

10.30972/eitt.827243

Laboratorio remoto: Rejillas periódicas submicrométricas

Ortiz, G. P. ^{1 2}, Romero, R. H. ¹, Ponce, C. A. ¹, Toranzos, V. J. ³ y Ríos, L. J. ⁴ (*)

Resumen

Presentamos un diseño de laboratorio remoto compatible con los requerimientos para establecer un nodo FaCENA en la red R-LAB administrada por el CONFEDI. Consideramos los antecedentes sobre la puesta en servicios de laboratorios remotos a nivel nacional e internacional; definiciones de metas y alcances pedagógicos; formulaciones del problema del diseño y la realización de una propuesta preliminar. Proponemos un prototipo con definiciones sobre dispositivos, instrumentos de medición, diseño y fabricación, mediante una interfaz del tipo modular muy versátil para conectividad vía páginas web.

Palabras claves: Laboratorio-remoto, difracción, raspberry, linux-Debian, Perl-Dancer2.

Abstract

We present the design of a remotely operated laboratory compatible with the requirements for setting a node at FaCENA within the R-LAB collaboration network coordinated by CONFEDI. We considered the antecedents on national and international experiences on running remote laboratories, definition of goals and pedagogical scope. We propose a prototype giving specifications on devices, measurement instruments and construction, operated through a modular versatile interface connected via web pages.

Keywords: Remote-lab, diffraction, raspberry, linux-Debian, Perl-Dancer2.

1. Departamento de Física de la Facultad de Ciencias Exactas y Naturales y Agrimensura de la UNNE.

2. Correo electrónico: gortiz@exa.unne.edu.ar

3. Departamento de Ingeniería de la Facultad de Ciencias Exactas y Naturales y Agrimensura de la UNNE.

4. Departamento de Informática de la Facultad de Ciencias Exactas y Naturales y Agrimensura de la UNNE.

(*) Cómo citar este artículo: Ortiz, G. P., Romero, R. H., Ponce, C. A., Toranzos, V. J. y Ríos, L. J. (2023). *Laboratorio Remoto: Rejillas Periódicas Submicrométricas*. Revista Extensionismo, Innovación y Transferencia Tecnológica: claves para el desarrollo, 8(2), 14-26. <https://doi.org/10.30972/eitt.827243>

1. Introducción

Los antecedentes muestran que son factibles las puestas en servicio de laboratorios remotos desde hace algo más de una década [1]. Tanto a nivel internacional como nacional se ofrecen hoy en día laboratorios que funcionan prácticamente en forma continua [2]. Los reportes sobre las propuestas existentes muestran que son posibles las realizaciones de experimentos con el control necesario para la observación del mismo y la generación de datos [3]. En la mayoría de las propuestas se implementan páginas web para el análisis de resultados en el tiempo real de la realización del experimento y también para modalidades de tiempo diferido sobre ensayos ya realizados [4]. La versatilidad, precisión y cantidad de datos parecen ser acordes con las características no presenciales y en todo caso se han reportado mejoras para los laboratorios en los que se detectaron necesidades de ajustes en relación a esas cuestiones [1]. Sin embargo, las experiencias reportadas en comparación con las mismas actividades realizadas en forma presencial muestran ventajas y desventajas entre ambas modalidades [5]. En esencia, la realización presencial contiene posibilidades de aprendizaje relacionadas en el *¿Cómo se hace?* que en el caso del laboratorio remoto se encuentran ya resueltas para garantizar un mínimo de robustez y estabilidad operativa. En contraparte, esta previsibilidad impone también limitaciones en las prestaciones que puedan requerirse para responder a otras cuestiones no contempladas o nuevas ideas.

Las preguntas necesarias para establecer las metas pedagógicas de un laboratorio remoto son sobre el tipo y la cantidad de experiencias, así como su complejidad, cantidad de usuarios y precisión requerida por las mediciones. Una de las cuestiones principales es sobre el aporte del laboratorio remoto en la adquisición de habilidades y competencias por parte de sus usuarios.

En la formulación del problema del diseño abordamos cuestiones operativas que están íntimamente relacionadas con las metas pedagógicas establecidas previamente. Es decir, debe quedar claro cuál es el objetivo pedagógico para el que se diseña el laboratorio remoto. Se establece entonces una interacción entre las necesidades pedagógicas y posibilidades tecnológicas que deben ser trabajadas en forma conjunta.

1.1 Definiciones de metas pedagógicas

Teniendo en cuentas las pautas mencionadas sobre la factibilidad de la creación de un laboratorio remoto establecemos primero una meta pedagógica a partir de un eje temático de asignaturas de Física del ciclo básico de carreras de FaCENA-UNNE.

