PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ

ESCUELA DE POSGRADO

AUTOMATIZACIÓN DE PROCESOS DE GENERACIÓN Y MEDICIÓN DE FASES GEOMÉTRICAS

Tesis para Optar el grado de Magíster en Física

AUTOR

Francis Luis Verástegui Salvador

ASESOR

Francisco de Zela

JURADO

XXXXXXXXXXXXXXXXX

XXXXXXXXXXXXXXXXXX

LIMA – PERÚ

2016

RESUMEN

Dentro de los diversos temas que investiga el grupo de óptica cuántica de la PUCP, se encuentra el estudio teórico y experimental de fases geométricas. La fase geométrica es estudiada a través de los métodos interferométrico y polarimétrico que dependen de la disposición de los componentes ópticos (placas retardadoras, polarizadores, etc.) según el arreglo experimental. Con estos componentes ópticos se logran manipular los estados de polarización de la luz. Para ello es de suma importancia controlar de manera precisa la ubicación angular de estos componentes ópticos. Esta posición angular debe ser variada continuamente, dependiendo del arreglo experimental que se esté usando, y se deben de registrar los datos de la intensidad del haz del láser.

Actualmente, estas posiciones angulares de las placas retardadoras, polarizadores, etc., son fijadas manualmente, lo cual genera dos problemas serios: si el arreglo consta de muchas placas retardadoras, entonces las mediciones toman mucho tiempo en ser realizadas; de otro lado, a los errores propios de los instrumentos se agregan fuentes de error experimental. Por estos motivos es que para el laboratorio de Óptica Cuántica resulta esencial tener un sistema automatizado el cual garantice que las tomas de mediciones sean más rápidas, la ubicación angular de los componentes ópticos sea precisa y que no se agreguen más fuentes de error. Una vez lograda esta automatización, realizada en LabVIEW, puede ser aplicada a diversos arreglos ópticos como por ejemplo en la calibración de placas retardadoras y de polarizadores.

El objetivo principal de esta tesis es la realización de un programa en donde todo el sistema de medición de fases geométricas esté automatizado, desde el movimiento de los motores rotatorios, el registro de la posición angular en la que se posiciona y hasta la medición de la intensidad del láser.

Esta tesis va dedicada a mis padres, mi hermano y Mily,
quienes son las personas más importantes en mi vida.

AGRADECIMIENTOS

Agradezco infinitamente a mis padres, David y Milagros, y a mi hermano David (junior) quienes siempre me motivan a seguir y a cumplir las metas que me he trazado. Sin el apoyo incondicional de mi familia nada de esto sería posible.

Agradezco muchísimo a Mily por aguantar mis malos humores, por la falta de tiempo, por apoyarme y animarme a no darme por vencido cuando creía que todo estaba perdido.

Quiero agradecer al Profesor Francisco de Zela por haber aceptado ser mi asesor de tesis y a la profesora Maria Elena López por brindarnos todas las facilidades que se encontraban a su alcance para que podamos estudiar con tranquilidad.

También agradezco enormemente a Omar Ortíz y Yonny Yugra por ayudarme a entender tanto la parte teórica como la parte experimental para que este trabajo sea posible.

Adicionalmente, agradezco a CONCYTEC que ha sido la institución que ha financiado este programa de maestría. ¡Muchas gracias!

También agradezco a mis amigos Alex Vega y Harold Orihuela quienes me ayudaron en momentos claves para finalizar esta tesis.

También agradezco a mis amigos Trigosinos de la promoción 2003, quienes siempre están ahí cuando uno los necesita… ¡les dije que los incluiría!

Y no quisiera olvidarme de agradecer al Club Universitario de Deportes por tantas alegrías que me ha dado en toda mi vida… ¡Y dale U!

ÍNDICE GENERAL

1. **Introduccion** **1**

2. **Marco teórico: Estados de polarización** **2**

2.1) INTERFERÓMETRO MACH-ZEHNDER 2
2.2) SUPERPOSICIÓN DE P-ESTADOS 3
2.3) PLACAS RETARDADORAS 4
2.4) PARÁMETROS DE STOKES Y ESFERA DE POINCARÉ 6
2.4.1) Parámetros De Stokes 6
2.4.2) La Esfera De Poincaré 8
2.5.2) FASES DE BERRY Y PANCHARATNAM 9
2.6.2) ARREGLO EXPERIMENTAL 11

3. **Marco experimental: Instrumentos, software y programación del sistema** **15**

3.1. COMPONENTES AGILIS: MOTOR PIEZOELÉCTRICO E INTERFACE DE CONEXIÓN MULTI-EJE 15
3.1.2. Motor piezoeléctrico 15
3.1.2. Interface de conexión multi-eje 16
3.2. SOFTWARE LABVIEW 17
3.2.1. Entorno 17
3.2.2. Panel frontal 18
3.2.3. Diagrama de bloques 22
3.3. PROGRAMACIÓN DEL MOTOR ROTATORIO EN LABVIEW 23
3.4. ACELERÓMETRO ADXL345 25
3.5. ARDUINO 26
3.5.1. El hardware 27
3.5.2. El entorno de programación 28
3.5.3. Comunidad 28
3.5.4. Conexión del acelerómetro ADXL345al arduino 29
3.5.5. Librería que conecta a arduino con LabVIEW 29
3.6. PROGRAMACIÓN DEL ACELERÓMETRO ACOPLADO AL MOTOR ROTATORIO 31
3.6.1. Programa para que LabVIEW lea los datos del acelerómetro ADXL345 31
3.6.2. Programa para que LabVIEW convierta bytes en valores de aceleración de la gravedad en cada eje coordenado y en grados angulares 32
3.6.3. Programa para leer datos desde un archivo de texto y procesarlos 33
3.6.4. Programa para girar al motor rotatorio y guardar las mediciones realizadas por el acelerómetro 34
3.6.5. Programa general para controlar los acelerómetros acoplados a los motores rotatorios y realizar mediciones de intensidad del láser 35

4. **Resultados De La Calibración Y Mediciones** **37**

4.1 CALIBRACIÓN DE PLACAS RETARDADORAS 37
4.2 MEDICIONES 39

5. **Conclusiones** **42**

6. **Bibliografia** **43**

7. **Apéndice: Cómo utilizar la librería LINX en LabVIEW** **44**

ÍNDICE DE FIGURAS

Figura 1. Interferómetro Mach-Zehnder. 2
Figura 2. Orientación del eje rápido y lento, x' e y', de la placa retardadora. 6
Figura 3. Esfera de Poincaré. 8
Figura 4. Motor piezoeléctrico NEWPORT. 15
Figura 5. Interface del motor piezoeléctrico. 16
Figura 6. LabVIEW 2014. 17
Figura 7. Panel Frontal (a) y Diagrama de Bloques (b). 17
Figura 8. Paleta de Controles. 18
Figura 9. Controles de la sub-opción Modern. 19
Figura 10. Controles de la sub-opción Numeric. 19
Figura 11. Controles de la sub-opción Classic. 20
Figura 12. Controles de la sub-opción Numeric. 21
Figura 13. Controles de la sub-opción Express, sub-opción Num Ctrls. 21
Figura 14. Paleta de Funciones. 22
Figura 15. Paleta de estructuras de programación de la opción Structures. 23
Figura 16. Algoritmo del movimiento del motor rotatorio en sentido horario (positivo). 24
Figura 17. Algoritmo del movimiento del motor rotatorio en sentido anti-horario (negativo). 24
Figura 18. Diseño del programa para controlar al motor rotatorio en sentido horario o anti-horario. 24
Figura 19. Acelerómetro ADXL345. 25
Figura 20. Orientación de los ejes coordenados en el sensor. 26
Figura 21. Respuesta de salida vs. Orientación de la gravedad. 26
Figura 22. Placa Arduino UNO. 28
Figura 23. Vista del entorno de programación de Arduino. 28
Figura 24. Conexión del acelerómetro ADXL345 a la placa del Arduino UNO. 29
Figura 25. Instalación de LINX mediante el JKI VI Package Manager 30
Figura 26. Acelerómetro acoplado al motor rotatorio. 30
Figura 27. Algoritmo para leer los datos que recibe LabVIEW del acelerómetro. 31
Figura 28. Diseño del programa para leer los datos que recibe LabVIEW del acelerómetro. 32
Figura 29. Algoritmo para convertir los bytes en valores de aceleración de la gravedad en cada eje coordenado; y se calcula en qué posición angular se encuentra ubicado. 32
Figura 30. Diseño del programa que nos da el valor de la aceleración de la gravedad en cada eje coordenado y el ángulo en el cual se encuentra ubicado el acelerómetro. 33
Figura 31. Algoritmo para leer datos de un archivo de texto y calcular promedios y desviaciones estándar. 33
Figura 32. Diseño del programa para leer datos de un archivo de texto y calcular promedios y desviaciones estándar. 33
Figura 33. Algoritmo realizado para ubicar al acelerómetro acoplado al motor rotatorio a posicionarse en un ángulo predeterminado. 34
Figura 34. Diseño del programa para posicionar al motor rotatorio acoplado al acelerómetro en el ángulo que se desea. 35
Figura 35. Datos de entrada. 35
Figura 36. Programa para controlar los ocho motores rotatorios y realizar mediciones de la intensidad del láser. 36
Figura 37. Calibración de placas retardadoras 37
Figura 38. Programa para calibrar placas retardadoras 38
Figura 39. Calibración de una placa retardadora - Posición angular vs. Intensidad. 38
Figura 40. Arreglo polarimétrico para calcular la fase de Pancharatnam Φp. 39

Figura 41. Arreglo polarimétrico armado en el Laboratorio (vista posterior). 39
Figura 42. Arreglo polarimétrico armado en el Laboratorio (vista frontal). 40
Figura 43. Interferograma. Probabilidad de detección vs. ∅. 40

CAPÍTULO 1

Introducción

La fase geométrica fue originalmente descubierta por Berry en 1984. Berry consideró una evolución cíclica, adiabática y unitaria de sistemas cuánticos dependientes del tiempo que obedecen la ecuación de Schrödinger. Luego el concepto fue generalizado, libre de las anteriores restricciones. Su existencia incluso en situaciones posiblemente no cíclicas, no adiabáticas y no unitarias ha sido demostrada. El trabajo original de Berry desencadenó un enorme interés en muchas áreas de la física, como por ejemplo en resonancia magnética [1], sistemas atómicos [2], interferometría con neutrones [3], etc. Muchas demostraciones experimentales de la fase geométrica también han sido realizadas en el contexto de la óptica clásica, donde se aprovechan los estados de polarización de la luz [4]. También se han producido numerosos estudios especialmente dentro de la óptica cuántica [5].

En el Laboratorio de Óptica Cuántica de la PUCP se realizan mediciones de fases geométricas a través de dos métodos: interferometría y polarimetría. Generalmente el método interferométrico es muy sensible a las perturbaciones mecánicas y/o térmicas que normalmente afectan al arreglo óptico y para contrarrestar estas perturbaciones se suelen utilizar métodos de estabilización que muchas veces no son sencillos de implementar.

Por otro lado, el método polarimétrico es robusto frente a este tipo de perturbaciones, lo cual es muy ventajoso. Sin embargo, para implementar éste método es necesario emplear gran cantidad de componentes ópticos, como lo son las placas retardadoras y polarizadores. Estos componentes ópticos deben ser colocados en una posición angular muy precisa, calculada teóricamente, y dependiendo del arreglo que se vaya a implementar los elementos ópticos deben ser fijados en otras posiciones angulares cubriendo un rango de valores entre 0° y 180°.

Actualmente este trabajo de ir fijando las posiciones angulares de los componentes ópticos se realizan manualmente, lo cual conlleva que las mediciones tomen mucho tiempo en realizarse y además a que se agreguen fuentes de error experimental, aparte de los errores propios de los equipos con los que se trabaja. Aprovechando que el Laboratorio de Óptica Cuántica cuenta con motores piezoeléctricos rotatorios, el software LabVIEW y un medidor de intensidad de láser (potenciómetro), se planteó la alternativa de automatizar todo este proceso: controlar hasta ocho componentes ópticos y registrar mediciones de intensidad del láser. Todo ello podría controlarse mediante un programa hecho en LabVIEW, al cual sólo se le ingresa datos de entrada, los cuales son las posiciones angulares que se calculan previamente.

El programa realizado se podría aprovechar para diversos arreglos experimentales, tanto para el método interferométrico como para el polarimétrico, y también para calibrar placas retardadoras y polarizadores. Por otro lado, en este trabajo se tomó un arreglo polarimétrico realizado por el grupo de Óptica Cuántica de la PUCP que ha sido publicado en Physical Review [6] el año 2009.

En el próximo capítulo se presenta un breve resumen de los conceptos básicos que son necesarios para entender el arreglo polarimétrico empleado. Tales conceptos son los estados de polarización de la luz, parámetros de Stokes, la acción de los retardadores y su representación en la esfera de Poincaré. Además se dará la definición de las fases de Berry y Pancharatnam.

CAPÍTULO 2

Marco teórico: Estados de polarización

2.1) Interferómetro Mach-Zehnder

El interferómetro Mach-Zehnder permite combinar dos estados para producir otro estado, que es superposición de los anteriores. El arreglo se muestra en la Fig. 1. M_1 y M_2 son espejos semitransparentes (divisores de haz) y M y M' son totalmente reflectores. Un haz monocromático que incide sobre M_1 es dividido y recombinado en M_2. A y A' son absorbedores cuya única función es controlar la intensidad de la luz en cada brazo. Los caminos $M_1M'M_2$ y M_1MM_2 son escogidos de tal manera que cada haz de luz dentro del aparato viaje la misma distancia. P, R y L son los filtros polarizadores.

El haz de luz que incide sobre M_1 está en un estado que denotamos como $|P\rangle$ dado que ha atravesado un filtro P. Si los absorbedores están variando de tal manera que sólo el haz superior es transmitido por el aparato, el haz de luz de salida estará en un estado que denotamos como $|R\rangle$; si sólo el haz inferior es transmitido, el haz de luz emergente estará en un estado denotado como $|L\rangle$. Ahora, si cada absorbedor permite pasar a los haces en su totalidad entonces el haz emergente estará en un estado $|P\rangle$, igual al haz incidente.

Este experimento muestra que un estado $|P\rangle$ puede ser representado de alguna manera como una superposición de los estados $|R\rangle$ y $|L\rangle$, es decir $|P\rangle = |R\rangle + |L\rangle$. Esto nos dice que un estado $|P\rangle$ puede ser obtenido por la superposición de un estado $|R\rangle$ y un estado $|L\rangle$.

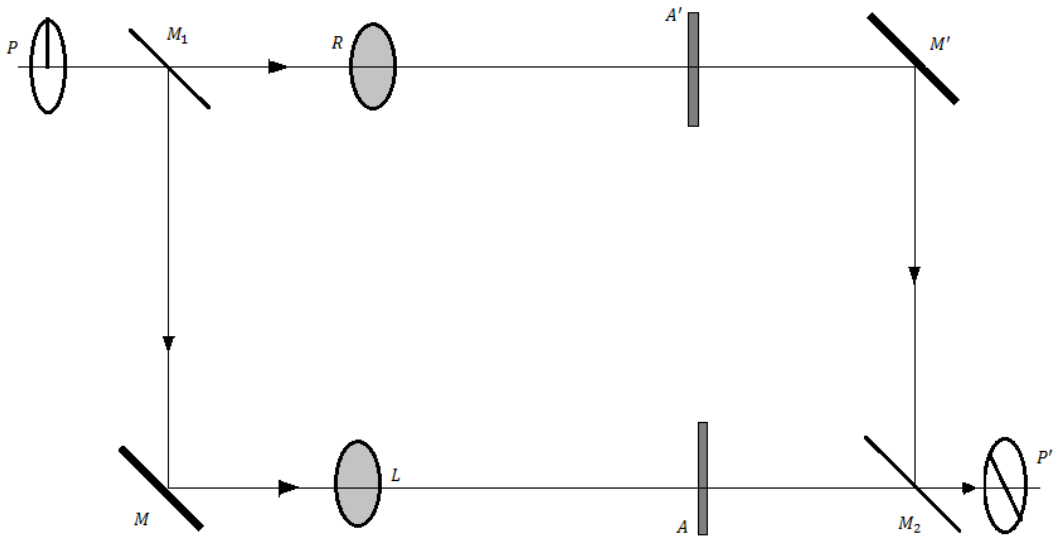

Figura 1. Interferómetro Mach-Zehnder.

Pero una simple superposición de un estado $|R\rangle$ y un estado $|L\rangle$ no necesariamente dan un estado $|P\rangle$. De hecho, una sucesión continua de estados de $|R\rangle$ a $|P\rangle$ a $|L\rangle$ puede ser producido por la variación del absorbedor de un brazo del aparato de 0 a 100% de transmisión, mientras que el absorbedor del otro brazo varía de 100 a 0%. Los absorbedores determinan cuánto de cada haz de luz contribuye a la superposición de estados, por ello la superposición la denotamos como $|E\rangle = a|R\rangle + b|L\rangle$, donde las constantes a y b dan la cantidad proporcional de $|R\rangle$ y $|L\rangle$ que contribuyen a la combinación final.

2.2) Superposición de P-estados

Hasta ahora se ha escrito los estados denotados por $|E\rangle$ como una superposición de los estados $|R\rangle$ y $|L\rangle$, pero esta no es la única posibilidad. Cuando un estado de polarización es escrito como una superposición de P-estados con las bases [7]:

$$|P_x\rangle = \begin{pmatrix}1\\0\end{pmatrix} \quad ; \quad |P_y\rangle = \begin{pmatrix}0\\1\end{pmatrix} \tag{1}$$

entonces el vector que representa al estado es llamado el vector de Jones.

Ahora, sean $|E_1\rangle$ y $|E_2\rangle$ dos estados ortogonales que se superponen:

$$|E\rangle = A|E_1\rangle + B|E_2\rangle \tag{2}$$

Las constantes A y B son números complejos y pueden ser escritos de la forma:

$$A = ae^{i\gamma} \qquad B = be^{i\rho} \tag{3}$$

Con a, b, γ y ρ reales. Esta superposición puede ser escrita como:

$$|E\rangle = e^{i\alpha}\{ae^{-i\beta/2}|E_1\rangle + be^{+i\beta/2}|E_2\rangle\} \tag{4}$$

donde

$$\alpha = \frac{1}{2}(\gamma + \rho), \qquad \beta = \rho - \gamma \tag{5}$$

El factor $e^{i\alpha}$ con frecuencia no es considerado y la forma estándar de la superposición coherente de dos estados ortogonales es la ecuación (4) sin considerar el factor $e^{i\alpha}$.

