

Arquitectura Hardware Asequible para Implementar Controles de Pitch en una Turbina Offshore

Martín Fernández de Diego

Computer Sciences Faculty

University Complutense of Madrid

28040-Madrid, España

martif02@ucm.es

Belén Sánchez Centeno

Computer Sciences Faculty

University Complutense of Madrid

28040-Madrid, España

belsan05@ucm.es

Segundo Esteban San Román

Faculty of Physics

University Complutense of Madrid

28040-Madrid, España

sesteban@ucm.es

Matilde Santos

Institute of Knowledge Technology

University Complutense of Madrid

28040-Madrid, España

msantos@ucm.es

Abstract—This article contains a computationally affordable and feasible proposal to implement pitch control in floating offshore wind turbines. For this, a hardware architecture that allows wind turbines to be monitored and controlled remotely through the figure of the digital twin is designed and implemented. As an example, the development of a practical implementation in a prototype that handles the pitch angle and collects inertial measurement data is detailed. The result is an affordable, flexible and modular hardware architecture for offshore wind turbines.

Keywords—Wind energy, floating wind turbine, control, hardware, pitch.

Resumen—Este artículo recoge una propuesta computacionalmente asequible y viable para implementar control de *pitch* en aerogeneradores *offshore* flotantes. Para ello, se diseña e implementa una arquitectura hardware que permite monitorizar y controlar las turbinas eólicas de manera remota a través de la figura del gemelo digital. A modo de ejemplo, se detalla el desarrollo de una implementación factible en un prototipo que maneja el ángulo de *pitch* y recopila datos de medición inercial. El resultado es una arquitectura hardware asequible, flexible y modular para turbinas eólicas marinas.

Palabras clave—Energía eólica, turbina eólica flotante, control, hardware, pitch.

I. INTRODUCCIÓN

El principal desafío al que se enfrenta la humanidad hoy en día es lograr un planeta verde y sostenible. Las energías renovables juegan un papel fundamental para alcanzar este hito y el esfuerzo en innovación por parte de instituciones y empresas está siendo particularmente alto.

Las turbinas eólicas o aerogeneradores aprovechan la energía cinética del viento para proporcionar fuerza motriz a las palas que hacen girar, a través de un eje de transmisión, un generador eléctrico [1].

Dependiendo de la ubicación de instalación, existen dos categorías: las turbinas eólicas *onshore* y *offshore*. El término *onshore* hace referencia a las turbinas localizadas en tierra o terrestres, generalmente posicionadas lejos de lugares poblados y fuera de espacios protegidos; el término *offshore*, en cambio,

hace referencia a las posicionadas en el agua, generalmente en el mar. Dentro de estas se pueden distinguir las costeras, que están ancladas al lecho marino, y las turbinas flotantes, más recientemente desarrolladas, que se instalan en alta mar y son de gran capacidad de producción.

La energía eólica *offshore* creció casi un 30 por ciento anual durante la última década [2]. Este incremento, apoyado en la reciente inversión de la industria y en los avances de la tecnología, viene motivado por el gran tamaño que pueden tener los aerogeneradores *offshore* y porque la orografía no impone restricciones físicas que bloqueen las corrientes de viento, que son mucho más intensas y constantes. Sin embargo, el sector *offshore* no consiguió ser competitivo respecto a las fuentes de energía convencionales hasta 2017 [3].

Existen claras desventajas que han impedido su desarrollo. La principal es que los costes de distribución y mantenimiento son más elevados a causa de que las condiciones meteorológicas marítimas son mucho más agresivas. El fuerte viento, las olas y las corrientes deterioran más rápido los aerogeneradores.

Como norma general, las turbinas eólicas ancladas al suelo marino se consideran económica y técnicamente viables en áreas con una profundidad de agua inferior a 50 metros [4]. Debido a esta restricción, los obstáculos para aprovechar el viento están más presentes que en mar abierto y, muchas veces, las turbinas siguen localizadas cerca de núcleos urbanos, áreas de mayor actividad pesquera y hábitats naturales protegidos.

