

Aplicación de IoT al control y monitoreo de un sistema de recuperación de energía

Valagao Sancho, Mario¹

Laboratorio de Automatización y Control, Facultad de Ingeniería,
Universidad Nacional de la Patagonia San Juan Bosco, Comodoro Rivadavia, Argentina.
<https://orcid.org/0000-0001-6918-9460>
mario.valagao.s@gmail.com

Ahrtz, Gerardo D.²

Laboratorio de Automatización y Control, Facultad de Ingeniería,
Universidad Nacional de la Patagonia San Juan Bosco, Comodoro Rivadavia, Argentina.
<https://orcid.org/0000-0003-3928-0145>
gerahrtz@gmail.com

Sosa Tellechea, Cristian A.³

Sociedad Cooperativa Popular Limitada, Comodoro Rivadavia, Argentina
<https://orcid.org/0000-0003-4966-0440>
cristiansosat@gmail.com

Resumen

En este trabajo se emplean conceptos de Internet de las cosas aplicados a un sistema de energía renovable basado en una bomba centrífuga operando como turbina. El objetivo del sistema es regular la presión del agua a la salida de la turbina, convirtiendo la potencia hidráulica involucrada en eléctrica, y reinyectando ésta última a la red eléctrica. Se desarrolla un puente entre el protocolo de comunicación industrial Modbus y OPC UA. El puente (o pasarela) OPC UA/Modbus es de bajo costo y se complementa con el diseño de una interfaz de usuario empleando software libre. El conjunto se utiliza para monitorear al sistema de recuperación de energía y para obtener el modelo matemático. A partir de lo anterior, se diseña el control a utilizar en el controlador industrial. Resultados de simulación, permiten observar el buen comportamiento del control diseñado.

Palabras clave: IoT, Recuperación de energía, regulación de presión.

¹ Aux. de primera, Circuitos electrónicos II Facultad de Ing. UNPSJB, Argentina.

² Prof. Adjunto en Electrónica Industrial, Aux. de primera en Circuitos Electrónicos II, jtp en Proyecto final de ingeniería Electrónica, Facultad de Ing. UNPSJB, Argentina.

³ Jefe del Dpto. de innovación y automatización Sociedad cooperativa popular limitada de la ciudad de Comodoro Rivadavia, Argentina.

Application of IoT to the control and monitoring of an energy recovery system

Abstract

This work presents the Internet of Things concepts applied in the framework of a renewable energy system based on a centrifugal pump operating as a turbine. The objective of the system is to regulate the pressure of the water at the output of the turbine, converting the hydraulic power involved into electricity, and reinjecting the latter into the electrical grid. A bridge is developed between the Modbus industrial communication protocol and OPC UA. The OPC UA/Modbus bridge (or gateway) is low cost and is complemented by the design of a user interface using free software. The set OPC UA/Modbus interface monitors the energy recovery system and allows obtaining the mathematical model giving place to the design of the control used in the industrial controller from experimental data acquisition. Simulation results allow observing the good behavior of the designed control.

Keywords: IoT, energy recovery, pressure regulation.

1. Introducción

Los conceptos de Industria 4.0 buscan la conexión de todo tipo de dispositivos a Internet con el objetivo de intercambiar datos contribuyendo tanto a la recolección como al análisis de los procesos productivos (Schleipen y cols., 2016). Es bien sabido que el concepto de Internet de las cosas (IoT) es el comienzo de la cuarta revolución industrial siendo considerada una parte clave del futuro (Geng, 2016). En este sentido, el concepto de ciudades inteligentes está fuertemente vinculado a los usos IoT. De manera genérica puede considerarse que una ciudad inteligente es una infraestructura compleja donde operan diferentes sistemas optimizando el uso de recursos (Gram-Hansen, 2016). En particular, las redes eléctricas y de distribución de agua son dos sistemas claves de cualquier ciudad.

Por otra parte, para la gestión del recurso hídrico en una ciudad se emplean válvulas que regulan la presión a valores adecuados. En la ciudad de Comodoro Rivadavia, que posee desniveles de altura del orden de los 100m, aproximadamente, es posible reemplazar en algunos lugares a las válvulas reguladoras de presión de agua por bombas que, además, inyecten energía en la red eléctrica. Por lo anterior, las bombas trabajando como turbinas (PAT), tienen el doble propósito de regular la presión y recuperar energía (Carravetta, 2018; Carravetta y cols., 2018; Ramos y Borga, 1999). La operación de la PAT en una red de distribución de agua es compleja debido a las variaciones del caudal y de presión de entrada y el valor de la presión de salida que el ente distribuidor de agua desee establecer. En efecto, pueden registrarse variaciones significativas de dichos parámetros a lo largo del día debido a que son función del consumo del lugar. Luego, un sistema de control debe diseñarse adecuadamente para asegurar presión constante en el punto considerado (Fontana y cols., 2016).

Con el objetivo de estudiar estos sistemas, y en el marco de un proyecto conjunto, se construyó un prototipo de un sistema de recuperación de energía en colaboración entre la Sociedad Cooperativa Popular Limitada de Comodoro Rivadavia (SCPLCR), responsable de la distribución de agua y de electricidad en la ciudad de Comodoro Rivadavia, y la Facultad de Ingeniería de la Universidad

Nacional de la Patagonia 'San Juan Bosco' (UNPSJB).

En el mencionado banco de pruebas, se han realizado actividades de investigación que han dado lugar al trabajo aquí presentado tanto desde el punto de vista asociado a IoT como a sistemas de control. En efecto, en (Yncio y cols., 2018) se ha desarrollado un cliente Modbus y se ha realizado una primera identificación del modelo matemático del sistema, mientras que en (Cadiboni y cols., 2018) se diseñó un primer control que fue mejorado en (Peña y cols., 2020) que tomó en cuenta tanto la presencia de retardos como la respuesta dinámica deseada. Finalmente, en (Peña y cols., 2019) se presentó una puerta de enlace OPC UA/Modbus en una Raspberry Pi 2 usando Slackware Linux como sistema operativo y utilizando Python como lenguaje de desarrollo.

Este trabajo, que presenta una aplicación de IoT en el control y monitoreo de un sistema de recuperación de energía, tiene varios objetivos:

1. proporcionar una interfaz OPC UA para los diferentes dispositivos,
2. desarrollar una interfaz de usuario amigable del sistema,
3. obtener los modelos matemáticos del sistema de recuperación de energía y
4. diseñar el control.

