir desde los 25km/h hasta los 65km/h aproximadamente. Como se puede ver en la distribución estacionaria, a lo largo de todo el año, la tendencia será que las velocidades superen los 20km/h en un 62,7%, que es la suma de los estados 4, 5, 6 y 7.

Además, también cabe destacar que aunque no se genere mucha energía, a partir de los 4m/s, es decir 14,4km/h, ya se genera. Por tanto, sabiendo que la velocidad será superior a 10km/h en un 90,5% del tiempo, sumando desde los estados 3 al 8, se puede decir que un molino eólico bien orientado generaría energía eólica prácticamente todo el tiempo, oscilando el 85% del tiempo.

5.3.2. Análisis de resultados para Direcciones

Para el caso de direcciones también se analiza la distribución estacionaria obtenida:

Distribución estacionaria de Direcciones =

(0.31353, 0.065764, 0.014831, 0.04729761, 0.234178, 0.12116, 0.08029, 0.12292)

A continuación se transforma la distribución obtenida en porcentajes asignados a los diferentes estados de direcciones del viento:

Estado de Dirección	%
1	31,3539771
2	6,57645326
3	1,48315199
4	4,72976056
5	23,4178544
6	12,1169321
7	8,02902144
8	12,2928491

Tabla 5: Resultados globales dirección

Así como al analizar los resultados de velocidades del viento, la tendencia temporal es positiva para colocar un molino o parque eólico, no se puede decir lo mismo respecto a las direcciones del viento. En este caso, no existe una estabilidad demasiado marcada hacia ninguna dirección en concreto. La dirección más probable a lo largo del tiempo es la del estado 1, es decir el Norte, con un 31,35% de probabilidad. Sin embargo, también hay una tendencia hacia el Sur, estado 5, es del 23,42%. Esto evidencia la inestabilidad en la dirección del viento.

De todas maneras, sí que se pueden sacar algunas conclusiones. En primer lugar, las direcciones asociadas a los estados 2, 3, 4 y 7, es decir Noreste, Este, Sureste y Oeste tienen muy poco peso, sumando entre los 4 estados un 20,8% tan solo. Esto quiere decir que el 79,2% del tiempo, el viento sopla en dirección Norte, Sur, Noroeste y Suroeste.

En segundo lugar, también es destacable el hecho de que los estados 1 y 5, Norte y Sur respectivamente, supongan más del 50% de la dirección del viento a lo largo del tiempo. Por ello, se puede decir, de manera muy general y aproximada, que en la estación de Iturgoyen, el viento o bien sopla al Norte, o bien al Sur, siendo el resto de direcciones mucho menos predominantes.

Finalmente, se puede decir que aunque no haya demasiada estabilidad en la dirección, al ser las velocidades las adecuadas, se podría colocar un molino eólico con la parte

superior giratoria, de tal manera que se orientase a la dirección deseada en función de hacia donde vaya, sabiendo que mayoritariamente el viento soplará al Sur y al Norte.

5.3.3. Conclusión global

Sabiendo por separado la tendencia a lo largo del tiempo de las velocidades y de las direcciones, se puede sacar una conclusión única sobre la situación meteorológica en la estación de Iturgoyen.

Se puede decir que por lo general, la el viento se dirigirá mayoritariamente al Norte o al Sur (55% del tiempo) y a una velocidad superior a los 20km/h (62,7%). Conociendo esta información, extraída analizando los datos desde Abril de 2015 hasta Marzo de 2016 de la estación meteorológica de Iturgoyen, se facilita la implementación de un posible parque eólico en este lugar.

Evidentemente, también cabe mencionar que los resultados obtenidos son tan solo una tendencia a lo largo del tiempo futuro, ya que este no se puede predecir con exactitud.

5.4. Ejemplos específicos

Todo el estudio anterior, se ha realizado con la totalidad de los datos recogidos de la Estación meteorológica de Iturgoyen. Se ha demostrado que el programa funciona correctamente con el análisis anterior y además los datos se han validado para ser modelados como una cadena de Markov. Por ello, a continuación se muestra el análisis de 2 días concretos del año elegidos al azar, el 14 de Abril de 2015 y el 22 de Noviembre de 2015. Esto supone una menor cantidad de datos por lo que los resultados son más uniformes.