Pero sigue habiendo nuevos avances, sobre todo en las turbinas eólicas flotantes. Aunque esta tecnología *offshore* todavía no está muy desarrollada, tiene un potencial excelente, principalmente porque puede instalarse en aguas de entre 50 y 1000 metros de profundidad, desbloqueando sitios de aguas profundas inadecuados para la tecnología de cimentación fija que ha dominado hasta la fecha [4].

Las principales características de estos nuevos aerogeneradores son:

- Sencillez en la fabricación e instalación. Las turbinas y plataformas flotantes se pueden construir y montar en tierra y luego remolcar al sitio de instalación en alta mar.
- Eficiencia energética. Aprovecha los fuertes vientos que soplan en áreas de las aguas más profundas.
- Impacto ambiental potencialmente bajo.

Esta tecnología, enfocada a aguas profundas y entornos difíciles más alejados de la costa, está avanzando de forma constante hacia la viabilidad comercial con diferentes prototipos y proyectos a pequeña escala ya en funcionamiento. De hecho, en los últimos diez años, los programas de I+i de la UE concedieron alrededor de 496 millones de euros a la energía eólica marina, incluyendo la flotante [5].

Queda por determinar cuál de los distintos diseños que se están desarrollando es más eficiente y cómo construir cientos de plataformas de forma rápida y barata.

En este proyecto se propone trabajar con un prototipo a pequeña escala, implementando una arquitectura hardware que monitorice y pruebe algoritmos de control [6] en ambientes controlados.

El artículo queda estructurado como sigue. La arquitectura hardware objeto de este trabajo se resume en la sección II. Los componentes utilizados, la implementación funcional y los resultados obtenidos hasta el momento se describen en la sección III. La sección IV termina el artículo con las conclusiones y los trabajos futuros.

II. ARQUITECTURA HARDWARE

El sistema que se propone desarrollar consta de una arquitectura para monitorizar y controlar una turbina eólica *offshore* flotante con el fin de detectar cambios en su estado y anticiparse a los problemas. Es lo que se denomina detección temprana de averías, lo que disminuye los costes de mantenimiento y mejora la eficiencia del aerogenerador.

El escenario de trabajo se muestra en la Figura 1. Como se puede observar, el objetivo es la monitorización remota de los parámetros y variables que puedan ser críticos en la turbina flotante. Esto plantea importantes retos de comunicación y computacionales.

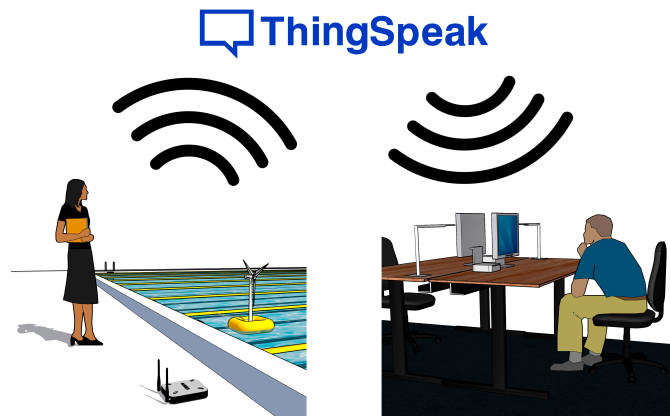


Fig. 1. Escenario de trabajo.

En este marco, por un lado, se aborda el objetivo a bajo nivel sobre el entorno hardware. Mediante sensores de medición inercial, como el acelerómetro y el giroscopio, se obtienen los datos necesarios para conocer el funcionamiento de la turbina y modificar el ángulo de rotación del servomotor que está unido al rotor. Es decir, se envían órdenes para el control del ángulo de pitch de las palas.

Por otro lado, también se incide en aspectos interesantes a alto nivel. Se van a realizar subidas periódicas a ThingSpeak de los datos recolectados para su posterior análisis y presentación. Esta visualización se materializa en un gemelo digital capaz de detectar patrones de comportamiento y de reaccionar de manera autónoma a las distintas eventualidades.

