

UNIVERSIDAD NACIONAL AUTÓNOMA DE
MÉXICO

FACULTAD DE INGENIERÍA



CONTROL EN TIEMPO REAL DE UN GIRÓSCOPO

TESIS

QUE PARA OBTENER EL TÍTULO DE
INGENIERO ELÉCTRICO ELECTRÓNICO
PRESENTA:

FRANCISCO JAVIER SERGIO SALAZAR
MELCHOR

ASESOR:

Ing. ALBERTO RAMIRO GARRIBAY MARTÍNEZ



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

Agradecimientos a:

**MAO. Guadalupe Melchor Díaz. Profesor de carrera de la Universidad
Autónoma del Estado de México**

**Ing. Francisco José Rodríguez Ramírez. Profesor de carrera de la
Facultad de Ingeniería de la Universidad Nacional Autónoma de México**

Índice General

Introducción

1. OBJETIVO

1.1. Tarjeta NI PCI-6024E y BNC2120

1.2. Módulo RT

1.3. LABVIEW

2. MARCO TEÓRICO Y CONCEPTUALES.

2.1. Giróscopo

2.2. Encoder

3. DESARROLLO DEL PROCESO

3.1. Descripción del Giróscopo (GYRO/ STABILIZED PLATAFORM)

3.2. Modelo del módulo SRV-02

3.3. Parámetros

4. Programación de la solución en LABVIEW

4.1. Control PD

4.11 Métodos de sintonización

4.12 Método de oscilaciones amortiguadas (Harriot 1957)

4.13 Método de la curva de reacción (Ziegler-Nichols 1942)

4.2 PROGRAMACIÓN DEL CONTROLADOR (PD) EN LABVIEW RUN-TIME 7.1

4.21. Ciclo While con Temporizador

4.22 Proceso de adquisición de lectura del encoder (Digital)

4.23 Rutina de conversión de lectura del Encoder a posición angular

4.24 RUTINA DE CONTROL (PD)

4.25 Rutina para el envío de la señal de control a los motores del SRV02

4.26 Rutina del proceso de adquisición del Tacómetro

4.27 Measurement & Automation Explorer

4.28 Descripción del panel Frontal.

4.3. PROGRAMACIÓN DEL CONTROLADOR (PD) EN LABVIEWRUN-TIME 7.1

4.4 Desempeño en el Tiempo del Controlador en Tiempo Real

5. Aplicación experimental y resultados

5.1. Análisis del Srv02-Gyro

5.2. PRUEBA EN LABVIEW 8.6

5.3 PRUEBA EN LABVIEWRUNTIME7.1.

6. Conclusiones

Bibliografía

Apéndice

Introducción

En sistemas de ingeniería es necesario que una tarea específica, como la operación de una máquina, o un determinado proceso, se desarrolle de acuerdo con un cierto desempeño deseado. Por ejemplo, un automóvil está diseñado para reaccionar a los comandos que el conductor decida realizar: ya sea a través del volante, para definir la dirección y sentido a tomar; o a través de los pedales de aceleración y de freno, para decidir la rapidez en el movimiento del mismo. Igualmente deberá proveer una operación segura y estable, a fin de reducir el riesgo de ocurrencia de un accidente. En suma, el conductor debe poseer el control absoluto en la operación del vehículo, estableciendo siempre la velocidad por él deseada, con el claro propósito de llegar a su destino sin haber sufrido percance alguno. Por lo tanto el control, como en éste y otros tantos casos, juega un papel primordial.

Controlar significa lograr que un sistema funcione de acuerdo con ciertas especificaciones establecidas por el operador o usuarios del mismo. Un sistema de control es la conjunción de métodos, técnicas, herramientas y equipo, orientados en el cumplimiento de este objetivo. Los sistemas de control son responsables en gran medida del desarrollo tecnológico y grado de bienestar actual, y seguramente lo seguirán siendo en el futuro. Prácticamente en cualquier sistema, equipo, o máquina está involucrado un sistema de control: desde el aire acondicionado que regula la temperatura de un cuarto, hasta un cohete espacial, en donde es fundamental controlar a cada instante su trayectoria, posición y velocidad.

Anteriormente se mencionó que un sistema de control no está solamente compuesto por equipo (de tipo mecánico, eléctrico, hidráulico o neumático, entre otros) e instrumentos, también se requieren métodos, técnicas, y algoritmos de control, en los cuales se establece cómo habrán de calcularse las acciones de control pertinentes, para llevar a un valor deseado las variables bajo control. Con base en el algoritmo, un microprocesador o cualquier otro circuito electrónico analógico y/o digital, por citar sólo una posibilidad, podrá realizar el cálculo de dichas acciones de control: a esta parte del sistema se le conoce como controlador o compensador.

En general, podemos diferenciar estos métodos o estrategias de control en dos categorías: control de lazo abierto y de lazo cerrado. La diferencia radica en la necesidad de los segundos de conocer el estado actual del proceso, a fin de calcular la acción de control, o dicho de otra manera, en el control de lazo cerrado debemos realimentar el estado presente del proceso.

Un sistema de tiempo real es aquél donde su correcta operación depende de la obtención de los resultados en un tiempo oportuno, a dicha característica en la operación en tiempo real se le conoce como determinismo. En el ejemplo del automóvil, el sistema de frenado del vehículo debe ser visto como un sistema de tiempo real, ya que el tiempo transcurrido entre la acción ejercida sobre el pedal de freno y la efectiva reducción de la velocidad del automóvil, no debe de rebasar cierto límite de tiempo. Asimismo, la mayoría de los sistemas de control realimentado de tipo industrial, requieren de una limitante o un rango en el tiempo de ejecución del lazo para la operación segura del proceso y evitar condiciones de riesgo y falla.

La característica de tiempo real en los sistemas de control, requieren en su implantación del uso de computadoras, microcontroladores o microprocesadores, los cuales, cuentan con la preferencia

de los desarrolladores por su bajo costo y versatilidad, de ahí que el presente trabajo refiera la comparación entre un sistema que no trabaja en tiempo real y otro que si lo realiza, para verificar la eficiencia del desempeño de un control.

1. OBJETIVO

El objetivo de este proyecto, es realizar el control para un giróscopo mediante un control PD programado en LabVIEW y mostrar la importancia del tiempo de adquisición de datos en los sistemas realimentados.

1.1.-TARJETA DAQ NI PCI-6024E y BNC2120

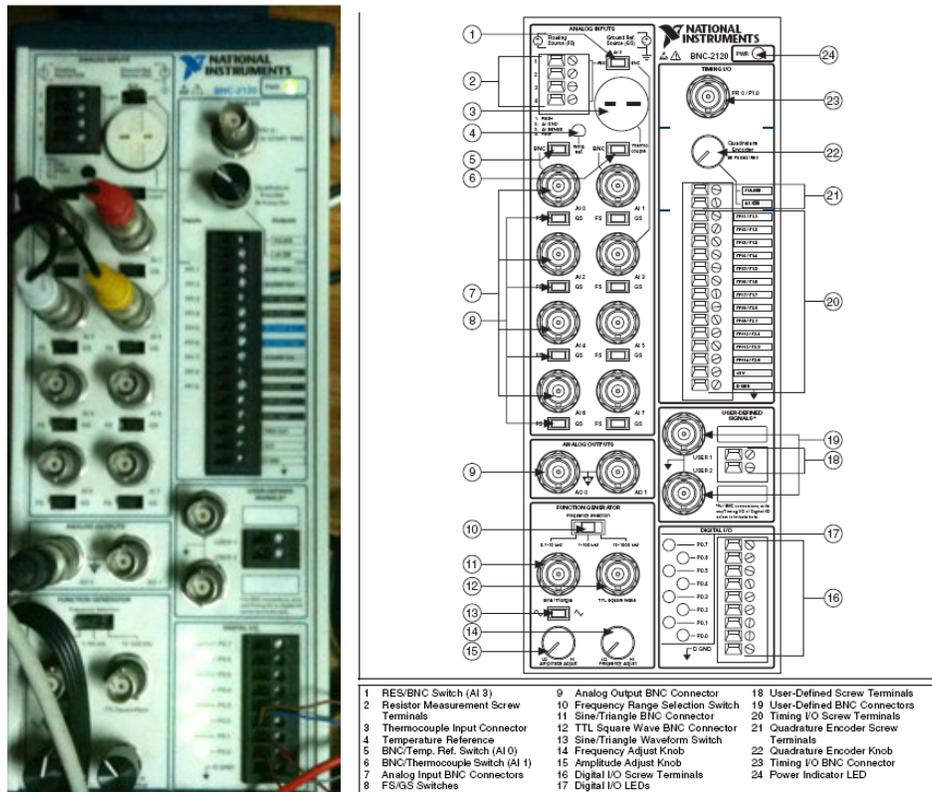


Figura 1: TARJETA BNC2120 NATIONAL INSTRUMENTS

La tarjeta BNC2120 consta de:

- 8 canales analógicos de entradas (BNC).
- 2 canales analógicos de salida (BNC).
- 1 generador de funciones.
- 8 entradas/salidas digitales.

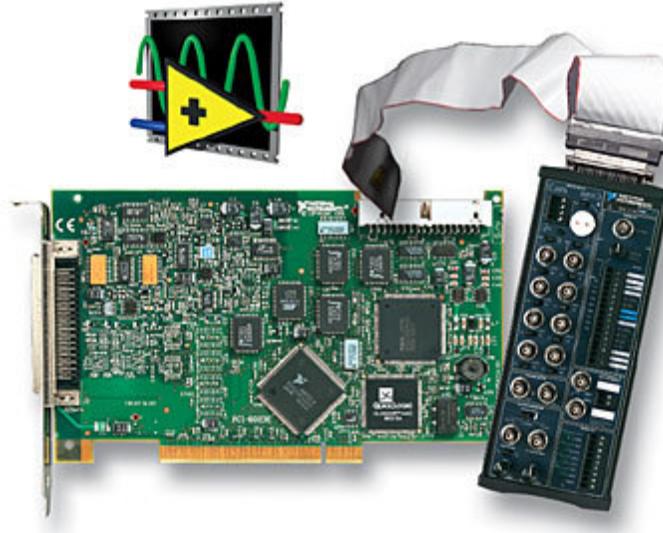


Figura 2: NI PCI-6024E/BNC-2120

La tarjeta NI PCI-6024E consta de:

- 16 entradas analógicas multifunción con resolución de 12 bits y un muestreo de 200 kS/s
- 2 salidas analógicas a 12 bits
- 8 líneas de E/S digitales.
- 2 contadores de 24 bits.

1.2 Módulo RT

En sistemas de ingeniería existen situaciones en las cuales el tiempo en el que una acción debe realizarse es muy importante por ejemplo supóngase un sistema de control que verifica la operación correcta de un artefacto, y que en presencia de una falla en dicho artefacto, deba generar una señal de alerta y posiblemente acciones, que le permitan evitar daños o inclusivamente la destrucción del equipo, al mismo tiempo de evitar daños a sus operadores. En este caso las acciones a realizar deben ser lo más rápido posible, ya que de llevarse a cabo en tiempo prolongado puede ocasionar severas consecuencias en el artefacto como en el operador, por ello es necesario asegurar que el sistema de control actué dentro de un límite de tiempo máximo con el propósito de lograr un funcionamiento efectivo del control de la falla (al referir al tiempo real implica que el tiempo es una variable de importancia crítica).

Un sistema de tiempo real es aquel sistema de procesamiento de la información que debe de responder a los estímulos de entrada externamente generados, dentro de un período de tiempo finito y especificado. El determinismo es el correcto funcionamiento de un sistema de tiempo real

que no solo depende de la obtención de los resultados esperados, sino de que estos se presenten en el tiempo en que deben de ser producidos.

El concepto de sistema de tiempo real no se limita a las aplicaciones de control, sino que también es aplicable a otros campos, como los sistemas de información, telecomunicaciones o de procesamiento de datos en general así como, en la respuesta a eventos.

Para el control de lazo cerrado de procesos, un sistema de control en tiempo real es aquel que monitorea a un proceso, compara el estado actual del proceso con el estado deseado, calcula la acción de control a realizar y ejecuta dicha acción sobre el proceso con el propósito de alterar la respuesta del mismo proceso bajo control. El tiempo que lleva ejecutar este ciclo es de consideración crítica y se conoce como tiempo de ciclo del lazo de control, el cual varía de acuerdo a la complejidad del sistema y el proceso bajo control.

Una primera característica de los sistemas en tiempo real es el determinismo porque se encarga de verificar el cumplimiento del intervalo de tiempo especificando dentro de ellos la tarea o acción que deben ejecutar y completar. Aunque esto es deseable, puede no siempre cumplirse y existir un cierto error que, que se denomina "jitter", el cual se calcula como la diferencia máxima entre cualquier tiempo de ejecución que rebasa a la restricción y el tiempo de ejecución especificado.

La segunda característica de los sistemas en tiempo real es la confiabilidad y seguridad de operación, ya que la operación de un proceso está confiada a que un sistema de control autónomo sistema no falle; y si lo hace, que lo haga de manera controlada. Es necesario asegurar que una probable equivocación humana no ponga en peligro la estabilidad del proceso, sin que ello signifique necesariamente una nula interacción con el operador.

La tercera característica de importancia es la implantación eficiente de los sistemas dado que, en una aplicación de tiempo real el tiempo es una consideración crítica por lo que debe buscarse la implementación, con el objetivo de cumplir con tales restricciones en el tiempo. En los sistemas de tiempo real, la eficiencia desempeña un papel de mayor relevancia, comparativamente a otros sistemas.

La cuarta característica es la existencia de un reloj de tiempo real, el cual utiliza como base de tiempo la operación de todo el sistema

Descripción de la plataforma de desarrollo LabVIEW REAL-TIME 7.1 y PCI 7041/6040E RT

Todas las plataformas de desarrollo de LabVIEW REAL-TIME están basadas en una misma arquitectura de hardware y software. Una computadora personal funciona como sistema de desarrollo, en donde se diseña y programa la aplicación en tiempo real; y desde donde se puede probar y depurar la misma operación, que se ejecuta en una tarjeta objetivo de tiempo real. El medio por el cual se comunican el sistema de desarrollo y la tarjeta de tiempo real es la memoria compartida.

En LabVIEW 7.1 se proporcionan todos los mecanismos para la carga y descarga de la aplicación en la tarjeta objetivo de tiempo real. Así como se prueba la comunicación entre la tarjeta y el sistema de desarrollo en etapas de prueba y depuración de código. También contiene ciertas herramientas y elementos de programación propios de una aplicación en tiempo real. En LabVIEW Real-Time Engine 7.1 se permite ejecutar la aplicación desarrollada en un tiempo que trabaje con un sistema operativo de tiempo real que este caso es la tarjeta PCI-7041/6040E.

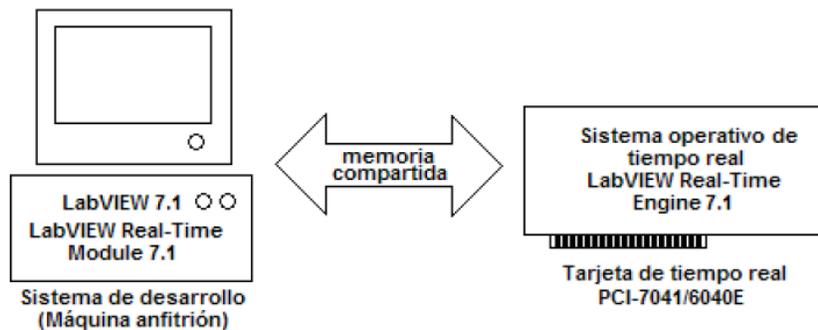


Figura 3: Arquitectura de la plataforma de desarrollo en tiempo real con LabVIEW

La tarjeta de tiempo real PCI-7041/6040 RT. está conectada al bus PCI de la maquina anfitrión, integrada por dos subtarjetas permanentemente unidas: la tarjeta procesadora 7041 y la multifunción 6040E

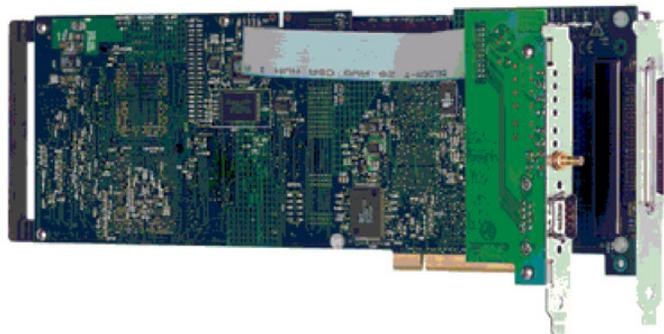


Figura 4: Tarjeta de tiempo real PCI-7041/6040E.

La parte procesadora de la tarjeta contiene los mismos elementos básicos de una computadora personal: un procesador, buses de control, de datos y de información, memoria volátil y no volátil. Es en esta parte donde la aplicación de tiempo real se ejecuta de forma completamente independiente y continua. Ya que en el procesador se ejecuta un sistema operativo de tiempo real la aplicación opera de acuerdo a las prioridades especificadas por el programador, resultando lo anterior en un grado de determinismo de la aplicación.

La tarjeta multifunción de adquisición de datos (DAQ) 6040E incluye medios de conectividad con 16 entradas analógicas y, 2 salidas analógicas, 1 puerto digital de 8 bits, y señales de temporizador. Para entradas y salidas analógicas, la tarjeta trabaja con un amplificador de instrumentación de ganancia programable, un multiplexor y un convertidor analógico/digital y otro digital/analógico. Dado que está conectada permanentemente a la tarjeta procesadora, entonces es posible operar en la aplicación de tiempo real sobre señales externas reales.

El software LabVIEW Real-Time Module 7.1 extiende el lenguaje gráfico de programación de LabVIEW a la construcción de aplicaciones en tiempo real. Con él se programa en el mismo entorno característico de LabVIEW además de que agrega elementos de programación típicos de las aplicaciones en tiempo real. También se realiza la descarga automática del programa del sistema de desarrollo a la tarjeta, además de proveer la comunicación entre ambos sistemas para la prueba y depuración del código de la aplicación.

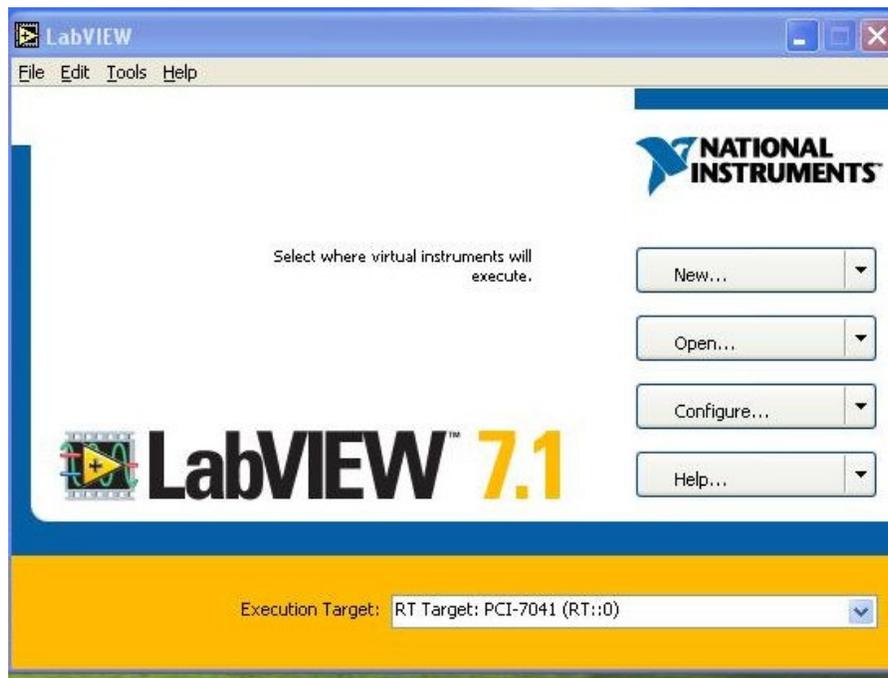


Figura 4: Pantalla de bienvenida de LabView7.1.