Evidentemente, se saltan todos los pasos previos y se muestra únicamente el resultado final obtenido con el programa de Matlab. Además, no hace falta validar de nuevo si los datos son modelables como una cadena de Markov, ya que si todos los datos lo son, una fracción de los mismos se puede modelar igualmente.

5.4.1. Día 14 de Abril de 2015

Analizando los resultados obtenidos para este día en concreto, se obtienen los siguientes resultados para dirección y para velocidad del viento.

Dirección	Estado	%
Norte	1	0
Noreste	2	0
Este	3	0
Sureste	4	22,9030339
Sur	5	77,0969661
Suroeste	6	0
Oeste	7	0
Noroeste	8	0

Tabla 6: Resultados Dirección 14 de Abril

Velocidad (km/h)	Estado	%
0< Valor <=0,3	1	0
0,3< Valor <=10	2	0
10< Valor <=20	3	14,3835616
20< Valor <=30	4	75,3424658
30< Valor <=40	5	10,2739726
40< Valor <=50	6	0
50< Valor <=60	7	0
60< Valor	8	0

Tabla 7: Resultados Velocidad 14 de Abril

Se puede observar, que si sólo se analizan los datos de este día, la velocidad predominante será la que vaya de 20 a 30 km/h y la dirección más habitual será el Sur, con un 77% de tendencia. Estos resultados son positivos, ya que indican que aunque, como se ha visto en la conclusión global, hay bastante inestabilidad en las direcciones, si se mira a corto plazo de tiempo sí que hay estabilidad.

Se ha visto anteriormente que mayoritariamente el viento iba o bien hacia el sur, o bien hacía el norte. Con este análisis se demuestra que es cierto, y que a corto plazo existe una alta estabilidad direccional que favorecería la implementación de un parque eólico. Además, la velocidad predominante coincide con la extraída del análisis global anterior, del estado 4.

5.4.2. Día 22 de Noviembre de 2015

A continuación se muestran los resultados obtenidos para este día.

Dirección	Estado	%
Norte	1	94,4055944
Noreste	2	0
Este	3	0
Sureste	4	0
Sur	5	0
Suroeste	6	0
Oeste	7	0
Noroeste	8	5,59440559

Tabla 8: Resultados Dirección 22 de Noviembre

Velocidad (km/h)	Estado	%
0< Valor <=0,3	1	0
0,3< Valor <=10	2	0
10< Valor <=20	3	0
20< Valor <=30	4	9,79020979
30< Valor <=40	5	74,8251748
40< Valor <=50	6	14,6853147
50< Valor <=60	7	0,6993007
60< Valor	8	0

Tabla 9: Resultados Velocidad 22 de Noviembre

En este caso, los resultados también son muy positivos y encajan a la perfección con la conclusión global realizada. En este caso, la dirección claramente predominante es el Norte, con una tendencia del 94,4%, y la velocidad es superior a 30 km/h gran parte del tiempo. Por tanto, analizando este día, se puede predecir a corto plazo que al viento soplará al norte y que tendrá una velocidad superior a 20km/h.

De nuevo, se ve una estabilidad en lo que se refiere tanto a la dirección como a la velocidad del viento. Esto, como ya se ha dicho, favorecería a un molino eólico ya que no existen cambios bruscos de temperatura.

La conclusión principal a sacar del estudio específico de ambos días es la estabilidad predecible a corto plazo, como ya se ha dicho. Esto quiere decir que en la estación de Iturgoyen, la dirección del viento va al sur o al norte, pero no existen cambios abruptos de dirección y por tanto un parque eólico funcionaría correctamente. Además, se corrobora que el viento tiene, gran parte del tiempo, la velocidad mínima necesaria para que un molino eólico funcione.

6. Planificación

La planificación del trabajo ha sido fundamental para poder organizar los tiempos e ir cumpliendo los objetivos. Teniendo en cuenta que el trabajo final de grado tiene un peso de 12 créditos, es decir, 300 horas aproximadamente (considerando 25 horas por crédito), esta planificación se adapta a este tiempo, y una duración de más de 3 meses.