Para ello hará falta conectar el aerogenerador *offshore* a un punto de acceso a Internet cercano, que permita transferir las mediciones al servidor que nutre al gemelo digital y actualizar el modelo, y así mostrar información relevante al usuario.

III. IMPLEMENTACIÓN Y RESULTADOS

El sistema implementado se encarga de monitorizar los sensores de medición inercial y de controlar el *pitch* de las palas del aerogenerador con el fin de aumentar o reducir la resistencia con el viento y así controlar la potencia generada.

Para el montaje del mecanismo hardware se han utilizado componentes estándar, fácilmente reemplazables y muy económicos.

A. Componentes

- Microcontrolador ESP32 [7].
- Sensor de medición inercial (IMU) MPU6050 [8].
- Microservomotor SG92R [9].

1) *Microcontrolador ESP32*: Es un sistema en chip (SoC) que integra, entre otras características, Wi-Fi de 2.4 GHz y dual 32-bit CPU core.

2) *Sensor de medición inercial (IMU) MPU6050*: Es un sistema de 6 ejes de rastreo del movimiento que combina los 3 ejes del acelerómetro, los 3 del giroscopio y un procesador digital de movimiento (DMP) en un mismo componente.

3) *Microservomotor SG92R*: Es un motor que permite un control preciso en términos de posición angular y, por tanto, en términos de aceleración y velocidad. Su pequeño tamaño lo hace especialmente útil para su uso en prototipos.

B. Esquema de conexiones

1) *Microcontrolador ESP32*: Está alimentado vía nano USB con un ordenador o con un transformador de 5V.

2) *Sensor de medición inercial (IMU) MPU6050*: Está alimentado con los 3.3V de la placa ESP32 y conectado a la masa común. Sus pines System Data (SDA) y System Clock (SCL) están enlazados con los respectivos pines de la ESP32.

3) *Microservomotor SG92R*: Está alimentado con los 5V de la placa y conectado a la masa común. Su pin de control debe estar enlazado con un pin compatible con Pulse Width Modulation (PWM).

En la Figura 2 se puede ver el esquema de conexiones implementado.

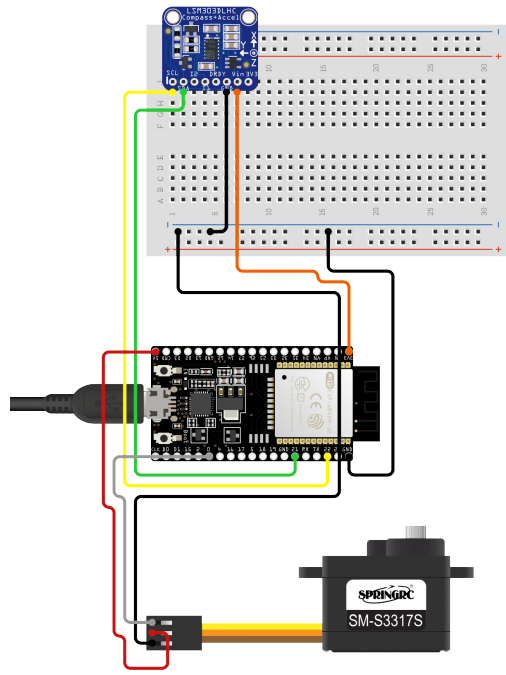


Fig. 2. Esquema de conexiones.

C. Entorno de desarrollo

La implementación se ha realizado en el entorno de desarrollo Arduino IDE, que permite compilar y grabar el código de manera sencilla en el firmware NodeMCU de la ESP32.

D. Desarrollo software

El código está escrito en lenguaje de alto nivel Arduino. En la Figura 3 se presenta un diagrama que muestra el flujo general de la ejecución del programa.

El flujo de instrucciones comienza enlazando el microcontrolador a la red WiFi con las credenciales preestablecidas en las constantes del programa. Una vez conectado, comenzará a realizar las tareas del lazo de control.