Se denota a $|P_x\rangle$ y $|P_y\rangle$ como estados puros, los cuales han sido preparados por filtros tipo P cuyos ejes están a lo largo de los ejes x e y. Usando estos dos estados como una base, se puede expresar cualquier estado $|E\rangle$ como:

$$|E\rangle = a_x e^{-i\phi/2}|P_x\rangle + a_y e^{+i\phi/2}|P_y\rangle \tag{6}$$

Se analiza el estado $|E\rangle$ usando filtros tipo P cuyo eje hace un ángulo θ con respecto al eje x, y por lo tanto un ángulo $(\pi/2 - \theta)$ con el eje y.

$$\langle P|E\rangle = a_x e^{-i\phi/2}\langle P|P_x\rangle + a_y e^{+i\phi/2}\langle P|P_y\rangle$$

$$\langle P|E\rangle = a_x e^{-i\phi/2}cos\theta + a_y e^{+i\phi/2}\cos(\pi/2 - \theta) \tag{7}$$

$$\langle P|E\rangle = a_x e^{-i\phi/2}cos\theta + a_y e^{+i\phi/2}\text{sen}\theta$$

La intensidad puede ser calculada de la siguiente manera:

$$I = |\langle P|E\rangle|^2 \tag{8}$$

Para nuestro caso se tiene:

$$|\langle P|E\rangle|^2 = a_x^2 cos^2\theta + a_y^2 sen^2\theta + a_x a_y cos\theta sen\theta cos\phi \tag{9}$$

Esta ecuación (9) indica la transmisión de un estado general $|E\rangle$ a través de un filtro tipo P.

Tomando la derivada, se obtiene:

$$\frac{\partial}{\partial\theta}|\langle P|E\rangle|^2 = -2a_x^2 cos\theta sen\theta + 2a_y^2 sen\theta cos\theta \\ +2a_x a_y cos^2\theta cos\phi - 2a_x a_y sen^2\theta cos\phi \qquad (10)$$

Igualamos a cero para encontrar el valor máximo y el valor mínimo, se obtiene:

$$\left(\frac{a_y}{a_x} - \frac{a_x}{a_y}\right) + (cot\theta - tan\theta)cos\phi = 0 \qquad (11)$$

Si $cos\phi = 0$, ($\phi = \pm\pi/2$), entonces a_x debe ser igual a a_y, y normalizando el estado a la unidad, se obtiene que $a_x = a_y = 1/\sqrt{2}$.

Cuando $\phi = \pi/2$, el estado corresponde a polarización circular derecha y cuando $\phi = -\pi/2$ el estado corresponde a polarización circular izquierda.

Entonces, matemáticamente se obtiene que:

$$|R\rangle = \frac{1}{\sqrt{2}}\{e^{-i\pi/4}|P_x\rangle + e^{+i\pi/4}|P_y\rangle\}$$
$$|L\rangle = \frac{1}{\sqrt{2}}\{e^{+i\pi/4}|P_x\rangle + e^{-i\pi/4}|P_y\rangle\} \qquad (12)$$

El caso general de la polarización elíptica se da cuando $a_x \neq a_y$ y ϕ debe tomar un valor entre el rango de 0 y $\pi/2$.

2.3) <u>Placas retardadoras</u>

El efecto de una placa retardadora sobre un estado $|E\rangle$ es matemáticamente representado por un factor multiplicativo de fase. Por lo tanto, si un haz en un estado representado por $|E\rangle$ atraviesa una placa retardadora, el estado emergente estará representado por $|E'\rangle = e^{iks}|E\rangle$, donde $s = t(n-1)$; en el cual n y t son el índice de refracción y el grosor de la placa retardadora, respectivamente; y $k = 2\pi/\lambda$, donde λ es la longitud de onda. Entonces, el estado queda representado de la siguiente manera:

$$|E'\rangle = e^{ikt(n-1)}|E\rangle \qquad (13)$$

Consideremos ahora un haz de luz preparado en una superposición de dos estados $|P_x\rangle$ y $|P_y\rangle$ como:

$$|E\rangle = A|P_x\rangle + B|P_y\rangle \qquad (14)$$

donde A y B son complejos.

Se inserta una placa retardadora de tal manera que su eje rápido sea paralelo al eje x y su eje lento sea paralelo al eje y. Además, denotamos como n_s y n_f a los índices de refracción en el eje lento y rápido, respectivamente. Luego de que el haz ha atravesado la placa birrefringente, su estado será:

$$|E'\rangle = Ae^{ikt(n_f-1)}|P_x\rangle + Be^{ikt(n_s-1)}|P_y\rangle$$
$$|E'\rangle = e^{-ikt}\{Ae^{iktn_f}|P_x\rangle + Be^{iktn_s}|P_y\rangle\} \tag{15}$$

El cual puede ser escrito de la siguiente manera:

$$|E'\rangle = e^{-i\tau}\{Ae^{-i\rho/2}|P_x\rangle + Be^{+i\rho/2}|P_y\rangle\} \tag{16}$$

donde

$$\tau = kt\left[1 - \frac{n_s + n_f}{2}\right]$$
$$\rho = kt[n_s - n_f] \tag{17}$$

siendo ρ el retraso relativo entre una y otra componente, que genera la placa birrefringente. Por lo tanto, si $t[n_s - n_f] = \lambda/2$ (placa de media onda de longitud) entonces $\rho = \pi$; sin embargo, si $t[n_s - n_f] = \lambda/4$ (placa de un cuarto de onda de longitud) entonces $\rho = \pi/2$.

La matriz que representa el efecto de la placa retardadora (siempre y cuando el eje rápido y el eje lento sean paralelos a los ejes x e y, respectivamente) será:

$$M = \begin{pmatrix} e^{-i\rho/2} & 0 \\ 0 & e^{+i\rho/2} \end{pmatrix} \tag{18}$$

Sin embargo, si los ejes rápido y lento son paralelos a los ejes y y x, su matriz será:

$$M^* = \begin{pmatrix} e^{+i\rho/2} & 0 \\ 0 & e^{-i\rho/2} \end{pmatrix} \tag{19}$$

Por otro lado, si el eje rápido de la placa retardadora hace un ángulo θ con el eje x, como se muestra en la Figura 2, la representación se obtiene primero aplicando una matriz R de rotación al estado $|E\rangle$, obteniéndose el estado $|E'\rangle = R|E\rangle$, luego se aplica la matriz M al estado $|E'\rangle$, obteniéndose el estado representado por $|E''\rangle = M|E'\rangle = MR|E\rangle$, y por último para volver al sistema original se aplica la matriz inversa R^{-1}, obteniéndose el estado final representado por $|E'''\rangle = R^{-1}MR|E\rangle = M'|E\rangle$, donde $M' = R^{-1}MR$.

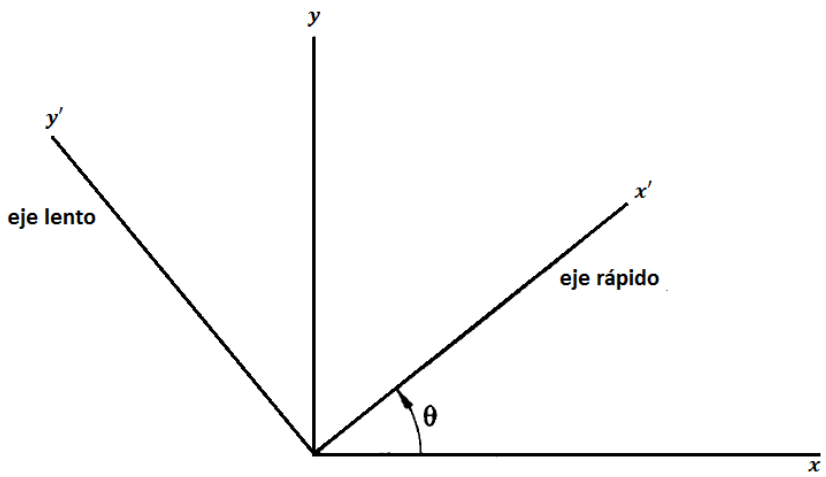

Figura 2. Orientación del eje rápido y lento, $\mathbf{x}'$ e $\mathbf{y}'$, de la placa retardadora.

La matriz de rotación R está representada por:

$$R = \begin{pmatrix} cos\theta & sen\theta \\ -sen\theta & cos\theta \end{pmatrix} \tag{20}$$

Y la matriz M' estará representada por:

$$M' = \begin{pmatrix} e^{-i\rho/2}cos^2\theta + e^{+i\rho/2}sen^2\theta & -2isen(\rho/2)cos\theta sen\theta \\ -2isen(\rho/2)cos\theta sen\theta & e^{-i\rho/2}sen^2\theta + e^{+i\rho/2}cos^2\theta \end{pmatrix} \tag{21}$$

2.4) <u>Parámetros de Stokes y esfera de Poincaré</u>

2.4.1) <u>Parámetros de Stokes</u>

El operador de proyección es una matriz de 2x2, que puede representarse de la siguiente manera [8]:

$$M(E) = |E\rangle\langle E| = \frac{1}{2}[P_0\sigma_0 + P_1\sigma_1 + P_2\sigma_2 + P_3\sigma_3] \tag{22}$$

donde P_0 y P_i $(i = 1,2,3)$ son escalares y σ_i son las matrices de Pauli.

$$\sigma_0 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad \sigma_1 = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}, \quad \sigma_2 = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad \sigma_3 = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} \tag{23}$$

Dado que $M(E)$ y las matrices de Pauli son hermíticas, los coeficientes escalares de las matrices de Pauli deben ser reales, tal que:

$$M(E) = \frac{1}{2}\sum_i P_i\sigma_i\,, \; M^\dagger(E) = \frac{1}{2}\sum_i P_i^*\sigma_i \qquad (i = 0,1,2,3) \tag{24}$$

Por lo tanto se debe cumplir, dado que $M(E)$ es hermítico:

$$\sum_i P_i\sigma_i = \sum_i P_i^*\sigma_i \tag{25}$$

Multiplicando a ambos lados de esta última ecuación por σ_j, y tomando la traza, se obtiene que $P_j = P_j^*$. Consecuentemente, cuando $M(E)$ es expresado como una combinación lineal de las matrices de Pauli, el operador de proyección es únicamente determinado por cuatro números reales (P_0, P_1, P_2, P_3) los cuales determinan el estado de polarización. Cuando el operador de proyección $M(E)$ es expresado en términos de $|P_x\rangle$ y $|P_y\rangle$ como una base, estos parámetros reales son llamados parámetros de Stokes. Ellos permiten una descripción del haz de luz en términos de un vector columna 4x1.

Se calcula los parámetros de Stokes para cualquier estado de polarización puro, de la siguiente manera:

$$|E\rangle = A|P_x\rangle + B|P_y\rangle; \qquad A = ae^{-i\phi/2} \quad y \quad B = be^{i\phi/2} \tag{26}$$

Entonces:

$$\langle E| = \langle P_x|A^* + \langle P_y|B^*$$

$$M(E) = \begin{pmatrix} AA^* & AB^* \\ BA^* & BB^* \end{pmatrix} = \frac{1}{2}\sum_i P_i\sigma_i \tag{27}$$

Utilizando las siguientes igualdades [9]:

$$\langle E|A|E\rangle = Tr[M(E)A], \qquad Tr(\sigma_i\sigma_j) = 2\delta_{ij} \tag{28}$$

se obtienen los siguientes resultados:

$$Tr[M(E)\sigma_j] = \frac{1}{2}Tr\sum_i \sigma_i\sigma_j P_i = P_j$$

$$\langle E|\sigma_j|E\rangle = P_j \tag{29}$$

Utilizando las ecuaciones (26) y (29), los parámetros de Stokes son:

$$P_0 = a^2 + b^2$$

$$P_1 = a^2 - b^2$$

$$P_2 = 2abcos\phi \tag{30}$$

$$P_3 = 2absen\phi$$

Representando lo anterior mediante un vector columna de 4x1, queda representado de la siguiente manera:

$$\begin{pmatrix} P_0 \\ P_1 \\ P_2 \\ P_3 \end{pmatrix} = \begin{pmatrix} a^2 + b^2 \\ a^2 - b^2 \\ 2abcos\phi \\ 2absen\phi \end{pmatrix} \tag{31}$$

Observamos que se cumple lo siguiente:

$$P_1^2 + P_2^2 + P_3^2 = (a^2 - b^2)^2 + 4a^2b^2cos^2\phi + 4a^2b^2sen^2\phi$$

$$P_1^2 + P_2^2 + P_3^2 = a^4 - 2a^2b^2 + b^4 + 4a^2b^2$$

$$P_1^2 + P_2^2 + P_3^2 = a^4 + 2a^2b^2 + b^4$$

$$P_1^2 + P_2^2 + P_3^2 = (a^2 + b^2)^2$$

$$P_1^2 + P_2^2 + P_3^2 = P_0^2 \tag{32}$$

2.4.2) <u>La esfera de Poincaré</u>

La ecuación (26) cumple con la condición de normalización, de tal manera que $\langle E|E\rangle = 1$, entonces:

$$\langle E|E\rangle = 1 = (a^2 + b^2)^2 = P_0^2 \tag{33}$$

Entonces la ecuación (32), puede ser escrita como:

$$1 = P_1^2 + P_2^2 + P_3^2 \tag{34}$$

Esto motiva considerar una esfera de radio 1, con los ejes coordenados P_1, P_2, P_3. Esta esfera es conocida como la esfera de Poincaré.

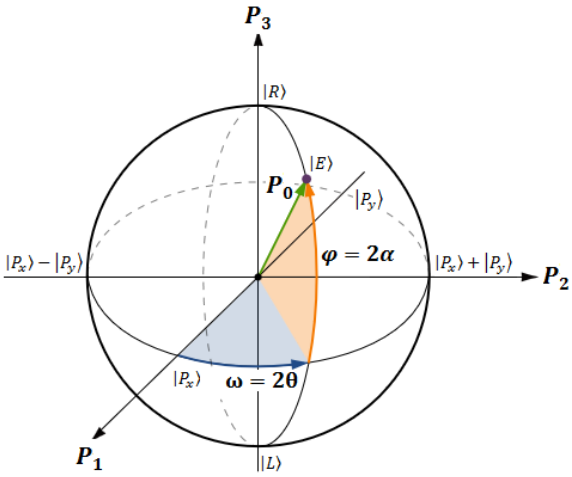

Figura 3. Esfera de Poincaré.

Los estados de polarización se encuentran ubicados en cualquier punto sobre la superficie de la esfera. Los estados de polarización lineal están sobre el ecuador, mientras que los estados de polarización circular derecha y circular izquierda se ubican sobre los polos norte y sur, respectivamente. En los puntos restantes se ubican los estados de polarización elíptica.

Con referencia a la esfera de Poincaré se pueden obtener los parámetros de Stokes en función de los ángulos polares:

$$P_1 = P_0 \cos(2\alpha)\cos(2\theta)$$

$$P_2 = P_0 \cos(2\alpha)\,\mathrm{sen}(2\theta) \qquad (35)$$

$$P_3 = P_0 sen(2\alpha)$$

2.5.2) <u>Fases de Berry y Pancharatnam</u>

En 1984 Berry estudió un sistema cuántico sometido a la acción de una evolución cíclica y unitaria bajo la acción de un Hamiltoniano dependiente del tiempo. El proceso correspondía al caso adiabático, es decir que el tiempo de la evolución del sistema era mucho más corto que el tiempo de evolución del Hamiltoniano. Berry descubrió que además de la fase dinámica que adquiría el sistema, existía una fase adicional a la que llamó fase geométrica. Esta fase geométrica, a diferencia de la fase dinámica, no podía ser eliminada mediante la aplicación de una transformada de fase.

Pancharatnam, anticipándose al descubrimiento de Berry, en 1956 propuso un método para determinar si dos haces polarizados se encontraban en fase. El método de Pancharatnam fue totalmente experimental, basado en la observación de si la intensidad de un interferograma formado por dos haces de luz era máxima. En tal caso estos haces se encontraban en fase. Esta fase, llamada "fase total", resultó tener una relación directa con la fase geométrica descubierta por Berry. A continuación veremos cómo, desde el punto de vista cinético, se definen la fase total, la dinámica y la geométrica.

Consideremos un espacio de Hilbert $\mathcal{H}$ en el cual se define $\mathcal{H}_0 \subset \mathcal{H}$ como un conjunto de vectores normalizados distintos de cero de tal manera que $|\Psi\rangle \in \mathcal{H}$. Además se define una curva $\mathcal{C}_0$ en $\mathcal{H}_0$ a través de vectores $|\Psi(s)\rangle$ que dependen del parámetro $s \in [s_1, s_2]$.

Debido a que $|\Psi(s)\rangle$ es normalizado, se cumple [10]:

$$\langle\Psi(s)|\dot{\Psi}(s)\rangle + \langle\dot{\Psi}(s)|\Psi(s)\rangle = 0 \qquad (36)$$

Entonces:

$$Re\langle\Psi(s)|\dot{\Psi}(s)\rangle = 0 \qquad \text{y} \qquad \langle\Psi(s)|\dot{\Psi}(s)\rangle = Im\langle\Psi(s)|\dot{\Psi}(s)\rangle \qquad (37)$$

Ahora consideremos el punto inicial $|\Psi(s_1)\rangle$ y el final $|\Psi(s_2)\rangle$ de la curva $\mathcal{C}_0$. De acuerdo a Pancharatnam, la fase total entre estos estados está dada por $\Phi_{tot}(\mathcal{C}_0) = arg\langle\Psi(s_1)|\Psi(s_2)\rangle$. Bajo una transformación gauge $|\Psi(s)\rangle \to |\Psi'(s)\rangle = exp(i\alpha(s))|\Psi(s)\rangle$, de manera que $\mathcal{C}_0 \to \mathcal{C}_0'$, se tiene:

$$\Phi_{tot}(\mathcal{C}_0) \to \Phi'_{tot}(\mathcal{C}_0) = arg\langle\Psi(s_1)|\Psi(s_2)\rangle + \alpha(s_2) - \alpha(s_1) \qquad (38)$$

Además:

$$Im\langle\Psi(s)|\dot{\Psi}(s)\rangle \rightarrow Im\langle\Psi'(s)|\dot{\Psi}'(s)\rangle = Im\langle\Psi(s)|\dot{\Psi}(s)\rangle + \dot{\alpha}(s) \qquad (39)$$

A partir de estas propiedades se puede definir la fase geométrica, la cual resulta ser invariante gauge:

$$\Phi_g(\mathcal{C}_0) = arg\langle\Psi(s_1)|\Psi(s_2)\rangle - Im\int_{s_2}^{s_1}\langle\Psi(s)|\dot{\Psi}(s)\rangle ds \qquad (40)$$

Además de ser invariante bajo re-parametrización, Φ_g también resulta ser invariante gauge. Es decir, que a pesar de haber sido definido en términos de $|\Psi(s)\rangle$ y $\mathcal{C}_0$, Φ_g efectivamente depende de una clase de equivalencia de $|\Psi(s)\rangle$ y $\mathcal{C}_0$, respectivamente. De hecho, el conjunto $\{|\Psi'\rangle = \exp(i\alpha)|\Psi\rangle\}$ constituye una clase de equivalencia y el espacio generado por esta clase de equivalencias es llamado el "espacio de rayos" $\mathcal{R}_0$. En lugar de trabajar con clases de equivalencias, trabajaremos con proyectores: $|\Psi\rangle\langle\Psi|$. El conjunto $\{|\Psi'\rangle = \exp(i\alpha)|\Psi\rangle\}$ proyecta sobre $|\Psi\rangle\langle\Psi|$ por medio de una proyección π: $\mathcal{H}_0 \rightarrow \mathcal{R}_0$.

