

Control del ángulo de Pitch de una Turbina Eólica mediante el uso del algoritmo Temple Simulado

Octavio Deshays Moreno
octaviodeshays@gmail.com

Control y Sistemas, Facultad de Ingeniería,
Universidad Nacional de Cuyo,
Mendoza, Argentina

Febrero de 2021

Resumen

Durante el siglo XX la principal fuente de energía utilizada para la generación de energía eléctrica fueron el petróleo y el carbón, fuentes de energía que son altamente nocivas para el medio ambiente en todos sus procesos, desde que se extraen hasta que finalmente son utilizadas como combustible. Al comenzar el siglo XXI, en especial en la última década, se tomó conciencia de la amenaza que representa el cambio climático originado por las emisiones de gases de efecto invernadero, lo que llevó a que las inversiones y desarrollos en energías renovables aumenten considerablemente, entre las que se destaca la energía eólica. En este proyecto se propone implementar un tipo de control que optimice la energía que una turbina eólica obtiene del viento, para lo que se utiliza un algoritmo de Inteligencia Artificial llamado Temple Simulado. Para ello se realizarán modelos en SIMULINK que permitan poner a prueba el controlador diseñado y compararlo con otros tipos de control. Se espera poder aumentar considerablemente la energía entregada por el generador.

1. Introducción

En la actualidad, el ser humano se encuentra frente a uno de los mas grandes desafíos en su historia: cambiar de un modelo económico de explotación a uno que sea sostenible con el medio ambiente, para así cumplir con los objetivos fijados en el Acuerdo de París y evitar las consecuencias catastróficas del Cambio Climático. Entre una de las tantas transiciones que deben ser realizadas, resulta fundamental la transición de una matriz energética basada en los hidrocarburos a una basada en energías limpias y renovables. Sin dudas, la energía eólica es una de las que está liderando esta transición (junto a las energías solar e hidroeléctricas).

De acuerdo a lo informado por el "Global Wind Energy Council" [1] en 2019 se instalaron 60.4 GW de capacidad en energía eólica, lo que representa un incremento del 19 por ciento respecto a las instalaciones realizadas en el año 2018. Esto representó también un aumento del 10 por ciento en la capacidad total instalada, haciendo que llegue a 651 GW. Para el 2020 se esperaban valores aún mayores, pero esto probablemente se vio afectado por la pandemia. Este considerable crecimiento es impulsado por el abaratamiento del costo para generar energía con recurso eólico, lo cual se puede observar en la Figura 1.

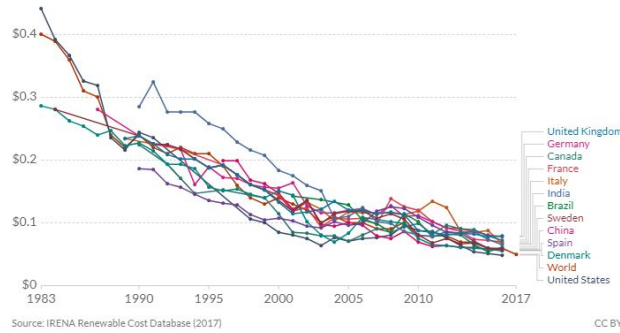


Figura 1: Precio de la energía renovable Onshore por kilowatt-hora [2]

Por el otro lado, si bien la energía eólica tiene el beneficio de ser limpia y poco costosa, presenta el inconveniente de depender de un recurso tan variable como lo es el viento, cuya velocidad puede variar a lo largo de un día en un rango que abarca desde 2 m/s hasta 25 m/s y, por ello, es muy importante poder aprovechar al máximo dicho recurso.

Teniendo en cuenta este contexto internacional y las limitaciones del recurso eólico, se busca desarrollar un método de control que optimiza la generación de energía para una velocidad de viento dada. Para ello se utiliza un controlador que combina un Algoritmo de optimización llamado Temple Simulado junto con un algoritmo de control PI que limita la energía generada a la nominal cuando se tienen velocidades de viento considerables. Para el desarrollo se utiliza el modelo de uno de los tipos de turbinas más utilizadas: Fixed Speed Variable Pitch (FSVP). Dicha turbina se caracteriza por ir conectada directamente a la red y, por lo tanto, su velocidad se mantiene constante. A su vez, se varía el ángulo de ataque de las aspas (Variable Pitch) para controlar la potencia entregada al generador eléctrico. Este último, el ángulo de ataque, es el valor que modificará nuestro controlador tanto para optimizar como para limitar el valor de potencia entregada. Para el desarrollo del proyecto se utilizó Matlab/Simulink.

El objetivo principal de este proyecto radica en optimizar el funcionamiento de una turbina eólica, haciendo que entregue mayor cantidad de energía con un mismo recurso eólico. Además, se busca integrar todos los conocimientos adquiridos en la Cátedra de Control y Sistemas para el entendimiento del problema, modelado de la planta, diseño del algoritmo de control y análisis de resultados.

2. Desarrollo

2.1. Modelo de la Planta

En una turbina eólica de velocidad constante, la energía generada dependerá directamente del torque entregado al generador. Dicho torque dependerá de las siguientes variables:

1. Ángulo de ataque β (Pitch Angle)
2. Ángulo de Yaw φ
3. Velocidad del Viento U

Dichas variables se pueden observar en la Figura 2.