A continuación se muestra una tabla y un diagrama de Gantt que indican las tareas realizadas, que prácticamente coinciden con los capítulos del trabajo, aunque si existe alguna diferencia es porque la planificación se realizó al inicio.

Actividad	Fecha inicio	Duración (días)	Fecha final
Introducción, Búsqueda de información	01/02/2016	17	18/02/2016
Lectura datos excel y tratamiento información	08/02/2016	13	21/02/2016
Análisis de datos: Generación de Matriz	21/02/2016	14	06/03/2016
Análisis de datos: post proceso	06/03/2016	13	19/03/2016
Validación cadena de markov	19/03/2016	13	01/04/2016
Estudio predictivo de la cadena de Markov	01/04/2016	19	20/04/2016
Resto (Impactos, presupuestos)	20/04/2016	6	26/04/2016
Finalización, detalles e impresión y corrección	26/04/2016	10	06/05/2016
Presentación	07/05/2016	6	13/05/2016

Tabla 10: Duración de las tareas realizadas

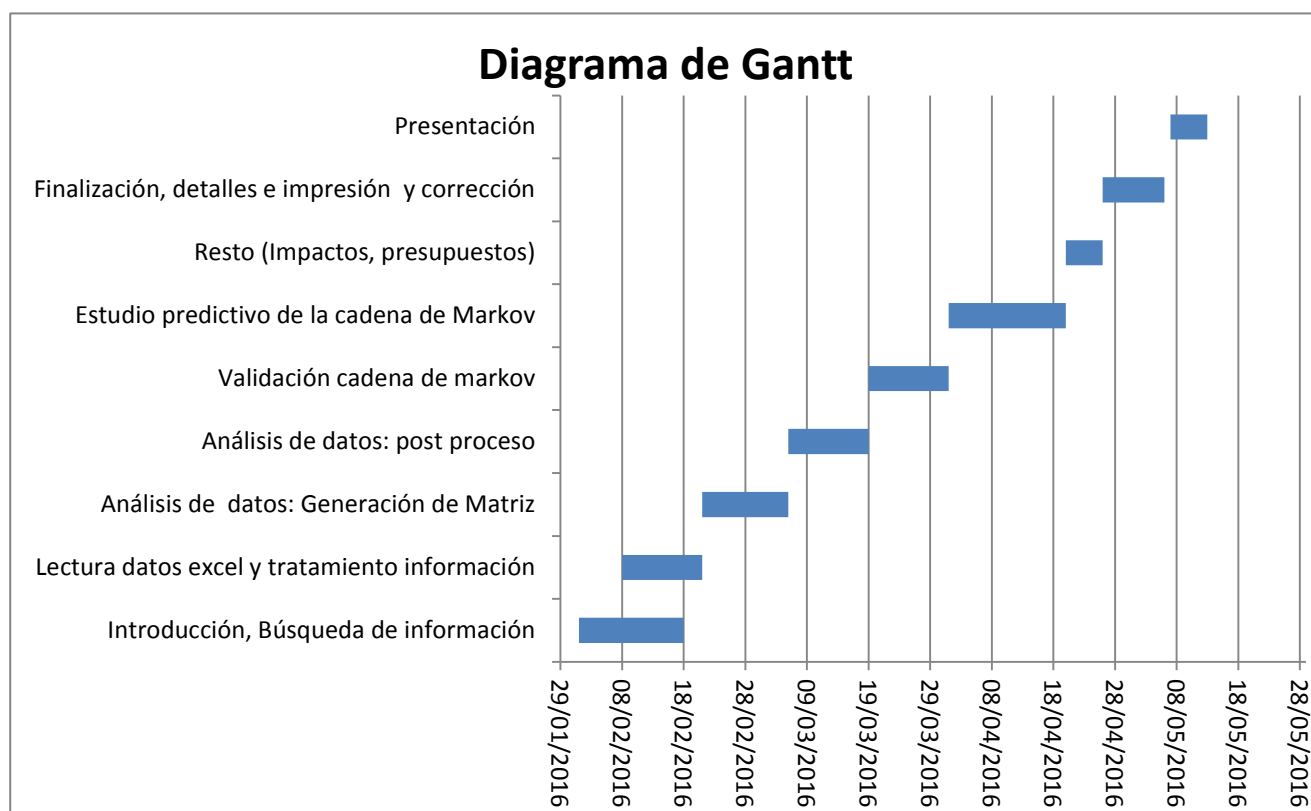


Figura 14: Diagrama de Gantt del proyecto

También cabe destacar que en lo que a la división de las 300 horas de duración del proyecto se refiere, la parte a la que más se ha dedicado es a la creación del programa para el análisis de datos junto con el estudio predictivo de los datos escogidos, suponiendo un total de 72 días de los aproximadamente 102 que ha durado todo el proyecto, y por tanto más de un 70% de la carga del trabajo.

7. Presupuestos

Se ha tenido en cuenta el coste que ha supuesto la realización de este trabajo. Básicamente se incluyen gastos como el de Ingeniero Industrial y Licencias, como se ve en la tabla siguiente.

Concepto	Utilización (horas)	Precio unitario (€/KWh)	Precio unitario (€/h)	Coste (€)
Ingeniero Industrial	300	-	22	6600
Licencia MATLAB R2016a educacional (1 año) [15]	-	-	-	35
Matlab Add-on product (Neural Network Toolbox)[15]	-	-	-	20
Matlab Add-on product (Symbolic Math Toolbox)[15]	-	-	-	20
Licencia MicrosoftOffice 2010 (1 año)[17]	-	-	-	110
Ordenador	-	-	-	600
Consumo eléctrico ordenador (0,15 KW)[18]	250	0,13	-	4,875
Consumo internet ADSL (cuota cuatrimestral)[19]	-	-	-	87,2
Material de oficina e impresiones	-	-	-	110
Coste total sin IVA (€)				7587,075
IVA (21%)				1593,2858
Coste total (€)				9180,3608

Tabla 11: Presupuestos

