UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

DISEÑO E IMPLEMENTACIÓN DE UN PROTOTIPO VIRTUAL DE UNA MESA DE COORDENADAS

TESIS PROFESIONAL

para obtener el título de

INGENIERO MECATRÓNICO

Presenta:

Cuauhtémoc Vladimir Luna Jiménez

Director de Tesis:

M.I. Serafín Castañeda Cedeño

AGRADECIMIENTOS

A la Universidad Nacional

Autónoma de México,

mi hermosa casa de estudios

y alma máter.

A mis padres,

los mejores maestros

que pude haber tenido.

Al M.I. Serafín Castañeda Cedeño,
por brindarme todo su apoyo y amistad
y guiarme en la realización de esta tesis.

Al M.I. Humberto Mancilla A.,
por compartirme su amistad y
guiarme en la realización de esta tesis.

Al gran arquitecto del universo.

Dedico esta tesis

A mí amada novia Giovanna, compañera y amiga incondicional en este hermoso sueño llamado vida.

A mis queridas hermanas Tania y Nadia, las dos lucecitas que iluminan mi camino en todo momento.

A mis adorados padres Martha y Enoc, arquitectos de mi pasado, maestros de mi futuro, mis más grandes ejemplos de vida.

A mis grandes e incomparables amigos, Arturo, Daniel, Fernando, Paola, Edwin, Dulce, Vianet y Elizabeth.

A todos los que me han acompañado en este gran viaje y no he podido mencionar aquí por falta de espacio, quienes han contribuido de alguna forma a la culminación de esta carrera.

“No espero nada. No temo nada. Soy libre.”

Nikos Kazantzakis

Contenido

CAPÍTULO 1

Introducción

Día a día el ser humano se ve en la necesidad de automatizar procesos de tareas rutinarias para enfocar su tiempo de una mejor forma o para hacer su vida más cómoda y más segura. La mecatrónica se encarga de dar soluciones reales a estas necesidades, por ejemplo, la automatización de un vehículo, es decir, que pueda conducirse por el camino sin la necesidad de que una persona esté al volante, utilizando sensores y cámaras para percibir su entorno y uno o varios controladores para tomar decisiones acerca de la velocidad, la dirección, frenado, etc. A quién no le gustaría poder viajar al trabajo mientras duerme o mientras se baña o se arregla dentro de su automóvil, o regresar de la escuela mientras se está sentado viendo la televisión o durmiendo una siesta, la mecatrónica hace posible que todo esto y más se esté desarrollando actualmente alrededor del mundo.

En la actualidad, la mecatrónica es parte fundamental de nuestra sociedad, debido a sus problemas crecientes y sus nuevas necesidades, en las que, los robots pueden ser la solución a muchos de ellos. Pero los robots necesitan ser diseñados e implementados por los ingenieros, los cuales ocupan todas las herramientas que tienen a su alcance, principalmente una computadora.

Una vez diseñado el robot o máquina se realiza un prototipo, el cual no siempre es funcional o tiene fallos en el diseño que lo hacen inoperable, lo que representa gastos de fabricación y ensamble desperdiciados, que, algunas veces llegan a ser grandes sumas de dinero. Por tal motivo se están implementando nuevas herramientas de software y hardware para la creación de prototipos rápidos y virtuales que disminuyan los costos de fabricación de nuestros prototipos, de tal forma que un día se pueda dar el salto del diseño virtual hasta la fabricación en serie del mismo, sin vernos en la

necesidad de crear un prototipo que, muchas veces por falta de recursos sólo se queda en una idea o en un diseño CAD (Diseño Asistido por Computadora, por sus siglas en inglés).

En esta tesis se diseñará un modelo virtual de un clásico diseño mecatrónico, una mesa de coordenadas, en la que, para probar su completo funcionamiento se resolverá uno de los problemas tradicionales usados en los concursos de robótica, el posicionamiento u ordenamiento de fichas de diferentes colores.

El modelo virtual (que por sus características también puede ser llamado robot virtual) que se presentará en esta tesis será capaz de reconocer el color y la posición de la ficha por medio de software de visión y luego organizar cada una de las fichas reconocidas en su depósito correspondiente.

También se busca que los alumnos de ingeniería (inicialmente mecatrónica) puedan usar esta tesis para ayudarles a crear sus diseños en CAD y operarlos a través de *LabVIEW* con su programación gráfica, para que ahorren tiempo y dinero en sus diseños, además de que, quizás algún día puedan entregar un prototipo virtual en lugar de uno físico.

1.1 Antecedentes

El término mecatrónica fue concebido por primera vez por un ingeniero llamado Tetsuro Moria de una compañía japonesa en 1969, como la combinación de mecanismos y electrónica. A través del tiempo, el significado de este término se ha ido ampliando y ahora es utilizado en el lenguaje técnico para describir la filosofía de la ingeniería en tecnología más que para la tecnología en sí. Como resultado de que el término mecatrónica ha sido ampliado, pueden encontrarse numerosas definiciones en la literatura. La definición más utilizada enfatiza a la sinergia y sus estados: "la mecatrónica es la integración sinérgica de ingeniería mecánica con electrónica y control inteligente por computadora para el diseño y manufactura de productos y procesos". (1)

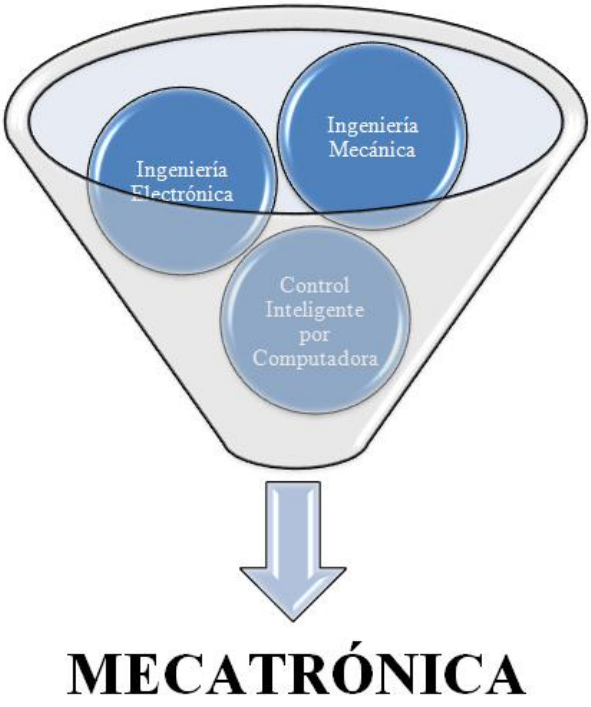


Figura 1.1 Mecatrónica, integración sinérgica de ingeniería electrónica, ingeniería mecánica y control inteligente por computadora.

En el siguiente diagrama se puede apreciar claramente la evolución de la tecnología a través de los años en el siglo XX hasta llegar a lo que hoy se conoce como mecatrónica.

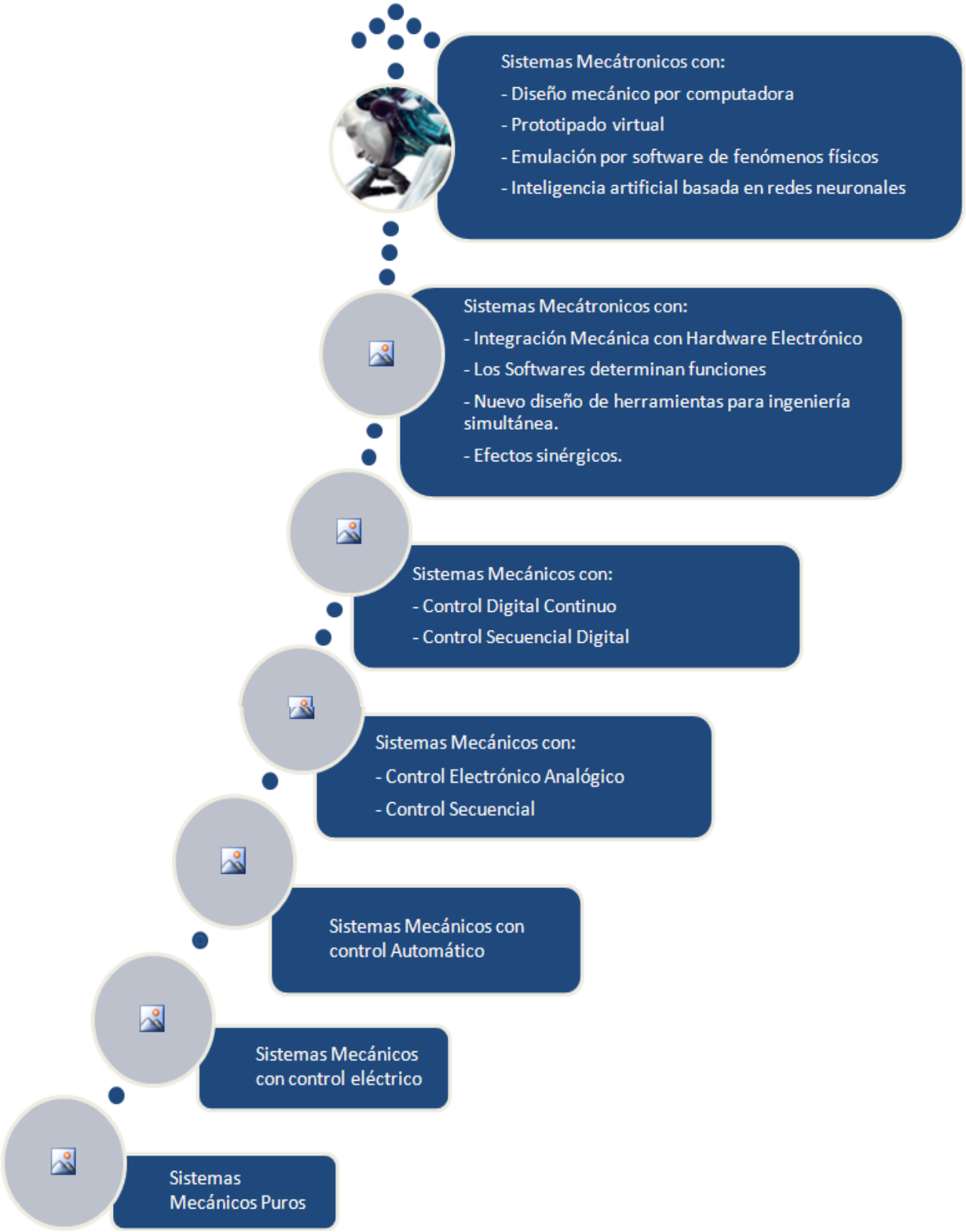


Figura 1.2 Desarrollo histórico de los sistemas mecánicos, eléctricos y electrónicos. (2)

Los sistemas mecatrónicos son implementados en casi cualquier sector de la sociedad, desde las grandes industrias en donde se emplean grandes, complejos y sumamente exactos robots y máquinaria CNC hasta los pequeños hogares donde se utilizan cientos de dispositivos mecatrónicos, por ejemplo una lavadora automática o un reproductor de discos compactos o blu-ray. Los campos en donde la mecatrónica actúa o puede actuar son simplemente infinitos.

1.2 Sistemas Mecatrónicos

Un sistema mecatrónico consiste en emular un sistema biológico utilizando diferentes herramientas para llegar a ese fin: sensores para percibir su entorno, controladores para tomar decisiones, acondicionamiento de señales para "traducir" o entender mejor lo que sus sensores identifican, hardware para protegerlos y soportarlos, dispositivos de interface para poder comunicarse, fuentes de poder para tener la energía necesaria para su funcionamiento, etc.

No está de más decir que, dentro de cada dispositivo hay también más mecatrónica implícita, puesto que, ingenieros han trabajado en su diseño y en su producción, ya que cada uno de ellos cuenta en su interior con diversos sensores, controladores, etc, lo que se asemeja aún más con los sistemas biológicos y sus subsistemas, formando así una cadena interminable de dispositivos trabajando en conjunto para que un sistema central funcione y "viva", en el sentido más ambiguo de la palabra.

Un sistema mecatrónico debe ser tratado como un sistema de control, cuyo objetivo es realizar tareas, cálculos y procesos, de forma autónoma, precisa y exacta. Normalmente los ingenieros se refieren a un proceso, máquina o dispositivo, a ser controlado como "planta" y se dedican a realizar un estudio previo o análisis de las entradas que tiene la planta y las salidas que se necesitan, por ejemplo, en una máquina para hacer café, la planta es la máquina para hacer café, las entradas son agua, café, azúcar y energía eléctrica, y la salida es el café ya preparado.

Se estudia la forma en la que se hace el café, la forma de mezclar los ingredientes, las cantidades, la temperatura necesaria, la energía requerida para alcanzar dicha temperatura y el tiempo necesario. Todo esto se traduce en fórmulas matemáticas para tener un solo modelo matemático general que nos servirá para controlar la máquina. A todo esto se le llama modelado de sistemas físicos, en donde se utiliza desde lógica y física, hasta álgebra y cálculo avanzado.

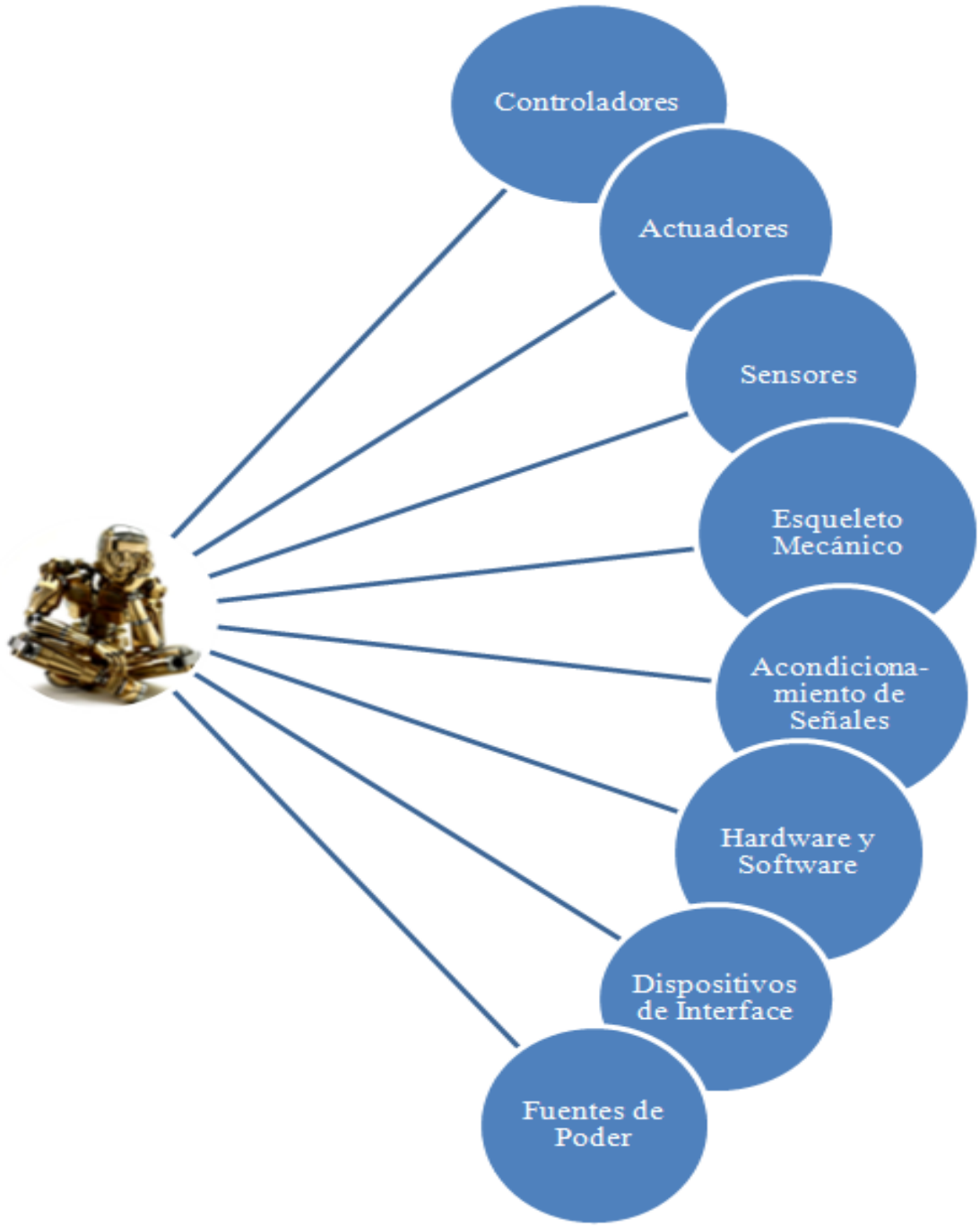


Figura 1.3 Componentes de un típico sistema mecatrónico.

La función del sistema mecatrónico se centra principalmente en la planta. Actuadores, sensores y los dispositivos de modificación de señales deberán integrarse a la planta, o deberán ser necesitados como componentes que son externos a la planta, para una apropiada operación de todo el sistema mecatrónico. El controlador es una parte esencial del sistema mecatrónico. Este genera las señales de control a los actuadores para operar la planta de la manera deseada. Las señales registradas pueden ser utilizadas para sistemas de monitoreo y control de conducta, y control de retroalimentación. (3)

En la Figura 1.4 se puede apreciar un diagrama de bloques de retroalimentación clásico de un sistema mecatrónico.