Presentamos el diseño de una propuesta didáctica con el fin de conceptualizar las propiedades ondulatorias de la luz. En este tema es de suma importancia articular

teoría y experimento, consideramos que resulta imprescindible la observación de los fenómenos estudiados a partir de simulaciones y/o experimentos. Una de las ventajas de la observación experimental por sobre las simulaciones numéricas es contar con condiciones empíricas para implementar el método científico. Mediante el cálculo de incertezas sobre mediciones se refuerzan los aprendizajes obtenidos en los laboratorios de asignaturas previas en la carrera. La presente propuesta pedagógica consiste en lograr una comprensión integral de un fenómeno transversal a varias asignaturas de Física, tal y como es el caso de superposición de ondas propagantes.

En esta instancia es importante considerar las características de los estudiantes que se constituyen en posibles usuarios del laboratorio remoto propuesto. Se supone que están cursando conjuntamente, o han cursado, las asignaturas del cálculo diferencial con contenidos mínimos donde se encuentran los temas teóricos necesarios para llevar a cabo las operaciones matemáticas del problema planteado. Por ejemplo, entre los conocimientos de un estudiante de las carreras de FaCENA, se espera que puedan realizar una regresión lineal de datos para encontrar una constante de proporcionalidad entre dos variables que dependen linealmente entre sí y su error cuadrático medio.

Los temas del trabajo a conceptualizar son los efectos de la coherencia espacial de las ondas electromagnéticas, superposición constructiva y destructiva. Interferencia y Difracción de la luz: la cuestión a dilucidar es la relación existente entre las propiedades ondulatorias de la luz y su interacción con rejillas periódicas, a través de la observación experimental de los puntos brillantes que se generan en una pantalla de observación con localizaciones regulares muy bien definidas, conocidas con el nombre de patrón o registro de difracción.

1.2. Formulación del problema de diseño

Proponemos un laboratorio remoto que permite al usuario desplazar mediante un motor paso a paso la posición de una rejilla (o red) de difracción por transmisión. Esta red se interpone en la trayectoria de un haz de luz proveniente desde un puntero láser y dirigiéndose hacia una pantalla de observación. El haz, al interactuar con la rejilla, se esparce iluminando esa pantalla con puntos brillantes que se observan mediante una cámara controlada por el mismo usuario. Se tiene previsto que el usuario pueda medir la separación entre aquellos puntos brillantes para diferentes distancias de la red de difracción a la superficie de observación. La figura 1 muestra un esquema del experimento propuesto.

El problema didáctico a resolver en primer lugar es encontrar qué relación existe en el fenómeno observado, definido por la separación D entre los dos primeros puntos brillantes para cada longitud L entre la rejilla de difracción y el plano de observación Σ . Conociendo esta relación, en una etapa posterior se propone investigar su conexión

con las otras escalas de tamaño relevantes del problema, esto es la separación entre líneas de la rejilla de difracción a y la longitud de onda λ del láser empleado.

Una cuestión para meditar es sobre los cuatro órdenes de magnitud de diferencia que existen entre las variables del experimento D y L que están en la escala del cm contra los parámetros a y λ que se encuentran en la escala micrométrica. Para guiar las ideas puede pensarse en los conceptos de superposición de ondas. Por ejemplo, aplicándolo a los casos de superposición constructiva y destructiva para dos fuentes coherentes y separadas por una distancia (como ser: a) comparable al tamaño de la longitud de onda λ y los efectos que pueden observarse a distancias muchos más grandes (como ser L) en direcciones específicas desde la mediatriz entre esas dos fuentes.