Se puede observar en la Tabla 9.1 que la mayor parte del precio proviene del sueldo de Ingeniero Industrial. De facto, del total del coste del proyecto, este sueldo tiene un peso de más del 70%. Se observa que el coste global del proyecto, incluyendo IVA (impuesto sobre valor añadido) es de 9180,36 Euros. En este TFG, además, las licencias MATLAB utilizadas son las versiones para estudiantes, por lo que no han tenido un alto coste como se observa en la tabla, sumando un total de 75 euros entre MATLAB y los paquetes adicionales usados.

8. Impacto Ambiental

Se ha decidido incluir un apartado que se refiera al impacto ambiental del proyecto realizado, ya que se considera un tema de vital importancia hoy en día y se debe tener en cuenta siempre que se realice cualquier tipo de trabajo, ya sea académico, profesional o de cualquier otra índole.

Sin embargo, en lo que se refiere a este Trabajo Final de Grado, el impacto ambiental es prácticamente nulo. Partiendo de la base de que el trabajo es fundamentalmente teórico, el único aparato usado a lo largo de todo el trabajo es el ordenador, en este caso portátil. De esta forma el único impacto ambiental es el consumo de energía de la red eléctrica, que evidentemente supone un impacto prácticamente despreciable en comparación a otros proyectos mucho más prácticos o aplicados. Sí que es cierto, que aun siendo un impacto pequeño, se debe tener en cuenta ya que la mayor parte de la energía generada en España no proviene de fuentes renovables, por lo que el consumo de energía de la red eléctrica del ordenador portátil tiene un impacto negativo ya que al ser usado durante muchas horas la energía gastada es suficientemente alta como para ser considerada para el impacto ambiental.

Si el proyecto alcanzara un objetivo futuro de búsqueda de un lugar idóneo para la construcción de un parque eólico entonces sí que se hablaría de un gran impacto ambiental, tanto positivo como negativo, al que no se entrará en detalle ya que no está incluido en este proyecto.



Figura 15: mundo sostenible [14]

Conclusiones

Como ya se ha expuesto al inicio y a lo largo del proyecto, el objetivo principal marcado ha sido el diseño y simulación de un modelo predictivo para la dirección y velocidad del viento mediante cadenas de Markov.