LabVIEW PID Control Toolset es un conjunto de herramientas que incluye bloques programados destinados a la creación de controladores PID, de la lógica difusa, o con base en algoritmos avanzados de control.

LabVIEWReal-Time Engine7.1 es una versión de LabVIEWReal-Time 7.1 diseñada para ejecutarse sobre un sistema operativo de tiempo real. Con ello se asegura que la ejecución de la aplicación cumpla con las restricciones en tiempo y prioridades especificadas por el programador.

1.3 LABVIEW

LabVIEW es un revolucionario entorno de desarrollo gráfico con funciones integradas para realizar adquisición de datos, control de instrumentos, análisis de medida y presentaciones de datos. LabVIEW le da la flexibilidad de un potente ambiente de programación, pero mucho más sencillo que los entornos tradicionales.

A diferencia de los lenguajes de propósito general, LabVIEW tiene funciones específicas para acelerar el desarrollo de aplicaciones de medida, control y automatización.

LabVIEW proporciona herramientas muy potentes para crear aplicaciones sin líneas de código además, permite colocar objetos ya construidos para crear interfaces de usuario rápidamente especificando las funciones del sistema mediante la construcción de diagramas de bloques.

LabVIEW puede conectarse de manera transparente con todo tipo de hardware incluyendo instrumentos de escritorio, tarjetas insertables, controladores de movimiento y controladores lógicos programables (PLCs). Además puede conectarse con otras aplicaciones y compartir datos a través de ActiveX, Web, DLLs, librerías compartidas, SQL, TCP/IP, XML, OPC y otros.

LabVIEW tienen velocidades de ejecución comparables con programas C compilados y permite desarrollar sistemas que cumplan con requisitos de desarrollo a través de las plataformas incluyendo Windows, Macintosh, UNIX y sistemas de tiempo real.

LabVIEW está integrado con el hardware de medida, por lo que se puede configurar y usar rápidamente cualquier dispositivo de medida que se tenga también, puede conectarse a miles de instrumentos para construir sistemas de medida completos, incluyendo desde cualquier tipo de instrumento autónomo hasta dispositivos de adquisición de datos, controladores de movimiento y sistemas de adquisición de imagen.

LabVIEW permite la construcción de sistemas definidos por el usuario más rápido que los métodos tradicionales, debido a que las necesidades de las aplicaciones cambian, los sistemas definidos por el usuario tienen la flexibilidad necesaria para poder modificarlos sin la necesidad de incorporar equipos nuevos. Utilizando un sistema basado en LabVIEW, tiene acceso a sistemas de

instrumentación completos con un coste mucho más bajo que un único instrumento comercial. National Instruments también asegura que los programas que desarrolla hoy pueden migrar para aprovechar las tecnologías del futuro.

LabVIEW está optimizado para el desarrollo de las aplicaciones de medida y automatización más exigentes. Debido a que la instrumentación virtual está basada en la tecnología informática estándar. Además, LabVIEW se caracteriza por un compilador gráfico optimizado en mutihilo para maximizar el rendimiento del sistema permitiendo desarrollar sistemas con el rendimiento necesario para las aplicaciones más exigentes, tanto en laboratorio como en producción, un sistema de medida más rápido se traduce en un aumento de la producción.

LabVIEW tiene extensas capacidades de adquisición, análisis y presentación disponibles en un sólo software, lo que permite crear una solución completa de manera única en la plataforma que se elija. Permite publicar aplicaciones de datos en la Web fácilmente o conectarse a otras aplicaciones a través de una variedad de tecnologías estándar, como TCP/IP, DLLs y ActiveX.

LabVIEW simplifica el desarrollo de sistemas y produce un código reutilizable que se ejecuta a velocidades de código compilado, también puede crear ejecutables autónomos o librerías compartidas y DLLs para poder llamar desde otros entornos como Microsoft Visual Basic o Measurement Studio de National Instruments.

2. MARCO TEÓRICO Y CONCEPTUALES.

2.1 Giróscopo

Un giróscopo es un artefacto que gira rápidamente sobre un eje libre que se mantiene en una dirección constante. Se utiliza en la estabilización de barcos y aviones.

Por ejemplo Podemos decir que el giróscopo es el piloto automático que nos va a mantener el rumbo ó dirección del helicóptero de una manera totalmente controlada. Dada la constante evolución que a través de los años han experimentado y según la tecnología empleada, vamos a clasificarlos en dos grandes grupos:

- A) Convencionales: - Mecánicos -Piezoeléctricos - Con bloqueo de la cabeza:
- B) Efecto " heading hold (también piezoeléctricos).

La diferencia fundamental entre los convencionales y los de bloqueo de la cabeza, consiste en que ante una misma perturbación producida por un agente exterior, como podía ser un golpe de viento, la reacción entre ambos será diferente y aunque los dos tratarán de corregir el rumbo alterado por dicho golpe de viento, los primeros lo harán transcurrido un tiempo, tiempo que aunque mínimo será el causante de lograr un rumbo diferente al inicial cuando la estabilidad se consiga, mientras que los segundos, "recuerdan siempre" la posición inicial que se tenía antes de la perturbación y allí dirigen la cabeza del helicóptero una vez pasada la misma.

Para entender mejor el efecto "heading hold", podemos imaginar que el giróscopo transforma al helicóptero en una auténtica brújula magnética, con la única diferencia que en una brújula real, la aguja señala siempre al invariable Polo Norte magnético de la Tierra y en cambio nuestro giróscopo, señalará la dirección que el helicóptero tenga en el momento de conectar la emisora; de ahí la importancia de no moverlo en los primeros segundos de conexión, momento que necesitan los circuitos electrónicos del giróscopo, para memorizar la posición del hipotético Polo Norte.

La rigidez en el espacio de un giróscopo es consecuencia de la primera ley del movimiento de Newton, que afirma que un cuerpo tiende a continuar en su estado de reposo o movimiento uniforme si no está sometido a fuerzas externas. Así, el volante de un giróscopo, una vez que empieza a girar, tiende a seguir rotando en el mismo plano en torno al mismo eje espacial.

Los giróscopos constituyen una parte importante de los sistemas de navegación automática o guiado inercial en aviones, naves espaciales, misiles teledirigidos, cohetes, barcos y submarinos. Los instrumentos de guiado inercial de esos sistemas incluyen giróscopos y acelerómetros que calculan de forma continua la velocidad y dirección exactas del vehículo en movimiento. Estas señales son suministradas a una computadora, que registra las desviaciones de la trayectoria y las compensa.

2.2 Encoder

El encoder es un transductor rotativo que transforma un movimiento angular en una serie de impulsos digitales. Se basa en la rotación de un disco graduado con un retículo radial formado por espacios opacos, alternados con espacios transparentes. Un sistema óptico de emisor receptor infrarrojo detecta el cambio en la superficie del disco, generando dos señales de cuadratura (desfasadas 90°), las señales se identifican como A y B.

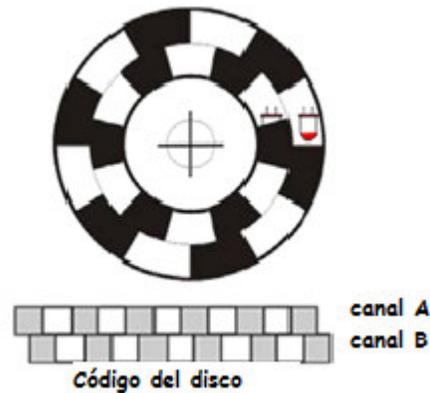


Figura 5 Encoder.

El encoder codifica la información del desplazamiento y su dirección, normalmente el mínimo desplazamiento es de codificado a partir de un ciclo completo de un ciclo completo de la señal A o B. observando detalladamente la señal de cuadratura se puede apreciar que hay información del desplazamiento en cada flanco de las señales A o B. observando detalladamente la señal de cuadratura se puede apreciar que hay información del desplazamiento en cada flanco de las señales A y B, por lo que es posible decodificar la información del desplazamiento y la dirección.

Incrementar la resolución de un encoder permite mayor precisión con el mismo dispositivo.

En las señales A y B en cuadratura se encuentra codificada la información correspondiente al avance y su dirección, la cual puede ser en sentido de la manecillas del reloj (Clockwise CW) o en sentido contrario (CounterClockWise CCW).

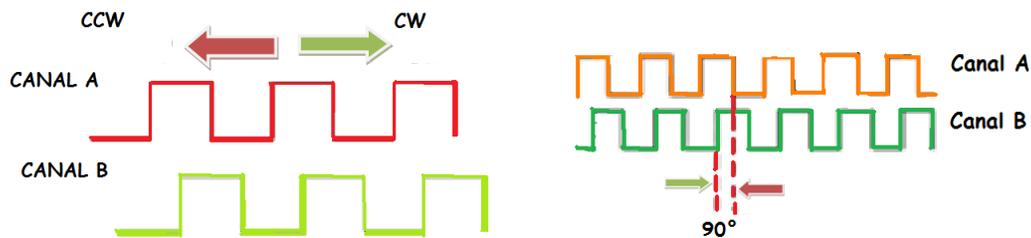


Figura 6 Señales de cuadratura

La información correspondiente al desplazamiento se obtiene directamente de A o B, un ciclo de la señal correspondiente al mínimo avance, se puede usar como referencia el flanco de subida o de bajada: para nuestro encoder que es de 4096 pulsos por revolución, el mínimo avance corresponde a $360^\circ/4096=0.087^\circ$. Para determinar la dirección del desplazamiento se requieren ambas señales; se tiene un circuito mediante el cual se determinan ambas señales.

Se implementa un Flip Flop tipo "D", la señal A se emplea como dato y B como señal de reloj, en sentido CW (Izquierda -derecha) se captura siempre un nivel alto de A. Para el sentido CCW (Derecha-Izquierda) el flanco de subida B coincide con el nivel bajo de A.

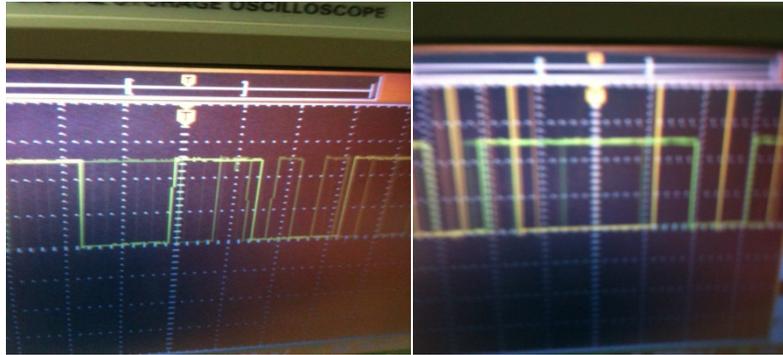


Figura 7 Señales del encoder del SRV-02.

3.-DESARROLLO DEL PROCESO

3.1 Giróscopo (GYRO/ STABILIZED PLATAFORM)

El del giróscopo ocupado consta de los siguientes componentes:

1. Placa base.
2. Placa Soporte.
3. Módulo del Giróscopo:
 - A) Motor de DC.
 - B) Disco de Inercia.
 - C) Resortes
 - D) Sensor y Engranés.

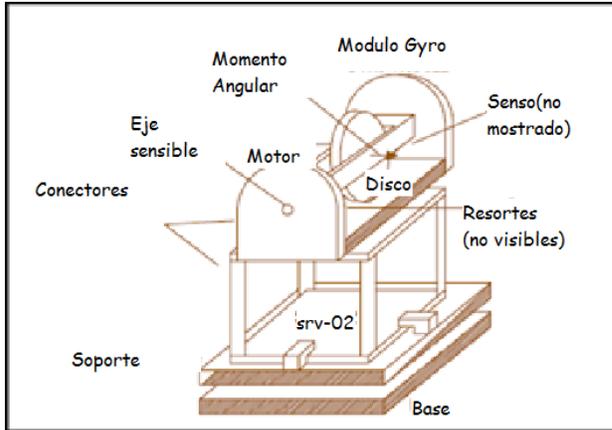


Figura 8 Módulo del giróscopo colocado sobre el servo SRV02
(Plataforma referenciada hacia una mesa o base cualquiera)

El experimento consiste en diseñar un control que mantenga una posición, mientras se rota el soporte de la base. Esto representa el problema de colocar un radar en una nave, donde se desea mantener la dirección de la luz en el radar independientemente de la pérdida del mantenimiento de curso de la nave debido a perturbaciones.

3.2 Modelo del módulo SRV-02

Conociendo modelo básico de funcionamiento de un giróscopo:

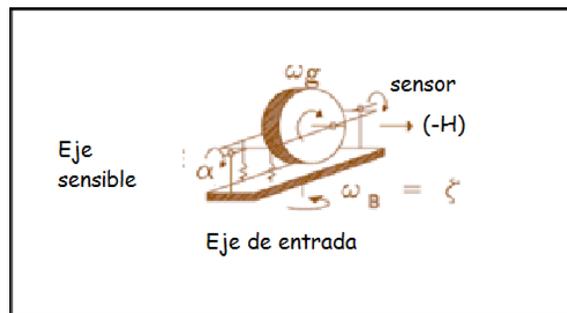


Figura 9 Modelo de un giróscopo

Considerando que el disco gira con una velocidad constante relativa ω_g . Cuando la base gira a una velocidad de ω_b se obtiene una fuerza de giro, referenciada al eje sensitivo, dada por:

$$T_g = W_b H$$

Donde H es el momento angular del volante que esta dado por:

$$H = J_f W_g$$

Los resortes montados en el giróscopo se oponen a la fuerza de giro y esta dado por:

$$T_s = K_R \alpha$$

$$T_s = T_g$$

Donde K_r es la tensión que ejercen los resortes.

Así el ángulo de inclinación α de con respecto al eje de giro está en función de W_b , que está dada por la función:

$$\alpha = \frac{W_b h_g}{K_g} (\text{rad})$$

La ganancia del giro está dada por:

$$G_g = \frac{\omega_b}{\alpha}$$

$$G_g = \frac{K_R}{H} \left(\frac{\text{°/s}}{\text{° de inclinación}} \right)$$

Así la inclinación al eje sensible del giróscopo es directamente proporcional a la velocidad a la velocidad de rotación de la base. (Solamente en el espacio de estados. Nótese que se ignora la dinámica en el eje sensible, un modelo más completo incluye la dinámica $\frac{\alpha(s)}{\omega_b(s)} = H(s)$), lo cual quiere decir que el ángulo de inclinación α puede ser usado para medir la rotación de la plataforma con relación a la base sin una medición directa.

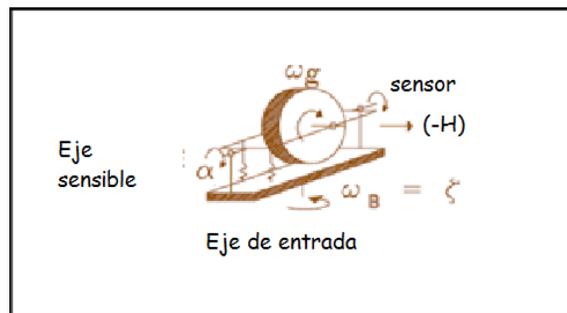


Figura 10 Modelo simplificado del giróscopo

El ángulo de inclinación α esta medido a través de un potenciómetro que es sensible al eje. Usando una escala de 4:1, el ángulo medido por el potenciómetro esta dado por $\beta = 4\alpha$, por lo cual el rango de giro de la base esta dado por:

$$\widehat{\omega}_b = G_g \alpha$$

$$\omega_b = G_g \frac{\beta}{4} = \frac{K_g}{4H} \beta$$

El giro que la base realiza es obtenido integrando la velocidad medida.

$$\xi = \int \widehat{\omega}_b \sigma t$$

3.3 Parámetros

Algunos de los parámetros del SRV-02 son:

- **Velocidad de Giro:**

El motor es uno de DC de 12V, cuya ecuación está dada por:

$$V_{gm} = I_{gm}R_{gm} - K_{gb}\omega_g$$

Donde el circuito ideal tiene corriente igual a cero, pero en este caso la fricción produce una pequeña corriente en dirección del motor. Para el espacio de estados tendremos que la velocidad angular será:

$$\omega_g = \frac{V_{gm} - I_{gm}R_{gm}}{K_{gb}}$$

Donde:

$$V_{gm} = 12 \text{ Volts DC}$$

$$R_{gm} = 5.3\Omega$$

$$K_{gb} = 0.0235 \frac{V}{rad/s}$$

Considerando que no hay fricción tenemos que la velocidad del disco será:

$$\frac{V_g}{K_g} = 510 \left(\frac{rad}{s} \right)$$

Una mejor estimación es obtenida midiendo la corriente utilizando un amperímetro, obteniendo una corriente de 0.23 amperes resultando una velocidad:

$$\omega_g = 510 \frac{rad}{s}$$

- **Momento de giro:**

La inercia del disco está dada por:

$$J_f = M_f \frac{r_f^2}{2}$$

$$J_f = 0.8 \frac{(2 \times 0.254)^2}{2} = 1.0323e^{-3} Kgm^2$$

Así el momento será:

$$h_g = 457 \times 1.0323e^{-3}$$

$$h_g 0.4718 Kgm^2/s$$

- **Rigidez común**

Los dos resortes se encuentran de la siguiente manera:

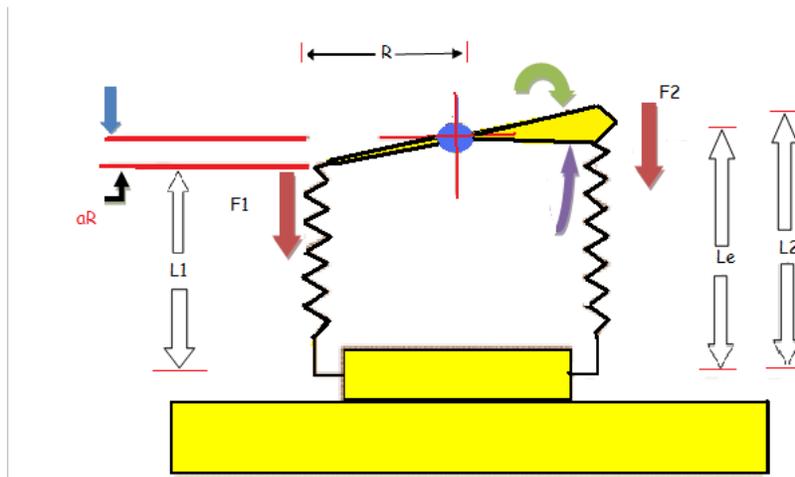


Figura 11 Modelo de puntos de rigidez

La rigidez al eje de rotación es interpretada de la siguiente manera:

Asumiendo que los coeficientes de los resortes son constantes K_S y una elongación de L_u , la elongación de los resortes en condiciones normales ($\alpha=0$) estará dada por L . suponiendo que el eje está girado una distancia α , entonces las 2 fuerzas estarán dadas por:

$$F_1 = K_s \delta L_1$$

$$F_1 = K_s \delta (L - L_u - R \alpha_s)$$

$$F_2 = K_s \delta L_2$$

$$F_2 = K_s \delta (L - L_u + R \alpha_s)$$

Entonces la fuerza de giro generado por las fuerzas será:

$$T = R(F_1 - F_2)$$

$$T = 2R^2 K_s \alpha$$

Entonces la rigidez rotacional estará dada por:

$$K_g = \frac{T}{\alpha}$$

$$K_g = 2R^2 K_s$$

$$K_g = 2(0.0254)^2 (1.9089e^3)$$

$$K_g = 2.4631 N - m/rad$$

- **Sensibilidad de giro**

La sensibilidad del giro con respecto al eje (α), será:

$$G_g = \frac{2.4631}{0.4718} \left[\frac{°/S}{°} \right]$$

$$G_g = 5.2205 \left[\frac{°/S}{°} \right]$$

Y la sensibilidad con respecto al potenciómetro (β) está dada por:

$$G = \frac{G_g}{4} \left[\frac{°/S}{°} \right]$$

$$G = 1.3051 \left[\frac{°/S}{°} \right]$$

- **Diseño del sistema de control**

Suponiendo que la placa del soporte (y el servo) giran con respecto a la base, el ángulo γ y el módulo de giro, giran con relación al módulo servo con el ángulo θ . El giro total del módulo del giróscopo con relación a la base está dado por:

$$\xi = \gamma + \theta$$

Queremos diseñar un controlador que mantenga ξ en cero grados independientemente de γ y que solo podemos utilizar la medida de β del sensor de giro.