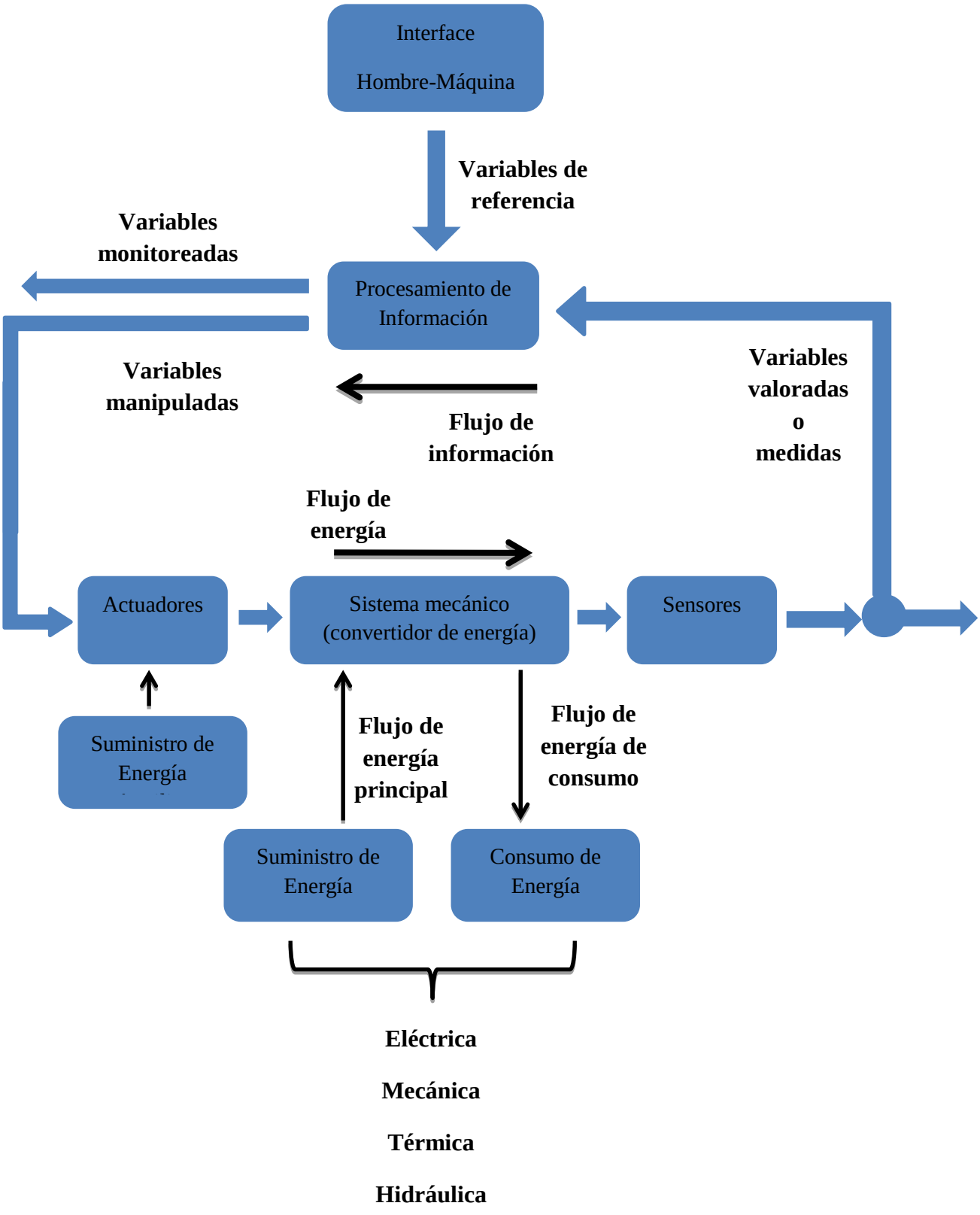


Figura 1.4 Diagrama de bloques de un típico sistema mecatrónico con retroalimentación.

1.3 Concepto de "Robot"

Un robot es una entidad mecánica o virtual inteligente programada para realizar tareas por su propia cuenta o de forma guiada, es decir que puede ser autónomo o semi-autónomo.

El término "robot" viene del término "roboti", introducido al público por el escritor checo, Karel Čapek, en su obra "Rossumovi univerzální roboti" (Robots Universales Rossum) la cual comienza en una fábrica en donde se construían "personas" artificiales llamadas robots. La palabra "roboti" fue inventada por su hermano Josef Čapek a partir de la palabra checa "robota" que significa trabajo.

1.4 Clasificación de los Robots

Por la versatilidad de su estructura, funciones y componentes, es difícil clasificar a los robots, pero para facilitar su análisis se presentará la siguiente clasificación:

1.4.1 Robots humanoides

Los robots humanoides son robots antropomorfos que además de imitar la apariencia humana, intentan imitar cualquier aspecto que defina a los seres humanos hasta el punto de confundirse o integrarse a ellos sin que la contraparte humana note la diferencia. En la Figura 1.5 se puede ver al robot ASIMO de la marca Honda como un ejemplo de robot humanoide.

Figura 1.5 Robot humanoide "ASIMO" de Honda.

1.4.2 Robots Móviles

Los robots móviles son los que tienen la habilidad de moverse alrededor de su entorno y no están fijos a un punto en concreto. Pueden desplazarse por medio de extremidades robóticas, ruedas o por cualquier otro tipo de dispositivo. Por ahora, muchos de estos robots tienen que guiarse a través de sensores que siguen líneas de un color en específico o con radares que les muestran el camino por donde van, pero en algunos años podrán percibir mejor su entorno gracias al avance de los sistemas de visión. Un ejemplo claro de estos robots, es el robot acuático 1KA Seaglider de la empresa iRobot (ver la Figura 1.6), que realiza un monitoreo autónomo en el golfo de México para colectar datos acerca de la situación del desastre ocasionado por la explosión de la plataforma petrolera Deepwater Horizon subarrendada a la empresa British Petroleum en abril del 2010. (4).

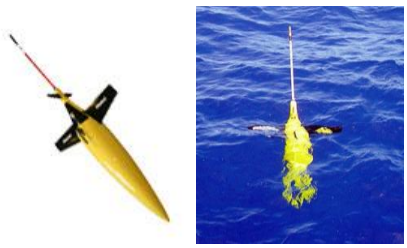


Figura 1.6 Robot acuático 1KA Seaglider.

1.4.3 Robots de servicio

Los robots de servicio son aquéllos que nos ayudan en la vida cotidiana, en nuestros hogares u oficinas, los que limpian por nosotros como la aspiradora automática Roomba (4) o el pequeño cortador de césped Automower Solar Hybrid (5), ambos mostrados en la Figura 1.7.
La Federación Internacional de Robótica (IFR, por sus siglas en inglés) ha propuesto una definición tentativa para el concepto de robot de servicio: “Un Robot de Servicio es un robot que opera semi o totalmente autónomo para realizar servicios útiles para el bienestar de las personas y equipos, con exclusión de las operaciones de fabricación”.

Figura 1.7 Robots de servicio, aspiradora Roomba y cortador de césped Automower.

1.4.4 Robots industriales

Como su nombre lo dice son aquellos que se desempeñan en las industrias y comúnmente poseen un brazo robótico con un efector final como una pinza (Gripper en inglés) para poder realizar una o varias actividades. (ver la Figura 1.8)

La Organización Internacional para la Estandarización (ISO, por sus siglas en inglés) define a un robot industrial como "un aparato manipulador programable en tres o más ejes, controlado automáticamente, reprogramable y de propósitos múltiples, el cual puede estar en un lugar fijo o móvil, para su uso en aplicaciones de automatización industrial" (6).

Figura 1.8 Robot "PUMA" de la empresa Unimation, utilizado ampliamente en el ensamblaje de autos en la década de los 80.

Por su tipo de estructura se puede subclasificar a los robots industriales en:

- **Robot Cartesiano**: Utilizado para trabajos de colocación de piezas, aplicación de sellador, operaciones de manufactura, manejo de herramientas y soldadura de arco. Es un robot que tiene 2 o más articulaciones prismáticas y sus ejes coinciden con el plano cartesiano.

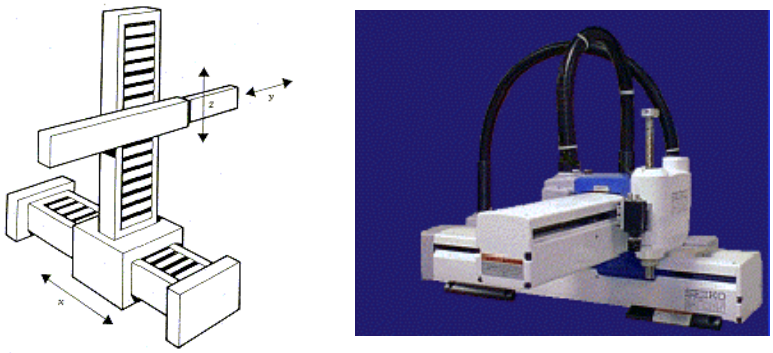


Figura 1.9 Robot Cartesiano

• **Robot Cilíndrico**: Usado en operaciones de montaje, soldadura, y fundición. Es un robot en que sus ejes forman un sistema coordinado cilíndrico.

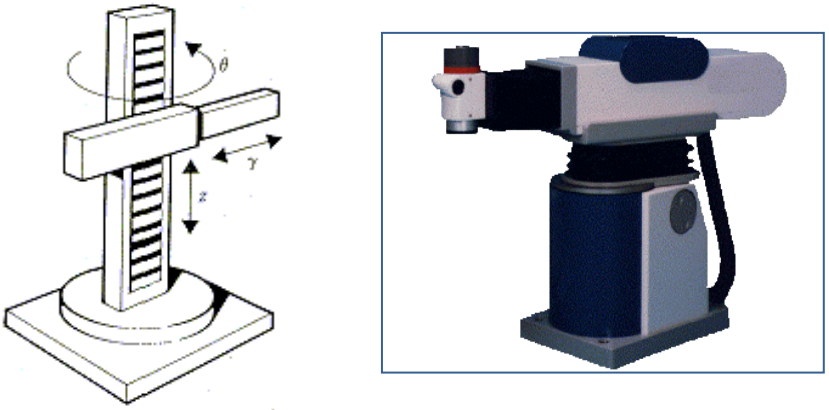


Figura 1.10 Robot Cilíndrico

• **Robot Esférico o Polar**: Usado en manejo de máquinaria, soldadura y fundición. A diferencia del robot anterior, sus ejes forman un sistema coordenado polar.

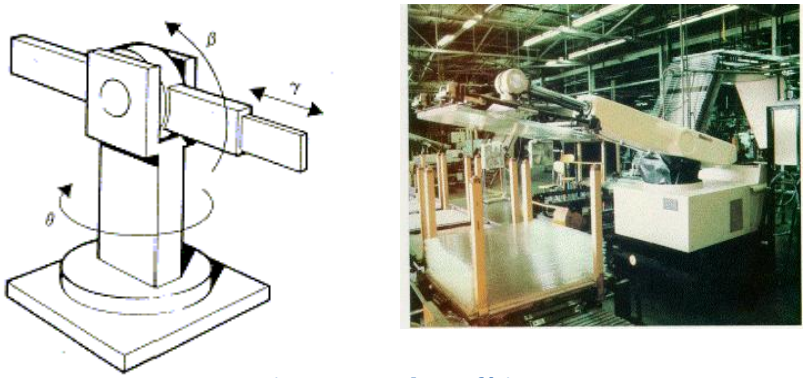


Figura 1.11 Robot Esférico

• **Robot SCARA**: Utilizado en labores de colocación de piezas, aplicación de selladores, operaciones de montaje y manejo de máquinaria. Es un robot que tiene dos articulaciones rotativas y paralelas.

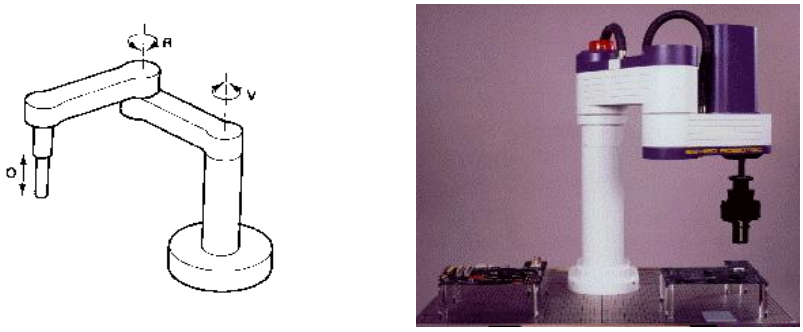


Figura 1.12 Robot SCARA

• **Robot Articulado**: Usado en montajes, fundición, soldadura, pintura. Es un robot compuesto por un brazo que tiene por lo menos tres articulaciones rotativas.

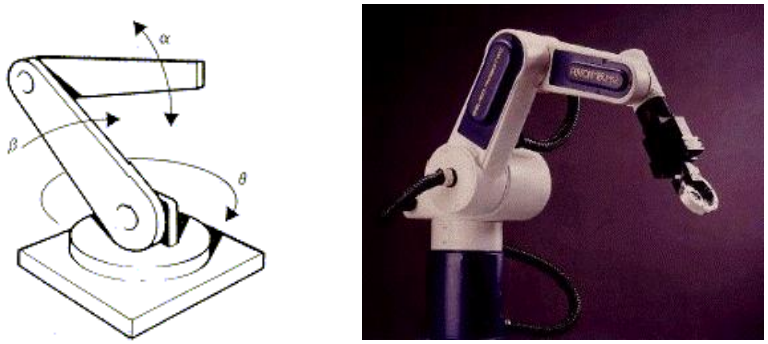


Figura 1.13 Robot Articulado

En conclusión, se puede ver que nuestro robot virtual o mesa de coordenadas virtual pertenece en teoría a los robots industriales.

1.5 Mesas de Coordenadas

Una mesa de coordenadas es sencillamente lo que se conoce como robot cartesiano y al igual que éstos, tiene una estructura morfológica muy sencilla. Sobresalen por sus altas prestaciones en cuanto a velocidad, precisión y exactitud, sin embargo, sus aplicaciones prácticas, requieren que los entornos de trabajo no sean excesivamente complejos, es decir, ambientes con pocos obstáculos. Tanto la sencillez de su construcción, como su control, son sus ventajas más destacadas, siendo idóneos en determinadas aplicaciones como la alimentación de máquinas herramientas o la creación de Tarjetas de Circuitos Impresos (PCB, por sus siglas en inglés). Claro está, que sus aplicaciones son, incontables.

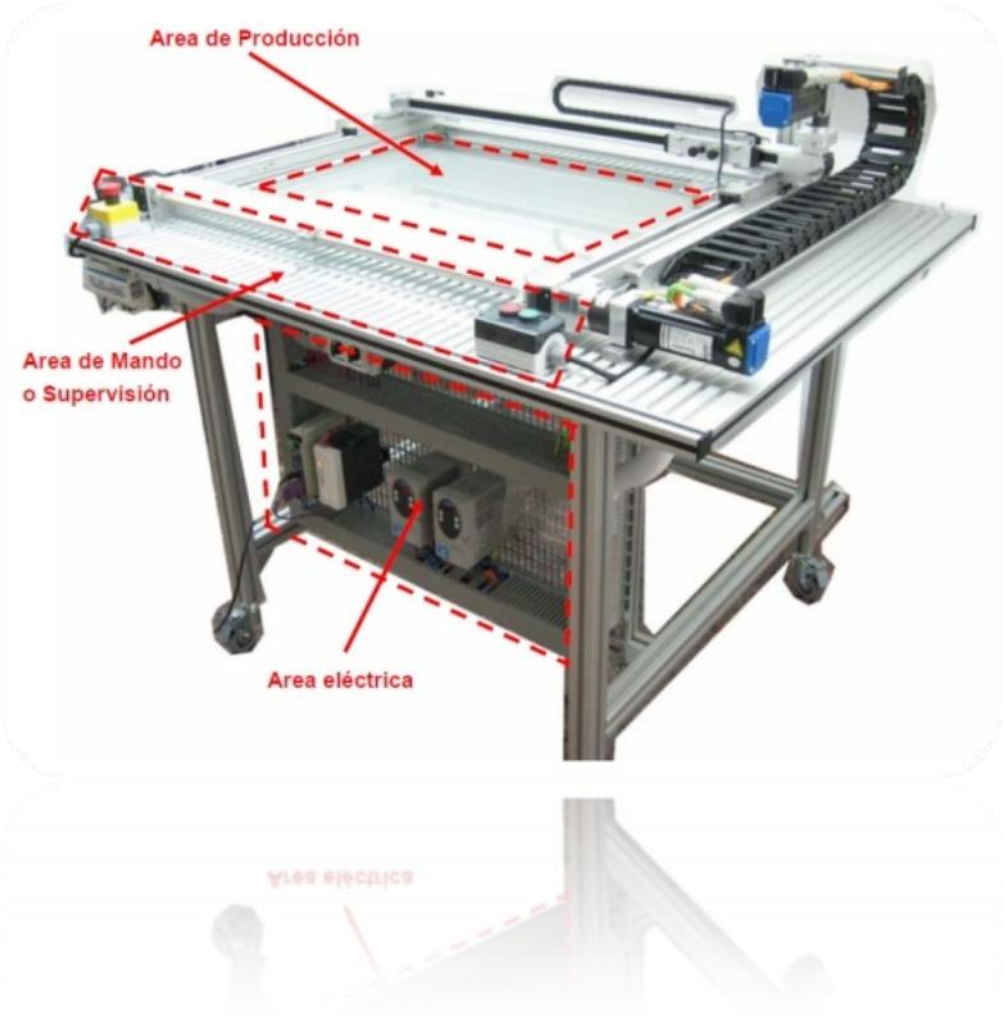


Figura 1.14 Mesa de coordenadas diseñada por la empresa Schneider Electric.

1.6 Prototipos Virtuales

Como ya se mencionó anteriormente, el paso anterior a construir una máquina, ya sea un auto, una lavadora o una computadora, es construir un prototipo. Pero, ¿qué es un prototipo?, bueno, pues no es otra cosa más que un borrador físico, una máquina hecha con distintos materiales y dispositivos, desde los que están a nuestro alcance o que nos sobraron de otras máquinas hasta los que son comprados o diseñados y construidos individualmente para usarse en nuestra máquina.

El prototipo se realiza para corroborar que la máquina diseñada en la computadora funciona en la vida real y realizar correcciones y mejoras en el diseño final. Pero muchas veces el prototipo no es lo que se

espera o no es funcional, por lo que se debe arreglarlo o muchas veces construirlo de nuevo, lo que implica pérdida de tiempo y dinero.