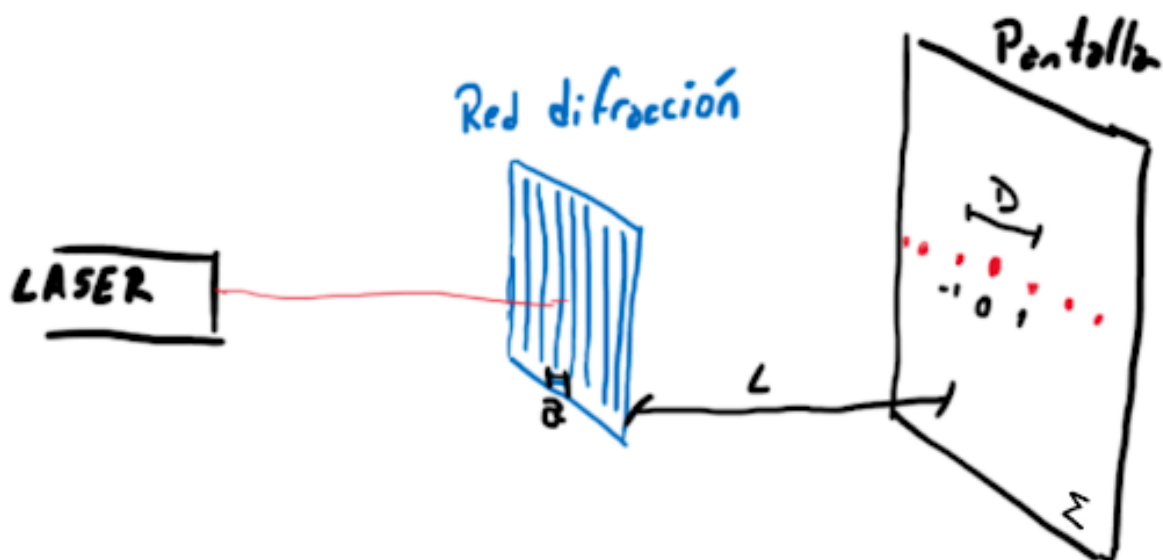


Fig. 1: Esquema del experimento de difracción donde se puede ver a la izquierda el puntero láser. El haz del puntero alcanza la red de difracción y se generan puntos brillantes en la pantalla de observación Σ . La separación entre líneas de la red es a , la distancia de la red al plano de observación es L y la distancia entre los primeros puntos brillantes es D , correspondientes a la distancia de separación entre los órdenes $m = -1$ y 1 de difracción.

Es deseable que el diseño experimental permita introducir más de una red de difracción y dos o tres punteros láseres, de este modo se podría contar con varios valores de los parámetros a y λ para lograr diferentes constantes de proporcionalidad entre D y L , no obstante, en una primer propuesta que permita una descripción completa del problema a resolver, consideraremos sólo una red y un puntero láser.

2. Diseño Preliminar del Laboratorio Remoto

El usuario ingresa al laboratorio remoto y observa desde una cámara web el dispositivo experimental. Puede identificar el puntero láser en la parte superior apuntando en dirección vertical hacia la parte inferior en la que encuentra primero la rejilla de difracción y luego una plataforma plana. El usuario puede prender y apagar el láser para verificar el funcionamiento y su interacción en tiempo real con el experimento.

También puede posicionar la rejilla de difracción a diferentes alturas respecto de la plataforma que tiene adherida una regla con escala milimétrica.

La ejecución del experimento es muy sencilla. Las acciones son prender o apagar el láser, cambiar la altura L de la rejilla de difracción y tomar una foto que contenga una clara visualización de los puntos brillantes y de la regla para determinar la relación entre píxeles y distancias en unidades de cm sobre la superficie de la plataforma. En la figura 2 se muestra una foto del prototipo experimental propuesto. El usuario puede determinar D definida por la distancia entre los órdenes 1 y -1 para cada L seleccionado. El laboratorio remoto le permite al usuario, para diferentes posiciones de la rejilla de difracción, generar una tabla de datos con los valores de L y D correspondientes, también le permite bajar esos datos a su computadora personal para analizarlos.