Derivando la ecuación anterior y reescribiéndola tenemos que:

Suponiendo que la placa del soporte (y el servo) giran con respecto a la base, el ángulo γ y el modulo de giro giran con relación al modulo servo con el ángulo θ . El giro total del modulo del giróscopo con relación a la base está dado por:

$$\dot{\xi} = \dot{\gamma} + \dot{\theta}$$

$$\dot{\xi} = \omega_b$$

$$\omega_b = G_g \frac{\beta}{4}$$

Resultando:

$$\beta = \frac{4}{G_g} (\dot{\gamma} + \dot{\theta})$$

$$\beta = \frac{4}{G_g} (s\gamma + s\theta)$$

Por lo cual se requiere controlar que la integral de β sea cero.

Las ecuaciones del servo motor (SRV-02) son:

$$V_m = I_m R_m + K_b K_g \dot{\theta} = I_m R_m + sK_b K_g \theta$$

$$T_m = I_m K_m K_g = J_t \ddot{\theta} = J_t s^2 \theta$$

Resolviendo para θ se tiene que:

$$\theta = \frac{V_m}{\frac{R_m J_t}{K_m K_g} s^2 - K_m K_g s}$$

Substituyendo en la ecuación de β se tiene que:

$$\beta = \frac{4}{G_g} \left(s^2 \gamma + \frac{V_m s}{\frac{R_m J_t}{K_m K_g} s^2 - K_m K_g s} \right)$$

$$\beta = \frac{4}{G_g} s^2 \gamma + \frac{\frac{4}{G_g} V_m s}{\frac{R_m J_t}{K_m K_g} s^2 - K_m K_g s}$$

Definiendo una nueva variable:

$$\varepsilon = \frac{\beta}{s}$$

Se obtiene:

$$\varepsilon = \frac{4}{G_g} s \gamma + \frac{\frac{4}{G_g} V_m}{\frac{R_m J_t}{K_m K_g} s^2 - K_m K_g s}$$

Diseñando un controlador para que ε se mantenga en cero, se define al controlador, como:

$$V_m = k_p \varepsilon + k_d s \varepsilon$$

Obteniendo la función de transferencia:

$$\frac{\varepsilon}{\gamma} = \frac{n_1 s^2 + n_0 s}{s_2 s^2 + s_1 s + s_0}$$

Donde:

$$n_1 = 4J_t R_m$$

$$n_0 = -4K_m^2 K_g^2$$

$$s_2 = G_g J_t R_m$$

$$s_1 = -G_g K_m^2 K_g^2 - 4K_m K_g K_d$$

$$s_0 = -4K_m K_g K_d$$

Nótese que el espacio de estados, la ganancia de la función de transferencia es cero, para un escalón en γ , ϵ , tiende a cero, esto significa que $(\gamma + \theta)$ tiende a cero. Esto quiere decir que θ gira en la dirección de $(-\gamma)$, que es exactamente el efecto deseado.

Ahora seleccionando un factor de relación (Z) de 0.707 y una frecuencia natural de 5Hz (es decir se limitan las perturbaciones a 5Hz) se obtiene la función de transferencia deseada:

$$s^2 + 2\omega_0 s + \omega_0^2$$

Ahora resolviendo para las ganancias de retroalimentación tenemos que:

$$K_p = -1/4 \frac{\omega_0^2 G_g J_t R_m}{K_m K_g}$$

$$K_p = -62.37 \text{ VI/rad}$$

$$K_d = -1/4 G_g \frac{(2z\omega_0 J_t R_m + K_m^2 K_g^2)}{K_m K_g}$$

$$K_d = 2.9159 \text{ VI(rad/sec)}$$

Convirtiendo a grados tenemos que:

$$K_p = -1.0 \text{ VI}^\circ$$

$$K_d = -0.5VI(^{\circ}/sec)$$

El diagrama de control se muestra a continuación:

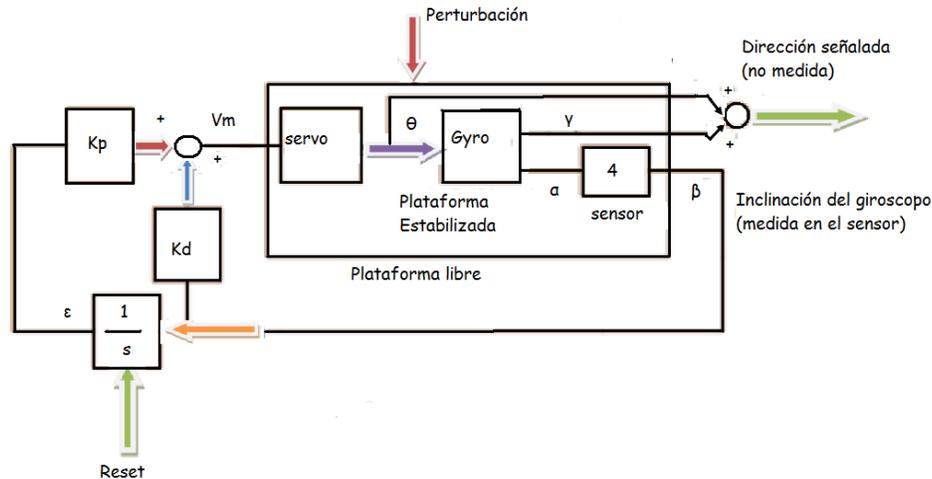


Figura12: Diagrama de control.

La perturbación del giro representa una perturbación externa aplicada al soporte de la base (por ejemplo nosotros) así el servo motor gira alrededor de un punto pivote en la base de la plataforma en un ángulo γ . Esta rotación ocasiona la inclinación de α en el eje sensible proporcional a la velocidad cambio de γ . la inclinación α es medida por el sensor que tiene una ganancia de 4 (relación de cambio) y el control de retroalimentación es el ángulo β .

β es integrada para obtener a ϵ . Nótese que ϵ no es derivada para obtener el termino $(k_d s \epsilon)$ en el lazo de retroalimentación. En lugar de esto β , es usada directamente esto, no necesariamente introducirá ruido. El voltaje V_m aplicado al servo resulta en un giro del modulo del giróscopo con relación al servo alrededor de θ que es compensado por γ y regresa a ξ a cero.

4. Programación de la solución en LABVIEW

4.1 Controlador PD.

Un controlador que funciona típicamente como un amplificador con una ganancia constante k se conoce formalmente como control proporcional, ya que la señal de control a la salida del controlador está relacionada con la entrada del controlador mediante una constante proporcional. El controlador de tipo Proporcional derivativo presenta la siguiente función de transferencia:

$$G(s) = K_p + K_D s$$

4.11 Métodos de sintonización

Los controladores actuales son dispositivos de aplicación general, cuya adaptación a un lazo en particular se realiza mediante la sintonización. El caso más general, es el controlador que actúa con las tres acciones: proporcional, integral y derivativa.

La sintonización consiste en asignar valores a las constantes K_c , T_i y T_d de forma que el lazo de control responda adecuadamente a las perturbaciones externas. Un controlador se puede considerar sintonizado, cuando un cambio en el punto de consigna o en las condiciones del sistema de lazo cerrado, produce como máximo 3 o 4 oscilaciones de la variable alrededor del punto de consigna.

Para la sintonización de los controladores existen diversos procedimientos, pero la sintonización finalmente se realiza de forma experimental, siendo muy importante la experiencia del operador que lo lleva a cabo. Una forma alternativa es mediante la técnica de contornos de las raíces, que es la misma del Lugar Geométrico de las Raíces pero con variación múltiple de parámetros.

Por otra parte, existen algunos procedimientos estándar, que están basados en criterios para minimizar la función de error de varias maneras. Es necesario mencionar que los resultados que se obtienen al aplicar estos procedimientos, sólo dan una indicación de los parámetros iniciales (base) y a partir de éstos, es necesario hacer un reajuste de los mismos para obtener la respuesta de lazo cerrado deseada.

Por el momento solo se presentan dos de los métodos más empleados: oscilaciones amortiguadas (Harriot 1957) y la curva de reacción (Ziegler-Nichols 1942)

4.12. Método de oscilaciones amortiguadas (Harriot 1957)

Consiste en operar el sistema en lazo cerrado con una acción proporcional, empleando la configuración de la figura

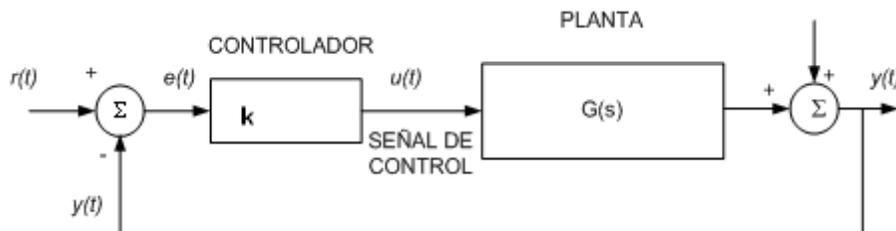


Figura13: Lazo cerrado.

Donde:

$r(t)$ es la señal de referencia, punto de ajuste (Set point) o punto de consigna.

$e(t)$ es la señal de error

$u(t)$ es la señal de salida del controlador o señal de control

$d(t)$ es la señal de perturbación

$y(t)$ es la señal de salida (variable controlada)

Aplicando en cada prueba una variación de referencia de tipo escalón, la ganancia del controlador proporcional se ajusta hasta que la respuesta del sistema realimentado tiene la forma de la figura

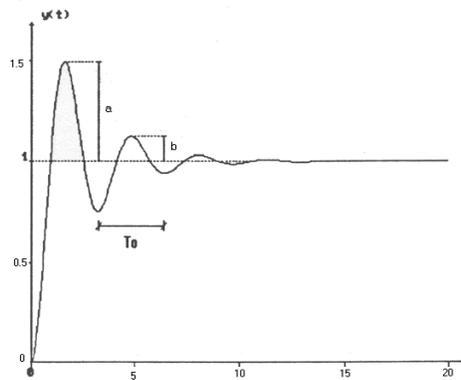


Figura14: Respuesta del sistema realimentado.

La relación entre el primer y segundo sobrepasos debe ser igual a 0.25, esto es:

$$\frac{b}{a} = 0.25$$

Esta condición se logra cuando el factor de amortiguamiento relativo de los polos dominantes es de 0.2176. En este caso, el proceso queda caracterizado en su dinámica, con los parámetros:

K_0 : es el valor de K_c para el cual la respuesta escalón del sistema realimentado presenta una relación entre el segundo y primer sobrepasos de 0.25.

T_0 : es el período de oscilación de la respuesta a escalón del sistema realimentado para las condiciones especificadas.

Las fórmulas de sintonización en este método son:

Tabla 1. Fórmulas de Sintonización

Tipo de controlador	K_c	T_i	T_d
P	K_o	-	-
PI	K_o	T_o	-
PID	K_o	$\frac{T_o}{1.5}$	$\frac{T_o}{6}$

4.13 Método de la curva de reacción (Ziegler-Nichols 1942)

En este método de sintonización se determinan las características dinámicas del proceso a controlar, a partir de la respuesta del sistema en lazo abierto. Una vez que se tienen estas características es posible obtener los parámetros del controlador. La caracterización del proceso es como sigue: la planta en lazo abierto es excitada con una función escalón y se obtiene su respuesta, teniéndose en este caso una curva como se muestra en la figura.

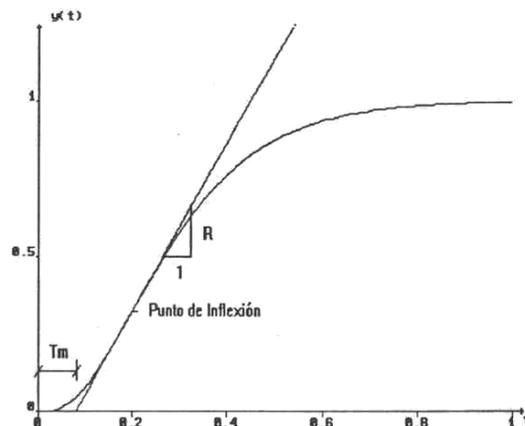


Figura 15: Respuesta escalón.

En esta curva se determina el punto de inflexión y sobre él se traza una recta tangente a la curva. Esta recta intersecta al eje de tiempo y el tiempo muerto que caracteriza al proceso se mide desde el origen hasta este punto de intersección, como se muestra en la figura. El otro parámetro que se requiere para la sintonización es la pendiente de la recta tangente en la respuesta graficada.

Las formulas de sintonización están en la siguiente tabla:

Tabla 2. Parámetros de Sintonización.

Tipo de controlador	K_c	T_i	T_d
P	$\frac{1}{RT_m}$	-	-
PI	$\frac{0.9}{RT_m}$	$\frac{T_m}{0.2}$	-
PID	$\frac{1.2}{RT_m}$	$\frac{T_m}{0.3}$	$0.8T_m$

T_m : es el periodo de oscilación de la respuesta a escalón del sistema realimentado para las condiciones especificadas.

4.2 PROGRAMACIÓN DEL CONTROLADOR (PD) EN LABVIEWRUN-TIME 7.1

La rutina de control para la ejecución en tiempo real se programo en esta versión de LabVIEW, debido a que los controladores de la tarjeta de tiempo real PCI7041/6040E solo se emitieron hasta la versión de LabVIEW-RUN-TIME7.1 debido a que el modelo fue descontinuado por el fabricante National Instruments.

A continuación se muestra el código realizado para llevar a cabo el proceso de control antes mencionado:

4.21 Ciclo While con temporizador

El programa principal esta contenido en un ciclo while, en el cual es configurado el tiempo en que se desea trabajar, como se muestra en la siguiente figura:

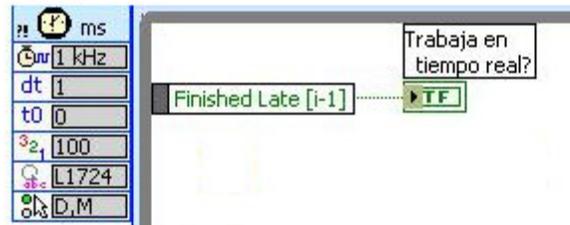
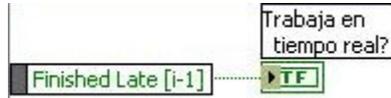


Figura 17. Configuración del ciclo while.

Las configuraciones importantes por realizar son:

1 kHz: Configura el reloj de la tarjeta.

1: Configura el tiempo de ejecución(en este caso 1ms)



: Esta parte del programa verifica que el mismo este ejecutandose en el tiempo de ejecución dado, en este caso es de 1ms, de no ser así una luz roja se encendera en el panel frontal del programa.

4.22 Proceso de adquisición de lectura del encoder(Digital)



Figura 18. Proceso de adquisición de datos del encoder

En esta parte el puerto digital es configurado para obtener la lectura de los pulsos del encoder del giróscopo y se realiza la creación del canal virtual.

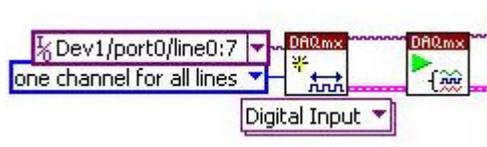


Figura 19. Creación del canal virtual

Al mismo tiempo se lleva a cabo la lectura de dichos pulsos, a través del puerto digital de la tarjeta BNC 2120 de National Instruments. También se separan los canales A y B en este caso en un arreglo donde usamos el canal 0 como canal A y el canal 1 como canal B.

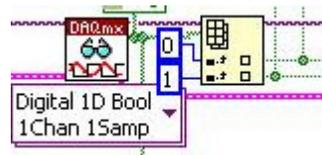


Figura 20. Lectura del canal digital

Se realiza un monitoreo visual de cómo se comportan los canales digitales en el panel frontal a través de un indicador booleano.

A la salida del ciclo detenemos el proceso y limpiamos el canal virtual creado.

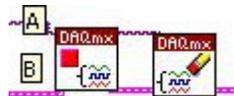


Figura 21. Paro de lecturas y limpieza de canal virtual

Después de haber obtenido los pulsos de los canales A y B procedemos a realizar la conversión de dichos pulsos a unidades de grados.

4.23 Rutina de conversión de lectura del Encoder a posición angular

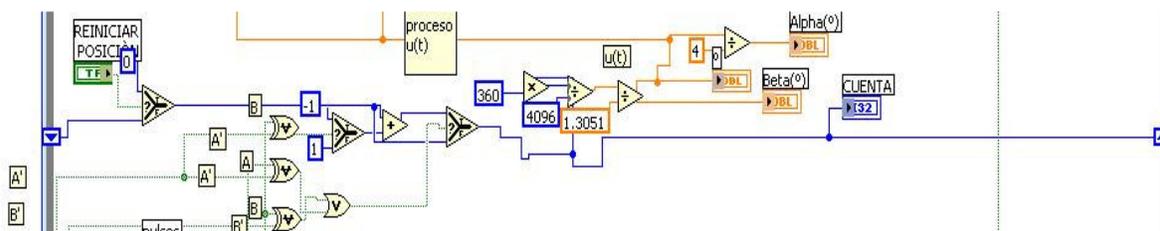


Figura 22. Rutina de conversión de pulsos a cuentas y grados.

Se lleva a cabo la retroalimentación de las lecturas a través del registro de corrimiento. La cual nos permite leer el estado actual del sistema.



Figura 23: Registro de corrimiento.

En esta sección se lleva a cabo la detección del número de cuentas (pulsos) y el sentido en el que está girando el módulo.

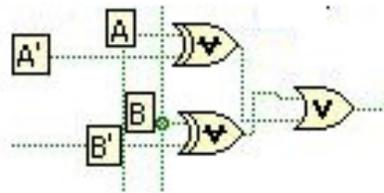


Figura 25. Lectura del estado actual y estado siguiente de los canales A y B

En esta sección del programa, se discriminan las lecturas a solo el paso de una señal con las compuertas XOR programadas, ya sea la señal actual del canal A o B (A' o B') o la del estado siguiente (A o B), para asegurar que la cuenta (pulso) no tenga error de traslape, entre el estado actual y el estado siguiente (se previene que A y A' o B y B' no ocurran al mismo tiempo) finalmente una compuerta OR es la que nos entrega el valor alto, para poder proseguir a realizar la cuenta de pulsos. (El estado Actual lo obtenemos a través de la retroalimentación)

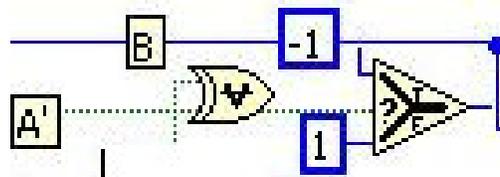


Figura 26. Determinación del sentido de giro.