En particular, las curvas $\mathcal{C}_0$ y $\mathcal{C}_0'$ están relacionadas mediante una transformación gauge, las cuales son también parte de una clase de equivalencia. Bajo la proyección π, ellos proyectan sobre una curva $C_0 \subset \mathcal{R}_0$, y de hecho, Φ_g no es una funcional de $\mathcal{C}_0$, sino que es una funcional de C_0 la curva definida por $|\Psi(s)\rangle\langle\Psi(s)|$. Esta es la razón de por qué Φ_g es una fase geométrica, ya que está asociada con la curva $C_0 \subset \mathcal{R}_0$. Entonces se debe escribir Φ_g de una mejor manera, aunque para calcularlo se requiere escoger un "lift" de C_0; es decir, se debe escoger una curva $\mathcal{C}_0$ de manera que $\pi(\mathcal{C}_0) = C_0$. Por lo tanto, $\Phi_g(C_0)$ es definido en términos de dos fases:

$$\Phi_{tot}(\mathcal{C}_0) = arg\langle\Psi(s_1)|\Psi(s_2)\rangle$$

Y (41)

$$\Phi_{dyn}(\mathcal{C}_0) = Im\int_{s_2}^{s_1}\langle\Psi(s)|\dot{\Psi}(s)\rangle ds$$

donde $\Phi_{tot}(\mathcal{C}_0)$ es la fase total o la fase de Pancharatnam de $\mathcal{C}_0$, el cual es el argumento α del número complejo $\langle\Psi(s_1)|\Psi(s_2)\rangle = |\langle\Psi(s_1)|\Psi(s_2)\rangle|e^{i\alpha}$; y $\Phi_{dyn}(\mathcal{C}_0)$ es la fase dinámica de $\mathcal{C}_0$. Puede observarse que tanto la fase dinámica como la fase total son funcionales de $\mathcal{C}_0$, sin embargo la resta entre ambas, la fase geométrica, es funcional de $C_0 = \pi(\mathcal{C}_0)$.

$$\Phi_g(C_0) = \Phi_{tot}(\mathcal{C}_0) - \Phi_{dyn}(\mathcal{C}_0) \qquad (42)$$

Se debe destacar que esta definición de la fase geométrica no se basa en las asunciones realizadas por Berry, debido a que $\Phi_g(C_0)$ ha sido introducido en términos de evolución de vectores de estado $|\Psi(s)\rangle$ y esta evolución no necesariamente es unitaria ni adiabática. Además, la curva $\mathcal{C}_0$ puede ser una curva abierta.

Dada una curva $C_0 \subset \mathcal{R}_0$, se pueden escoger diferentes "lifts" para calcular $\Phi_g(C_0)$ y aprovechar esa libertad para expresar $\Phi_g(C_0)$ de acuerdo a las necesidades requeridas. Por ejemplo, se puede hacer $\Phi_{tot}(\mathcal{C}_0) = 0$ escogiendo de manera adecuada la fase de $|\Psi(s_2)\rangle$. En este caso, $\Phi_g(C_0) = -\Phi_{dyn}(\mathcal{C}_0)$. Alternativamente, se puede hacer $\Phi_{dyn}(\mathcal{C}_0) = 0$, de tal manera que $\Phi_g(C_0) = \Phi_{tot}(\mathcal{C}_0)$ escogiendo un "lift horizontal" satisfaciendo que $Im\langle\Psi(s)|\dot{\Psi}(s)\rangle = 0$. Debido a que $Re\langle\Psi(s)|\dot{\Psi}(s)\rangle = 0$, entonces se cumple que $\langle\Psi(s)|\dot{\Psi}(s)\rangle = 0$.

Para conseguir un "lift horizontal" se puede aplicar, en caso sea necesario, un "lift" $|\Psi(s)\rangle \to |\Psi'(s)\rangle = \exp(i\alpha)|\Psi(s)\rangle$, de manera que $Im\langle\Psi(s)|\dot{\Psi}(s)\rangle \to Im\langle\Psi'(s)|\dot{\Psi}'(s)\rangle = Im\langle\Psi(s)|\dot{\Psi}(s)\rangle + \dot{\alpha}(s)$. Entonces se requiere que $Im\langle\Psi'(s)|\dot{\Psi}'(s)\rangle = 0$, y asumiendo que $\alpha(s_1) = 0$, es decir, fijando $|\Psi'(s_1)\rangle = |\Psi(s_1)\rangle$ al escoger de manera adecuada la fase inicial, se logra obtener:

$$\alpha(s) = -Im\int_{s_2}^{s_1}\langle\Psi(s)|\dot{\Psi}(s)\rangle ds \tag{43}$$

Debido a que $\Phi_g(C_0)$ depende sólo de las cantidades del espacio de rayos, debe ser posible obtener una expresión que refleje este hecho. Esta expresión puede ser obtenida al considerar el operador $K(s) = \dot{\rho}(s) = d\left(\left|\langle\Psi(s_1)|\dot{\Psi}(s_2)\rangle\right|\right)/ds$, cuya acción sobre $|\Psi(s)\rangle$ da:

$$K(s)|\Psi(s)\rangle = |\dot{\Psi}(s)\rangle - \langle\Psi(s)|\dot{\Psi}(s)\rangle|\dot{\Psi}(s)\rangle \tag{44}$$

Donde $K(s)$ es invariante gauge. Por lo tanto, la ecuación (44) es invariante gauge bajo cantidades transformadas. Escogiendo un "lift horizontal" $\langle\Psi(s)|\dot{\Psi}(s)\rangle = 0$, resulta:

$$\frac{d}{ds}|\Psi(s)\rangle = \dot{\rho}(s)|\Psi(s)\rangle \tag{45}$$

La solución de la ecuación (45) puede realizarse mediante las series de Dyson: $|\Psi(s)\rangle = P\left(exp\int_{s_1}^{s}\dot{\rho}(s)ds\right)|\Psi(s_1)\rangle$, donde P es el operador "parameter-ordering". Haciendo el "lift horizontal", la fase geométrica se reduce a $\Phi_g(C_0) = \Phi_{tot}(\mathcal{C}_0) = arg\langle\Psi(s_1)|\Psi(s_2)\rangle$. Ahora, $\langle\Psi(s_1)|\Psi(s_2)\rangle = Tr|\Psi(s_2)\rangle\langle\Psi(s_1)|$, y escogiendo $|\Psi(s_2)\rangle = P\left(exp\int_{s_1}^{s_2}\dot{\rho}(s)ds\right)|\Psi(s_1)\rangle$, se obtiene:

$$\Phi_g(C_0) = argTr\left\{P\left(exp\int_{s_1}^{s_2}\dot{\rho}(s)ds\right)\rho(s_1)\right\} \tag{46}$$

La ecuación (46) expresa la fase geométrica en términos de proyectores únicamente.

2.6.2) <u>Arreglo experimental</u>

A continuación se desarrollarán los principios básicos que sustentan la técnica de polarimetría para medir la fase geométrica.

Se considera un estado polarizado inicial denotado por $|i\rangle = |+\rangle_z$ y se le aplica una rotación de $\pi/2$ alrededor del eje perpendicular x. Como resultado se obtiene el estado $(|+\rangle_z - i|-\rangle_z)/\sqrt{2}$. Ahora, se le aplica a este estado el operador $exp(-\phi\,\sigma_z/2)$, obteniéndose el estado:

$$V|+\rangle_z \equiv exp(-\phi\,\sigma_z/2)exp(-\pi\,\sigma_x/4)|+\rangle_z$$

$$V|+\rangle_z \equiv \left(e^{-i\phi/2}|+\rangle_z - ie^{i\phi/2}|-\rangle_z\right)/\sqrt{2} \tag{47}$$

$$V|+\rangle_z \equiv e^{-i\phi/2}\left(|+\rangle_z - ie^{i\phi}|-\rangle_z\right)/\sqrt{2}$$

Por lo tanto, se ha generado una fase relativa ϕ entre los estados representados por $|+\rangle_z$ y $|-\rangle_z$, lo cual es similar a lo que se consigue en un interferómetro por el cambio de longitud de uno de los caminos ópticos. Luego, aplicando $U \in SU(2)$ se obtiene el siguiente estado $UV|+\rangle_z = \left(e^{-i\phi/2}U|+\rangle_z - ie^{i\phi/2}U|-\rangle_z\right)/\sqrt{2} \equiv |\chi_+\rangle_z + |\chi_-\rangle_z$. A partir de este último estado se puede extraer la fase de Pancharatnam midiendo la intensidad. Con el fin de lograr este objetivo, se proyecta $|\chi_+\rangle_z$ y $|\chi_-\rangle_z$ sobre el estado $V|+\rangle_z$, de manera que la intensidad del estado proyectado es:

$$I = \left|{}_z\langle +|V^\dagger(|\chi_+\rangle + |\chi_-\rangle)\right|^2 \tag{48}$$

Se denota $V|+\rangle_z = \left(e^{-i\phi/2}U|+\rangle_z - ie^{i\phi/2}U|-\rangle_z\right)/\sqrt{2} \equiv |\varphi_+\rangle_z + |\varphi_-\rangle_z$ y se toma la siguiente parametrización:

$$U(\beta;\gamma;\delta) = e^{i\left(\frac{\delta+\gamma}{2}\right)\sigma_z}e^{-i\beta\sigma_y}e^{i\left(\frac{\delta-\gamma}{2}\right)\sigma_z} = \begin{pmatrix} e^{i\delta}cos\beta & -e^{i\gamma}sen\beta \\ e^{-i\gamma}sen\beta & e^{i\delta}sen\beta \end{pmatrix} \tag{49}$$

Al calcular la amplitud ${}_z\langle +|V^\dagger(|\chi_+\rangle + |\chi_-\rangle) = (\langle\varphi_+| + \langle\varphi_-|)(|\chi_+\rangle + |\chi_-\rangle)$, y utilizando las siguientes relaciones:

$$\langle\varphi_\pm|\chi_\pm\rangle = \frac{e^{\pm i\delta}\cos(\beta)}{2} \quad ; \quad \langle\varphi_\mp|\chi_\pm\rangle = \frac{ie^{[\mp i(\gamma+\phi)]}sen(\beta)}{2} \tag{50}$$

se obtiene:

$$(\langle\varphi_+| + \langle\varphi_-|)(|\chi_+\rangle + |\chi_-\rangle) = \cos(\beta)\cos(\delta) + isen(\beta)\cos(\gamma+\phi) \tag{51}$$

La ecuación (51) contiene la fase de Pancharatnam $\delta = \Phi_{tot}$, que puede ser determinada a partir de medir la intensidad. De hecho, se observa en la ecuación (51) que las intensidades mínima y máxima están dadas por:

$$I_{min} = \cos^2(\beta)\cos^2(\delta) \tag{52}$$

y

$$I_{max} = \cos^2(\beta)\cos^2(\delta) + sen^2(\beta) \tag{53}$$

Entonces la fase de Pancharatnam es:

$$cos^2(\delta) = \frac{I_{min}}{1-I_{max}+I_{min}} \tag{54}$$

El arreglo experimental empleado fue propuesto por J.C. Loredo, O. Ortíz, R. Weingärtner y F. de Zela [6], el cual paso a detallar.

En el arreglo óptico se implementan U y V como placas retardadoras. Para comenzar:

$$exp(-i\pi\sigma_x/4) = Q\left(\frac{\pi}{4}\right) \qquad ; \qquad exp(-i\pi\sigma_z/2) = Q\left(\frac{\pi}{4}\right)H\left(\frac{\phi-\pi}{4}\right)Q\left(\frac{\pi}{4}\right) \tag{55}$$

Donde Q y H se refieren a "quarter" y "half"; es decir, retardadores de cuarto y media onda, respectivamente.

Usamos las siguientes relaciones:

$$Q^2\left(\frac{\pi}{4}\right) = H\left(\frac{\pi}{4}\right) \qquad ; \qquad exp(+i\pi\sigma_z/2) = Q\left(-\frac{\pi}{4}\right)H\left(\frac{\phi+\pi}{4}\right)Q\left(-\frac{\pi}{4}\right) \tag{56}$$

Se calcula $U_{tot} \equiv V^{\dagger}UV$

$$U_{tot} = H\left(-\frac{\pi}{4}\right)H\left(\frac{\phi+\pi}{4}\right)Q\left(-\frac{\pi}{4}\right).U.Q\left(\frac{\pi}{4}\right)H\left(\frac{\phi-\pi}{4}\right)Q\left(\frac{\pi}{4}\right) \tag{57}$$

Se emplea la siguiente forma de U que puede ser traducido directamente en un arreglo de placas retardadoras:

$$U(\xi;\eta;\zeta) = Q\left(\frac{-3\pi+2\xi}{4}\right)H\left(\frac{\xi-\eta-\zeta-\pi}{4}\right)Q\left(\frac{\pi-2\zeta}{4}\right) \tag{58}$$

Al insertar la ecuación (58) dentro de la ecuación (57), se obtiene un arreglo que consiste de 9 placas retardadoras. Con el fin de reducir el número de las placas retardadoras, se aplican las siguientes relaciones $Q(\alpha)H(\beta) = H(\beta)Q(2\beta-\alpha)$ y $Q(\alpha)H(\beta)H(\gamma) = Q(\alpha+\pi/2)H(\alpha-\beta+\gamma-\pi/2)$. La reducción final consiste en un arreglo de cinco placas retardadoras.

$$\begin{aligned} U_{tot} = Q\left(-\frac{3\pi}{4}-\frac{\phi}{2}\right)Q\left(-\frac{5\pi+2\xi}{4}-\frac{\phi}{2}\right)Q\left(-\frac{9\pi+2(\xi+\eta)}{4}-\frac{\phi}{2}\right) \\ xH\left(-\frac{7\pi+\xi+\eta-\zeta}{4}-\frac{\phi}{2}\right)Q\left(-\frac{\pi}{4}-\frac{\phi}{2}\right) \end{aligned} \tag{59}$$

Hay que tener en cuenta que este arreglo podría implementarse mediante el montaje de cinco placas retardadoras que tienen un eje de rotación en común, de modo que todas las placas retardadoras se pueden girar el mismo ángulo $\phi/2$ a través de algún aparato mecánico.

La intensidad que se debe medir depende de ξ, η y ζ de acuerdo con la siguiente expresión:

$$I = \left|{}_z\langle +|V^{\dagger}|+\rangle_z\right|^2$$

$$I = cos^2\left(\frac{\eta}{2}\right)cos^2\left(\frac{\xi+\zeta}{2}\right) + \left[cos\left(\frac{\eta}{2}\right)sen\left(\frac{\xi+\zeta}{2}\right)cos\phi + sen\left(\frac{\eta}{2}\right)sen\left(\frac{\xi-\zeta}{2}\right)sen\phi\right]^2 \quad (60)$$

A través de este arreglo con cinco placas retardadoras se puede calcular la fase de Pancharatnam de la ecuación (54). Por otro lado, por simplicidad se puede considerar casos particulares de U_{tot} con el objetivo de reducir el número de placas retardadoras para simplificar el trabajo experimental.

El arreglo se hace fijando el valor de $\xi = -\pi$, de manera que U_{tot} se simplifica a:

$$U_{tot}^{\xi=-\pi} = Q\left(\frac{3\pi+2\eta}{4} - \frac{\phi}{2}\right)H\left(\frac{-4\pi+\zeta-\eta}{4} - \frac{\phi}{2}\right)Q\left(-\frac{\pi}{4} - \frac{\phi}{2}\right) \quad (61)$$

De la ecuación (54) se encuentra que:

$$cos^2(\delta) = cos^2\left(\frac{\eta}{2}\right) \quad (62)$$

Entonces la fase de Pancharatnam resulta ser:

$$\Phi_p = \frac{\eta}{2} \quad (63)$$

La intensidad para este caso particular es calculada por:

$$I = cos^2\left(\frac{\zeta}{2}\right)cos^2\left(\frac{\eta-2\phi}{2}\right) + sen^2\left(\frac{\zeta}{2}\right)cos^2\left(\frac{\eta}{2}\right) \quad (64)$$

Como se puede observar, para este arreglo experimental las placas retardadoras deben ser giradas un ángulo $\phi/2$. En el Laboratorio de Óptica Cuántica de la PUCP se cuenta con motores rotatorios piezoeléctricos los cuales pueden ser programados. En el siguiente capítulo se mostrarán los programas que se hicieron en LabVIEW para poder controlar a los motores rotatorios con el fin de automatizar la generación y medición de fases geométricas.

CAPÍTULO 3

Marco experimental: Instrumentos, software y programación del sistema

En éste capítulo se describen los instrumentos y software que se han empleado en el desarrollo de este trabajo, y además se muestra la evolución del programa que se ha realizado para las mediciones de fases geométricas que se llevan a cabo en el Laboratorio de Óptica Cuántica.

3.1. Componentes AGILIS: motor piezoeléctrico e interface de conexión multi-eje

Los instrumentos que se describen a continuación son de la marca NEWPORT, los cuales constan de motores piezoeléctricos rotatorios y de la interface de conexión multi-eje que los conecta al ordenador.

3.1.2. Motor piezoeléctrico

Este instrumento es un motor piezoeléctrico rotatorio de modelo AG-PR100, el cual tiene inscrito en su superficie una escala angular de 0° a 360°, y posee un movimiento mínimo gradual de 5 µrad.

Figura 4. Motor piezoeléctrico NEWPORT.

A continuación se muestran las especificaciones técnicas principales del motor rotatorio [11].

Especificaciones	Medidas
Diámetro óptico	1,0 in (25,4 mm)
Máximo espesor óptico	0.40 in (10 mm)
Rango de Movimiento	360°
Movimiento mínimo gradual	5 µrad
Velocidad máxima	2°/s
Resolución	2°

Tabla 1. Especificaciones técnicas del motor piezoeléctrico.

Estos motores rotatorios tienen como principal característica el moverse por una cantidad de pasos predeterminado. Estos pasos se programan, como se verá más adelante, en el software LabVIEW.

3.1.2. Interface de conexión multi-eje

La interface que se emplea para controlar los movimientos del motor rotatorio a través de un ordenador es de modelo AG-UC8. Esta interface posee 4 canales (CHANNEL). A su vez, cada canal posee 2 ejes (AXIS), como se muestra en la Figura 5, haciendo posible que se pueda controlar hasta 8 motores rotatorios.

Figura 5. Interface del motor piezoeléctrico.

Esta interface es capaz de conectarse a través de tres diferentes protocolos: USB, RS232 o RS485. En este trabajo se ha realizado la conexión a través del protocolo USB.

A continuación se presentan los parámetros de comunicación USB que requiere el ordenador (puerto COM) para que la conexión a través de USB sea exitosa [12]:

Configuración del puerto COM	
Velocidad de Transmisión	921600
Bits de datos	8
Paridad	Ninguna
Bits de parada	1
Control de flujo	Ninguna

Tabla 2. Configuración del puerto COM.

El fabricante de estos instrumentos (NEWPORT) brinda las librerías para poder controlar a los motores mediante el software LabVIEW, es por ello que la programación se ha realizado en este lenguaje de programación. Para más información acerca del motor piezoeléctrico rotatorio y su interface se puede revisar el manual del instrumento brindado por el fabricante NEWPORT.