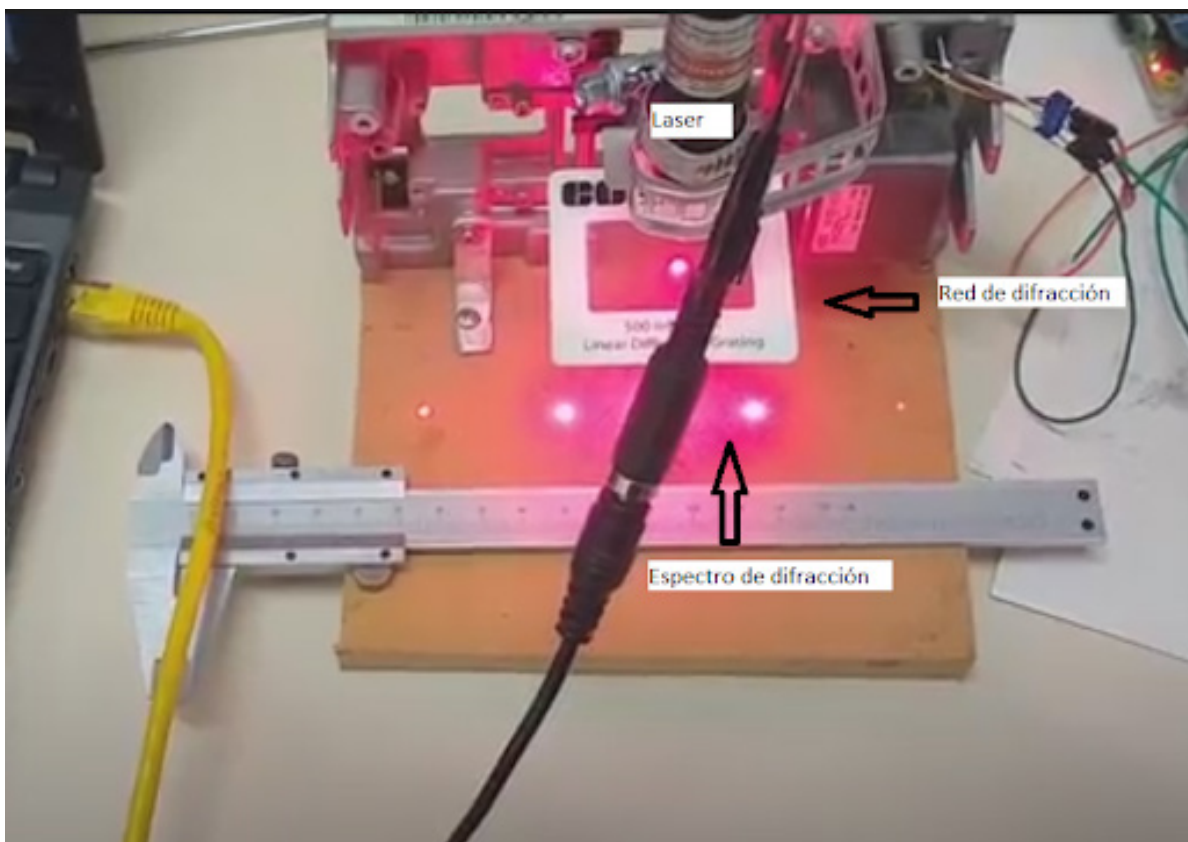


Fig. 2: Prototipo experimental en desarrollo para la implementación del laboratorio remoto de difracción de la luz por una rejilla de transmisión. Se puede ver el puntero láser en la parte superior iluminando la red de difracción y los primeros órdenes difractados sobre la superficie inferior.

2.1. Integración y componentes del dispositivo

Proponemos integrar la operación del laboratorio remoto con tecnología Raspberry PI 3 [6, 7]. Esta arquitectura por ser de código abierto es muy robusta y adecuada por sus capacidades computacionales y tiempos de respuesta para la interacción y comunicación entre periféricos locales y/o remotos. La versión actual de esta micro computadora cuenta con las opciones de 1, 2 y 4GB de RAM en un procesador BCM2711

de cuatro núcleos ARM de 64 bits a la frecuencia de 1.5GHz. El sistema operativo y el almacenamiento de datos se montan via tarjeta micro SD. Dispone de adaptadores de doble banda (2.4Ghz y 5.0 GHz) de red inalámbrica, Bluetooth 5.0 y conexión Ethernet para red por cable. Tiene 4 puertos USB, 2 puertos de video micro-HDMI, 1 puerto DSI para display, 1 puerto CSI para cámara y 1 puerto audio estéreo. Entrada para fuentes de alimentación de 5V DC via conectores USB-C, via GPIO y sobre Ethernet (PoE). El ingrediente principal de esta minicomputadora son las 40 conexiones para entradas y salidas de periféricos [8] conocido con el acrónimo GPIO (*del inglés General Propose Input Output*). Estas conexiones cubren muy diversos protocolos y puertos de comunicación entre los que podemos mencionar aquellos que se utilizan para controlar sensores y motores mediante el empleo de pulsos de anchos modulados y otras señales de comunicación y control [8]. En la figura 3a) se muestra un esquema del GPIO de las versiones PI 3 y PI 4 de la minicomputadora Raspberry. Se utiliza la nomenclatura BCM para numerar los pines. Todos los pines que se utilizan son inicializados como pines de salida. Los valores posibles son bajo y alto. Los pines 18, 19 y 22 se utilizan para comandar el motor. Los pines 23, 24 y 25 son para seleccionar el tamaño del paso menor al nominal. Con el pin 27 se comanda el laser.