En esta sección se lleva a cabo la detección del sentido en el que gira el modulo, se obtiene a través de una suma o resta de cuentas (pulsos) los cuales nos entregaran ya se a un signo positivo o negativo según sea su caso, esto quiere decir que en caso de que el estado siguiente del canal B y el estado actual del canal A (A') ocurran al mismo tiempo tendremos una cuenta con signo negativo, esto quiere decir que el canal A va en dirección del B, esto indica un cambio en el sentido del giro. Mientras esto no ocurra se tiene que el valor de la cuenta será positivo, lo cual indica un giro en sentido positivo.

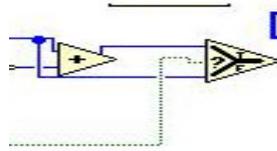


Figura 26. Generación de las cuentas.

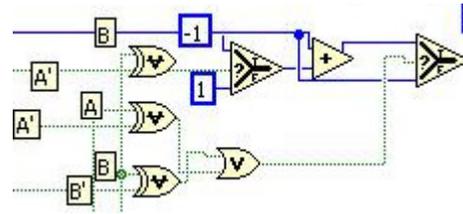


Figura 27. Detección de cuentas.

En esta parte del programa se lleva a cabo la generación de cuentas, utilizando una retroalimentación, es decir al estado actual de la cuenta, se le sumara o restara una cuenta, según sea el caso, de acuerdo con el resultado obtenido en la parte anterior del proceso, es decir la determinación del signo de la cuenta, esta sección funciona como un contador (con incremento o decremento de una unidad), a estado actual se le suma el estado siguiente.

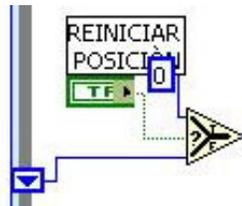


Figura 28. Reinicio de cuentas

El programa consta de un reinicio en la posición, es decir, se puede reiniciar el numero de cuentas que se han producido hasta el momento, mandando un valor de cero al valor de las cuentas.

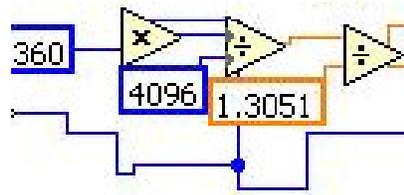


Figura 29 Conversión de cuentas a grados.

En esta sección, se lleva cabo una conversión de cuentas a grados, la cual nos permite, llevar a cabo el proceso de control, en la primer parte el numero de cuentas se multiplica por 360, que equivale a los grados que lleva una vuelta, y se dividen entre los 4096 cuentas que equivalen a tener una vuelta ($4096[\text{cuentas}] = 1[\text{RPM}]$). En la siguiente partes toma en cuenta la sensibilidad del aparato, es decir el valor de lectura que tenemos no es absoluto debido a la sensibilidad del giro,

por tanto se divide entre 1.3051 (sensibilidad de β), que es la sensibilidad del encoder para corregir la medición de los grados.

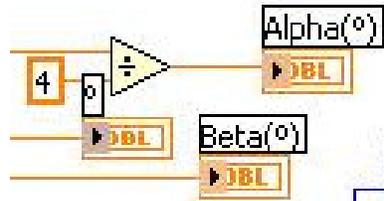


Figura 30. Despliegue de datos

En esta sección se lleva a cabo el despliegue de datos leídos en la pantalla frontal, la primer lectura, se obtiene directamente del encoder inferior (Encoder del srv-02) es la de beta (β), a partir de esta medición se puede conocer el valor de la variable alpha (α) gracias a la relación ya antes mencionada.

4.24 RUTINA DE CONTROL (PD)

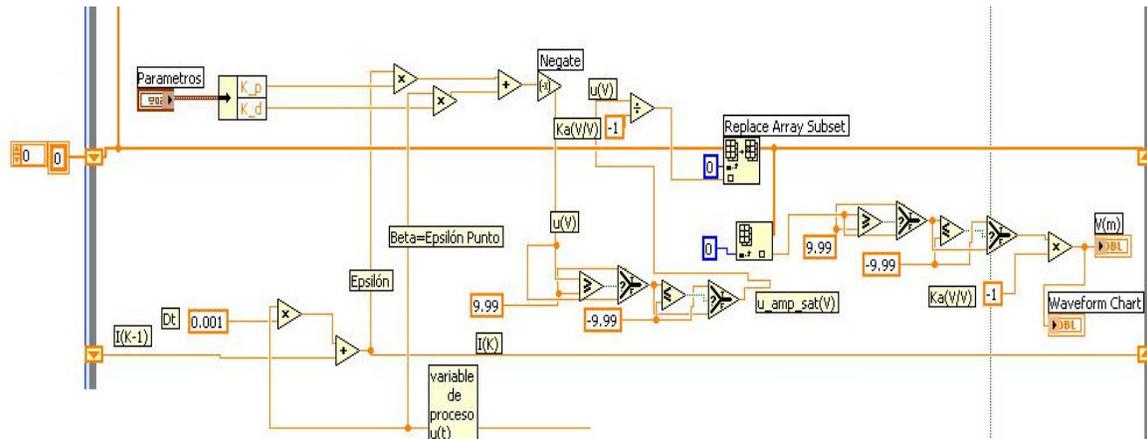


Figura 31. Control PD

El proceso de control es realizado con respecto a la variable β .

Para este tipo de sistemas, se tiene una variación considerable en la variable de proceso, es por eso que en la retroalimentación se lleva a cabo la integración de la misma, lo cual nos permite tener un mejor desempeño de control en este tipo de situaciones.

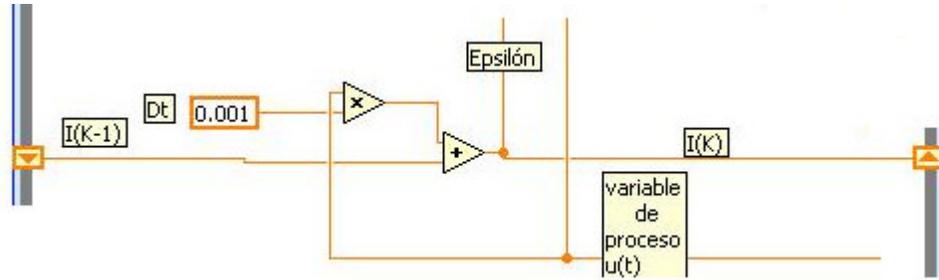


Figura 32. Integrador. Obtención de épsilon (ϵ)

En esta parte se lleva a cabo el proceso de integración de la variable de proceso, multiplicando por la diferencial de tiempo, que en este caso corresponde al valor de 1ms(0.001s), y se lleva a cabo la retroalimentación de la misma, a la salida de este proceso obtenemos una nueva variable llamada épsilon(ϵ).

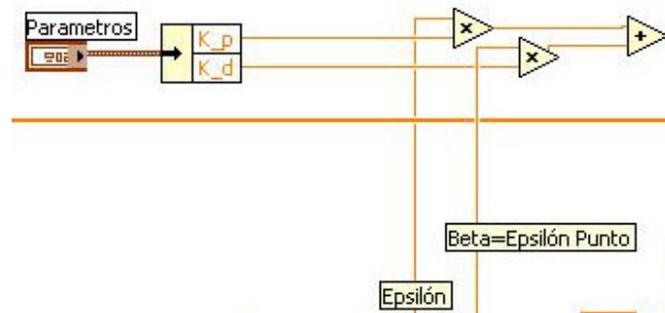


Figura 33. Control PD

La variable leída directamente del encoder se le nombrara ahora épsilon punto, para poder llevar a cabo la parte proporcional derivativa, es decir, que beta, será la velocidad con la que gira la base, en esta parte del proceso se lleva a cabo el control PD (proporcional derivativo), en este se multiplica el parámetro de proporcionalidad y de derivación., finalmente las dos señales se suman a la salida. Siendo esta nuestra variable de control.

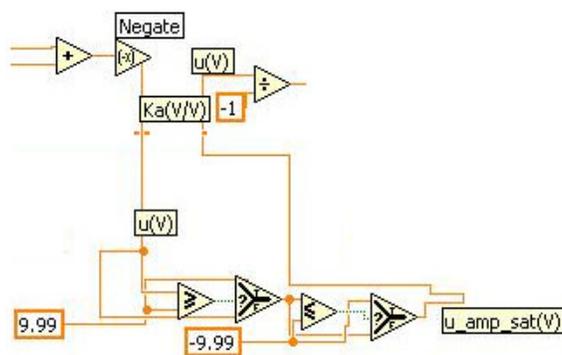


Figura 34. Salida del control.

La salida de control es negada, debido a que el giro de α que está relacionado con β es negativa (-H), después de esto la señal pasa por un saturador, para evitar que la señal de control sobrepase los +10 (V) o los -10(V)(este voltaje es contemplado en el modulo UPM), ya que esto puede dañar el aparato, en esta sección se obtiene la señal de control, saturada, después esta señal contempla la ganancia que tiene el cable con el enviamos la señal de la tarjeta al modulo UPM(valor de -1),



Figura 35. Modulo UPM

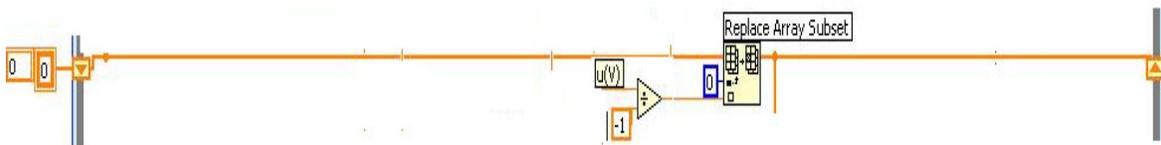


Figura 36. Envió de datos al motor.

Una vez finalizado el proceso de control, se envían los datos a un arreglo, que esta inicializado en cero, después este arreglo comenzara a variar de acuerdo a la variable de control, este proceso se retroalimenta para poder mantener la señal de control, debido a que como anteriormente se había mencionado, este es un sistema con el cual la variable del proceso cambia en cada instante, y no da tiempo para lograr una estabilización del sistema, sin embargo con esta retroalimentación se logra mantener estable la señal de control.

4.25 Rutina para el envió de la señal de control a los motores del SRV02

Inicialización del canal analógico (para enviar los datos al motor)



Figura 37 Creación del canal virtual analógico de salida.

En esta sección se crea el canal de salida anagógico, en la primera parte, determinamos el canal que enviara la señal (Dev/1/a₀0), el voltaje máximo (+9[V]) y el voltaje mínimo (-9 [V]) unidades de salida (Volts[V]) y finalmente el tipo de voltaje de salida (RSE).todo esto en un clúster, el cual después se separa por variable y se conecta a la función de creación de canal virtual.



Figura 38. Envío de datos analógicos en múltiples muestras al motor.

El envío de datos obtenidos del control hacia el motor se lleva a cabo por medio de un canal digital, en la parte de estado(status) se verifica que no exista algún problema durante la escritura en el motor, de ser así envía una señal de paro de programa.



Figura 39. Detención del canal analógico limpiado del mismo.

El motor se detiene una vez que se envía un voltaje de 0[V] al mismo, después se procede a detener el canal analógico ya la limpieza del mismo.

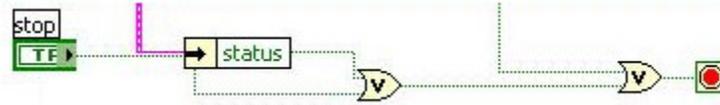


Figura 40. Sección para Detener el Programa.

El paro del programa se realiza mediante el botón de paro en el menú, o debido a alguna falla durante la lectura y escritura de datos.

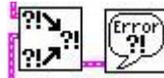


Figura 41. Salida de error

Estas funciones sirven para desplegar, si es que existiese algún error por el cual se halla detenido el programa.

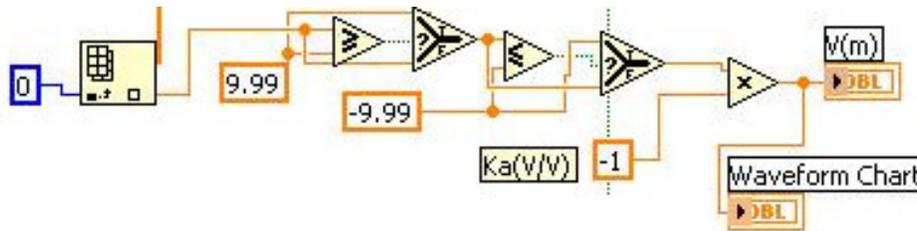


Figura 42. Despliegue de datos en el panel frontal (V (m))

En esta sección se analiza el comportamiento de la variable de control, que en este caso es V (m), y se hace un despliegue en un indicador numérico y uno gráfico.

4.26 Rutina del proceso de adquisición del Tacómetro

Inicialización del canal analógico (para enviar los datos al motor).

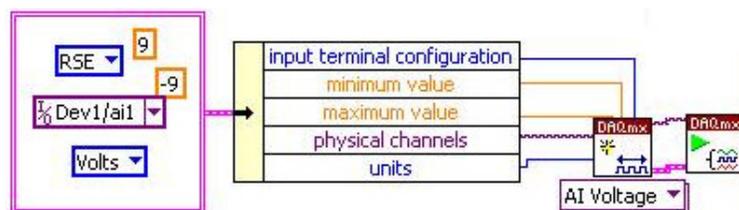


Figura 43. Creación de un canal virtual analógico de entrada.

Esta sección corresponde a los parámetros de configuración de entrada, tales como valor máximo de entrada como valor mínimo (+9V para este caso), también se configura el tipo de unidades que se manejan (en este caso es voltaje), también se configura el tipo de voltaje que se maneja (en este caso es con referencia a una tierra), también se define el canal de entrada de la tarjeta que se tiene (en este caso el canal 1). Lo cual se incluye en un clúster, el cual se separará por nombre de cada una de las variables .y el cual se conectara a las entradas del canal virtual creado y configurado como entrada.

Una vez terminado el proceso de creación del canal virtual, se procede a obtener las lecturas de la tarjeta, lo cual se lleva a cabo con la función de lectura la cual se configura como un canal de entrada y una muestra.

De aquí se obtiene un voltaje, el cual se compara con el voltaje que se envía al motor, para verificar que el control realizado, funciona adecuadamente.

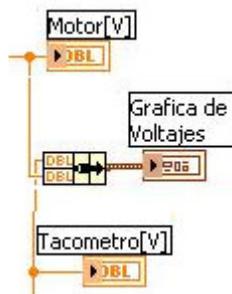


Figura 44. Graficas de voltajes.

Una vez terminada la lectura, se utilizan los datos para llevar a cabo la interpolación de la lectura de voltaje obtenida en el canal 1, proveniente del tacómetro del srv02 y transformarla a velocidad angular, a través del modelo matemático $y = 46.75x - 0.556$ el cual se obtuvo a través de una serie de muestras.

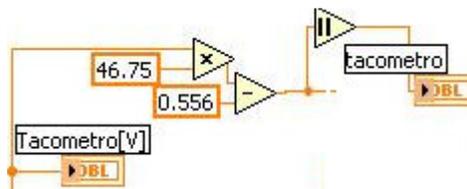


Figura 45: Transformación de voltaje a velocidad

 ω (es el valor absoluto del valor de giro).

 Despliega la velocidad de giro de la base en el panel frontal.

Al final de este programa obtenemos la velocidad angular a la que la base del gir6scopo se encuentra girando, esto es muy importante para nosotros, ya que por medio del tac6metro podemos calcular la posici6n final que posee el gir6scopo, es decir la posici6n de α , de la siguiente manera:

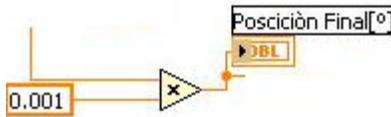


Figura 46. Obtenci6n de la posici6n angular final de α a trav6s del Tac6metro.

Dicho c6lculo se lleva a cabo de la multiplicaci6n de la posici6n angular por el tiempo en el que estamos trabajando, en este caso es 1ms.

Una vez obtenida la posici6n, esta se grafica al mismo tiempo que se grafica la posici6n de alpha, para verificar que realmente se est6 manteniendo en la posici6n deseada.

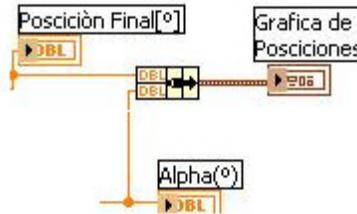


Figura 47: Realizaci6n de las graficas de posici6n final.

(Tac6metro y alpha (obtenida del encoder)).

4.27 Measurement & Automation Explorer

Antes de comenzar a ejecutar el programa es necesario verificar la conexi6n con la tarjeta PCI7041/6040E de National Instruments a trav6s del Measurement & Automation Explorer

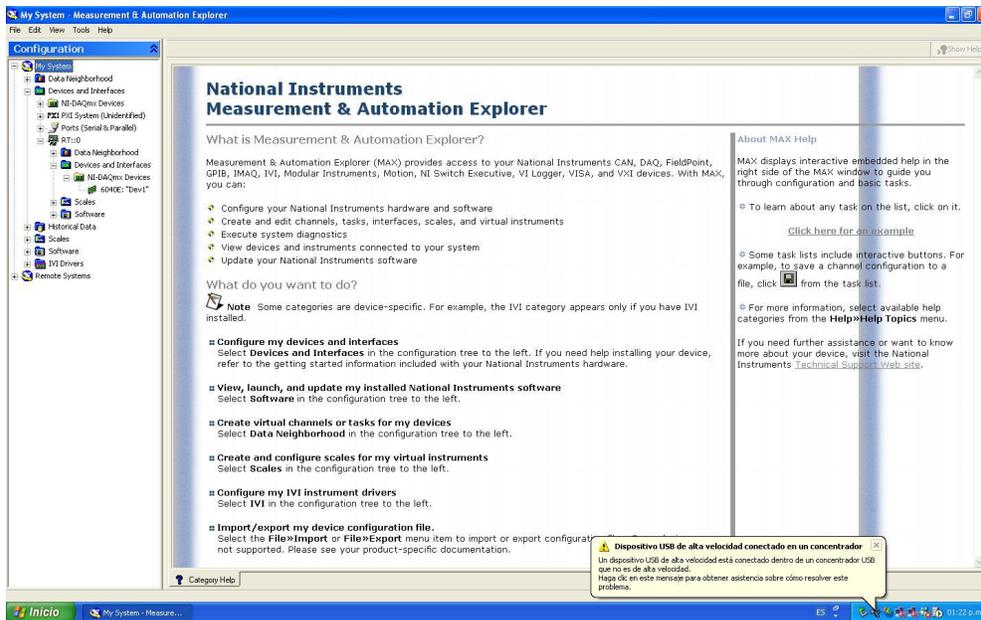


Figura 48. Pantalla inicio de “Measurement & Automation Explorer”

Después se selecciona la parte de NI Devices and Interfaces seguida de la sección de 640E-dev., después presionamos el botón “selftest” y finalizamos la revisión de conectividad.