En este momento, se crean dos tareas asignadas a diferentes procesadores o cores. La tarea de control asignada al core 0 es la encargada de realizar la lectura de los valores de la unidad de medición inercial (IMU), de mover el microservomotor tras una pequeña transformación de datos y, por último, de enviar los datos previos a la transformación a la tarea encargada de subir la recopilación a ThingSpeak. La tarea de subida a ThingSpeak asignada al core 1 es la encargada de acumular la recepción de datos de la primera tarea y subirlos a ThingSpeak una vez transcurrido el tiempo de espera impuesto por el servidor.

Dividir el flujo en dos tareas o hilos de ejecución evita que al ejecutar el programa de manera secuencial el microservomotor deje de reaccionar a los cambios de la IMU durante unos segundos mientras la ESP32 sube los datos al servidor de ThingSpeak.

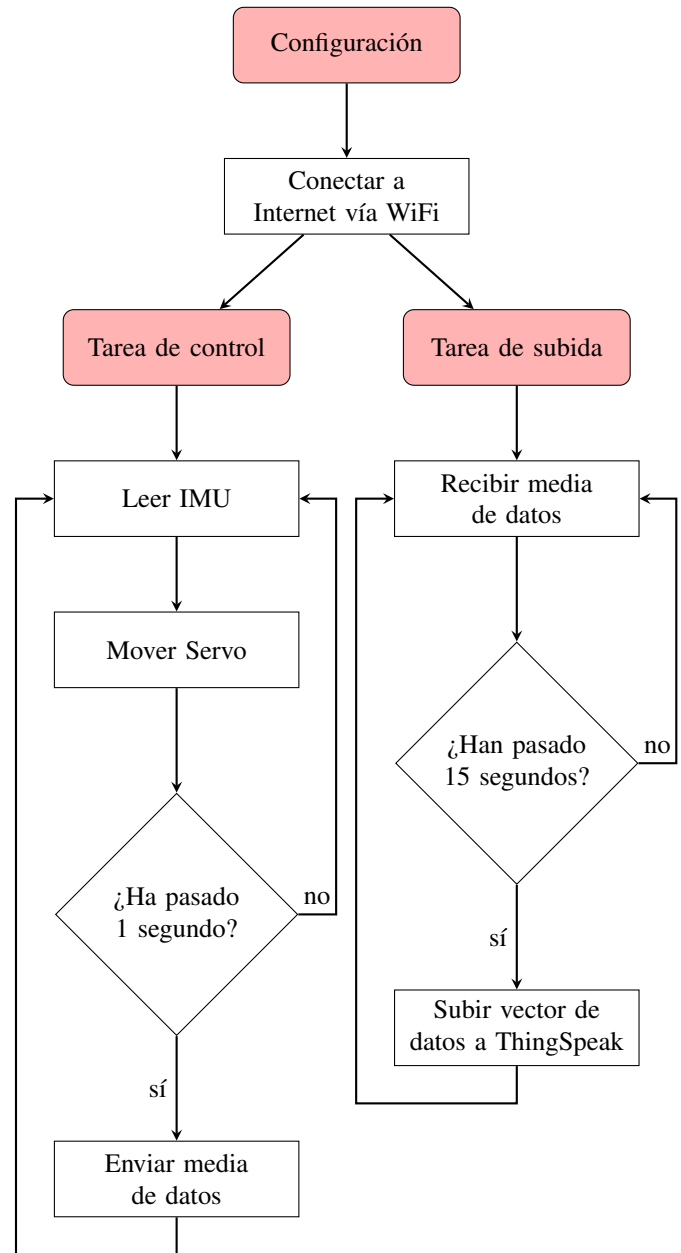


Fig. 3. Diagrama de flujo.

1) *Tarea de control:* Comienza realizando una medida en crudo (RAW) de los valores del acelerómetro y del giroscopio de la IMU MPU6050 a través de los pines I2C de comunicación que nos proporciona el microcontrolador ESP32.

Se realiza una transformación de los valores suministrados por el sensor del eje X/Y a valores comprendidos entre 0 y 180 grados para modificar la posición del microservomotor a través de los pines especializados PWM. De esta forma, se establece una relación entre la inclinación sobre un eje de la IMU con el grado de rotación del servomotor.