Este problema se afronta de diferentes formas, por ejemplo, existen máquinas de prototipos rápidos que forman una pieza o modelo físico a partir de un modelo virtual, utilizando diversos materiales como plástico y metal para su construcción. Estás máquinas utilizan sistemas láser o estéreo-litografía para ir dando forma a las piezas o modelos. Pero la limitante es que estas máquinas son costosas y los modelos son pequeños o a escala, además de que muchas piezas móviles no pueden construirse. En la **Figura 1.15** se puede ver una máquina de prototipos rápidos por láser.

Figura 1.15 Impresora 3D para prototipado rápido 3D In Vision HR

Afortunadamente hoy en día la tecnología nos permite manipular y hasta programar un prototipo virtual, lo que implica que se puede poner a prueba un modelo hecho por computadora para analizar su comportamiento y posibles errores como si el prototipo estuviera construido físicamente, con la gran ventaja de que los efectos de las leyes de la física pueden analizarse de manera aislada o en conjunto. El software de CAD está en constante actualización, por lo que, cada vez hay más pruebas que se pueden aplicar a nuestros modelos, desde aplicar gravedad hasta esfuerzos y temperaturas, con lo que el diseño virtual se acerca cada vez más a la realidad.

Aunque el software CAD ha existido desde hace décadas, ahora no sólo se aplican pruebas a nuestros modelos, sino que además se puede moverlos y programarlos a nuestro gusto, como si fuera un modelo

físico real. Gracias a los avances en el software de programación, se pueden transmitir instrucciones para controlar el modelo realizado en el software CAD, es decir, las máquinas virtuales ahora también son inteligentes.

Y no todo acaba ahí, también es posible "combinar" lo virtual con la realidad y poner "los ojos", "los oídos", etc. del modelo virtual en cualquier tipo de cámara o sensor, para que perciba el entorno al que está expuesto y realice alguna acción a partir de ello. Evidentemente, todo es posible, sólo depende de nuestra imaginación.

En la Figura 1.16 se puede ver un modelo virtual y un poco de su programación gráfica.

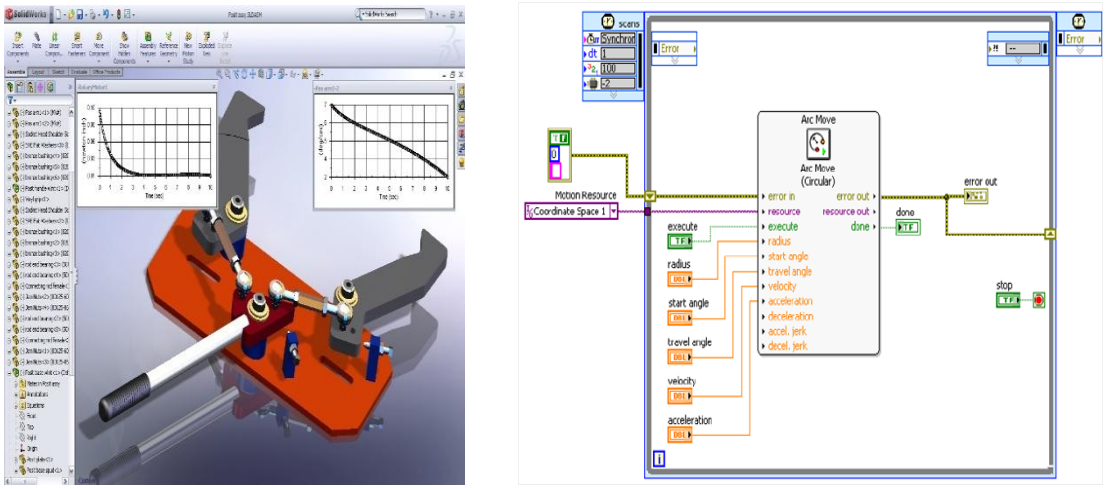


Figura 1.16 Prototipo virtual realizado con SolidWorks y su programación gráfica en *LabVIEW*.

1.7 Sistemas de Visión

Los sistemas de visión utilizan cámaras para detectar objetos y realizar alguna acción a partir de ello, por ejemplo, medir una distancia o comprobar que una botella tenga el líquido requerido dentro de ella. Los sistemas de visión son para las máquinas, lo que los ojos son para los seres vivos.

Los sistemas de visión miden sin la necesidad de contacto físico. Esto es posible con una cámara y un lente que nos entrega una imagen a escala. Teniendo esta imagen acondicionada en una computadora y desarrollando algoritmos de imágenes, se puede trabajar el objeto como si lo dimensionáramos físicamente. Este proceso se logra haciendo una comparación de pixeles que contiene la imagen tomada, con las dimensiones de un objeto conocido a una distancia fija. En la Figura 1.17 se puede observar un típico sistema de visión.

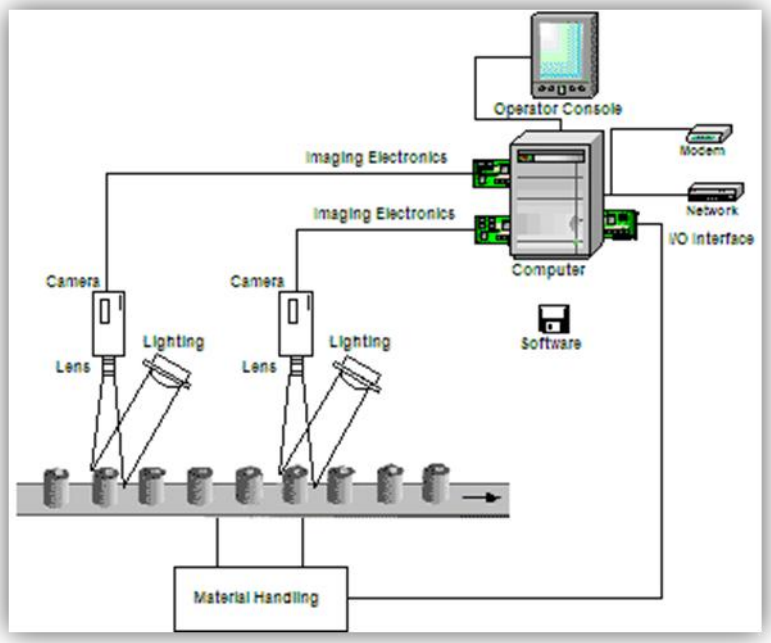


Figura 1.17 Un típico sistema de visión.

La forma tradicional de medir cualquier objeto involucra contacto físico entre éste y el instrumento con el que se efectúa la medición. Existe otra forma de efectuar esta acción sin necesidad de tocar el

objeto, lo que se conoce como “medición sin contacto” ó “medición a distancia”.

Todo esto requiere métodos de prueba sofisticados como, por ejemplo, el que brinda el procesamiento industrial de imágenes, tecnología que avanza en forma continua para convertirse en un componente integral de las soluciones de automatización, desarrollándose cada vez más como disciplina estándar para el control de calidad.

1.8 Software de Diseño Asistido por Computadora

El software de Diseño Asistido por Computadora es usado ampliamente por ingenieros, arquitectos y otros profesionales para dibujar en 2 dimensiones (2D) o moldear en 3 dimensiones (3D) desde piezas o herramientas simples hasta ensambles complejos de máquinaria o casas en el caso de los arquitectos. El software en 2D se basa en entidades geométricas vectoriales y se añaden superficies y sólidos para el 3D.

Existen varias marcas de software CAD en el mercado, como por ejemplo, SolidEdge, AutoCAD, *SolidWorks* y AutoDesk Inventor, sin embargo durante el desarrollo de este proyecto se utilizaron únicamente 2 paquetes de software CAD, SolidEdge y *SolidWorks*.

1.9 Software de programación gráfica

El software de programación gráfica como su nombre lo dice utiliza un entorno gráfico para asociar instrucciones, variables, archivos, objetos y subprogramas de manera simple valiéndose de expresiones visuales, arreglos espaciales de texto y símbolos gráficos. Entre los paquetes más comunes de programación gráfica se encuentran *NI LabVIEW*, Simulink y Agilant VEE.

Para este proyecto se utilizó el programa *LabVIEW* de la empresa *National Instruments*, porque permite la comunicación con *SolidWorks* a través de su herramienta *NI SoftMotion*.

1.10 Aplicaciones

Las aplicaciones de un modelo virtual son bastas, sin embargo, hoy en día la industria que más usa los prototipos virtuales es la de automoción, en donde las grandes empresas de automóviles ahorran millones de dólares al crear sus prototipos y ponerlos a prueba tanto física como estéticamente en un entorno completamente virtual.

En la propia Facultad de Ingeniería se han realizado ya algunos prototipos virtuales. Es el caso de la escultura transformable "*Brancusi*" diseñada primeramente por el escultor Enrique Carbajal González mejor conocido como *Sebastián* y rediseñada mecatrónicamente por el equipo de la sala de proyectos liderada por el ingeniero Serafín Castañeda. En donde el prototipo virtual fue sometido a varias pruebas de esfuerzo y deformación con el software Inventor, *SolidWorks* y SolidEdge.

Ahora el proyecto presentado en esta tesis intenta ser un hito en cuanto a prototipos virtuales se refiere, ya que se integran también la programación y los sistemas de visión, combinando así tecnologías que ayudarán a la rapidez en la realización de proyectos y al ahorro de recursos.

Algún día este proyecto podrá servir en el Laboratorio de Automatización, Instrumentación y Control Avanzado (LAICA) para que los alumnos de la Facultad de Ingeniería de la UNAM puedan programar esta mesa de coordenadas virtual y después ellos puedan diseñar y realizar sus propios prototipos virtuales, ahorrándoles tiempo y dinero en la entrega de sus proyectos.

CAPÍTULO 2

Diseño de la mesa de coordenadas

2.1 Metodología

Para diseñar la mesa de coordenadas (o MC por sus siglas en español) se utilizó el método de diseño de Pahl y Beitz utilizado ampliamente en la Sala de Proyectos del Centro de Diseño Mecánico e Innovación Tecnológica en donde se llevaron a cabo los siguientes pasos.

2.1.1 Estudio Previo.

El estudio previo es una pequeña investigación de la institución para la cual se está trabajando, es decir, el cliente. Este primer paso se realiza para conocer más a fondo al cliente y principalmente conocer los detalles de su actividad y su necesidad.

Antes de diseñar la mesa de coordenadas se realizó un estudio previo para conocer las necesidades del cliente que en este caso es el Laboratorio de Automatización, Instrumentación y Control Avanzado (LAICA) que inicialmente quería construir una mesa de coordenadas física para que transportara engranes desde un pequeño almacén, esto serviría para que los alumnos aprendieran el uso de servomotores, así como el control y programación de la mesa. El estudio previo que se hizo en ese puede verse en la Tabla 1.

EL CLIENTE	**Nombre o razón social:** División de Ingeniería Mecánica e Industrial **Actividad: Impartir y coordinar académica y administrativamente las carreras de ingeniería mecánica, ingeniería industrial e ingeniería mecatrónica.** **Antigüedad/experiencia:** 30 Años **Dirección: Circuito exterior s/n Cd. Universitaria, delegación Coyoacán CP. 04510 México D.F.** **Teléfono: Tel. 52 (55) 56-22-31-00, 01 y 02** **Contacto:** No se menciona
EL PRODUCTO	**Descripción: Engranes rectos hechos de acero de 22 cm de diámetro, un paso diametral de 20 y un módulo de 40.** **Dirigido a: Industrias** **Tamaño del mercado: Nacional** **Ventas estimadas: No se menciona**
EL PROCESO	**Antes de proceder al mecanizado de los dientes, los engranes han pasado por otras máquinas como son torno y fresadora donde se les han mecanizado todas sus dimensiones exteriores y agujeros si los tienen, dejando los excedentes necesarios en caso de que tengan que recibir tratamiento térmico y posterior mecanizado de alguna de sus zonas. El mecanizado de los dientes se realiza en máquinas talladoras construidas especialmente para este fin, llamadas fresas madres.**
PRIMER CONTACTO	**Medio: Personal** **Fecha: 11 de Febrero de 2009** **Contacto: Ing. Humberto Mancilla Alonso**
LA NECESIDAD	**Descripción/contexto: Mover las piezas de las fresas madres a un almacén y acomodarlas de forma rápida y precisa para su posterior empaquetamiento y venta.** **Las ideas del cliente: No se menciona** **Identificación: Diseño y automatización**
DIAGNÓSTICO:	**Descripción: Se necesita una máquina mecatrónica programable que enganche los engranes que salgan de las fresas madre y los coloque en el almacén cada cierto tiempo.** **Tratamiento: Una mesa de coordenadas programable solucionaría el problema, ya que realizaría las tareas necesarias para mover los engranes de las fresas madre al almacén de una manera rápida y precisa.**

Tabla 1. Estudio previo realizado a la DIMEI

A partir del estudio previo anterior se comenzó a trabajar en el diseño de la mesa de coordenadas.

2.1.2 Diagrama de funciones

Se buscaron todas las funciones que debe de tener la MC como: el movimiento en el eje “X”, el movimiento en el eje “Y”, el movimiento en el eje “Z”, dibujar, transmisión, comunicación, etc. y se trazaron en un diagrama para poder separar todo el problema y así darle una mejor solución.

2.1.3 Configuración

Una vez que el problema estaba dividido en funciones, se comenzaron a dar ideas de posibles configuraciones de la máquina, dónde puede estar cada elemento, de qué forma consumiría menos energía, cómo podría ser más compacto, etc. Todo esto nos llevó a realizar varios bosquejos, algunos de ellos se pueden ver en las siguientes figuras.

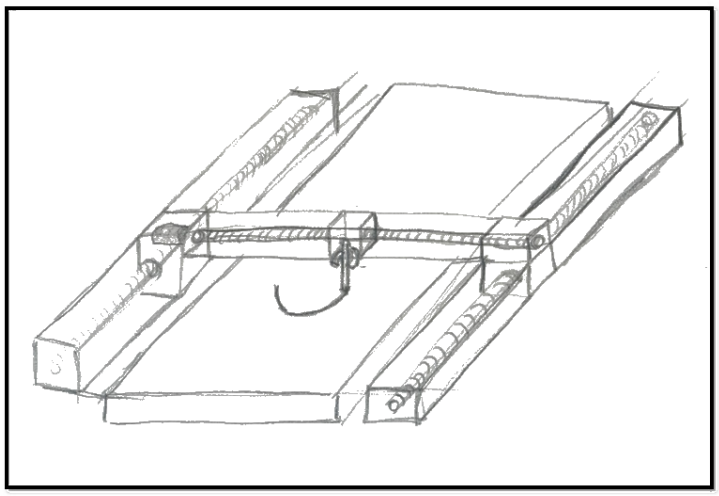


Figura 2.1 Bosquejo de configuración 1.

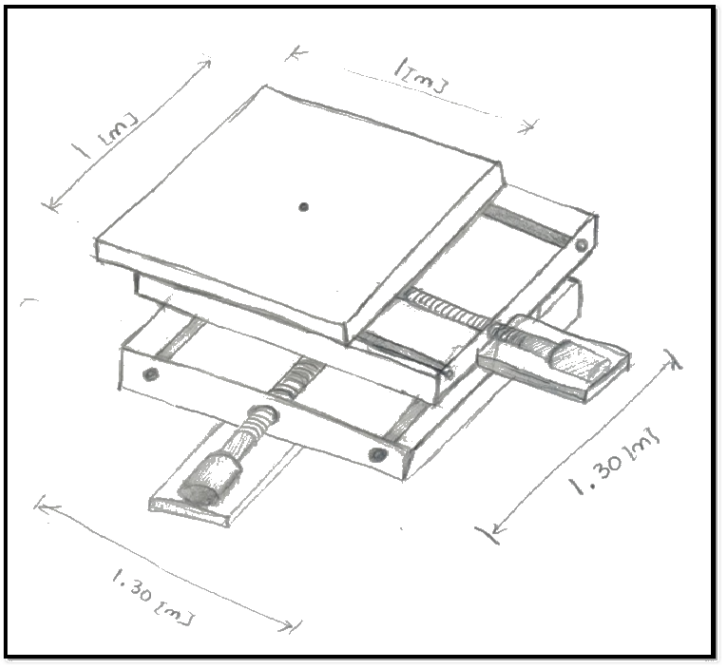


Figura 2.2 Bosquejo de configuración 2.

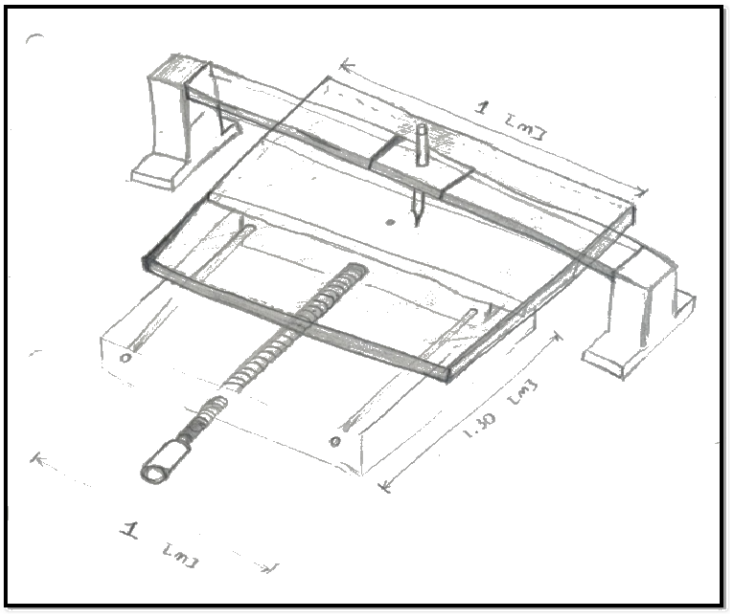


Figura 2.3 Bosquejo de configuración 3.

2.1.4 Composición

Se escogió la configuración número dos que aparece en la Figura 2.2 y se procedió a usar un software de CAD para hacer la composición y empezar realmente a diseñar las partes que conformarán la MC. Durante el proceso de dibujado en el software se modificó la

configuración para que varias de estas partes fueran más fáciles de manufacturar o en su defecto comprar, en caso de que se quisiera construir (como lo requería originalmente la DIMEI, ver tabla 1).