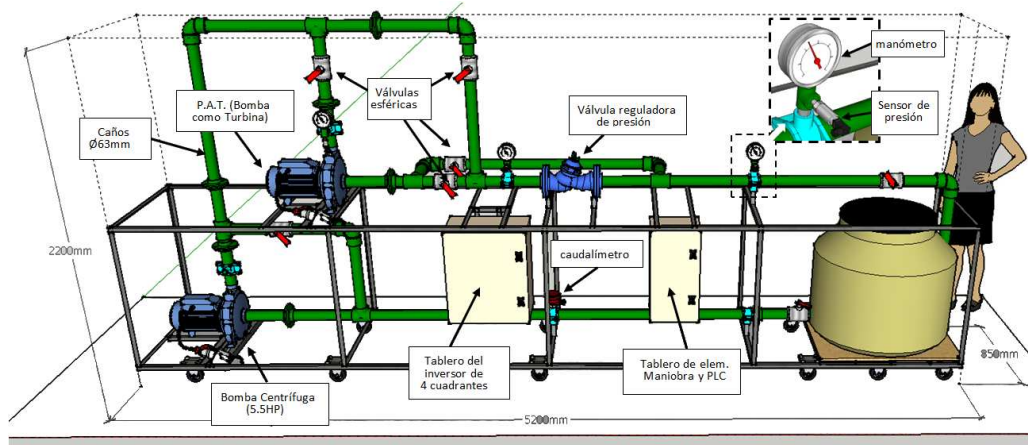
La estructura de este trabajo es como sigue. En la Sección 2 se introduce el sistema de recuperación de energía y el banco de ensayos desarrollado. En la Sección 3.1 se presentan los protocolos de comunicación industriales utilizados y el desarrollo de la pasarela OPC UA/Modbus. En la Sección 3.2 se incorpora la interfaz de usuario (HMI). En la Sección 4 se muestran ensayos experimentales a la vez que se obtienen los modelos matemáticos del sistema. En la Sección 5 se diseña el control y en la Sección 6 se muestran los resultados de simulación. Finalmente, se presentan las conclusiones de este trabajo en la Sección 7.

2. Sistema de recuperación de energía

Un esquema general del prototipo desarrollado en el Laboratorio de Automatización y Control de la Facultad de Ingeniería UNPSJB (FI-UNPSJB) es mostrado en la Figura 1. Cabe destacar que el banco de ensayos se encuentra conectado a la red ethernet de la UNPSJB. En la misma se puede observar el circuito hidráulico. Este prototipo tiene dos bombas centrifugas Speroni® Rotorpump 5,5 HP una se utiliza como turbina y la otra se utiliza para hacer circular el fluido en el circuito. Una fotografía del banco de prueba se presenta en la Figura 2, mientras que un diagrama del prototipo se indica en la Figura 3. El diagrama muestra una bomba hidráulica que funciona como una turbina (PAT) conectada a un variador de velocidad Siemens® de cuatro cuadrantes lo que le permite un flujo bi-direccional de energía. Como se observa en la Figura 3, el variador de velocidad de cuatro cuadrantes está conectado a la red eléctrica mediante un filtro de línea y un medidor de energía iEM3155 de Schneider Electric® está conectado aguas arriba. El sistema es controlado por medio de un controlador lógico programable (PLC) de la marca Allen Bradley®, modelo Micro 820. El PLC, Figura 3, se comunica con los distintos dispositivos por medio de cinco lazos de corriente de 4-20 mA:

- Velocidad de referencia de la turbina.
- Estimación de la velocidad de rotación PAT.

Figura 1: Esquema del prototipo del sistema de recuperación de energía



Fuente: Elaboración propia.

Figura 2: Banco de ensayos en el Laboratorio de Automatización y Control, FI-UNPSJB.

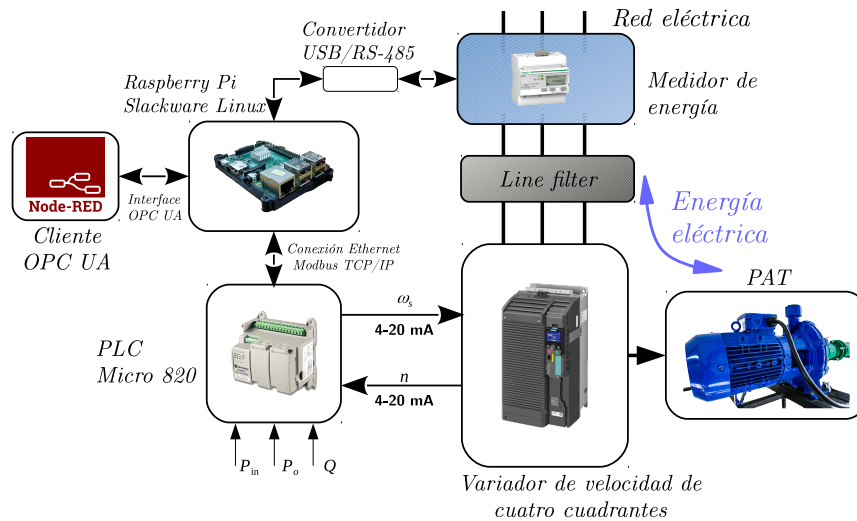


Fuente: Elaboración propia.

- Sensores de presión de entrada y de salida de la PAT.
- Caudalímetro.

Debido a que el objetivo principal es controlar la presión de salida y considerando que el flujo de agua lo imponen los consumidores, la acción de control para regular la presión de salida es la velocidad de la PAT. El medidor de energía, mostrado en la Figura 3, se conecta a una computadora Raspberry Pi 2 (RPi2) mediante un convertidor USB/RS-485. La RPi2 corre el sistema operativo Slackware Linux y permite el acceso, mediante ethernet, a operar y monitorear externamente las variables del banco de ensayos. En la Figura 3 se observa que mientras el PLC utiliza el protocolo de comunicación Modbus (*Modbus Foundation, 2019*), el medidor de energía emplea el protocolo de comunicación industrial serie RS-485. De allí la necesidad del convertidor USB/RS-485 para conectar el medidor de energía y la RPi2. El PLC también posee una interfaz ethernet que se puede conectar a la infraestructura de red de la UNPSJB.

Figura 3: Diagrama del prototipo



Fuente: Elaboración propia.

El medidor de energía iEM3155 de Schneider Electric ® proporciona las capacidades de medición esenciales que son: energía activa y reactiva, corrientes, tensiones, comunicación a través del protocolo Modbus (serie RS-485) y una pantalla o display (*iEM3155, 2019*). Los mapas Modbus con variables de interés para este trabajo se presentan en el Cuadro 1.

Resumen	Registros	Tipo	Unidad
Potencia Real	3060	Float 32	kW
Potencia Reactiva	3068	Float 32	kVAr
Factor de potencia	3192	Float 32	-
Frecuencia	3110	Float 32	Hz

Cuadro 1: Mapa Modbus del medidor de energía iEM3155

El medidor de energía posee un registro que indica si la energía es entregada o recibida, es decir si el sistema funciona como generador o como consumidor de energía. En el Cuadro 2 se resume esta información que corresponde al registro Modbus 3192 del medidor de energía (*iEM3155*, 2019). Además, el mapa Modbus del PLC es presentada en el Cuadro 3.

Cuadrante	Flujo de Potencia real	PF reg.	Energía entregada (importada) o recibida (exportada)
cuad. 1	positivo	0 to +1	Energía entregada (importada).
cuad. 2	negativo	0 to -1	Energía recibida (exportada).
cuad. 3	negativo	-2 to -1	Energía recibida (exportada).
cuad. 4	positivo	+1 to +2	Energía entregada (importada).

Cuadro 2: Energía entregada/recibida, Factor de potencia.

