

# **Control en tiempo real de un mecanismo XY virtual por medio de un joystick**

## ***Gustavo Aquino Mendieta***

Instituto Politécnico Nacional, ESIME UA, Av. de las Granjas 682, Col. Santa Catarina, C.P. 02250,  
Deleg. Azcapotzalco, México D.F., Teléfono: 55-16-80-53-30  
*gustavo\_aquino@hotmail.es*

## ***Miguel Magos Rivera***

Universidad Autónoma Metropolitana-Azcapotzalco, Av. San Pablo No.180 Col. Reynosa Tamaulipas,  
C.P.02200, Deleg. Azcapotzalco, México D.F., Teléfono: 53-18-90-00  
*mrm@correo.azc.uam.mx*

## ***Ricardo Godínez Bravo***

Universidad Autónoma Metropolitana-Azcapotzalco, Av. San Pablo No.180 Col. Reynosa Tamaulipas,  
C.P.02200, Deleg. Azcapotzalco, México D.F., Teléfono: 53-18-90-00  
*rgb@correo.azc.uam.mx*

## **Resumen**

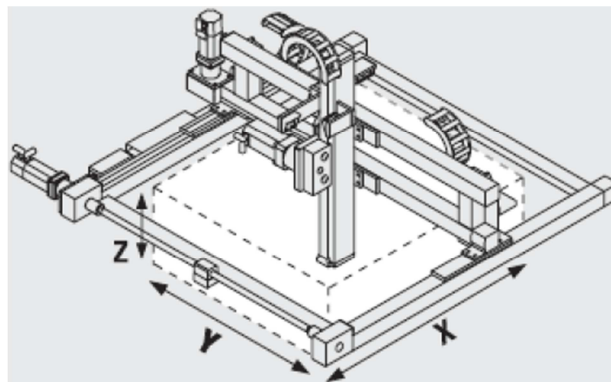
En este artículo se presenta el desarrollo de un mecanismo XY virtual creado en SolidWorks<sup>®</sup>, el cual es controlado en tiempo real por medio de un joystick USB utilizando LabVIEW<sup>®</sup> como interfaz. Se describe la forma en que se realiza la interconexión entre el dispositivo físico (joystick) y el programa desarrollado en LabVIEW<sup>®</sup>, así como de este con el dispositivo virtual en SolidWorks<sup>®</sup>. El trabajo incluye una interfaz gráfica por medio de la cual se puede interactuar con el mecanismo virtual. Se observó que SolidWorks<sup>®</sup> puede funcionar como un entorno de simulación virtual que puede ser de gran utilidad para su aplicación en el desarrollo de interfaces hombre maquina, entre muchas otras aplicaciones.

**Palabra(s) Clave(s):** joystick, LabVIEW®, mecanismo XY, prototipo virtual, SolidWorks®.

## **1. Introducción**

### **1.1. Robots cartesianos**

Los robots que alguna vez se encontraron casi exclusivamente en empresas con grandes volúmenes de producción, están siendo cada vez más utilizados en pequeñas fábricas en una gran variedad de tareas. Lo anterior debido, entre otras cosas, a las altas exigencias de calidad que demanda el mercado, además que la relación beneficio/costo es cada vez más atractiva. En general, el costo de implementación de casi cualquier sistema robótico ha disminuido en los últimos años, especialmente para los de tipo cartesiano. Además de que estos a menudo tienen las mismas capacidades (velocidad, carga, etc.) y tiempos de amortización que otros robots con diferentes configuraciones (SCARA, PUMA, etc.), con la ventaja de una gran variedad de modos de personalización. Estos beneficios han alentado a las pequeñas compañías a explorar el uso de robots en nuevas aplicaciones. Lo anterior es particularmente importante en el caso de los robots cartesianos, pues algunas personas, aún si haberlos utilizado, son capaces de crear nuevos usos para estos. A diferencia de un robot PUMA ó SCARA que no puede ser reconfigurado y posee un conjunto de parámetros y direcciones de movimientos establecidos, un robot cartesiano puede ser fácilmente reconfigurado para ajustarse a los requerimientos de la producción [1]. La figura 1. Muestra un robot cartesiano de 3 grados de libertad.



**Fig. 1. Robot cartesiano [3].**

Los robots cartesianos son generalmente utilizados en la industria en tareas tales como: el ensamble de circuitos electrónicos, operaciones de mecanizado (máquinas CNC) y más recientemente impresoras tridimensionales. Este tipo de robots suelen tener 3 e incluso 4 o 5 grados de libertad.

## **1.2. Prototipos virtuales**

Disciplinas tales como la Ingeniería Mecánica, Electrónica y Computación son combinadas en la creación de productos modernos, esta combinación ha sido conocida como mecatrónica. Comúnmente durante el desarrollo de un sistema mecatrónico, las decisiones de diseño son tomadas de forma independiente por cada uno de los equipos especializados en una de las disciplinas mencionadas, resultando en largos tiempos de diseño y altos costos. Es en este contexto donde la habilidad de crear prototipos virtuales se vuelve útil [2], pues estos ayudan a los ingenieros a diseñar, optimizar, validar y visualizar el desempeño en el mundo real, de máquinas y sistemas de movimiento antes de la creación de prototipos físicos [3]. El prototipado virtual implica el diseño y generación de un producto con características tales que permitan tener un comportamiento lo más apegado posible al modelo físico, excepto que el primero sólo está disponible en un ambiente virtual. De esta manera la realidad virtual (VR) intenta reemplazar la percepción del usuario del mundo circundante, por un ambiente generado por computadora, a menudo llamado entorno virtual 3D. Los usuarios pueden ver e

interactuar con el entorno virtual y manipular directamente objetos en el mismo, este mundo virtual puede ser un modelo generado mediante CAD, una simulación científica, o un gráfico. Con los avances en los gráficos por computadora, la VR se está convirtiendo en la tecnología favorita en muchas aplicaciones industriales que requieren interfaces realistas generadas por computadora [4]. Teóricamente cualquier elemento se puede llevar a su forma de prototipo virtual, la limitante radica en la capacidad de computación y de las herramientas disponibles [5]. Es así que pueden existir prototipos virtuales en campos tales como la medicina, creando atlas virtuales de anatomía humana [6], prototipos de realidad virtual para la medición de expresiones características en situaciones emocionales [7] e incluso simuladores y telesimuladores de cirugías virtuales [8,9].