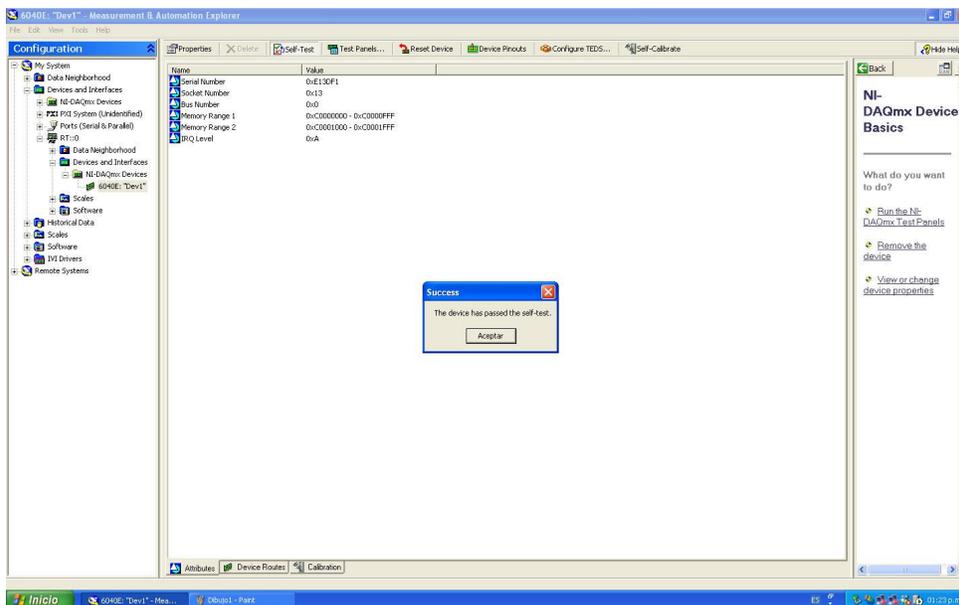


Figura 49. “Measurement & Automation Explorer”

4.28 Descripción del panel Frontal.

Después de esto se procede a comenzar con la descarga del programa a la tarjeta de tiempo real.

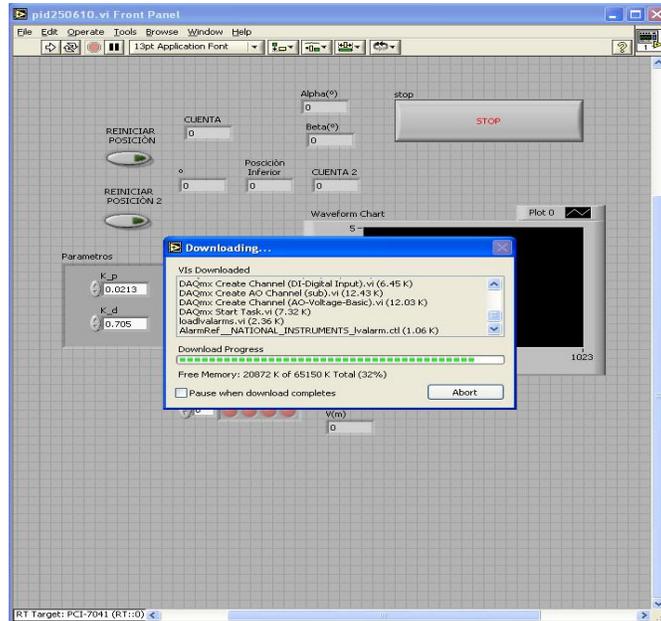


Figura 50. Descarga del programa.



Figura 51. Panel frontal del programa de control. (Programa funcionando)



Figura 52. Parámetros de control.

Controles numéricos para cambiar los parámetros del control K_p y K_d .

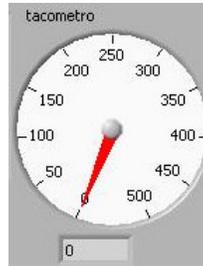


Figura 53. Tacómetro.

Despliegue de la velocidad medida en un indicador analógico y digital.



Figura 54. Alarma de funcionamiento del programa en tiempo real.



Figura 55. Botón para detener la ejecución del programa.



Figura 56. Botón para reiniciar el número de cuentas, utilizado durante el proceso de pruebas para corregir posiciones mal medidas, debidas a diferentes circunstancias.

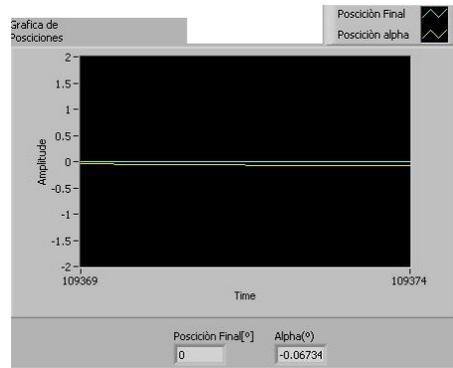


Figura 57. Despliegue de las graficas de posiciones Finales, con sus indicadores digitales en la parte de la gráfica.

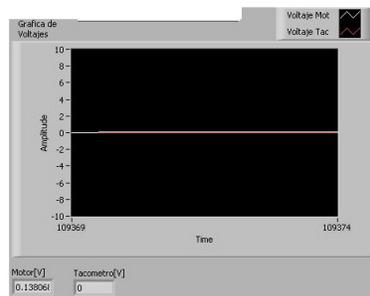


Figura 58. Despliegue de las graficas de voltajes tanto del motor como del tacómetro con indicadores digitales.

4.3 PROGRAMACIÓN DEL CONTROLADOR (PD) EN LABVIEW 8.6

Se ocupa esta versión de LabVIEW, ya que es una plataforma que cuenta con los controladores de la tarjeta PCI6024E, la cual trabaja a una con el tiempo de ejecución de Windows (no tiempo real) se realiza la programación en la versión 8.6 de LabVIEW, la cual se muestra a continuación:



Figura 59. Pantalla de inicio de LabVIEW 8.6

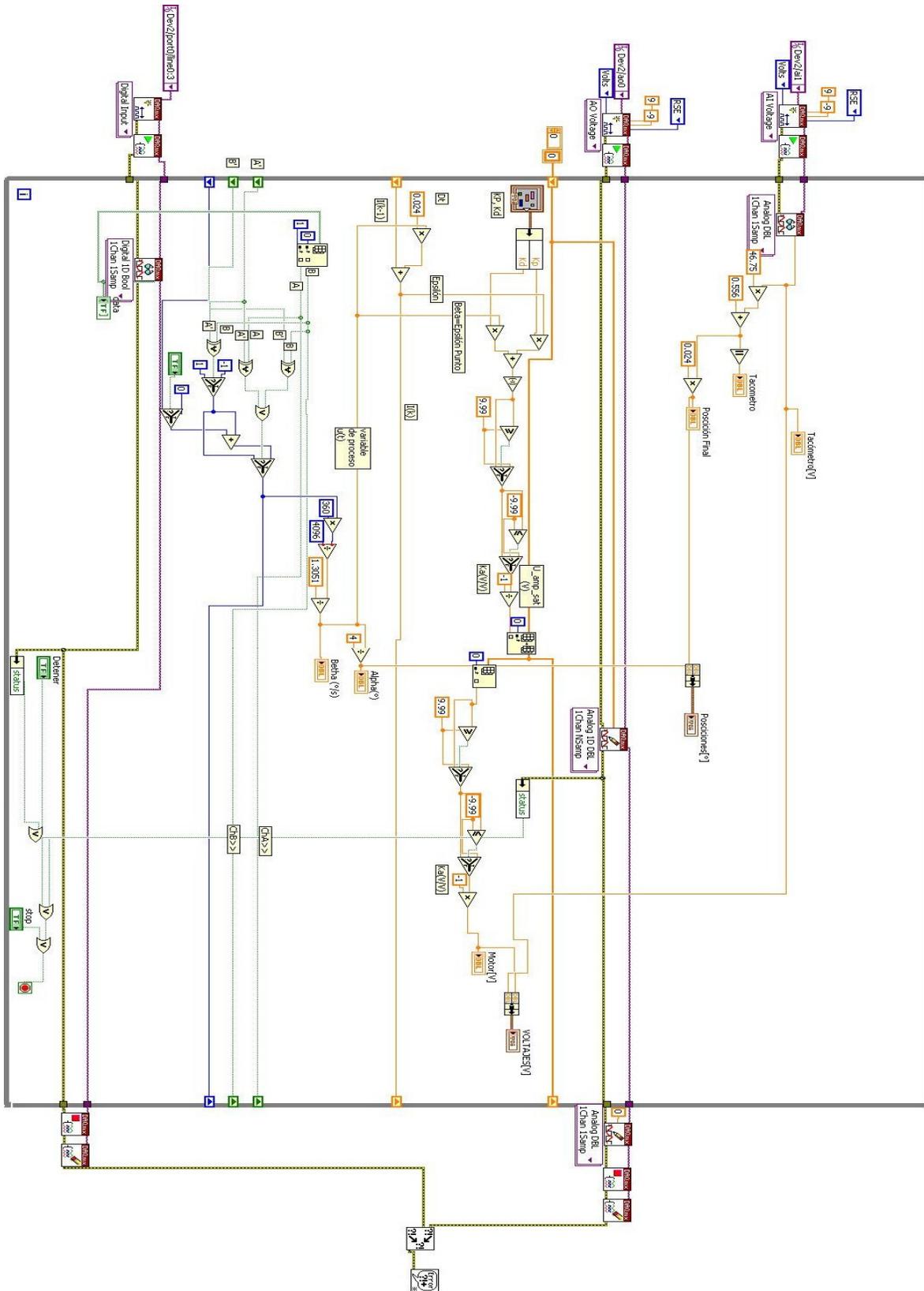


Figura 60. Programa de control PD en LabVIEW 8.6

Se observa que la programación es muy similar a la de labviewrun-time7.1 lo único diferente es ciclo while, ya que en este no se puede tener un tiempo de muestreo deseado, este ciclo while trabaja a un tiempo de 24ms (que es el tiempo de ejecución de Windows).

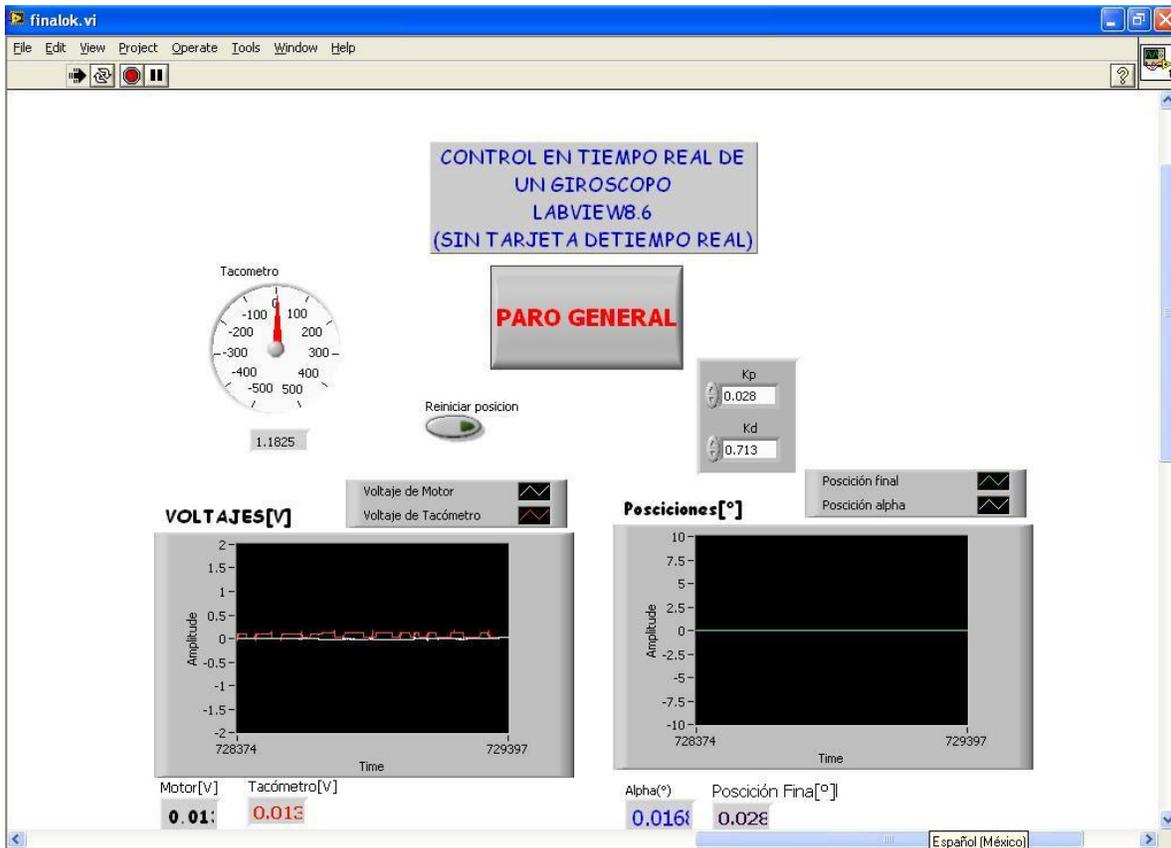


Figura 61. Panel frontal del control PD en LabVIEW 8.6

El panel frontal es similar visto anteriormente en LabVIEWRUNTIME7.1. El proceso para la verificación de la tarjeta PCI6024E se lleva a cabo de manera similar que en labVIEWRUNTIME7.1.

4.4 Desempeño en el Tiempo del Controlador en Tiempo Real

Para medir el desempeño del programa en el tiempo, fue necesario agregar los bloques TraceTool Start Trace VI y TraceTool Stop Trace and Send VI del LabVIEW Execution Trace Tool. Se muestra en la Figura la adición de dichos bloques, a fin de recabar los datos a ser vistos después en la sesión dentro del LabVIEW Trace Tool.

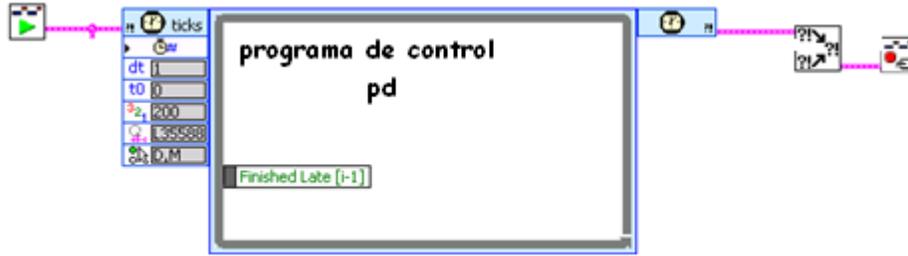


Figura 62. Rutina para la medición de desempeño del programa.

Con la inclusión de las funciones del Execution Trace Toolkit podemos verificar el tiempo de ejecución real de cualquier ciclo dentro del programa, en este caso nuestro proceso de control llamado PD, en la figura verificamos que el ciclo de control tarda en ejecutarse 1 [ms] en promedio.

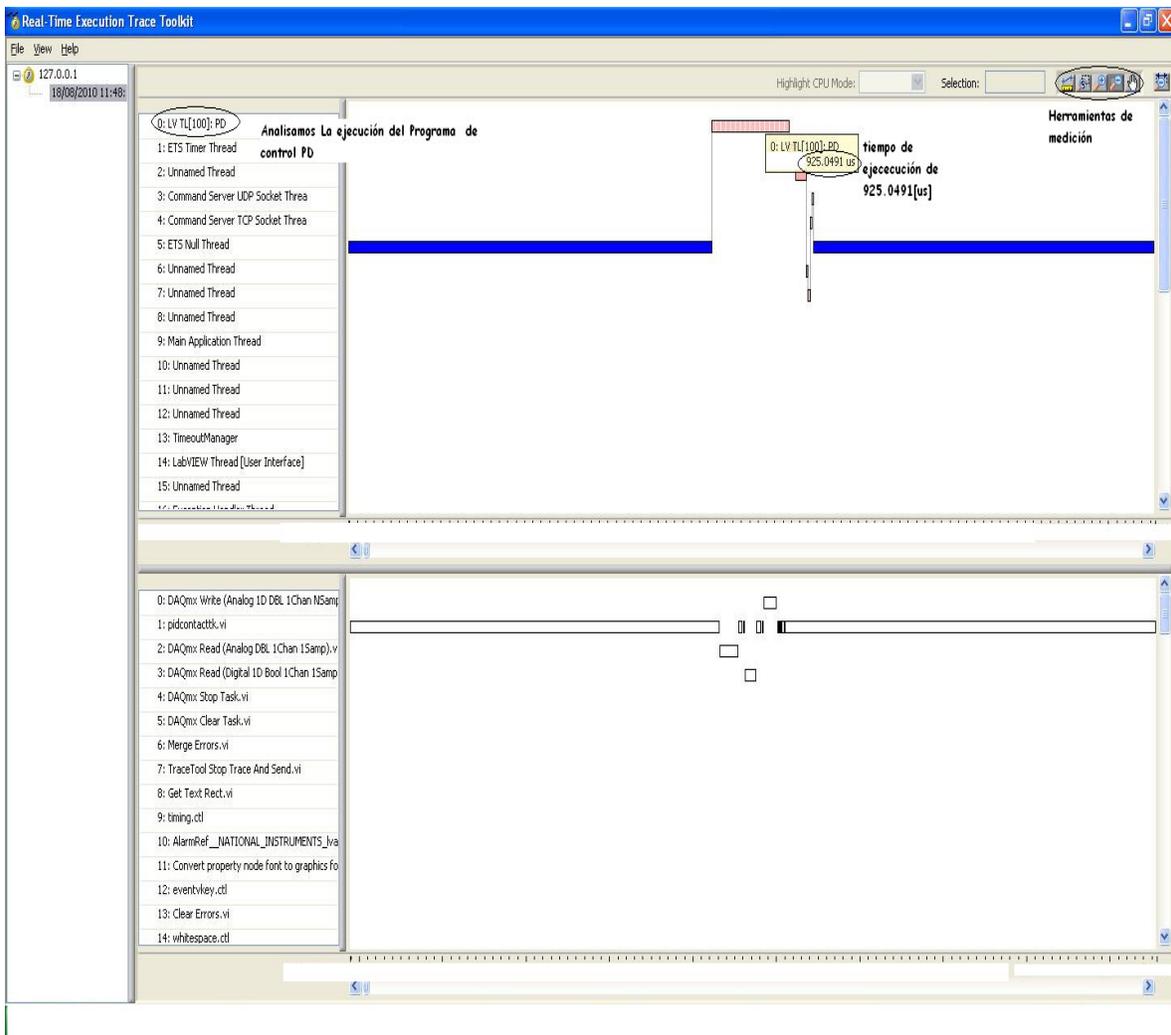


Figura 63. Execution Trace Toolkit para el ciclo de control en la tarjeta NI PCI-6024E.

5.-Aplicación experimental y resultados.