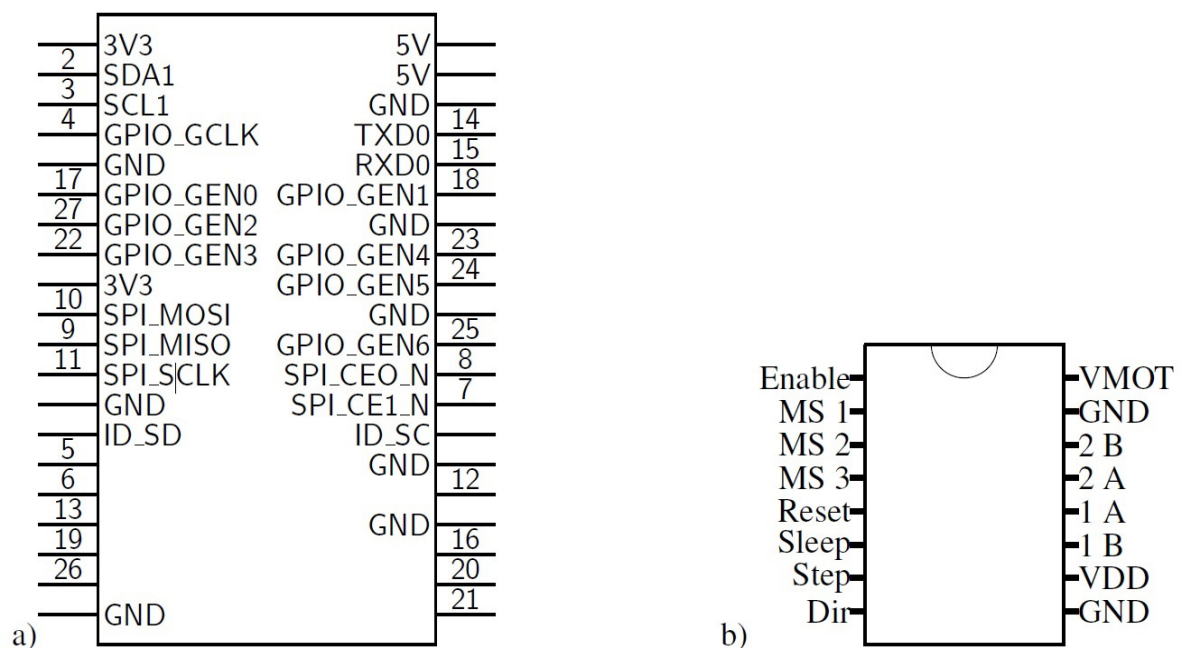


Fig 3. a) GPIO Raspberry PI. Los pines en nomenclatura BCM (GPIO). Se utilizan en modo de salida con valores binarios: alto o bajo. El pin 18, 19 y 22 se utilizan para comandar el motor. Los pines 23, 24 y 25 para seleccionar el tamaño del paso. El pin 27 opera el laser. Fig 3. b) Controlador del motor paso a paso pololu A4988. Los pines MS 1,2 y 3 se utilizan para definir tamaños de pasos menor al nominal. El pin Reset en alto permite la traslación. El pin Step ejecuta la traslación. El pin Dir permite seleccionar el sentido de giro del motor. Los pines VMOT y GND se utilizan para entregar la potencia al motor en el rango 8-35V. Los pares de pines A y B corresponden a las terminales para cada uno de los bobinados ortogonales 1 y 2. Los pines VDD se utiliza para alimentar este controlador desde la terminal de 3.3V de la Raspberry PI (primero a la izquierda en el panel a)) junto con el el pin GND para establecer una referencia de voltaje entre ambos dispositivos (ultimo a la izquierda en el panel a)).

El motor paso a paso es marca NEC 1.8DEG de 12V DC bipolar controlado mediante un integrado con puente H y corriente límite ajustable marca Allegro A4988 caracterizado por ser de traslación para realizar micropasos que facilita su operación, robustez y estabilidad [9]. En la figura 3b) se muestra un esquema de este integrado conocido también con el nombre pololu A4988, el cual se utiliza frecuentemente para controlar cabezales en impresoras 3D.

Se requiere un puntero laser de baja potencia y una pequeña fuente de corriente para alimentar este puntero. En la figura 4 se muestra un esquema para esta fuente. Se propone una configuración de seguimiento con un transistor NPN BC337. En esta configuración la tensión V_C , provista por la Raspberry PI, decae en 0.7V al pasar de B a E. Por lo tanto, la resistencia R se elige para determinar la corriente deseada para alimentar el diodo-laser en la parte de arriba de ese diagrama. La tensión en V_C es aproximadamente 3.5V por lo que un valor típico de $R=330\Omega$ determinan una corriente $i=8.8\text{mA}$. El valor de referencia de voltaje se establece al mismo nivel de la Raspberry PI y el controlador pololu conectando éste al pin GND de ambos. La configuración de seguimiento es deseable porque para una ganancia típicamente de dos órdenes de magnitud entre la corriente en la base B y el emisor E asegura una corriente con niveles de seguridad muy adecuados para el control desde la Raspberry.

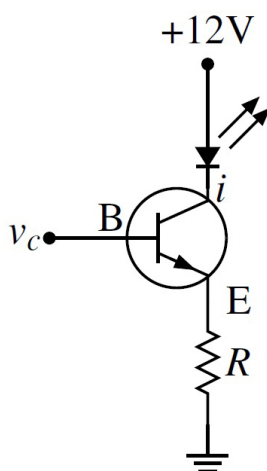


Fig. 4: Esquema del diagrama eléctrico de la fuente de corriente para el puntero diodo-laser. Se emplea un transistor NPN BC337 en configuración de seguimiento. La tensión de control V_C proviene del pin 27 de la Raspberry PI (figura 3a). Se elige R considerando una caída de 0.7V entre B y E para determinar la corriente i . La referencia de voltaje se establece al mismo nivel del controlador pololu y la Raspberry PI indicados con el pin GND en las figura3.