3.2. Software LabVIEW

LabVIEW (Laboratory Virtual Instrument Engineering Workbech) es un lenguaje de programación gráfico que se utiliza fundamentalmente para el control de instrumentación; es decir, se utiliza para diseñar instrumentos virtuales (Virtual Instruments - VI) y se ejecuta bajo el sistema operativo Windows. A diferencia de los lenguajes de programación como Pascal, C++, Java, Visual Basic, etc., que se programan a través de comandos, en LabVIEW se programa mediante comandos y funciones gráficas [13]. National Instruments es la empresa desarrolladora y propietaria de LabVIEW.

En este trabajo se empleó la versión 2014 de LabVIEW.

Figura 6. LabVIEW 2014.

3.2.1. Entorno

Como se mencionó antes, LabVIEW es una herramienta de programación gráfica, la cual consta de dos ventanas principales: un instrumento real tiene un Panel Frontal donde están sus botones, pantallas, etc. y una circuitería interna. En LabVIEW estas partes reciben el nombre de Panel Frontal y Diagramas de Bloques, respectivamente.

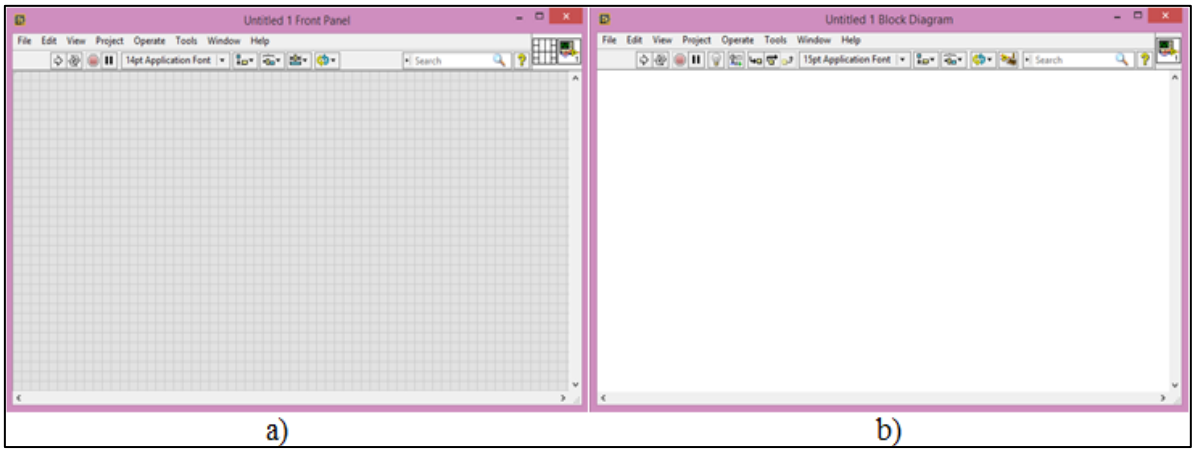

Figura 7. Panel Frontal (a) y Diagrama de Bloques (b).

3.2.2. Panel frontal

El Panel Frontal es la interfaz gráfica del VI (Virtual Instrument) con el usuario; esta interfaz recoge las entradas procedentes del usuario o datos adquiridos a través de las tarjetas de adquisición de datos (DAQ-data acquisition) del LabVIEW y muestra los resultados proporcionados por el programa [14].

El Panel Frontal sirve para diseñar un VI, el cual se construye en base a controles (datos de entrada) e indicadores (datos de salida). Todos los controles e indicadores se encuentran en la ventana "Controls Palette" (paleta de controles) que se muestran en la Figura 8.

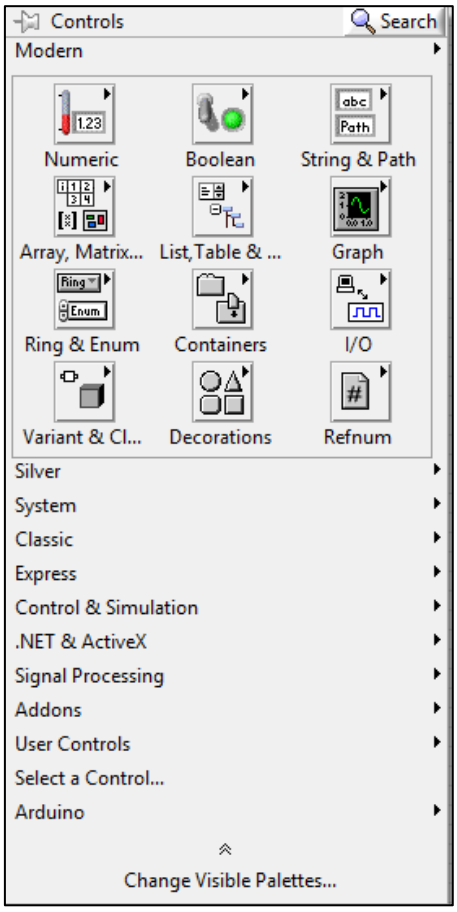

Figura 8. Paleta de Controles.

Cada uno de las opciones que se muestran en la Figura 8, tienen sub-opciones o sub-controles (objetos) que aparecen cuando el cursor del mouse se posiciona encima de cada uno de ellos. De todos los sub-controles que se muestra en la Figura 8, los más importantes son: **Modern, Classic y Express**; cada uno de los cuales se muestran a continuación.

En la sub-opción **Modern** de la Figura 8, se tienen los controles que se muestran en la Figura 9.

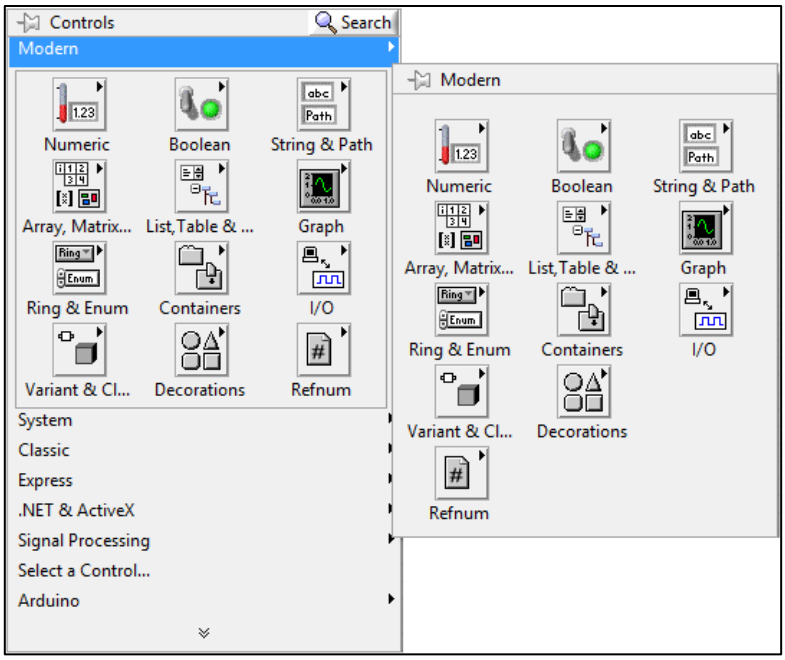

Figura 9. Controles de la sub-opción Modern.

Adicionalmente, cada uno de los controles que se muestran al lado derecho de la Figura 9, tienen sus controles e indicadores, los cuales se pueden observar ubicando el cursor del mouse sobre cada uno de las sub-opciones; por ejemplo los controles e indicadores de la sub-opción **Numeric**, se muestran en la Figura 10 y simplemente haciendo clic en uno de los controles e indicadores y arrastrándolo hacia el Panel Frontal se lo tendrá listo para usarlo.

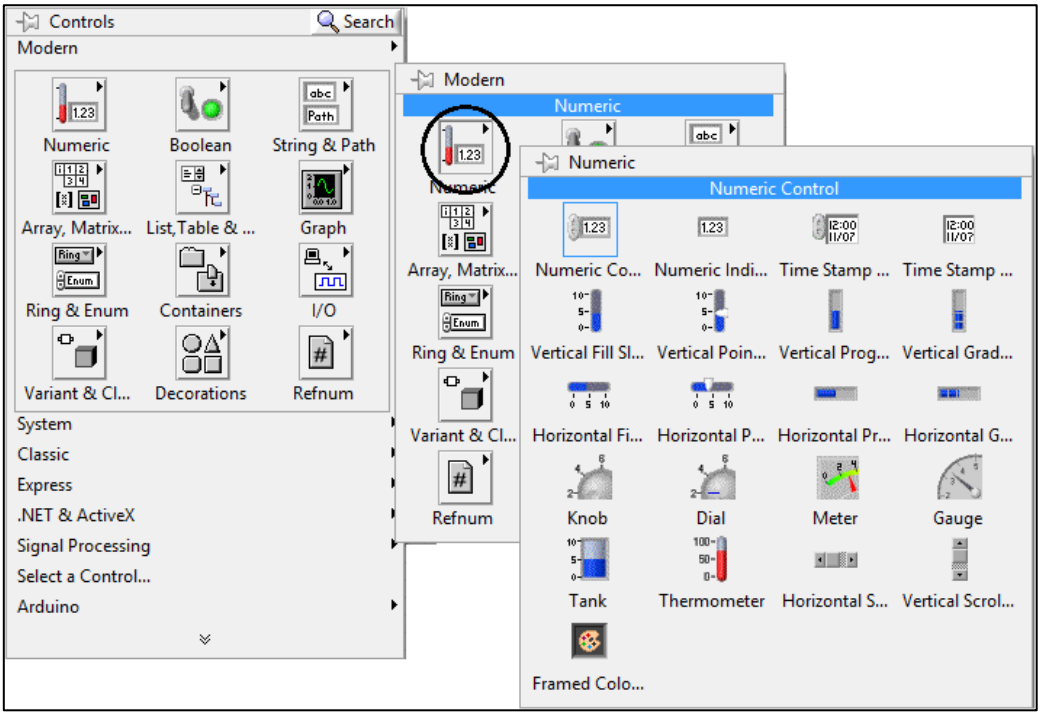

Figura 10. Controles de la sub-opción Numeric.

En la sub-opción **Classic** de la Figura 8, se tienen los controles que se muestran en la Figura 11, estos controles son los controles clásicos que se usaban en las versiones anteriores del LabVIEW, pero en la versión 2014 se ha incluido por razones de compatibilidad con las versiones anteriores

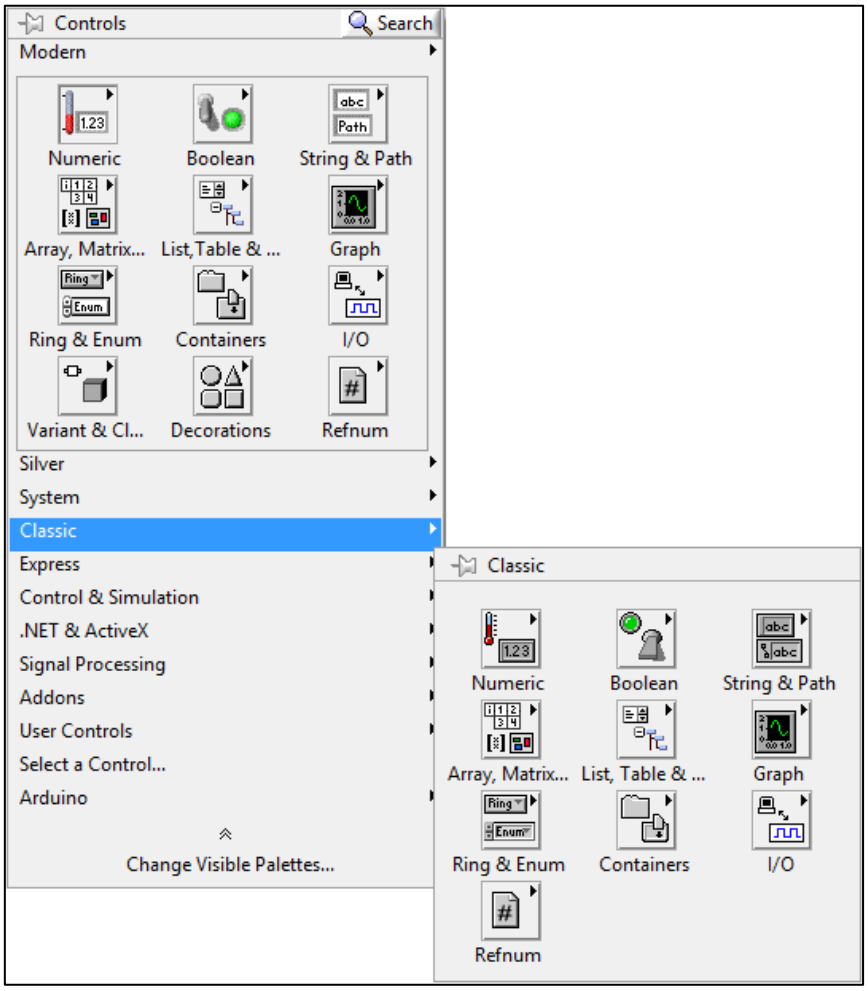

Figura 11. Controles de la sub-opción Classic.

Adicionalmente, cada uno de los controles que se muestran al lado derecho de la Figura 11 tienen sus controles e indicadores, los cuales se pueden observar ubicando el cursor del mouse sobre cada uno de las sub-opciones; por ejemplo los controles e indicadores de la sub-opción **Numeric**, se muestran en la Figura 12 y simplemente haciendo clic en uno de los controles o indicadores y arrastrándolo hacia el Panel Frontal se tendrá listo para usarlo. Observando las Figuras 10 y 11 se aprecia la diferencia que existe entre los controles **Numeric** de las sub-opciones **Modern** y **Classic**.

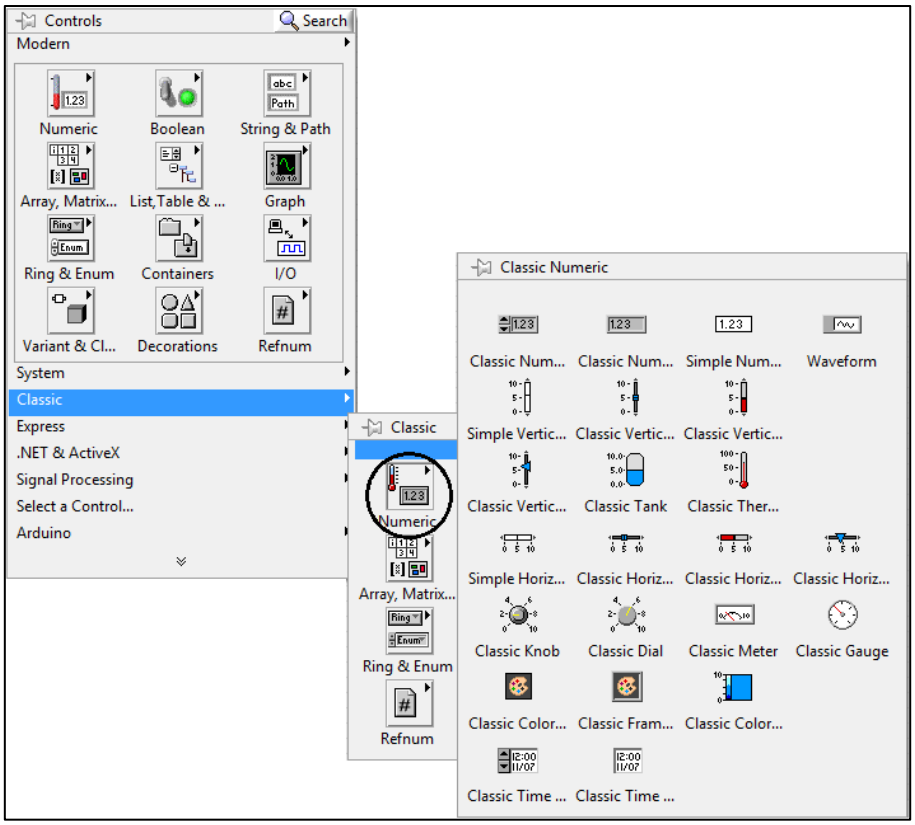

Figura 12. Controles de la sub-opción Numeric.

En la sub-opción **Express** de la Figura 8, se tienen los controles que se muestran en la Figura 13, estos controles son los que más se usan en la versión 2014, pero en realidad todos los controles de esta sub-opción también se encuentran en la sub-opción **Modern**.

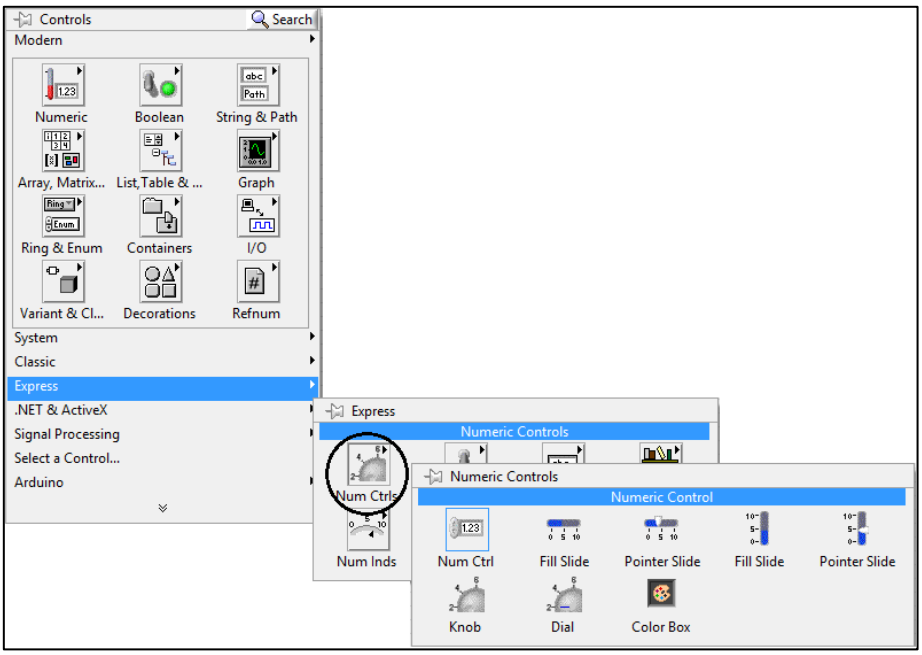

Figura 13. Controles de la sub-opción Express, sub-opción Num Ctrls.

3.2.3. Diagrama de bloques

El Diagrama de Bloques constituye el código fuente del VI, aquí es donde se realiza la implementación del programa del VI para controlar o realizar las conexiones, operaciones o cálculos que debe realizar el VI con los controles e indicadores que se crearon en el Panel Frontal; es decir, los controles e indicadores que se colocaron previamente en el Panel Frontal comienzan a tener vida en el Diagrama de Bloques, cuando se conectan los distintos objetos (controladores e indicadores) a través de cables por los cuales fluyen los datos. Los cables son las trayectorias que siguen los datos desde su origen hasta su destino; cada cable tiene un color o un estilo diferente, el cual sirve para diferenciar diversos tipos de datos que proporciona el LabVIEW [15].