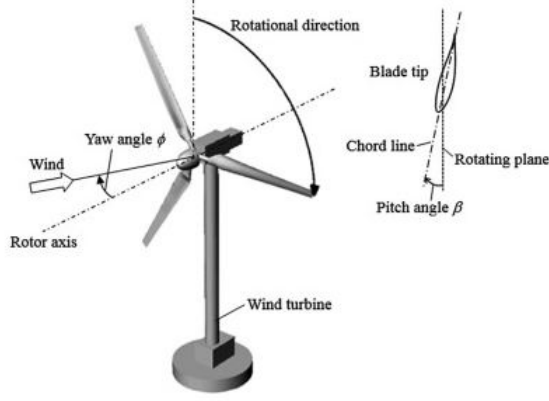


Figura 2: Esquema de turbina eólica.

El ángulo de Yaw se modifica para que el plano definido por las aspas se encuentre siempre perpendicular a la componente horizontal de la velocidad del viento. En este proyecto el control de dicha variable no será realizado, es decir, se considera que la turbina se encuentra siempre en dicha posición.

Para el modelado del sistema se utilizan los modelos ya provistos por Simulink, tanto para la turbina como para el generador eléctrico. De todas maneras, antes de comenzar con el modelado en Simulink se analiza el modelo matemático que utiliza el bloque de la turbina eólica para así poder entender las variables implicadas y su influencia en el torque entregado al generador.

2.1.1. Modelo matemático

La potencia entregada por la turbina se puede calcular como (3):

$$P(U) = \frac{1}{2} \cdot \rho \cdot A \cdot C_p(\beta, \lambda) \cdot U^3 \quad (1)$$

Donde:

1. ρ es la densidad del aire
2. A es el área de la turbina
3. C_p es el coeficiente entre la potencia en el rotor de la turbina y la potencia presente en el viento
4. U es la velocidad del viento

Según la documentación provista por Matlab sobre el modelo de la turbina eólica [4], el valor C_p puede ser calculado como se indica en la ecuación (2).

$$C_p(\beta, \lambda) = c_1 \cdot \left(\frac{c_2}{\lambda_i} - c_3 \cdot \beta - c_4 \right) \cdot e^{\frac{-c_5}{\lambda_i}} + c_6 \cdot \lambda \quad (2)$$

Donde:

$$\lambda = \frac{\omega R}{U} \quad (3)$$

$$\frac{1}{\lambda_i} = \frac{1}{\lambda + 0,08 \cdot \beta} + \frac{0,035}{\beta^3 + 1} \quad (4)$$

El valor C_p puede tomar un valor máximo de 0.59, esto se conoce como el Límite de Betz y es la mayor cantidad de energía que puede extraer una turbina del viento. Este coeficiente tiene un rol central en el proyecto, ya que es el valor que se buscará maximizar. Esto se ve en más detalle en la sección 2.2.

2.1.2. Modelado utilizando Matlab/Simulink

El modelo que se utiliza se puede dividir en tres bloques fundamentales:

1. Turbina Eólica: recibe como entrada la velocidad del viento, el ángulo de ataque β y la velocidad de giro del generador, y entrega a la salida el torque motor T_m que la turbina obtiene del viento y el valor λ en ese momento. Se trabajará con una turbina de 2 MW de potencia nominal.
2. Generador Eléctrico: se utiliza un Generador de inducción de tres polos con rotor de Jaula de Ardilla. Este bloque recibe como entrada el Torque motor generado por la turbina y entrega a la salida los valores de: Potencia activa (P), Potencia reactiva (Q), velocidad del generador (ω_r) y terminales de fase A, B y C. Se trabajará con un generador de potencia nominal de 2.2 MW.
3. Controlador PI: Este bloque recibe como entrada la potencia activa que está entregando el generador, la potencia de referencia (un valor levemente inferior a la potencia máxima a la que dicho generador puede trabajar) y entrega a la salida un valor de ángulo de ataque β . Este valor de β se entrega como consigna a un actuador que modifica dicho valor en la turbina y, de este modo, se controla la potencia entregada por la misma.

En la Figura 3 se puede observar un diagrama de bloques que representa dicho sistema.

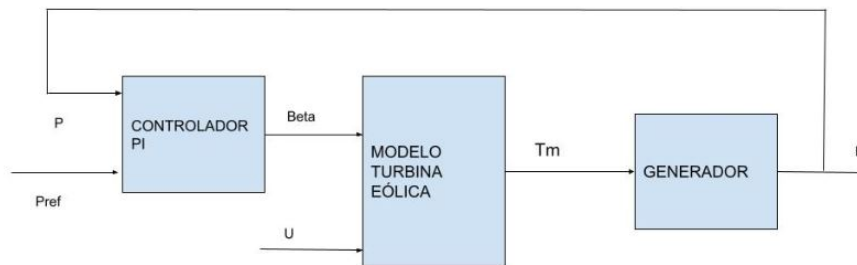


Figura 3: Diagrama de bloques de la planta

Posteriormente, se implementó dicho modelo en Simulink, donde, para la turbina, el generador y el controlador PI se utilizaron bloques ya presentes en la herramienta. Dicho modelo se puede observar en la Figura 4. Además, se realiza la conexión directa del generador a la red, haciendo que el sistema se mueva a velocidad constante.