El software de CAD elegido inicialmente fue Solid Edge ST debido a la experiencia con él pero a lo largo del proyecto se decidió migrar toda la configuración a *SolidWorks* ya que es el software que permite la comunicación con *LabVIEW*.

Se realizaron algunas composiciones para poder observar la interacción con los elementos y las posibles medidas y modo de funcionamiento de la MC.

En la Figura 2.4 puede apreciarse la primera composición.

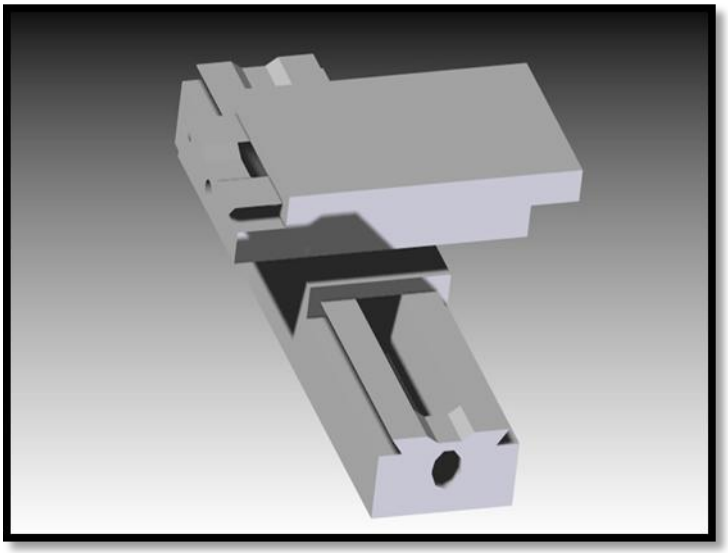


Figura 2.4 Primera composición realizada con el software Solid Edge ST.

2.1.5 Diseño de detalle

Después de haber realizado varias composiciones se procedió a realizar el diseño de detalle, con medidas exactas y bien planeadas para el correcto funcionamiento de la MC.

Se realizaron diversas correcciones al diseño, se implementaron nuevos dispositivos y partes. En la Figura 2.5 se ve el diseño implementando un arreglo de poleas para transmitir el movimiento del motor al tornillo sinfín.

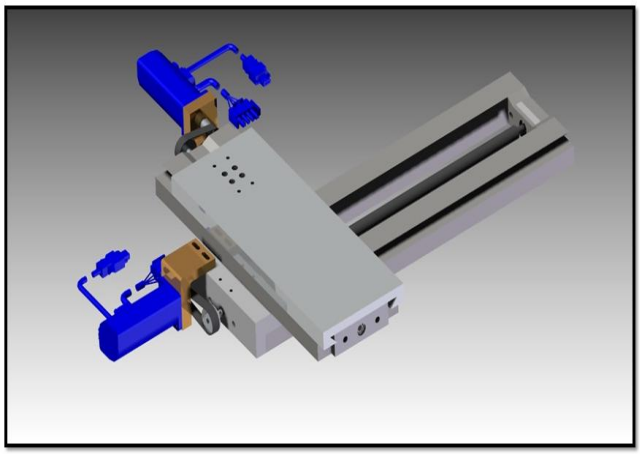


Figura 2.5 Composición número 2 con un arreglo de poleas para transmitir el movimiento.

Este arreglo de poleas fue desechado porque no ofrecía la potencia necesaria al tornillo sinfín para poder mover los soportes.

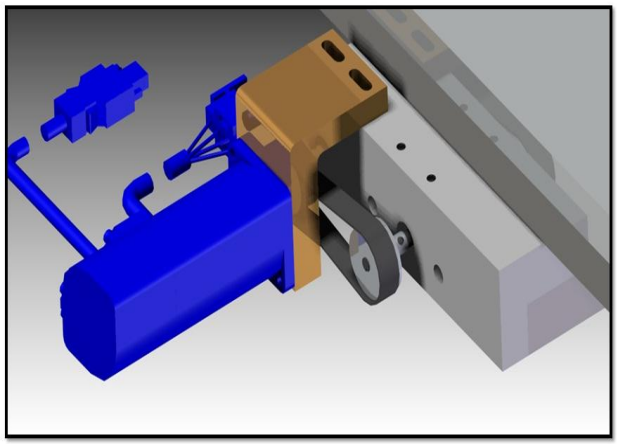


Figura 2.6 Arreglo de poleas.

En la Figura 2.7 se puede ver corregido el problema de la transmisión de potencia, gracias al acoplamiento de un reductor tipo Gearhead de la marca *Shimpo Drives*. (7)

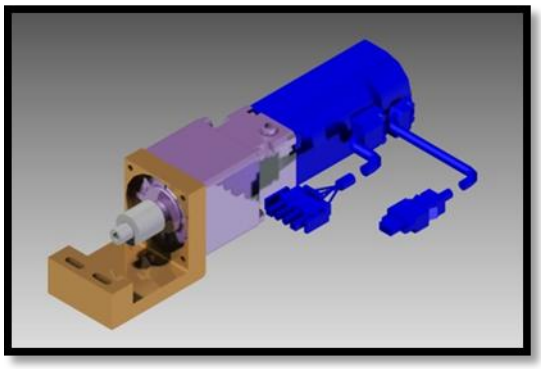


Figura 2.7 Reductor *Shimpo Drives* unido al servomotor Yaskawa.

La idea original era que la mesa de coordenadas tuviera un lápiz como efector final, es decir, que tenía que ser capaz de dibujar, y tenía que hacerlo en una superficie del tamaño de una hoja tamaño carta, 21.59 cm x 27.94 cm. Por lo que se implementó un simple y pequeño sistema a base de un solenoide para que el lápiz subiera y bajara con una sencilla instrucción, se puede ver en la Figura 2.8.

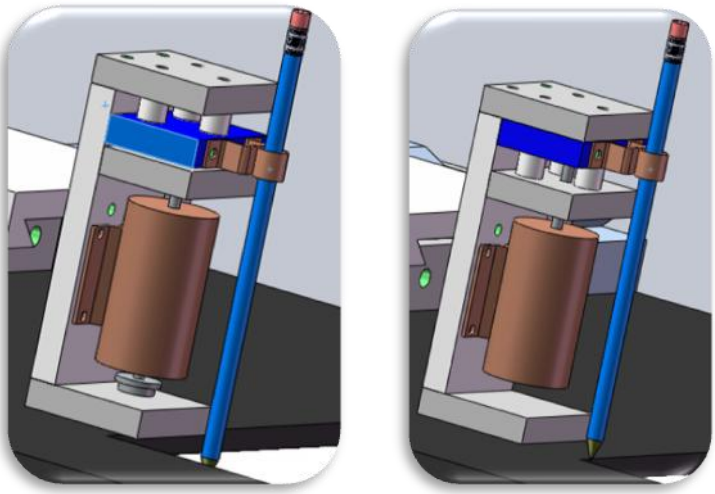


Figura 2.8 A la izquierda el solenoide en su posición natural (abajo) y a la derecha en su posición con circuito cerrado (arriba).

Para este momento la mesa de coordenadas se veía así:

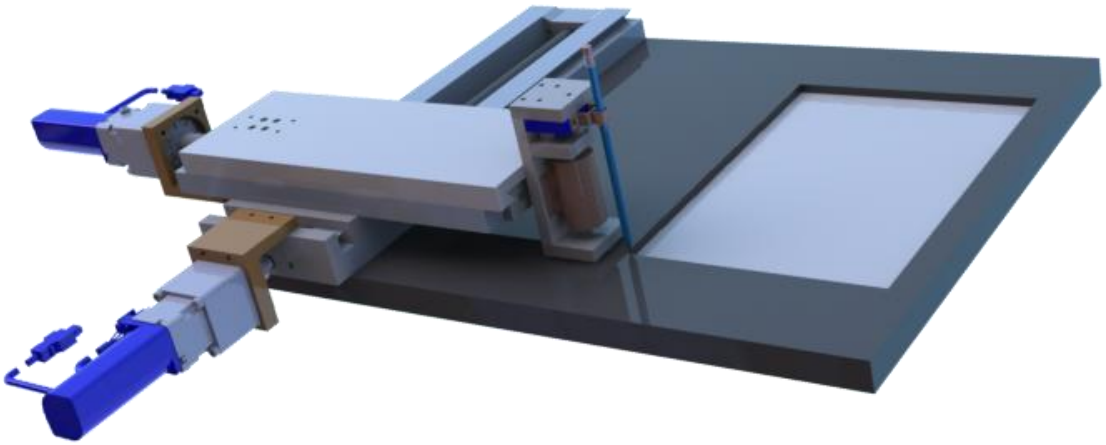


Figura 2.9 *Render* de la mesa de coordenadas hecho con PhotoView 360 de *SolidWorks*.

En la última composición realizada se decidió hacer los ejes aún más ligeros y transportables por lo que la mesa sufrió grandes cambios, debido a que se decidió no construir la mesa por falta de recursos. Así que el proyecto tomó un giro inesperado y se decidió realizar un prototipo virtual que sólo funcionara como ejercicio de programación. Así se ve el último diseño de detalle de la mesa de coordenadas.

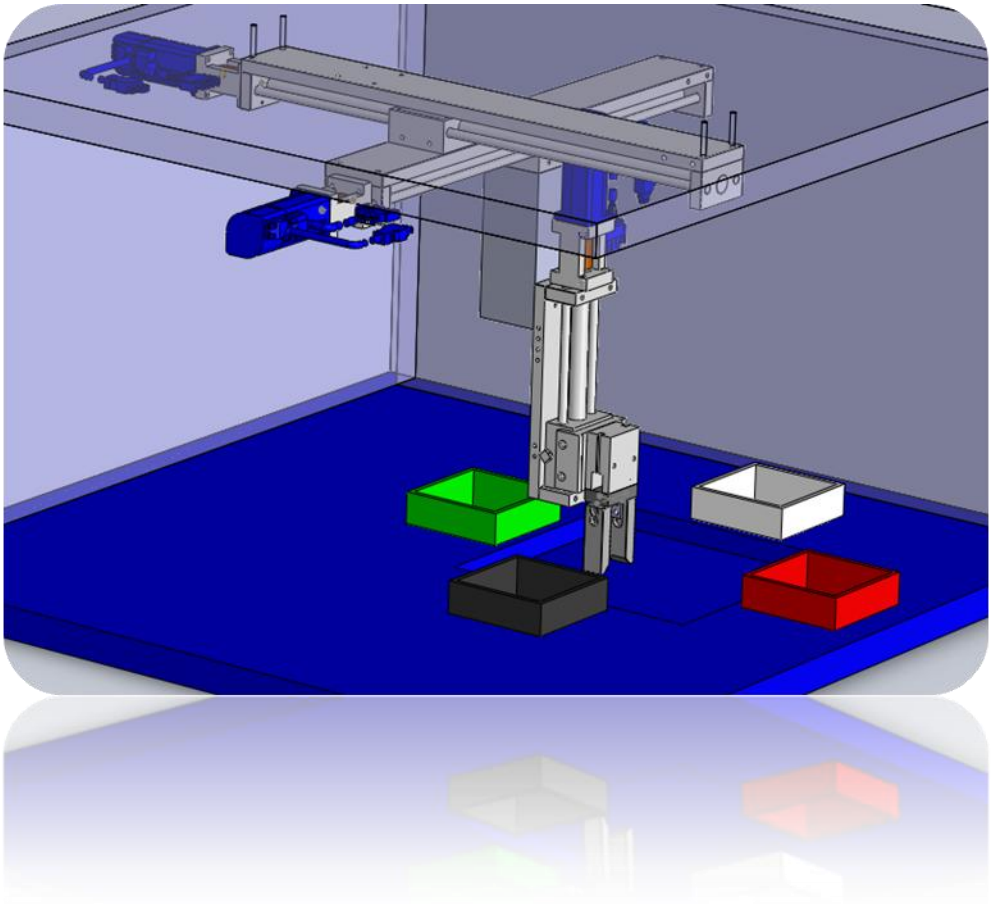


Figura 2.10 Diseño final de la Mesa de Coordenadas.

Como puede apreciarse en la Figura 2.11, los reductores (Figura 2.7) fueron desechados y se agregó un gripper o pinza para manipulación de objetos (Figura 2.11). La mesa de coordenadas ahora puede clasificar objetos según su color, se le agregaron cuatro cajas de cuatro colores distintos para que clasifique fichas y las ordene en su caja correspondiente.

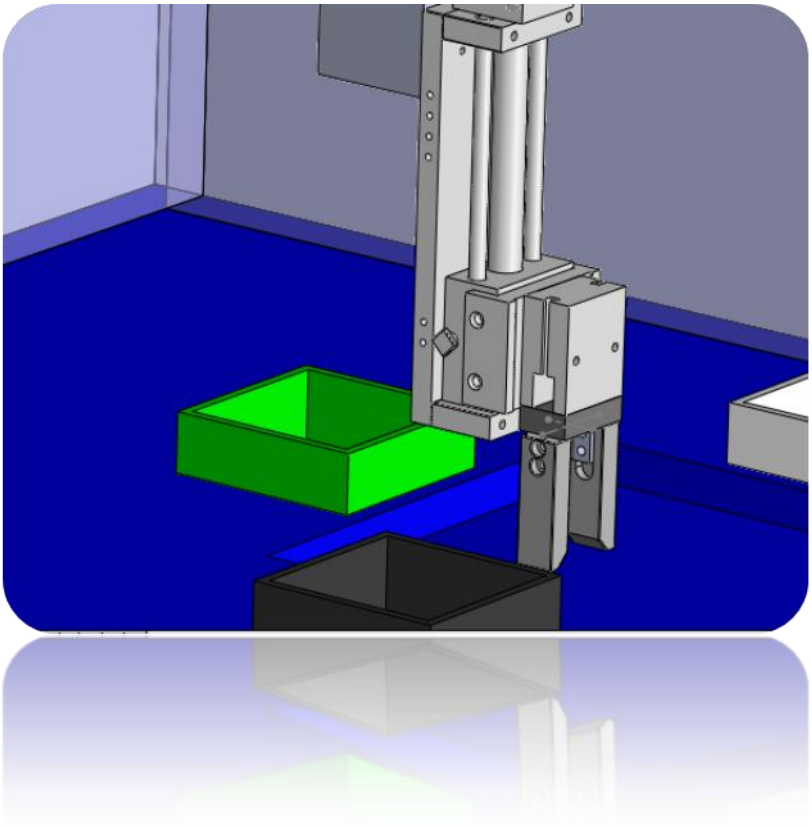


Figura 2.11 Se agregó un gripper o pinza a la mesa de coordenadas.

CAPÍTULO 3

Instalación y preparación del software

Como ya se mencionó la mesa de coordenadas pasó a ser un prototipo virtual por falta de recursos para su construcción, así que se decidió programar y probar la mesa de coordenadas sólo con software, por lo que se tuvo que implementar un módulo llamado *SoftMotion* de la empresa *National Instruments.* Éste módulo permite la comunicación con *SolidWorks* a través de *LabVIEW*. En este capítulo se dará la explicación completa de cómo se instala este modulo además del software adicional necesario, dónde conseguirlo y cómo usarlo a manera de manual para que los lectores de esta tesis puedan usarla de manera práctica.

3.1 Dónde conseguir el software

1. *SolidWorks* 2010. Para este proyecto se usó la versión 2010 del conocido software de la empresa *Dassault Systèmes SolidWorks* Corp. Se puede comprar una versión académica a un bajo costo a través de la siguiente página:

   http://www.*SolidWorks*.com/sw/products/student-software-3d-mcad.htm

   Después de comprar es necesario enviar una copia de credencial de estudiante y una tira de materias del semestre o año en curso.

2. *LabVIEW* 2009. Para este proyecto se usó la versión 2009 del conocido software de la empresa *National Instruments*, en la siguiente página puede conseguirse una prueba de 30 días de la última versión del software:

   http://www.ni.com/try*LabVIEW/*

3. *LabVIEW NI SoftMotion* Module 2009 SP1. Para este proyecto se usó la versión 2009 SP1 del módulo *SoftMotion* para *LabVIEW*. En la siguiente página puede conseguirse gratuitamente y ahí mismo esta la explicación de cómo activarlo si no se sabe cómo hacerlo:

   http://joule.ni.com/nidu/cds/view/p/id/1509/lang/en

4. *Vision Builder* for Automated Inspection 2009 SP1. Para este proyecto se usó la versión 2009 SP1 del conocido software de la empresa *National Instruments*. En la siguiente página se puede comprar una licencia académica o bajarse una prueba de 30 días:

   http://www.ni.com/vision/software/

3.2 Recomendaciones al instalar el software

El software puede instalarse en cualquier orden y sólo hay que seguir las instrucciones en pantalla de cada uno de los gestores de instalación, salvo una excepción al instalar *SolidWorks*:

En la última pantalla de instalación en la parte de "Productos" se debe dar clic en "Cambiar" (ver la Figura 3.1).

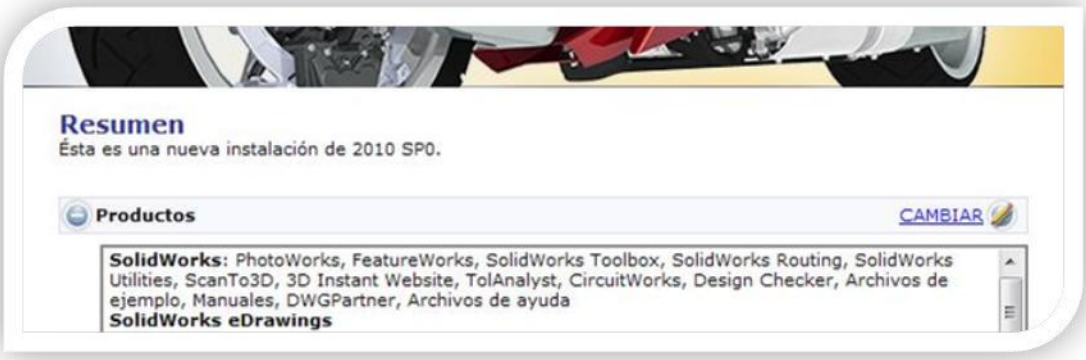


Figura 3.1 Resumen de la instalación de *SolidWorks* 2010

Una vez en la pantalla de “Selección de Productos” hay que activar la casilla de “COSMOSM” y asegurarse que la casilla de “*SolidWorks* Simulation” esté activada (ver la Figura 3.2).

