



Departamento de Electrotecnia
Facultad de Ingeniería
Universidad Nacional de La Plata



Emulador de turbina para OWC

Barrios, Leonardo N.º 71278/2

Director: Ing. Anderson, Jorge
Codirector: Dr. Puleston, Paul

23 de marzo de 2024

Resumen

Este proyecto se centra en la programación y puesta en marcha de un emulador de turbinas diseñadas para sistemas de columna de agua oscilante (OWC) y su interfaz gráfica. El objetivo es reproducir en tiempo real las curvas de cupla-velocidad de estas turbinas utilizando un perfil de recurso seleccionado, que consiste en el flujo producido por las olas que pasa a través de la turbina.

El microcontrolador utilizado para la emulación es el DSP TMS320C28335, de Texas Instruments, que fue implementado previamente en un proyecto para emular curvas de turbinas eólicas, por lo que se aprovechó como punto de partida para este. Permite la emulación de las curvas deseadas en tiempo real al realizar los cálculos necesarios en un tiempo acotado. La señal de referencia proporcionada por el módulo de control varía de 0 a 20 [mA], siendo proporcional a la cupla calculada según las características del modelo seleccionado, la velocidad de rotación de la turbina y el perfil de ola dado.

Para la configuración del dispositivo, la interfaz de usuario se comunica mediante un USB, lo que permite seleccionar el modelo a emular, los rangos de operación, las características del banco de trabajo y el perfil de olas, entre otros parámetros. Además, ofrece la posibilidad de visualizar en tiempo real los resultados de la emulación para analizar el comportamiento del generador. También permite guardar los datos obtenidos de la emulación, incluyendo recurso, tiempo, cupla y velocidad, en un archivo de tipo CSV para su posterior análisis.

Agradecimientos

Quiero expresar mi más profundo agradecimiento a todas las personas que han sido parte fundamental de mi trayectoria académica y profesional. En primer lugar, a mi familia, por su inquebrantable apoyo, paciencia y comprensión en cada etapa de este proceso.

Agradezco también al Instituto LEICI, donde tuve el privilegio de conocer a mis directores, Jorge y Paul. Quienes me presentaron el mundo de las energías renovables y me guiaron hacia la energía marina. Su mentoría experta y dedicación fueron fundamentales para mi crecimiento y desarrollo profesional. Asimismo, no puedo pasar por alto el apoyo de Claus que su ayuda y compromiso fueron fundamentales para alcanzar mis objetivos y para resolver los problemas que surgieron durante este proyecto. También quiero expresar mi más sincero agradecimiento a la Comisión de Investigaciones Científicas (CIC) por otorgarme la beca de entrenamiento. Este apoyo fue esencial para poder llevar a cabo este proyecto y continuar con mi formación profesional en el campo de la energía marina.

A mis estimados profesores, les estoy profundamente agradecido por su dedicación, conocimiento y pasión por la enseñanza. Y a mis compañeros y amigos, quienes compartieron conmigo desafíos y momentos inolvidables a lo largo de estos años. Gracias a ellos que me brindaron siempre su ayuda durante estos años. Espero haberles sido de ayuda tanto como ellos lo fueron para mí.

Índice general

1	Energía marina	4
1.1	Introducción	4
1.2	Extracción de energía marina	5
2	Extracción de energía marina	9
2.1	Modelo OWC	9
2.1.1	Mecánicas del agua	9
2.1.2	Turbina autorrectificadora y cámara OWC	10
2.2	Modelado de curvas	11
2.2.1	Ajuste por modelo de Gauss	11
2.2.2	Ajuste por modelo racional	12
2.2.3	Comparación de ajustes	14
3	Emulador	17
3.1	Banco de emulación	17
3.2	Módulo de control de la máquina impulsora	20
3.2.1	Unidad de procesamiento	20
3.2.2	Timer	21
3.2.3	PIE	22
3.2.4	Módulo de Encoder de Pulsos de cuadratura (eQEP)	23
3.2.5	Módulo de interfaz de comunicación serial (SPI)	23
3.2.6	Módulo de interfaz de comunicación serial (SCI)	24
3.3	Generación de señal de referencia en el DAC	24
3.4	Cálculo de cupla motora	25
3.5	Firmware del módulo de control	26
3.5.1	Estados del emulador y de la configuración	26
3.5.2	Comunicación serial	27
4	Interfaz de usuario	28
4.1	Protocolo de comunicación con el módulo de control	29
4.2	Gráficas	30
4.3	Lectura del recurso y guardado de los datos de emulación	30
4.4	Función de la interfaz durante la emulación	31
4.4.1	Hilo de comunicación para la emulación	31
5	Resultados experimentales	33
5.1	Prueba del microcontrolador con el DAC	33
5.2	Prueba del emulador completo	36
5.2.1	Recurso escalonado	37
5.2.2	Recurso sinusoidal	38
5.3	Análisis de resultados	40
6	Conclusiones y trabajos futuros	42
6.1	Conclusiones	42
6.2	Trabajos futuros	42
	Bibliografía	44

Capítulo 1

ENERGÍA MARINA

1.1. Introducción

En los últimos años la problemática sobre el agotamiento de las energías no renovables y su contaminación al medio ambiente es un tema de alta prioridad a tratar. Es por esto que se empezó a tener un mayor enfoque en las energías renovables [1]. En este sentido, la energía marina se presenta como una fuente de energía con un potencial significativo en varias formas, las cuales si se aprovecharan contribuirían a satisfacer una gran parte de los requerimientos energéticos mundiales. Dentro de estas formas de energía marina se incluyen la energía eólica offshore, de las mareas, de las corrientes por mareas, de las olas, térmica y gradiente salino.

Si bien la extracción de energía del mar aún está en sus primeras etapas, exceptuando alguna medida de la energía eólica offshore, los avances tecnológicos y las políticas estratégicas están induciendo a la utilización de los recursos energéticos marinos. Con respecto a la Argentina, la generación de energía eléctrica a partir de energía marina está en etapas muy prematuras todavía. Sin embargo, existen estudios sobre las posibilidades de su aprovechamiento a nivel nacional [2, 3].

Teniendo en cuenta el potencial de las energías renovables, la construcción de un emulador para sistemas de extracción de energía marina, como el OWC (Columna de Agua Oscilante), es muy útil para la investigación energética. Este emulador, diseñado para simular el funcionamiento de dispositivos que capturan energía del movimiento de las olas del mar, representa un avance significativo en la búsqueda de alternativas sostenibles y limpias para satisfacer las necesidades energéticas mundiales. Es importante destacar que este emulador se construye sobre la base de otro previamente desarrollado para turbinas de energía eólica [4]. Aunque los principios fundamentales pueden ser similares, adaptar el emulador para funcionar con sistemas marinos implica consideraciones específicas, como las características del comportamiento y variaciones del entorno marino. Además, también se deben tener en cuenta las características propias de las turbinas típicas utilizadas en estos sistemas OWC, como la turbina de Wells.

La utilización de este emulador permite a los investigadores y diseñadores probar y optimizar sus sistemas de extracción de energía marina en un entorno controlado y reproducible. Al simular el comportamiento de un OWC, el emulador ofrece la oportunidad de evaluar el rendimiento del sistema bajo diversas condiciones marinas y ajustar su diseño y configuración en consecuencia. Además, proporciona una plataforma para investigar y desarrollar estrategias de control eficientes que maximicen la captura y conversión de energía de las olas en electricidad utilizable. Con esta herramienta, se espera avanzar en la viabilidad y eficacia de los sistemas de energía marina, acercando la realidad de su implementación a nivel mundial.

1.2. Extracción de energía marina

El mar y el océano ofrecen diversas formas de energía, pero convertirla en electricidad plantea un desafío tecnológico significativo. Para abordar este desafío, se han desarrollado varios dispositivos de extracción de energía marina, cada uno diseñado para aprovechar diferentes aspectos de las olas y corrientes marinas. En el caso de las olas, se pueden aprovechar aspectos como el empuje de la ola por su frente, las variaciones de altura y la presión bajo la superficie de las propias ondas. Sin embargo, la transformación de esta energía en electricidad es el principal desafío a enfrentar. Estos dispositivos se pueden clasificar según el tipo de energía marina explotada, así como también según su ubicación en el océano para su funcionamiento: en la costa, cerca de la costa y mar adentro (onshore, near-shore y offshore, respectivamente).

En el contexto de los sistemas de extracción undimotrices, se encuentran diversas soluciones para aprovechar la energía marina, aunque todas comparten un proceso general. La mayoría de los dispositivos extractores siguen un esquema de conversión en dos etapas: la primera convierte la energía cinética de las olas en un movimiento mecánico oscilatorio, mientras que la segunda convierte este movimiento en energía eléctrica utilizable mediante un generador eléctrico [5].

Se nombran a continuación algunos dispositivos que trabajan con el problema y brindan soluciones de la conversión de la energía de las olas.

- *Pelamis (serpiente marina)*: Este sistema consiste en una serie de cilindros articulados y parcialmente sumergidos, alineados en paralelo a la dirección de la ola. Cada eje de movimiento cuenta con un generador asociado. La fuerza de la ola induce un movimiento relativo entre los cilindros, activando un sistema hidráulico que bombea aceite a alta presión a través de un conjunto de motores hidráulicos conectados a los generadores encargados de producir electricidad. Este tipo de sistema suele ser utilizado en zonas con condiciones marinas muy adversas. Se puede ver un esquema de su funcionamiento en la Figura 1.1, y una foto en la Figura 1.2

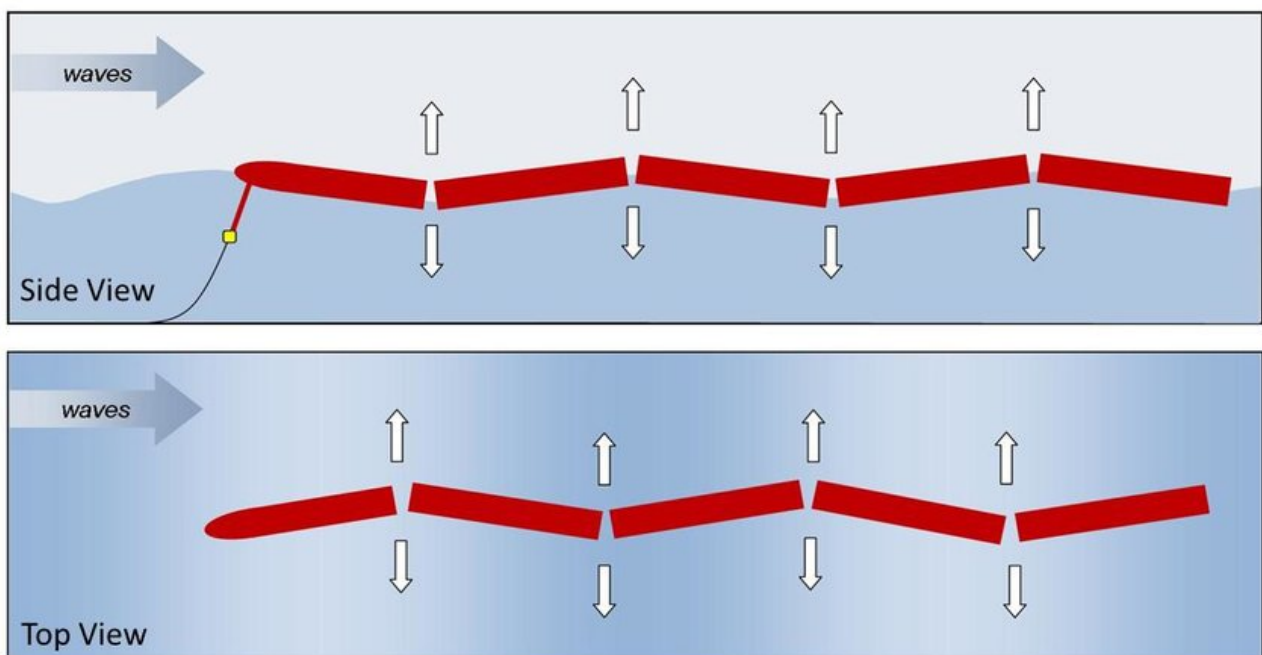


Figura 1.1: Una vista esquemática del funcionamiento del Pelamis.

1.2. EXTRACCIÓN DE ENERGÍA MARINA

- *Columpio de olas de Arquímedes (AWS: Archimedes Wave Swing)*: es un dispositivo totalmente sumergido entre los 40-100 metros de profundidad. Está formado por una estructura fija anclada al fondo oceánico con hormigón y una estructura móvil llena de aire que se mueve verticalmente por la acción de las olas. El movimiento entre la parte móvil o flotador y la parte fija hace que el aire que contiene el flotador se comprima para equilibrar las presiones. Este movimiento se transforma en electricidad por la acción de un sistema hidráulico y un conjunto motor-generator. Se puede ver un esquema de su funcionamiento en la Figura 1.4.

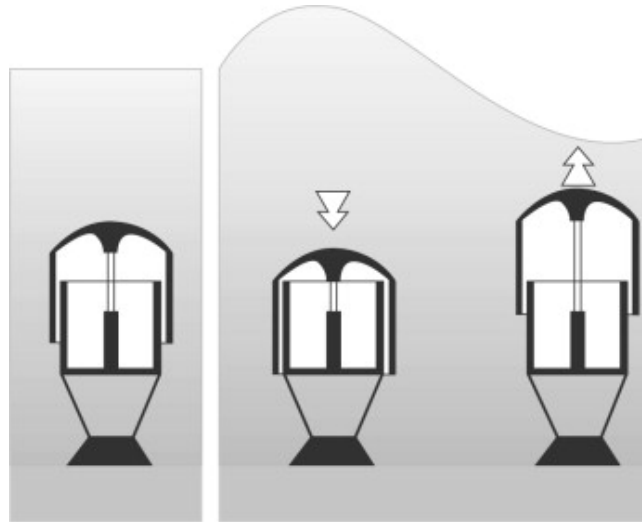


Figura 1.4: Columpio de Arquímedes.

- *Columna de agua oscilante (OWC: Oscillating water column)*: Este sistema consiste en una cámara que captura las olas, formada por una estructura fija con la parte inferior abierta por debajo del nivel del agua, lo que hace su funcionamiento muy similar al de la boya de tubo de cola o neumática. Las olas que entran en esta cámara comprimen y descomprimen el aire en su interior, el cual atraviesa un orificio, generando un flujo de aire oscilante. Este orificio está conectado a una turbina acoplada a un generador, convirtiendo así el movimiento en energía eléctrica. Se puede ver un esquema en la Figura 1.5.

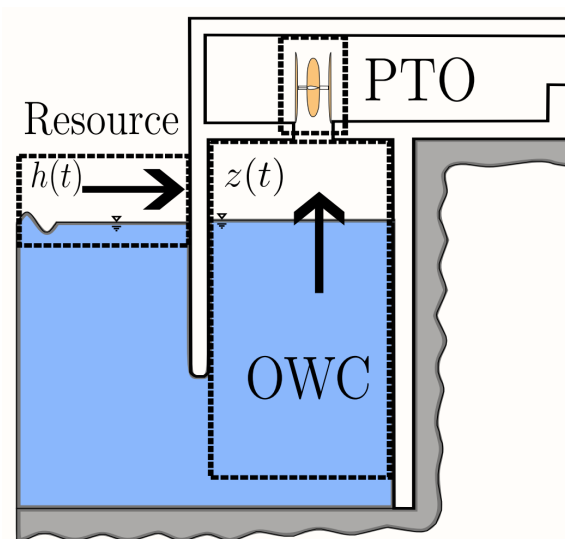


Figura 1.5: Esquema de funcionamiento del sistema OWC.

1.2. EXTRACCIÓN DE ENERGÍA MARINA

Estos sistemas de extracción de energía marina requieren control para lograr una operación eficiente, lo que implica la aplicación de técnicas de control. Generalmente, se emplean simulaciones o emulaciones para diseñar los algoritmos de control. Las simulaciones son útiles en las etapas iniciales del diseño, pero cuando se busca analizar el comportamiento real del generador, la mejor opción es llevar a cabo una emulación, donde se prueba físicamente.