Resumen	Registros	Tipo	Unidad
Presión de entrada	40001	Unsigned 16 bit	kg/cm^2
Presión de salida	40002	Unsigned 16 bit	kg/cm^2
Velocidad	40003	Unsigned 16 bit	RPM
Caudal	40004	Unsigned 16 bit	$m^3/hour$

Cuadro 3: Mapa Modbus del PLC

3. Desarrollo de una pasarela OPC UA/Modbus y un HMI en Node-RED

3.1. Desarrollo de una pasarela OPC UA/Modbus

El protocolo industrial Modbus fue desarrollado por la empresa Modicon® (1979) y fue diseñado para la supervisión y control de controladores lógicos programables (PLC's) y otros dispositivos de supervisión. A pesar del tiempo, este protocolo se usa ampliamente en la industria debido a su flexibilidad. El protocolo industrial fue diseñado como una arquitectura maestro-esclavo. Existen numerosas versiones del protocolo Modbus, incluida la versión para redes de comunicaciones de puerto serie RTU (acrónimo en inglés de Remote Terminal Unit) y TCP (acrónimo en inglés de Transmission Control Protocol) (*Modbus Foundation*, 2019).

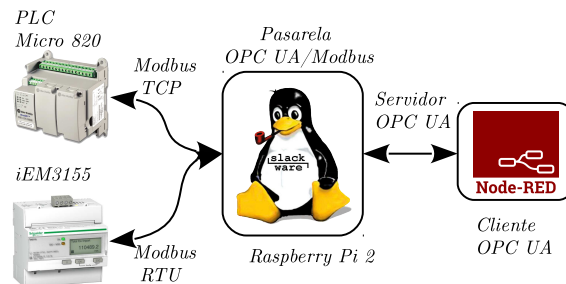
En 2006, la Fundación OPC propuso el protocolo de comunicación industrial OPC UA. Entre las características sobresalientes se establecieron la conexión a través de cortafuegos sin perder funcionalidades (*OPC Foundation*, 2019). Además, también implementa la firma y el cifrado del flujo de datos, lo que garantiza la seguridad y confiabilidad de la información industrial. Otra característica distintiva es la posibilidad de agregar marcas de tiempo al protocolo de datos OPC UA (Mahnke, 2009). Una de las principales ventajas de este protocolo de comunicación industrial es que permite conectar varios sistemas operativos y plataformas (Cavalieri y Chiacchio, 2013).

Una pasarela para los protocolos OPC UA y Modbus fue desarrollada en (Peña y cols., 2019) y es modificada en este trabajo. Esta herramienta proporciona una conexión OPC UA para que un cliente pueda adquirir datos y operar el prototipo.

La computadora Raspberry Pi puede utilizar diferentes sistemas operativos, muchos de ellos basados en GNU/Linux. En este trabajo se empleó un sistema operativo Slackware ARM Linux (*Slackware Linux Project*, 2019) para la computadora RPi2 debido a la facilidad de uso y estabilidad. Por ello se hizo una instalación completa de la versión 14.2 de Slackware ARM Linux en la computadora RPi2.

En la Figura 4, se puede observar un PLC Allen Bradley® Micro 820 conectado via ethernet a la RPi2. Además, un contador de energía de la empresa Schneider electric® se conecta mediante un adaptador RS-485/USB a la RPi2.

Figura 4: Esquema de la comunicación entre el PLC, iEM3155 y RPi2



Fuente: Elaboración propia.

Mientras que la computadora RPi2 es el maestro Modbus, los dispositivos conectados vía ethernet y RS-485 son esclavos Modbus. Por otra parte, para completar la pasarela, el servidor OPC UA acepta conexiones via ethernet sobre un puerto determinado. De esta manera, el sistema puede atender diferentes clientes. En este caso, se construye un cliente OPC-UA empleando Node-RED como se observa en la Figura 4. Para el desarrollo de la pasarela OPC UA/Modbus se utilizó el lenguaje de programación Python debido a su abundante documentación y flexibilidad (Hughes, 2010). El desarrollo del cliente OPC UA es explicado en detalle en la Sección 3.2. Hay diferentes implementaciones del protocolo Modbus para Python, en este trabajo se usan las librerías “*pymodbus*” con “*Free OPC-UA*” para crear la puerta de enlace. En esta implementación de la puerta de enlace OPC UA/Modbus se acceden a dos dispositivos, como se observa en la Figura 4, con diferentes velocidades de acceso.

Uno de los dispositivos, el medidor de energía, proporciona un gran número de variables y los datos son enteros sin signo de 64 bits, números de punto flotante de 32 bits y 16 bits (*iEM3155*, 2019). El segundo dispositivo conectado a la computadora RPi2 es un PLC Micro 820 con variables de 16-bit en su mapa Modbus. Entonces, debido a que el medidor de energía es más lento que el PLC para intercambiar datos, no es posible asegurar un tiempo de muestreo adecuado si el acceso es de forma secuencial.

Los sistemas operativos modernos como GNU/Linux pueden ejecutar múltiples procesos concurrentes. Para lograr esto, la CPU cambia la ejecución de un proceso a otro. Un proceso está compuesto por su espacio de memoria, las variables de entorno del sistema y los recursos de entrada/salida

(Hughes, 2010; Tanenbaum, 2015). Como en este caso, hay procesos que es menester realizar al mismo tiempo. Estas actividades se dividen en miniprocesos denominados hilos. Por otra parte, un hilo puede verse como un proceso liviano debido a que los hilos generalmente se ejecutan dentro de un contexto de un proceso padre. Una característica interesante de los hilos es que pueden compartir sus datos con el proceso principal y que son más livianos que los procesos (Tanenbaum, 2015). El lenguaje de programación Python tiene soporte para programación concurrente e hilos. Estos pueden ser implementados mediante la librería “*Threading*” (Hughes, 2010). El pseudocódigo de la pasarela OPC UA/Modbus usado en este trabajo se presenta en el algoritmo 1. En la primer parte del algoritmo (1), se muestra como el programa es inicializado: comunicación serie para Modbus RTU y TCP para medir los datos del medidor de energía y del PLC, respectivamente. A continuación, el servidor OPC UA es configurado para escuchar en el puerto 4840. Entonces, se inicia el hilo1 responsable de leer los datos del medidor de energía y copiar los datos obtenidos al servidor OPC UA. A continuación, el programa ingresa en un bucle donde se leen las variables del PLC a través del protocolo Modbus, estas variables se escriben en el servidor OPC UA. Se incrementa un contador i , que representa la cantidad de mediciones realizadas, y se produce un retraso t_d . La variable NM es el valor máximo de mediciones configurado previamente. Cuando la condición $i < NM$ es falsa el algoritmo (1) termina.

Algorithm 1 Software escrito en Python que se ejecuta sobre la computadora RPi2.

procedure PASARELA OPC UA/MODBUS

Configuración del maestro Modbus.

Inicializa el servidor OPC-UA.

Inicializa el hilo1

while $i < NM$ **do**

Lee las variables del PLC.

▷ programa principal.