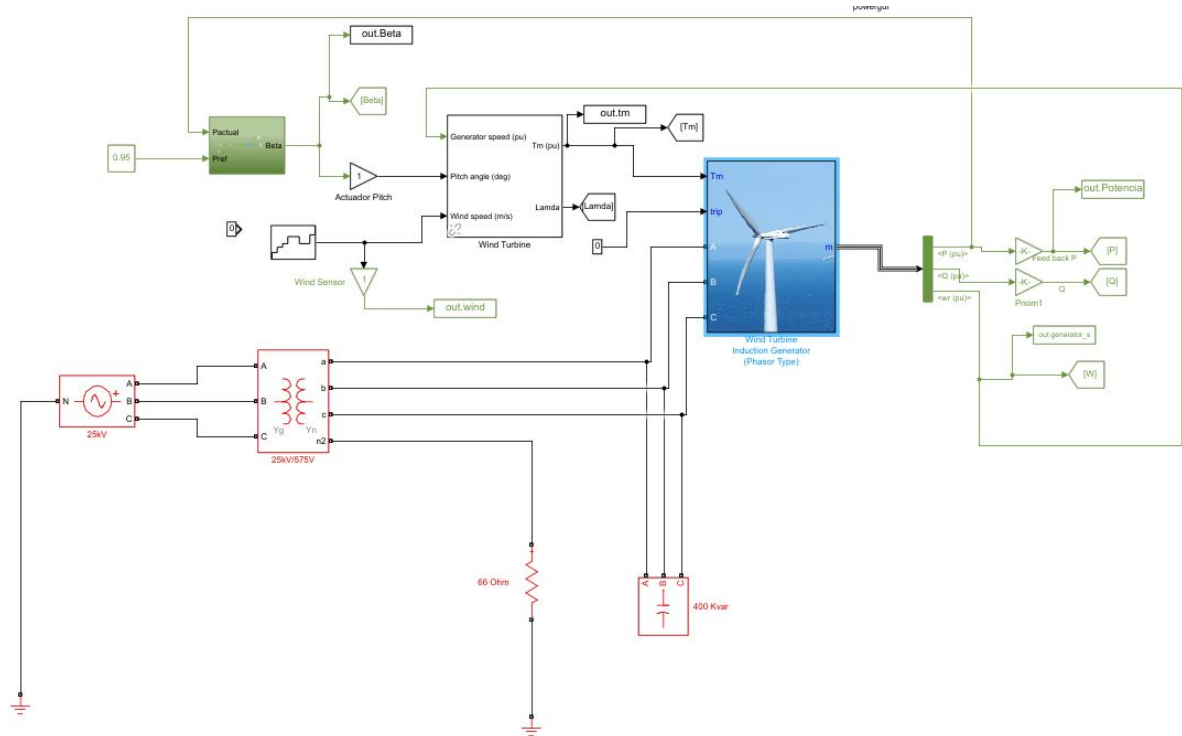


Figura 4: Implementación del modelo de la planta en Simulink. Verde: controlador, Negro: turbina eólica, Azul: generador y Rojo: conexión a la red.

Luego, se simula dicho modelo para las velocidades de viento de la Figura 5 y se pueden observar los resultados para la potencia generada en la Figura 6. Se puede ver que, la potencia generada, aumenta cuando aumenta la velocidad del viento hasta que la potencia a la salida del generador supera el valor del 95 % de la potencia nominal, allí comienza a actuar el controlador PI, que modifica el valor de β para limitar el torque que entrega la turbina y, en consecuencia, limitar la potencia generada. Cuando la potencia entregada es menor que el 95 % de la nominal, el ángulo de ataque simplemente se mantiene en 0. En la Figura 7 se puede observar cómo varía el valor de β durante la simulación.

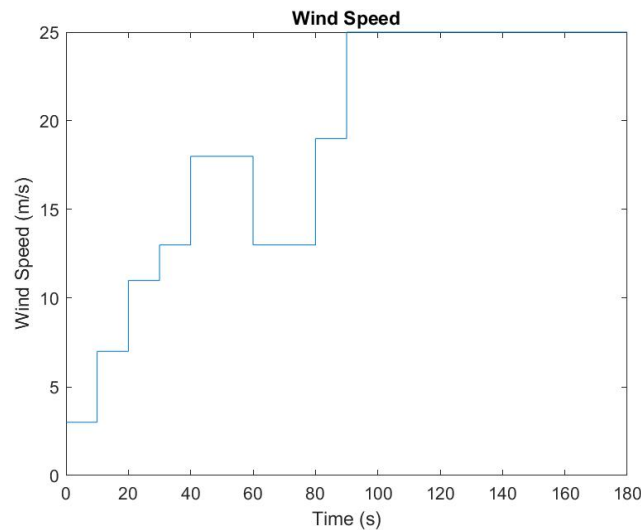


Figura 5: Velocidad del viento utilizada en Simulación.

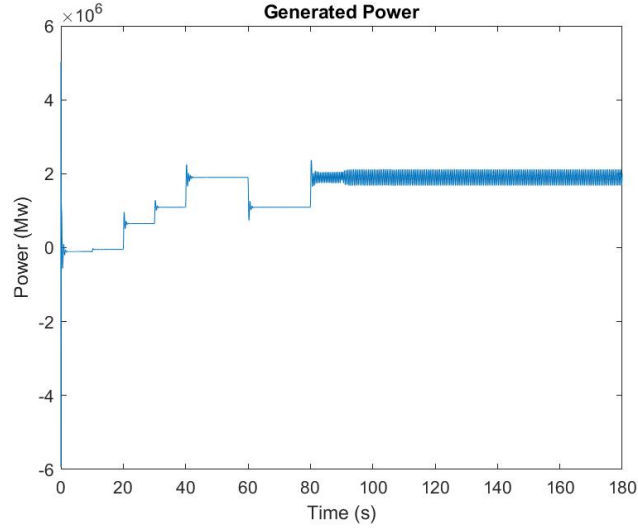


Figura 6: Potencia entregada por el generador durante la simulación.