Es por esto que se llevan a cabo pruebas en el generador en laboratorios con variables controladas. Para realizar estas pruebas de manera efectiva, se requiere un emulador que simule el funcionamiento del generador una vez instalado. Por esta razón, este trabajo se enfoca en la emulación del sistema de extracción de energía undimotriz mediante el dispositivo de columna de agua oscilante (OWC).

Capítulo 2

EXTRACCIÓN DE ENERGÍA MARINA

2.1. Modelo OWC

El sistema de aprovechamiento de energía marina de columna de agua oscilante, como se explicó previamente en la Sección 1.2, implica captar energía a través del flujo de aire generado por las oscilaciones del agua. Se instala una turbina en un conducto por donde fluye el aire oscilante generado por las olas, permitiendo así la extracción de energía mediante la rotación de la turbina. Para calcular las magnitudes del torque y el flujo de aire generados por las propias olas, se emplean ecuaciones descriptivas. A continuación se detallan estas ecuaciones.

2.1.1. Mecánicas del agua

La representación más básica de una ola es una senoide variable en la superficie del agua. Esta descripción puede definirse teniendo en cuenta la altura del agua, H , que es la distancia entre la cresta de la ola hasta el canal, y el período de la ola, T , que es el tiempo que se toma la ola para volver a pasar, como se puede ver en la Figura 2.1 de ejemplo.

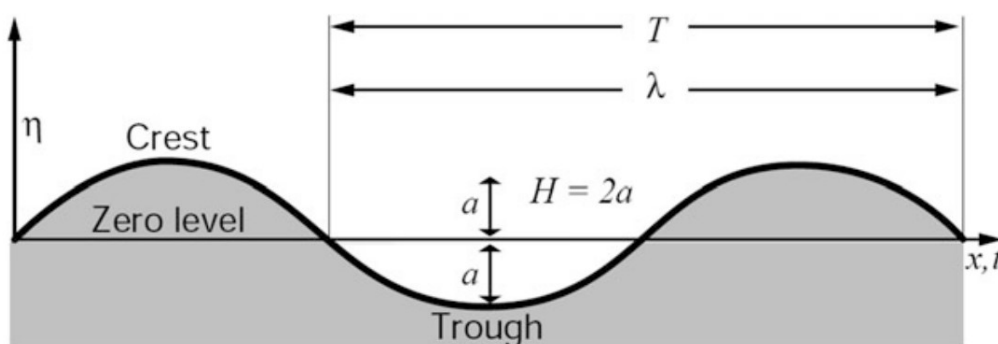


Figura 2.1: Ejemplo de ola sinusoidal con el valor de H y T definido.

Una forma más realista de representar las olas sería considerar estas variables como aleatorias, cambiando de altura y frecuencia constantemente a lo largo del tiempo. Para ello habría que describir el perfil de la ola de una manera más estocástica, representándola a través de su densidad espectral, $S(\omega)$. Entre algunas de estas representaciones se encuentran los espectros de Pierson-Moskowitz, JONSWAP y Brestchneider [6]. Dentro de este conjunto de espectros, el de Pierson-Moskowitz es semi-empírico, y su definición se puede ver en la siguiente ecuación [7].

$$S_{pm}(\omega) = \frac{0,11H_s^2T_z}{2\pi} \left(\frac{\omega T_z}{2\pi} \right)^{-5} e^{-0,44\left(\frac{\omega T_z}{2\pi}\right)^{-4}} \quad (2.1)$$

En el proceso de pruebas, se optará por emplear una referencia sinusoidal como recurso, ya que su simplicidad permite un análisis detallado del comportamiento del sistema con una predicción suficiente.

2.1.2. Turbina autorrectificadora y cámara OWC

Para poder convertir el flujo bidireccional (ν_t) en movimiento rotatorio unidireccional, se utilizan turbinas auto-rectificantes, de los cuales la mayoría son máquinas de flujo axial de dos tipos: turbinas de Wells [8] y turbinas de impulso. En este trabajo se hará el análisis teniendo en cuenta las turbinas de Wells, que están construidas a partir de palas escalonadas a 90° respecto del flujo, teniendo así un funcionamiento sencillo y buena fiabilidad [9].

Las ecuaciones para modelar el sistema son las siguientes [10]:

- Potencia disponible para el dispositivo:

$$P_{in} = q\Delta p [W] \quad (2.2)$$

donde $\Delta p [Pa]$ y $q [m^3/s]$ son la caída de presión y la tasa de flujo respectivamente.

- Tasa de flujo.

$$q = \nu_x a \quad (2.3)$$

donde $\nu_x = |\nu_t|$ es el flujo de aire unidireccional en dirección perpendicular al área de la pala, y $a [m^2]$ es el área del conducto de la turbina. ν_x es unidireccional debido a la propiedad auto-rectificante de la turbina.

- Caída de presión:

$$\Delta p = C_a(\phi)k\frac{1}{a}[\nu_x^2 + (r\Omega_r)^2] \quad (2.4)$$

donde $k = \rho bnl/2 [kg/m^3]$, $\rho [kg/m]$ es la densidad del aire, $b [m]$ es el largo de la pala, n es el número de palas y $l [m]$ es la longitud de la cuerda. $\Omega_r [rad/s]$ es la velocidad rotacional de la turbina, y $C_a(\phi)$ es el coeficiente de potencia (ver Figura 2.2), que depende de ϕ , el coeficiente de flujo. El valor de este coeficiente es particular de cada dispositivo.

- Coeficiente de flujo:

$$\phi = \frac{\nu_x}{r\Omega_r} \quad (2.5)$$

donde $r [m]$ es el radio de la pala. Notar la semejanza con la relación de velocidad periférica λ o TSR, propia de las turbinas eólicas.

- Torque de turbina:

$$T_t = C_t(\phi)kr[\nu_x^2 + (r\Omega)^2] [Nm] \quad (2.6)$$

donde $C_t(\phi)$ es el coeficiente de torque, que también depende de ϕ . Se puede ver una gráfica de un perfil típico de $C_t(\phi)$ en la Figura 2.2.

- Eficiencia de la turbina:

$$\eta = \frac{P_{out}}{P_{in}} = \frac{T_t\Omega_r}{q\Delta p} = \frac{C_t}{C_a\phi} \quad (2.7)$$

lo cual relaciona la potencia mecánica de la salida con la potencia disponible de la diferencia de presión y la tasa de flujo. Ver Figura 2.3

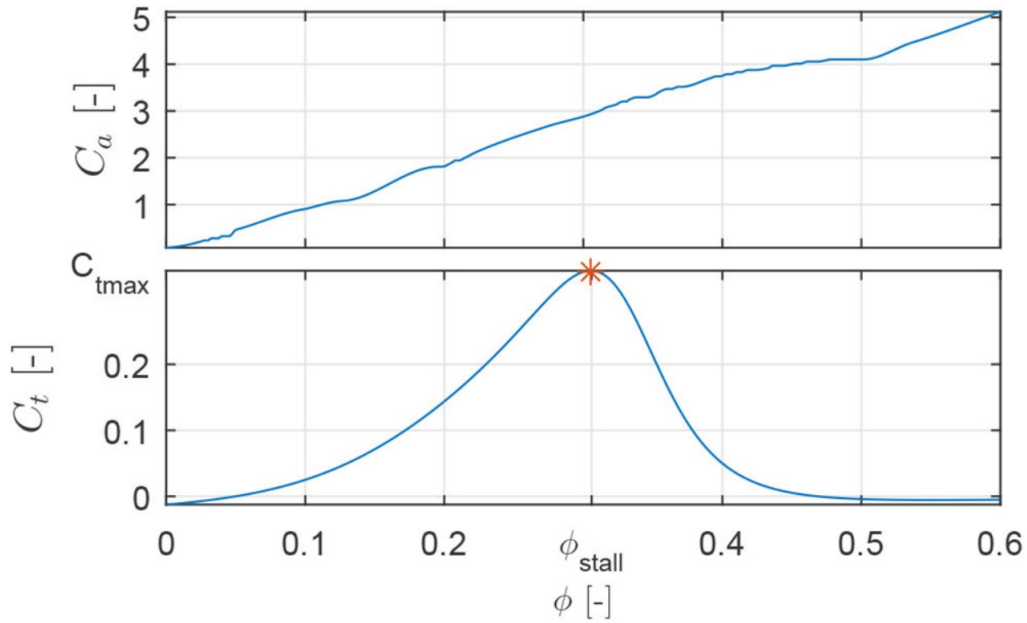


Figura 2.2: Superior: coeficiente de potencia, inferior: coeficiente de torque.

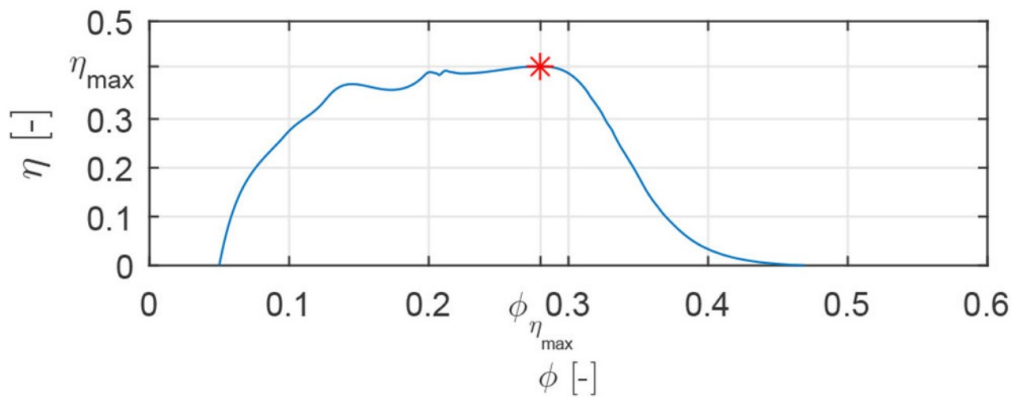


Figura 2.3: Eficiencia de la turbina de Wells.

2.2. Modelado de curvas

Para el cálculo de la cupla que se debe imponer utilizando la máquina impulsora (la cual está estrechamente relacionada con la corriente que se le entrega) se calcula el valor de este una vez que se tiene el dato del recurso y la velocidad, que teniendo en cuenta la Ecuación 2.6 es necesario caracterizar la curva $C_t(\phi)$.

Para ello se realiza un ajuste de la curva $C_t(\phi)$, en donde se pueden usar distintos tipos. Observando su forma en la Figura 2.4, se podría aplicar un ajuste por gaussiana o por alguna otra forma como la racional. En este trabajo se tuvieron en cuenta ambos ajustes y se realizaron comparaciones para seleccionar el más adecuado.

2.2.1. Ajuste por modelo de Gauss

El ajuste gaussiano es muy utilizado para distribuciones que presentan uno o varios picos. Por definición el modelo de Gauss viene dado por:

2.2. MODELADO DE CURVAS

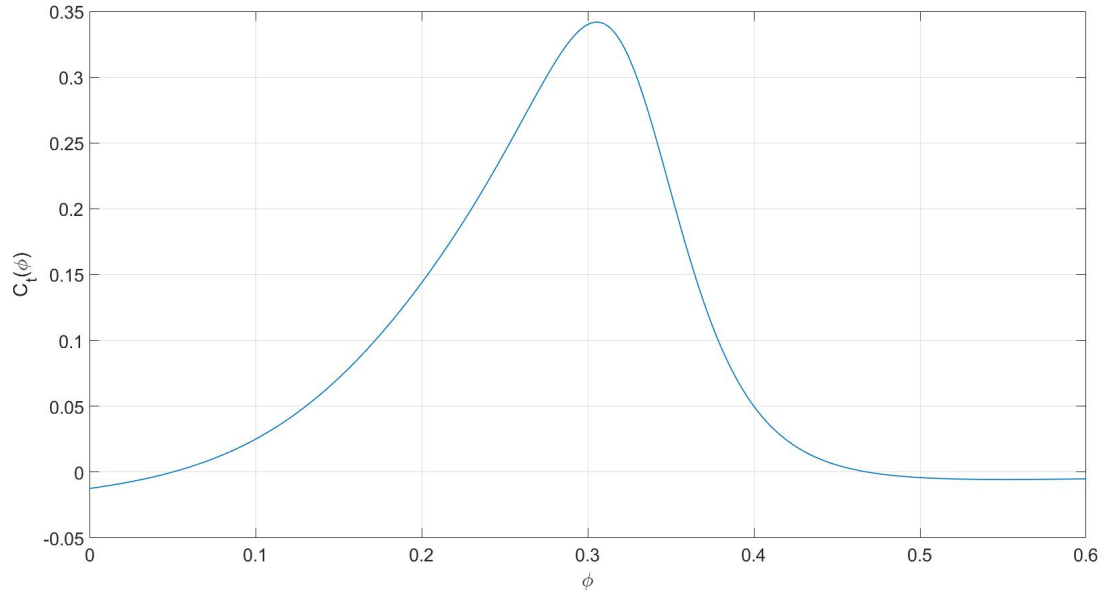


Figura 2.4: Curva $C_t(\phi)$, extraído por datos experimentales.

$$y = \sum_{i=1}^n a_i e^{-\left(\frac{x-b_i}{c_i}\right)^2} \quad (2.8)$$

donde a es la amplitud, b es el centroide (ubicación), c está relacionado a la anchura del pico y n es el número de picos que ajustar.

Observando la forma de la curva $C_t(\phi)$ en la Figura 2.4 se puede deducir que ajustando con al menos 2 picos por el modelo de Gauss, los cuales se superponen, se podría llegar a un buen ajuste. Considerando entonces $n = 2$ se tiene la ecuación:

$$C_{tG} = a_1 e^{-\left(\frac{x-b_1}{c_1}\right)^2} + a_2 e^{-\left(\frac{x-b_2}{c_2}\right)^2} \quad (2.9)$$

Utilizando la herramienta que ofrece Matlab para ajustar curvas, “*Curve Fitting*”, los valores de las variables calculados son:

$$\begin{aligned} a_1 &= 0,1996 \\ b_1 &= 0,3141 \\ c_1 &= 0,05651 \\ a_2 &= 0,1856 \\ b_2 &= 0,2541 \\ c_2 &= 0,1057 \end{aligned}$$

que describen la curva de la Figura 2.5.

2.2.2. Ajuste por modelo racional

Por otro lado, el ajuste por modelo racional, también conocido por ecuaciones racionales, son razones entre polinomios dados por:

2.2. MODELADO DE CURVAS

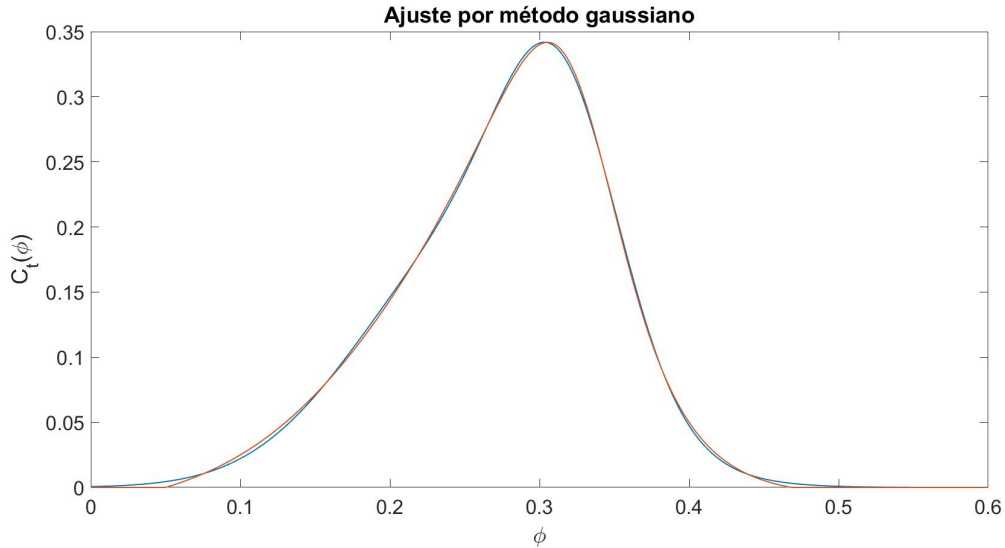


Figura 2.5: Comparación de la curva C_t original con el ajuste realizado por método de Gauss. Naranja: curva C_t original, azul: ajuste gaussiano de la curva C_t .

$$y_R = \frac{\sum_{i=1}^{n+1} p_i x^{n+1-i}}{x^m + \sum_{i=1}^m q_i x^{m-i}} \quad (2.10)$$

donde n es el grado del numerador, y m es el grado del denominador.