Mediante una investigación de los programas CAD existentes en el mercado, se encontró que, desde la versión 2009 de la edición profesional del software de diseño y simulación SolidWorks®, es posible manipular ensamblajes desde LabVIEW® [10]. Lo anterior abre todo un abanico de posibilidades, ya que este programa de cómputo ofrece una gran cantidad de herramientas para conectar dispositivos de entrada y salida, así como para el desarrollo de interfaces de usuario.

En este artículo se busca explorar los alcances de las herramientas que cuentan, tanto SolidWorks® como LabVIEW®, para la creación de interfaces hombre máquina. Por tal motivo, se implementó el control en tiempo real de un mecanismo XY virtual creado en SolidWorks®, utilizando el módulo de movimiento NISoftMotion® en LabVIEW® y empleando un joystick USB como dispositivo de entrada. El prototipo virtual es manipulado a través de una de las palancas analógicas del joystick, así como con cuatro de sus botones. El sistema desarrollado permite generar en el prototipo virtual movimientos lineales, ya sea en un solo eje, o interpolando trayectorias lineales en un plano. La sección 2 presenta la descripción general del sistema así como sus principales componentes. El punto 3 aborda la comunicación del joystick USB con LabVIEW®. Por su parte la sección 4 detalla la conexión del software de diseño y

simulación SolidWorks® con LabVIEW®, enseguida se muestra la generación de trayectorias. Finalmente las secciones 6 y 7 presentan los resultados y conclusiones del trabajo realizado.

## 2. Descripción general del sistema

Para facilitar la explicación del sistema desarrollado, este puede ser separado en tres bloques principales: Dispositivo de Entrada, Cálculo de trayectorias y Prototipo Virtual, tal y como se aprecia en la figura 2.

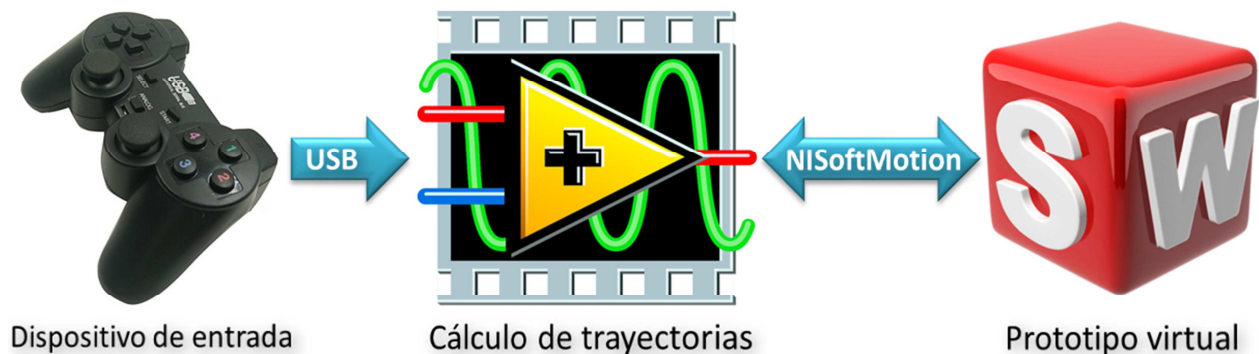


Fig. 2. Diagrama de bloques del sistema desarrollado.

### 2.1. Dispositivo de entrada

Como dispositivo de entrada se empleó un joystick, el cual se conectó con LabVIEW® utilizando el protocolo HID-USB (Dispositivo de Interfaz Humana por sus siglas en inglés), el cual es utilizado para la conexión de interfaces de usuario con computadoras. El principal objetivo de este protocolo es el de homogeneizar la forma en la que los diversos dispositivos de interfaz humana existentes en el mercado se comunican. Antes de la existencia del HID, cada dispositivo tenía su propio protocolo, el cual era útil solo para un tipo de interfaz (ratones, teclados, joysticks, etc.) [11]. Por lo cual la creación de un nuevo dispositivo con nuevas funciones requería el desarrollo de un controlador especial para este. Los dispositivos HID-USB se comunican a través de paquetes auto descriptivos, los cuales pueden contener una gran cantidad de tipos de datos y

formatos. Lo anterior permite la utilización de un solo controlador para una gran cantidad de dispositivos diferentes[12].

El protocolo HID posee dos partes principales: el anfitrión o host y el dispositivo. El dispositivo es aquel que interactúa directamente con el ser humano, tal como los ratones o los joysticks. Por su parte el host se comunica con el dispositivo y recibe los datos que este envía de acuerdo a las acciones ejecutadas por el usuario. Los principales hosts son las computadoras, pero esto está cambiando rápidamente con la inclusión de las nuevas tecnologías móviles tales como los celulares y tabletas. La mayoría de los ratones y teclados en el mercado utilizan un CPU de 8 bits, dado que normalmente se asume que el host es más complejo que el dispositivo [12].

Para el desarrollo de la aplicación presentada en este artículo, se utilizó como interfaz de entrada de datos un joystick USB comercial como el que se muestra en la figura 3. Este cuenta con dos palancas analógicas (cuatro ejes) y doce botones, por medio de estos elementos se controla la posición en la cual se desea ubicar el efector del mecanismo XY virtual.



**Fig. 3. Joystick USB utilizado.**

## **2.2. Cálculo de trayectorias**

Usualmente, una tarea de planificación suele tener tres tipos de resultados: rutas, caminos y trayectorias. Una ruta es un conjunto ordenado de configuraciones que

deben ser alcanzadas por el robot. Por un camino se entiende la discretización de una función continua que interpola las configuraciones definidas en una ruta y finalmente, cuando se habla de una trayectoria, se hace referencia a un camino que tiene asociado un perfil cinemático, es decir, a cada configuración perteneciente al camino se le asocia una velocidad [13].

Es así que para generar el movimiento de un robot en una trayectoria específica es necesario conocer las ecuaciones que definen el camino y determinar las velocidades de los actuadores en función de estas. Si las trayectorias a seguir siempre serán en línea recta sobre el plano definido por los ejes X-Y, es posible utilizar la herramienta “Straight-Line Move”, la cual es parte del complemento NISoftMotion<sup>®</sup> para LabVIEW<sup>®</sup>, para esto basta con definir los ejes que forman el plano en el cuales se generará la trayectoria.