5.1. Análisis del Srv02-Gyro

Para la sintonización del control, utilizaremos el método de oscilaciones sostenidas, para lo cual es necesario encontrar el valor de K_{Os} , que es la ganancia para la cual el sistema presenta oscilaciones sostenidas. Es posible obtener este valor mediante el Lugar Geométrico de las Raíces, o a partir del Criterio de Estabilidad de Routh:

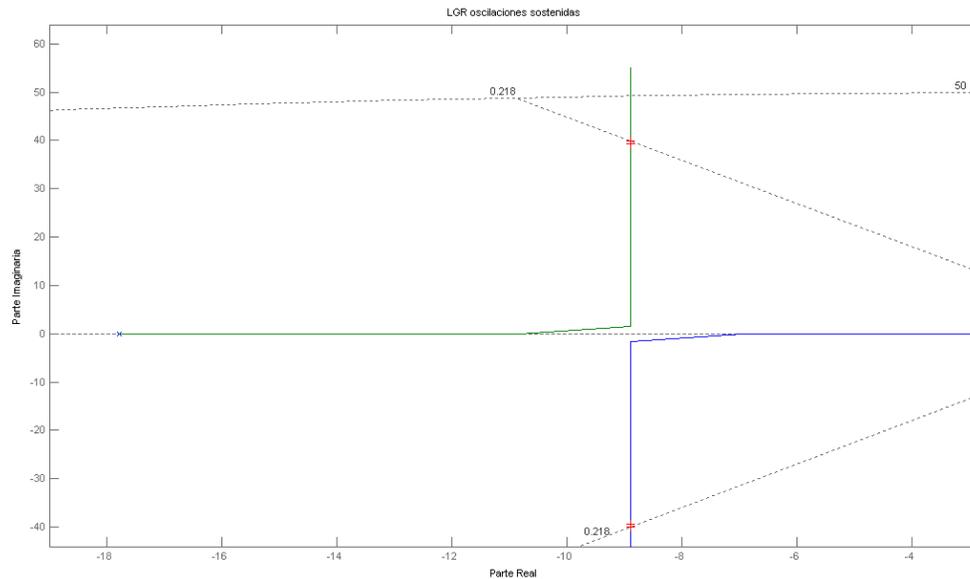


Figura 64. Gráfica del Lugar Geométrico de las Raíces realizado en matlab.

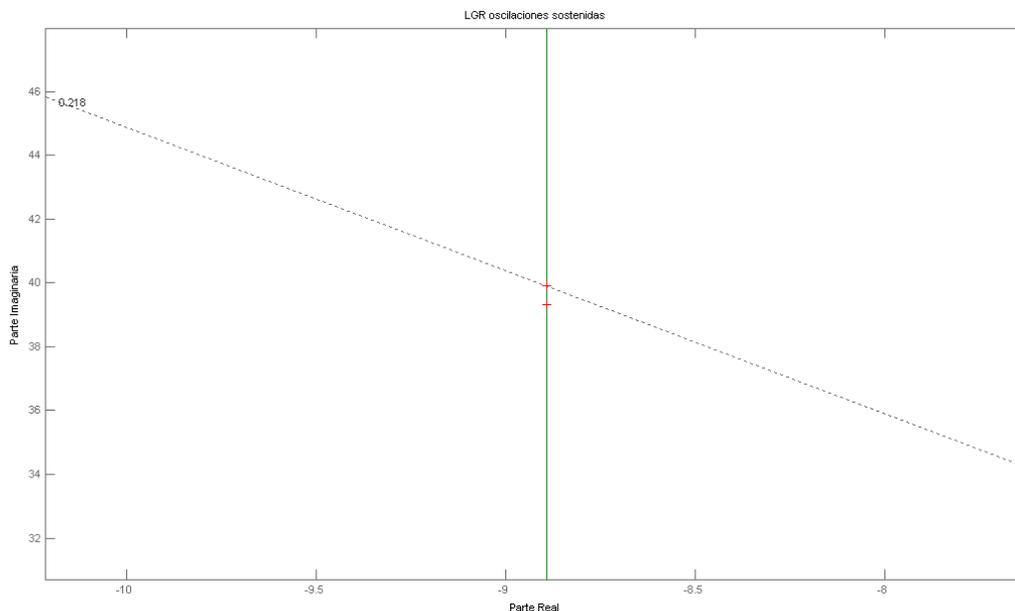


Figura 65. Acercamiento (zoom) de la gráfica del Lugar Geométrico de las Raíces realizado en matlab.

Con la herramienta de MatLab *rlocfind* se determinan los puntos y el valor de la ganancia para dichos puntos, siendo éstos:

$$k_{os} = 53.4148$$

$$s_0 = -8.8993 \pm j39.9122$$

Este valor de $j39.9122$ es la frecuencia de oscilación, por lo tanto:

$$\omega_d = 39.9122 = 2\pi f_{os} = \frac{2\pi}{T_{os}}$$

$$T_{os} = \frac{39.9122}{2\pi} = 6.35222[s]$$

Por lo cual los parámetros de sintonización son:

$$T_{os} = 6.35222[s] \quad y \quad k_{os} = 53.4148$$

La tabla a utilizar es:

Tabla 3: Parámetros de control:

Tipo de Controlador	K_c	T_i	T_d
P	K_o		
PI	K_o	T_o	
PID	K_o	$\frac{T_o}{1.5}$	$\frac{T_o}{6}$

Y los valores finales son:

Tabla 4. Parámetros para el giróscopo.

Tipo de Controlador	K_c	T_i	T_d

P	53.4148	-	-
PI	53.4148	6.35222	-
PID	53.4148	4.23482	1.0587

Realizando la transformación de los parámetros debido a que no es directamente el envío de la salida de control, se tiene que transformar conforme a la siguiente relación (debida a los parámetros del giróscopo):

$$k_p = \frac{k'_p}{1907.6} \quad k_d = \frac{k'_d}{8.909}$$

Ahora transformando los parámetros, se obtiene:

$$k_p = \frac{53.4148}{1907.6} \quad k_d = \frac{6.35222}{8.909}$$

$$K_p=0.028 \quad \text{y} \quad K_d=0.713$$

5.2 PRUEBA EN LABVIEW 8.6

En esta versión, se llevo a cabo la ejecución del programa descrito anterior mente en labVIEWRUNTIME7.1. A continuación se muestran los gráficos obtenidos durante el desempeño del programa de control obtenido a través de la plataforma (LabVIEW). Siendo importante la reproducción de datos obtenidos en LabVIEW en otras plataformas, se creo utilizo la rutina que ofrece este paquete de obtención de archivos de texto“TXT” para posteriormente realizar los respectivos gráficos en la plataforma de matlab.

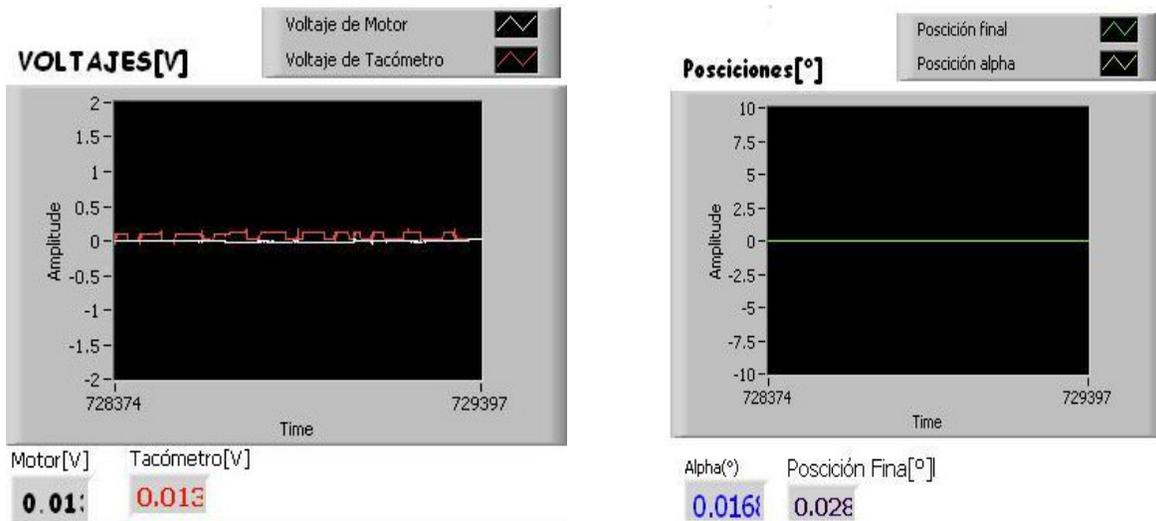


Figura 66. Graficas (voltajes y posiciones) obtenidas con LabVIEW 8.6 al Inicio del programa.

En estas imágenes se muestra el programa para el giróscopo en una posición inicial.

A continuación, se muestra el comportamiento del controlador en cuanto ocurre un desplazamiento en la base del giróscopo:

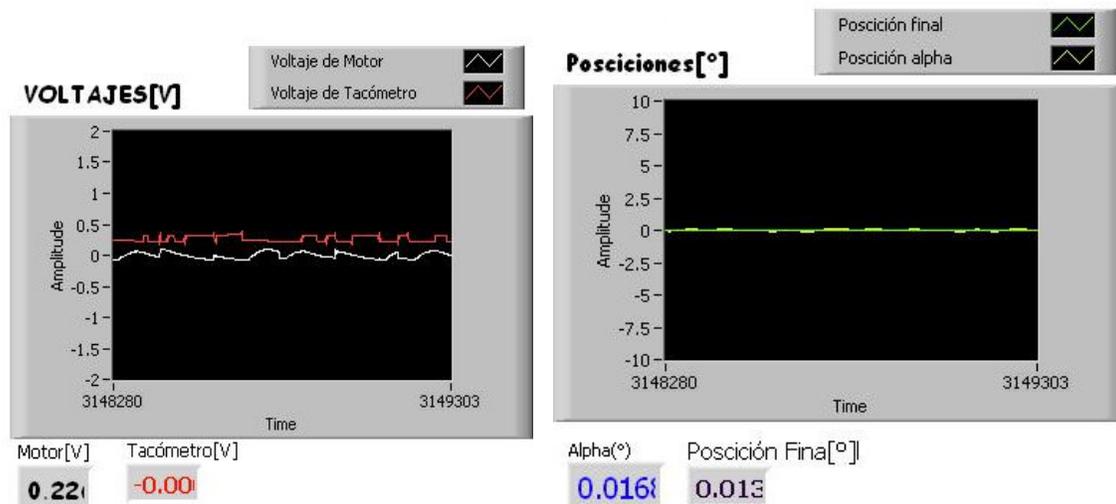


Figura 67. Graficas (voltajes y posiciones) obtenidas con LabVIEW durante la Perturbación en el giróscopo.

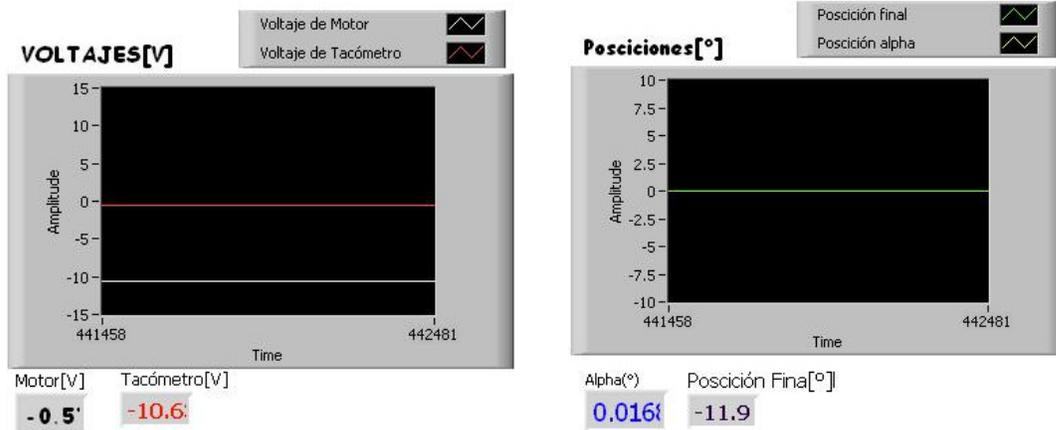


Figura 68. Graficas (voltajes y posiciones) obtenidas con LabVIEW al fin del programa.

Se muestra que el gir6scopo al recibir una perturbaci6n no es capaz de mantener una relaci6n congruente entre los datos de voltaje y los de las posiciones, por lo cual es claro resultado de un mal funcionamiento de la lectura de datos y de la rutina de control.

Corroborando lo en matlab, se observa que las mediciones no equivalen a las obtenidas con el programa de LabVIEW.

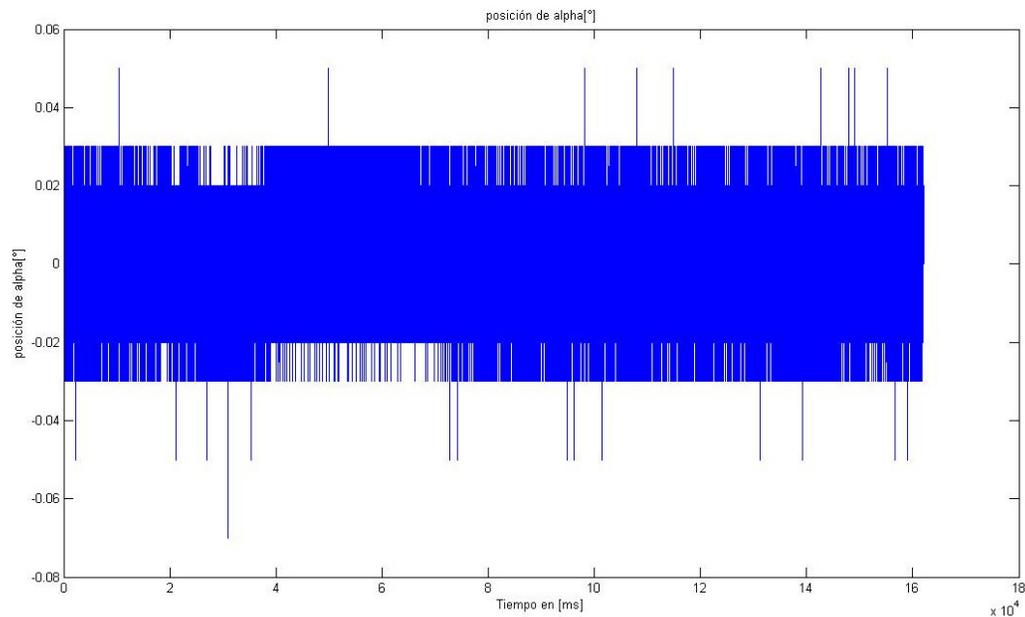


Figura 68. Gr6fica de posici6n del 6ngulo α (correspondiente a la inclinaci6n del disco giratorio) realizada en matlab.

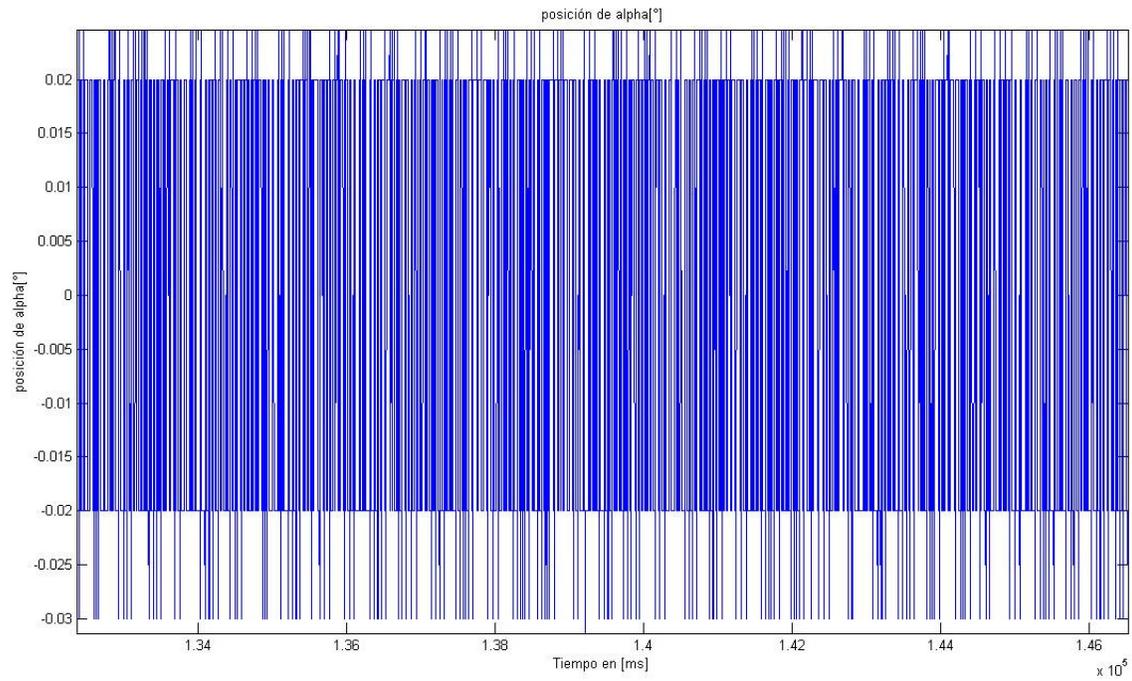


Figura 69. Acercamiento (zoom) de la gráfica de posición del ángulo α (correspondiente a la inclinación disco giratorio) realizada en matlab.

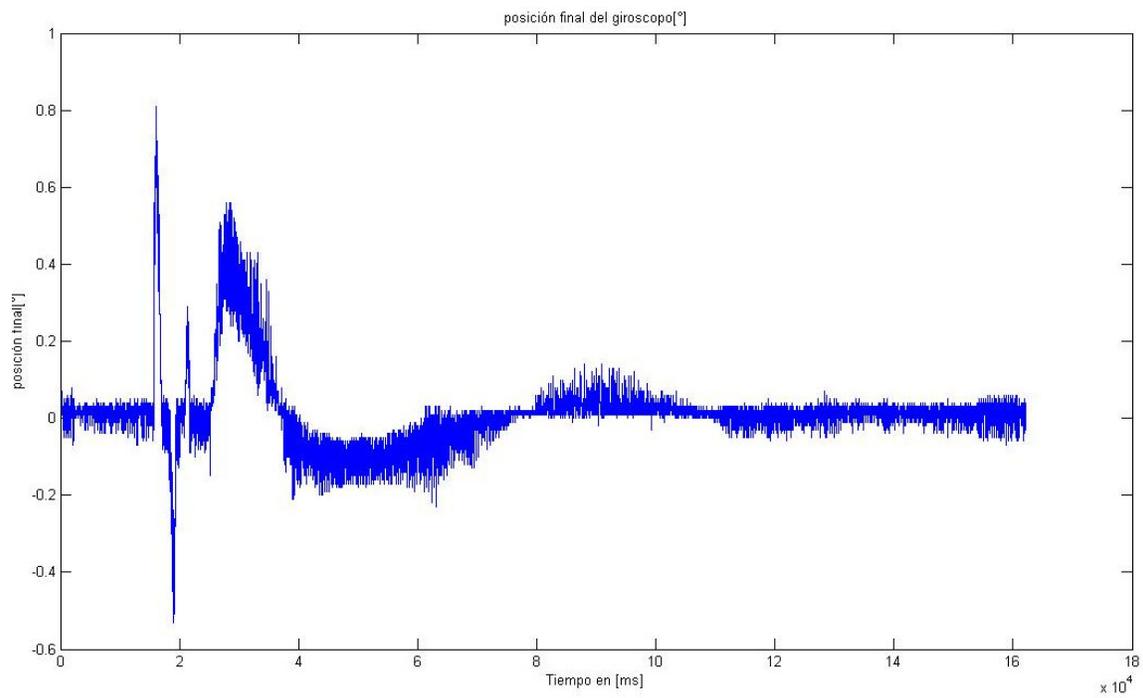


Figura 70. Gráfica de la posición final del giróscopo (correspondiente a la inclinación disco giratorio)