Utilizando la herramienta de Matlab, “*Curve Fitting*” se obtuvo que el mejor resultado con la menor cantidad de parámetros es teniendo un polinomio de grado 2 en el numerador, y uno de grado 4 en el denominador. Teniendo entonces la siguiente expresión:

$$C_{tR} = \frac{p_1 x^2 + p_2 x + p_3}{x^4 + q_1 x^3 + q_2 x^2 + q_3 x + q_4} \quad (2.11)$$

Agregando 2 parámetros adicionales de normalización para que los valores de estos sean más sencillos de manipular, desviación estandar, σ y media, μ , se considera $x = \left(\frac{\phi - \mu}{\sigma}\right)$.

Los valores obtenidos fueron:

$$\begin{aligned} p_1 &= -0,2096 \\ p_2 &= 0,1659 \\ p_3 &= 0,573 \\ q_1 &= -1,459 \\ q_2 &= 2,409 \\ q_3 &= -4,349 \\ q_4 &= 3,991 \\ \sigma &= 0,1159 \\ \mu &= 0,2 \end{aligned}$$

que describen la curva de la Figura 2.6.

2.2. MODELADO DE CURVAS

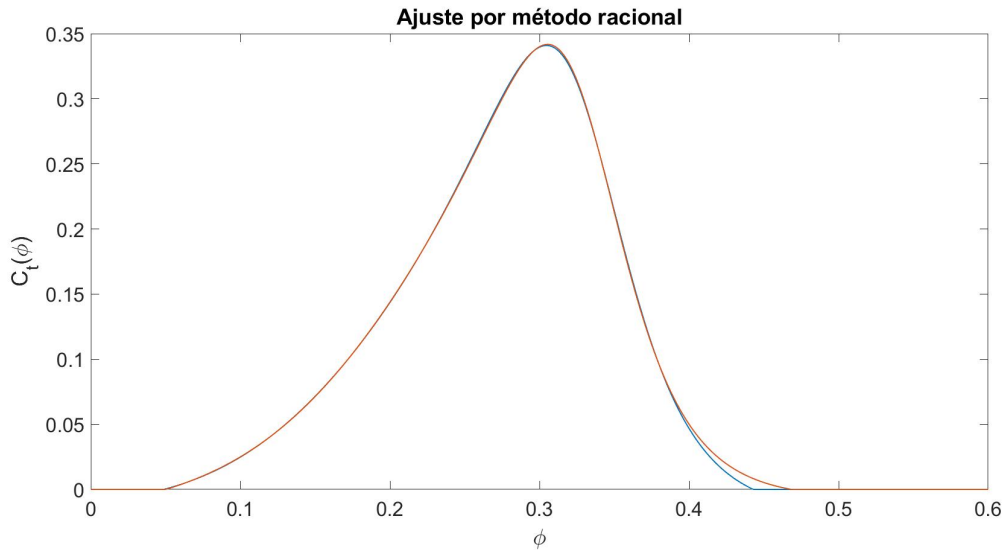


Figura 2.6: Comparación de la curva C_t original con el ajuste realizado por método racional. Naranja: curva C_t original, azul: ajuste racional de la curva C_t .

2.2.3. Comparación de ajustes

Para verificar y comparar el buen ajuste a la curva C_t de ambos se calculó la curva Cupla-Velocidad y la curva Potencia-Velocidad de la turbina con cada C_t obtenido, y se los comparó con el valor original utilizando la Ecuación 2.6. El resultado se puede ver en las Figuras 2.7 y 2.8.

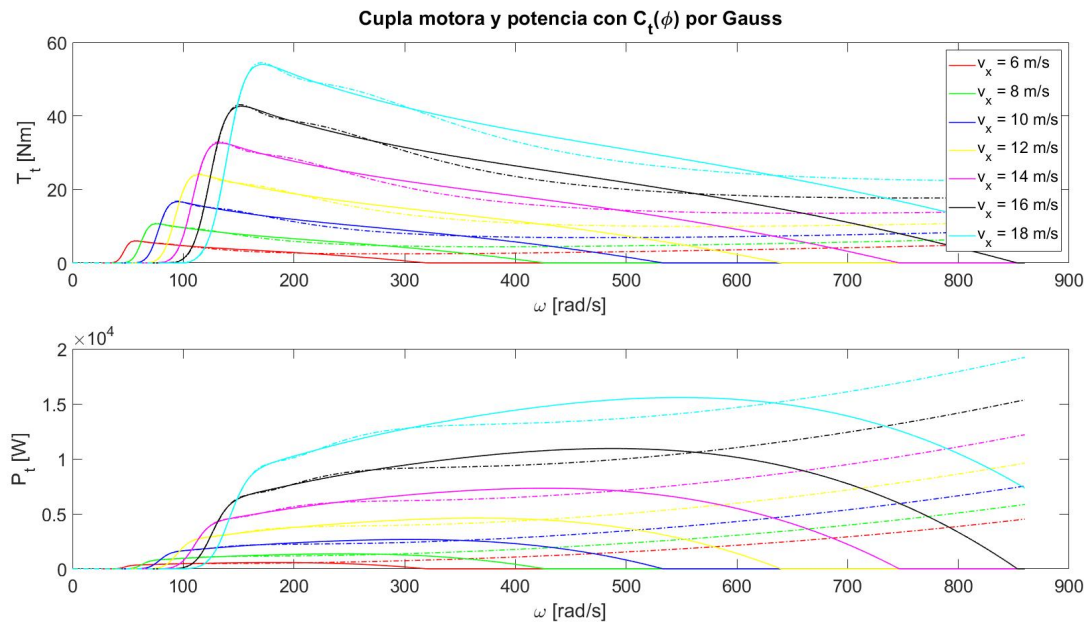


Figura 2.7: Comparación del torque y potencia extraída de la turbina para diferentes recursos calculando con el ajuste C_t gaussiano (línea punteada y discontinua) y C_t original (línea continua).

Considerando las curvas de torque de ambos ajustes, se observa que en la zona ascendente hay un bajo error, pero luego del pico se aprecia una diferencia considerable entre el ajuste gaussiano y el ajuste racional. Mientras que el ajuste gaussiano muestra un error significativo

2.2. MODELADO DE CURVAS

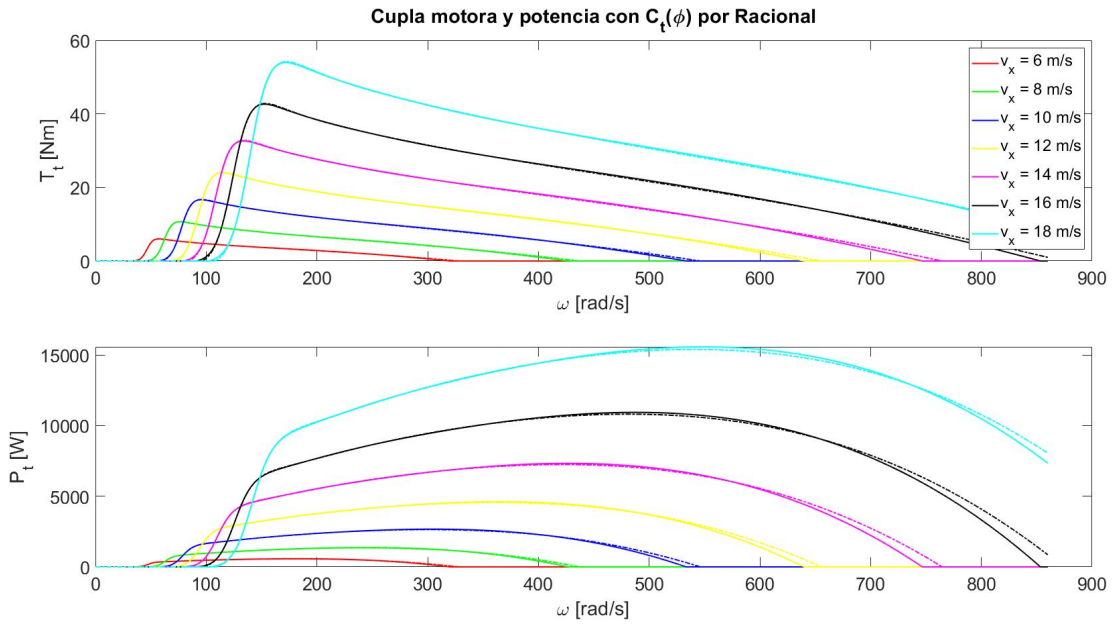


Figura 2.8: Comparación del torque y potencia extraída de la turbina para diferentes recursos calculando con el ajuste C_t racional (línea punteada y discontinua) y C_t original (línea continua).

en esta área, el ajuste racional presenta un error aceptable. La zona descendente de la curva es crítica, ya que es donde se logra la mayor extracción de potencia.

Teniendo en cuenta la zona de trabajo considerada por F.D. Mosquera et al. 2020 [11] para extraer la máxima potencia disponible, indicado como $T_{t_{opt}}$ en la Figura 2.9. La opción más adecuada entre estos dos ajustes es el modelo racional, ya que es el que mejor se ajusta a los valores de torque utilizados.

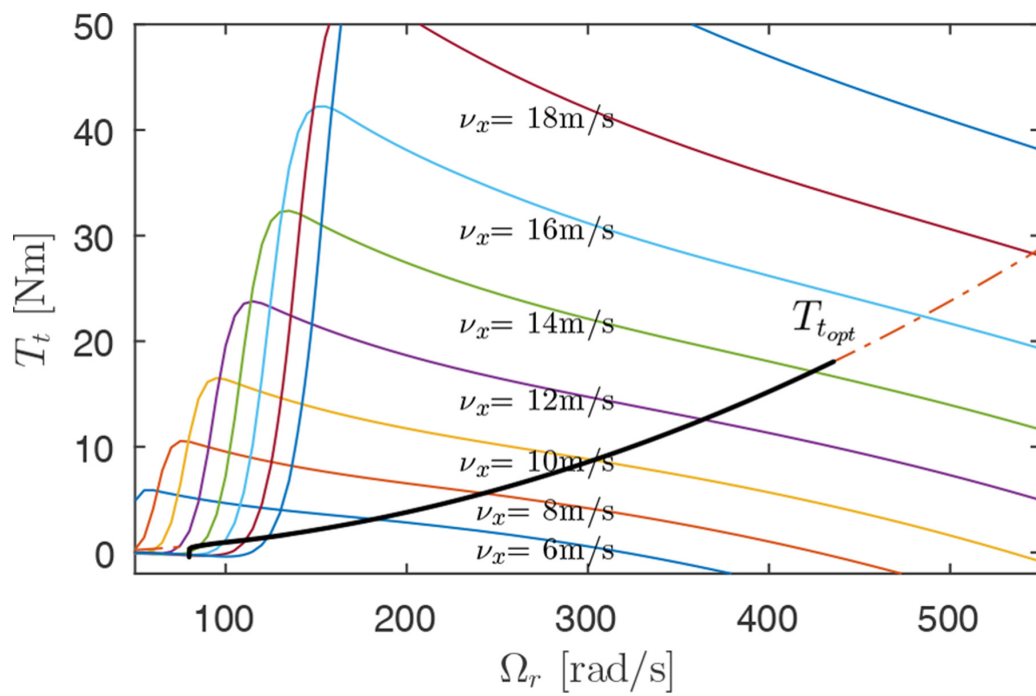


Figura 2.9: Curva característica del sistema (línea sólida negra) en el plano Cupla-Velocidad.
 Lugar de máxima extracción de energía de las olas (línea punteada y discontinua).
 Característica de la turbina para distintos recursos, ν_x (líneas sólidas coloreadas) [11]

Capítulo 3

EMULADOR

3.1. Banco de emulación

Un emulador de turbinas para sistemas OWC permite representar las características cupla-velocidad de un OWC real en el eje de una máquina eléctrica. De este modo es posible reproducir el funcionamiento de diferentes tipos de turbinas implementadas dentro de estos OWC. Así se podrían ensayar estrategias de control sobre el generador eléctrico asociado a la turbina, para diferentes condiciones difíciles de reproducir en un sistema OWC real.

En la Figura 3.1 se pueden ver las diferentes partes del emulador, y en la Figura 3.2 se muestra una fotografía.

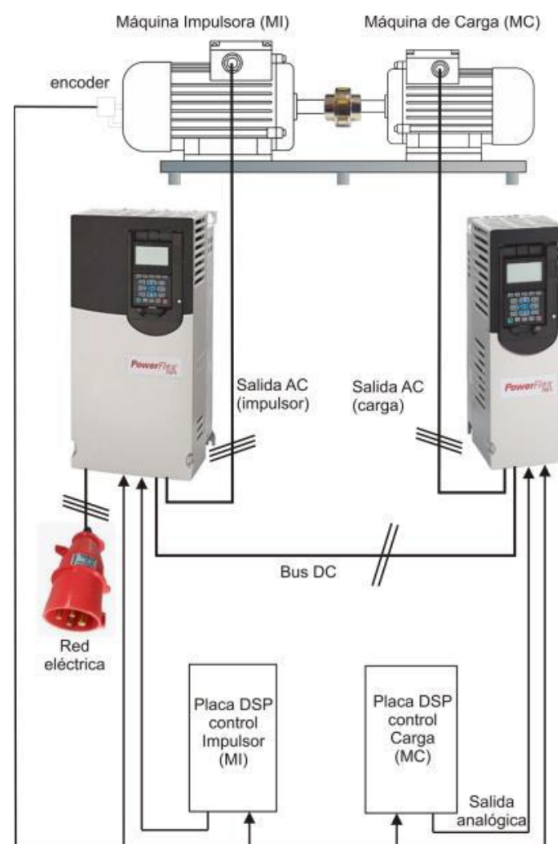


Figura 3.1: Esquemático del emulador de turbinas para OWC.

3.1. BANCO DE EMULACIÓN



Figura 3.2: Fotografía del emulador con sus partes.

En primer lugar, teniendo en cuenta la Figura 3.1, se observa el generador eléctrico o máquina de carga (MC) que se encuentra hacia la derecha, acoplado mecánicamente con la máquina impulsora (MI) a su izquierda. Por un lado la máquina impulsora es la encargada de imponer en su eje la cupla de la turbina emulada, de acuerdo a su velocidad de rotación y la curva característica cupla-velocidad. Por el otro, el generador absorbe la energía entregada por la máquina impulsora y la reinyecta a la red. Las máquinas impulsora y de carga son marca WEG de propósito general tipo W22-IE1-132M-cast iron (15HP) y W21-IE1-132M-aluminio (10HP), respectivamente. Ambas poseen dos pares de polos, siendo la velocidad nominal aproximada de ambas 1500RPM [4]. Se puede ver una imagen en la Figura 3.3



Figura 3.3: Máquinas acopladas mecánicamente. Izquierda, máquina de carga. Derecha, máquina de impulsión.

3.1. BANCO DE EMULACIÓN

Para la excitación de ambas máquinas se utilizan variadores de velocidad, de características adecuadas, de acuerdo a la máquina empleada (máquina de inducción, de alimentación trifásica, etc.). Los variadores de velocidad de las máquinas controlan las cuplas motoras impuestas por estas en su eje, según una referencia que es provista por el módulo de control de cada uno. Los que se utilizaron en este proyecto son Marca Allen-Bradley modelo PowerFlex 755 con corrientes de salida máximas de 30A y 22A para las máquinas impulsora (MI) y de carga (MC) respectivamente [4]. Se puede ver una fotografía en la Figura 3.4



Figura 3.4: Variadores de velocidad utilizados en el banco de emulación.