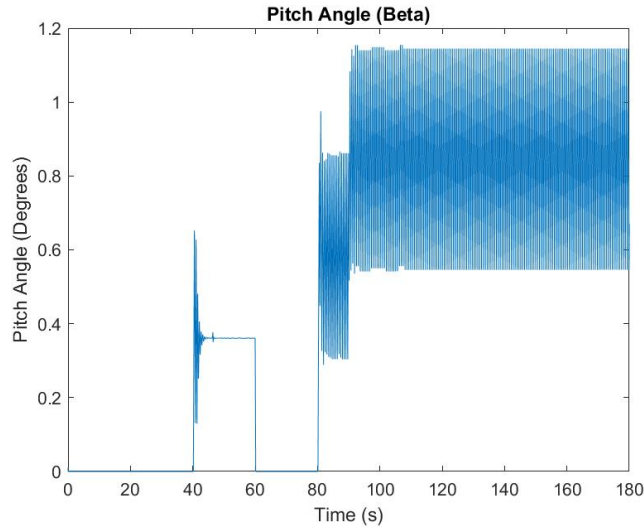


Figura 7: Valor del ángulo de ataque β durante la simulación

2.2. Algoritmo Temple Simulado

2.2.1. Desarrollo y testeo del Algoritmo

Para cumplir con el objetivo de optimizar la producción de energía, se utiliza el Algoritmo Temple Simulado para maximizar el valor del coeficiente de potencia C_p (ver sección 2.1.1). Dicho algoritmo es un algoritmo de búsqueda local, es decir que evalúa y modifica un conjunto reducido de estados (se mueve entre estados "vecinos"), este tipo de algoritmos permite manejar espacios de estados muy grandes o incluso infinitos, lo cual resulta ideal para esta aplicación ya que se trata de un espacio de búsqueda continuo. El funcionamiento del algoritmo Temple Simulado se puede resumir como:

1. Se generan estados vecinos de forma aleatoria.
2. También de forma aleatoria, se selecciona uno de los estados vecinos.
3. Se calcula un valor de Energía tanto para el estado actual como para el seleccionado en el paso 2. En este caso particular, el valor de energía está dado por el coeficiente C_p , calculado con la ecuación (2).

4. Si el valor de Energía del estado vecino es mayor que la del estado actual, entonces es aceptado como el nuevo estado del sistema.
5. Si el valor de Energía del estado vecino es peor que la del estado actual, se acepta con una probabilidad:

$$e^{\frac{\delta E}{T}} \quad (5)$$

Donde T es la "temperatura" del algoritmo. Este valor T decrece en cada iteración, de esta manera al principio se tiene una alta probabilidad de aceptar un estado peor y, al final, esta probabilidad es muy baja. Esto permite que el algoritmo escape de máximos locales en el inicio y limita la oscilación hacia el final para poder converger a una solución. Tanto el valor inicial de T como la velocidad de decrecimiento de la misma son hiperparámetros del algoritmo, es decir, deben ser seteados para que su funcionamiento sea el correcto.

6. Finalmente, cuando se cumple la condición de convergencia (cuando T es menor a un valor definido), se conserva el último estado encontrado.

El valor de λ depende de la velocidad de rotación (que se mantiene constante) y de la velocidad del viento, por ello λ no puede ser modificado. Esto quiere decir que el Algoritmo debe modificar el valor de β para buscar el valor óptimo de C_p , es decir, los estados vecinos descriptos anteriormente son nuevos valores para el ángulo de ataque del sistema.

Luego, este algoritmo se implementa escribiendo dos funciones en matlab:

1. `s_annealing()`: donde se realiza todo el algoritmo explicado anteriormente
2. `fitness()`: se realiza el calculo de la función de energía.

El código perteneciente a ambas se puede encontrar en el repositorio del proyecto [5].

Finalmente, se procede a testear el algoritmo construido antes de incorporarlo al modelo de la planta. Para ello se ejecuta la función `s_annealing()` con 1000 combinaciones de λ y β . Luego se calcula los valores de C_p iniciales y los correspondientes a los nuevos valores de β calculados con el algoritmo de temple simulado, lo que arroja un aumento del 5.2% de los valores nuevos respecto de los iniciales, por lo tanto, el algoritmo optimiza el valor de C_p .