Conviene en este punto destacar varias observaciones:

- El eje de referencia se establece en función de la orientación de la IMU sobre el prototipo final.

- El rango de giro del microservomotor puede verse alterado para adecuarse a las limitaciones del mecanismo de la pala del aerogenerador.

Para enviar los datos a la tarea de subida, se realiza la media aritmética de las mediciones que se han tomado durante el último segundo y se empaquetan junto con la hora actual en formato EPOCH. Para conocer la hora EPOCH, el sistema accede al servidor de consultas libre y colaborativo pool.ntp.org.

2) *Tarea de subida*: Comienza leyendo los 15 datos más antiguos de la cola y guardándolos en un vector. Recopila los datos equivalentes a 15 segundos de lectura de la IMU. Si en algún momento la cola estuviera vacía y no le llegaran nuevos datos, esta tarea se quedaría bloqueada a la espera de recepción.

A continuación, estos datos se convierten a formato JSON para que el servidor de almacenamiento sea capaz de cargarlos e interpretarlos.

Por último, los datos son subidos a ThingSpeak vía HTTP POST. Esta fase consume unos segundos y la ejecución secuencial del hilo queda bloqueada hasta que finaliza la carga en el servidor.

3) *Comunicación entra tareas*: La comunicación entre hilos debe realizarse atendiendo a los principios de programación concurrente y paralela para evitar pérdidas o corrupción de datos.

La estructura de comunicación se traduce, en este caso, en una cola *First In First Out* (FIFO) accesible por ambos procesos, como se observa en la Figura 4. El paquete con las mediciones de la tarea de control se introduce en una cola y, más tarde, es recogido por la tarea de subida de datos.

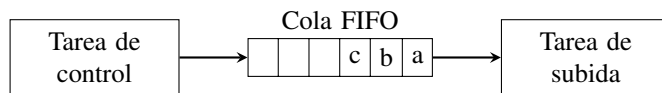


Fig. 4. Esquema de funcionamiento.

Con esta estructura, conseguimos:

- Que la tarea de control no tenga que esperar para enviar los datos y, por tanto, no vea interrumpida su función de mantener la fluidez en la relación IMU-microservomotor.
- Que la tarea de subida no pierda los datos que recibe cuando está realizando otras tareas que no sean de lectura (p.e. subiendo la recopilación anterior a ThingSpeak).

4) *Servidor de ThingSpeak*: El servicio gratuito del servidor de ThingSpeak restringe las subidas de modo que solo se puedan realizar cada 15 segundos.

Para crear un historial de datos suficientemente rico, como la precisión del intérprete está limitada a segundos, la recopilación que se sube cada 15 segundos consta de 15 datos.

Cada paquete subido en formato JSON contiene: una clave API de escritura de un canal de ThingSpeak y una lista de datos, cada uno con un tiempo EPOCH diferente y 6 campos que corresponden con la información de la IMU.

E. Resultados

Los resultados obtenidos proporcionan una arquitectura hardware que implementa un algoritmo de control del *pitch* en función de la inclinación. Incorpora además subida en tiempo real de las medidas inerciales a un servidor de Internet, desde donde son fácilmente accesibles para su análisis.

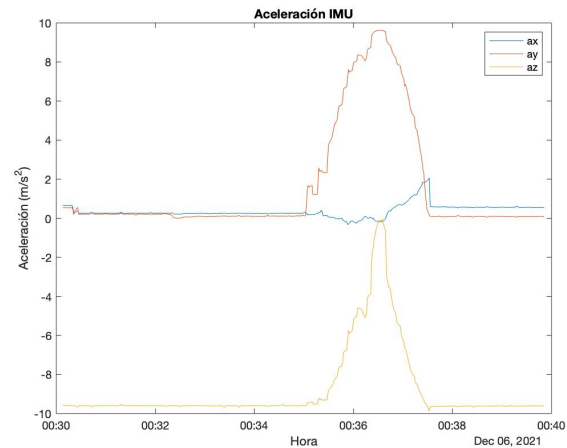


Fig. 5. Gráfica generada con MATLAB.