En cuanto a las variables medidas, la velocidad del eje se realiza mediante un encoder de 1000 pulsos por vuelta marca HEDSS modelo ISC5210-001G. La señal de velocidad es enviada a ambos variadores y a los sistemas de comando de los mismos, los cuales consisten en dos plataformas de cómputo independientes basadas en hardware DSP.

Finalmente, estos controladores dedicados se encargarán de generar la señal de referencia para cada uno de los convertidores. Dicha señal de referencia es calculada a partir de la velocidad de rotación de la máquina y el modelo de turbina emulado, en conjunto con los parámetros del banco de emulación, y el perfil de recursos, es decir el perfil de olas que es enviado a través de un puerto serie de una computadora hacia el microcontrolador. Ver Figura 3.5.

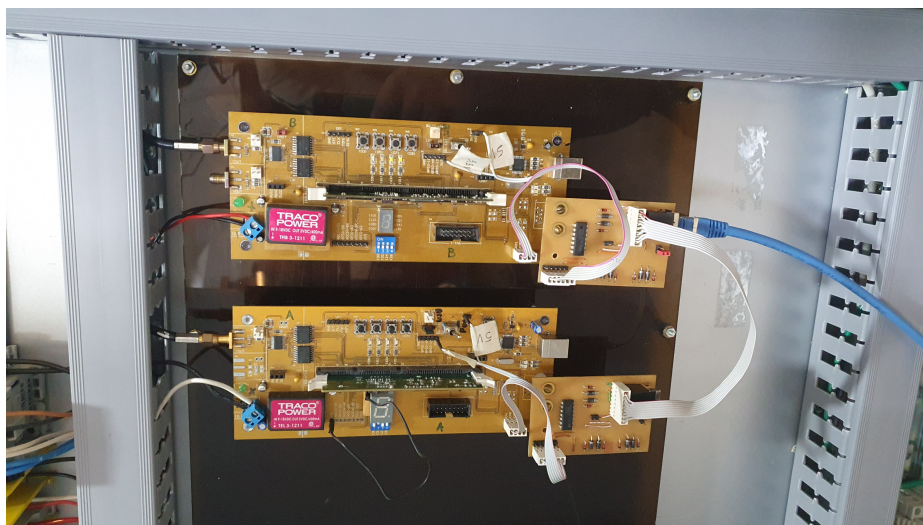


Figura 3.5: Variadores de velocidad utilizados en el banco de emulación.

3.2. MÓDULO DE CONTROL DE LA MÁQUINA IMPULSORA

Entonces, para la generación de la cupla de referencia, el módulo de control puede descomponerse básicamente en:

- Unidad de procesamiento.
- Medición de velocidad.
- Generación de señal de referencia.
- Recepción de recurso mediante la interfaz de usuario.
- Interfaz de usuario.

Dado que el banco emulador, o estructura dual para la emulación, ya está en funcionamiento en el laboratorio LEICI, será utilizado en este proyecto [4]. Entonces la parte del emulador que se desarrolla en este trabajo involucra el módulo de control de la máquina impulsora (tomando de partida lo realizado para la emulación de turbinas eólicas), la interfaz gráfica, y la comunicación entre ambos. A continuación se desarrollará sobre este módulo de control, y en el siguiente capítulo sobre la interfaz gráfica.

3.2. Módulo de control de la máquina impulsora

Este módulo es el que utilizó para el desarrollo de este proyecto. A continuación, en esta sección se describe su implementación, la cual es la misma que la ya hecha para el emulador de turbinas eólicas, pero con la implementación de la comunicación serial con la interfaz de usuario.

3.2.1. Unidad de procesamiento

Este módulo es responsable de llevar a cabo los cálculos según los modelos descritos en la sección 2.1.2. Su función abarca desde representar las curvas del modelo de turbina de Wells a emular hasta comunicarse con la interfaz gráfica. Además, se encarga del procesamiento de la información relacionada con la rotación y el viento.

Está conformado por un microcontrolador DSP F28335, de la familia TMS320C28x de Texas Instruments, que tiene una unidad de punto flotante para una mayor eficiencia en la realización de los cálculos, como también una optimización en los tiempos de implementación [12].

Algunas características generales del DSP utilizado son:

- Frecuencia de trabajo hasta 150MHz, y tensión de trabajo de 3.3V.
- Unidad de punto flotante IEEE-754 de simple precisión.
- Arquitectura Harvard.
- Memoria Flash de 256k x 16, y 34k x 16 de SARAM.
- Hasta 18 salidas para PWM.
- 2 módulos de interfaz para lectura de encoders de cuadratura.
- 3 Timers de 32 bits.
- Periféricos de comunicación serial: CAN, SCI (UART), SPI e I2C.

3.2. MÓDULO DE CONTROL DE LA MÁQUINA IMPULSORA

- Un conversor ADC de 12 bits con 16 canales.
- 88 puertos de propósito general (GPIO), con filtro de entrada programable.
- Soporte JTAG, para depuración y programación.

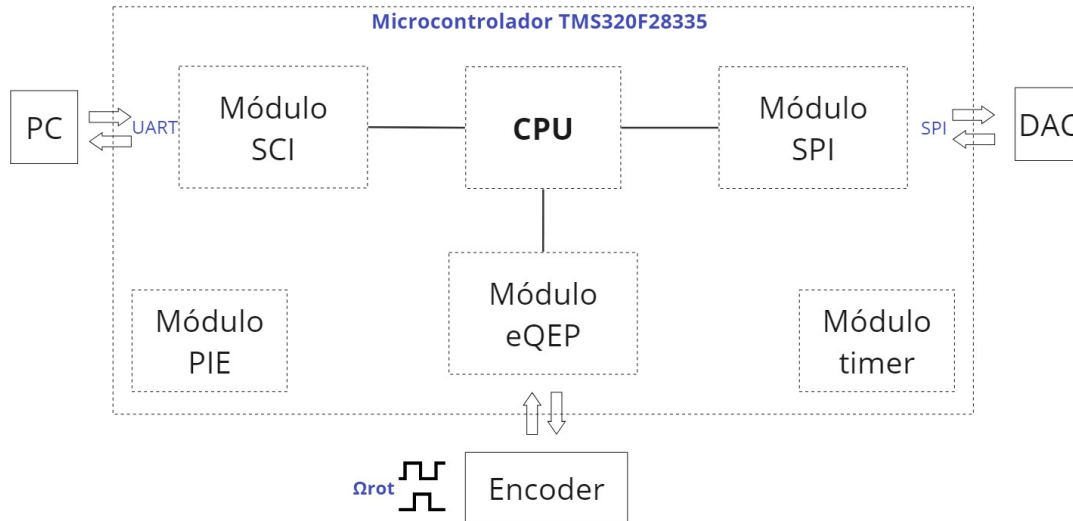


Figura 3.6: Diagrama del DSP con los módulos utilizados en la unidad de procesamiento.

En la Figura 3.6 se puede ver un diagrama del funcionamiento del microcontrolador los módulos utilizados.

La Unidad Central de Procesamiento (CPU) procesa toda la información, utilizando también dos módulos complementarios necesarios para el manejo de las tareas del sistema, el módulo temporizador y el módulo de interrupciones PIE. El temporizador es utilizado para administrar los tiempos del sistema, mientras que el módulo de interrupciones se utiliza para manejar las tareas de ejecución con sus prioridades.

El módulo eQEP es el encargado de interpretar la señal recibida del encoder y procesa la velocidad de rotación. Esta velocidad será utilizada por el mismo CPU para calcular los valores de cupla del modelo del emulador. Una vez obtenida la cupla que debe imponer la máquina impulsora en el eje debe ser transferido al DAC para que el valor llegue a los variadores de velocidad. Para ello se utiliza el módulo de la interfaz serial de periféricos, SPI. Para finalizar, la comunicación con la computadora se realiza mediante la interfaz comunicación serial, o SCI, para permitir la configuración del DSP y transferencia del recurso para cálculos de cupla con la interfaz de usuario.

3.2.2. Timer

El TMS320F28335 posee tres temporizadores de 32 bits, dos reservados para aplicaciones de usuario y uno reservado para el modo BIOS [12].

La frecuencia de funcionamiento del módulo se basa en el clock principal de 150 MHz. Esta frecuencia se ajusta mediante un prescaler configurable a través del registro PSC.

Al iniciar su funcionamiento, el módulo carga el valor almacenado en el registro PRD al registro TIM. El contador disminuye el valor de dicho registro a intervalos definidos por el

3.2. MÓDULO DE CONTROL DE LA MÁQUINA IMPULSORA

registro TDDR. Cuando el registro TIM alcanza el valor cero, el módulo temporizador activa una bandera de aviso y dispara la interrupción correspondiente, según el número de módulo determinado.

El timer 1 se utiliza para las aplicaciones de usuario, lo que permite la ejecución periódica de estas tareas sin interrumpir las tareas de control. Por otro lado, el timer 0 se reserva para las tareas de control, que requieren un tiempo de ejecución optimizado. Estas tareas son de alta prioridad y se ejecutan individualmente, sin que otras actividades del sistema las interrumpían.

3.2.3. PIE

El módulo encargado de gestionar las interrupciones del sistema, incluida la asignación de prioridades de ejecución [12], consta principalmente de un multiplexor que convierte las interrupciones del sistema en un número reducido de grupos de interrupciones, como se observa en la Figura 3.7. De esta forma, el módulo puede administrar y distribuir hasta 96 interrupciones individuales, agrupándolas en 12 grupos más pequeños.

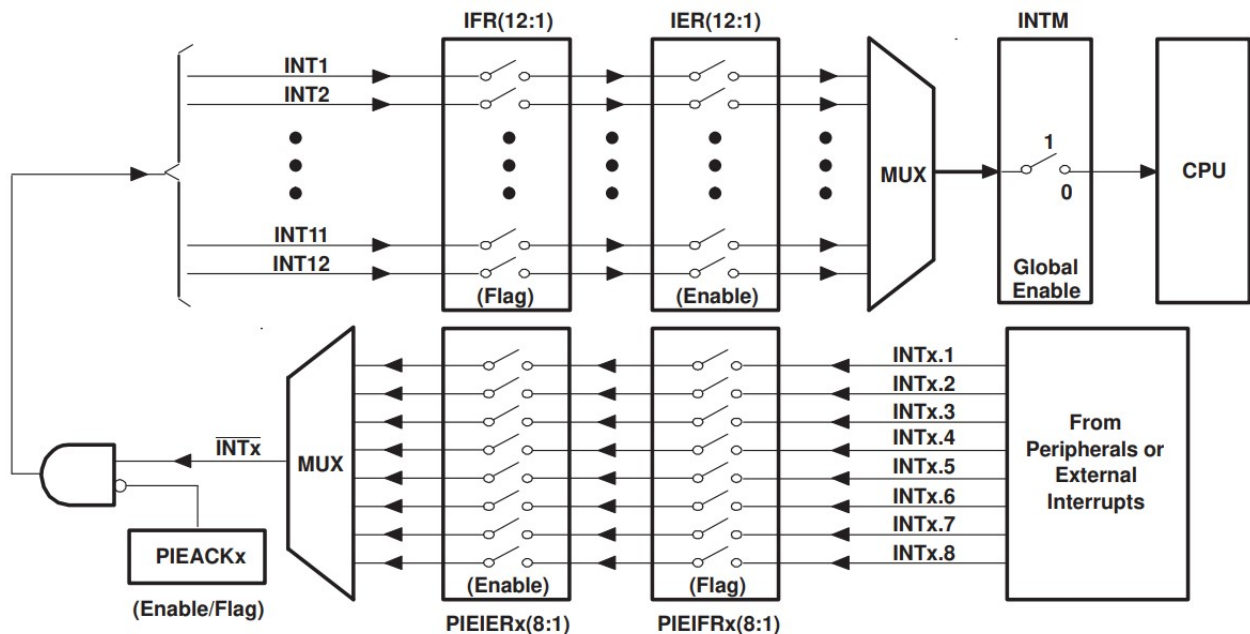


Figura 3.7: Multiplexor de interrupciones utilizando los bloques PIE.

Este módulo también permite la utilización de interrupciones definidas por el usuario, lo que posibilita forzar su ejecución cuando sea necesario, especialmente útil en tareas de usuario de baja prioridad. Comprender el funcionamiento de este módulo es esencial para optimizar el sistema y garantizar respuestas de operación acordes a los tiempos requeridos por la aplicación.

El sistema fue diseñado bajo la gestión de tareas mediante interrupciones, lo que permite su ejecución en tiempo real con una prioridad superior sobre las tareas de control. Estas últimas tienen un límite de tiempo de ejecución menor en comparación con otras aplicaciones. Por lo tanto, las tareas de control son manejadas por el timer 0, cuya interrupción se encuentra en el grupo 1 del PIE, con una prioridad mayor en el vector de interrupciones.

3.2.4. Módulo de Encoder de Pulsos de cuadratura (eQEP)

El módulo eQEP se utiliza con un encoder incremental para obtener la velocidad, posición y dirección de rotación de una máquina impulsora. Sin embargo, en el caso del emulador, solo es necesario conocer la velocidad de rotación, por lo que no es necesario calcular la posición ni la dirección [12].

En la Figura 3.8 se puede ver el disco de un encoder incremental con una serie de ranuras a lo largo de su periferia, que crean un patrón de líneas oscuras y claras. Se pueden ver dos señales típicas que tienen de salida este tipo de encoders, las señales QEPA y QEPB, que están desfasadas entre sí 90° (muy útil para obtener la dirección de giro). Además, algunos encoders incluyen la señal QEPI, la cual genera un pulso por revolución.

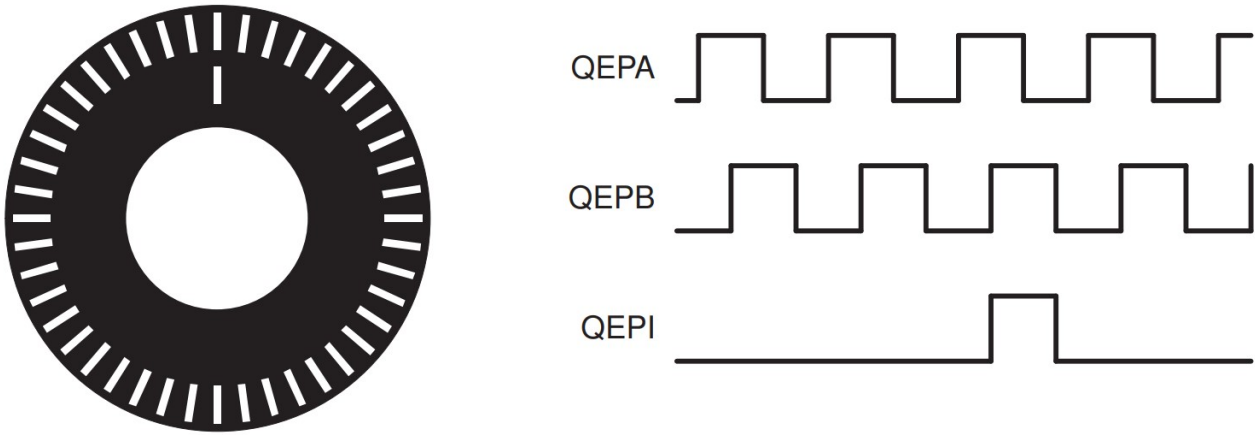


Figura 3.8: Disco de encoder incremental con pulsos en cuadratura.

Básicamente existen dos métodos de primer orden para la medición de la velocidad, denominadas de alta velocidad y de baja velocidad. La medición de alta velocidad se realiza mediante el conteo de cantidad de pulsos por una unidad de tiempo (Ecuación 3.1), mientras que el de baja velocidad se basa en la medición del tiempo entre pulsos (Ecuación 3.2).