2.2.2. Incorporación del algoritmo al modelo

Ahora se procede a incorporar el algoritmo desarrollado al modelo de la Figura 4, para lo que se utiliza el bloque "Matlab Function" que permite incorporar funciones desarrolladas en un script a un modelo de Simulink. Finalmente se llega al modelo que se puede observar en la Figura 8:

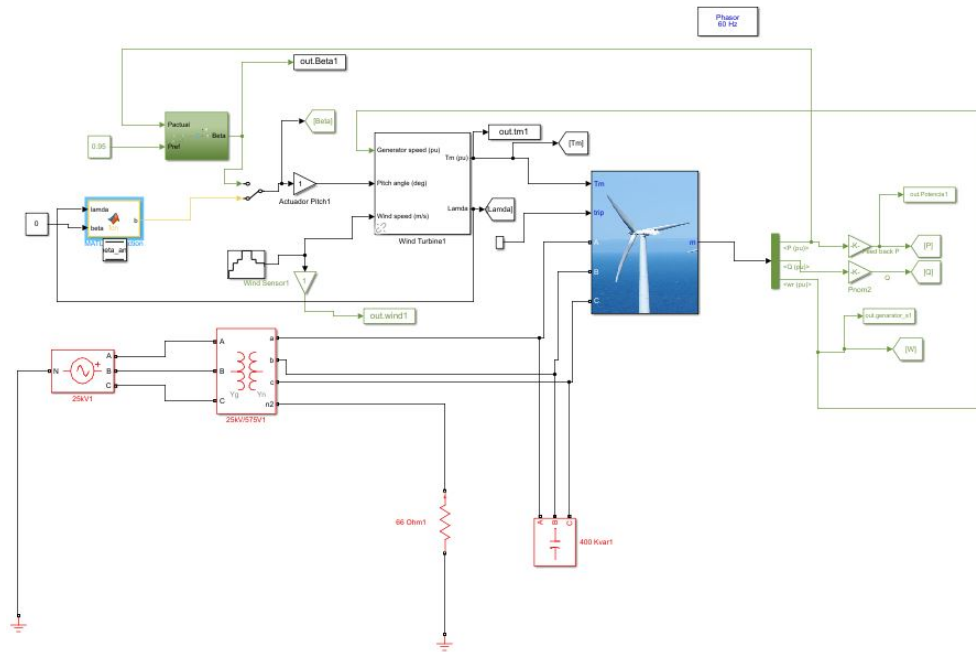


Figura 8: Modelo de la planta con Algoritmo Temple Simulado (Amarillo)

Se simula el modelo para las velocidades de viento de la Figura 9 y se obtienen los siguientes resultados:

1. Utilizando control PI: la potencia promedio entregada por el generador es de 6.14×10^5 W.
2. Utilizando algoritmo Algoritmo de Temple Simulado: la potencia promedio entregada por el generador es de 6.80×10^5 W.

Es decir que, al calcular el valor de β utilizando el algoritmo Temple Simulado se obtiene un aumento de la potencia promedio del 10.8 %. Esto puede parecer un número no tan alto, pero, al trabajar con un generador de alta potencia, este aumento representa una ganancia considerable en la cantidad de energía entregada a la red.

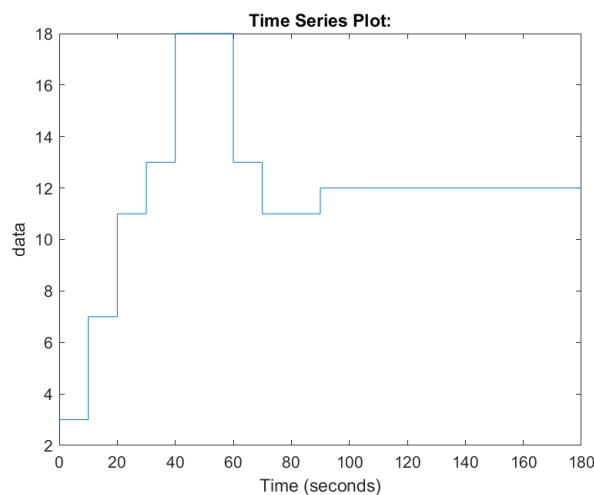


Figura 9: Velocidad del viento para simulación.

2.3. Control supervisor

En un sistema mecatrónico, por lo general, se encuentran dos tipos de control:

1. Control Supervisor: es un autómata discreto activado por evento, maneja los distintos modos de operación, diagnóstico de estado, monitoreo de fallas, etc.
2. Control a Lazo Cerrado: es un automata continuo, con implementación digital en tiempo discreto, que se encarga de controlar las variables del sistema como velocidad, torque, potencia, etc.

En esta aplicación se tiene dos tipos controles del tipo continuo para la potencia: control PI y control mediante Algoritmo de optimización. Para poder utilizar ambos controladores, se implementa un Control Supervisor que pasará a la turbina al estado que corresponda.

El automata discreto tendrá tres estados:

1. Estado OFF: estará en este estado cuando la velocidad del viento sea menor a 4 m/s y mayor a 26 m/s. Estas velocidades representan el valor mínimo para entregar energía a la red y el máximo que la turbina puede tolerar sin sufrir fallas.
2. Estado ON con control óptimo: estará en este estado cuando la velocidad del viento sea mayor que la mínima y menor que la velocidad para la cual se obtiene la potencia nominal en la salida.
3. Estado ON con control PI: estará en este estado cuando la velocidad del viento sea mayor que la necesaria para alcanzar la potencia nominal y menor que la máxima admitida por la turbina.

Para implementar este controlador discreto en el modelo de Simulink se utiliza la herramienta StateFlow. El modelo realizado se puede observar en la Figura 10.