Figura 3.2 Selección de Productos de la instalación de *SolidWorks* 2010.

COSMOSM es una interfaz autónoma de Análisis de Elementos Finitos (FEA, por sus siglas en inglés) con solvers de simulación que permite el uso de Motion en *SolidWorks*.

Se da clic en “regresar a resumen” y luego en “instalar ahora”.

Una vez que se han instalado todos los programas y componentes necesarios se procederá a configurar el software para poder lograr comunicación entre *LabVIEW* y *SolidWorks* a través de *SoftMotion* de NI con unos sencillos pasos:

1. Abrimos *SolidWorks*
2. Se da clic en el menú emergente “Herramientas”
3. Se da clic en “Complementos”, esto abrirá una ventana (Ver Figura 3.3)
4. Se activan las casillas *SolidWorks* Motion y *SolidWorks* Simulation, tanto de la izquierda como de la derecha. La casilla derecha se activa para

que cada vez que se abra *SolidWorks*, se inicien automáticamente estos complementos.

5. Se da clic en "Aceptar"

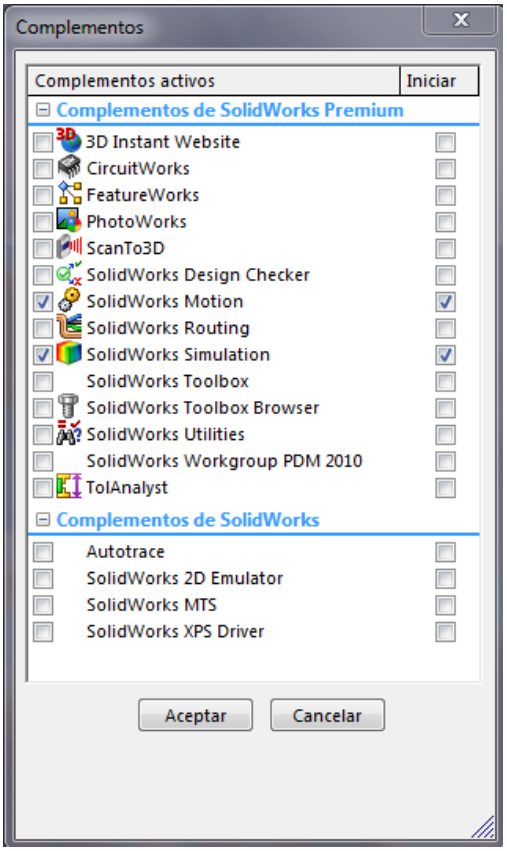


Figura 3.3 Menú de Complementos en *SolidWorks*.

3.3 Preparación del ensamble para la simulación

Una vez instalado y configurado el software en la computadora se procederá a programar y mover el ensamble, pero antes se tiene que prepararlo o ajustarlo en *SolidWorks*. Es importante mencionar que el ensamble pudo haberse diseñado en cualquier software de CAD y luego exportarse a *SolidWorks* pero pueden haber pequeños desajustes, es recomendable crear el ensamble en *SolidWorks* desde un principio para evitar contratiempos.

Recomendaciones

Para que el ensamble funcione perfectamente con motion y *LabVIEW* hay que seguir sólo 3 recomendaciones:

1. Definir todas las relaciones de posición y revisar que estén correctas.
2. Definir todas las relaciones mecánicas necesarias y revisar que estén correctas (si se desconoce cómo definir las relaciones mecánicas en *SolidWorks* favor de visitar este tutorial http://zone.ni.com/wv/app/doc/p/id/wv-1417/nextonly/y en inglés)
3. Para evitarse problemas de movimiento y conflictos con las relaciones de posición y mecánicas es mejor realizar el ensamble sin subensambles, es decir, empezar desde cero colocando pieza por pieza hasta terminar todo el ensamble en un solo archivo.

Relaciones mecánicas.

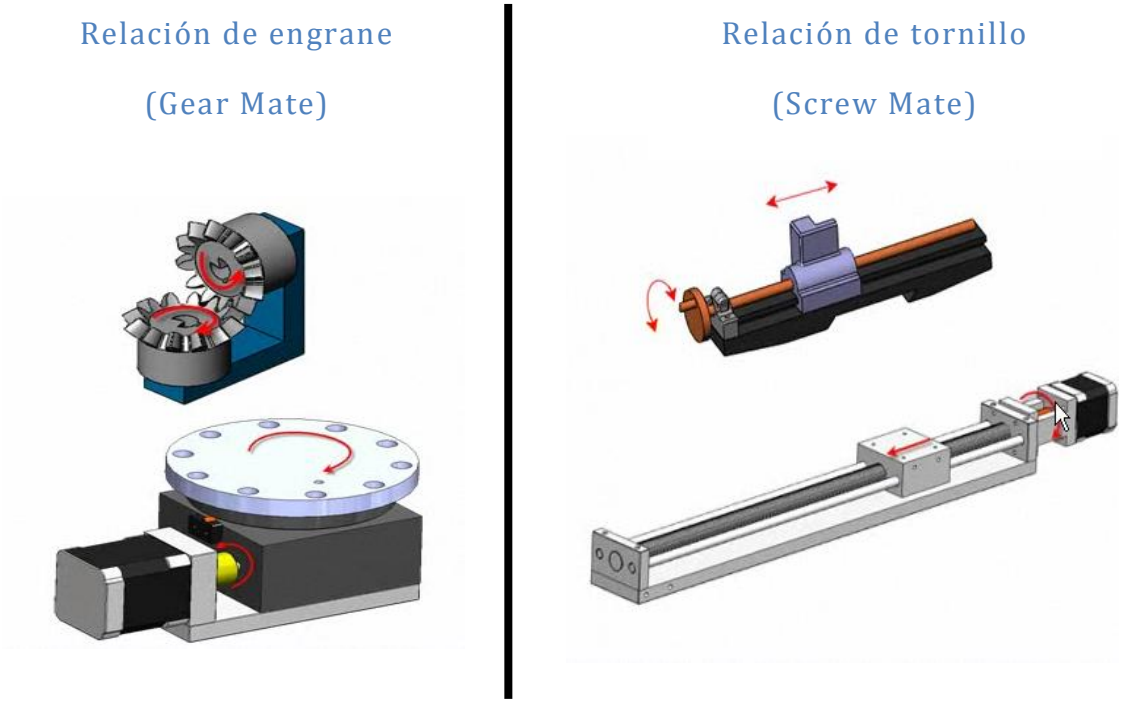


Figura 3.4 Relaciones mecánicas en un ensamble de CAD.

3.3.1 Estudio de movimiento

Un “estudio de movimiento” es una simulación gráfica de movimiento para modelos de ensamblaje. Se puede incorporar en un estudio de movimiento propiedades visuales, como iluminación y perspectiva de cámara. Los estudios de movimiento no modifican un modelo de ensamblaje ni sus propiedades sino que simulan y animan el movimiento prescrito para un modelo. (8)

Se pueden utilizar relaciones de posición de *SolidWorks* para restringir el movimiento de componentes en un ensamblaje al modelar movimiento.

Después de que el ensamble está listo y con todas las relaciones mecánicas y de posición definidas se procederá a programar el estudio de movimiento.

En la parte inferior de la pantalla dentro de *SolidWorks* se encuentran dos pestañas, una que dice “Modelo” (que es el área de trabajo donde se realizó el ensamble) y otra pestaña que dice “Estudio de Movimiento”. Se da clic en esta pestaña y aparecerán una serie de herramientas y una línea de tiempo (Ver la Figura 3.5).

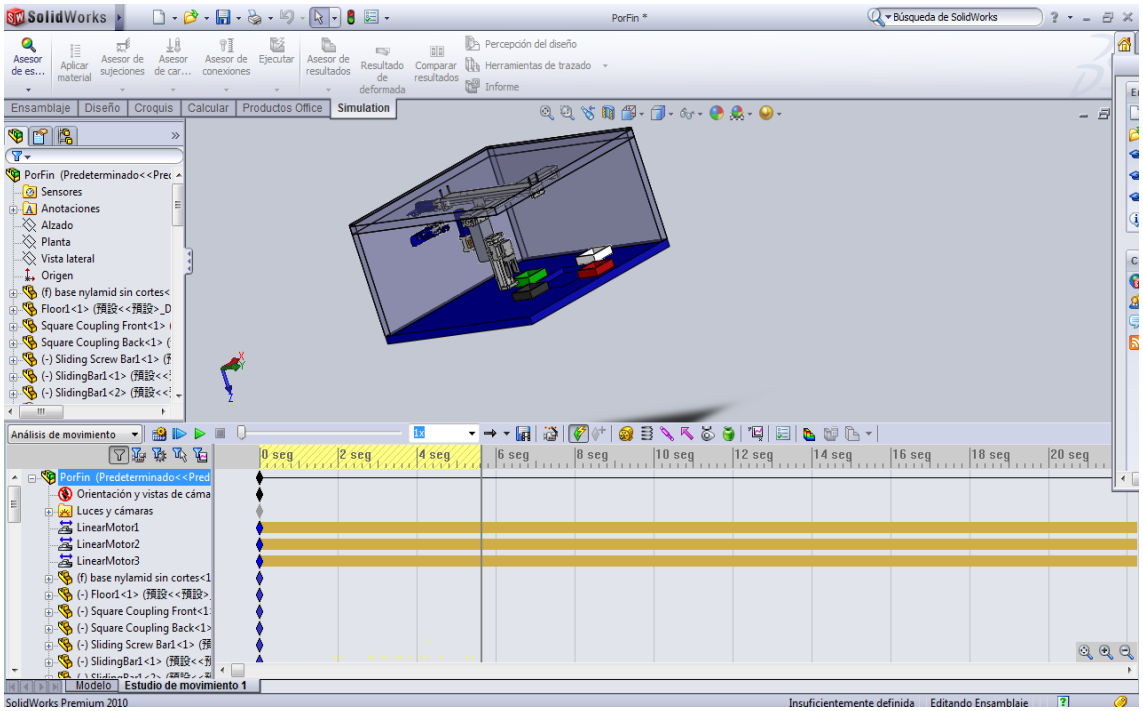


Figura 3.5 Espacio de trabajo del “Estudio de Movimiento” en *SolidWorks*.

3.3.2 Tipos de estudio de movimiento

El siguiente paso es elegir el tipo de "estudio de movimiento" que se usará para el proyecto, como se puede apreciar en la Figura 3.6, existen 3 tipos, Animación, Movimiento Básico y Análisis de Movimiento. Para el proyecto se usará Análisis de movimiento pero no está demás explicar también los otros tipos.

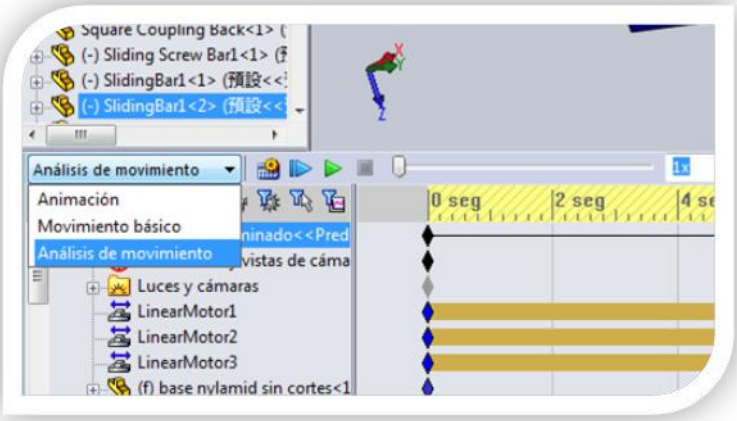


Figura 3.6 Tipos de Estudio de Movimiento

1. Animación

Se utiliza el tipo "Animación" para animar el movimiento de ensamblajes:

- Agregar motores para conducir el movimiento de una o varias piezas de un ensamblaje.

- Prescribir las posiciones de los componentes del ensamblaje en varios momentos mediante la utilización de marcas. "Animación" utiliza interpolación para definir el movimiento de los componentes de un ensamblaje entre marcas. En este tipo de "Estudio de movimiento" no es posible el uso de fuerzas, resortes, contactos, etc. Su uso se limita a observar el movimiento de los componentes en el ensamblaje.

2. Movimiento Básico

Se utiliza "Movimiento básico" en ensamblajes para simular los efectos de motores, resortes, colisiones y gravedad. A la hora de calcular el

movimiento, “Movimiento básico” tiene en cuenta la masa. El cálculo es relativamente rápido, por lo que puede utilizar este estudio de movimiento para crear animaciones tipo presentación mediante simulaciones basadas en leyes físicas.

3. Análisis de Movimiento

“Análisis de movimiento” se utiliza para simular y analizar de forma precisa en un ensamblaje los efectos de elementos de movimiento (incluyendo fuerzas, resortes, amortiguadores y fricción). Este tipo de movimiento utiliza solvers cinemáticos potentes, desde el punto de vista del cálculo, y tiene en cuenta propiedades materiales así como la masa e inercia. También puede utilizar “Análisis de movimiento” para trazar resultados de simulación para análisis adicionales.

La barra de herramientas del menú “*SolidWorks* Motion” también se puede utilizar para:

- Cambiar puntos de vista.
- Mostrar propiedades.
- Crear animaciones con calidad de presentación que muestren el movimiento de un ensamblaje. (9)

3.3.3 Inserción de motores virtuales

Como ya se dijo, en éste caso se tiene que elegir “Análisis de movimiento” para poder comenzar a trabajar en el estudio de movimiento. En este espacio de trabajo se programarán los motores virtuales que darán movimiento a la mesa de coordenadas. Como sólo tiene tres ejes se programarán sólo 3 motores.

Figura 3.7 Símbolo del botón “Motor” dentro del Estudio de Movimiento en *SolidWorks*.

Dentro de *SolidWorks*, un motor es un elemento presente en un estudio de movimiento que mueve componentes en un ensamblaje simulando los efectos que provocaría. En *SolidWorks* Motion hay dos tipos de motores en función del movimiento que se les quiera dar a los componentes:

- Motor Rotatorio
- Motor Lineal

Es importante tener en cuenta que los motores mueven componentes en una dirección seleccionada, pero no son fuerzas. El movimiento originado por motores prevalece sobre el originado por otros elementos de simulación.

No se debe agregar más de un motor del mismo tipo al mismo componente.

Se da clic en el botón "Motor" dentro de la barra de herramientas del estudio de movimiento. Aparecerá un menú del lado izquierdo de la pantalla (Ver la Figura 3.8).

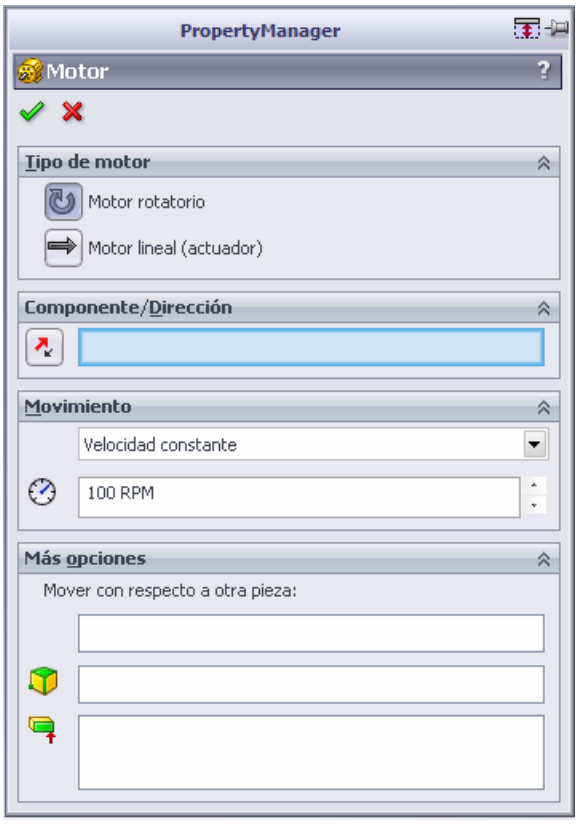


Figura 3.8 Menú de propiedades de un motor dentro de *SolidWorks*.

Se escoge "Motor Lineal" y se da clic en una de las caras de la pieza móvil, la cara en donde se da clic tiene que ser perpendicular a la dirección del movimiento de la pieza (ver la Figura 3.9).

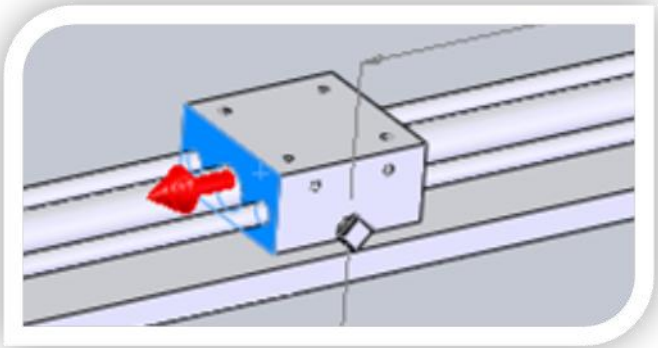


Figura 3.9 Elección de la dirección del motor lineal en un estudio de movimiento en *SolidWorks.*

A continuación en la casilla de “Movimiento” elegimos el tipo de motor “Distancia”, se deja el valor por default “100mm”, se cambia la hora de inicio a “0 [s]” y la duración se cambia a “2 [s]”. Se da clic en la palomita verde para guardar los cambios y listo, el primer motor está configurado.

Se realiza este mismo paso para los otros dos motores.