Alta velocidad:

$$v(k) \approx \frac{x(k) - x(k-1)}{T} = \frac{\Delta X}{T} \quad (3.1)$$

donde se mantiene fijo el tiempo de medición T , y se mide la diferencia de pulsos entre la posición $x(k)$ y $x(k-1)$.

Baja velocidad:

$$v(k) \approx \frac{X}{t(k) - t(k-1)} = \frac{X}{\Delta T} \quad (3.2)$$

donde se mantiene fijo la cantidad de eventos X , y se mide el tiempo ΔT que tarda en producirse esa cantidad de eventos.

En este proyecto el método de cálculo de velocidad utilizado es el de baja velocidad, ya que proporciona una mayor precisión. El encoder implementado es el propio del banco de emulación, de la marca HEDSS modelo ISC5210-001G, de 1000 pulsos por vuelta.

3.2.5. Módulo de interfaz de comunicación serial (SPI)

La interfaz periférica serial es un puerto de entrada y salida (I/O) serial sincrónica de alta velocidad que permite un flujo de bits en serie de longitud programada (hasta 16 bits) que se

3.3. GENERACIÓN DE SEÑAL DE REFERENCIA EN EL DAC

transmite hacia dentro o fuera del dispositivo a una tasa de transferencia de bits programada. Se basa en el protocolo de comunicaciones del tipo maestro/esclavo de 4 líneas, con 2 líneas de comunicación, una de reloj y una línea de habilitación [13].

Este es el módulo utilizado para la comunicación con el DAC, que genera la señal para que el convertidor electrónico alimente a la máquina impulsora con la corriente necesaria. Está configurado para trabajar con palabras de 8 bits de longitud como maestro, con un baud-rate de 292,96875 kHz.

3.2.6. Módulo de interfaz de comunicación serial (SCI)

La interfaz de comunicación serial es un puerto de dos cables de datos, una para la recepción RX y otra para la transmisión TX [12]. Incorpora el protocolo de comunicaciones UART, que envía tramas de 8 bits, añadiendo un bit de comienzo y otro de parada, para señalar el comienzo y el final de la transmisión de datos.

Dado que el DSP TMS320F28335 trabaja con variables de punto flotante de simple precisión, es decir de 32 bits, se necesita enviar 4 tramas para representar correctamente las variables. Esto aumenta considerablemente los tiempos de ejecución. Por lo que la tasa de transmisión de bits en este caso debe ser seleccionado con cautela para obtener el comportamiento deseado.

Este módulo fue utilizado para establecer la comunicación entre el microcontrolador y la interfaz de usuario mediante el protocolo UART a una velocidad de baudios de 128000. Esta configuración específica se seleccionó con el objetivo de garantizar que, durante el estado de emulación (que se explicará más adelante), la comunicación se realice en un tiempo considerablemente inferior a 10 ms. Esta rápida comunicación es fundamental para mantener un rendimiento eficiente del sistema durante la emulación.

3.3. Generación de señal de referencia en el DAC

El generador de señal de referencia está compuesto por el convertor digital-analógico DAC8760, de Texas Instruments. Este DAC es de 16 bits con un único canal, con salida de corriente y tensión programable, que pueden utilizarse de manera simultánea [14]. El diseño del módulo de control contempla la utilización del DAC8760 en su modo de generador de corriente, permitiendo operar en un rango de 0 a 20 mA.

Como se mencionó previamente, el convertor digital-analógico utiliza el protocolo de comunicación SPI para su configuración. La trama del mensaje tiene una longitud de 24 bits, y se divide principalmente en dos partes: una primera parte de 8 bits que contiene el código de comando o dirección, seguida por una segunda parte de 16 bits que incluye los datos o la palabra a enviar, como se puede ver en el Cuadro 3.1. Y en el Cuadro 3.2 se ven los comandos reconocidos por el dispositivo.

BIT 23: BIT 16	BIT 15: BIT 0
Byte de dirección.	Palabras de datos.

Cuadro 3.1: Trama SPI predeterminada.

3.4. CÁLCULO DE CUPLA MOTORA

Byte de dirección	Función
0x00	Sin acción. (NOP)
0x01	Escritura en los registros de datos del DAC.
0x02	Lectura de registros.
0x55	Escritura del registro de control.
0x56	Escritura del registro de reset.
0x57	Escritura del registro de configuración.
0x58	Escritura del registro de calibración de ganancia del DAC.
0x59	Escritura del registro de calibración del cero del DAC.
0x95	Reset del error CRC.

Cuadro 3.2: Comandos de escritura de dirección.

La configuración utilizada en el dac, mediante el comando 0x55, es la siguiente:

- Rango de corriente de salida: 0 a 20 mA.
- Salida habilitada.
- Slew rate deshabilitado.
- Sin resistencia externa.

Cuando el microcontrolador envía el dato de la corriente de salida, lo hace utilizando el comando 0x01 del Cuadro 3.2, escribiendo en el registro el valor de corriente necesario. Este número debe estar comprendido entre 0 y 65535, debido a la resolución de 16 bits del DAC, lo que corresponde a un rango de 0 a 20 mA, respectivamente. Esta señal de corriente es la que luego los variadores de velocidad toman como referencia para imponer la cupla en el eje.

3.4. Cálculo de cupla motora

Se implementa mediante una interrupción que se ejecuta por defecto cada 1ms, con posibilidad de modificarlo. Esta interrupción consiste en la lectura de velocidad, cálculo de ϕ (Ecuación 2.5), con lo que se obtiene el valor de C_t mediante el ajuste descrito en la Sección 2.2.2. Luego se calcula el valor de cupla con la Ecuación 2.6, y finalmente con los comandos descritos en la Sección 3.2 se envía al DAC el valor de torque calculado. Se puede ver un diagrama de flujo en la Figura 3.9.

3.5. FIRMWARE DEL MÓDULO DE CONTROL

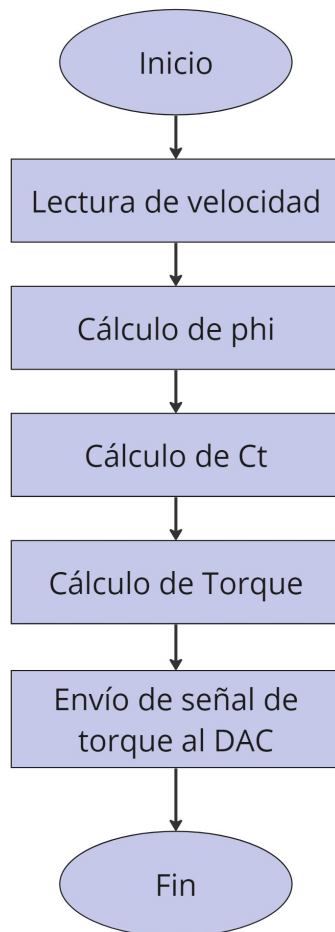


Figura 3.9: Diagrama de flujo de la interrupción de cálculo de cupla.

3.5. Firmware del módulo de control

La administración de las tareas de ejecución y optimización de los tiempos de cómputos el programa se basa en la ejecución por medio de interrupciones.

3.5.1. Estados del emulador y de la configuración

Existen diferentes estados que determinan el funcionamiento en el que está el emulador. En el Cuadro 3.3 se pueden ver los estados con sus descripciones.

Estado	Descripción
Inactivo (IDLE)	Sin actividad, espera acción del usuario remoto.
Configuración (CFG)	Cambio de parámetros del sistema.
Emulación (EMU)	Máquina emulando.
Detención (STOP)	Detiene el envío de señal de referencia de cupla.

Cuadro 3.3: Estados del emulador con sus descripciones.

Por otro lado, también se determina el estado de la configuración para poder seleccionar entre las 4 diferentes. Ver Cuadro 3.4.

3.5. FIRMWARE DEL MÓDULO DE CONTROL

Estado	Descripción
Indefinido (UNDEF)	Sin configuración seleccionada, indefinida.
Cambiando configuración (CONFIG)	Se entró en el estado de configuración en el emulador, se están cambiando parámetros de las configuración.
Configuración 1 (CFG1)	Seleccionada la configuración número 1.
Configuración 2 (CFG2)	Seleccionada la configuración número 2.
Configuración 3 (CFG3)	Seleccionada la configuración número 3.
Configuración 4 (CFG4)	Seleccionada la configuración número 4.

Cuadro 3.4: Estados de la configuración con sus descripciones.

Es imprescindible que ambos estados estén definidos antes de iniciar la emulación, ya que resultan fundamentales para la detección de errores y para conocer el estado actual del emulador, así como qué cálculos está llevando a cabo en ese momento.

3.5.2. Comunicación serial

La interrupción de comunicación serial tiene varios comandos para configuración y para la emulación. Se nombran y se explican estos comandos a continuación:

- STATECONFIG: Coloca el estado de configuración en CONFIG e ingresa en el modo de configuración para cambiar valores.
- STATECFGx: Coloca el estado de configuración en CFGx (siendo x un número entre 1 y 4).
- STATEIDLE: Coloca el estado de configuración en UNDEF y el estado de emulación en IDLE.
- STARTFRAME: Coloca el estado de emulación en EMU, y configura la interrupción de cálculo de cupla según el estado de la configuración (CFG1, CFG2, CFG3 o CFG4).
- STOPFRAME: Coloca el estado de emulación en STOP y el de configuración en UNDEF. A su vez detiene la ejecución de la interrupción de cálculo de cupla.
- SRESFRAME: Recibe el valor del recurso por la comunicación serial, guardándolo en memoria, y envía la velocidad de rotación y la cupla actual.

La interfaz de usuario tiene la responsabilidad de enviar estos comandos a medida que el usuario interactúa con ella. Además, se encarga de garantizar que el envío y la recepción de los datos se realicen en el orden adecuado y correctamente. Entre sus funciones más cruciales se encuentra el comando "SRESFRAME". Este comando es especialmente crítico, ya que se envía cada 10 ms como mínimo, por lo que es necesario tener especial cuidado en su funcionamiento. El desarrollo de la interfaz gráfica se detalla en el siguiente capítulo.

Capítulo 4

INTERFAZ DE USUARIO

La interfaz de usuario, o interfaz gráfica, es fundamental para simplificar el uso del emulador, permitiendo al usuario centrarse en los datos gestionados por este. Se optó por diseñarla en Python debido a su amplia variedad de librerías con una documentación detallada y accesible. Además, Python ofrece una sintaxis clara y legible, lo que facilita el desarrollo y mantenimiento de la interfaz, y su flexibilidad permite adaptarla a las necesidades específicas del proyecto con facilidad.

Las librerías utilizadas incluyen:

- `pyserial`: para la comunicación serial.
- `pyqt6`: para la ventana interactiva de la interfaz de usuario.
- `numpy`: para el manejo de datos numéricos.
- `matplotlib`: para la generación de gráficas.
- `pandas`: para la lectura y escritura de archivos, en este caso archivos `.csv`.
- `pyarrow`: complementaria a `pandas`.

El emulador consta de distintas partes, que incluyen:

1. Conexión del microcontrolador.
2. Modificación de los parámetros de configuración y del banco de emulación, junto con la carga y descarga de estos.
3. Generación de gráficas de torque-velocidad, potencia-velocidad, recurso-tiempo, torque-tiempo, etc.
4. Lectura del recurso en función del tiempo, guardado de los datos de emulación, inicio y detención de la emulación.

En la interfaz aquí desarrollada, se han abordado los problemas identificados en la versión anterior, que fue desarrollada en MATLAB. Algunos de estos problemas incluyen la falta de distribución libre debido a su dependencia de MATLAB, la fragmentación en múltiples ventanas que dificultaba la navegación del usuario, y una tasa de actualización de recurso limitada a 200 ms, lo que afectaba la resolución de los datos en las emulaciones. La nueva interfaz ha superado estos problemas al integrar todas las funciones en una sola ventana, aumentar la tasa de actualización del recurso a 10 ms y aprovechar la naturaleza de código abierto y desarrollo libre de Python para permitir una distribución más flexible y accesible.

A continuación se detalla el funcionamiento de estas partes.

4.1. PROTOCOLO DE COMUNICACIÓN CON EL MÓDULO DE CONTROL

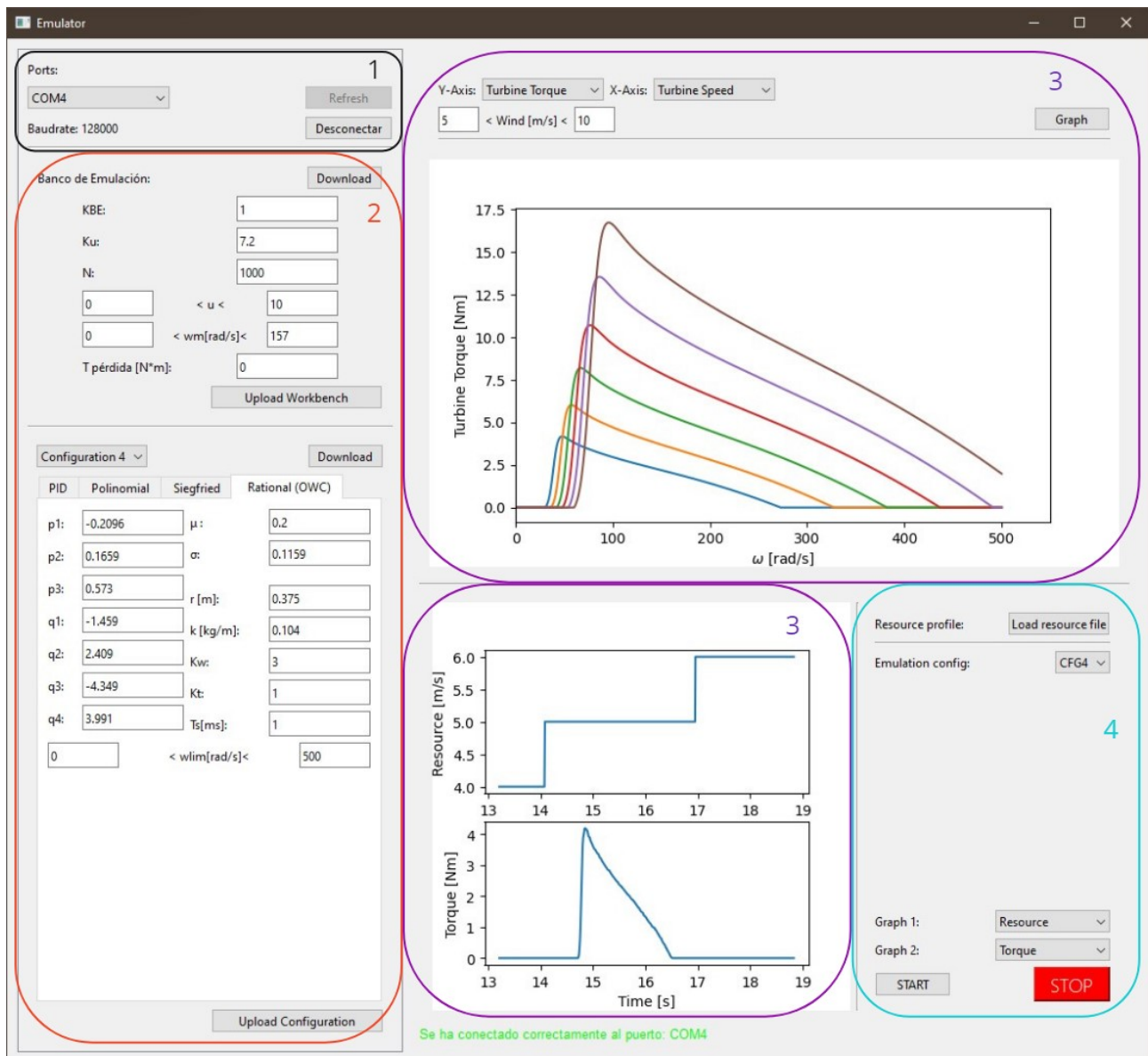


Figura 4.1: Captura de la interfaz de usuario. 1- Conexión con el microcontrolador, 2- Modificación de los parámetros de configuración y del banco de emulación junto con la carga y descarga de estos, 3- Generación de gráficas (divididos en 2), 4- Lectura y guardado de datos, inicio y detención de la emulación.