2.2 Diseño y fabricación

Determinadas las características necesarias de los dispositivos para el desplazamiento controlado de la red de difracción, establecemos las siguientes conexiones. El pin 18 de la Raspberry PI se conecta con el pin Step del integrado pololu. El pin 19 con el Dir y el pin 22 con el Reset, respectivamente. Estas

conexiones permitan mediante el código que describimos en el Apéndice, ejecutar los movimientos del motor paso a paso. El pin 18 al permutarse entre bajo y alto ejecuta cada paso. El pin 19 al cambiar entre bajo y alto conmuta las direcciones de giro en un sentido u otro. El pin 22 al estar conectado en estado alto permite todas las secuenciaciones mencionadas.

A diferencia de los motores empleados para transmisiones continuas de movimiento los motores, paso a paso emplean electroimanes que interactúan con un imán permanente orientado perpendicularmente y solidario al eje de giro del motor. Cada paso del giro consiste en un cambio sincronizado de la dirección de esa interacción magnética comandada por los electroimanes. Existen diferentes configuraciones y diseños de esos electroimanes básicamente constituidos por bobinados. El caso más simple es contar con dos bobinados, denominados 1 y 2, para inducir campos magnéticos en direcciones ortogonales entre sí y como se menciona más arriba también perpendiculares al eje de giro. El sentido y la intensidad de esos campos inducidos se controla siguiendo diferentes estrategias. En el motor paso a paso bipolar se controlan las tensiones de los extremos de cada bobina, denominadas terminales A y B. Un paso nominal del motor NEC que estamos utilizando es de 1.8° . Esto se establece porque la diferencia de fase entre las bobinas 1 y 2 son discretizadas con intervalo mínimo de las diferencias de potenciales entre sus terminales A y B. Mediante los pines MS 1, 2 y 3 se logra reducir el intervalo de discretización y codificar otras diferencias de potenciales que establecen pasos menores al nominal. Por ejemplo, conmutando a nivel alto los pines 23, 24 y 25 conectados respectivamente a los pines MS 1, 2 y 3 se puede dividir en una dieciseis avas parte el valor nominal de giro, es decir un micropaso de 0.1125° .

3. Interfaz a páginas web

La conectividad es uno de los puntos principales en la propuesta del laboratorio remoto. Implementamos el protocolo `http` del formato web para permitir a los usuarios conectarse al servidor que administra el laboratorio remoto y ejecutar las instrucciones descritas en el Apéndice. Para esto utilizamos el paquete de cómputo `Dancer2` escrito en lenguaje `PERL` y basado en el concepto de controladores de rutas [10]. Se trata de una programación orientada a objetos mediante el intermediario `Mojo` que le da seguridad y robustez a las interacciones entre diferentes instancias de comunicación.

Instalamos en la Raspberry PI los paquetes `Plack` [11] y `Dancer2` que son módulos de `PERL` para construir y administrar aplicaciones web. Una forma conveniente de hacer esta instalación es utilizando la herramienta `cpanm` que es una utilidad de `CPAN` para instalar códigos `PERL` [12]. Es posible que otras dependencias sean requeridas y el mismo instalador permita detectarlas e instalarlas automáticamente.

Una vez instalado, desde la línea de comandos y ejecutando `dancer2 gen -a labRemoto` se obtiene el directorio `./labRemoto` junto con los subdirectorios de

vista, archivos de configuración y ejecutables necesarios para montar la página web. Ubicándose en el directorio `./labRemoto` y ejecutando desde la línea de comando `plackup -p 5000 bin/app.psgi` se obtiene una vista de la página web generada en el URL `http://ip.de.la.RPI:5000` del navegador disponible en alguna PC que compartan misma red con la Raspberry PI empleando protocolo TCP, donde `ip.de.la.RPI` representa la dirección IP de la Raspberry PI.

Entre los detalles de la configuración mencionamos el que corresponde a la base de datos con la que se administra el historial de usuarios ingresados y los datos generados en el laboratorio remoto. En el subdirectorio `environments` del directorio `labRemoto` se encuentra el archivo `development.yml`. En ese archivo se configuran los complementos (comunmente conocidos con el nombre en inglés *plugins*) tal y como es en nuestro caso para el manejo de la base de datos. Por tal motivo se debe instalar en la Raspberry PI previamente el módulo `Dancer2::Plugin::Database` con la utileria mencionada `cpanm` y el paquete `sqlite3` con el sistema de repositorios del sistema operativo en uso, por ejemplo para Linux/Debian 11 sería `sudo apt install sqlite3`. En el archivo `development.yml` se deben incorporar las líneas de configuración para crear la base de datos, indicando su nombre, el programa administrador de la base de datos denominado en inglés *driver* que es el `SQLite` en nuestro caso. Existe una sintaxis específica para los parámetros de creación de la base de datos [13].