En el Diagrama de Bloques se incluyen funciones, constantes y estructuras. Las estructuras son similares a los bucles de los lenguajes de programación convencionales, estos ejecutan el código que contienen de forma condicional o repetitiva (bucle for, while, case, etc.). Por otro lado, se debe resaltar que LabVIEW posee una extensa biblioteca de funciones, entre ellas: funciones aritméticas, funciones lógicas (comparaciones), funciones de entrada/salida, funciones de análisis de datos, etc.; para acceder a ellos se hace clic derecho del ratón cuando el cursor está sobre la pantalla del Diagrama de Bloques, en ese momento se visualiza la ventana **Functions Palette** (Paleta de Funciones) que se muestra en la Figura 14.

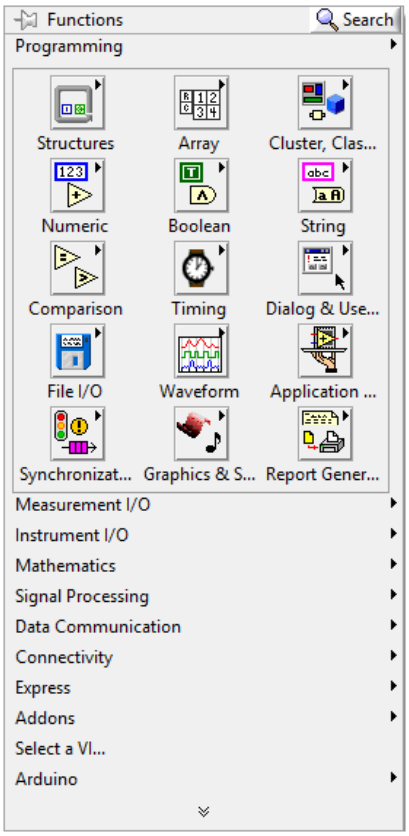

Figura 14. Paleta de Funciones.

Cada una de las opciones que se muestran en la Figura 14 tiene sub-funciones que aparecen cuando el cursor del mouse se posiciona sobre cada uno de ellos; por ejemplo, si posicionamos el cursor del mouse sobre la opción **Programming** se visualiza las funciones que se muestran en la Figura 15, cada una de ellas contienen funciones y estructuras de programación; entre las más importantes se tienen las funciones Structures, Numeric, Boolean; Comparison y Array.

En la Figura 15, se muestran las estructuras de programación While, Case Structure, Flat Sequence, etc. que se emplean en la programación para controlar los motores rotatorios.

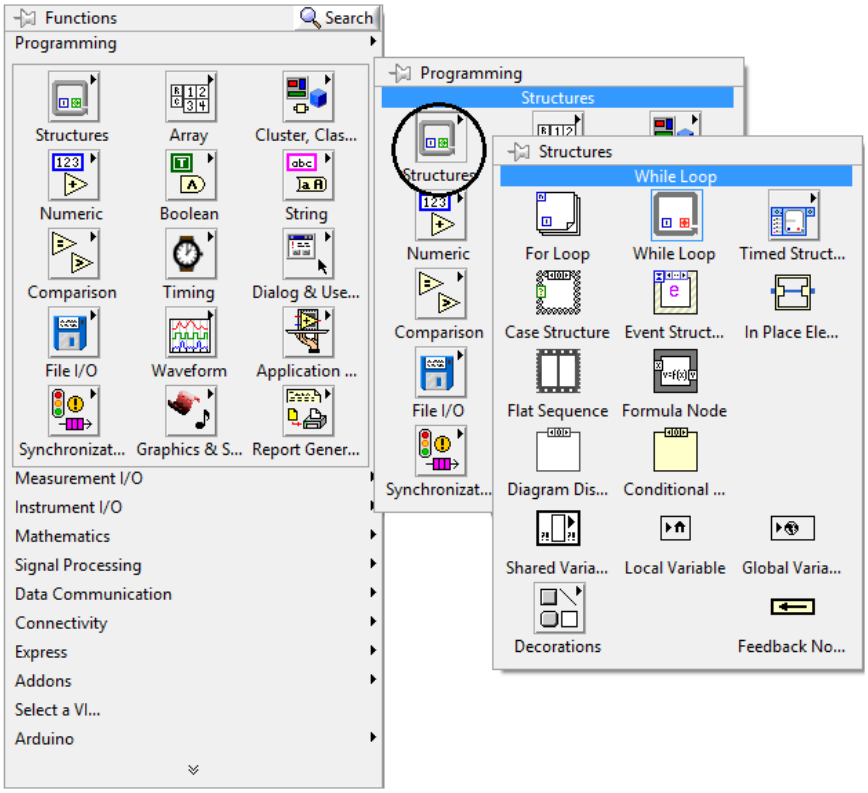

Figura 15. Paleta de estructuras de programación de la opción Structures.

Ya descritos los motores rotatorios con su interface y el lenguaje de programación (LabVIEW), a continuación se muestra el primer programa que se realizó para controlar a los motores rotatorios vía un ordenador.

3.3. Programación del motor rotatorio en LabVIEW

El programa que se realizó consistía en hacer girar al motor rotatorio tanto en sentido horario (positivo) como en sentido anti-horario (negativo) los cuales se muestran en las Figuras 16 y 17, respectivamente. El sentido del giro del motor rotatorio se puede cambiar a través de un interruptor.

El algoritmo consiste en los siguientes 6 pasos:

1) Se abre el puerto COM del ordenador para que se pueda controlar al motor rotatorio a través de la interface.
2) Se selecciona el Channel en el cual está conectado el motor rotatorio en la interface.
3) Se selecciona el Axis en el cual está conectado el motor rotatorio en la interface, y la cantidad de pasos que se le desea desplazar.
4) Se selecciona el sentido del giro del motor rotatorio a través del interruptor.
5) Las mediciones para este ejemplo se realizaran cada 100 milisegundos.
6) Se realizaron 50 mediciones para este ejemplo.
7) Se cierra el puerto del ordenador.

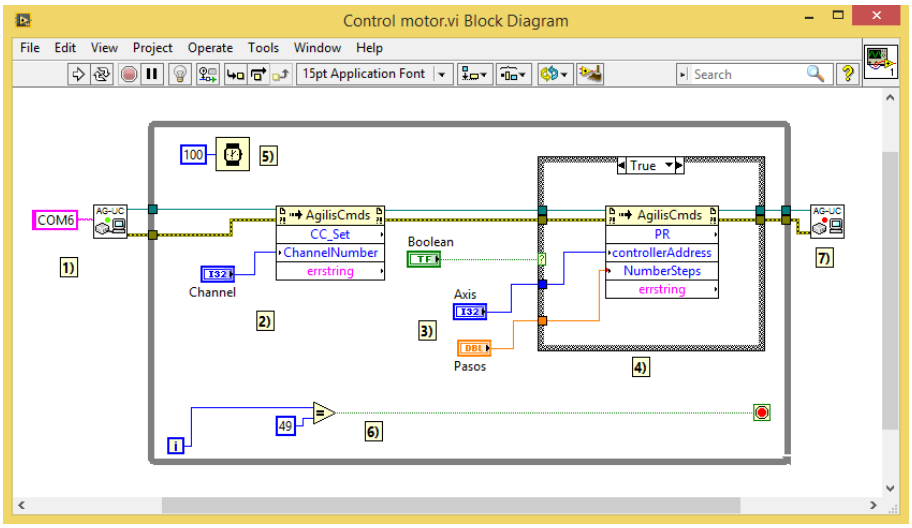

Figura 16. Algoritmo del movimiento del motor rotatorio en sentido horario (positivo).

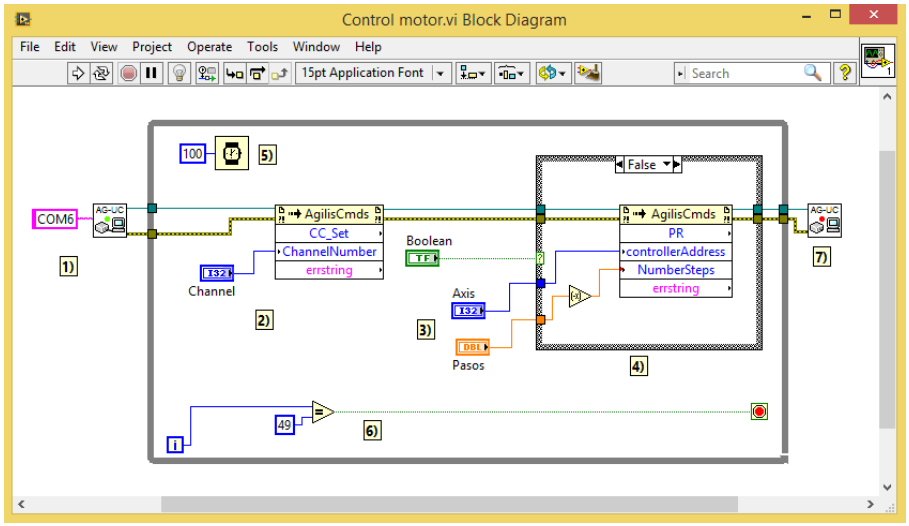

Figura 17. Algoritmo del movimiento del motor rotatorio en sentido anti-horario (negativo).

El programa en el Panel Frontal tiene la siguiente forma:

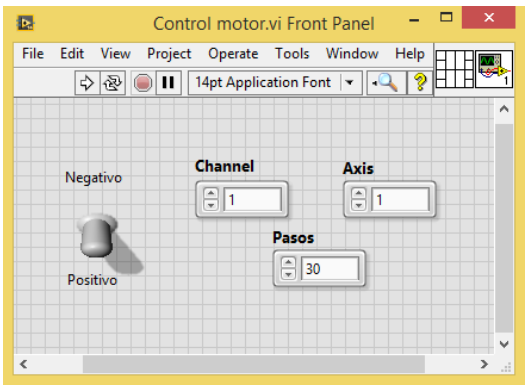

Figura 18. Diseño del programa para controlar al motor rotatorio en sentido horario o anti-horario.

En este programa primero se debe escoger el Channel y el Axis en donde se encuentra conectado el motor rotatorio en la interface, la cantidad de pasos que se desea desplazar, y a través del interruptor se escoge el sentido del giro que se desea realizar.

Con este programa se realizaron mediciones tanto en sentido horario como en sentido anti-horario, con la finalidad de obtener la razón de pasos que se le programaba al motor rotatorio por los grados sexagesimales que se desplazaba. Al revisar los resultados se obtuvieron las siguientes observaciones acerca del comportamiento del motor rotatorio:

- El número de pasos por grado sexagesimal en sentido horario (positivo) es diferente al del sentido anti-horario (negativo).

- El motor rotatorio tiene un comportamiento errático ya que el promedio del número de pasos por grado sexagesimal no es constante. En diferentes días en que se realizaron las mediciones se encontró que, tanto en sentido horario como anti-horario, los valores promedio cambiaban.

- El movimiento del motor rotatorio en sentido anti-horario es más estable que en sentido horario.

- El tiempo que emplea el motor rotatorio en realizar una misma cantidad de pasos en sentido horario y anti-horario es el mismo.

Con las mediciones realizadas en los diferentes días, no se encontró una razón de pasos por grado sexagesimal que sea constante debido a que el motor rotatorio tiene un comportamiento errático, es por ello que se optó por usar un sensor acelerómetro ADXL345 para determinar la cantidad de grados sexagesimales que gira el motor rotatorio y/o en qué posición angular se encuentra ubicado.

3.4. Acelerómetro ADXL345

El acelerómetro ADXL345 es un sistema completo de medición de aceleración como vector tridimensional. Este sensor mide tanto la aceleración dinámica (por ejemplo, movimiento o vibraciones) y la aceleración estática (por ejemplo la gravedad). Ello permite que el dispositivo sea usado como un sensor de inclinación [16].

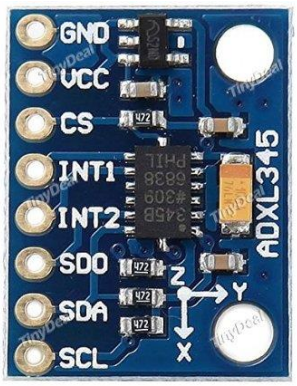

Figura 19. Acelerómetro ADXL345.

El acelerómetro contiene placas capacitivas internamente. Algunas de estos son fijas, mientras que otras están unidas a resortes minúsculos que se mueven internamente conforme las fuerzas de aceleración actúan sobre el sensor. Como estas placas se mueven una respecto a la otra, la capacitancia entre ellas cambia. A partir de estos cambios en la capacitancia, se puede determinar la aceleración.

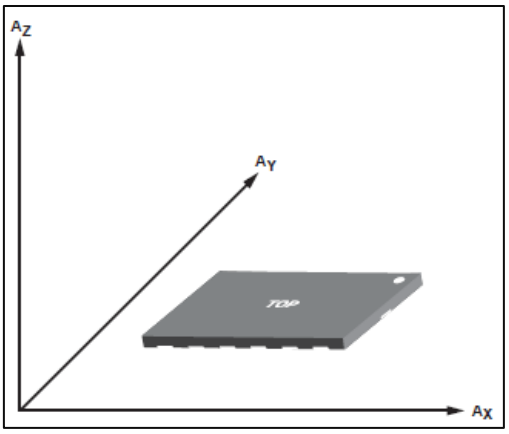

Figura 20. Orientación de los ejes coordenados en el sensor.

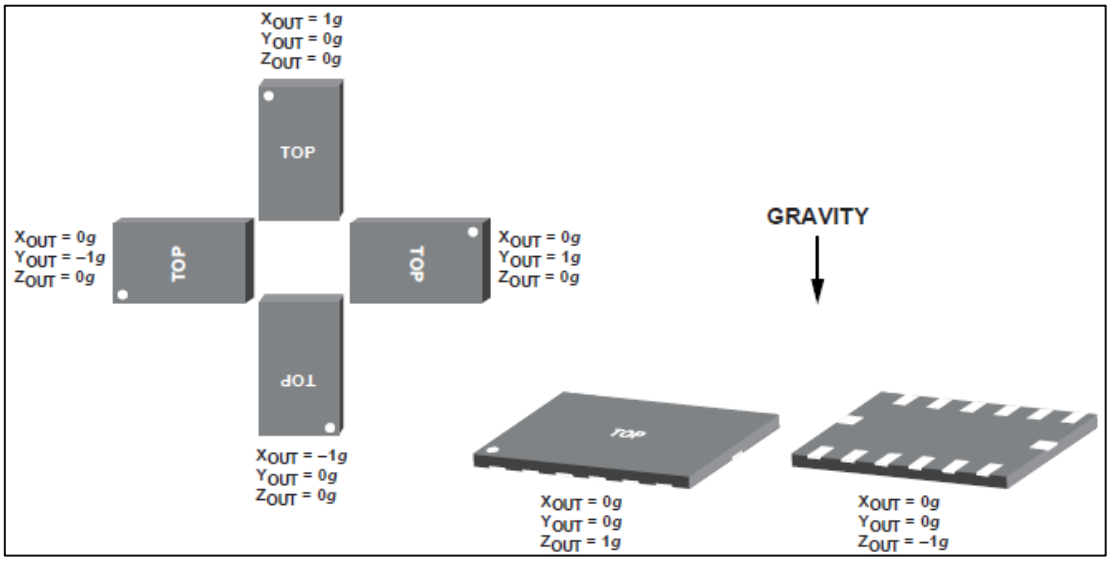

Figura 21. Respuesta de salida vs. Orientación de la gravedad.

En la Figura 20 se muestra la orientación de los ejes coordenados en el sensor; mientras que en la Figura 21 se muestra cómo va cambiando la respuesta de salida a medida que la orientación del sensor va cambiando.

En este acelerómetro la interfaz de comunicación digital puede ser I2C o SPI. En el presente trabajo la conexión que se utiliza es mediante la interfaz SPI.

Se debe de tener en cuenta que al estar usando una interfaz digital los valores arrojados por el acelerómetro ya no son voltajes variables; ahora son bytes. El acelerómetro, al arrojar estos bytes, necesita un hardware que los lea, el cual puede ser el Arduino.

Para más información acerca del acelerómetro ADXL345 se puede revisar el data sheet del sensor (referencia 11).

3.5. Arduino

Arduino es una plataforma electrónica de código abierto basada en hardware y software fáciles de usar. Las placas Arduino son capaces de leer señales de entrada (luz de un sensor, un dedo en un botón, o un mensaje en Twitter) para convertirlas en una salida (la activación de un motor, encender un LED, publicar algo en línea). Uno puede decirle qué hacer a su tarjeta Arduino mediante el envío de un conjunto de instrucciones para el micro-controlador en la

tarjeta. Para ello se usa el lenguaje de programación Arduino (basado en Wiring), y el software de Arduino (IDE), basado en Processing [17].

Arduino se creó con un fuerte enfoque en el movimiento del código abierto, esto es, la compartición del conocimiento. Desde entonces Arduino ha ido creciendo sin parar gracias al aporte de investigadores, empresas y aficionados de todo el mundo. El "ecosistema" Arduino está formado por tres elementos básicos:

3.5.1. El hardware

Placa electrónica básica para hacer desarrollos rápidos y económicos. Este hardware es libre y permite una forma simple de aprender electrónica básica desmenuzando los componentes que forman sus circuitos. También permite ampliar el hardware mediante el uso de otras placas (shields) que se conectan a la del microprocesador.

En el mercado existen diversas placas Arduino. Para el desarrollo de este trabajo se ha utilizado la placa Arduino UNO.

Arduino UNO es una placa electrónica que posee el micro-controlador ATmega328P. Cuenta con 14 pines de entrada/salida (de los cuales 6 pueden ser usados como salidas PMW), 6 entradas analógicas y una velocidad del reloj de 16 MHz, una conexión USB, un conector de alimentación, una cabecera ICSP y un botón de reinicio. Contiene todo lo necesario para apoyar al micro-controlador, conectarlo a un ordenador con un cable USB

Especificaciones técnicas [18].

Micro-controlador	ATmega328P
Voltaje de operación	5 V
Voltaje de entrada	7 – 12 V
Voltaje de entrada (límite)	6 – 20 V
Pines I/O	14
Pines PWM	6
Pines de entrada analógica	6
Corriente DC por Pin I/O	20 mA
Corriente DC por Pin 3.3 V	50 mA 32 KB (ATmega328P)
SRAM	2 KB (ATmega328P)
EEPROM	1 KB (ATmega328P)
Velocidad del reloj	16 MHz
Largo	68.6 mm
Ancho	53.4 mm
Peso	25 g

Tabla 3. Especificaciones técnicas de la placa Arduino UNO

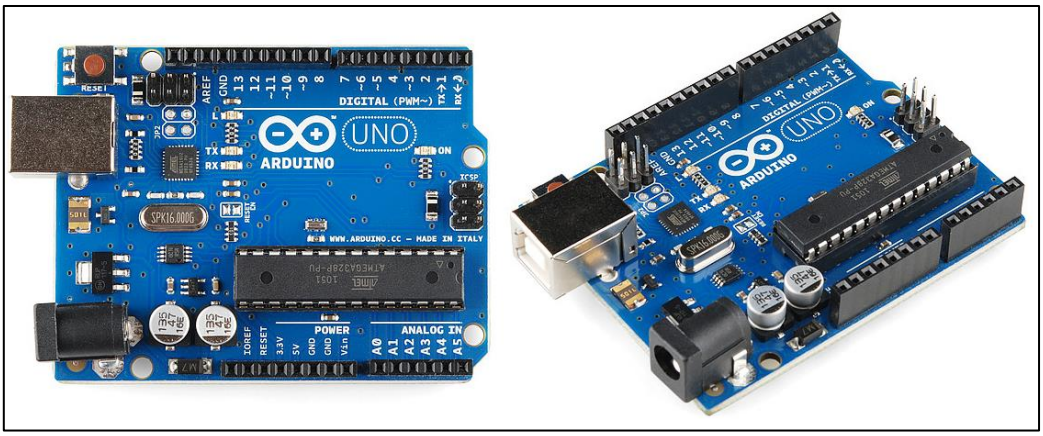

Figura 22. Placa Arduino UNO.