Tomando en cuenta lo anterior se desarrolló una interfaz gráfica en LabVIEW<sup>®</sup>, la cual se encarga de controlar el movimiento del mecanismo XY virtual. El programa utiliza coordenadas relativas para calcular la posición final del efector a partir del movimiento de la palanca analógica izquierda (ejes “X” & “Y”). La figura 4 muestra la interfaz gráfica del programa.

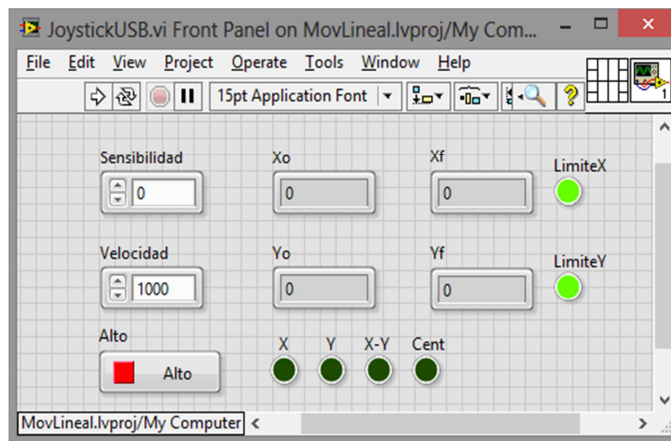
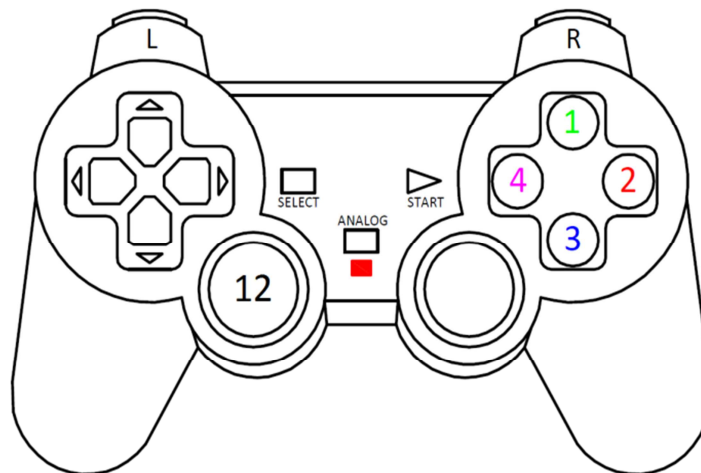


Fig. 4. Interfaz gráfica del programa de generación de trayectorias.

El programa revisa continuamente la posición actual del efector del mecanismo XY y calcula la posición final de este a partir de los datos obtenidos del joystick. El mecanismo XY virtual posee tres tipos de movimiento: en un eje (X ó Y), en ambos ejes (interpolación lineal) y centrado. Cada uno de estos desplazamientos es controlado por un botón distinto del joystick. Es así que el botón “1” mueve el efector a la posición final calculada en el eje Y manteniendo constante la posición del eje X. El botón “2” mueve el efector a la posición calculada en el eje X conservando la posición del eje inicial del eje “Y”. Por su parte el botón “4” genera una trayectoria lineal desde la posición inicial del efector hasta la posición final en el espacio generado por los ejes X y Y. Cuando la palanca izquierda es presionada (botón “12”), el efector se desplaza al centro geométrico del mecanismo XY (215, 215 mm.) sin importar la posición en la que este se encuentre originalmente. La figura 5 muestra la distribución de los botones del joystick utilizado.



**Fig. 5. Vista superior del Joystick USB.**

El objeto *sensibilidad* de la interfaz modifica la resolución del movimiento del efector, mientras que *velocidad* cambia la rapidez con la que este se desplaza en el espacio virtual.



La interfaz cuenta con una serie de indicadores, los cuales encienden de acuerdo al movimiento que se esté realizando, es decir, cuando el efector se encuentra moviéndose en el eje X encenderá el led con la etiqueta “X”, de igual forma para cada uno de los demás movimientos posibles. Cuenta también con dos indicadores que señalan que se llegó al borde del espacio de trabajo virtual.

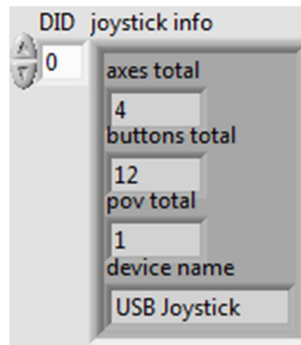
En esta sección se presentaron las principales partes del sistema y se definió la forma en la que funciona la interfaz gráfica del programa de generación de trayectorias. A continuación se describe la conexión del joystick con LabVIEW®, así como las principales características del protocolo utilizado para tal fin.

### **3. Conexión del joystick USB con LabVIEW®**

El joystick se comunica con LabVIEW® utilizando las funciones “Input Device Control”, las cuales forman parte de la suite estándar del entorno de desarrollo. Por medio de estas es posible conectarse con el controlador del dispositivo HID-USB.

Existen 2 propiedades de importancia en todos los dispositivos USB, estas son: su Vendor ID (VID) y su Product ID (PID), ya que estos identifican el fabricante y el tipo de equipo. Además todos los dispositivos USB poseen un GUID o identificador mundial, el cual es único e intransferible. Cada vez que se conecta un dispositivo USB al host, este le asigna un identificador DID [15], el cual es una cadena asignada por el enumerador de dispositivos del Sistema Operativo, un periférico puede tener un solo DID.

El DID del joystick se determinó por medio de la función “Query Input Device”, la cual permite conocer las propiedades del dispositivo HID que se desea conectar a LabVIEW®. La figura 6 muestra la respuesta de esta función, en ella se puede observar el número de ejes y botones que posee el dispositivo. En este caso, como ya se mencionó, el joystick utilizado cuenta con 4 ejes y 12 botones y su DID corresponde, en este caso, a “0”.



**Fig. 6. Propiedades del joystick obtenidas mediante la función “Query Input Device”.**

Una vez que se conoce este parámetro es posible realizar la conexión del joystick por medio de la función “Acquire Input Data”.