Lee el medidor de energía vía Modbus RTU.

▷ hilo 1.

Escribe el valor de las variables al servidor OPC-UA.

$i = i + 1$;

delay(t_d);

Cierra hilo1.

Cierra las conexiones Modbus.

Cierra el servidor OPC-UA.

end.

return ;

3.2. Diseño del Cliente OPC UA y HMI en Node-RED

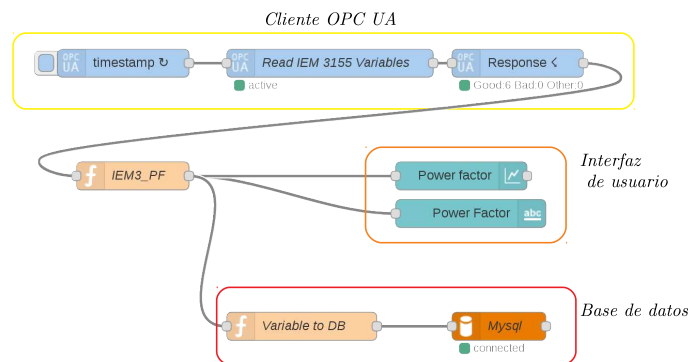
Node-RED es una herramienta de desarrollo ampliamente utilizada en aplicaciones de IoT (*Node-RED official webpage*, 2019; Kodali y Anjum, 2018), fue creada por IBM® y actualmente es un proyecto de la fundación OpenJS (*web OpenJS Foundation*, 2020). Una de las características atractivas de este entorno de desarrollo es que la edición de la aplicación se puede realizar diagramando un flujo de datos en un navegador web. La aplicación puede ser desplegada en un solo click y construida simplemente arrastrando elementos (nodos) desde una paleta y conectándolos (*Node-RED official webpage*, 2019).

En este trabajo se muestra como se desarrollo una interfaz amigable para el sistema de recuperación de energía. Esta herramienta es usada para visualizar los datos en tiempo real y salvar los datos en

una base de datos para su posterior análisis. En la Figura 5, para mayor claridad, se muestra parte del flujo de datos creado en el entorno de desarrollo Node-RED. El cliente OPC UA fue desarrollado a partir de los nodos de la paleta de Node-RED. La primera parte del flujo de datos dentro del cuadro amarillo muestra el cliente OPC UA, en el que se accede al servidor OPC UA a través de la dirección IP y el puerto. En el lado izquierdo del gráfico, el nodo de marca de tiempo identifica las variables OPC para su posterior uso en el siguiente nodo, cuya función es recuperar los valores reales de esas variables y luego enviarlos al último nodo de filtro, para comprimir y simplificar los datos que se utilizarán y almacenarán. Además, este bloque se ejecuta cada cierto intervalo de tiempo. El intervalo que se ajusta desde el entorno de desarrollo. Luego se configura el conector OPC UA y se leen las variables del servidor. Posteriormente, los datos pasan por un nodo de función que extrae el valor y luego la ruta se divide en dos, uno de ellos va a los nodos de comunicación de la base de datos MySQL (cuadro rojo) y el otro camino va a los nodos de creación de la interfaz de usuario (cuadro naranja). El recuadro naranja en la Figura 5 muestra el bloque “Variable a DB”. Este bloque toma el mensaje con la variable y los guarda en la tabla de base de datos especificada. El siguiente bloque “MySQL” es el bloque que gestiona la conexión a un servidor de base de datos, local o remoto. En las Figuras 6 y 7 se presenta los flujos de datos completos para el medidor de energía y el PLC, respectivamente.

Los nodos de interfaz de usuario poseen varias opciones para la personalización de la visualización de datos, lo que hace posible una HMI realmente útil. De esta forma, los usuarios pueden analizar información en tiempo real obtenida de diferentes dispositivos en un conjunto de pestañas o incluso en diferentes pantallas, lo que resalta los beneficios de esta plataforma en una implementación de IoT. Los resultados completos de esta aplicación HMI se muestran en la siguiente Sección.

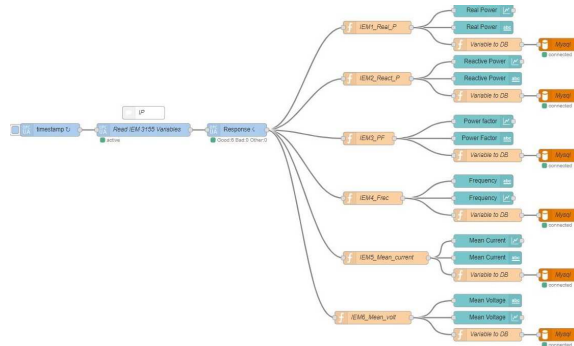
Figura 5: Una parte de un flujo de datos creada con Node-RED



Fuente: Elaboración propia.

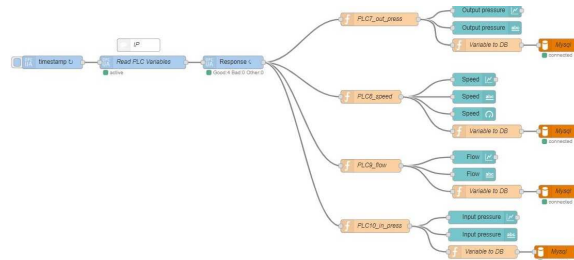
Cabe señalar que la utilización de software libre permite implementar estas ideas de forma modular y económica. Es decir, el servidor de base de datos y Node-RED pueden ejecutarse en un servidor dedicado. Este servidor puede tener la capacidad de realizar copias de seguridad con las que se puede proteger la información. Luego, la carga de la computadora RPi2 se puede optimizar y los datos de operación del sistema se pueden respaldar.

Figura 6: Flujo de datos del medidor de energía



Fuente: Elaboración propia.

Figura 7: Flujo de datos del PLC



Fuente: Elaboración propia.

4. Modelo del sistema

Los objetivos de control para un sistema de recuperación de energía construido con una PAT no se pueden lograr sin una técnica de control adecuada. Como el diseño de un controlador está basado en modelos, la calidad del modelo influye en el desempeño del sistema. En este trabajo se utiliza una regresión lineal por mínimos cuadrados para obtener un modelo matemático del sistema. De esta manera, un sistema con múltiples entradas y una salida (MISO) puede ser escrito de forma general como:

$$y(k) = \sum_{j=1}^{\nu} H_{j,1}(z)u_j(k), \quad (1)$$

donde $k = 1, 2, \dots, N$, $y(k)$ es la secuencia de salida, $u_j(k)$ son las entradas de la secuencia, ν es el número de entradas y $H_{j,1}(z)$ es una función de transferencia discreta dada por:

$$H_{j,1}(z) = \frac{B(z)}{A(z)}. \quad (2)$$

Entonces,

$$A(z)y(k) = \sum_{i=1}^{\nu} B_i(z)u_i(k) + e(k) \Rightarrow y(k) = -(a_1y(k-1) + \dots + a_ny(k-n)) + b_{1,1}u_1(k-1) + \dots + b_{1,n}u_1(k-n) + \dots + b_{\nu,1}u_{\nu}(k-1) + \dots + b_{\nu,n}u_{\nu}(k-n) + e(k). \quad (3)$$

La expresión anterior puede escribirse de forma compacta de la siguiente forma:

$$y = x\theta + E, \quad (4)$$

donde $x = [y(k-1), \dots, y(k-n), -(u_1(k-1), \dots, u_1(k-n)), \dots, -(u_{\nu}(k-1), \dots, u_{\nu}(k-n))]^T$ y θ son los parámetros desconocidos de la expresión (1).

$$\theta = [a_1, \dots, a_n, b_{1,1}, \dots, b_{1,n}, b_{\nu,1}, \dots, b_{\nu,n}]^T. \quad (5)$$

Es posible obtener los parámetros desconocidos del sistema dinámico aplicando el método de mínimos cuadrados de la siguiente forma:

$$\hat{\theta} = (x^T x)^{-1} x^T y. \quad (6)$$

Si el producto $x^T x$ es no singular, es decir invertible, es posible encontrar los parámetros del sistema (Åström, 1995).