3.5.2. El entorno de programación

Permite de forma muy sencilla (teniendo conocimientos básicos de lenguaje de programación) la programación del hardware escogido. Además, este entorno es multiplataforma (Windows, Mac y Linux) y únicamente necesita un cable USB para conectarse a la placa [19].

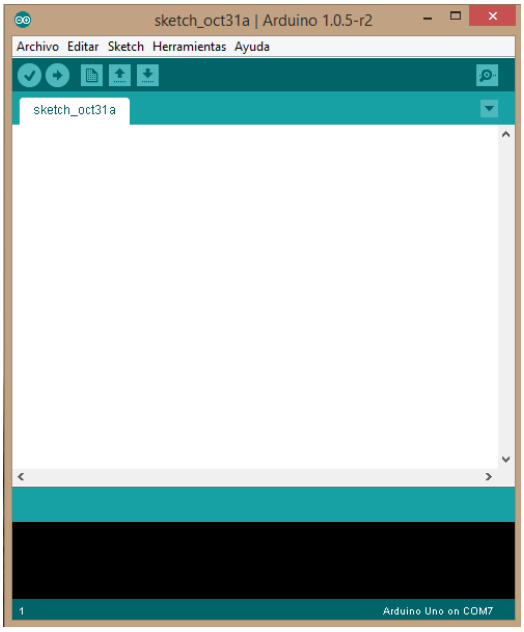

Figura 23. Vista del entorno de programación de Arduino.

Arduino dispone de un sencillo entorno de programación que permite escribir programas (Sketchs), compilarlos y transferirlos al hardware. Al igual que el resto del entorno Arduino, se puede acceder al código fuente, el cual está escrito en C, C++ y Java.

El IDE (Integrated Development Environment) se compone del propio editor de texto, un conjunto de librerías y las herramientas de compilación y programación (toolchain) necesarias para trabajar.

3.5.3. Comunidad

Personas que participan activamente en nuevos desarrollos, ideas y test. Se puede consultar a través del foro y de la wiki tanto el hardware como el software de forma libre, tanto en la web oficial de Arduino como de otros realizados por la comunidad.

3.5.4. Conexión del acelerómetro ADXL345 al arduino

Se debe conectar el acelerómetro a la placa del Arduino UNO, tal como se muestra en la Figura 24 [20].

Arduino Pin	ADXL345 Pin
10	CS
11	SDA
12	SDO
13	SCL
3V3	VCC
Gnd	GND

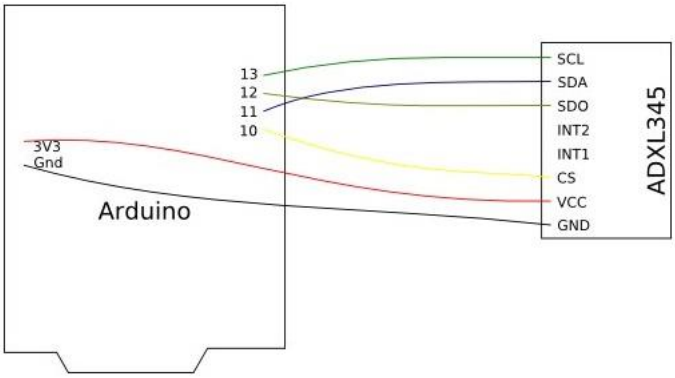

Figura 24. Conexión del acelerómetro ADXL345 a la placa del Arduino UNO.

Una vez conectado el acelerómetro a la placa del Arduino UNO, se procedió a realizar toda la programación en LabVIEW empleando la librería LINX proporcionado por National Instruments.

3.5.5. Librería que conecta a arduino con LabVIEW

La librería que se ha empleado para conectar Arduino y LabVIEW se llama “LINX de LabVIEW MakerHub”. El objetivo es poder conectarse y controlar Arduino sin tener que realizar un programa para él.

Para instalar LINX se debe de visitar la siguiente página web http://sine.ni.com/nips/cds/view/p/lang/es/nid/212478 y descargarlo utilizando el gestor de paquetes JKI VIPM.

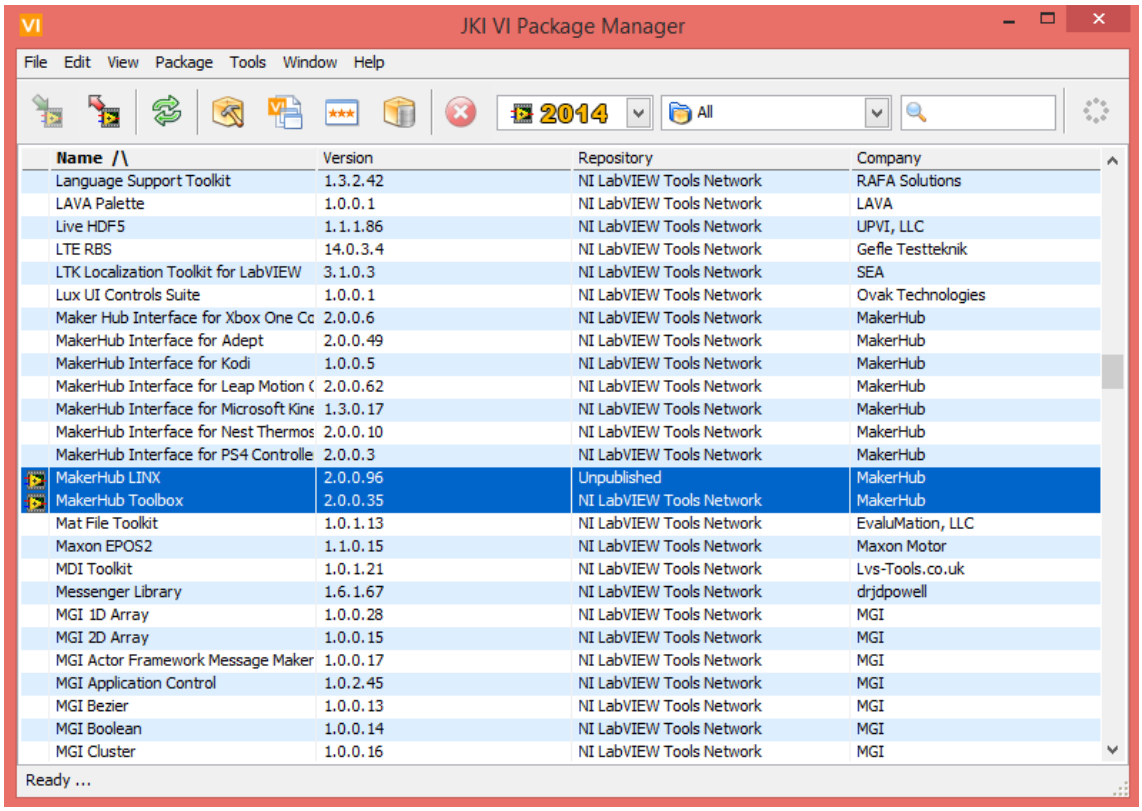

Figura 25. Instalación de LINX mediante el JKI VI Package Manager

En LabVIEW aparecerá un nuevo menú con las nuevas funciones. Se dispone de las típicas funciones de leer/escribir pines digitales y analógicos, pero también funciones de alto nivel para trabajar con sensores, motores, pantallas, etc. El método de trabajo sería utilizar la función de inicializar, después las de escribir/leer y finalmente la de cierre.

El siguiente paso fue acoplar el acelerómetro al motor rotatorio, como se muestra en la Figura 26. De ahora en adelante, por cada giro que se le programe al motor rotatorio será también girado el acelerómetro. De esta manera y a través del programa que se ha realizado en LabVIEW se podrá cuantificar la cantidad de grados que gira el motor/acelerómetro y/o en qué posición angular se encuentra ubicado.

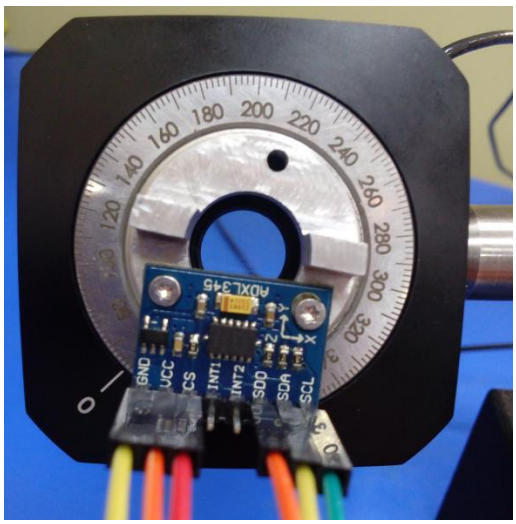

Figura 26. Acelerómetro acoplado al motor rotatorio.

3.6. Programación del acelerómetro acoplado al motor rotatorio

A continuación se detallan los diversos programas que se han realizado por separado en LabVIEW, los cuales serán utilizados en el programa final.

3.6.1. Programa para que LabVIEW lea los datos del acelerómetro ADXL345

El algoritmo que se muestra en la Figura 27, hace que LabVIEW lea los datos que le está enviando el acelerómetro en valores de bytes en cada eje coordenado.

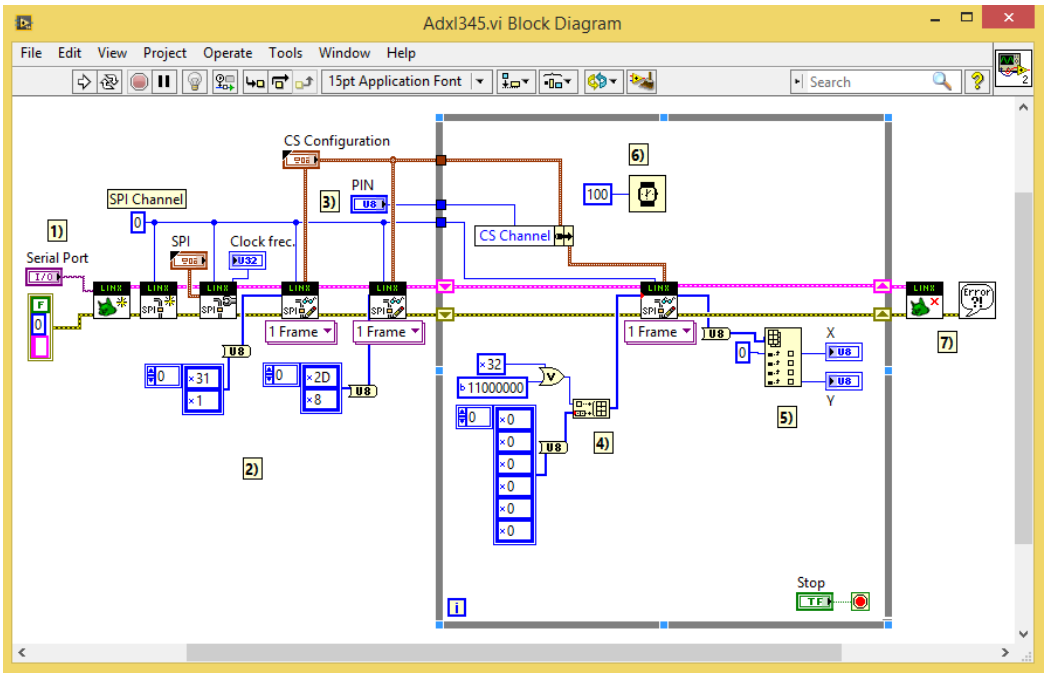

Figura 27. Algoritmo para leer los datos que recibe LabVIEW del acelerómetro.

El algoritmo consiste en los 7 pasos:

1) Se abre el puerto COM por el que está conectado el Arduino UNO al ordenador.
2) Se hace la configuración brindada en el data sheet del acelerómetro.
3) Se selecciona el PIN de entrada del Arduino UNO en el cual está conectado el PIN CS del acelerómetro.
4) El acelerómetro empieza a enviarle datos al LabVIEW.
5) En LabVIEW se comienzan a registrar los datos enviados por el acelerómetro.
6) Estos datos son registrados cada 100 milisegundos.
7) Se cierra el puerto en donde está conectado el Arduino UNO.

En la Figura 28, se muestra el diseño para este programa en el panel frontal. Los indicadores “X” e “Y” muestran los bytes en los ejes X e Y.

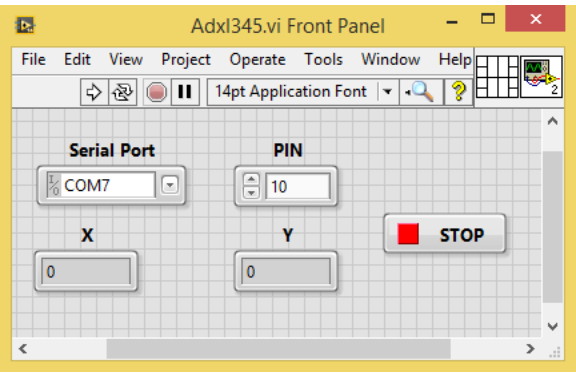

Figura 28. Diseño del programa para leer los datos que recibe LabVIEW del acelerómetro.

En esta ventana se debe seleccionar el puerto COM al que se le conectó el Arduino UNO al ordenador; y también el PIN de entrada en el Arduino UNO al cual se le conectó el PIN CS del acelerómetro.

3.6.2. <u>Programa para que LabVIEW convierta bytes en valores de aceleración de la gravedad en cada eje coordenado y en grados angulares</u>

El algoritmo que se muestra en la Figura 29 hace que LabVIEW convierte los bytes que le envía el acelerómetro en valores de la aceleración de la gravedad en cada eje coordenado, y también la posición angular en la que se encuentra ubicado.

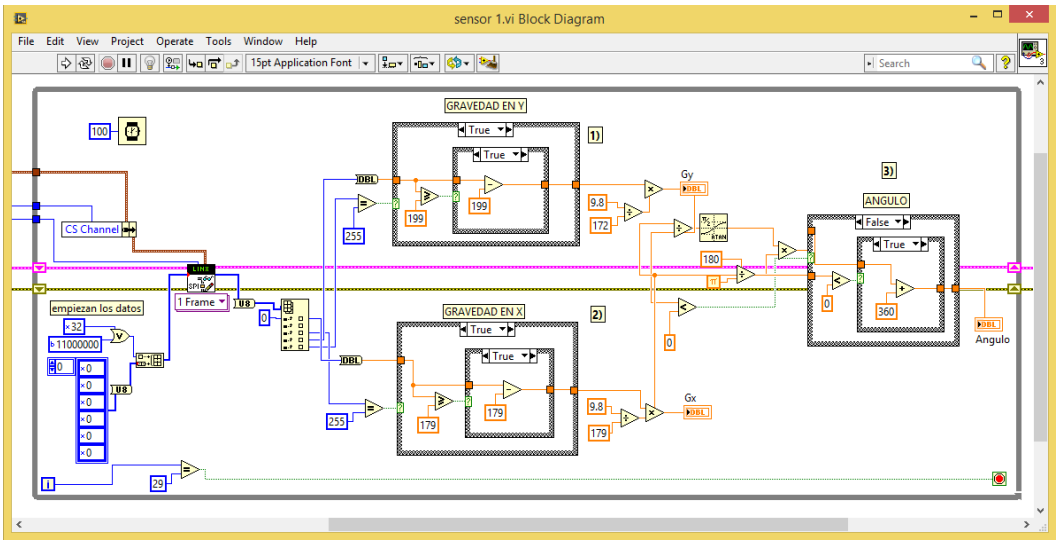

Figura 29. Algoritmo para convertir los bytes en valores de aceleración de la gravedad en cada eje coordenado; y se calcula en qué posición angular se encuentra ubicado.

El programa consiste en los siguientes tres pasos:

1) Se calcula el valor de la aceleración de la gravedad en el eje Y.
2) Se calcula el valor de la aceleración de la gravedad en el eje X.
3) Se calcula la posición angular en la que se encuentra el acelerómetro.

En la Figura 30, se muestra el diseño para este programa en el panel Frontal. Los indicadores “Gx” y “Gy” muestran los valores de la aceleración de la gravedad en los ejes coordenados X e Y, respectivamente; el indicador “Angulo” muestra el ángulo en el cual se encuentra ubicado el acelerómetro. Se realizan 30 mediciones y son guardadas en un archivo de texto.

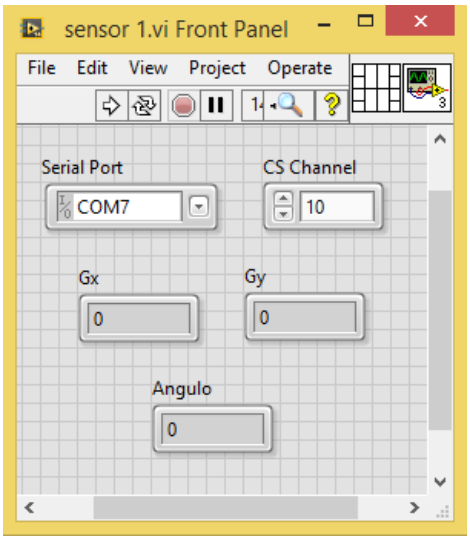

Figura 30. Diseño del programa que nos da el valor de la aceleración de la gravedad en cada eje coordenado y el ángulo en el cual se encuentra ubicado el acelerómetro.

3.6.3. <u>Programa para leer datos desde un archivo de texto y procesarlos</u>

El siguiente paso, fue realizar el algoritmo para leer los datos que están siendo guardados en el archivo de texto para procesarlos, es decir calcular promedios y desviaciones estándar.

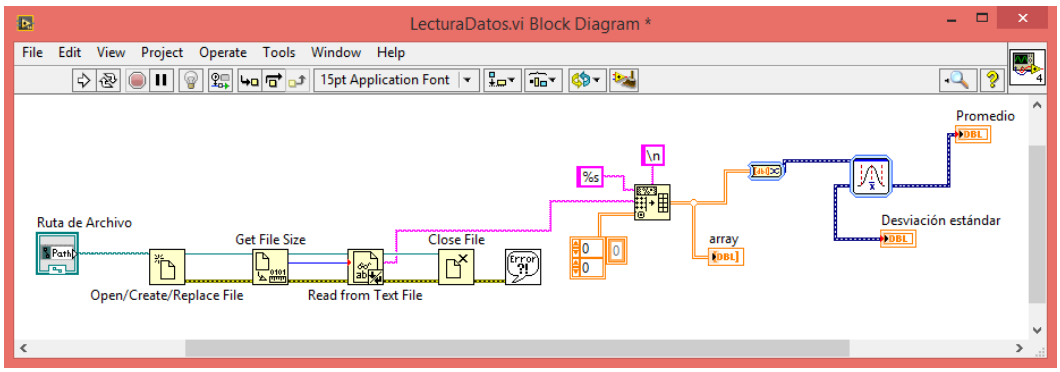

Figura 31. Algoritmo para leer datos de un archivo de texto y calcular promedios y desviaciones estándar.