En esta sección se mostraron las principales características de los dispositivos HID-USB, así como la forma en la que se realiza la conexión entre LabVIEW® y el joystick USB utilizado. El siguiente punto aborda la interfaz entre el entorno de desarrollo de LabVIEW® y el programa de diseño y simulación SolidWorks®.

#### **4. Integración de LabVIEW® y SolidWorks®**

Para realizar la comunicación entre LabVIEW® y SolidWorks®, se necesitan complementos para ambos. En el caso de LabVIEW®, se utiliza NISoftMotion®, el cual crea la interface con el entorno virtual. Mientras que para SolidWorks® se emplea el complemento Análisis de Movimiento, en este se implementan los motores virtuales que generarán el movimiento del ensamble, mismos que serán controlados desde el entorno de desarrollo.

##### **4.1. Complemento NISoftMotion para LabVIEW®**

La integración de LabVIEW® con SolidWorks® se realiza a través de módulo NISoftMotion, el cual permite el desarrollo de aplicaciones para control de movimiento, generación de trayectorias, interpolación spline y control de posición y velocidad. Todo esto utilizando el estilo de programación gráfica de LabVIEW®. El complemento

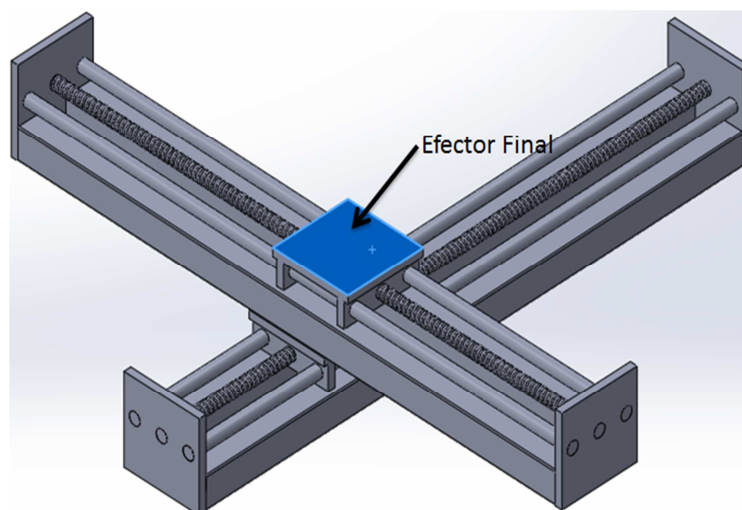
NISoftMotion cuenta con herramientas que permiten la conexión con SolidWorks® a través de la manipulación de motores virtuales previamente definidos en el entorno de diseño. Dichos motores virtuales son las variables que serán controladas desde LabVIEW® para generar los movimientos. Es posible generar desplazamientos en un solo eje, definido por uno de los motores lineales, o en el espacio generado por dos o más de estos. En este trabajo se contempla únicamente el desplazamiento en dos dimensiones.

#### **4.2. Análisis de movimiento en SolidWorks®**

NISoftMotion se interconecta con el complemento SolidWorks® Motion, este módulo de análisis de movimiento puede ser utilizado para estudiar los desplazamientos, velocidades y aceleraciones de los componentes móviles de un ensamble [14]. A través de este complemento se creará un “estudio de movimiento” en SolidWorks®, en el cual serán generados los motores virtuales lineales que se controlarán desde LabVIEW®. Es importante señalar que, previo a crear los motores virtuales, se requiere asignar las restricciones apropiadas al ensamble con el fin de que este se mueva como es debido. Se programó un algoritmo en LabVIEW® que evita la interferencia entre piezas en SolidWorks®, dado que las funciones que este último paquete cuenta para esta situación no aplican cuando el ensamble es manipulado desde LabVIEW®.

#### **4.3. Conexión de LabVIEW® y SolidWorks®**

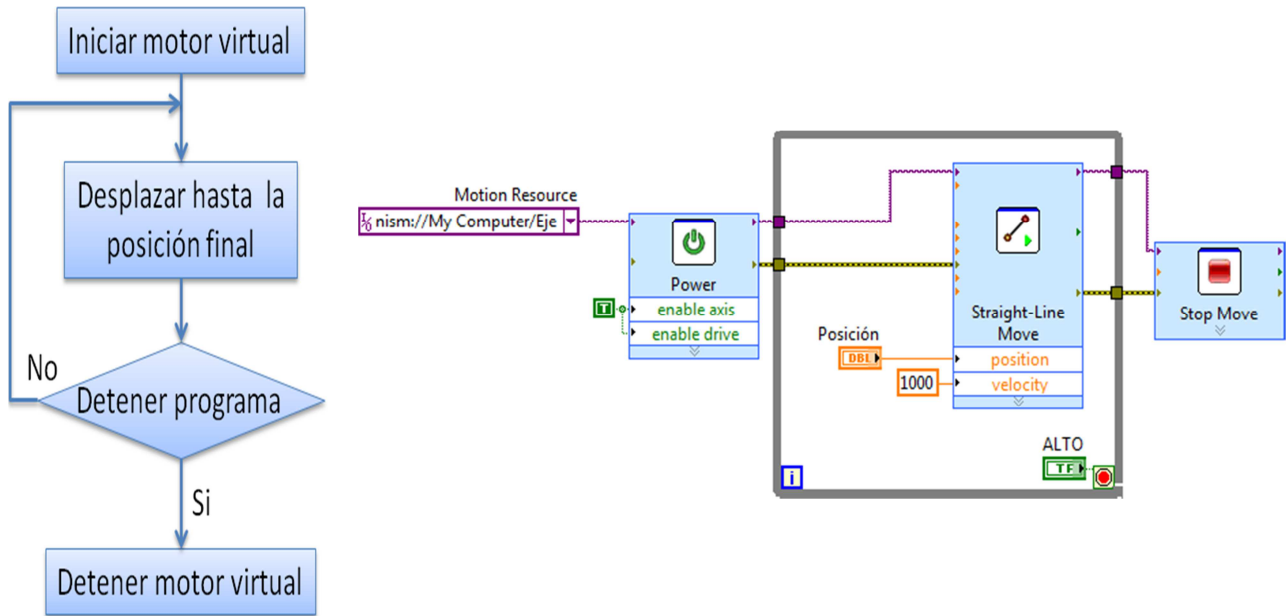
Con el fin de probar las herramientas de movimiento de LabVIEW® se creó un mecanismo XY virtual en SolidWorks® como el que se muestra en la figura 7. Las restricciones de movimiento fueron delimitadas de tal forma que solo se permite el desplazamiento del efector final en los ejes definidos por las guías lineales.