4.1. Protocolo de comunicación con el módulo de control

El protocolo de comunicación serial de la interfaz está estrechamente vinculado a las funciones descritas en 3.5. En este caso la interfaz se encarga de comandar al microcontrolador mediante el protocolo UART nombrado en 3.2.6.

Para el envío de datos se tiene un protocolo implementado con un Byte de inicio y otro de fin, junto con un *acknowledge* (ACK) desde la parte que recibe. Los datos pueden ser de tipo entero de 16 bits o de punto flotante 32 bits, por lo que existen 2 tipos, cada uno con su Byte de inicio. En los Cuadros 4.1 y 4.2 se detallan la tramas utilizadas.

4.2. GRÁFICAS

Byte de inicio	Dato de tipo float de 32bits	Byte de fin
Inicio de envío (1)	Dato de tipo float32 (4)	Fin de envío (1)
0x11	0xXX XX XX XX	0x10

Cuadro 4.1: Trama entre emulador e interfaz gráfica para datos de tipo punto flotante de 32 bits.

Byte de inicio	Dato de tipo entero de 16bits	Byte de fin
Inicio de envío (1)	Dato de tipo int16 (2)	Fin de envío (1)
0x12	0xXX XX	0x10

Cuadro 4.2: Trama entre emulador e interfaz gráfica para datos de tipo entero de 16 bits.

4.2. Gráficas

La generación de las gráficas es de gran utilidad para tener seguimiento del punto de operación de la turbina emulada. Por un lado, se dispone de una gráfica, ubicada en la parte superior, que permite visualizar las curvas características de la turbina y ofrecen varias opciones para seleccionar en ambos ejes. Y además, se puede visualizar el punto de operación en tiempo real sobre la gráfica de las curvas características mientras la emulación está en curso.

Para el eje de abscisas, se pueden seleccionar:

- Velocidad del generador.
- Velocidad de la turbina.

Y para el eje de las ordenadas, se pueden seleccionar:

- Torque del generador.
- Torque de la turbina.
- Potencia del generador.
- Control del valor de cupla.

Por otro lado, se tienen las gráficas de tiempo real, ubicadas en la parte inferior, en las que se disponen de dos representaciones para visualizar el recurso respecto al tiempo, el torque respecto al tiempo y la velocidad respecto del tiempo a elección del usuario.

4.3. Lectura del recurso y guardado de los datos de emulación

La lectura de los recursos se realiza desde la aplicación a un archivo de tipo CSV (Comma-Separated Values), el cual debe ser previamente creado (con la posibilidad de realizarlo con Matlab). Este tipo de archivo se caracteriza por ser un documento de texto con los datos separados por comas ampliamente utilizado debido a su simplicidad y compatibilidad.

En caso de utilizar Matlab para crear los valores del recurso en función del tiempo, se puede emplear la función “writematrix()”, **almacenado el vector de tiempo en la primera**

columna y los valores de los recursos en la segunda. Es importante destacar que el vector de tiempo debe tener un paso mínimo de 10 ms, ya que es el valor más pequeño con el que el emulador es capaz de trabajar debido a las prioridades de las tareas del microcontrolador.

Adicionalmente, una vez completada la emulación, existe la opción de guardar los datos de tiempo, recurso, torque y velocidad en otro archivo CSV para su análisis posterior en Matlab, utilizando la función “`readmatrix()`”, si así se desea.

4.4. Función de la interfaz durante la emulación

Para iniciar la emulación, se requiere establecer la conexión con el microcontrolador, cargar el archivo de recursos y seleccionar la configuración deseada. Una vez realizadas estas selecciones, la emulación comienza cuando el usuario presiona el botón **START**, estableciendo la configuración seleccionada en el DSP y luego enviando la señal de inicio de emulación.

Posteriormente, la interfaz se encarga de enviar el recurso en el intervalo definido por el archivo CSV hasta que se agoten los datos o hasta que se presione el botón **STOP**. Durante este proceso, se grafican los datos en tiempo real en las dos gráficas inferiores, y se resalta el punto de operación con un punto rojo en la gráfica superior.

La utilización de hilos de ejecución es necesaria en este caso para mantener la interfaz en funcionamiento mientras se ejecuta la emulación y para tener un buen manejo de los tiempos en el envío de datos. Se generan 2 hilos de ejecución, uno encargado de la comunicación de los recursos con el DSP, y el otro encargado de graficar.

4.4.1. Hilo de comunicación para la emulación

Esta ejecución es crucial para el correcto funcionamiento del emulador, y para ello la interfaz utiliza el reloj interno de la computadora para gestionar el tiempo de ejecución. Esto garantiza que el código sea coherente con el envío y la recepción de datos. Se crea un hilo de ejecución al iniciar la emulación, el cual consiste en un bucle *while* que ejecuta el envío y la recepción de datos a una tasa de tiempo fija hasta que se complete el envío del recurso o se detenga el emulador. Se proporciona un diagrama de flujo en la Figura 4.2.

El tiempo de actualización del recurso en el DSP está determinado por el tiempo asignado para la utilización del módulo de comunicación serial SCI. En este caso, el módulo tiene un intervalo asignado de utilización de 2 milisegundos, lo que significa que el envío de recursos puede realizarse como mínimo cada 4 milisegundos. Por lo tanto, se estableció un paso mínimo de actualización de recurso de 10 milisegundos, lo cual es considerablemente rápido.

4.4. FUNCIÓN DE LA INTERFAZ DURANTE LA EMULACIÓN

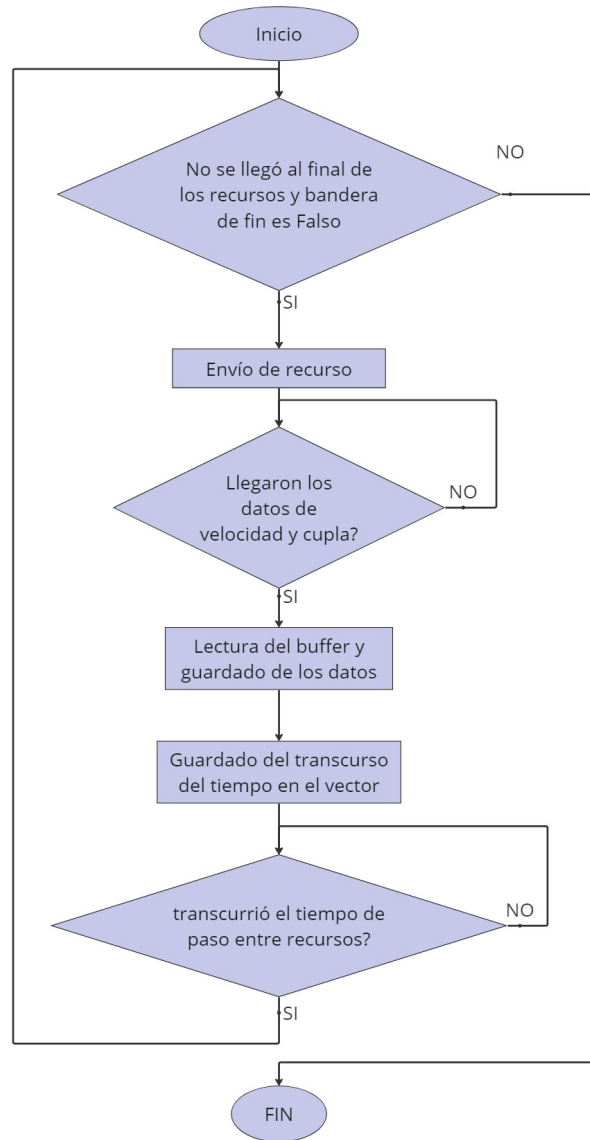


Figura 4.2: Diagrama de flujo del hilo de comunicación para la emulación.

Capítulo 5

RESULTADOS EXPERIMENTALES

Se recopilaron resultados en 2 etapas, la primera consta de muestras de la salida del DAC funcionando con referencia de tensión, con valores entre 0 y 10V, para corroborar los valores de cupla calculados en el DSP. La segunda parte está formada por los resultados del emulador funcionando con referencias de recursos escalonadas para recorrer las diferentes curvas de cupla-velocidad; y con otra referencia más realista sinusoidal monocromática, replicando de forma genérica y básica el comportamiento de las olas. En esta última parte se tienen en cuenta los valores de cupla y velocidad obtenidos, junto con la referencia aplicada.

5.1. Prueba del microcontrolador con el DAC

Para realizar pruebas del buen funcionamiento del microcontrolador junto a su DAC, que conforma toda la parte del emulador previa hasta obtener la señal de entrada a los convertidores electrónicos. Para obtener una buena muestra para cada recurso se tuvieron en cuenta algunos parámetros:

- Se realiza un barrido de velocidades desde 0 rad/s hasta 300 rad/s en bucle para observar las curvas completas.
- Se señala con un pico el comienzo del recorrido de velocidades.
- Durante cada muestra se mantiene el valor del recurso invariante.

Los valores de la configuración utilizada son los siguientes:

$$\begin{aligned}KBE &= 1 \\Ku &= 0,8 \\N &= 1000 \\u_{min} &= 0 \text{ V} \\u_{max} &= 10 \text{ V} \\w_{m \ min} &= 0 \text{ rad/s} \\w_{m \ max} &= 157 \text{ rad/s} \\T_p &= 0\end{aligned}$$

5.1. PRUEBA DEL MICROCONTROLADOR CON EL DAC

Y los valores de la configuración son:

$$\begin{aligned}p_1 &= -0,2096 \\p_2 &= 0,1659 \\p_3 &= 0,573 \\q_1 &= -1,459 \\q_2 &= 2,409 \\q_3 &= -4,349 \\q_4 &= 3,991 \\ \sigma &= 0,1159 \\ \mu &= 0,2 \\ r &= 0,375 \text{ m} \\ k &= 0,104 \\ Kw &= 1 \\ Kt &= 1 \\ Ts &= 10\text{ms} \\ w_{lim \text{ min}} &= 0 \text{ rad/s} \\ w_{lim \text{ max}} &= 300 \text{ rad/s}\end{aligned}$$

Teniendo en cuenta los resultados obtenidos con un recurso de $v_x = 3\text{m/s}$ y $v_x = 7\text{m/s}$, se comparan las señales de las Figuras 5.1 y 5.3 con las Figuras 5.2 y 5.4, respectivamente. En estas últimas se observan las señales que se esperaba ver en el osciloscopio.

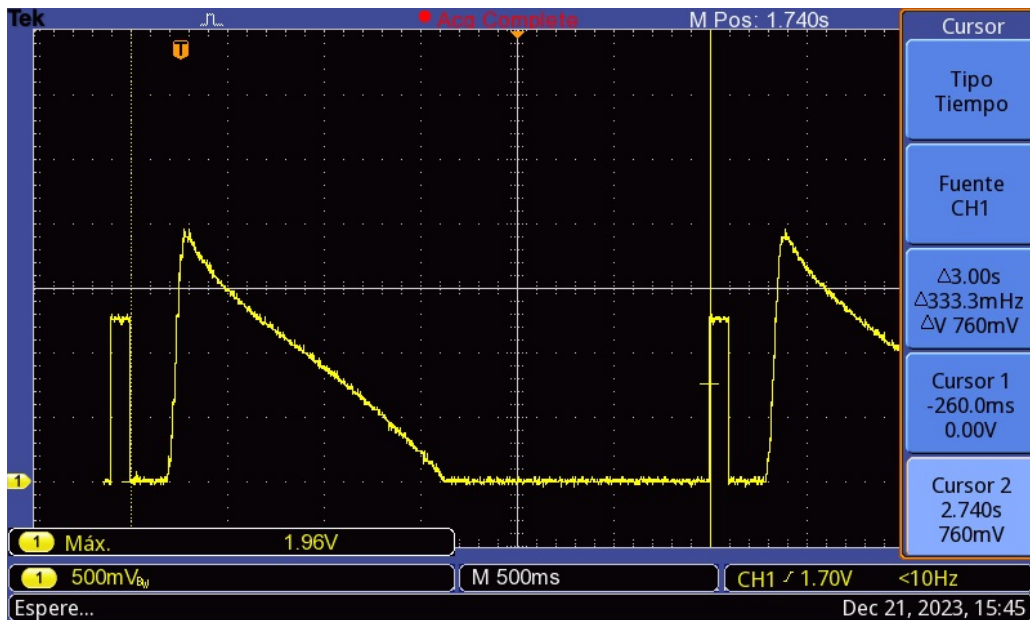


Figura 5.1: Salida del DAC con un recurso de 3.

Al analizar los resultados, se observa que la tensión de salida del DAC coincide con lo esperado en términos de amplitud y velocidad. Por lo tanto, la señal que llegaría al variador de velocidad es la correcta, lo que permite aplicar la cupla motora deseada de manera precisa y eficiente.

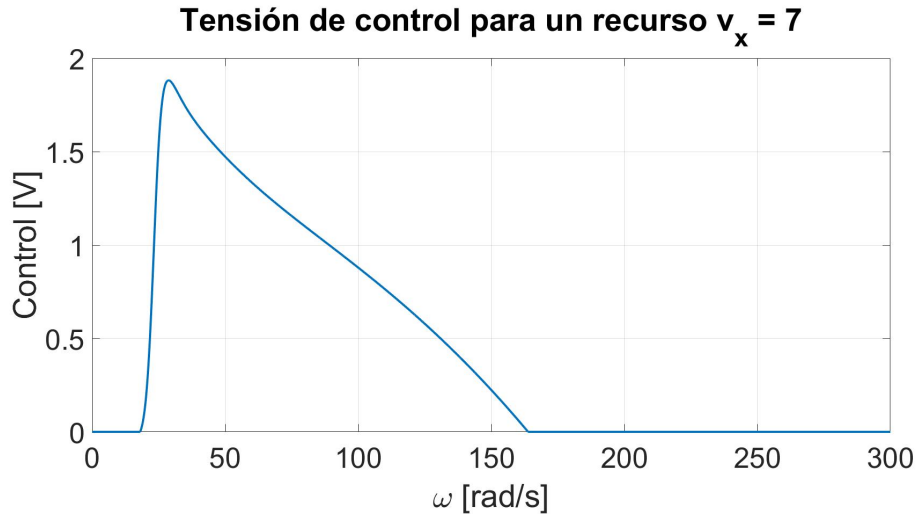


Figura 5.2: Cupla calculada con un recurso de 3.

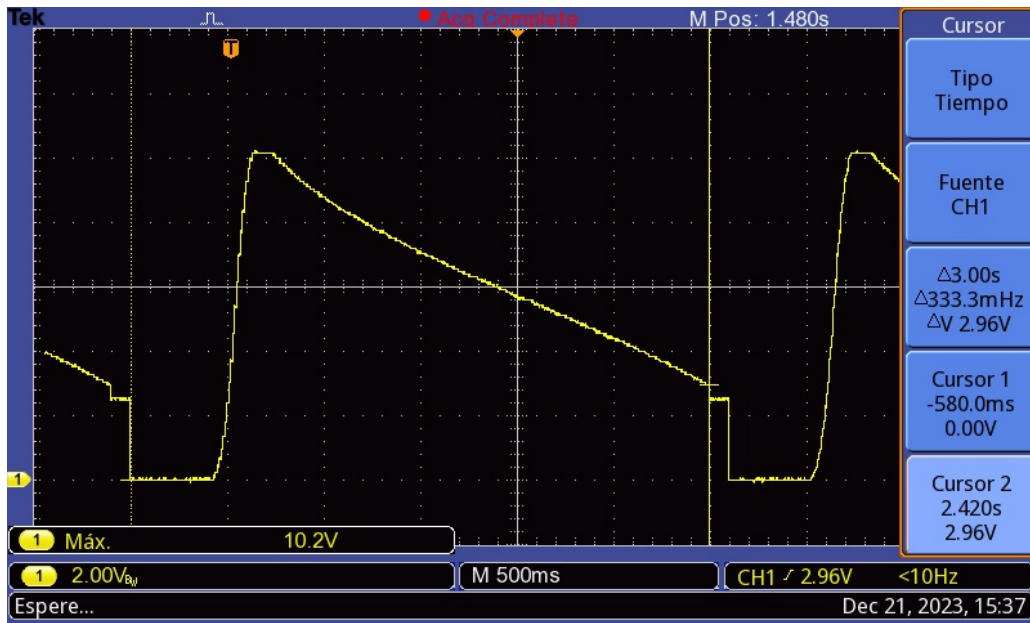


Figura 5.3: Salida del DAC con un recurso de 7.