Basado en este objetivo, y una vez finalizado el proyecto, se extraen las siguientes conclusiones:

En primer lugar, se ha cumplido con el objetivo propuesto. Se han creado dos programas (uno para velocidades y otro para direcciones) con Matlab que son capaces de, a partir de unos datos meteorológicos importados desde una estación meteorológica y que vienen dados en formato Excel, analizarlos y guardar una distribución estacionaria que indique la tendencia de la velocidad y la dirección del viento en esa zona. El buen funcionamiento se ha corroborado con la simulación realizada con los datos recogidos de la estación meteorológica de Iturgoyen, Navarra. El programa ha leído, clasificado y analizado los datos de tal manera que se han podido obtener resultados coherentes y deseados, visto tanto en el caso con todos los datos como en ejemplos específicos con datos de días concretos del año.

De forma paralela, pero no independiente, se ha programado la validación de los datos como Cadena de Markov. El motivo de realizar el programa aparte es básicamente porque no tiene sentido estudiar la evolución de las condiciones del viento si los datos no son markovianos. Además, se evita ralentizar los cálculos.

Por tanto, sencillamente lo primero que hay que hacer con los datos meteorológicas a estudiar es validarlos como una cadena de Markov. Una vez validada la cadena, entonces se utiliza el programa principal para predecir la tendencia futura de las condiciones del viento.

En segundo lugar, a nivel académico, este proyecto ha supuesto la superación de un reto personal y ha sido muy útil. Se han repasado muchos conceptos teóricos de Álgebra Lineal y se han aprendido muchos nuevos. Además, se ha mejorado ostensiblemente en la utilización de Matlab, tanto a nivel general como a nivel de programación. Como bien se expone en la motivación del proyecto, se escogió este tema precisamente por las ganas de mejorar tanto en el ámbito de Álgebra como en el de programación, y esto se ha conseguido con creces. Junto con esto, Matlab es un programa que tiene muchas aplicaciones y es muy utilizado en muchos ámbitos, por lo que la utilidad del trabajo es inmensa, ya que servirá, seguro, para el futuro. De facto, se puede ver una gran progresión en el nivel de los ficheros realizados a medida que se avanza en el proyecto, empezando con una programación más básica pero ineficiente y terminando con una programación mucho más simple y eficiente.

Finalmente, este trabajo puede tener aplicaciones reales para decidir zonas donde se puedan situar parques eólicos. Sí que es cierto que el diseño se puede refinar, pero lo que está claro es que funciona y se puede utilizar para fines relacionados con la energía eólica, y esto se ha demostrado sobradamente con este proyecto, analizando la evolución de las condiciones de velocidad y dirección del viento en la estación de Trinidad de Iturgoyen.

Agradecimientos

En primer lugar, este proyecto no hubiese sido posible sin la ayuda de mi tutora, María Isabel García Planas. Durante todo el proceso, siempre ha estado ayudándome activamente y ha estado a disposición tanto por email como presencialmente con más de una reunión por semana. Quiero agradecerle su interés y ganas mostradas a lo largo de estos últimos meses.

En segundo lugar, quiero agradecer, aunque haya estado fuera prácticamente todo el proyecto, la ayuda del Sr. Jose Luis Domínguez, codirector de mi trabajo también. Durante todo el tiempo ha estado disponible vía email y me ha resuelto con prontitud todas las dudas que me han aparecido.

Finalmente, agradecer a todos mis seres cercanos, incluyendo familiares y amigos, que me han apoyado moralmente todo el tiempo. Quiero destacar dos nombres propios entre ellos. Por un lado Marc, que me ha dedicado un buen rato para animarme en algunos momentos, y por otro lado, Carlos, con el que he estudiado los últimos 15 años, día a día, hasta ahora.