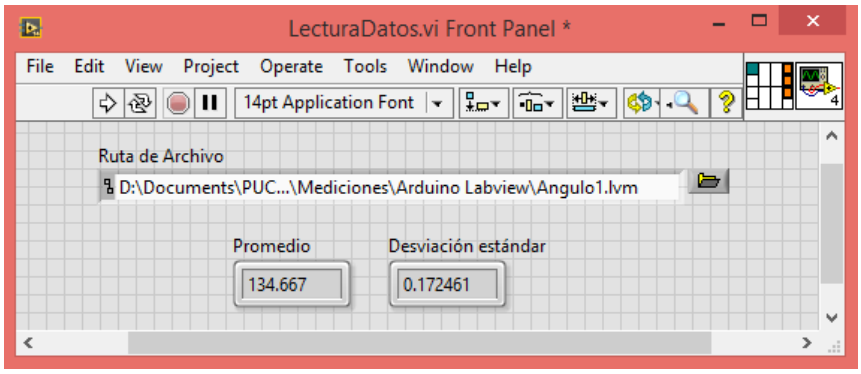

Figura 32. Diseño del programa para leer datos de un archivo de texto y calcular promedios y desviaciones estándar.

Hasta el momento tenemos tres programas realizados:

a) Programa para girar al motor rotatorio en sentido horario o anti-horario.
b) Programa para calcular la posición angular en la cual se encuentra ubicado el acelerómetro.

c) Programa para leer datos que están siendo guardados en un archivo de texto, y calcular promedios y desviaciones estándar.

El paso siguiente consiste en unir estos tres programas en uno solo.

3.6.4. Programa para girar al motor rotatorio y guardar las mediciones realizadas por el acelerómetro

El algoritmo que se muestra en la Figura 33 hace que el acelerómetro acoplado al motor rotatorio se posicione en el ángulo al cual se le desea ubicar. En este programa se han combinado los tres anteriores programas que ya se han realizado.

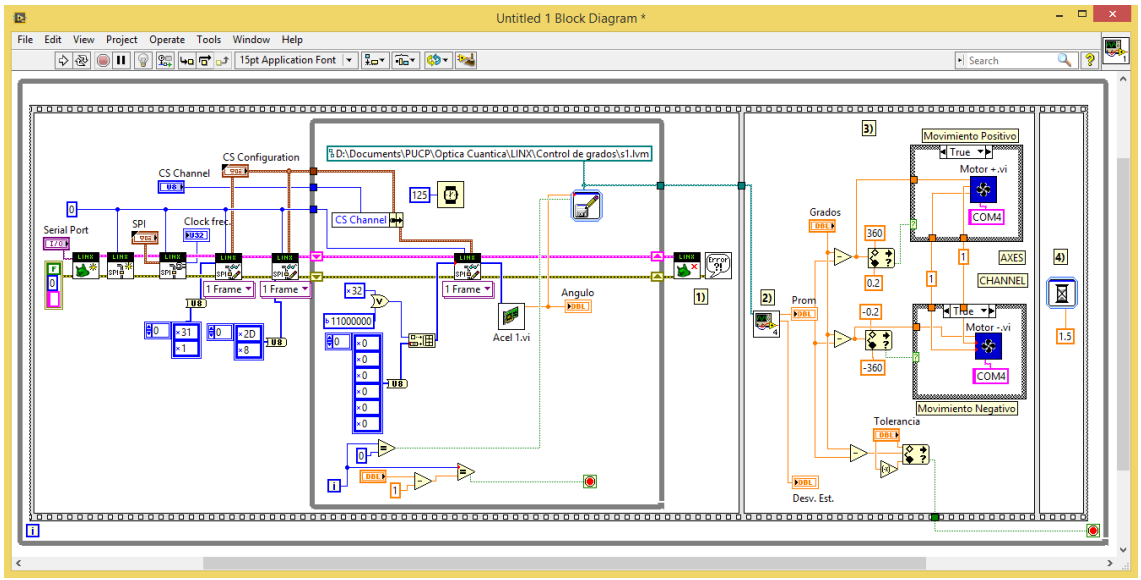

Figura 33. Algoritmo realizado para ubicar al acelerómetro acoplado al motor rotatorio a posicionarse en un ángulo predeterminado.

El programa que se realizó consiste en 3 pasos, uno detrás del otro:

1. En el control "Grados", mostrado en la Figura 34, se introduce el valor del ángulo al cual se desea posicionar al acelerómetro acoplado al motor rotatorio; en el control "Tolerancia" se introduce el valor de tolerancia que se le desea dar al ángulo introducido. Ambos indicadores se encuentran en el panel Frontal. LabVIEW mide 30 veces la posición inicial angular en la que se encuentra el acelerómetro acoplado al motor rotatorio y los datos son guardados en un archivo de texto
2. Se leen los datos guardados en el archivo de texto y se calcula el promedio y la desviación estándar de estas 30 mediciones.
3. Si el promedio calculado resulta ser mayor o menor al valor introducido en el control "Grados", incluida la tolerancia, el motor rotatorio comenzará a realizar el movimiento en sentido negativo o positivo según sea el caso, y el ciclo se repetirá hasta que el promedio se encuentre dentro de la tolerancia dada que es cuando el programa se detiene.

El diseño del programa tiene la siguiente forma:

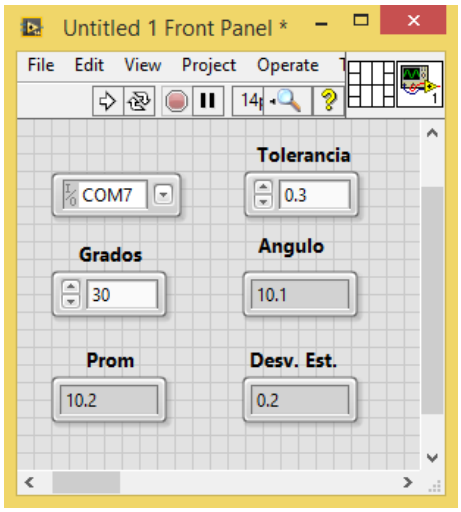

Figura 34. Diseño del programa para posicionar al motor rotatorio acoplado al acelerómetro en el ángulo que se desea.

El indicador "Angulo" nos muestra las 30 mediciones que realiza el programa (cada medición se realiza cada 125 milisegundos). El indicador "Prom" y "Desv. Est." indican el promedio y la desviación estándar de las 30 mediciones realizadas.

Hasta aquí tenemos un programa para controlar un solo motor rotatorio en el que ha sido acoplado un acelerómetro. Con este programa podemos indicarle al acelerómetro acoplado al motor rotatorio que se ubique en el ángulo que se desee, y se obtienen luego de realizar 30 mediciones su promedio y desviación estándar.

3.6.5. Programa general para controlar los acelerómetros acoplados a los motores rotatorios y realizar mediciones de intensidad del láser

El siguiente paso es generalizar el último programa para los 8 motores rotatorios con sus respectivos acelerómetros acoplados y agregarle un algoritmo para medir la intensidad del láser. En la Figura 36 se muestra el diseño del programa final que se ha realizado.

El programa consiste en los siguientes pasos:

1) En una hoja de texto se deben introducir los valores de los ángulos para los ocho motores rotatorios como se muestra en la Figura 35. Se debe de colocar la ubicación del archivo de texto en el control "DATOS DE ENTRADA", mostrado en la Figura 36.

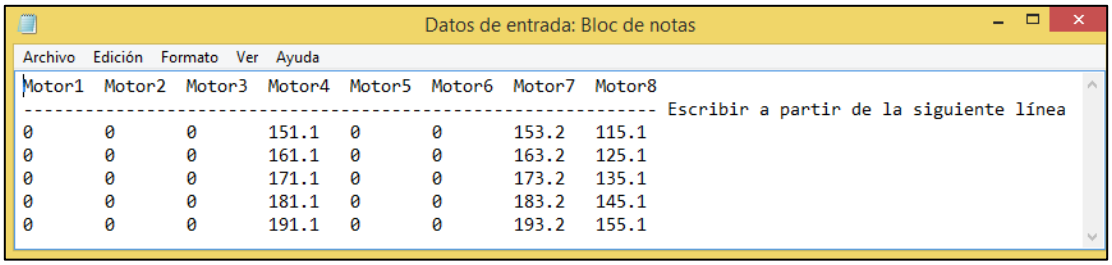

Figura 35. Datos de entrada.

2) Se selecciona el puerto en el cual está conectado el Arduino UNO al ordenador.
3) Se introduce la tolerancia deseada en el control "TOLERANCIA".
4) En el control "CICLOS" se introduce la cantidad de las distintas posiciones angulares en las cuales se desea ubicar al mismo motor rotatorio.

5) Un LED de color verde indica que su respectivo motor se encuentra “encendido” y que se desplazará durante la medición; sin embargo, un LED de color negro indica que su respectivo motor se encuentra “apagado” y que no se desplazará durante la medición. Por default, al abrir el programa todos los motores estarán apagados, y haciendo un clic sobre el LED se podrá encender.
6) Los motores encendidos comenzarán a desplazarse al ángulo introducido deseado, uno después del otro. Una vez que los motores encendidos se hayan ubicado en la posición angular deseada, se registrarán 30 datos leídos por el potenciómetro de la intensidad del láser.
7) Por último, en un archivo de texto se guardarán los promedios y desviaciones estándar de las mediciones angulares de los motores encendidos y de la lectura del potenciómetro.

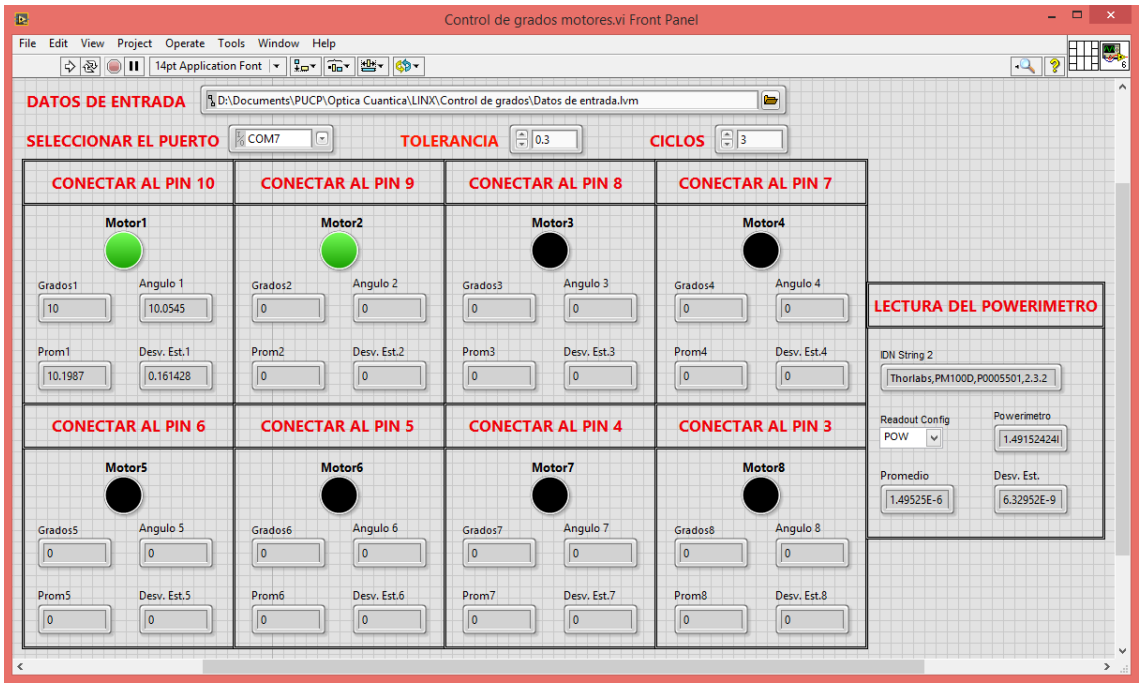

Figura 36. Programa para controlar los ocho motores rotatorios y realizar mediciones de la intensidad del láser.

En la siguiente sección se muestra cómo han sido utilizados nuestros programas en las diversas mediciones que se realizaron.

CAPÍTULO 4

RESULTADOS DE LA CALIBRACIÓN Y MEDICIONES

En esta sección se describen las mediciones realizadas con los programas desarrollados en la sección anterior y los resultados obtenidos.

4.1 Calibración de placas retardadoras

Antes de realizar cualquier medición, lo primero que se debe conocer es la posición angular del eje rápido de las placas retardadoras que se emplearán en el arreglo experimental.

La calibración de las placas retardadoras consiste en lo siguiente: en la Figura 37 se muestra un haz de luz láser, el cual se encontrará en el estado de polarización horizontal luego de atravesar el polarizador horizontal. Al incidir sobre la placa retardadora de, por ejemplo, media longitud de onda, cuyo eje rápido forma un ángulo de 45° respecto del eje x, su estado de polarización cambia al estado vertical, el cual es identificado con la ayuda de otro polarizador (analizador) y un detector. Cuando el eje de transmisión del analizador es vertical, paralelo al estado de polarización del haz incidente, se produce un máximo en intensidad que es registrado por el detector. Si el haz incidente no hubiera sufrido cambio alguno en su estado de polarización horizontal e incidiera sobre el analizador, sería absorbido completamente y el detector (idealmente) no registraría intensidad alguna [21]. Entonces, en esta posición se encontraría el eje rápido de la placa retardadora.

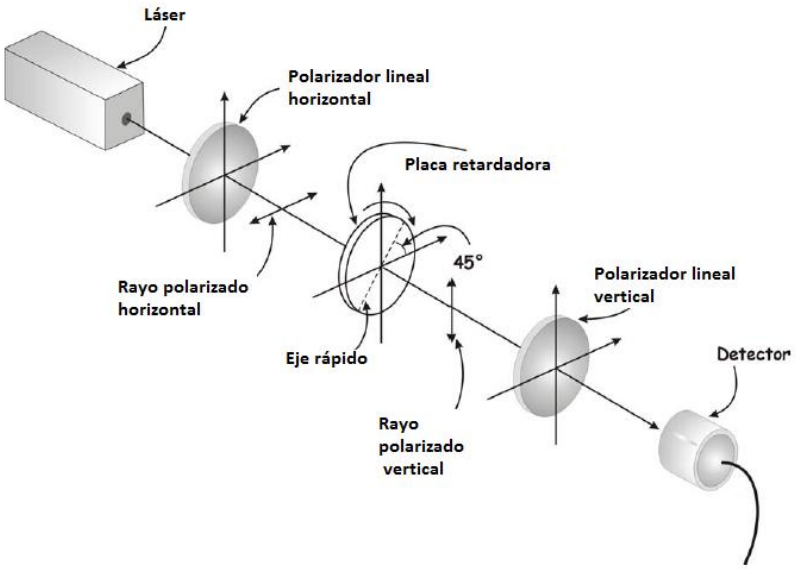

Figura 37. Calibración de placas retardadoras

Para la calibración de placas retardadoras se realizó el programa mostrado en la Figura 38. Este programa hace que el acelerómetro acoplado al motor rotatorio en el cual se encuentra a la vez una placa retardadora, se desplace aproximadamente 0.5°. Luego de moverse, el motor se detiene y se comienza a medir en qué posición angular se encuentra y se registra la intensidad del láser, tomando 30 datos para ambos. Se calculan los promedios angulares y de intensidades, y son guardados en una hoja de texto. Si el promedio de la medición angular tomada es mayor al valor del ángulo que se introduce en el control “Grado final”, mostrado en la Figura 38, el

programa se detendrá, de caso contrario se volverá a realizar el mismo ciclo hasta que el promedio de las 30 mediciones angulares sea mayor al valor angular introducido.

Figura 38. Programa para calibrar placas retardadoras

Cuando las mediciones han finalizado, el programa automáticamente determina cuál ha sido el menor valor de intensidad del haz del láser medido y en qué posición angular se encontró. Es decir, nos indicará en qué posición angular se encuentra el eje rápido de la placa retardadora.

Al calibrar una placa retardadora ubicada en un motor rotatorio, se encontró la siguiente gráfica:

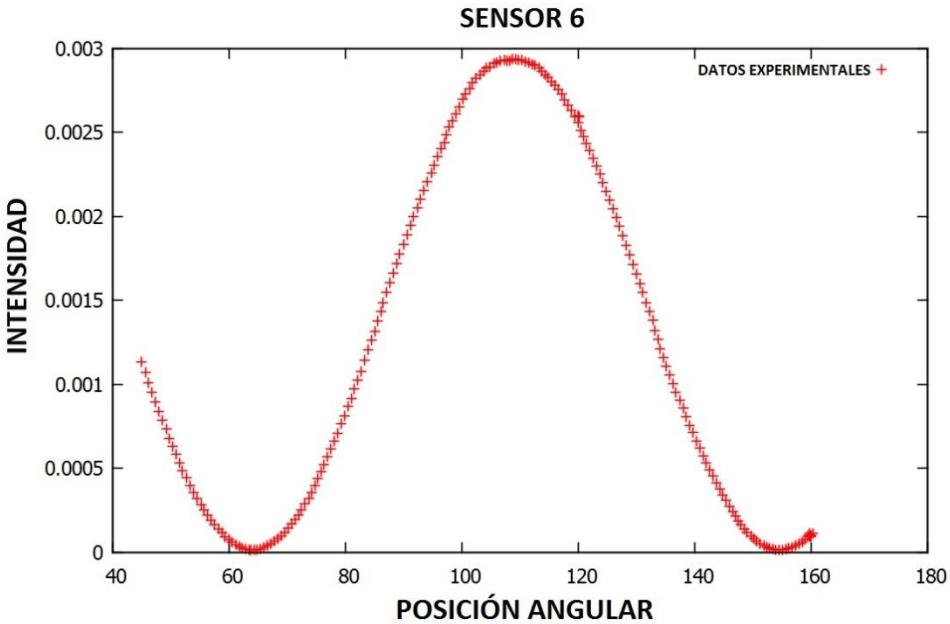

Figura 39. Calibración de una placa retardadora - Posición angular vs. Intensidad.

En la Figura 39 se ha graficado las mediciones realizadas de intensidad del láser versus la posición angular. Para esta calibración se encontró que el eje rápido está ubicado en la posición angular de 63.8°, ya que para este ángulo se encontró la menor intensidad medida.

4.2 Mediciones

Las mediciones que se realizaron son de acuerdo al arreglo experimental desarrollado en la sección 2 de este trabajo, descrito por la ecuación (61):

$$U_{tot}^{\xi=-\pi} = Q\left(\frac{3\pi+2\eta}{4}-\frac{\phi}{2}\right)H\left(\frac{-4\pi+\zeta-\eta}{4}-\frac{\phi}{2}\right)Q\left(-\frac{\pi}{4}-\frac{\phi}{2}\right) \qquad (65)$$

el cual se puede representar por la disposición de tres placas retardadoras: una placa retardadora de media onda (H) colocada entre dos placas retardadoras de un cuarto de onda (Q) las cuáles serán giradas una misma cantidad de ángulos.