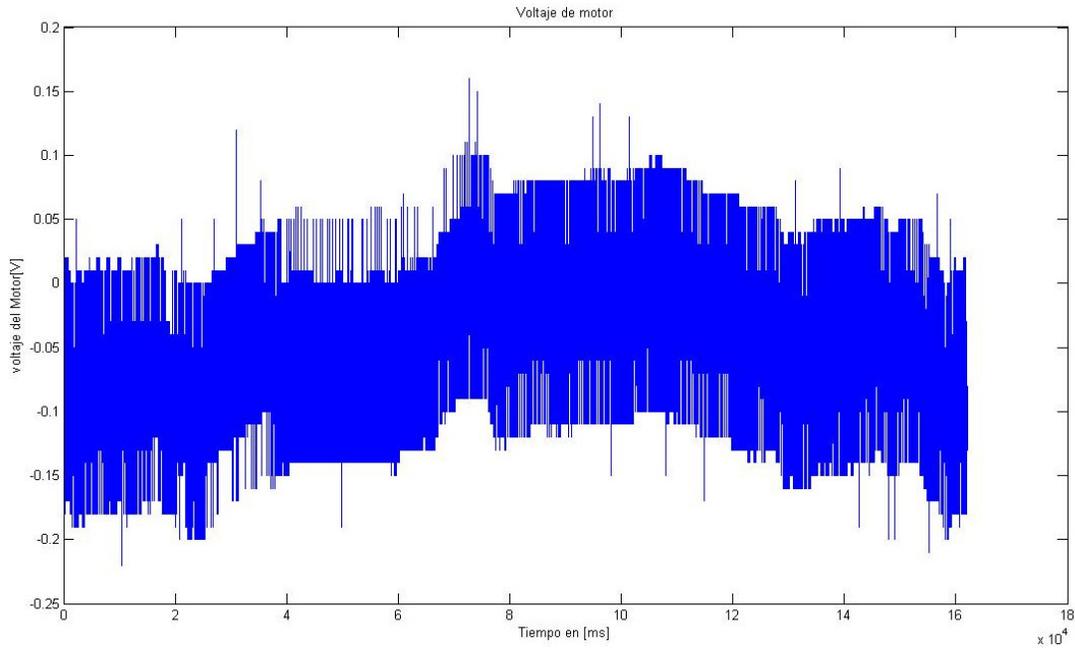


Figura 71. Gráfica del voltaje suministrado al motor

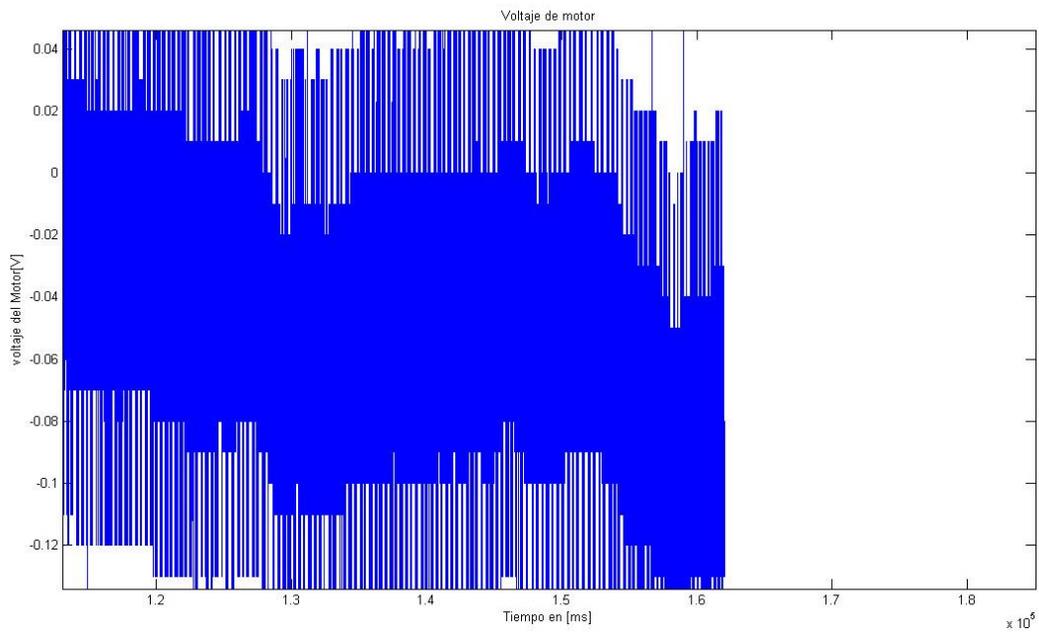


Figura 72. Acercamiento (zoom) de la gráfica del voltaje suministrado al motor

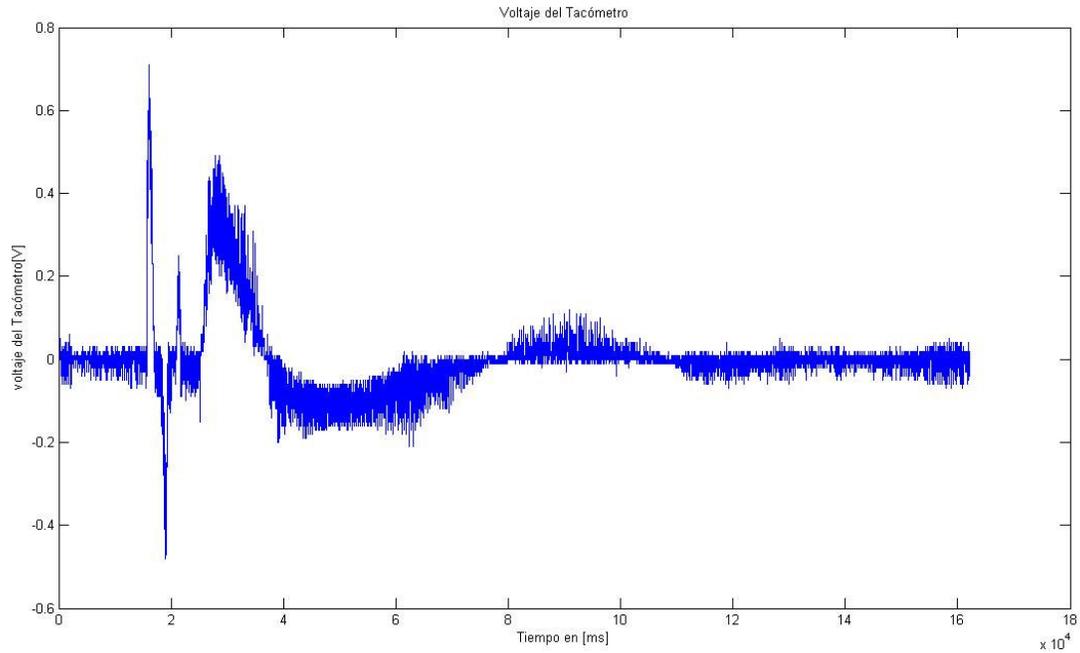


Figura 73. Gráfica del voltaje emitido por el tacómetro.

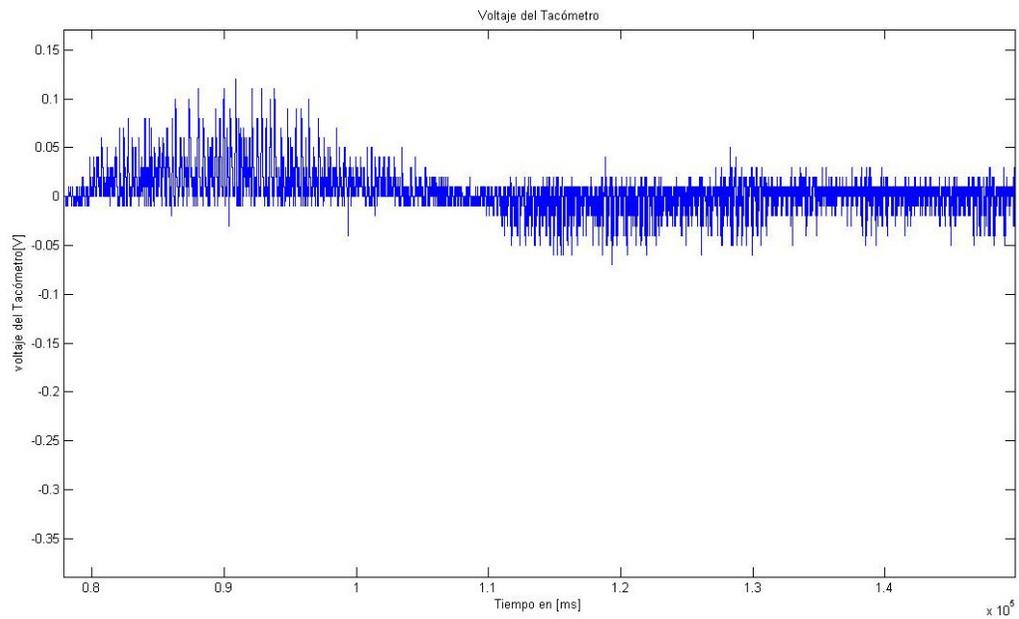


Figura 74. Acercamiento (zoom) de la gráfica del voltaje emitido por el tacómetro.

5.3 PRUEBA EN LABVIEWRUNTIME7.1.

En esta versión, se llevo a cabo la ejecución del programa elaborado anterior mente, y el cual es similar al realizado en LabVIEW 8.6, destacando que la diferencia radica en el tiempo en que se ejecutó, debido a que en LabVIEWRUNTIME7.1 se permite una ejecución de hasta 1[ms]. A

continuación se muestran los gráficos obtenidos con LabVIEWRUNTIME7.1 y de la misma manera que en el caso anterior de resaltar la importancia de la reproducción de datos que se ofrece en todos los paquetes de LabVIEW se ocupó la misma rutina de creación de archivos de texto "TXT" para la realización de los gráficos en matlab.

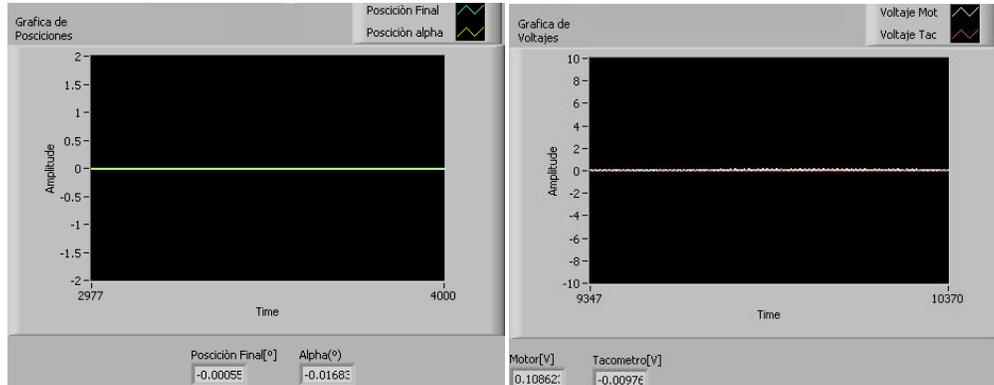


Figura 75. Graficas (voltajes y posiciones) obtenidas con LabVIEWRUNTIME 7.1 al Inicio del programa.

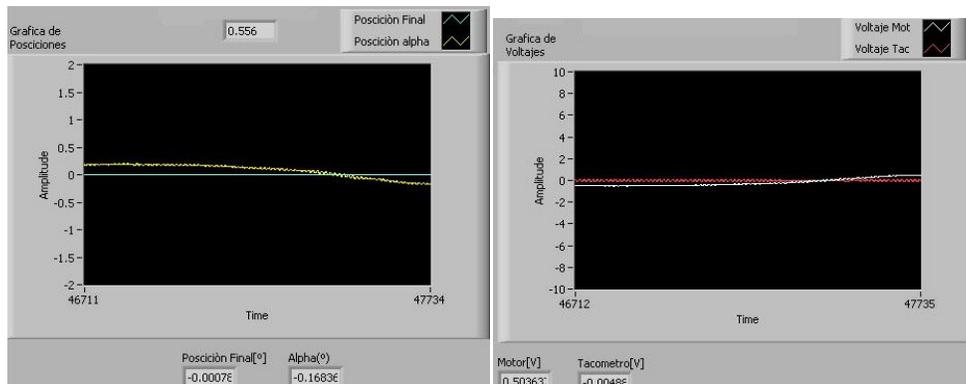


Figura 76. Graficas (voltajes y posiciones) obtenidas con LabVIEWRUNTIME 7.1 durante una perturbación en el sistema.

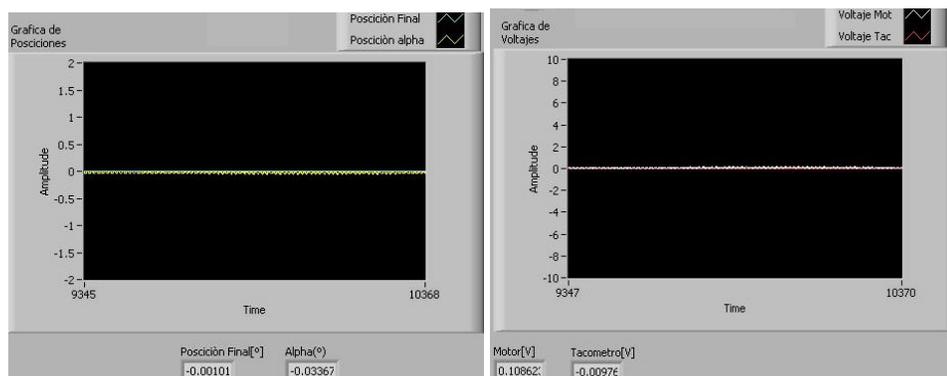


Figura 77. Graficas (voltajes y posiciones) obtenidas con LabVIEWRUNTIME 7.1 al fin del programa.

Obsérvese que al momento de ocurrir una perturbación en el sistema, la posición del ángulo de inclinación alpha sufre una variación muy cercana a cero, corroborándose esto con la posición final que debe de tener el módulo al termino de la rutina, corroborando lo anterior con los voltajes que

emiten el tacómetro y el proceso de control. Realizando un análisis de los datos obtenidos con LabVIEWRUNTIME7.1 en matlab, se obtuvo lo siguiente:

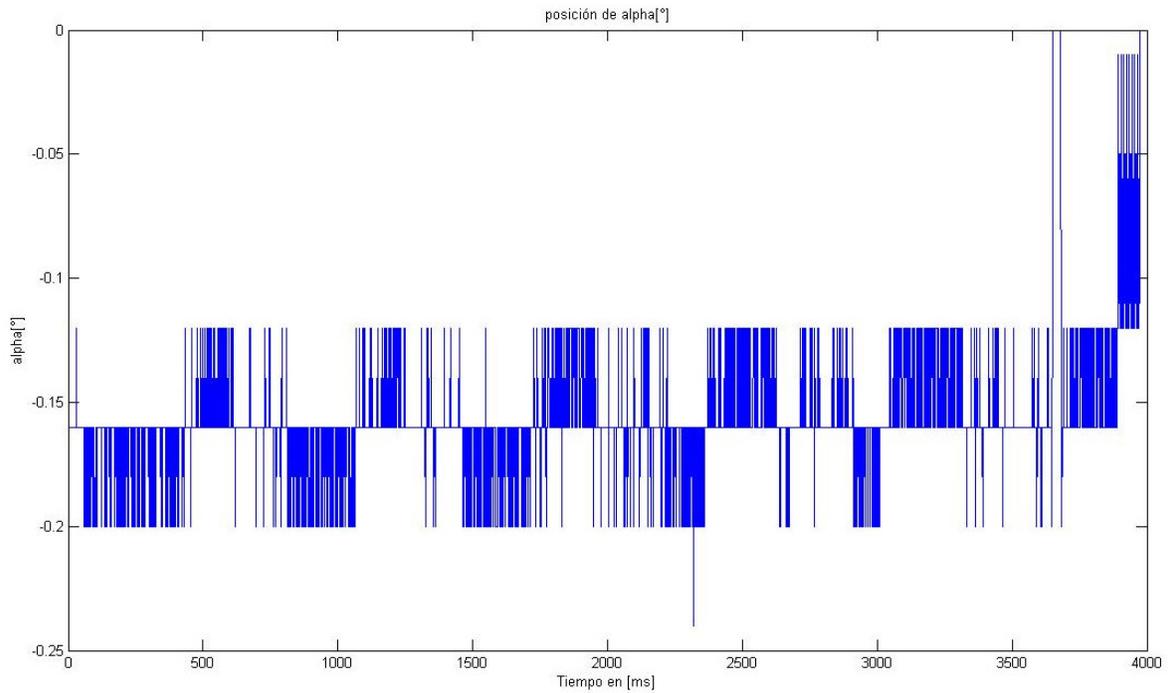


Figura 78. Gráfica de posición del ángulo α (correspondiente a la inclinación del disco giratorio) realizada en matlab.

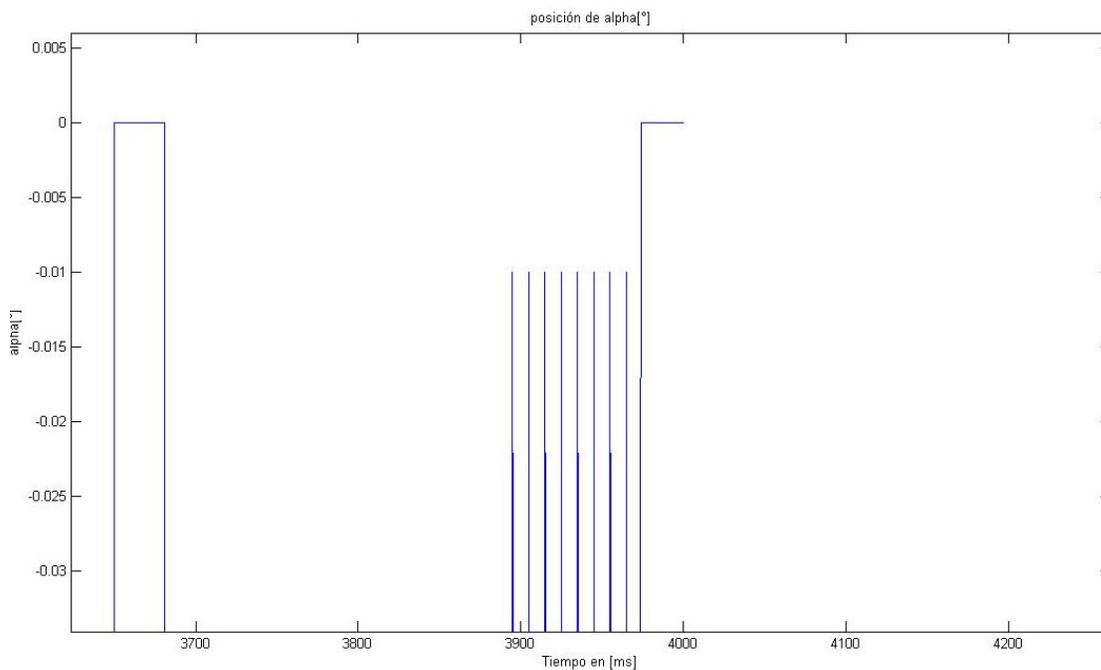


Figura 79 Acercamiento (zoom) de la gráfica de posición del ángulo α (correspondiente a la inclinación del disco giratorio) realizada en matlab.