**Fig. 7. Mecanismo XY virtual.**

Los motores lineales creados para generar el movimiento del efector final del mecanismo, fueron agregados al proyecto de LabVIEW<sup>®</sup> para poder ser utilizados en el entorno de desarrollo. Los motores lineales son reconocidos por LabVIEW<sup>®</sup> como ejes. La figura 8. muestra el diagrama de flujo así como el código G de un programa en LabVIEW<sup>®</sup>, que permite la manipulación del efector final en un solo eje utilizando un motor lineal virtual. Las funciones que se utilizan para mover el efector final son:

- *Power*: Habilita el eje o espacio coordinado que será utilizado, así como su correspondiente controlador. Se requiere uno de estos bloques por cada eje que se desee manipular.
- *Straight-Line Move*: Realiza el movimiento del efector en una línea recta sobre un eje o un espacio coordinado. En este caso la variable posición le indica al programa la ubicación final a la cual será movido el efector líneal, mientras que el valor "1000", corresponde a la velocidad a la cual se desplazará el efector en el espacio virtual.
- *Stop Move*: Detiene el movimiento del efector, cuando se deja de ejecutar el programa.



**Fig. 8. diagrama de flujo y código G del programa de prueba.**

En esta sección se explicó la forma en la que se realiza la conexión entre LabVIEW® y SolidWorks®, así como la importancia de las restricciones de movimiento del ensamble.

El siguiente punto describe el funcionamiento del programa que genera las trayectorias del mecanismo virtual XY y las diversas subrutinas que lo componen.

## 5. Generación de trayectorias

Tal como se mencionó en la sección 2, el mecanismo virtual XY puede desplazarse en un eje independiente o crear una trayectoria lineal interpolada en el espacio definido por los ejes X y Y. Además es posible colocar el efector en el centro geométrico del ensamble al presionar un botón del joystick. El programa de generación de trayectorias, está compuesto por una serie de subrutinas, mismas que se muestran en la figura 9. y se describen en las secciones subsecuentes.

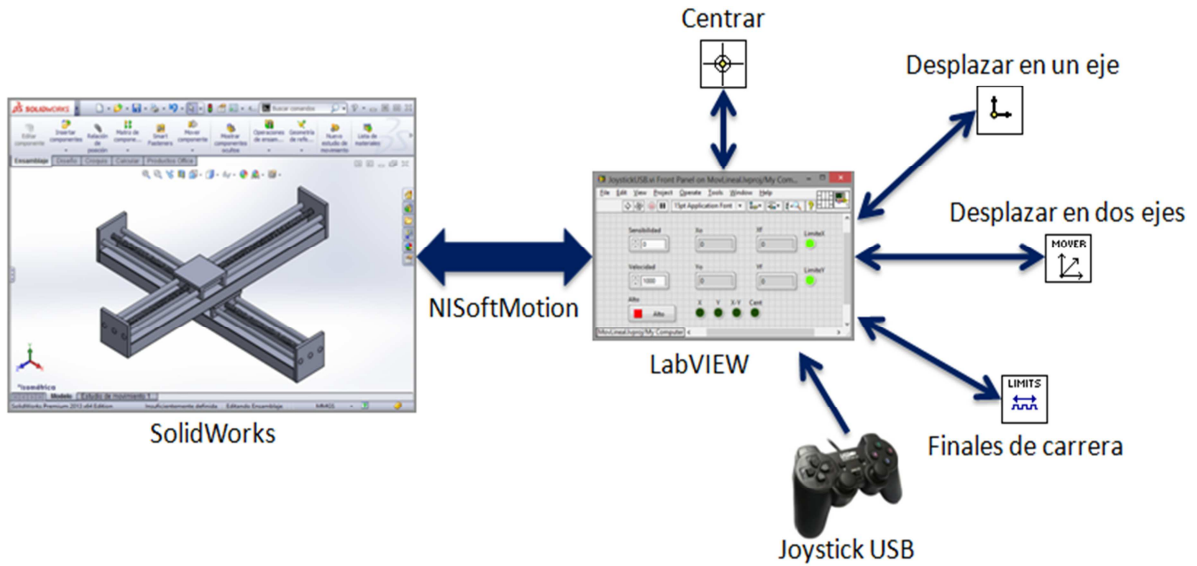


Fig. 9. Arquitectura del programa principal.

### 5.1. Desplazar efector en un eje

En esta subrutina se mueve el efector en un solo eje a velocidad constante, desde la posición en la que este se encuentra inicialmente, hasta la posición final calculada a partir de los valores leídos del joystick. La figura 10. muestra el diagrama de flujo del subprograma y su código G. Esta rutina es una variante del programa descrito en el punto 4.2, con la posibilidad de modificar la velocidad de movimiento del efector final desde la interfaz de usuario por medio del objeto *Velocidad Eje*.

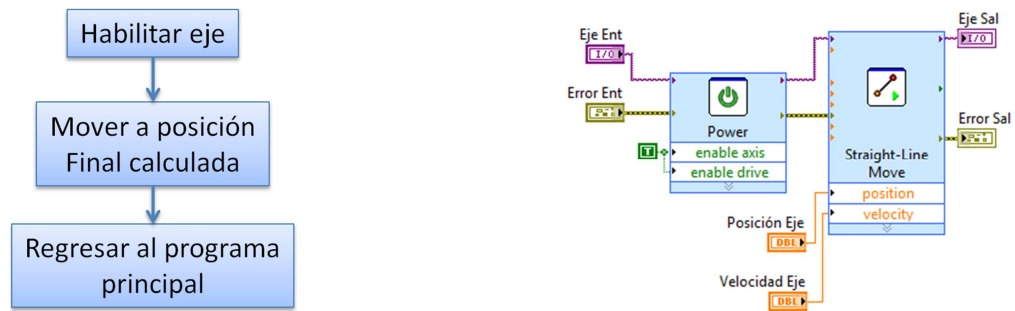


Fig. 10. Diagrama de flujo y código G, del subprograma mover en un solo eje.