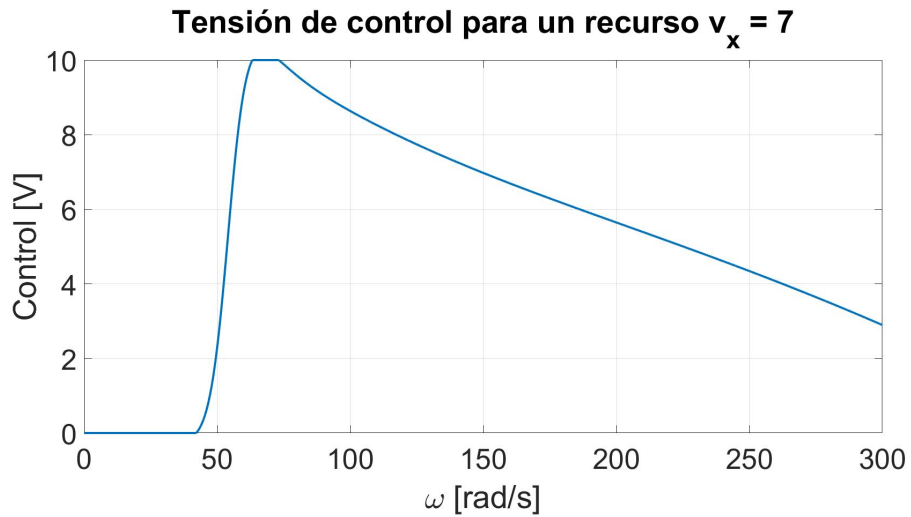


Figura 5.4: Cupla calculada con un recurso de 7.

5.2. Prueba del emulador completo

Esta prueba, a diferencia de la anterior, se realizó con toda la instalación emuladora funcionando. El banco de emulación en el que se tomaron las pruebas es el especificado en el Capítulo 3, disponible en el laboratorio LEICI.

Se realizó una serie de pasos cruciales y necesarios para el correcto funcionamiento de ambos motores y del microcontrolador. Teniendo ambos variadores conectados mediante la red a la computadora se encienden estos, y se coloca la máquina que funciona como generador en control de velocidad y la máquina impulsora controlando torque. La referencia de velocidad es impuesta mediante el software de los variadores, y la referencia del torque de la máquina impulsora es tomada desde el DAC.

Los valores utilizados para la configuración y el banco de emulación fueron iguales para ambas pruebas.

Banco de emulación:

$$\begin{aligned}
 KBE &= 1 \\
 Ku &= 7,2 \\
 N &= 1000 \\
 u_{min} &= 0 \text{ V} \\
 u_{max} &= 10 \text{ V} \\
 w_{m \ min} &= 0 \text{ rad/s} \\
 w_{m \ max} &= 157 \text{ rad/s} \\
 T_p &= 0
 \end{aligned}$$

Configuración:

$$\begin{aligned}
 p_1 &= -0,2096 \\
 p_2 &= 0,1659 \\
 p_3 &= 0,573 \\
 q_1 &= -1,459 \\
 q_2 &= 2,409 \\
 q_3 &= -4,349 \\
 q_4 &= 3,991 \\
 \sigma &= 0,1159 \\
 \mu &= 0,2 \\
 r &= 0,375 \text{ m} \\
 k &= 0,104 \\
 Kw &= 3 \\
 Kt &= 2 \\
 Ts &= 1 \text{ ms} \\
 w_{lim \ min} &= 0 \text{ rad/s} \\
 w_{lim \ max} &= 500 \text{ rad/s}
 \end{aligned}$$

5.2.1. Recurso escalonado

Primero se realizó con unos escalones de recursos, como se puede ver en la Figura 5.5, y se obtuvieron los resultados que se muestran a continuación.

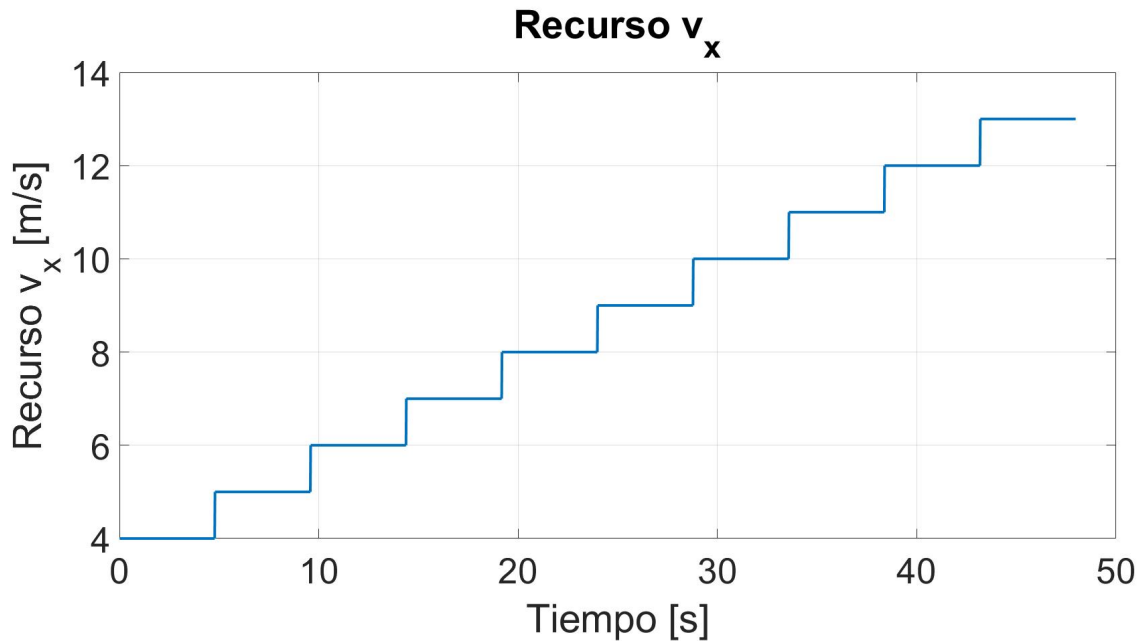


Figura 5.5: Recurso escalonado.

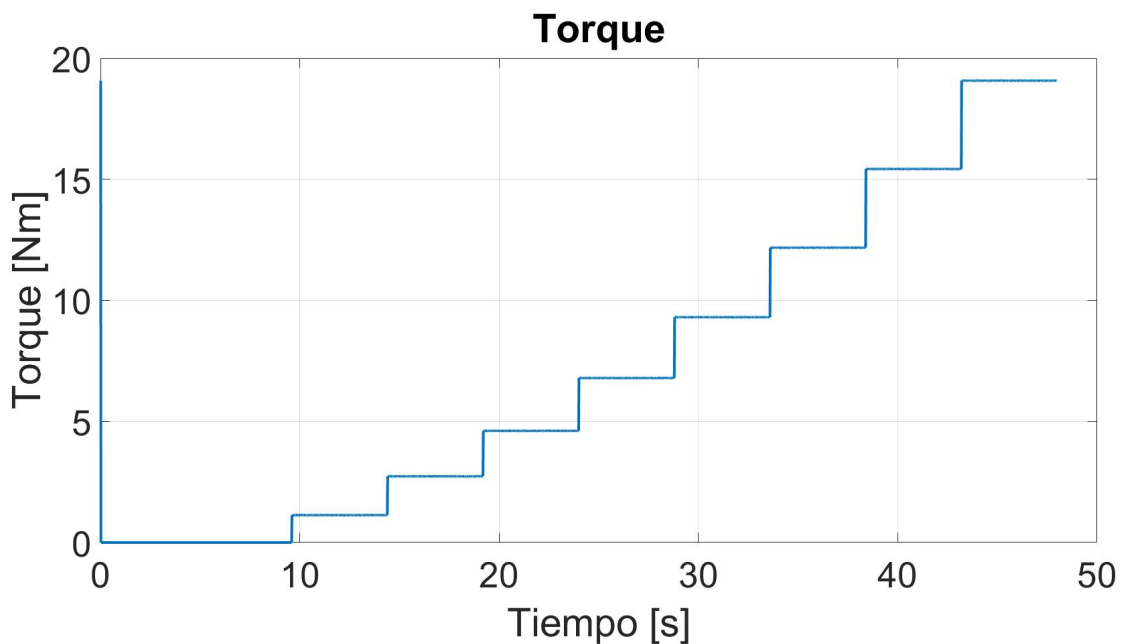


Figura 5.6: Torque calculado a lo largo del tiempo.

Observando las Figuras 5.5 y 5.6, se puede apreciar que al mantener un recurso constante, se obtiene un torque también constante, lo cual es consistente con lo esperado, ya que la velocidad está siendo controlada por la máquina de carga. Al relacionar estos resultados con la Figura 5.7, se observa que se van recorriendo las distintas curvas de cupla-velocidad para una velocidad fija, dado que se dejó el generador controlando a una velocidad constante.

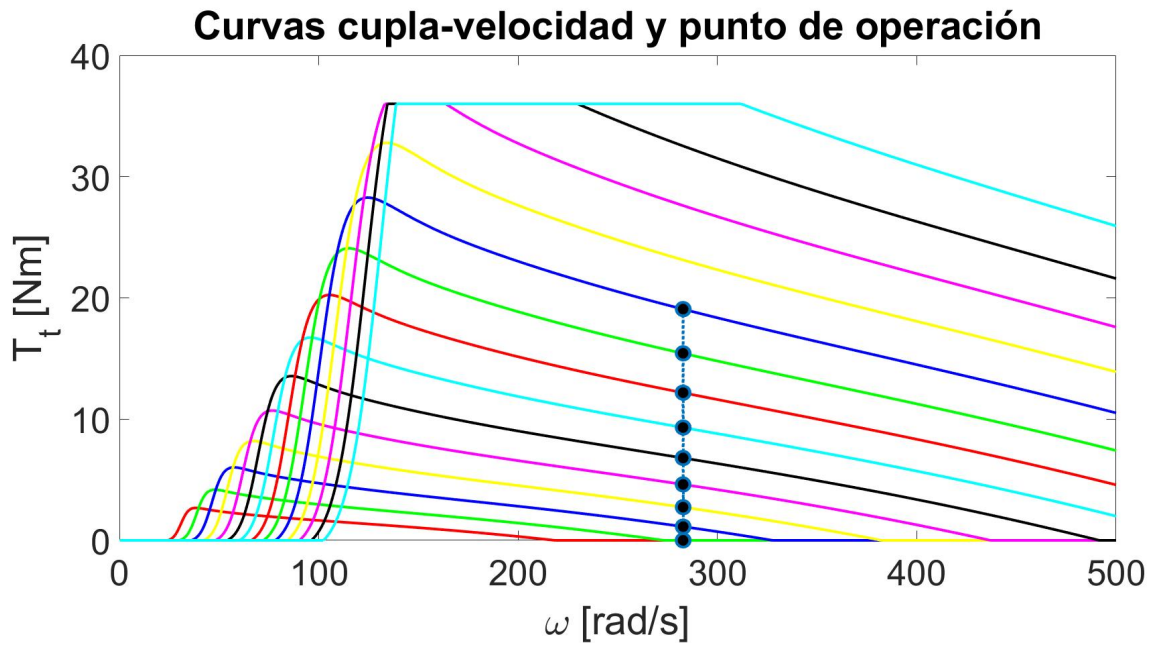


Figura 5.7: Punto de operación con recurso escalonado.

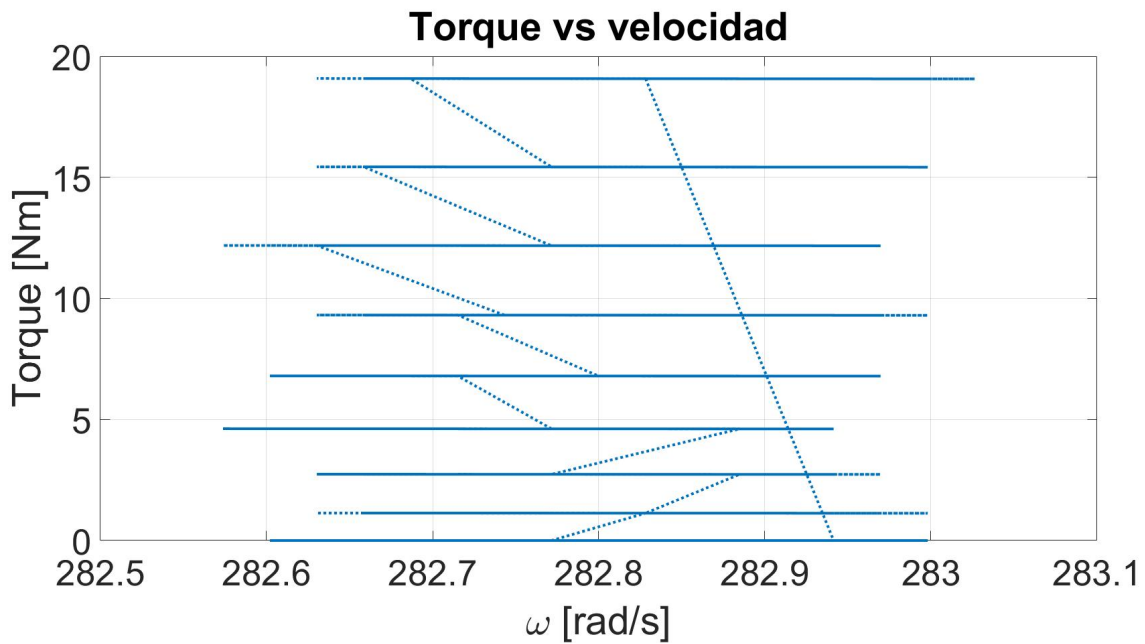


Figura 5.8: Velocidad vs torque.

Por otro lado, en la Figura 5.8 se pueden observar la dispersión de los puntos de operación de la Figura 5.7. Se puede apreciar los saltos en los valores de cupla debido a los valores de la referencia.

5.2.2. Recurso sinusoidal

En última instancia, se realizó una prueba utilizando una curva sinusoidal monocromática rectificada como recurso, donde después de un intervalo de tiempo, su amplitud aumenta. Esta curva se rectifica asumiendo que la turbina de Wells rectifica perfectamente el flujo de aire que atraviesa por ella.

5.2. PRUEBA DEL EMULADOR COMPLETO

A continuación se grafican los resultados obtenidos de la emulación.