Los controladores de ruta tienen por objeto hacer que la aplicación web escrita en `dancer2` interprete los cinco métodos del protocolo `http` asignando un camino y una referencia a un código que regresa la respuesta.

4. Conclusiones

En este trabajo exponemos las etapas de definición de las características técnicas de un laboratorio remoto para su uso en asignaturas de física de ciclos básicos de carreras de ingeniería y ciencias de FACENA, hasta su implementación en un prototipo completamente funcional operable a través de un navegador web.

El prototipo satisface los requerimientos pedagógicos de uso, monitoreo y repetibilidad en tiempo real a fin de proporcionar una experiencia cercana a la de un laboratorio presencial. La electrónica de alimentación y control usa componentes estandar comercialmente disponibles, integradas mediante un dispositivo Raspberry PI para procesamiento de administración, monitoreo, control de las partes móviles, y comunicación via internet.

La experiencia de difracción de luz a través de una red de transmisión, elegida como ejemplo de aplicación de la tecnología de operación remota, puede ser extendida a otras funcionalidades de mayor complejidad y permite la posibilidad de uso extendido de estas innovaciones didácticas.

Referencias

- [1] Javier García Zubía and Gustavo R. Alves, editors. *Using Remote Labs in Education*. University of Deusto, Bilbao, 2011.
- [2] <https://www.ises.info/index.php/en> *International School Experimental System*.
- [3] Faisal Mahmud, Mohd Nashrul Mohd Zubir, Mohd Yazed Bin Ahmad, Zuraidah Binti Abdullah, Nahrizul Adib Bin Kadri, Unaizah Hanum Binti Obaidellah, and Tousif Amin Faisal. Development of web based real time remote laboratory for teaching and learning. *IOP Conference Series: Materials Science and Engineering*, 854(1):012025, may 2020.
- [4] Francisco Esquembre. Facilitating the creation of virtual and remote laboratories for science and engineering education. *IFAC-PapersOnLine*, 48(29):49–58, 2015. IFAC Workshop on Internet Based Control Education IBCE15.
- [5] C.S. Tzafestas, N. Palaiologou, and M. Alifragis. Virtual and remote robotic laboratory: comparative experimental evaluation. *IEEE Transactions on Education*, 49(3):360–369, 2006.
- [6] <https://www.raspberrypi.org/>.
- [7] <https://github.com/raspberrypi/documentation>.
- [8] <http://www.mosaic-industries.com/embedded-systems/microcontroller-projects/raspberry-pi/gpio-pin-electrical-specifications>.
- [9] Allegro MicroSystems LLC. *DMOS Microstepping Driver with Translator and Overcurrent Protection*, 2021. <https://www.allegromicro.com/en/products/motordrivers/brush-dc-motor-drivers/a4988>.
- [10] <https://metacpan.org/dist/Dancer2/view/lib/Dancer2/Manual.pod>.
- [11] <https://metacpan.org/dist/Plack/view/script/plackup>.
- [12] <https://metacpan.org/dist/App-cpanminus/view/bin/cpanm>.
- [13] <https://metacpan.org/pod/Dancer2::Plugin::Database>

5. Apéndice: Código de control del posicionamiento de la red de difracción, del puntero láser y de la cámara web

A continuación describiremos el código empleado para interactuar con los periféricos mencionados y que forma parte del módulo `labRemoto.pm` que se encuentra en el subdirectorio `lib` del directorio `labRemoto` creado con `dancer2` como se menciona en la Sec. 3. Debe notarse en el ejemplo de abajo que empleando el comando `system` se establece una salida a la interfase de comandos del tipo `bash` para comandar el GPIO y los periféricos. El uso del lenguaje `Perl-PDL` se establece por compatibilidad con la interfase `dancer2`.

```
#código que permite encender/apagar el laser, tomar y guardar imágenes,
#desplazar hacia arriba/abajo la red de difracción
#y nuevamente encender/apagar el laser y tomar/guardar imagenes
#
use strict;
use warnings;
use PDL::Constants qw(PI);
system("gpio -g mode 18 output;");#step
system("gpio -g mode 19 output;");#dir
system("gpio -g mode 27 output;");#laser
system("gpio -g mode 22 output;");#reset
system("gpio -g mode 23 output;");#ms1
system("gpio -g mode 24 output;");#ms2
system("gpio -g mode 25 output;");#ms3
```

Las primeras líneas definen paquetes de ayudas para la sintaxis y la constante π . Luego inicializa los pines a utilizar en modo de salida. Todas las instrucciones son ejecutadas a través del `bash` del sistema operativo de la Raspberry PI, lo cual destaca versatilidad en implementación de algún lenguaje deseado.