## 5.2. Desplazar efector en dos ejes

En esta rutina se mueve el efector en ambos ejes al mismo tiempo desde la posición inicial del efector hasta la posición final calculada a partir de los datos leídos del joystick, generando así una trayectoria lineal con velocidad constante. La figura 11. muestra el diagrama de flujo del subprograma y el código G correspondiente. En esta rutina, a diferencia de los programas descritos en los puntos 4.2 y 5.1, el bloque *Power* habilita un espacio coordinado en lugar de un solo eje. Además, en este caso, la variable posición es un arreglo de 2x1, pues el bloque *Straight-Line Move* requiere dos parámetros, uno por cada coordenada de la posición a la cual se desee desplazar el efector final.

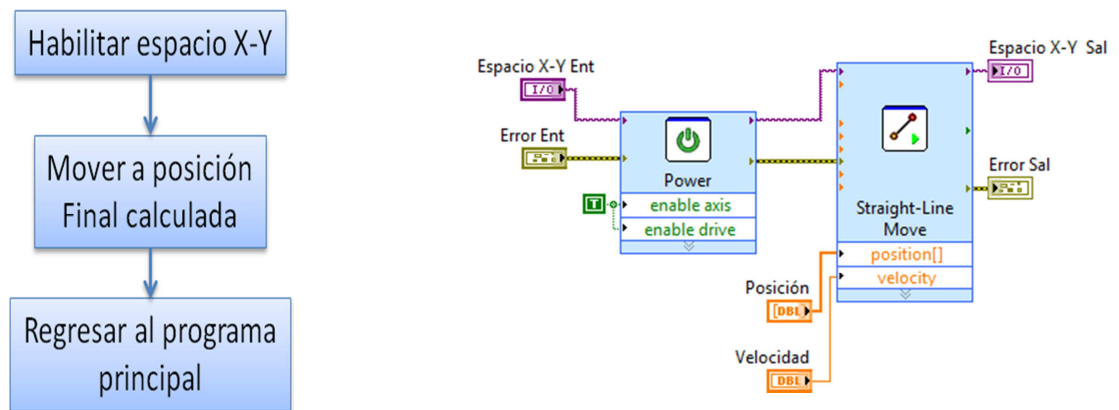


Fig. 11. Diagrama de flujo y código G, del subprograma de mover en dos ejes.

## 5.3. Centrar efector

Esta rutina mueve el efector a la posición central del mecanismo XY virtual a velocidad constante, esta posición corresponde al punto (215, 215 mm.), puesto que el área de trabajo es de 430mm. para ambos ejes. La figura 12 muestra el diagrama de flujo de la rutina y su código G, puede observarse que, a diferencia del programa descrito en la sección 5.2, el parámetro de entrada "position" del bloque *Straight-Line Move* es un arreglo constante.

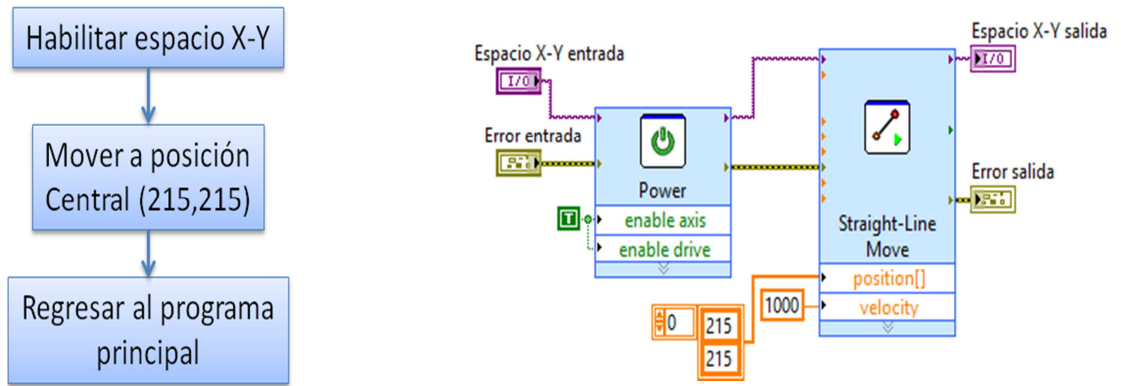


Fig. 12. Diagrama de flujo y código G, del subprograma de centrado.

### 5.4. Interruptores de final de carrera

Este subprograma se encarga de evitar interferencias entre el efector final y los extremos del modelo virtual, para ello compara la posición final a la cual llegará el efector al ser desplazado con los límites superior e inferior del espacio de trabajo del mecanismo XY (0-430mm.). Además enciende el indicador “LIMITE” en la interfaz de usuario, cuando se ha alcanzado cualquiera de los bordes del área de trabajo. La figura 13. muestra el diagrama de flujo del subprograma.

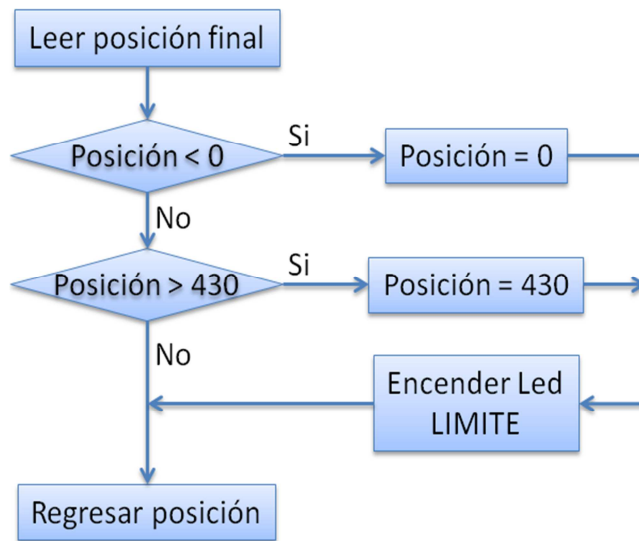


Fig. 13. Diagrama de flujo del programa de revisión de interferencia.



En esta sección se describió la forma en que funciona el programa de generación de trayectorias, además de las partes que la componen. A continuación se presentan y se discuten los resultados obtenidos.