Como último paso de la inserción de los motores se da clic en “Calcular” para calcular el estudio de movimiento (Ver la Figura 3.10).

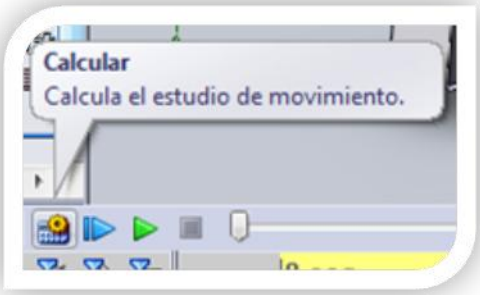


Figura 3.10 Botón “Calcular” en el estudio de movimiento en *SolidWorks.*

CAPÍTULO 4

Programación de la mesa de coordenadas

4.1 Programación en *LabVIEW*

4.1.1 ¿Qué es *LabVIEW*?

LabVIEW es un entorno de programación gráfica usado por miles de ingenieros e investigadores para desarrollar sistemas sofisticados de medida, pruebas y control usando íconos gráficos e intuitivos y cables que parecen un diagrama de flujo. (10)

Al crear un nuevo programa o Instrumento Virtual (VI, por sus siglas en inglés) aparecerán dos ventanas, una con un fondo blanco y otra con fondo gris, la primera de ellas se conoce como “Block Diagram” o diagrama de bloques en español y la segunda se conoce como “Front Panel” o panel frontal en español.

Diagrama de bloques

panel frontal

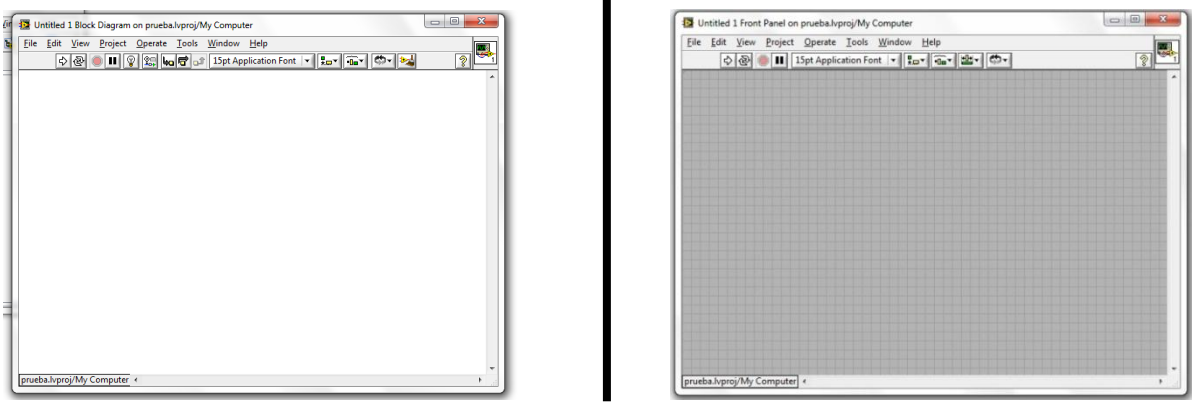


Figura 4.1 Diagrama de bloques y Panel Frontal en *LabVIEW*.

En el diagrama de bloques se colocarán las funciones y bibliotecas que harán funcionar al programa, dando clic derecho en cualquier parte libre en el diagrama de bloques se abrirá un menú emergente donde se puede tener acceso a las cientos de funciones que tiene *LabVIEW*.

En el panel frontal, se encontrarán todo tipo de controles o indicadores, donde cada uno de estos elementos tiene asignado en el diagrama de bloques una terminal, es decir el usuario podrá diseñar un proyecto en el panel frontal con controles e indicadores, donde estos elementos serán las entradas y salidas que interactuarán con la terminal del VI. Se pueden observar en el diagrama de bloques todos los valores de los controles e indicadores y como van fluyendo los datos e instrucciones entre ellos cuando se está ejecutando un programa VI. (11)

4.1.2 Proyecto de *LabVIEW*

Un proyecto en *LabVIEW* se usa para agrupar archivos de *LabVIEW* y archivos externos a él, crear especificaciones, e implementar o descargar archivos a los objetivos.

Se debe usar un proyecto para construir aplicaciones y compartir bibliotecas, así como para trabajar con RT, FPGA, Mobile, Touch Panel, DSP ó un Módulo Embebido. (12)

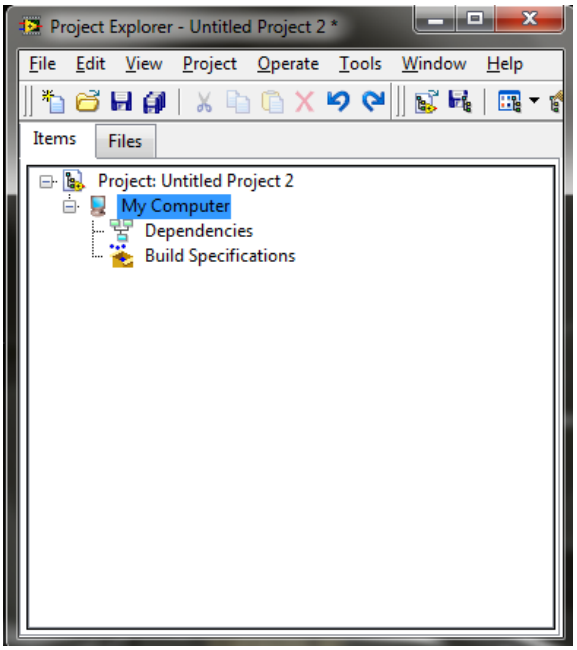


Figura 4.2 Explorador del Proyecto de *LabVIEW*.

4.1.3 Creación del nuevo proyecto y adición de elementos ó *ítems*

Abrimos *LabVIEW* y se da clic en "Empty Project", nos abrirá una nueva ventana de proyecto.

Dentro del proyecto existe el elemento "My Computer", dentro de él se guardarán todas las bibliotecas, dependencias, especificaciones de construcción, archivos externos, ensambles, etc.

Se pueden agregar archivos, folders e hipervínculos, pero además es posible crear archivos vinculados con *LabVIEW*.

Se debe agregar el ensamble al proyecto en *LabVIEW* para poder trabajar en él con los motores que se agregaron anteriormente en el estudio de movimiento en *SolidWorks*, para poder hacer esto, es necesario que el ensamble esté abierto en *SolidWorks*.

Se da clic derecho en "My Computer", se elige "New" y luego "*SolidWorks Assembly*" (Ver la Figura 4.3).

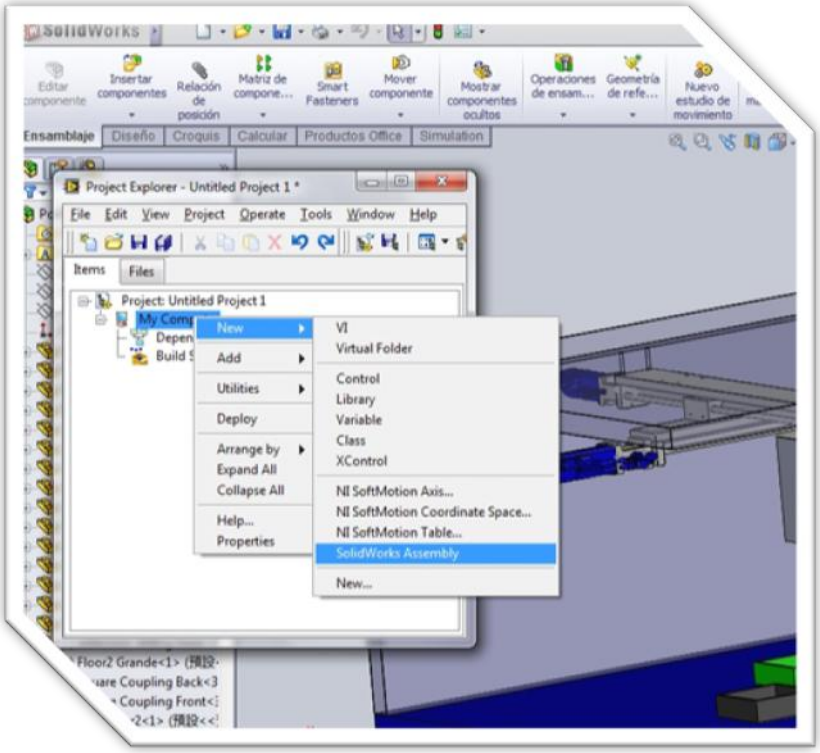


Figura 4.3 Al frente: Agregar ensamble al proyecto de *LabVIEW*. Atrás: *El* ensamble en *SolidWorks*.

Inmediatamente nos aparecerá un cuadro de diálogo mostrando la dirección donde está guardado el ensamble, si es correcto dar clic en "OK", si está mal, hay que buscarlo manualmente dando clic en "Browse" (Ver la Figura 4.4).

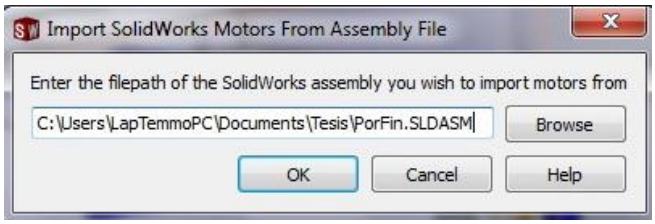


Figura 4.4 Cuadro de diálogo de confirmación de dirección del ensamble en *SolidWorks*.

Al dar clic en "OK" *SolidWorks* cambiará automáticamente al ambiente de Estudio de Movimiento (si es que no se está en él) y quizás aparezca un nuevo cuadro de diálogo pero esta vez de parte de *SolidWorks* preguntando si se desean actualizar las marcas afectadas, esto aparece sólo si la posición en la que se guardó el ensamble difiere a la posición del mismo en el Estudio de Movimiento, se da clic en "Sí" y se continua (ver la Figura 4.5).

Figura 4.5 Cuadro de diálogo de actualización del estado inicial de la animación.

Ahora se tendrá el ensamble dentro del proyecto de *LabVIEW*, se puede ver que dentro del ensamble se encuentran los 3 motores que definimos anteriormente (ver la Figura 4.6).

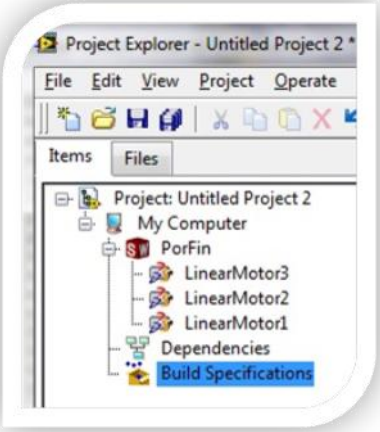


Figura 4.6 Motores dentro del proyecto de *LabVIEW*.

El siguiente paso es agregar los ejes que *SoftMotion* va a controlar, estos ejes son los que los motores van a mover. Se da clic derecho en "My Computer", se entra al menú desplegable "New" y se da clic en "*NI SoftMotion* Axis" (Ver la Figura 4.7).

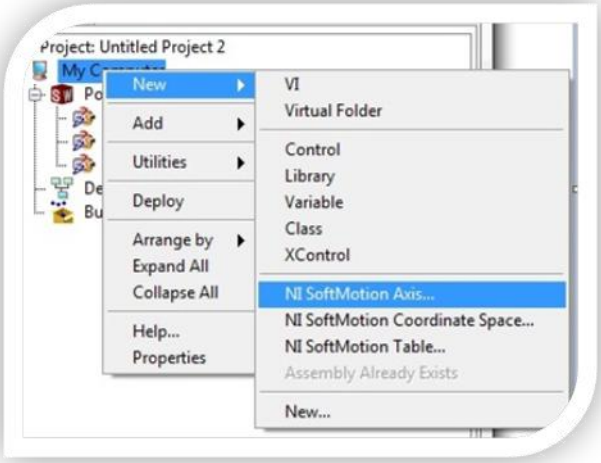


Figura 4.7 Agregar los ejes de *SoftMotion* en el proyecto de *LabVIEW*.

Nos aparecerá un cuadro de diálogo llamado Axis Manager, ahí se agregarán los tres ejes dando clic en "Add New Axis", una vez agregados dar clic en "OK" (Ver la Figura 4.8).

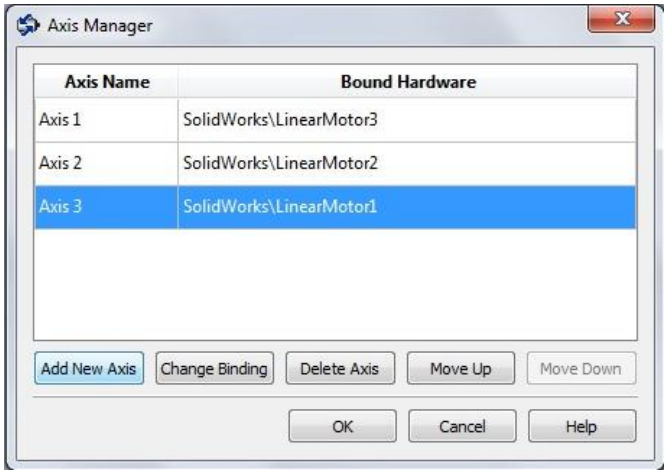


Figura 4.8 Cuadro de diálogo Axis Manager para agregar los ejes de *SoftMotion*.

Los tres ejes aparecerán dentro de "My Computer" llamados Axis 1, 2 y 3. El siguiente paso es habilitar el control sobre estos ejes para poder controlarlos (valga la redundancia) cuando *NI SoftMotion* esté en modo Activo. Para realizar este paso se da clic derecho en el primer eje "Axis 1" y se da clic en la pestaña "Properties" para entrar en las propiedades del eje (Ver la Figura 4.9).

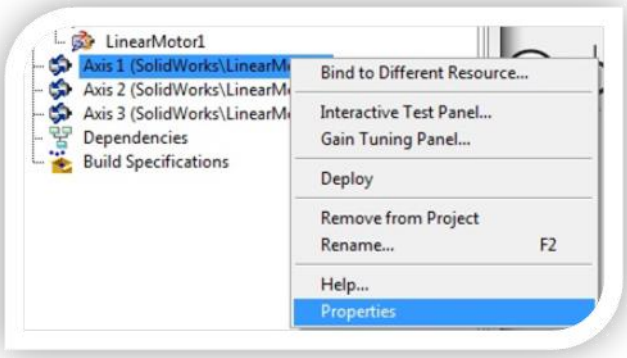


Figura 4.9 Menú desplegable de los ejes de SoftMotion.

Se abrirá un cuadro de diálogo en donde se puede ver la configuración del eje en modo gráfico o estándar, por default nos mostrará la configuración en modo gráfico, en donde se puede apreciar un diagrama de control con cinco botones para modificar desde los valores de configuración del eje hasta incluso modificar los valores del ciclo de control PID y sus ganancias (Ver la Figura 4.10).

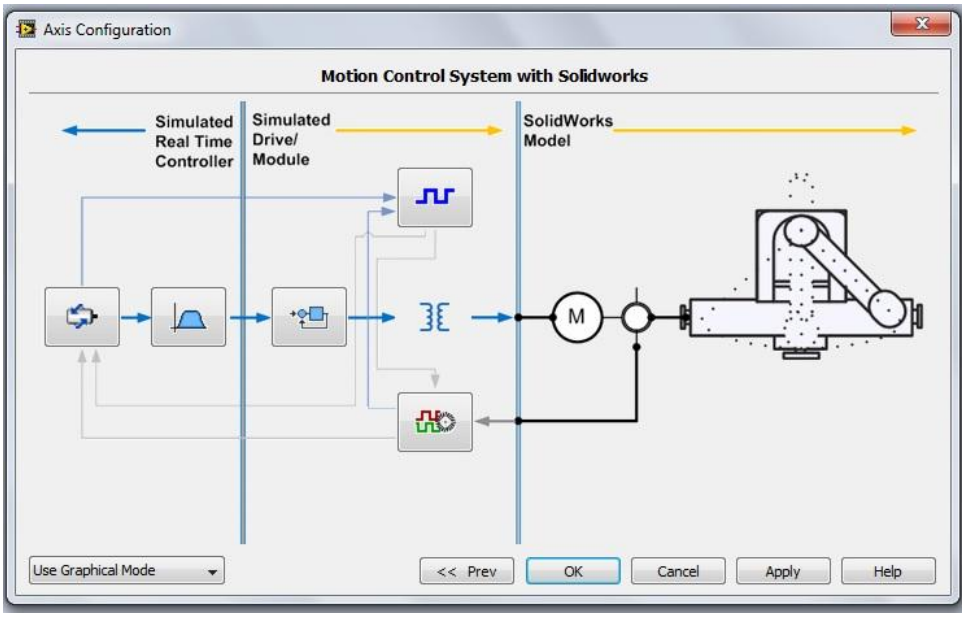


Figura 4.10 Cuadro de diálogo "Axis Configuration".

El primer botón es llamado "Axis Setup"  , dando clic en él se puede acceder a la configuración del eje en donde se pueden modificar parámetros como:

Activar o desactivar el eje.

Activar el control para poder manipular el eje.

El tipo de retroalimentación, que en éste caso es un encoder emulado.

El tipo del eje, que en éste caso es un servomotor.

El modo de ciclo de control, que en éste caso será Posición Interpolada.

El número de escaneos por cada ciclo.

El segundo botón es llamado "Trajectory"  y dando clic en él se pueden modificar criterios de movimiento, filtros, umbrales, retrasos y algoritmos de cálculo de velocidad a los que el modelo estará sujeto durante el estudio de movimiento.

El tercer botón es llamado “Control Loop”  y como su nombre lo dice, aquí se pueden modificar todos los parámetros con respecto al ciclo de control del modelo dentro de *SoftMotion*, al dar clic sobre él, nos mostrará el siguiente cuadro de diálogo (ver la Figura 4.11).