Para obtener un modelo matemático del sistema de recuperación de energía se diseñó un experimento a partir de las herramientas desarrolladas en la Sección 3. Diferentes velocidades de referencia fueron aplicadas a la PAT para caracterizar el sistema.

Estos ensayos, en los cuales se cambio el valor de referencia de velocidad de la turbina (1000-1500, 1500-2000 y 2000-2500 RPM), y la interfaz de usuario creada con Node-RED son presentadas en las siguientes figuras. La Figura 8 muestra las mediciones de la presión de salida, velocidad del eje de la PAT (formato de línea de tendencia y como un indicador en tiempo real), caudal y presión de entrada a la PAT, respectivamente. La Figura 9 muestra las mediciones del medidor de energía de acuerdo a el Cuadro 1.

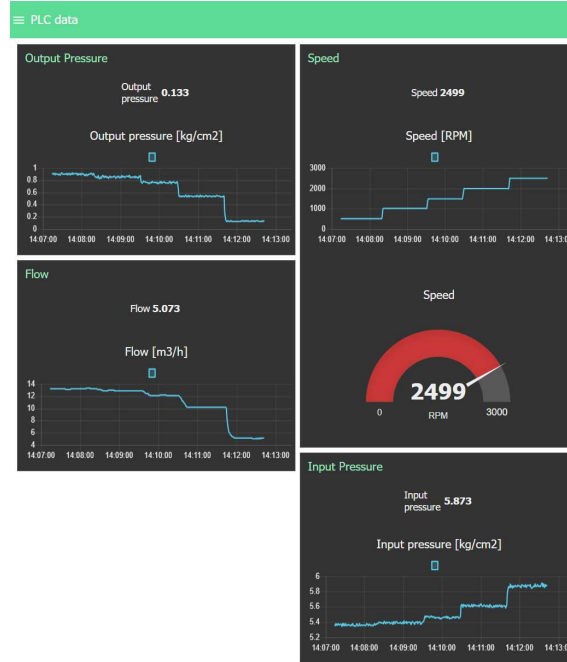
En el banco de ensayos, cada cambio en la velocidad de la turbina da lugar a variaciones en la presión de entrada y en el caudal en la PAT. Estas condiciones no se obtienen comúnmente en el campo debido a que el caudal depende de los consumidores y la presión en la entrada se puede determinar a partir del caudal. Sin embargo, en el banco de pruebas, los cambios en la velocidad PAT permiten determinar las funciones de transferencia considerando la presión de entrada y caudal.

Las variables de interés para el modelo del sistema de recuperación de energía son las variables hidráulicas del sistema y la velocidad de la PAT. El modelo, el cual consta de tres funciones de transferencia es:

$$y(k) = [H_{1,i}(z) \quad H_{2,i}(z) \quad H_{3,i}(z)] \begin{bmatrix} n \\ Q \\ P_{in} \end{bmatrix}, \quad (7)$$

donde $H_{1,i}(z)$, $H_{2,i}(z)$ y $H_{3,i}(z)$, son las funciones de transferencia con una condición de operación i . La salida del sistema es la presión de salida, las entradas del mismo son: n velocidad de rotación del eje de la PAT (500, 1000, 1500, 2000, 2500 RPM, respectivamente, Figura 8), Q es el caudal y P_{in} es la presión de entrada. A partir de las evoluciones de las variables se proponen las siguientes

Figura 8: Interfaz de usuario desarrollada con Node-RED, mediciones del PLC.



Fuente: Elaboración propia.

funciones de transferencia:

$$H_{1,i}(z) = z^{-r_1} \frac{n_1}{1 + d_{11}z^{-1} + d_{12}z^{-2}}, \quad (8)$$

$$H_{2,i}(z) = z^{-r_2} \frac{n_2}{1 + d_{21}z^{-1} + d_{22}z^{-2}}, \quad (9)$$

$$H_{3,i}(z) = z^{-r_3} \frac{n_3}{1 + d_{31}z^{-1} + d_{32}z^{-2}}. \quad (10)$$

Los parámetros obtenidos para las expresiones anteriores, aplicando mínimos cuadrados son presentados en el Cuadro 4 donde los pasos 1, 2 y 3 indican los cambios de referencia para el variador de velocidad (1000-1500,1500-2000 y 2000-2500 RPM, respectivamente). El tiempo de muestreo considerado es de 0,1 segundos.

Por otra parte, en el Cuadro 5, se muestran los porcentajes de ajuste de los modelos propuestos con los datos experimentales obtenidos en el banco de ensayos.

Es importante estudiar cómo se vincula el sistema de recuperación de energía con el sistema eléctrico de distribución. En este sentido, es interesante analizar las mediciones de energía y los modelos propuestos. En la parte inferior izquierda de la Figura 9 se puede observar la evolución del registro del factor de potencia. En la figura se observa un salto consecuencia de un cambio importante en el estado de funcionamiento del sistema de recuperación de energía. En efecto, como se indica en el Cuadro 2, un coeficiente +2 indica que el sistema esta funcionando como una carga eléctrica. Este

Figura 9: Interfaz de usuario desarrollada con Node-RED (Medidor de energía).



Fuente: Elaboración propia.

Cuadro 4: Parámetros del modelo

Pasos	Parámetros de los modelos propuestos											
	d_{11}	d_{12}	d_{21}	d_{22}	d_{31}	d_{32}	r_1	r_2	r_3	n_1	n_2	n_3
1	0,004	-0,73	-0,15	0,7	-0,1	-0,9	7	3	14	-6,8E-05	-0,06	-0,01
2	-0,006	-0,68	-1,43	0,4	0,009	0,4	7	16	1	-1,1E-04	2,3E-08	-0,2
3	0,108	0,006	-1,82	0,8	0,006	-0,9	13	3	1	-4,0E-04	2,3E-07	-0,009

cambio en las características operativas del sistema también se observa en la identificación del mismo según se presenta en (Peña y cols., 2020). De esta manera, la identificación de sistemas aplicando mínimos cuadrados puede aplicarse para obtener un modelo que se va a utilizar para sintonizar un controlador y también puede aplicarse para detectar cuando el sistema está trabajando como carga o como generador.