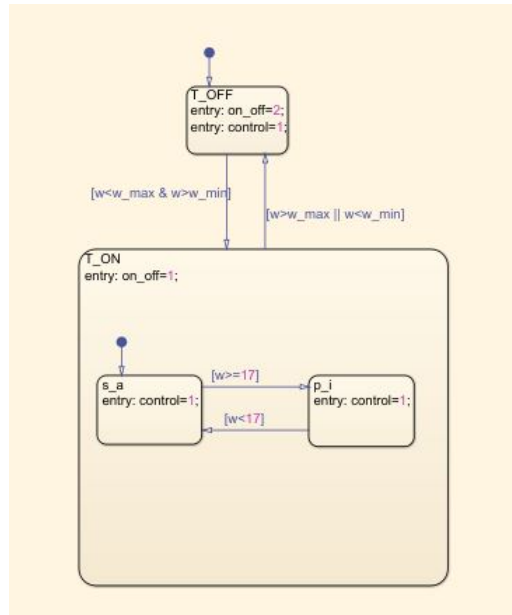


Figura 10: Control supervisor diseñado en StateFlow

Luego, este control supervisor se incorpora al modelo en simulink, el cual se puede observar en la Figura 11. Dicho modelo resulta el modelo final de este proyecto y se utiliza para simular el controlador diseñado con una señal de viento real y, así, poder analizar su desempeño.

Ahora, se toma la señal de viento medida a lo largo de un 3 días (12/03/18 al 15/03/18). Esta señal será la utilizada para realizar la simulación del sistema. En la Figura 12 se puede observar la distribución de la velocidad para este período. Además, en la Figura 13, se puede observar la señal de viento a lo largo del tiempo.

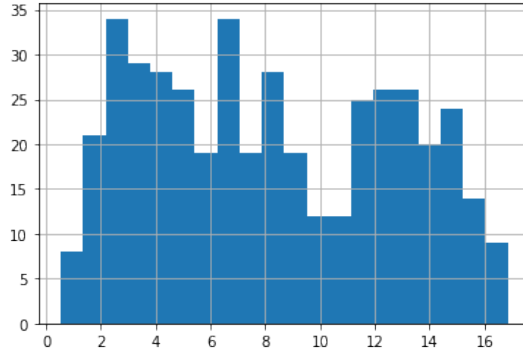


Figura 13: Distribución de la velocidad del viento a lo largo del período que se utiliza en simulación

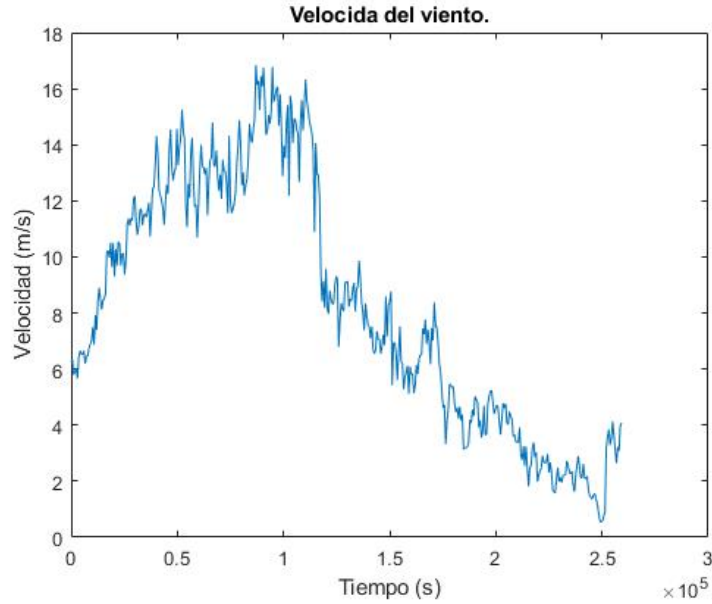


Figura 14: Velocidad del viento que se utiliza en simulación

3. Resultados

Se procede a simular el modelo completo (Figura 11) utilizando como señal de velocidad de viento la señal descrita en la sección 2.4 y que se puede observar en la Figura 14. Se puede observar en la Figura 15, la señal de control del ángulo de ataque β durante la simulación. Podemos ver como es continuamente modificada para optimizar la energía generada.

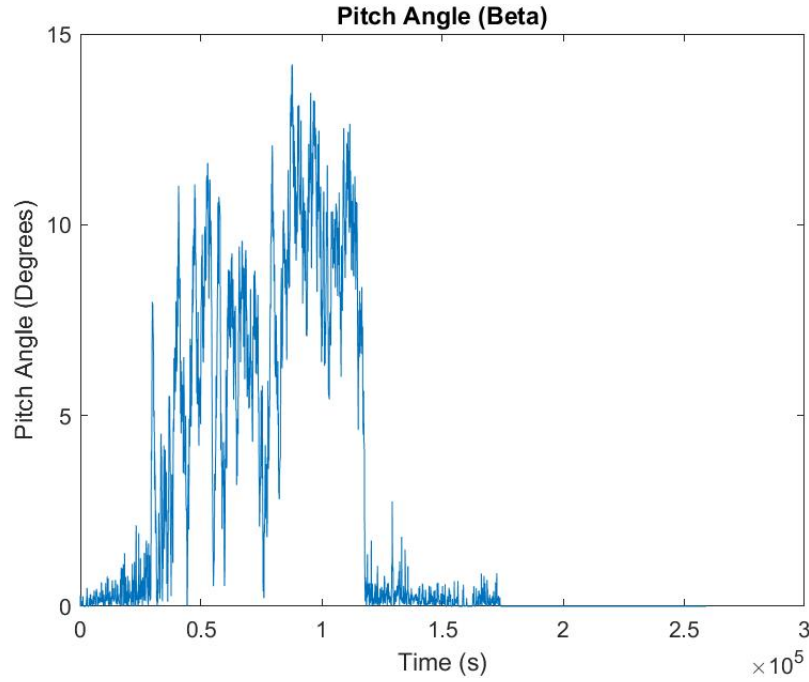


Figura 15: Ángulo de ataque durante simulación.