## 6. Resultados

Las pruebas efectuadas mostraron un movimiento estable del efector debido a que el joystick USB elimina el ruido ocasionado durante el muestreo de las palancas analógicas. Lo anterior permite obtener desplazamientos uniformes del efector en el área de trabajo. Las figuras 14 a 17 muestran el mecanismo durante diversas pruebas de movimiento. En estas se pueden observar dos flechas rojas encima del efector final, estas representan los motores virtuales con los que cuenta el ensamble en SolidWorks®.

La Figura 14. muestra el efector moviéndose en un solo eje, se puede observar que el indicador con la etiqueta “Y” se encuentra encendido, esto señala que el efector se encuentra desplazándose, lo anterior sin tomar en cuenta un posible cambio del valor de la posición del eje “X”.

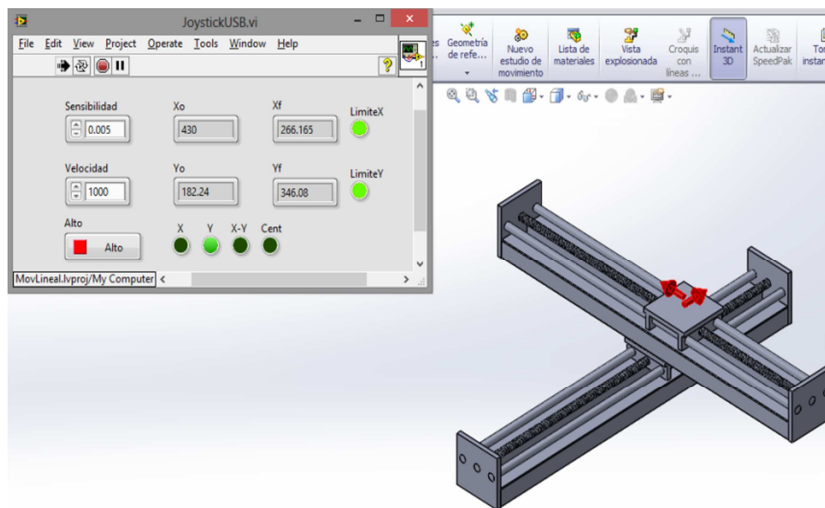
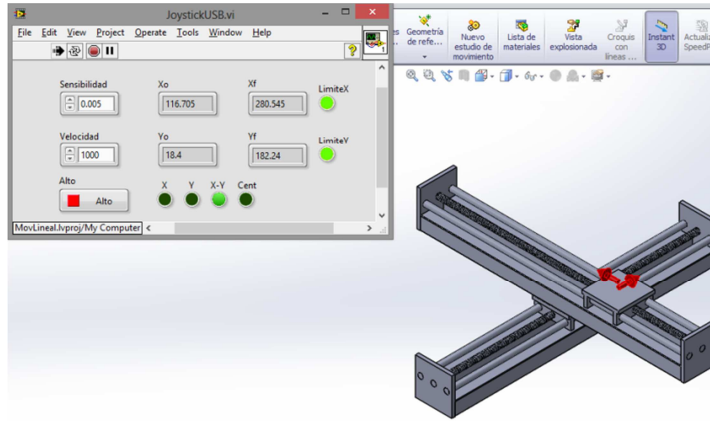


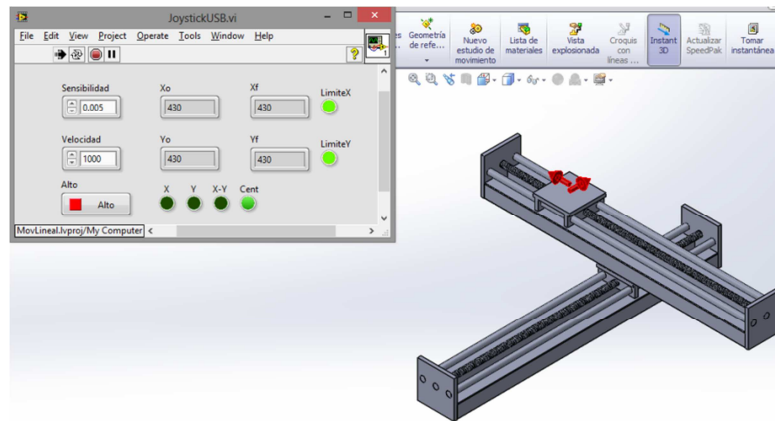
Fig. 14. Efector moviéndose a lo largo del eje “Y”.

En la figura 15 se observa el efector final moviéndose en una trayectoria lineal en el plano X-Y desde la posición inicial (116.705, 18.4) hasta la posición final (288.545, 182.24), se puede observar que el indicador con la etiqueta “X-Y” se encuentra encendido, señalando así que el efector se encuentra desplazándose.



**Fig. 15. Efector moviéndose en ambos ejes al mismo tiempo.**

Se puede apreciar en la figura 16 al efector final mientras se desplaza desde la posición inicial correspondiente a los límites superiores de ambos ejes (430, 430), hasta el centro geométrico del mecanismo XY virtual. Se observa también que el indicador con la etiqueta “Cent” se encuentra encendido, señalando que el efector se está desplazando hacia la posición central del área de trabajo.



**Fig. 16. Efector desplazándose a la posición central.**

La figura 17 muestra el efector final en la posición correspondiente a los límites superiores de ambos ejes, por ello los indicadores con las etiquetas “LímiteX” y “LímiteY” se encuentran encendidos en color rojo. El subprograma que revisa la posición instantánea del efector evita que exista interferencia en el ensamble.

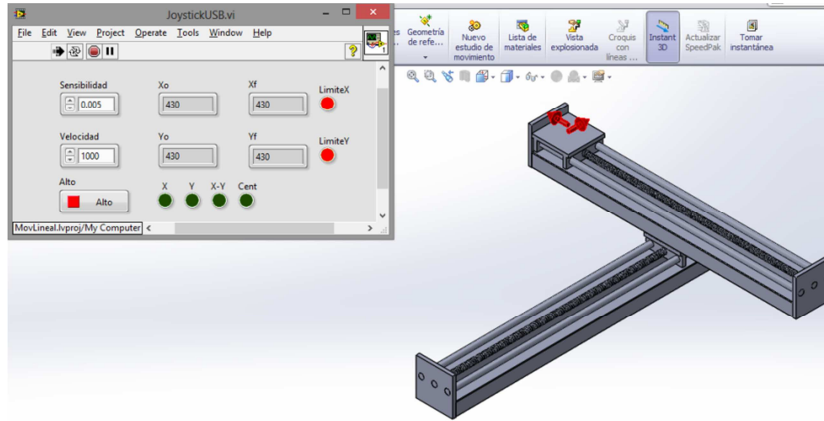


Fig. 17. Efector final en los extremos de los ejes X & Y.