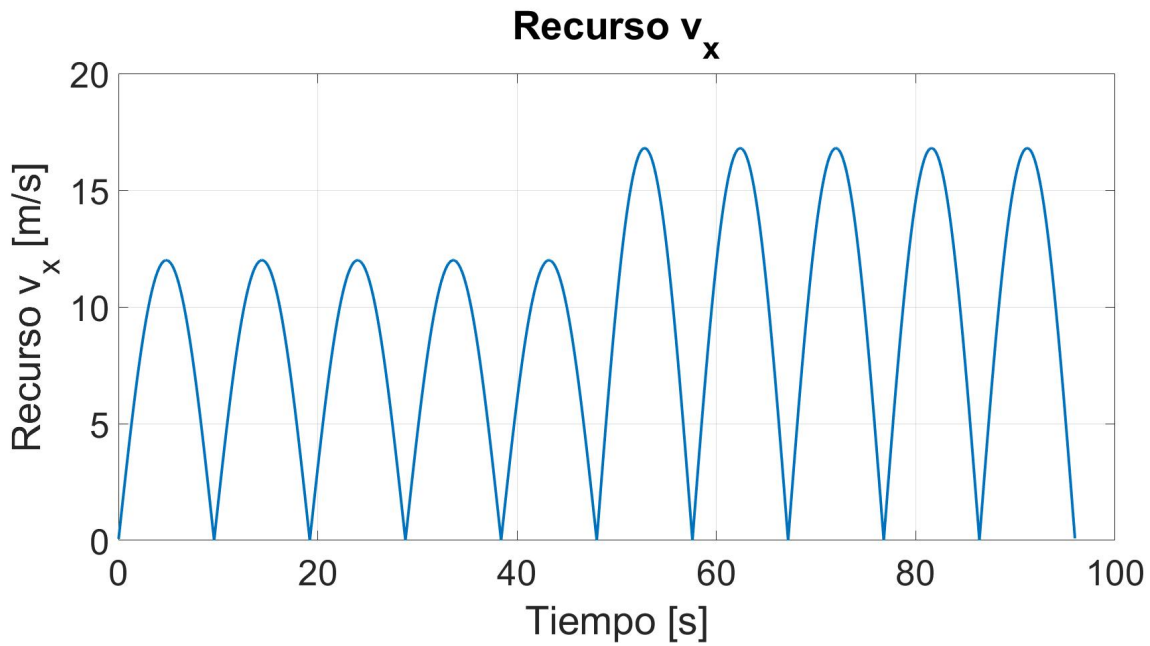


Figura 5.9: Recurso escalonado.

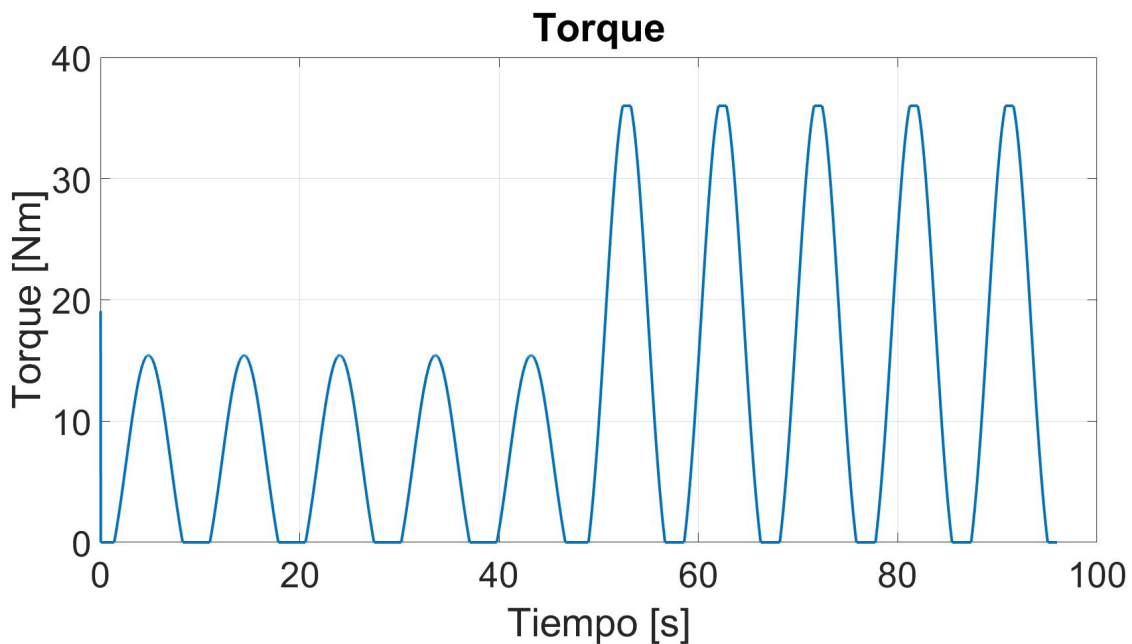


Figura 5.10: Torque calculado a lo largo del tiempo.

Al igual que en la prueba anterior, se mantuvo constante la velocidad durante toda la emulación, siendo controlada por el generador. Al analizar la Figura 5.10, se observa que cuando el recurso mantiene su primera amplitud, el torque adopta una forma similar, pero con zonas donde su valor es cero debido al bajo valor del recurso. A medida que el recurso aumenta en amplitud, el torque conserva esta forma, pero satura debido a que alcanza su valor máximo.

Por otro lado, al examinar la Figura 5.11, se puede apreciar que la línea vertical representa los distintos puntos de operación alcanzados por la máquina. Además, en la Figura 5.12 se puede observar con mayor precisión la dispersión de los valores de velocidad en relación con los valores de torque.

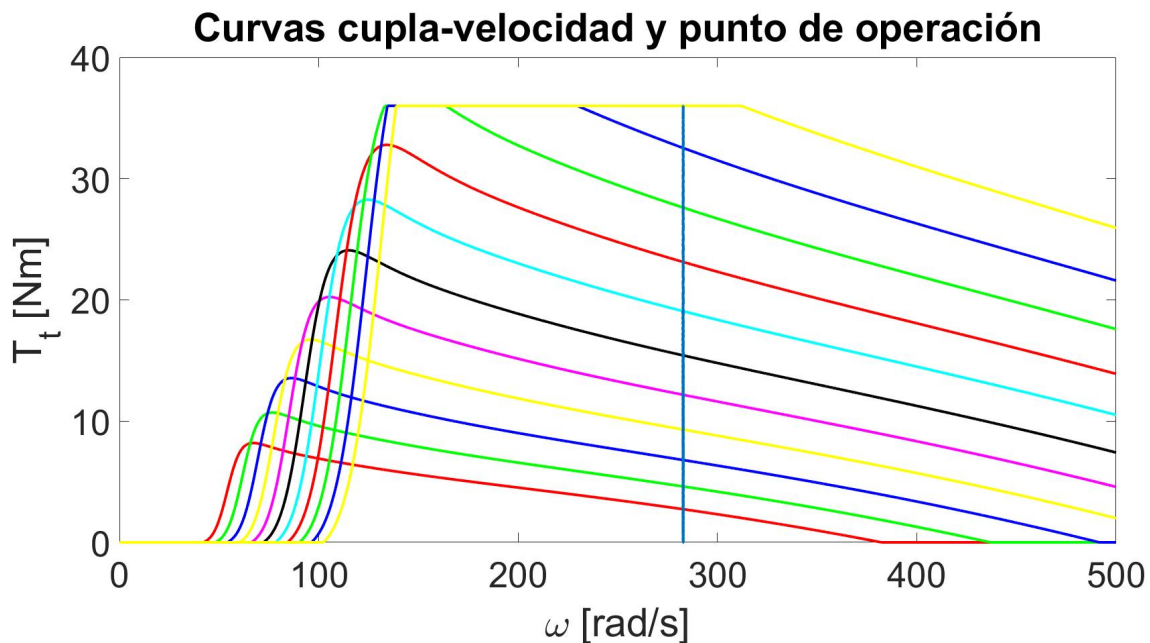


Figura 5.11: Punto de operación con recurso escalonado.

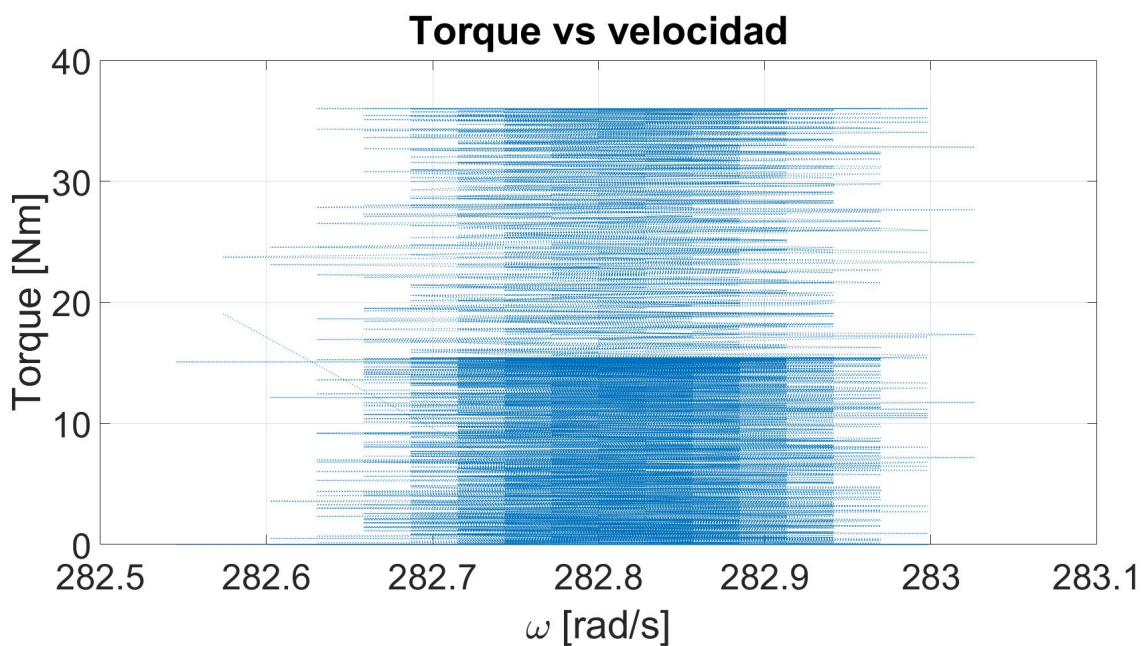


Figura 5.12: Velocidad vs torque.

5.3. Análisis de resultados

Los ensayos realizados con el emulador desarrollado arrojaron resultados generalmente satisfactorios. Este dispositivo permite llevar a cabo ensayos de emulación tanto para turbinas de Wells en sistemas de columna de agua oscilante (OWC) como para turbinas eólicas.

La señal de referencia en la salida del convertidor digital-analógico mostró respuestas adecuadas al seguir las variaciones de la velocidad del recurso, lo que resultó en una curva que se relaciona con la curva de cupla-velocidad de la turbina emulada. Este primer ensayo confirmó el correcto cálculo de los valores de cupla en el DSP y la suficiente rapidez de actualización del DAC para alcanzar el objetivo deseado.

5.3. ANÁLISIS DE RESULTADOS

En cuanto a la interfaz gráfica remota, cumplió eficazmente con su propósito. Esta interfaz representa una notable mejora con respecto a la versión anterior desarrollada en MATLAB, ya que utiliza Python, un software libre y ampliamente utilizado en la comunidad de desarrollo. Se logró una significativa mejora en su tasa de actualización del recurso, que ahora se envía cada 10 ms como mínimo, lo que significa un incremento de 20 veces en la cantidad de muestras con respecto a la versión anterior, de 200ms. Es importante destacar la versatilidad de esta interfaz, ya que es compatible tanto con sistemas operativos Windows como Linux, lo que amplía su accesibilidad y utilidad para diferentes usuarios y entornos de trabajo.

Capítulo 6

CONCLUSIONES Y TRABAJOS FUTUROS

6.1. Conclusiones

Como conclusión principal del trabajo, se puede destacar que se logró cumplir con el objetivo principal de diseñar y desarrollar una estación emuladora de conversión de energía marina funcional y versátil, apta para su utilización en entornos de laboratorio. La programación de las curvas y la interfaz de usuario se caracterizan por su fácil adaptación a diferentes configuraciones, lo que permite ajustarlo según las características específicas de cada aplicación.

La interfaz de usuario, implementada en Python, resultó ser muy versátil y fácil de usar, cumpliendo satisfactoriamente con su principal objetivo de proporcionar un control total sobre el emulador. Además, resulta muy simple de utilizar para la carga de perfiles de recurso mediante archivos CSV, y su capacidad de almacenar los resultados obtenidos en este tipo de archivos, los cuales son muy simples de manejar en MATLAB para su posterior análisis. Esta funcionalidad añadida proporciona una mayor flexibilidad y eficiencia en el análisis de datos y resultados de los ensayos realizados.

A lo largo de este trabajo, se abordaron diversos aspectos de la ingeniería electrónica, incluyendo la programación de microcontroladores, la electrónica de potencia, la electrónica digital y analógica, así como la programación de alto nivel. Por lo tanto, el proyecto no solo cumplió con su objetivo específico, sino que también proporcionó una oportunidad invaluable para aplicar y expandir los conocimientos en diferentes áreas de la ingeniería electrónica.

6.2. Trabajos futuros

Esta interfaz de usuario y el módulo de control desarrollados cumplen con los objetivos propuestos, ofreciendo una herramienta funcional y versátil. No obstante, hay varios aspectos que podrían mejorarse como:

- Rediseñar el protocolo de comunicación UART para evitar el desperdicio de bits y una mayor estabilidad.
- Implementar la interfaz gráfica en un sistema operativo de tiempo real para un control más preciso de los tiempos.
- Reemplazar el manejo de tareas de usuario en el DSP por la lectura de datos a partir de la recepción directa mediante interrupciones, optimizando así el procesamiento de la información.

6.2. TRABAJOS FUTUROS

Presentado el trabajo, se seguirán implementando actualizaciones con el objetivo de mejorar el emulador y garantizar que sea una herramienta confiable y con todas las prestaciones necesarias.

BIBLIOGRAFÍA

- [1] Hannah E Murdock et al. «Renewables 2021-global status report». En: (2021).
- [2] Carlos Victor Manuel Labriola y Cecilia Elizabeth Peralta. «Análisis de sistema de convertidores fluido-dinámicos de energía renovable para la Patagonia Austral de Argentina». En: *Informes Científicos Técnicos-UNPA* 4.1 (2012), págs. 128-143.
- [3] Mario Pelissero et al. «Aprovechamiento de la energía undimotriz». En: (2018).
- [4] Santiago Andrés Verne et al. «Desarrollo e implementación de una estructura dual para emulación de sistemas de conversión basados en energías renovables». En: *VII Jornadas de Investigación, Transferencia y Extensión de la Facultad de Ingeniería (La Plata, 25 al 27 de abril de 2023)*. 2023.
- [5] Umesh A Korde y John Ringwood. *Hydrodynamic control of wave energy devices*. Cambridge University Press, 2016.
- [6] Johannes Falnes. «A review of wave-energy extraction». En: *Marine structures* 20.4 (2007), págs. 185-201.
- [7] G Nolan, John Ringwood y B Holmes. «Short term wave energy variability off the west coast of Ireland». En: (2007).
- [8] António FO Falcão y Joao CC Henriques. «Oscillating-water-column wave energy converters and air turbines: A review». En: *Renewable energy* 85 (2016), págs. 1391-1424.
- [9] Tiziano Ghisu, Pierpaolo Puddu y Francesco Cambuli. «A detailed analysis of the unsteady flow within a Wells turbine». En: *Proceedings of the institution of mechanical engineers, part A: journal of power and energy* 231.3 (2017), págs. 197-214.
- [10] M. Alberdi et al. «Complementary Control of Oscillating Water Column-Based Wave Energy Conversion Plants to Improve the Instantaneous Power Output». En: *IEEE Transactions on Energy Conversion* 26.4 (2011), págs. 1021-1032. DOI: 10.1109/TEC.2011.2167332.
- [11] FD Mosquera et al. «Optimal wave energy extraction for oscillating water columns using second-order sliding mode control». En: *IET Renewable Power Generation* 14.9 (2020), págs. 1512-1519.
- [12] Texas Instruments. «TMS320x2833x, TMS320x2823x Technical Reference Manual». En: *Tms320x2833x Datasheet* (2020).
- [13] Texas Instruments. *TMS320x281x Serial Peripheral Interface Reference Guide, Rev. February 2009*. 2009.
- [14] Texas Instruments. «Datasheet DAC7760, DAC8760». En: *Tms320x2833x Datasheet* (2021).