Bibliografía

- [1] R.B. Bapat, T.E.S. Raghavan, Nonnegative Matrices and Applications, Encyclopedia of Mathematics and its Applications (No. 64), Cambridge University Press, 1997.
- [2] T. Burton, D. Sharpe, N. Jenkins, E. Bossanyi. Wind Energy Handbook. John Wiley & Sons, Ltd. 2001.
- [3] 'Ayuda de Matlab' [en línea] [Visitado a lo largo de todo el trabajo], Disponible en: <<http://es.mathworks.com/help/matlab/index.html>>
- [4] A. Berman, R.J. Plemmons, Non-negative Matrices in Mathematical Sciences, SIAM. Philadelphia 1994.
- [5] J.L. Domínguez García, M.I. García Planas. Introducción a la teoría de matrices positivas. Aplicaciones. Iniciativa Digital Politècnica, 2013.
- [6] M.I. García Planas, M.D. Magret Planas. Eines d'àlgebra lineal i matricial per a l'enginyeria. Editado por las autoras. Barcelona, 2014.
- [7] M.C. Palafox Duarte. Inferencia Estadística para Cadenas de Markov. Tesis, México, (2009).
- [8] 'Linpack' : [en línea] [Visitado el 01/02/2016], Disponible en: <<http://www.netlib.org/linpack/>>
- [9] 'Eispack' : [en línea] [Visitado el 01/02/2016], Disponible en: <<http://www.netlib.org/eispack/>>
- [10] 'Energía eólica en España' : [En línea] [Visitado el 03/02/2016], Disponible en: <<http://www.aeeolica.org/es/sobre-la-eolica/la-eolica-en-espana/>>
- [11] 'Imagen 360 grados' [En línea] [Visitado el 18/02/2016], Disponible en: <<http://www.cienciacierta.org/2009/11/los-360-grados-que-te-dejan-igual.html>>
- [12] 'Rosa de los vientos' [En línea] [Visitado el 18/02/2016], Disponible en: <<http://geografia.laguia2000.com/general/rosa-de-los-vientos>>
- [13] 'Datos trinidad de Iturgoyen' [En línea] [Visitado el 19/02/2015], Disponible en: <<http://meteo.navarra.es/estaciones/estacion.cfm?IDEstacion=29>>
- [14] 'Mundo sostenible' [En línea][Visitado el 22/04/2016], Disponible en: <<http://www.responsabilidadsocial.mx/106-articulos/fundamentos/736-impacto-ambiental-definicion-e-instrumentos-de-evaluacion.html>>

- [15] 'Matlab' [En línea] [Visitado el 23/04/2016], Disponible en:
<http://es.mathworks.com/pricing-licensing/?prodcode=ML&s_iid=main_pl_ML_nav>
- [16] Juan Miguel García, Matrices no negativas, paseos aleatorios y cadenas de Markov, Universidad del País Vasco, Matemática aplicada y Estadística, 2002
- [17] Microsoft Office Pack. [Paquete de software informático]. Disponible en:
<<https://products.office.com/es-es/home>>
- [18] "Catálogo tarifas luz". [En línea]. [Visitado el 22/04/2016]. Disponible en:
<<https://www.endesaclientes.com/catalogo/luz.html>>
- [19] "Tarifas internet ADSL movistar". [En línea]. [Visitado el 22/04/2016]. Disponible en: <<http://www.movistar.es/particulares/internet/adsl-fibra-optica/>>.

Anexos

A. Matrices de probabilidad 1

Matriz de probabilidad 1 para velocidades: 'MatrizProbabilidad1'

0,66666667	0,00165803	0	0	0	0	0	0
0,29166667	0,82694301	0,05839881	0,00019101	0	0	0	0
0	0,16994819	0,84016422	0,09079333	0,00113601	0	0	0
0,04166667	0,00124352	0,10079989	0,83242073	0,12351544	0,00096061	0,00066578	0
0	0	0,00063708	0,07614924	0,80016524	0,17387128	0,00332889	0,00124844
0	0,00020725	0	0,00038202	0,07425385	0,74471662	0,22237017	0,00374532
0	0	0	6,37E-05	0,00092946	0,07877041	0,67376831	0,1897628
0	0	0	0	0	0,00168108	0,09920107	0,80524345

Matriz de probabilidad 1 para direcciones: 'MatrizProbabilidad1D'