A continuación, en las Figuras 16 y 17 se pueden observar las señales de salidas del Controlador Supervisor: on/off y Temple Simulado/PI. En el caso de la señal on/off, se puede ver que cuando la velocidad del viento es menor que la mínima, el generador es desconectado (señal pasa al valor 2). Cuando la velocidad del viento es mayor que la máxima soportada por la estructura, el generador se coloca en posición de "bandera", en esta posición el plano de las aspas se encuentra paralelo a la componente horizontal de la velocidad del viento, lo que limita la fuerza que el viento realiza sobre la turbina.

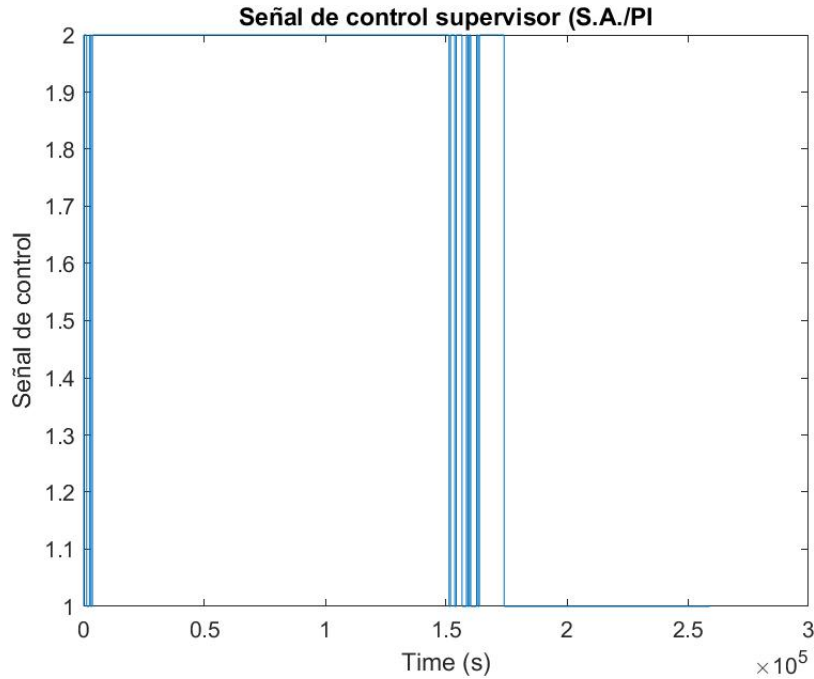


Figura 16: Señal de salida del Controlador Supervisor que permite modificar el control aplicado.

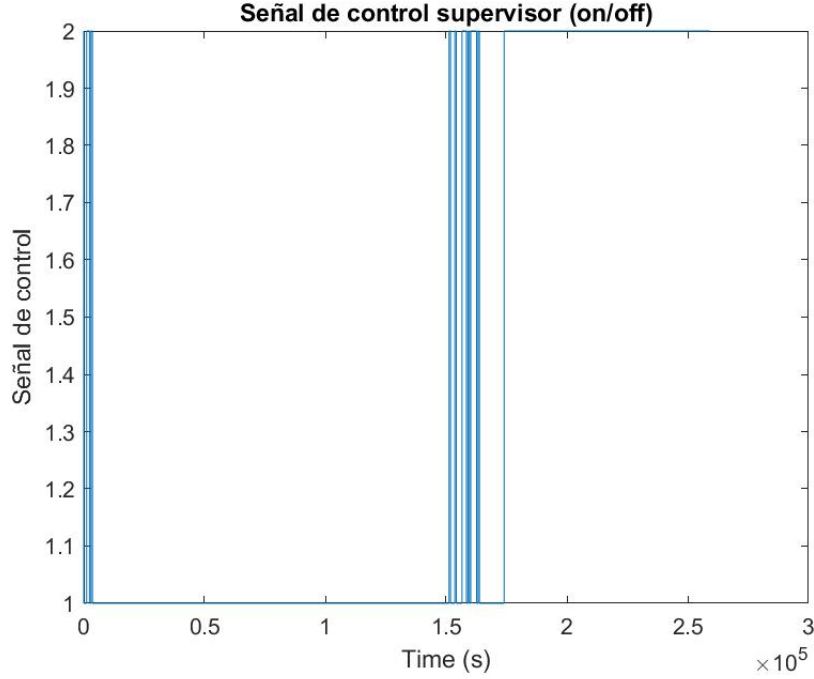


Figura 17: Señal de salida del Controlador Supervisor que permite modificar el control aplicado.

Finalmente, se puede observar la potencia generada con el Modelo Completo en la Figura 18.