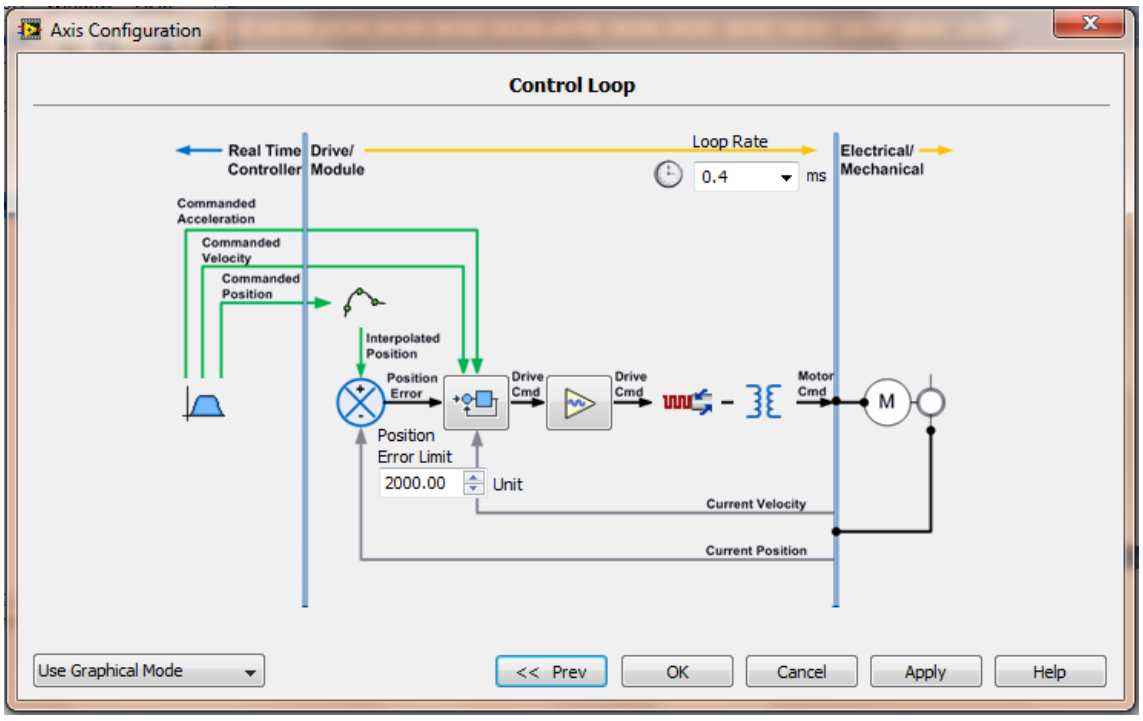


Figura 4.11 Cuadro de diálogo “Control Loop”.

Aquí se pueden modificar los milisegundos que se quiere que tarde el ciclo de control en dar una vuelta y el límite de error de la posición del eje. Además de que se encontrarán dos botones más, el primero es igual al “control loop”  y al acceder se pueden cambiar las ganancias del controlador PID y el segundo botón  sirve para modificar los niveles de voltaje (simulado) máximo y mínimo que recibirá el motor a la salida del ciclo de control.

Dando clic en el primer botón se abrirá un cuadro de diálogo como se puede apreciar en la Figura 4.12.

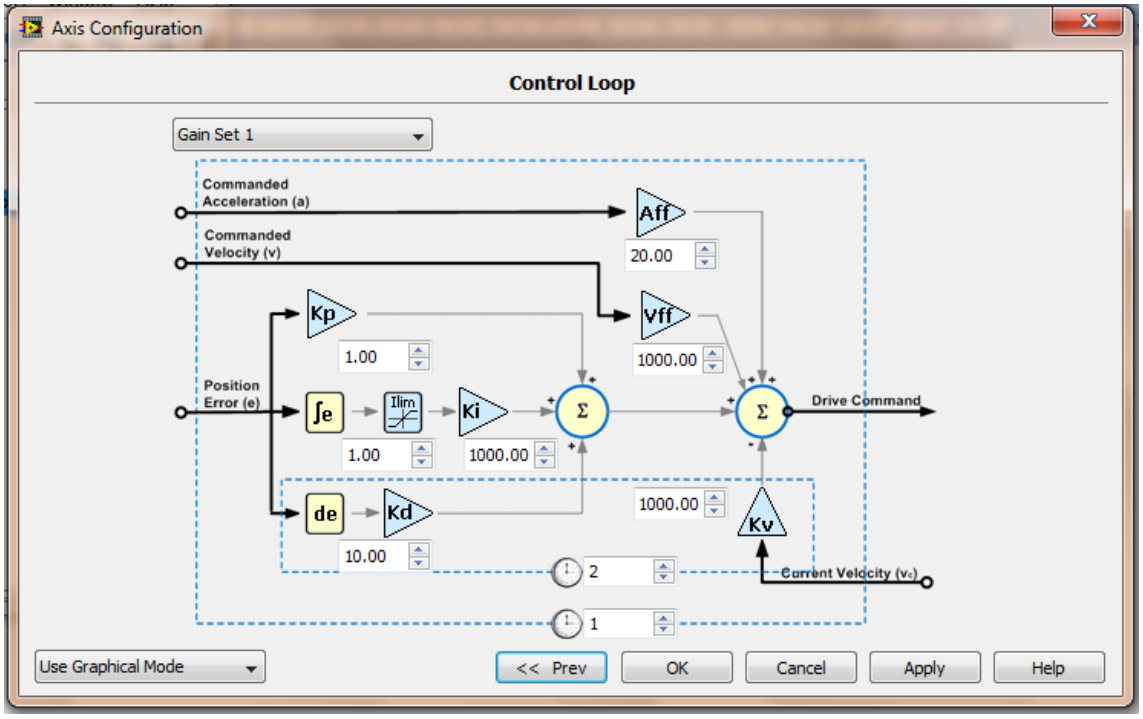


Figura 4.12 Modificación de las ganancias del control PID dentro de "Control Loop".

En este cuadro de diálogo se pueden modificar las ganancias del control PID para modificar el comportamiento de las curvas de reacción, en este caso Posición Vs Tiempo, Velocidad Vs Tiempo y Aceleración Vs Tiempo, además de modificar las ganancias del PID también es posible agregar ganancias prealimentadas o "FeedForward" para mejorar la reacción del sistema. A continuación una breve explicación de la simbología.

Especifica la ganancia Proporcional, su valor puede ir desde 0 hasta 32,767. Su valor por default es 100.

Especifica la ganancia Integral, su valor puede ir desde 0 hasta 32,767. Su valor por default es 0.

Especifica la ganancia Derivativa, su valor puede ir desde 0 hasta 32,767. Su valor por default es 1000.

Especifica el límite de integración, su valor puede ir desde 0 hasta 32,767. Su valor por default es 1000.

Especifica la velocidad de retroalimentación, su valor puede ir desde 0 hasta 32,767. Su valor por default es 0.

Especifica la ganancia de velocidad de prealimentación, su valor puede ir desde 0 hasta 32,767. Su valor por default es 0.

Especifica la ganancia de aceleración de prealimentación, su valor puede ir desde 0 hasta 32,767. Su valor por default es 0. (13)

Regresando a la primera pantalla del cuadro de diálogo "Axis configuration", el cuarto botón es llamado "Digital Input/Output"  , dando clic en él se pueden configurar los parámetros de las líneas de entrada y de salida digitales, como el mapeo y configuración de las mismas.

El quinto y último botón es llamado "Encoder"  , dando clic en él se permite configurar los parámetros del o de los encoders, ya que se pueden tener hasta dos de ellos por cada eje. Se puede configurar el modo de comparación y captura del encoder, así como las unidades de medición, los conteos por unidad y la velocidad del mismo.

Pero para fines prácticos del proyecto por ahora sólo nos limitaremos a usar el primero de los botones del cuadro de diálogo "Axis Configuration" en donde sólo se habilitará el control del eje, se da clic en el botón "Axis Setup"  .

Se abrirán las propiedades del eje y se activará la casilla de "Enable Drive on Transition to Active Mode", si se hiciera este paso los ejes no se moverían en *SolidWorks* al ejecutar la simulación.

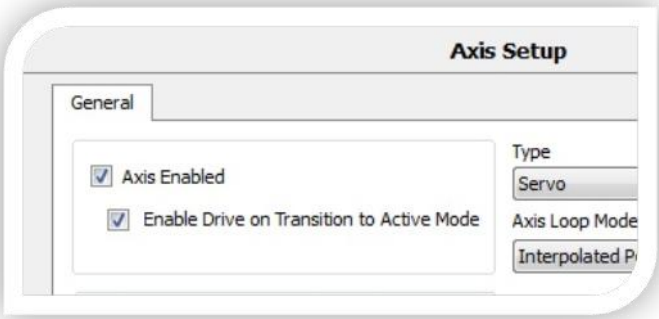


Figura 4.13 Cuadro de diálogo Axis Setup dentro de Axis Configuration en SoftMotion.

Repetimos el paso anterior para los otros dos ejes (Axis 2 y Axis 3).

El siguiente paso es activar el "Scan Engine", para activarlo se da clic derecho en "My Computer" y se escoge la pestaña "Properties" del menú desplegable. Nos aparecerá el cuadro de diálogo "My Computer Properties", una vez ahí se da clic en la categoría "Scan Engine" y se activa la casilla "Start Scan Engine on Deploy", uno de los errores más comunes es el olvido de la activación de esta casilla (Ver la Figura 4.14).

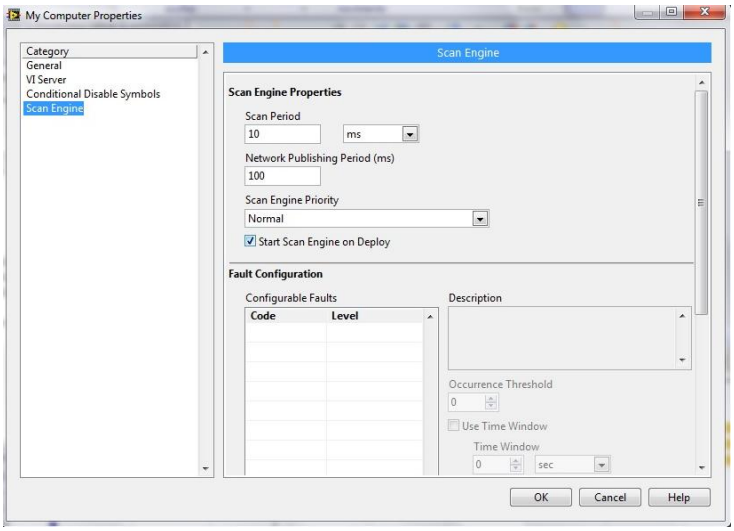


Figura 4.14 Cuadro de diálogo My Computer Properties.

Una vez activado el Scan Engine se puede empezar con la programación en *LabVIEW*.

4.1.4 Adición y programación de un VI en el proyecto de *LabVIEW*

Continuando con el proyecto, el siguiente paso es agregar un VI o "Instrumento Virtual", para hacerlo se da clic derecho en "My Computer" y se coloca el puntero sobre el menú emergente "New", elegimos "VI".

Se abrirán dos ventanas, el panel frontal y el diagrama de bloques.

Se da clic derecho en cualquier parte libre del diagrama de bloques para abrir el menú de funciones, expandimos el menú dando clic en la doble flecha, se entra a la categoría "Vision and Motion" (Si no aparece esta categoría hay que reinstalar *NI SoftMotion*), ahora accedemos a la categoría "*NI SoftMotion*", por último se entra a la categoría "Function Blocks" y se da clic en el botón "Line" ó "Straight-Line Move" (ver la Figura 4.15).

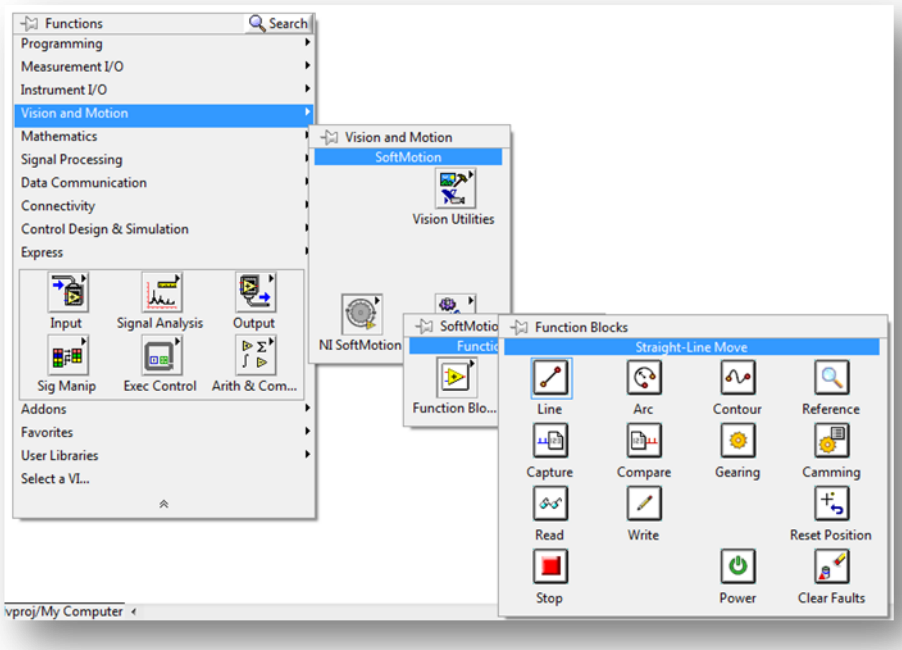


Figura 4.15 Agregar función Straight-Line Move al diagrama de bloques en *LabVIEW*.

Se coloca el cuadro de control en cualquier parte del diagrama de bloques, a la izquierda se pueden ver las entradas que tiene la función y a la derecha las salidas, por ahora sólo se usarán las entradas "error

in", "resource", "execute" y "position" y las salidas " error out" y "done" (ver la Figura 4.16).

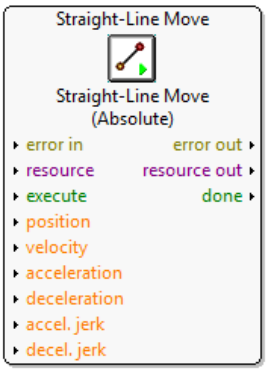


Figura 4.16 Cuadro de control Straight-Line Move.

El siguiente paso es agregar uno de los ejes de *SoftMotion* ubicados en el proyecto, para esto literalmente se arrastrará el "Axis 1" desde la ventana del proyecto hasta el diagrama de bloques (Ver la Figura 4.17).

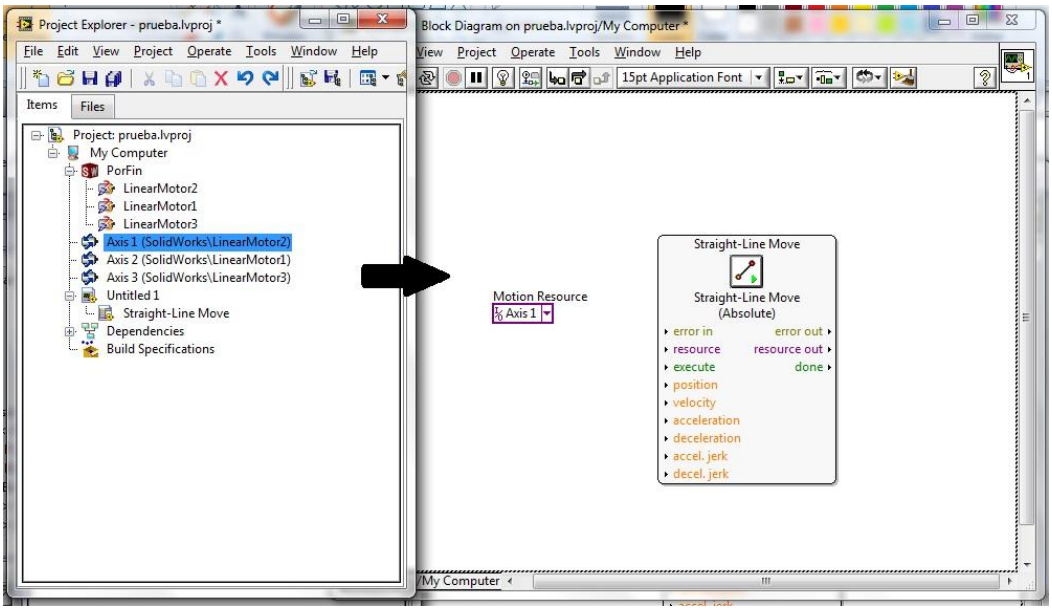


Figura 4.17 Arrastrando un eje desde el proyecto hasta el Block Diagram en *LabVIEW*.

Unimos la salida de "Motion Resource" del eje 1 a la entrada "resource" del cuadro de control por medio de un cable.

Ahora se creará un control para la entrada "execute", esto hará que aparezca un botón de control en el panel frontal para que se pueda ejecutar la aplicación o detenerla cuando se desee, para hacer esto se da clic derecho en la entrada "execute", posicionamos el mouse en "create", y se da clic en la casilla "control" (Ver la Figura 4.18).

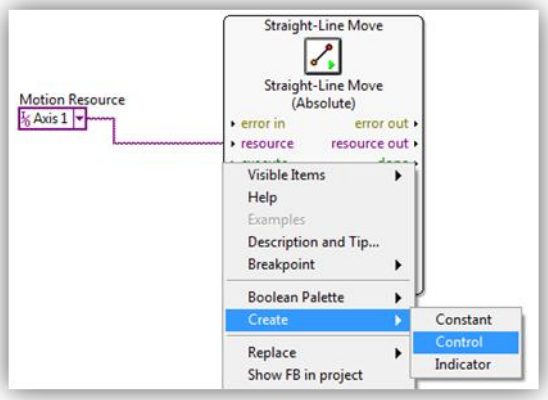


Figura 4.18 Creación del botón de control para la entrada Execute.