5. Diseño del control

A partir de la expresión (8) se observa la presencia del retardo r_1 en la respuesta del sistema que es presentado en el Cuadro 4. Es sabido que los retardos tienden a inducir comportamientos inestables en sistemas a lazo cerrado y por ello requieren un tratamiento particular en algunos casos. En general, los retardos están asociados a fenómenos de transporte, sistemas de comunicación, procesamiento digital de una señal, etc. Una mejora en el rendimiento de un controlador para

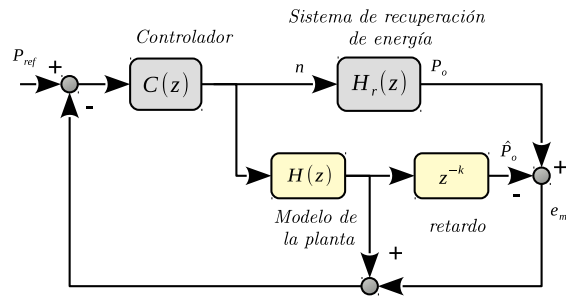
Cuadro 5: Porcentaje de ajuste de los modelos.

Step	Porcentaje de ajuste
1	69,04 %
2	78,95 %
3	88,46 %

un sistema que presenta retardos es la capacidad de predicción. Una aproximación ampliamente utilizada es el Predictor de Smith (Smith, 1958). Esta idea es la que emplea en este trabajo y una ventaja que tiene es que permite diseñar un controlador sin tener en cuenta el retardo.

Un esquema de un sistema típico con un predictor de Smith es presentado en la Figura 10, donde $C(z)$ es un controlador, H_R es la función de transferencia que representa el sistema de recuperación de energía, $H(z)$ y z^{-k} es una estimación de la planta y del retardo de la misma, respectivamente. Como se puede observar en la Figura 10, la diferencia entre la salida de la planta (P_o) y la estimación de la misma (\hat{P}_o) es la señal de error e_{mp} . Si esta diferencia es cero entonces el proceso ($H_r(z)$) y su estimación son iguales. En este caso, el controlador puede ser diseñado o sintonizado sin tener en cuenta el retardo ya que el lazo cerrado queda definido por el controlador y el modelo de la planta. De esta forma, la adición de un predictor de Smith mejora el desempeño del sistema cuando el mismo presenta retardos (Rico, 2007).

Figura 10: Diagrama del sistema con un predictor de Smith.



Fuente: Elaboración propia.

Si bien el modelo presentado en la expresión (7) posee múltiples entradas, dado que la presión de entrada y el caudal son impuestos externamente, solo la velocidad de giro de la turbina es la variable controlada. Entonces, el sistema se considerara de una entrada y una salida (SISO) para el diseño del control, siendo las variables presión de entrada y caudal perturbaciones del sistema. Debido a que las perturbaciones tienen una dinámica más lenta que el control, es posible considerarlas como constantes. El modelo general de un sistema SISO está dado por:

$$Ay(t) = B(u(t) + v(t)), \tag{11}$$

donde u , v e y son la entrada, una perturbación aplicada a la entrada, y la salida del sistema, respectivamente; A y B son polinomios.

Un controlador lineal para el sistema (11) se puede escribir como:

$$Ru(t) = Tu_c(t) - Sy(t), \quad (12)$$

donde u_c es la referencia, R , T y S son polinomios, respectivamente. El sistema a lazo cerrado es:

$$y(t) = \frac{BT}{AR + BS}u_c(t) + \frac{BR}{AR + BS}v(t). \quad (13)$$

Siendo el polinomio característico de lazo cerrado:

$$AR + BS = A_c \quad (14)$$

donde A_c es el polinomio con la dinámica deseada. La expresión (14) es conocida como la ecuación diofantina (Åström, 1995).

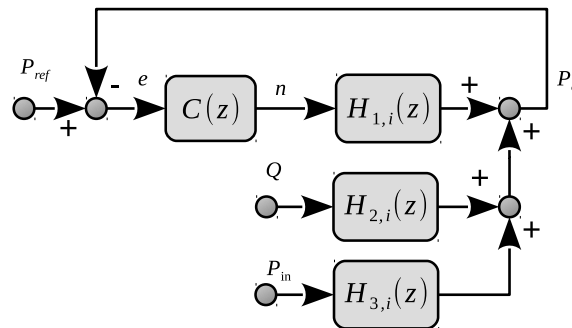
5.1. Diseño de un controlador para el sistema de recuperación de energía

Si consideramos la siguiente función de transferencia discreta para el controlador:

$$C(z) = \frac{S(z)}{R(z)}, \quad (15)$$

con $S(z) = s_0z^2 + s_1z + s_2$ y $R(z) = (z - 1)(z + p_1)$ el numerador y el denominador del controlador, respectivamente, el objetivo del diseño es determinar los parámetros s_0 , s_1 , s_2 y p_1 de $C(z)$. Un diagrama a lazo cerrado del sistema recuperación de energía, donde se observan las entradas de perturbación P_{in} y Q , se muestra en la Figura 11.

Figura 11: Diagrama del sistema de recuperación de energía.



Fuente: Elaboración propia.

Un primer paso para el diseño del controlador es seleccionar la respuesta dinámica deseada para el sistema a lazo cerrado. En este sentido, existen diferentes tipos de respuestas. Un análisis acerca de la síntesis de respuestas transitorias óptimas fue desarrollada en (Graham y Lathrop, 1953). En este trabajo, se elige una respuesta transitoria para minimizar la integral del valor absoluto del error

multiplicada por el tiempo (ITAE, por sus siglas en inglés) (Graham y Lathrop, 1953; Franklin, 1994), conforme a la siguiente expresión:

$$I = \int_0^{\infty} t |e| dt, \quad (16)$$

donde e es la señal error, Figura 11.

De forma general, la función de transferencia de la dinámica deseada de lazo cerrado, de tiempo continuo, puede representarse como:

$$H_{LC}(s) = \frac{b_1 s^n + b_2 s^{n-1} + \dots + b_{n+1}}{a_1 s^n + a_2 s^{n-1} + \dots + a_{n+1}}, \quad (17)$$

donde a_1, \dots, a_{n+1} y b_1, \dots, b_{n+1} son los coeficientes del polinomio numerador y denominador de la función de transferencia, respectivamente. Dichos coeficientes son conocidos y se desprenden del criterio ITAE antes mencionado. La ecuación (17), posee un equivalente en tiempo discreto:

$$H_{LC}(z) = \frac{b_1 z^n + b_2 z^{n-1} + \dots + b_{n+1}}{a_1 z^n + a_2 z^{n-1} + \dots + a_{n+1}}, \quad (18)$$

que permite resolver la ecuación diofantina (14):

$$(z^2 + d_{11}z + d_{12})R(z) + n_1 z^2 (s_0 z^2 + s_1 z + s_2) = A_c(z), \quad (19)$$

donde s_0, s_1, s_2 y p_1 son los coeficientes del controlador a ser determinados. Como el criterio ITAE se calcula para sistemas continuos, en el diseño del controlador se consideraron un par de polos complejos conjugados cuya ubicación es $-0,7071 \pm i0,7071$ y dos polos rápidos fueron incorporados a los efectos de resolver la ecuación (19). Los coeficientes para el controlador son obtenidos empleando el modelo cuando el sistema opera en las velocidades 1000-1500 RPM. Los parámetros obtenidos, para el sistema discreto (15) y con un periodo de muestreo de 0,1 segundos, son: $s_0 = -788,4571$, $s_1 = 1,9781e + 03$, $s_2 = -1,5686$ y $p_1 = 0,4784$, respectivamente.