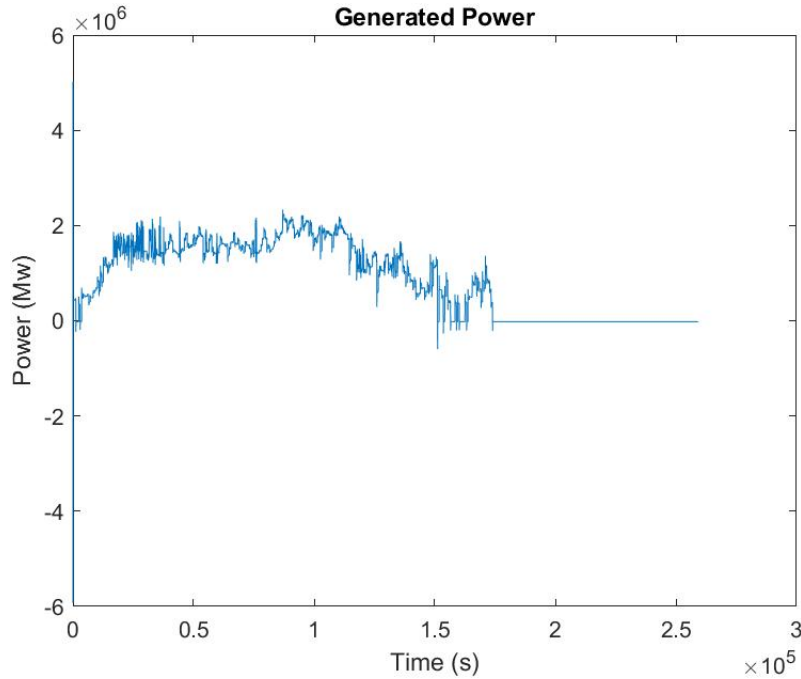


Figura 18: Potencia generada en el Modelo completo para señal de viento real.

Se puede observar que, efectivamente, cuando la velocidad de viento es menor que la mínima el generador es desconectado y, por lo tanto, no genera potencia. El resultado más importante es que la potencia promedio entregada por el generador para este controlador es de 1.2428 MW, mientras que si se utiliza solo el control PI, para la misma señal de viento, la potencia entregada es de 1.1068 MW. Esto representa una mejora del 12.28 % de la potencia promedio entregada por la turbina. Si se considera un consumo de 10 KW como normal para una casa, el aumento de potencia obtenido permite darle energía a 13.42 casas por cada turbina presente en el parque eólico. Por lo tanto, frente

a lo observado podemos afirmar que la implementación del algoritmo Temple Simulado produce un impacto considerable en el rendimiento de un parque eólico.

4. Conclusiones

El desarrollo del proyecto se llevó a cabo en dos etapas principales. Una primera etapa en la que se procedió a investigar sobre los distintos tipos de Turbinas Eólicas, los métodos de control que se aplican y los modelos matemáticos que rigen el funcionamiento de las mismas, para luego proceder a elaborar un modelo en Simulink con el objetivo de entender aún más su funcionamiento. En una segunda etapa, se procedió a investigar sobre distintos algoritmos de Inteligencia Artificial que permitan mejorar el rendimiento de la turbina modelada para finalmente implementar el algoritmo Temple Simulado dentro del controlador de Pitch de la misma.

Habiendo concluido las etapas anteriores, se puede decir que los objetivos planteados al inicio del proyecto fueron cumplidos, ya que, en primer lugar, se logró aumentar la potencia promedio generada en un 12.28 % respecto a la obtenida con un control PI clásico, por lo que se cumplió con el objetivo principal de optimizar el rendimiento. Para lograr este resultado, fue necesario aplicar todos los conocimientos aprendidos en la Cátedra de Control y Sistemas, lo que lleva a concluir que también se cumplió con el segundo objetivo del proyecto.

Por último, se espera que este proyecto sirva como punto de partida para que más estudiantes de Ingeniería en Mecatrónica se interesen e investiguen sobre las energías renovables, ya que estas resultarán fundamentales para el futuro de la humanidad.

5. Referencias

- [1] Global Wind Energy Council "Global Wind Report" <https://gwec.net/global-wind-report-2019/>
- [2] Renewable Energy Reports .Our World in Data" <https://ourworldindata.org/renewable-energy>
- [3] J.F. Manwell and J.G. Mcgowan and A.L. Rogers "Wind Energy Explained: Theory, Design and Application. Second Edition. "pp 79.
- [4] Wind Turbine Implementation "Matlab Documentation" <https://www.mathworks.com/help/physmod/sps/powers>
- [5] Repositorio de GitHub del Proyecto.
- [6] Set de datos con información de sistema SCADA de una Turbina Eólica. <https://www.kaggle.com/berkerisen/wind-turbine-scada-dataset>
- [7] Consumo medio de Electrodomésticos. <https://www.argentina.gob.ar/enre/uso-eficiente-y-seguro/consumo-basico-electrodomesticos>

6. Agradecimientos

1. Al Dr. Rodrigo Gonzalez, profesor de la cátedra de Control y Sistemas, por su disposición para el dictado de clases durante la Pandemia y por siempre contestar a todas las dudas planteadas.
2. Al Ing. Gabrien Julian, profesor de la cátedra de Automatas y Control Discreto, por aconsejarme la bibliografía utilizada.

7. Bibliografía

1. Fernando D. Bianchi and Hernan De Battista and Ricardo J. Mantz "Wind Turbine Control Systems: Principles, Modelling and Gaing Scheduling Desing".
2. S. Sumathi and L. Ashok Kumar and P. Surekha "Solar PV and Wind Energy Conversion Systems: an introduction to Theory, Modeling with MATLAB/SIMULINK, and Soft Computing Techniques.
3. Documentación de Matlab.
4. Videos del curso Wind Energy de la TECHNICAL UNIVERSITY OF DENMARK. Coursera.
5. Presentación de clase de la cátedra Inteligencia Artificial II, elaborados por el Dr. Martín Marchetta.