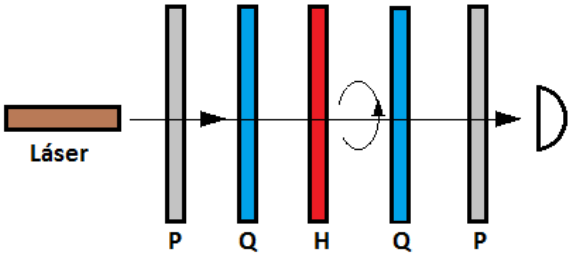

Figura 40. Arreglo polarimétrico para calcular la fase de Pancharatnam $\Phi_{\mathbf{p}}$.

En las siguientes figuras se muestra cómo fue la disposición de las placas retardadoras en el Laboratorio de Óptica Cuántica.

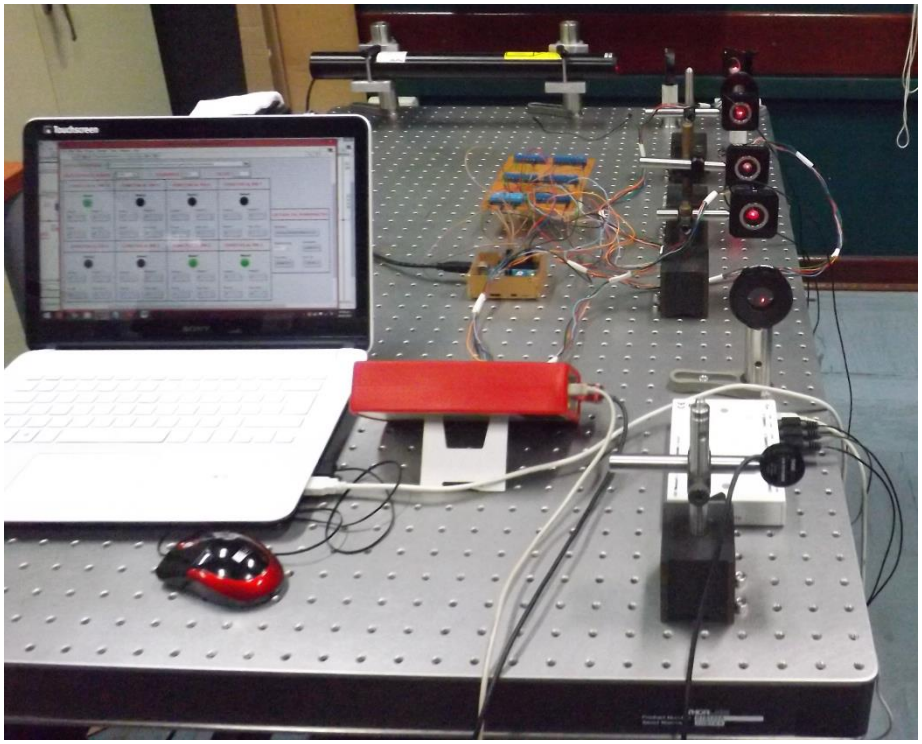

Figura 41. Arreglo polarimétrico armado en el Laboratorio (vista posterior).

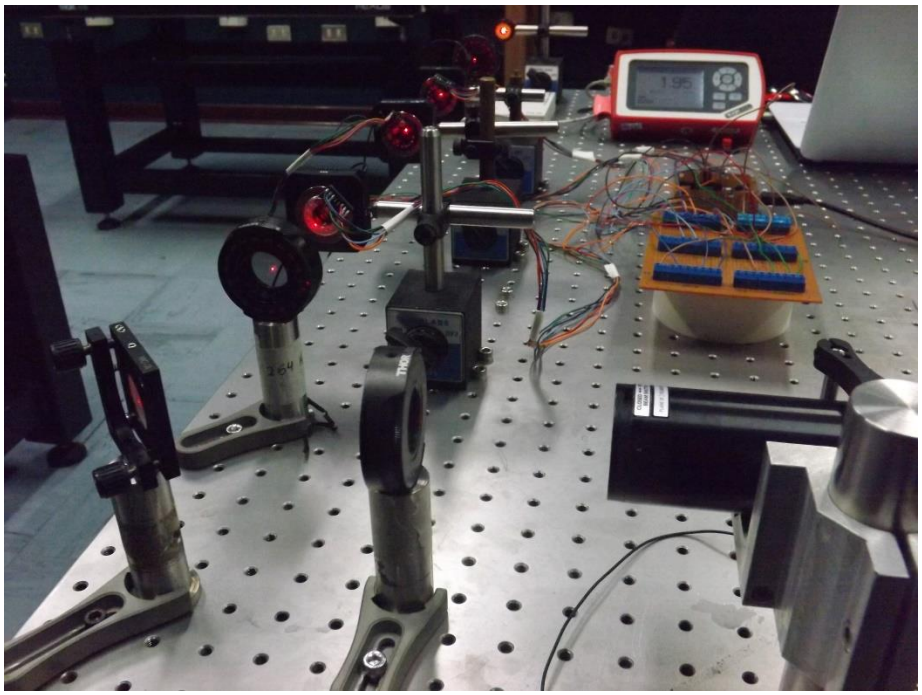

Figura 42. Arreglo polarimétrico armado en el Laboratorio (vista frontal).

Se realizaron las pruebas fijando los siguientes valores: $\xi = -\pi, \zeta = 0$, $\eta = \pi$ variando ϕ cada $\pi/6$. Estos valores fueron reemplazados en la ecuación (64), y se obtuvo la siguiente gráfica de resultados del interferograma:

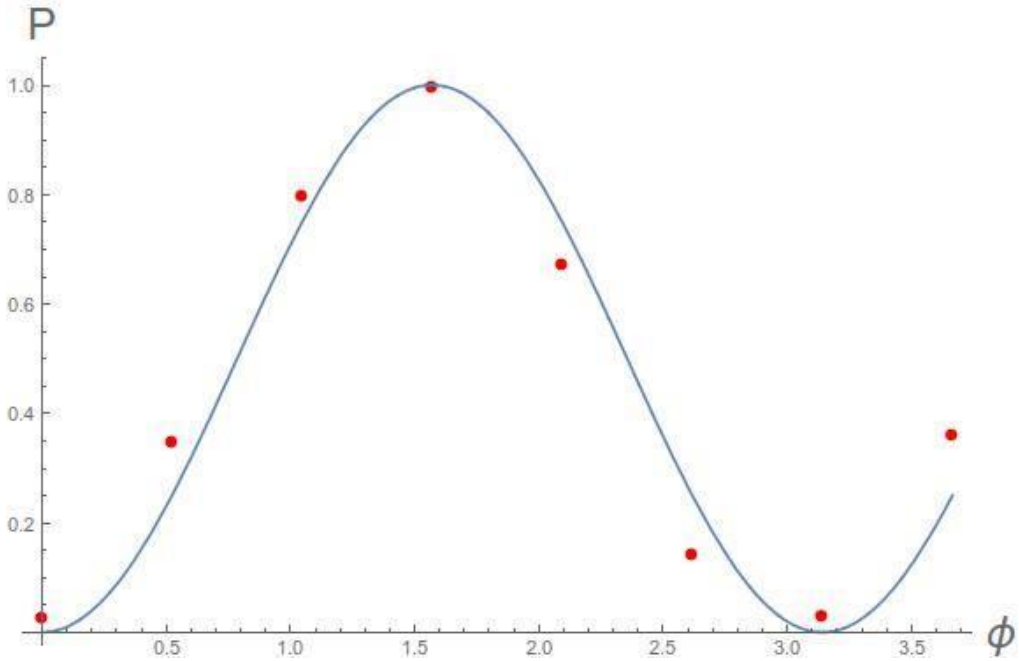

Figura 43. Interferograma. Probabilidad de detección vs. ϕ.

La curva de color azul corresponde a los valores teóricos, mientras que los puntos de color rojo son los datos experimentales. A cada dato experimental se le puso barras de error. Sin embargo, estas no se pueden observar ya que los errores son demasiado pequeños y el punto rojo cubre estas barras de error, lo cual nos indica que las mediciones realizadas son bastante exactas.

Los datos experimentales no coinciden exactamente con la curva teórica debido a que cuando uno calibra las placas retardadoras, el mínimo valor registrado de intensidad no alcanza a 0 mW. ya que el potenciómetro utilizado es bastante sensible. Además, a los motores se les giró cada 0.5° lo cual introduce una pequeña fuente de error acumulable.

Adicional a estas fuentes de error, una más que se incluye es que en el proceso de la medición, utilizando los tres motores rotatorios, se dio una tolerancia de 0.3° a las posiciones angulares a las cuales se deben girar a los motores.

CONCLUSIONES

El presente trabajo tuvo por objetivo principal el desarrollo de un programa que permite controlar hasta 8 motores rotatorios a los cuales se les acopló un acelerómetro ADXL345. En cada motor rotatorio, además, se colocó una placa retardadora. Tal tarea se llevó a cabo utilizando el software LabVIEW, una placa Arduino UNO conectada a través de un cable USB, los motores rotatorios y su interface. Se logró que los motores rotatorios se muevan, uno tras otro, a una posición angular previamente fijada, a la vez que se mide con un potenciómetro la intensidad de un haz láser que atraviesa a las placas retardadoras.

La primera tarea que se realizó consistió en calibrar las placas retardadoras para encontrar en qué posición angular se encontraba su eje rápido, lo cual fue la base para poder hacer las siguientes tareas.

La segunda tarea realizada consistió en reproducir una medición hecha en el Laboratorio de Óptica Cuántica, que fue publicada en el año 2009 en Physical Review [6]. Los resultados de esta medición resultaron ser muy exactos, ya que su error es muy pequeño.

Se tenía planeado inicialmente realizar mediciones que incluyeran más motores rotatorios con sus respectivos acelerómetros y placas retardadoras. Sin embargo, al momento de realizar la soldadura de los acelerómetros con cables de larga extensión no se empleó una técnica adecuada ya que algunos sensores no enviaban datos coherentes a la PC. Se recomienda que esta soldadura las realice un especialista, ya que los sensores son sensibles y podrían ser dañados.

Por otra parte, el algoritmo empleado para el sensor ADXL345 tiene la desventaja de pasar de la indicación de 359.9° a 0°, y viceversa. Por ésta razón, se recomienda ubicar al eje rápido de las placas retardadoras entre los valores de 150° y 210°, lo cual permitiría desplazar al motor en un rango de aproximadamente 150° sin ningún inconveniente.

BIBLIOGRAFIA

[1] D. Sutter, G. C. Chingas, R. A. Harris, and A. Pines, Mol. Phys. 61 (1987), 1327.

[2] G. S. Agarwal, Phys. Rev. A 38 (1988), 5957.

[3] A. G. Wagh and V. C. Rakhecha, Phys. Lett. A 148 (1990), 17.

[4] R. Y. Chiao and Y. S. Wu, Phys. Rev. Lett 57 (1986), 933.

[5] G. S. Agarwal and R. Simon, Phys. Rev. A 42 (1990), 6924.

[6] J.C. Loredo, O. Ortíz, A. Ballón, and F. De Zela, Phys. Rev. A 80, 012113 (2009).

[7] J. SIMMONS, M.GUTTMANN, *States, waves and photons: A modern introduction to light*. (Estados Unidos. Addison-Wesley Publishing Company, Inc. 1970), p. 53

[8] Ref. [7], p 42

[9] Ref. [7], p 75

[10] De Zela, Francisco. "The Pancharatnam-Berry Phase: Theoretical and Experimental Aspects". EN: *Theoretical Concepts of Quantum Mechanics*. InTech. 2012. p. 294

[11] Agilis TM Series, Piezo Driven Components, p. 5

[12] Agilis-UC2 LabVIEW driver recommendations, p. 5

[13] R. Galeas, E. Llana, Programación gráfica con LabVIEW (Perú. Megabyte S.A.C., 2011), p. 3

[14] Ref. [13], p. 17

[15] Ref. [13], p. 29

[16] ANALOG DEVICES Digital accelerometer ADXL345, p. 1

[17] https://www.arduino.cc/ - revisado el 30/10/2015

[18] https://store.arduino.cc/product/GBX00066 - revisado el 30/10/2015

[19] J. Lajara, J. Pelegri, Sistemas integrados con Arduino (México. Alfaomega Grupo Editor), pag. 3

[20] http://www.elecfreaks.com/wiki/index.php?title=Triple_Axis_Accelerometer_-_ADXL345 – revisado el 08/09/2015

[21] M. Pérez, M. Ortíz, Película de Celofán como retardador de media longitud de onda de amplio espectro, p. 23

APÉNDICE: Cómo utilizar la librería LINX en LabVIEW

Una vez conectado el sensor acelerómetro ADXL345 al Arduino, y estando éste conectado a la PC, se siguen los siguientes pasos para que LabVIEW pueda leer los datos enviados por el sensor:

1) Hacer clic en “Tools” ubicado en la barra de herramientas, luego se hace clic en la opción “MakerHub”, y para finalizar clic en “LINX Firmware Wizard”, como se muestra en la siguiente figura.

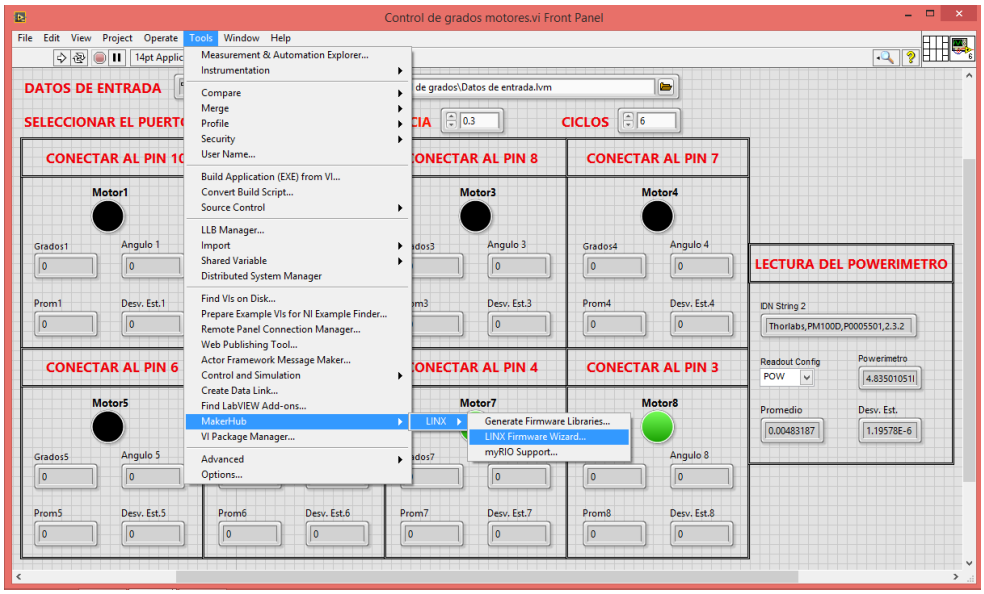

Figura A1.

2) Aparecerá la siguiente ventana. Aquí en la opción “Device Family” uno debe seleccionar la familia Arduino. En la opción “Device Type” se selecciona el tipo de tarjeta de Arduino que se está utilizando; en nuestro caso fue el Arduino UNO. Y en la opción “Firmware Program Interface” se debe seleccionar el tipo que conexión que se emplea; para nuestro caso Serial/USB.

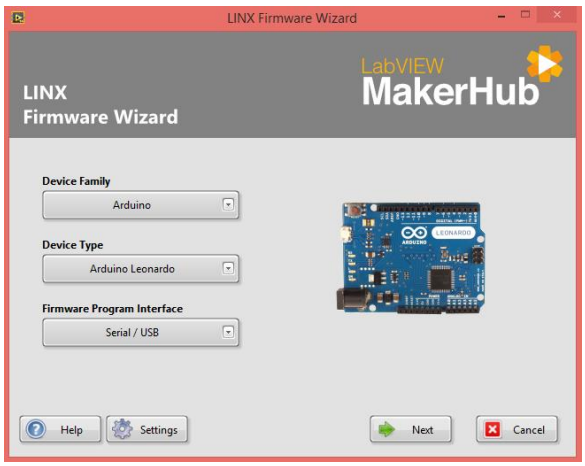

Figura A2.

3) En la siguiente ventana se selecciona el puerto en el cual la PC reconoce a nuestro hardware del Arduino.

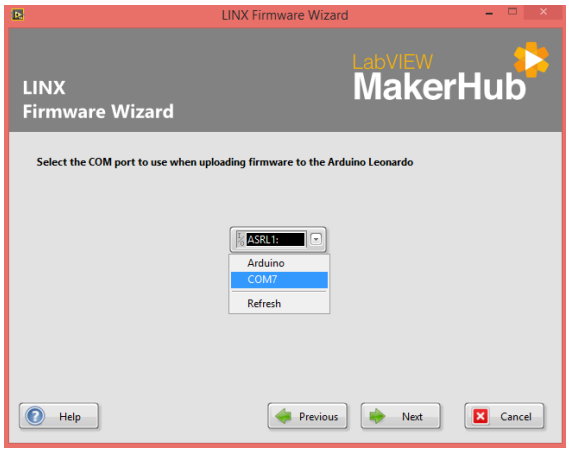

Figura A3.

4) Aquí la librería está cargándose para que LabVIEW reconozca los datos provenientes del Arduino.

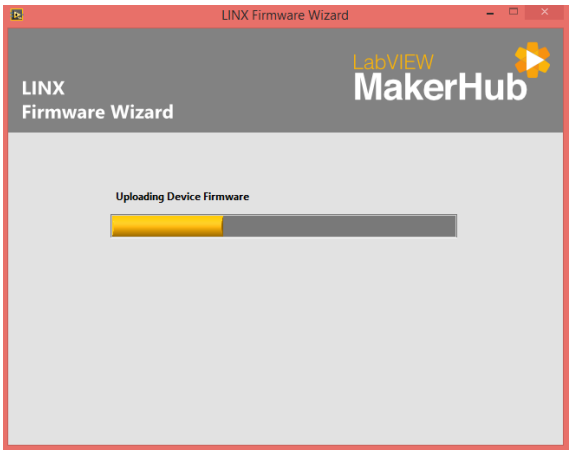

Figura A4.

5) Haciendo clic en “Finish”, LabVIEW se encuentra listo para leer los datos provenientes del Arduino.

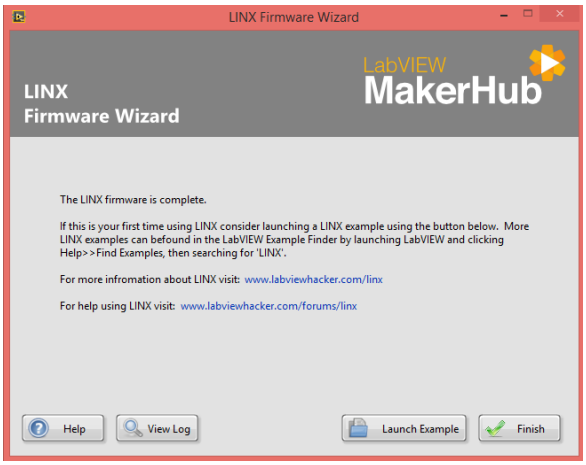

Figura A5.