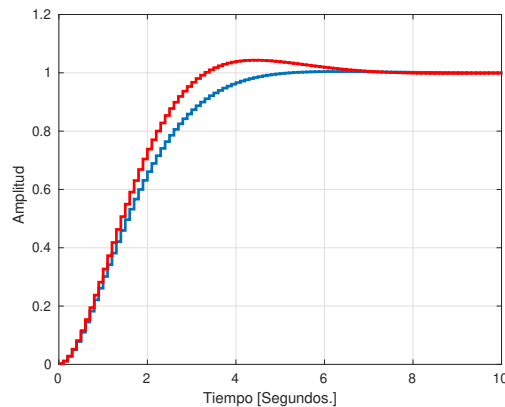
6. Resultados de Simulación

En esta Sección se simula el modelo obtenido del sistema y se evalúa el desempeño del controlador diseñado. Primero se compara la síntesis de un controlador empleando el criterio ITAE y empleando una función de transferencia de filtro de tipo Bessel. Por último se analiza como responde el sistema a lazo cerrado frente a variaciones en el retardo e incertidumbres en el modelo.

Es deseable que el sistema de recuperación de energía siempre opere como generador, es decir, suministrando energía a la red eléctrica. Por lo anterior, es razonable limitar la velocidad de funcionamiento del sistema en función de las mediciones eléctricas del registro factor de potencia. A los efectos de evitar que estos sistemas consuman energía de la red eléctrica, trabajan en serie con una válvula reguladora de presión en la red de distribución de agua y una válvula de bypass (Carravetta, 2018). Este tipo de arreglo permite una mejor operación del sistema de recuperación de energía, permitiendo operar en puntos de mejor eficiencia, y asegura el control de presión de agua frente a cortes del suministro eléctrico, etc.

En la Figura 12 se puede observar una comparación de la sintonía de un controlador empleando los coeficientes de un filtro de Bessel, empleado en (Peña y cols., 2020) y empleando el criterio ITAE

Figura 12: Comparación del desempeño de un controlador diseñado usando un filtro Bessel y el criterio ITAE.



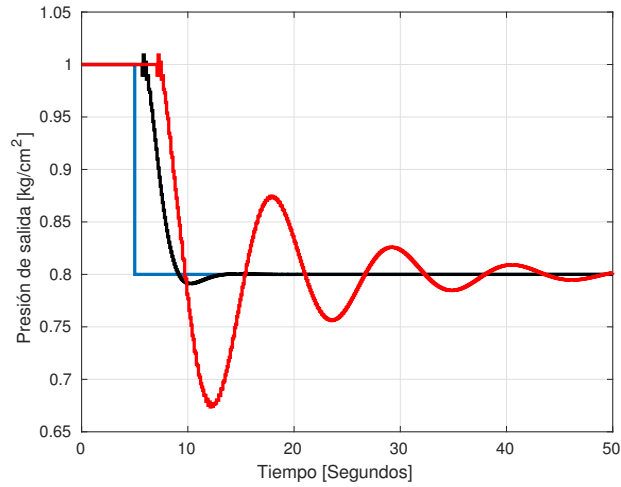
Fuente: Elaboración propia.

presentado en la Sección 5. En la Figura 12 se observa la evolución del sistema a lazo cerrado frente a un escalón en su entrada. En esta figura, con línea de color azul, se representa la respuesta del sistema a lazo cerrado con un controlador sintonizado empleando los coeficientes de un filtro de Bessel. Mientras que con línea de color rojo se presenta la respuesta de un controlador empleando el criterio ITAE. Como se puede observar en los resultados de simulación la respuesta de tipo ITAE tiene un pequeño sobrepaso mientras que el diseño de tipo Bessel casi no lo tiene. No obstante, la respuesta usando el criterio ITAE para el diseño del controlador es más rápida que la de tipo Bessel. En la Figura 13 se muestra una simulación con la respuesta temporal del sistema frente a un cambio de la referencia de presión de entrada. Además, se introduce una variación en el retardo de la planta. En la Figura 13 se observa con línea de color azul, negro y rojo el valor de la presión de referencia, la evolución del sistema a lazo cerrado y la evolución del sistema con un retardo 3 veces más grande que el que posee el modelo, respectivamente. En la Figura 13, se puede observar que el sistema a lazo cerrado (línea de color negro) responde según la dinámica diseñada. Además, frente a una variación del retardo (tres veces más grande) el sobrepaso fue mayor al diseñado pero el sistema sigue siendo estable.

En la Figura 14 se presenta una simulación con un cambio en la entrada de referencia del sistema para una variación de parámetros en la planta de $\pm 10\%$ en uno de los polos. Estas variaciones se presentan en líneas de color rojo ($+10\%$) y negro (-10%).

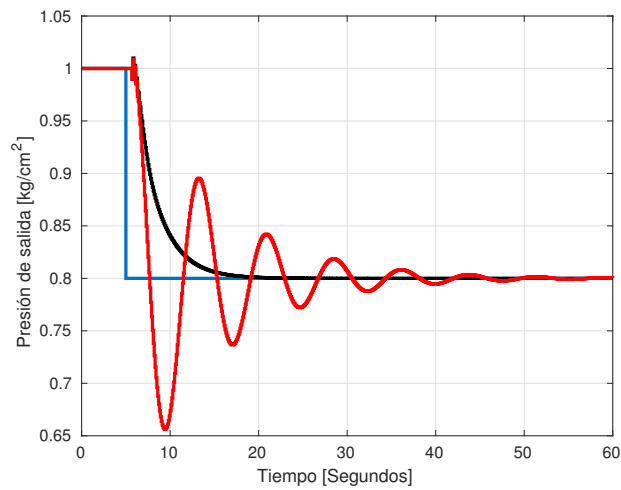
Frente a la entrada escalón en la referencia, aunque la respuesta se ha degradado fuertemente para el cambio $+10\%$, el sistema sigue siendo estable. A partir de los resultados de simulación, se puede observar que es necesario tener una buena identificación de la planta y también de los retardos de la misma para utilizar correctamente el predictor de Smith. De esta forma, se puede asegurar que el sistema a lazo cerrado tenga la respuesta dinámica diseñada.

Figura 13: Simulación del sistema con el controlador.



Fuente: Elaboración propia.

Figura 14: Simulación del sistema con variaciones en los parámetros de la planta.



Fuente: Elaboración propia.