Las líneas siguientes corresponden al encendido del laser, la introducción de un tiempo de espera de 0.5 s y la toma de una imagen mediante el comando `raspistill` con la opción `-roi` que permite determinar una región de interés y la opción `-o` para definir un archivo de salida `rem1.jpg` y guardar la imagen tomada.

```
system("gpio -g write 27 1;");#turn on laser
delay(0.5);
system("raspistill -roi 0,0,0.8,0.7 -o rem1.jpg;"); #toma una foto
system("gpio -g write 27 0;");#turn off laser
```

Y luego apagar el laser comandando a V_c de la figura 4 al valor de referencia (GND).

A continuación, se establece el pin 22 en nivel alto para permitir todas las operaciones de traslación del controlador pololu (Reset). Luego se fija el micropaso en 0.1125° mediante los niveles altos de los pines 23, 24 y 25.

```
system("gpio -g write 22 1;"); #reset
system("gpio -g write 23 1;"); #MS 1
system("gpio -g write 24 1;"); #MS 2
system("gpio -g write 25 1;"); #MS 3
```

A continuación se establece el pin 18 en nivel bajo (0) y el sentido del giro mediante el pin 19. Notar el ciclo `for` para incrementar de uno en uno cada paso permutando el pin 18 entre bajo (0) y alto (1). Notar que se introduce un tiempo de espera de $50 \mu s$.

```
system("gpio -g write 18 0;"); #step
system("gpio -g write 19 1;"); #dir
my $r=8.425; # radio de giro en mm
my $v=47.5; # altura vertical mínima desde  $\Sigma$  hasta la
red
printf "Pos. Vertical: %.3f \n", $v;
for(0..1240) {
system("gpio -g write 18 1;");
delay(0.00005);
system("gpio -g write 18 0;");
delay(0.00005);
$v+=0.1125*PI/180*$r;
}
printf "Pos. Vertical: %.3f \n", $v;
```

Entre la permutación de los niveles del pin 18. Esto se hace teniendo en cuenta el tiempo de respuesta posible del integrado pololu [9] y la rapidez deseada para el movimiento de la red de difracción desde un extremo al otro del recorrido total, un poco mas de 2cm. En cada paso ese movimiento corresponde a un desplazamiento $0.1125*PI/180*\$r$ de la red (aproximadamente 16.5μ) porque esta recibe la transmisión mediante una cinta que se enrolla en un radio $\$r=8.425$ mm desde el eje de giro del motor paso a paso. Luego de alcanzar el paso final la posición vertical se obtiene del acumulado de todos los pasos sobre la altura inicial de $\$v$ que se imprime en pantalla con formato de número real con 3 decimales.

```
system("gpio -g write 27 1;"); #turn on laser
delay(0.5);
system("raspistill -roi 0,0,0.8,0.7 -o rem2.jpg;");
system("gpio -g write 27 0;");
```

Luego, se enciende el laser, se toma una nueva imagen que se guarda ahora en `rem2.jpg` y se apaga el laser.

Las líneas siguientes muestran el cambio de dirección en el movimiento.

```
system("gpio -g write 19 0;");# dir
for(0..1200){
system("gpio -g write 18 1;");# step
delay(0.00005);
system("gpio -g write 18 0;"); #step
delay(0.00005);
$v-=0.1125*PI/180*$r;
}
```

El ciclo `for` es equivalente al de mas arriba, pero al conmutar primero el pin 19 a nivel bajo se invierte el sentido de giro del motor logrando que la red de difracción regrese a la posición inicial. La posición vertical se obtiene luego de descontar los desplazamientos sobre todos los pasos dados. Por lo que debería verificarse que $v=47.5$ mm. Las líneas siguientes ya la hemos descripto antes, se agregan solo por completez.

```
printf "Pos. Vertical: %.3f \n", $v;
system("gpio -g write 27 1;"); #laser
delay(0.5);
system("raspistill -roi 0,0,0.8,0.7 -o rem3.jpg;");
system("gpio -g write 27 0;");
system("gpio -g write 22 0;");#reset
```

Para verificar el regreso a la posición inicial y el correcto funcionamiento esperado mediante el encendido del laser, la toma de una nueva imagen `rem3.jpg` que debe arrojar resultados idénticos a los de `rem1.jpg`, el apagado del laser y la desactivación del integrado pololu permutando a nivel bajo el pin 22 que libera el motor. Las líneas siguientes definen la subrutina `delay`.

```
sub delay{
my $time=shift;
select(undef, undef, undef, $time);
```