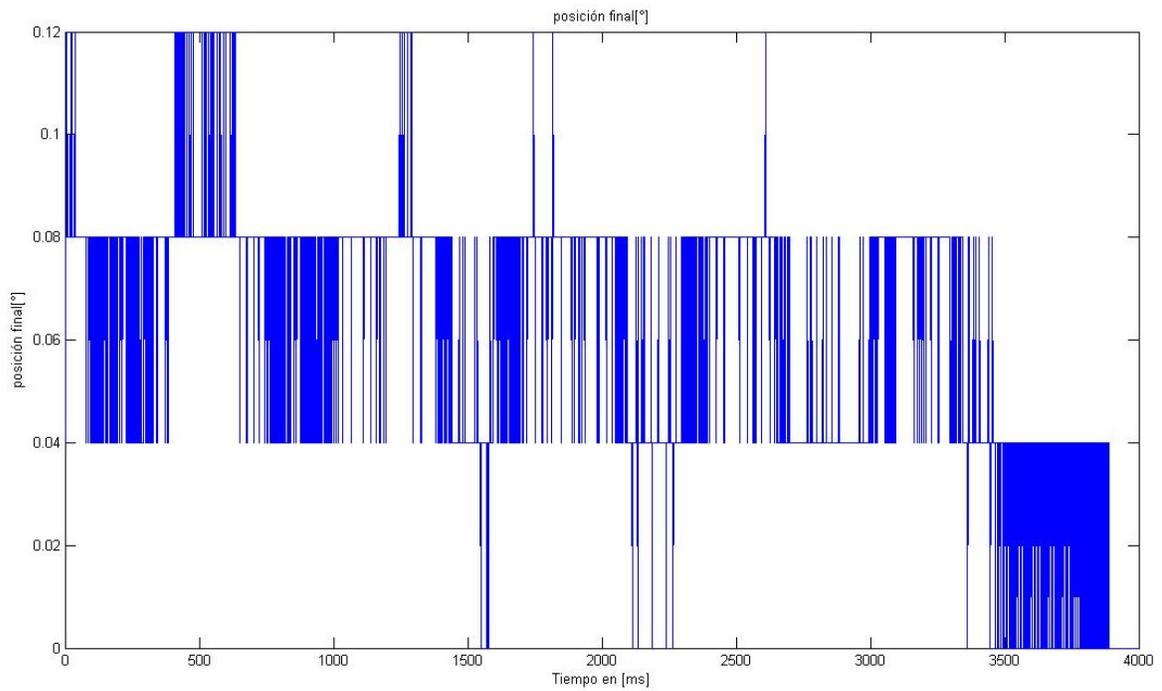


Figura 80. Gráfica de la posición final (correspondiente a la inclinación del disco giratorio) realizada en matlab.

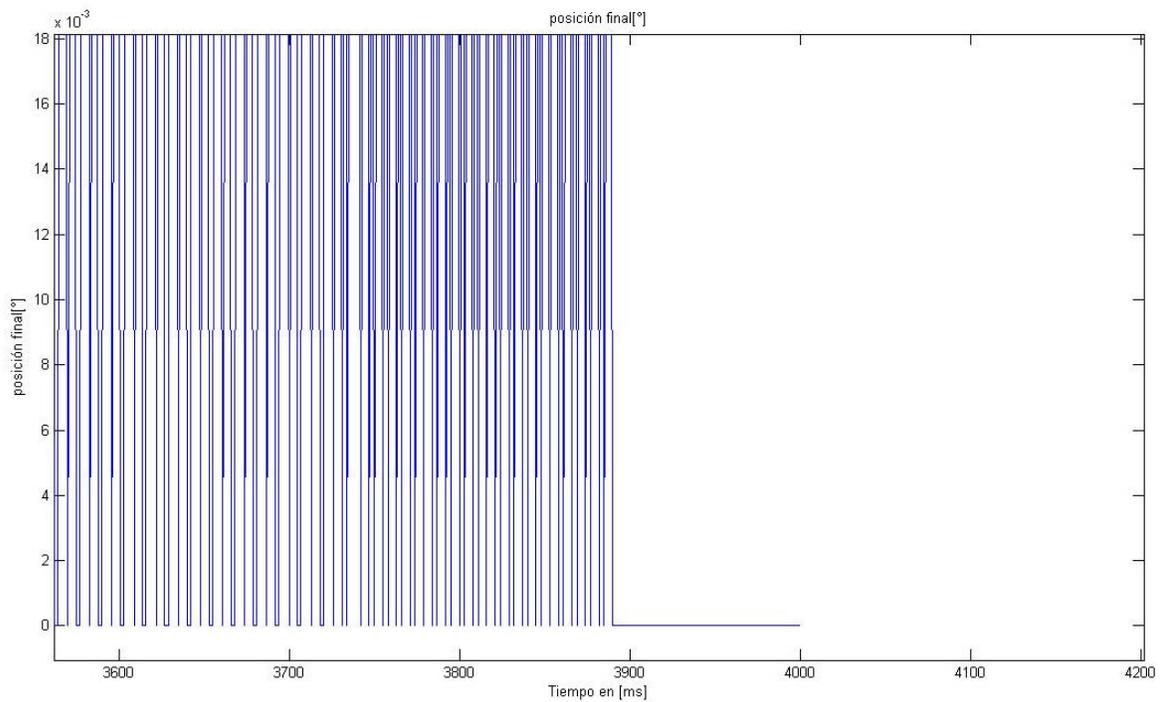


Figura 81. Acercamiento (zoom) de la gráfica de la posición final (correspondiente a la inclinación del disco giratorio) realizada en matlab.

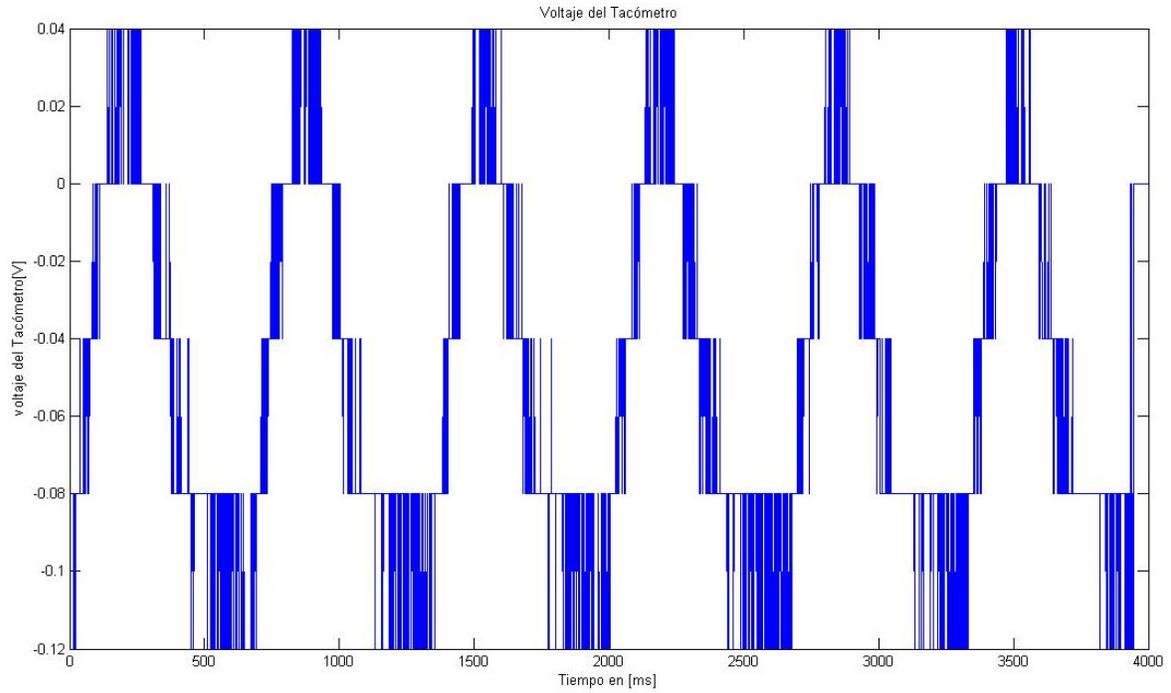


Figura 82. Gráfica del voltaje del Tacómetro obtenida en matlab.

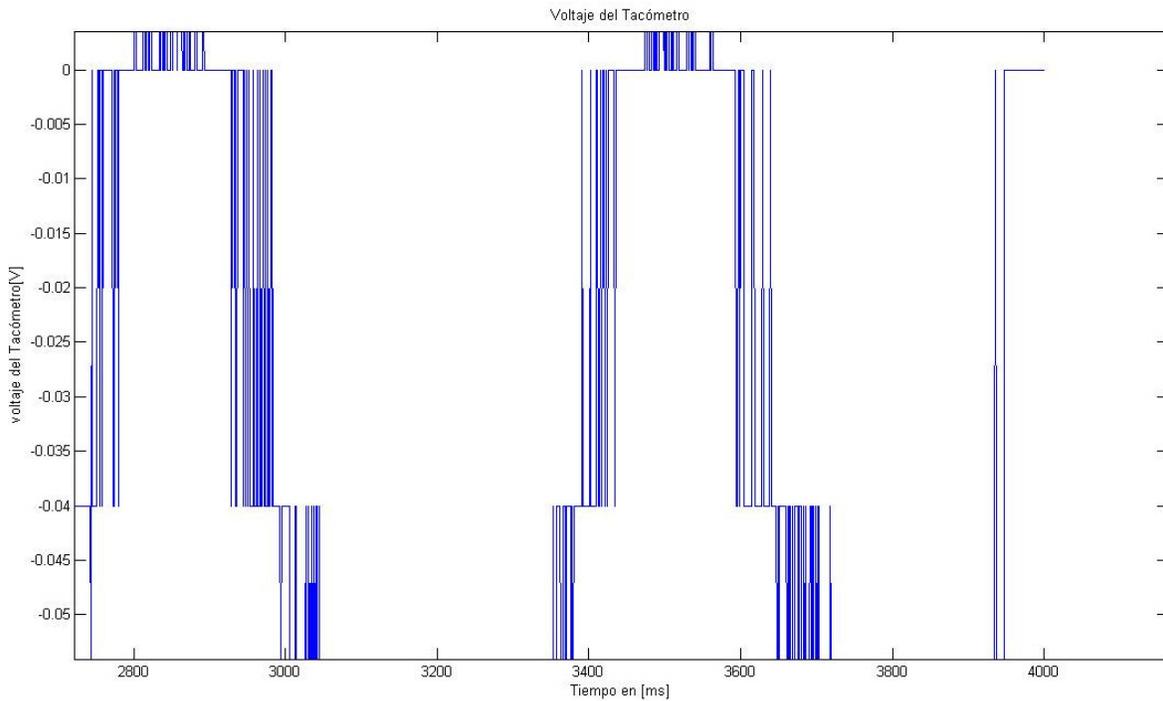


Figura 83. Acercamiento (zoom) de la gráfica del voltaje del Tacómetro obtenida en matlab.

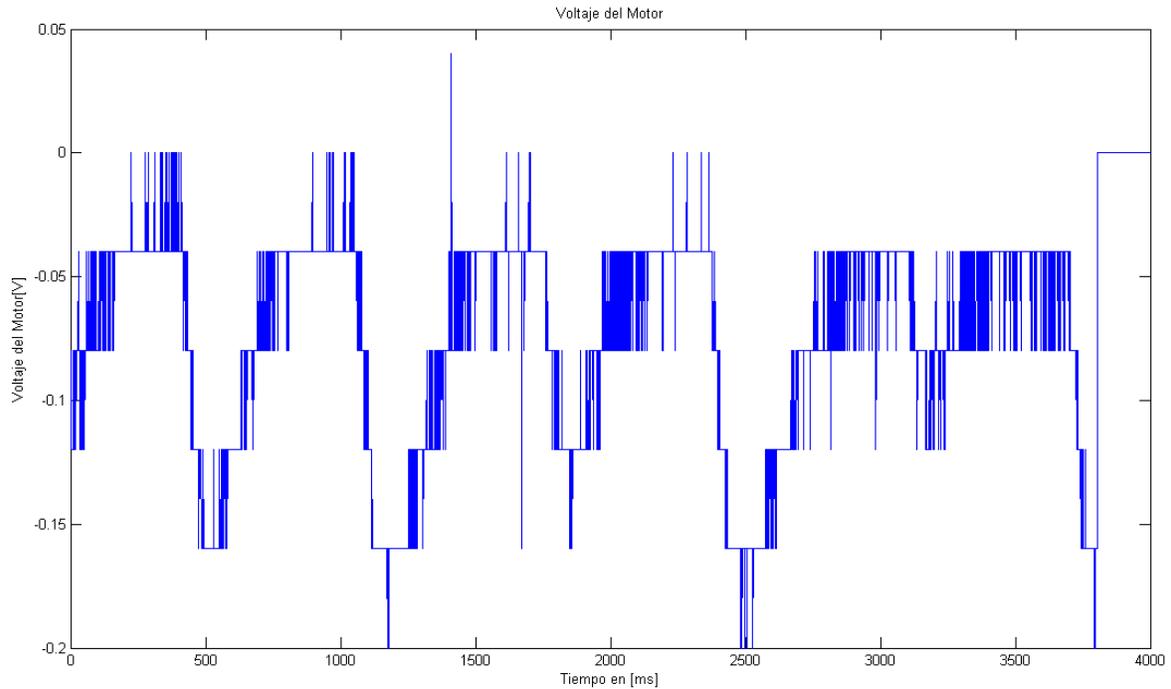


Figura 84. Gráfica del voltaje del Motor obtenida en matlab.

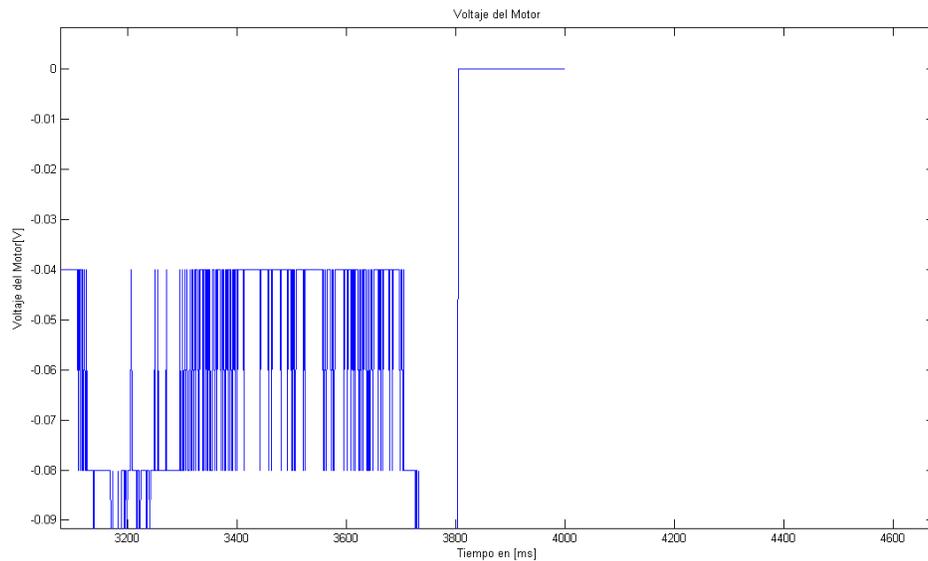


Figura 85. Acercamiento (zoom) de la gráfica del voltaje del Motor obtenida en matlab.

Con ayuda de las gráficas realizadas en matlab se puede corroborar que el control funciona satisfactoriamente, ya que el módulo “GYRO” muestra una variación en la posición angular muy cercana a cero, por lo cual el control está cumpliendo con el objetivo de mantener la posición inicial.

6. Conclusiones

Al término de la investigación se denota que:

El giróscopo se controló de manera satisfactoria gracias a la tarjeta de tiempo real PCI-7041/6040E logrando la estabilidad del mismo. Lo cual consistía en que el ángulo α (correspondiente a la inclinación del disco giratorio) no debe de variar de la posición inicial, pese al movimiento de la base del giróscopo.

El giróscopo requiere de una acción de control inmediata, debido a que el artefacto es sensible a los cambios de posición.

Al trabajar con la Tarjeta 6221E se obtuvo un proceso lento de adquisición de datos, afectando el desempeño de la rutina de control ejecutada en LabVIEW8.6 al mismo tiempo afectando el funcionamiento del giróscopo, debido a que no se estaba trabajando en tiempo real. Mientras que en el proceso realizado con la PCI7041/6040E el comportamiento de la rutina de adquisición de datos fue satisfactorio lo cual permitió que el control se desempeñara de manera adecuada, alcanzando la estabilidad debida a perturbaciones en el giróscopo. Además de que la ejecución de las rutinas se vio afectada en ambas plataformas por lo obsoleto de la computadora en la cual se trabajó.

El giróscopo puede ser controlado a través de otros algoritmos de control, como lo son el PID, modos deslizantes, etcétera; además de otros dispositivos, lo cual puede dar pauta a la continuación de un trabajo futuro implementado con otras técnicas por ejemplo en los laboratorios de control de la Facultad de Ingeniería.

Bibliografía

Optical encoders' applications. Technical Articles, Computer Optical Products, Inc., 1995.

Kees van der Pool. High Resolution Optical encoders. Technical

W. Bolton. Ingeniería de Control. Alfa Omega grupo editor. México 2005

Conversión del encoder de cuadratura.

www.ni.com/pdf/newsletters/esa/q4_2002_esa.pdf

National Instruments Corporation; DAQ Getting Started Guide, junio 2005.

Installation Guide BNC-2120, junio 2003.

<http://www.ni.com/pdf/manuals/372123c.pdf>

LabVIEW™ Execution Trace Toolkit User Manual, abril 2004.

LabVIEW™ Measurements Manual, abril 2003.

LabVIEW™ Real-Time Module User Manual, abril 2004.

PCI E Series User Manual, julio 2002.

PCI-7041 User Manual, octubre 2002. Real-Time Implementation

<http://www.ni.com/pdf/manuals/373360a.pdf>

<http://www.ni.com/labview/embeddedcontrol/implementation>

NI Real-Time Execution Trace Toolkit

http://www.ni.com/toolkits/lv_exec_trace.htm

LabVIEW PID Control Toolkit for Windows

http://www.ni.com/toolkits/lv_pid_control.htm

2D double array shared variable in real-time application

<http://forums.ni.com/ni/board/message?board.id=170&message.id=162426>

Gyro SRV-02 Quanser.

http://www.adeptscience.co.uk/products/controls/quanser/p_quan_R_gyro.html

http://www.quanser.com/english/html/products/fs_product_challenge.asp?lang_code=english&product_code=exp-rot&prod_code=R6-gyro&tmpl=1

<http://www.multion.com.mx/>

APÉNDICE

A.1 PCI 7041

Procesador

Procesador.....	Intel Pentium III
Velocidad del reloj del procesador.....	700 MHz
Velocidad del procesador de Bus.....	100MHz
Memoria.....	32MB DRAM (expansible a 128MB)
Velocidad del Bus de memoria.....	100MHz
Caché on-chip.....	32KB
Unidad de punto flotante.....	si

Memoria compartida

Tipo de memoria compartida.....	SRAM
Tamaño de memoria compartida.....	512Kb

Interface del Bus

Tipo de la interface de Bus.....	3.3V o 5V.PCI esclavo o maestro.
----------------------------------	----------------------------------

Watch dog

Ancho del contador.....	16 bits
Tiempo base.....	Seleccionable (480ns a 15.7ms)
Periodo de espera.....	480ns a 17.2min

Requerimientos de energía

PCI7041 con 6040E

+5 VDC ($\pm 5\%$)	
MAX.....	4.0 A
Típica.....	3.5 A
+12 VDC ($\pm 5\%$).....	80 mA
-12 VDC ($\pm 5\%$).....	12 mA

A.2 PCI 6024E

Bus.....	PCIA
Entradas analógicas.....	16 SE/8DI
Resolución de la entrada.....	12bits
Velocidad máxima de muestreo.....	200KS/s
Salidas analógicas.....	2
Resolución de la salida.....	12bits
Rango de la entrada.....	$\pm 0.05V$ a $\pm 10V$

Muestreo a la salida.....10KS/S
Rango de la salida.....±10V
Rango de la salida digital.....±10V
Contadores/timers.....2, 24 bit