7. Conclusiones

En este trabajo, se aplicó Internet de las cosas a control y monitoreo de un sistema de recuperación de energía. Para llevar a cabo lo anterior, se creó una pasarela para los protocolos de comunicación

Revista REDIUNP

Revista Electrónica de divulgación de Metodologías Emergentes en el Desarrollo de STEM
Aprobada en Consejo Directivo de la Facultad RCDFI-419-2018
ISSN:2683-8648
Vol. 3 N° 1(2021)



Valagao Sancho, M.; Ahrtz, G.D. y Sosa Tellechea, C.A.
Aplicación de IoT al control y monitoreo de un sistema de recuperación de energía.

industrial OPC UA y Modbus. Esto permitió conectar diferentes dispositivos Modbus a un servidor OPC UA. De esta forma, cualquier cliente OPC UA pudo acceder a las variables del sistema de recuperación de energía accediendo al servidor. Además, se diseñó un cliente OPC UA empleando el software Node-RED que permite monitorear el funcionamiento del sistema y almacenar los datos de los ensayos. Cabe destacar que estas herramientas fueron diseñadas empleando software libre, lo cual aseguró su bajo costo.

A partir del uso de IoT, se pudo caracterizar al sistema de recuperación de energía en diferentes puntos de funcionamiento mediante una aproximación por mínimos cuadrados. De igual manera las mediciones llevadas a cabo con el medidor de energía fueron analizadas y tenidas en cuenta para encontrar el rango de funcionamiento más adecuado para el sistema de recuperación de energía. Un controlador fue diseñado resolviendo una ecuación diofantina y el retardo fue tenido en cuenta en el diseño empleando un predictor de Smith. El controlador fue validado mediante simulaciones. Trabajos futuros incluyen la generación de código para la implementación de identificación en línea y la síntesis de un controlador adaptativo que pueda manejar de forma óptima al sistema de recuperación de energía bajo cualquier condición de funcionamiento.

Agradecimientos

Este trabajo se ha realizado con el soporte del proyecto de investigación (PI-1542), UNPSJB y la SCPL de Comodoro Rivadavia.

Referencias

- Cadiboni, A., Yncio, F., Fernández, R., Ahrtz, G., Peña, R., Sosa Tellechea, C., y Vásquez, M. (2018, Nov). Control and modeling of a centrifugal pump used as a turbine in an energy recovery system. En *2018 IEEE 9th Power, Instrumentation and Measurement Meeting (EPIM)* (p. 1-6). doi: 10.1109/EPIM.2018.8756435
- Carravetta, A. (2018). *Pumps as turbines : fundamentals and applications*. Cham, Switzerland: Springer.
- Carravetta, A., Fecarotta, O., y Ramos, H. (2018). A new low-cost installation scheme of pats for pico-hydropower to recover energy in residential areas. *Renewable Energy*, 125, 1003 - 1014. doi: <https://doi.org/10.1016/j.renene.2018.02.132>
- Cavalieri, S., y Chiacchio, F. (2013). Analysis of OPC UA performances. *Computer Standards & Interfaces*, 36(1), 165 - 177. doi: <https://doi.org/10.1016/j.csi.2013.06.004>
- Fontana, N., Giugni, M., Glielmo, L., y Marini, G. (2016). Real time control of a prototype for pressure regulation and energy production in water distribution networks. *Journal of Water Resources Planning and Management*, 142(7), 04016015. doi: 10.1061/(ASCE)WR.1943-5452.0000651
- Franklin, G. (1994). *Feedback control of dynamic systems*. Reading, Massachusetts: Addison-Wesley.
- Geng, H. (2016). *Internet of things and data analytics handbook*. Somerset: John Wiley & Sons, Incorporated.

- Graham, D., y Lathrop, R. C. (1953). The synthesis of 'optimum' transient response: Criteria and standard forms. *Transactions of the American Institute of Electrical Engineers, Part II: Applications and Industry*, 72(5), 273-288. doi: 10.1109/TAI.1953.6371346
- Gram-Hansen, R. (2016). Digital services and sustainable solutions. En *Internet of things and data analytics handbook* (p. 29-40). John Wiley & Sons, Ltd. doi: 10.1002/9781119173601.ch2
- Hughes, J. (2010). *Real world instrumentation with python*. Boston, MA: O'Reilly Media.
- iem3155*. (2019). Descargado de <https://www.se.com/ar/es/product/A9MEM3155>
- Kodali, R. K., y Anjum, A. (2018, Aug). IoT based home automation using node-red. En *2018 second international conference on green computing and internet of things (icgciot)* (p. 386-390). doi: 10.1109/ICGCIoT.2018.8753085
- Mahnke, W. (2009). *Opc unified architecture*. Berlin Heidelberg: Springer-Verlag.
- Modbus Foundation*. (2019). Descargado de <http://www.modbus.org/>
- Node-RED official webpage*. (2019). Descargado de <https://nodered.org/>
- OPC Foundation*. (2019). Descargado de <https://opcfoundation.org/>
- Peña, R. R., Fernández, R. D., Lorenc, M., y Cadiboni, A. (2019, Sep.). Gateway OPC UA/Modbus applied to an energy recovery system identification. En *2019 xviii workshop on information processing and control (rpic)* (p. 235-240). doi: 10.1109/RPIC.2019.8882175
- Peña, R. R., Fernández, R. D., Valagao Sancho, M., Lorenc, M. A., y Sosa Tellechea, C. (2020, Oct.). Control of a centrifugal pump as a turbine in an energy recovery system. En *2020 argentine conference on automatic control (aadeca)*.
- Ramos, H., y Borga, A. (1999). Pumps as turbines: an unconventional solution to energy production. *Urban Water*, 1(3), 261 - 263. doi: [https://doi.org/10.1016/S1462-0758\(00\)00016-9](https://doi.org/10.1016/S1462-0758(00)00016-9)
- Åström, K. J. (1995). *Adaptive control*. Reading, Mass: Addison-Wesley.
- Rico, J. E. (2007). *Control of dead-time processes*. Berlin London: Springer.
- Schleipen, M., Gilani, S.-S., Bischoff, T., y Pfrommer, J. (2016). OPC UA & industrie 4.0 - enabling technology with high diversity and variability. *Procedia CIRP*, 57, 315 - 320. (Factories of the Future in the digital environment - Proceedings of the 49th CIRP Conference on Manufacturing Systems) doi: <https://doi.org/10.1016/j.procir.2016.11.055>
- Slackware linux project*. (2019). Descargado de <http://www.slackware.com>
- Smith, O. J. (1958). *Feedback control systems*. New York: McGraw-Hill.
- Tanenbaum, A. (2015). *Modern operating systems*. Boston: Pearson.
- web OpenJS Foundation*. (2020). Descargado de <https://openjsf.org/projects/>
- Yncio, F., Peña, R., Cadiboni, A., Fernández, R., Ahrtz, G., y Sosa Tellechea, C. (2018, Nov). A modbus client for the identification of an energy recovery system for a water distribution network. En *2018 ieee 9th power, instrumentation and measurement meeting (epim)* (p. 1-6). doi: 10.1109/EPIM.2018.8756337