## 7. Conclusiones

En este artículo se presentó el desarrollo de una interfaz hombre máquina para el control de un mecanismo virtual XY utilizando un joystick USB. Se evaluaron las herramientas de movimiento lineal, tanto en uno como en dos ejes, con la finalidad de crear una interfaz hombre máquina que permita la manipulación de objetos virtuales utilizando un dispositivo de entrada.

El software CAD existente en el mercado es una herramienta muy poderosa y versátil, sin embargo es limitada cuando se requiere la manipulación de un entorno virtual a partir de estímulos externos. Es en este contexto donde la utilización de la herramientas NISoftmotion® y SolidWorks® Motion obtiene particular importancia, ya que permiten la integración de elementos virtuales y físicos, posibilitando una simulación mucho más interactiva para el usuario.

La sinergia de los dos paquetes de cómputo empleados en este trabajo, ofrece amplias posibilidades para desarrollar y probar controladores más sofisticados, ya que SolidWorks® cuenta con herramientas tales como sensores virtuales de posición, velocidad, aceleración y fuerza, que pueden ser leídos utilizando LabVIEW®. Esto aunado a la gran versatilidad de diseño de SolidWorks®, lo que ofrece una amplia gama de posibilidades para crear y validar prototipos virtuales antes de incurrir en altos costos de fabricación.

Hasta el momento sólo se han generado trayectorias en línea recta, el trabajo a futuro consiste en modificar la interfaz gráfica dándole la capacidad al usuario de generar trayectorias en curvas suaves. También se plantea la posibilidad de conectar diversos dispositivos a la computadora tales como PLC's, tarjetas de adquisición de datos entre otros, aumentando la interactividad del entorno virtual con el usuario.

## **8. Reconocimientos**

Este trabajo ha sido posible gracias al apoyo económico del CONACYT a través del programa de becas para estudios de posgrado inscritos al PNPC #591875 y al Instituto Politécnico Nacional a través del proyecto 25151609 "Caracterización por TEMP-PEELS y AFM de los precipitados inducidos por deformación de dos aceros de ultra-bajo Carbono al Niobio".

## **9. Referencias**

- [1] Drive & Control profile. Bosch Group. USA. 2013.
- [2] Robot Cartesiano XY Guía aplicativa. Schneider Electric. España. 2010.
- [3] Que es y para qué sirve el prototipado virtual. <http://www.utp.edu.co/~gtorres/VARDOCS/Virtual%20prot.pdf>. Revisado: 09/06/2015.

- [4] P. McHugh, Rynedel 1er autor, Virtual Prototyping of a Mechatronics Device. 1a edición. 2011. Purdue e-Pubs. Purdue Indiana. 8-15 pp.
- [5] Kiwan Han, Jeonghun Ku, "Virtual reality prototype for measurement of expression characteristics in emotional situations". *Computers in Biology and Medicine*. No. 39. 08/12/2008. 173-179 pp.
- [6] Generación de prototipos virtuales. <http://www.ni.com/virtualprototyping/esa/>. 09/06/2015.
- [7] A. H. Meier, C. L. Rawn, "Virtual Reality: Surgical Application-Challenge for the New Millennium". *Journal of the American College of Surgeons*. Vol. 192. No. 3. 04/2001. 372-384 pp.
- [8] W. Qing Hui, L. Jing Rong, "Interactive visualization of complex dynamic virtual environments for industrial assembliesulo". *Computers in industry*. Vol. 57. 25/11/2005. 366-377 pp.
- [9] A. H. Filip Górski, D.G. Przemyslaw Zawadzki, "Virtual 3D Atlas of a Human Body-Development of an Educational Medical Software Application". *Procedia Computer Science*. Vol. 23. 2013. 302, 314 pp.
- [10] Kiwan Han, Jeonghun Ku, "Virtual reality prototype for measurement of expression characteristics in emotional situations". *Computers in Biology and Medicine*. Vol. 39. 08/12/2008. 173-179 pp.
- [11] Introduction to NI SoftMotion for SolidWorks. National Instruments. USA. 2014.
- [12] HDI Class Device Interface Guide. Apple Inc.. USA. 2009.
- [13] Device Class Definition for Human Interface Devices (HID). Firmware Specification, USB Implementers.1996-2001.

- [14] J. J. Craig, *Robótica*. Tercera edición. 2006. Prentice Hall. México D.F. 2001-230 pp.
- [15] *Introducción a las aplicaciones de análisis de movimiento con SolidWorks® Motion*. Dassault Systèmes SolidWorks® Corporation. Massachusetts. EE. UU. 2010.
- [16] J. A. Vicaria Flores, “Aplicaciones del controlador Bluetooth en robótica”. Proyecto de fin de carrera. ETSI de la Universidad de Sevilla. 2007.

## **10. Autores**

Ing. Gustavo Aquino Mendieta es Ingeniero en Mecánica por la Universidad Autónoma Metropolitana, actualmente se encuentra realizando la Maestría en Ingeniería de Manufactura en la SEPI ESIME Unidad Azcapotzalco del IPN.

Dr. Miguel Magos Rivera es Ingeniero en Electrónica por la Universidad Autónoma Metropolitana, Maestro y Doctor en Control Automático por la Universidad Claude Bernard de Lyon (Francia). Actualmente es profesor investigador en la Universidad Autónoma Metropolitana Unidad Azcapotzalco.

Ing. Ricardo Godínez Bravo es Ingeniero en Electrónica por la Universidad Autónoma Metropolitana. Actualmente se encuentra cursando la Maestría en Mecatrónica en el Instituto Tecnológico de Ecatepec además de ser profesor-investigador en la Universidad Autónoma Metropolitana unidad Azcapotzalco.