0,91206724	0,11423445	0,01061008	0,00207987	0,0004201	0,00227347	0,00710784	0,15340272
0,02728299	0,84210526	0,08753316	0,00374376	0,00025206	0,00064956	0,0004902	0,00144115
0,00043904	0,03199761	0,71485411	0,03244592	0,00109225	0,00097434	0,0004902	0,00032026
0,00018816	0,00179426	0,15251989	0,74084859	0,03990926	0,0034102	0,0004902	0,00016013
0,0003136	0,00179426	0,0265252	0,21048253	0,90438582	0,09645989	0,00147059	0,00016013
0,0006272	0,00328947	0,0066313	0,00873544	0,0525962	0,82461838	0,09387255	0,00384307
0,00094079	0,00119617	0,00132626	0,00124792	0,00075618	0,06706723	0,83406863	0,03714972
0,05814099	0,00358852	0	0,00041597	0,00058814	0,00454693	0,0620098	0,80352282

B. Fichero 'EsMarkovianoDireccion'

```
%carácter markoviano dirección
matrizDeTransicionDireccion
%comprobación de carácter markoviano
DsaleDe1 = size(find(vectorDViento == 1),2);
DsaleDe2 = size(find(vectorDViento == 2),2);
DsaleDe3 = sumaColumnas(3);
DsaleDe4 = sumaColumnas(4);
DsaleDe5 = sumaColumnas(5);
DsaleDe6 = sumaColumnas(6);
DsaleDe7 = sumaColumnas(7);
DsaleDe8 = sumaColumnas(8);
MatrizProbabilidad1D = zeros(8,8);
MatrizProbabilidad1D(:,1) = matrizTransicionD(:,1)/DsaleDe1;
MatrizProbabilidad1D(:,2) = matrizTransicionD(:,2)/DsaleDe2;
MatrizProbabilidad1D(:,3) = matrizTransicionD(:,3)/DsaleDe3;
MatrizProbabilidad1D(:,4) = matrizTransicionD(:,4)/DsaleDe4;
MatrizProbabilidad1D(:,5) = matrizTransicionD(:,5)/DsaleDe5;
MatrizProbabilidad1D(:,6) = matrizTransicionD(:,6)/DsaleDe6;
MatrizProbabilidad1D(:,7) = matrizTransicionD(:,7)/DsaleDe7;
MatrizProbabilidad1D(:,8) = matrizTransicionD(:,8)/DsaleDe8;
%Ya tengo una matriz con las probabilidades de pasar de un estado X a
un
```

```
%estado Y dividido por las veces que se sale de X
MatrizProbabilidad1DGrafico = reshape (MatrizProbabilidad1D, 64,1);
%Me he generado la columna deseada para hacer el gráfico en
%Ahora debo buscar matriz probabilidad 2, tal que se llegue de estado
z a
%x a y
condicionesProbabilidad
%hago una búsqueda de todas las ocurrencias/condiciones
i=1;
VectorProbabilidad2D = [];
matrizCantidadesD = [];
while i<=size(M,2)
    buscado = M(:,i)';
    numeroRepeticiones = length(strfind(vectorDViento, buscado));
    VectorProbabilidad2D(i) =
numeroRepeticiones/matrizTransicionD(buscado(2),buscado(1));
    matrizCantidadesD(i) =
numeroRepeticiones/matrizTransicionD(buscado(2),buscado(1));
    i = i+1;
end
VectorProbabilidad2D(isnan(VectorProbabilidad2D))=0;
matrizCantidadesD(isnan(matrizCantidadesD))=0;
matrizFinalPr2D = reshape (matrizCantidadesD,64,8);
%al final consigo una matriz 64x8 donde se se tienen las 512
probabilidades
%calculadas y divididas, 111,112,113.....121,122 (columna 1),
%211,212,...221 (columna 2).....
%Finalmente dibujo el gráfico que enseña si se puede modelar como una
%cadena de markov
plot (MatrizProbabilidad1DGrafico, matrizFinalPr2D, '.', 'markersize',
12)
title ('Comprobación de datos Markovianos para las direcciones')
grid on;
refline(1,0)
```