La Figura 5 es un ejemplo de representación de datos. Consiste en una gráfica que muestra las medidas obtenidas del acelerómetro en los últimos 10 minutos.

IV. CONCLUSIONES Y TRABAJOS FUTUROS

Se ha comprobado que es posible diseñar una arquitectura hardware asequible, flexible y modular.

Los próximos pasos del desarrollo atajarán el problema de la cantidad de ruido que genera la IMU de bajo coste aplicando técnicas y filtros como el de Kalman.

El objetivo a continuación es implementar un modelo capaz de mostrar, almacenar e interpretar los datos del servidor. También debería ser capaz de modificar el estado del aerogenerador (desactivar su funcionamiento, modificar *pitch*, etc.) con el objetivo de poder mantener el control ante situaciones adversas.

El modelo se materializaría en un gemelo digital. Esta versión digital del aerogenerador *offshore* simularía lo que sucede con la versión original en tiempo real. Sería capaz de mostrar el histórico de estados en detalle, reconocer patrones o incluso predecir comportamientos recurrentes.

Como última etapa, sería deseable estudiar la implementación desarrollada en un prototipo de turbina eólica situada en una plataforma flotante sobre el agua (p.e. en una piscina). Así se comprobaría su utilidad para tomar mediciones de forma remota del estado del prototipo, para enviarle comandos automáticos y para mantener un registro de su actividad.

AGRADECIMIENTOS

Este trabajo de investigación ha sido desarrollado con financiación parcial del Proyecto MCI AEI/FEDER RTI2018-094902-B-C21, del Ministerio de Ciencia, Innovación y Universidades.

REFERENCIAS

- [1] M. Arshad, and B. C. O’Kelly, “Offshore wind-turbine structures: A review,” *Proceedings of the Institution of Civil Engineers – Energy*, vol. 166, p. 139, 2013.
- [2] S. Reed, “A New Weapon Against Climate Change May Float,” *The New York Times*, no. 4, para. 9, June 2020. Accessed on: Dec. 2, 2021. [Online]. Available: <https://www.nytimes.com/2020/06/04/climate/floating-windmills-fight-climate-change.html>
- [3] D. Milborrow, “At the tipping point: 2017 wind cost analysis,” *Wind-power Monthly*, para. 1, Feb. 2018. Accessed on: Dec. 2, 2021. [Online]. Available: <https://www.windpowermonthly.com/article/1455361/tipping-point-2017-wind-cost-analysis>
- [4] World Bank, “Going Global: Expanding Offshore Wind To Emerging Markets,” World Bank Group, Washington D.C., Oct. 31, 2019. Accessed on: Dec. 2, 2021. [Online]. Available: <http://documents.worldbank.org/curated/en/716891572457609829/Going-Global-Expanding-Offshore-Wind-To-Emerging-Markets>
- [5] Comisión Europea, “Una Estrategia de la UE para Aprovechar el Potencial de la Energía Renovable Marina para un Futuro Climáticamente Neutro,” Bruselas, 2020. [Online]. Available: <https://eur-lex.europa.eu/legal-content/ES/TXT/HTML/?uri=CELEX:52020DC0741&from=EN>
- [6] J. E. Sierra-García, and M. Santos, “Improving Wind Turbine Pitch Control by Effective Wind Neuro-Estimators,” *IEEE Access*, vol. 9, pp. 10413-10425, 2021.
- [7] Espressif Systems, “ESP32-WROOM-32”, 2021. Accessed on: Dec. 4, 2021. [Online]. Available: https://www.espressif.com/sites/default/files/documentation/esp32-wroom-32_datasheet_en.pdf
- [8] InvenSense Inc., “MPU-6000 and MPU-6050 Product Specification Revision 3.4”, PS-MPU-6000A-00 datasheet, Nov. 2010 [Revised Aug. 2013].
- [9] Torq Pro & Tower Pro, “SG92R – Tower Pro”, 2021. Accessed on: Dec. 4, 2021. [Online]. Available: <https://www.towerpro.com.tw/product/sg92r-7/>