Se creará un cuadro de control en el diagrama de bloques unido a la entrada execute por un cable y también se creará un botón llamado "execute" en el panel frontal. (Ver la Figura 4.19)

Diagrama de bloques

Panel frontal

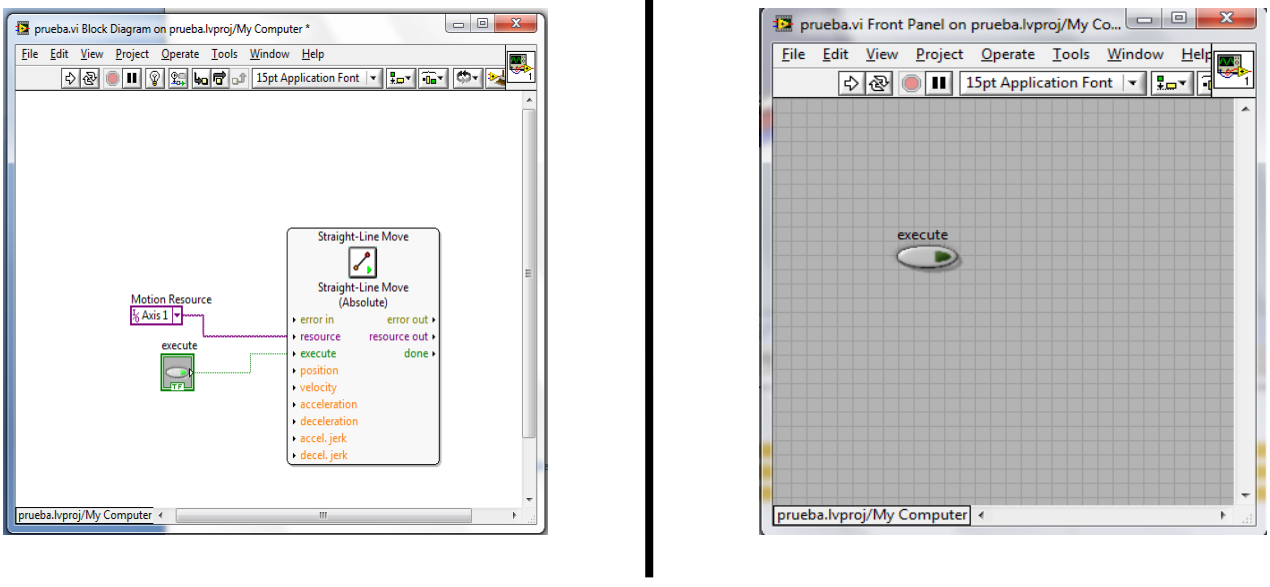


Figura 4.19 Creación del botón de control "execute"

El siguiente paso es crear de igual forma un control para la entrada "position" pero en este caso la entrada no es un valor booleano (verdadero o falso) como "execute", sino un valor numérico. El control se crea exactamente de la misma forma, dando clic derecho en la

entrada “position” y eligiendo “create” y “control”. En el diagrama se creará un cuadro de control unido a la entrada y en el Panel se creará un control numérico en donde se podrá escribir la posición a la que se desee que se mueva el eje (ver la Figura 4.20).

Figura 4.20 Control de posición en el Panel Frontal.

El siguiente paso es crear una estructura de ciclo llamada “Timed Loop” ó Ciclo de Tiempo, esto hará que el sistema retroalimentado funcione y que al mismo tiempo esté sincronizado con el “Scan Engine”.

Para agregar el “Timed Loop” se da clic derecho en cualquier parte libre del diagrama de bloques y expandimos el menú de funciones con la doble flechita hacia abajo  para poder ingresar al menú desplegable “Programming”, dentro de él se escoge “Structures”, luego “Timed Structures” y por último se da clic en “Timed Loop” (Ver la Figura 4.21).

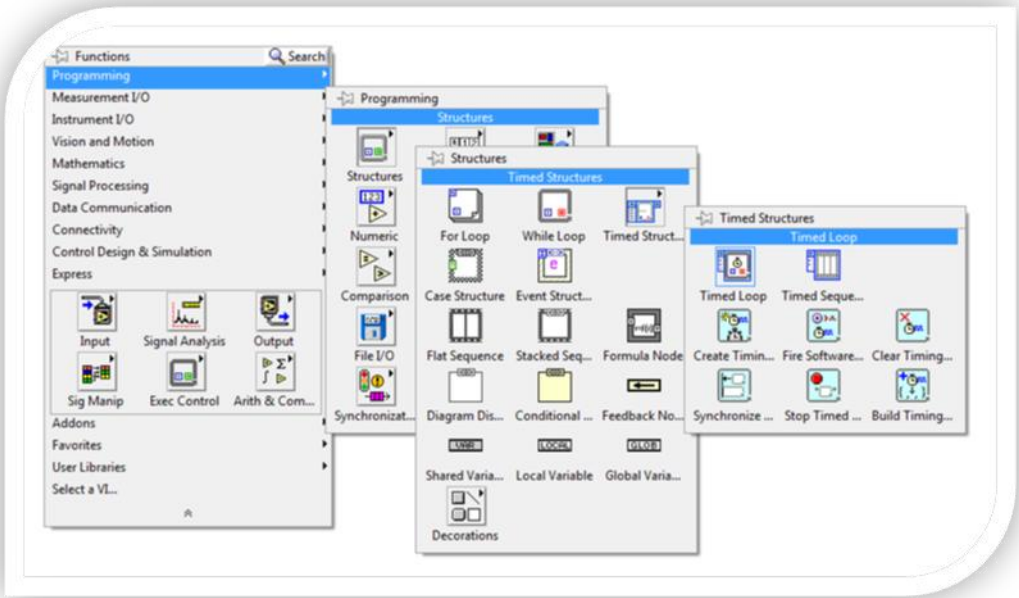


Figura 4.21 Insertar un Timed Loop en *el* diagrama de bloques.

Para colocar el *Timed Loop* se debe literalmente encerrar todo lo que se tiene en el diagrama de bloques para que todas las funciones entren en el ciclo (Ver la Figura 4.22).

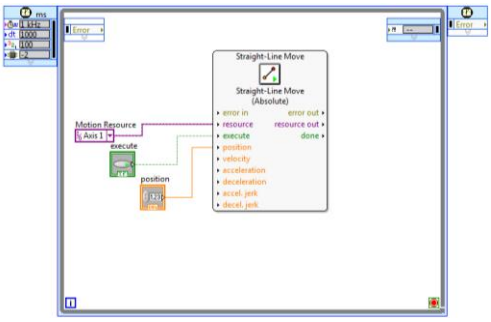


Figura 4.22 Timed Loop encerrando las funciones en *el* diagrama de bloques.

Ahora se configurará el "Timed Loop" para que esté sincronizado con el *Scan Engine* dando doble clic en el reloj que dice "ms" ms en la pestaña izquierda del ciclo. Se abrirá un cuadro de diálogo llamado "Configure Timed Loop" y en la categoría "Loop Timing Source" debe estar escogida la casilla "Use Built-In Timing Source", se cambia el Source Type a "Synchronize to Scan Engine" y se da clic en "OK" (Ver la Figura 4.23).

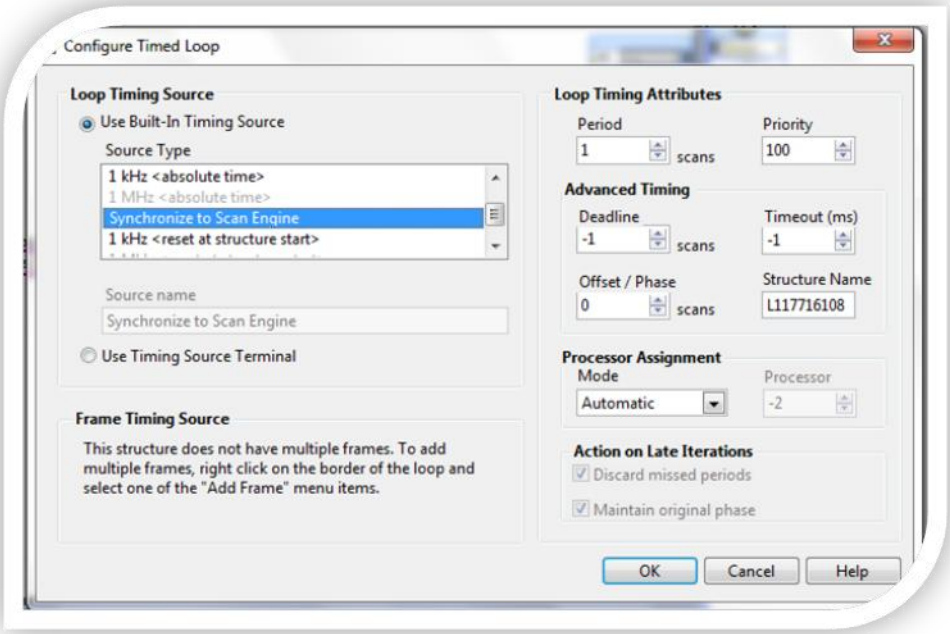


Figura 4.23 Cuadro de Diálogo "Configure Timed Loop".

Ahora el siguiente paso es unir la salida "error out" con una de las paredes de la parte derecha del ciclo. Se da clic en la salida "error out"

y unimos el cable con la pared derecha. Se debe crear una retroalimentación del error dando clic en el cuadro que se formó en la pared del ciclo y se abrirá un menú desplegable donde se dará clic en “Replace with Shift Register” (ver la Figura 4.24).

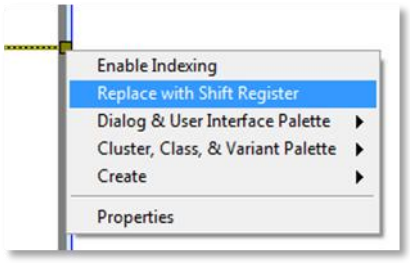


Figura 4.24 Insertar Shift Register.

El cuadro de color marrón cambiará por un cuadro con una flecha hacia arriba y aparecerá al mismo tiempo un cuadro muy parecido en la pared izquierda del ciclo pero con una flecha hacia abajo. Cada vez que el ciclo dé una vuelta, la salida “error out” mandará información de si hay error o no y cuando empiece el nuevo ciclo o vuelta, esta información le llegará a la entrada “error in”. Para poder hacer esto basta con unir la salida del “shift register” (flecha hacia abajo) con la entrada “error in” (Ver la Figura 4.25).

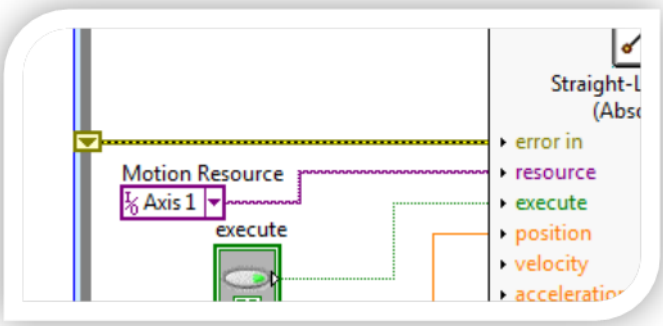


Figura 4.25 Salida del Shift Register a la entrada error in.

Como último paso, unimos la salida “done” a la condición del ciclo o “Loop Condition”, esto hará que cuando el eje llegue a la posición deseada y mande la señal “done”, ó “hecho” en español, el ciclo se detenga (ver la Figura 4.26).

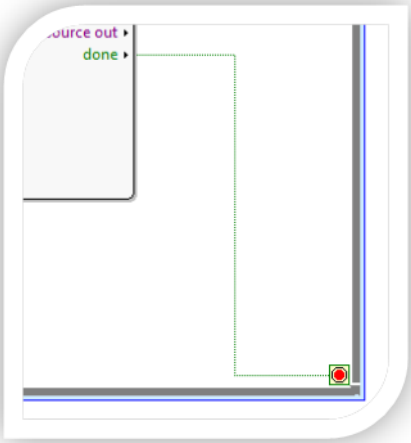


Figura 4.26 Salida “done” manda la señal de paro del ciclo.

4.2 Prueba de la programación de un eje

Ahora se aprobará la programación que se acaba de hacer. Pero antes de dar clic en el botón “run” en el programa se tienen que iniciar algunas cosas.

El primer paso es moverse al proyecto de *LabVIEW*, ahí se seleccionarán los elementos “My Computer”, el ensamble junto con sus tres motores y los tres ejes de *SoftMotion*, se da clic derecho sobre ellos y se da clic en “Deploy”, que en español significa implementar (Ver la Figura 4.27).

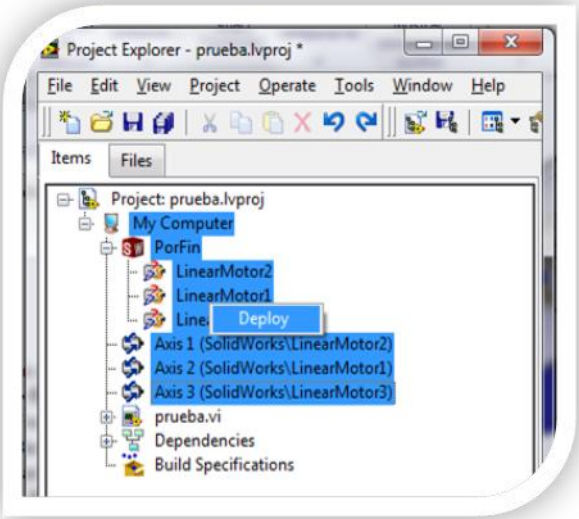


Figura 4.27 Deploy de los elementos del proyecto de *LabVIEW*.

Si todo está bien aparecerá un cuadro de diálogo llamado “Deployment Progress” y después de completar el proceso dirá “Deployment completed successfully”, se da clic en “Close” y continua al siguiente paso.

Nota:

Si se tiene errores en el proyecto, al implementar los ejes (es decir, al hacer clic en “deploy”) aparecerá un cuadro de diálogo (ver la Figura 4.28) mostrando algunos errores y la manera de corregirlos.

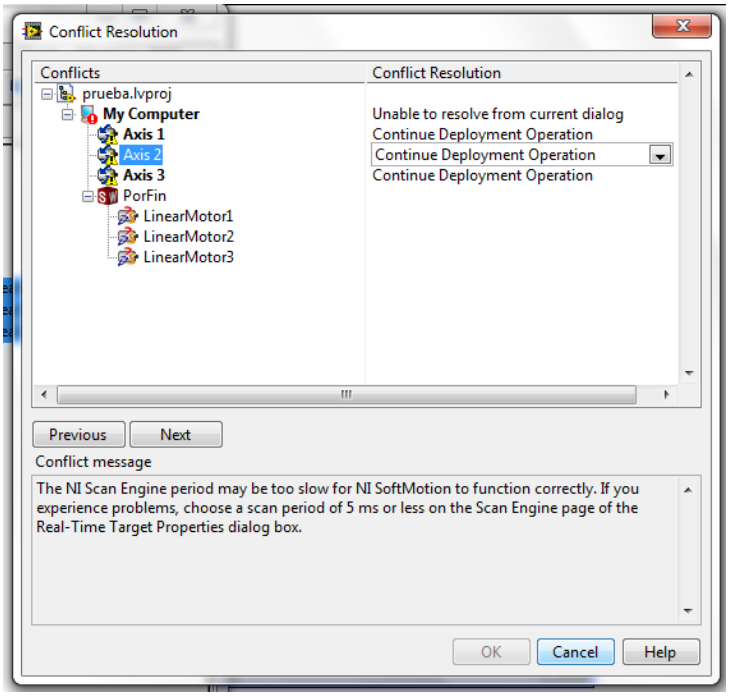
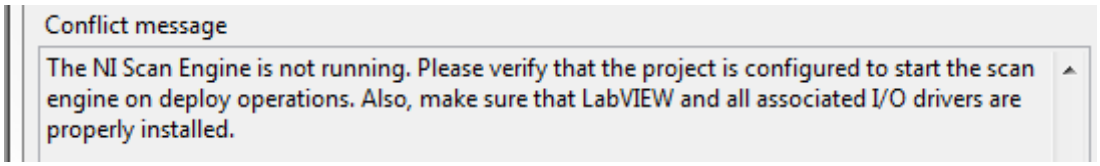


Figura 4.28 Cuadro de diálogo “Conflict Resolution”.

Se presentan a continuación los dos errores más comunes y la manera de solucionarlos:

1) Si aparece este símbolo  y el siguiente mensaje:



Quiere decir que no han activado el “Scan Engine”. Si no saben cómo activarlo vayan al último paso del punto 5.2, en la figura 5.10 pueden ver cómo debe de estar la configuración.

2) El error dice: “The Maximum Step Size configured for this *SolidWorks* Assembly is greater than or equal to the NI Scan Engine Scan Period”.

   Para solucionarlo, se da clic derecho en el ensamble dentro del proyecto y se accede a “properties”. En la casilla “Maximum Step Size” y se cambia el número por 0.001, se da clic en “OK” y listo, problema solucionado (ver la Figura 4.29).

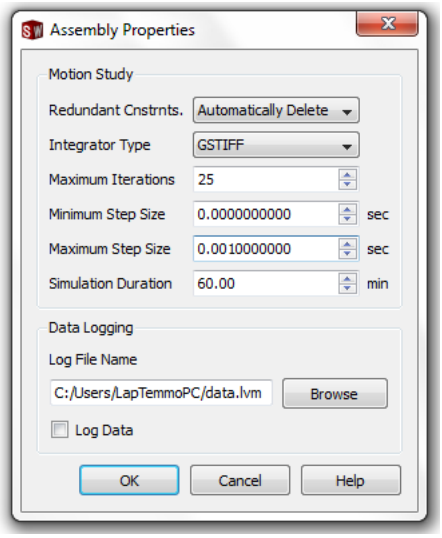


Figura 4.29 Cuadro de diálogo “Assembly Properties”

Antes de ejecutar la prueba se reduce el panel frontal a lo más compacto posible sin que dejen de observarse los controles y se puede ocultar o minimizar el diagrama de bloques, esto con la finalidad de ver el ensamble al mismo tiempo que se controla el panel frontal, así se podrá ver como se mueve el ensamble (ver la Figura 4.30).

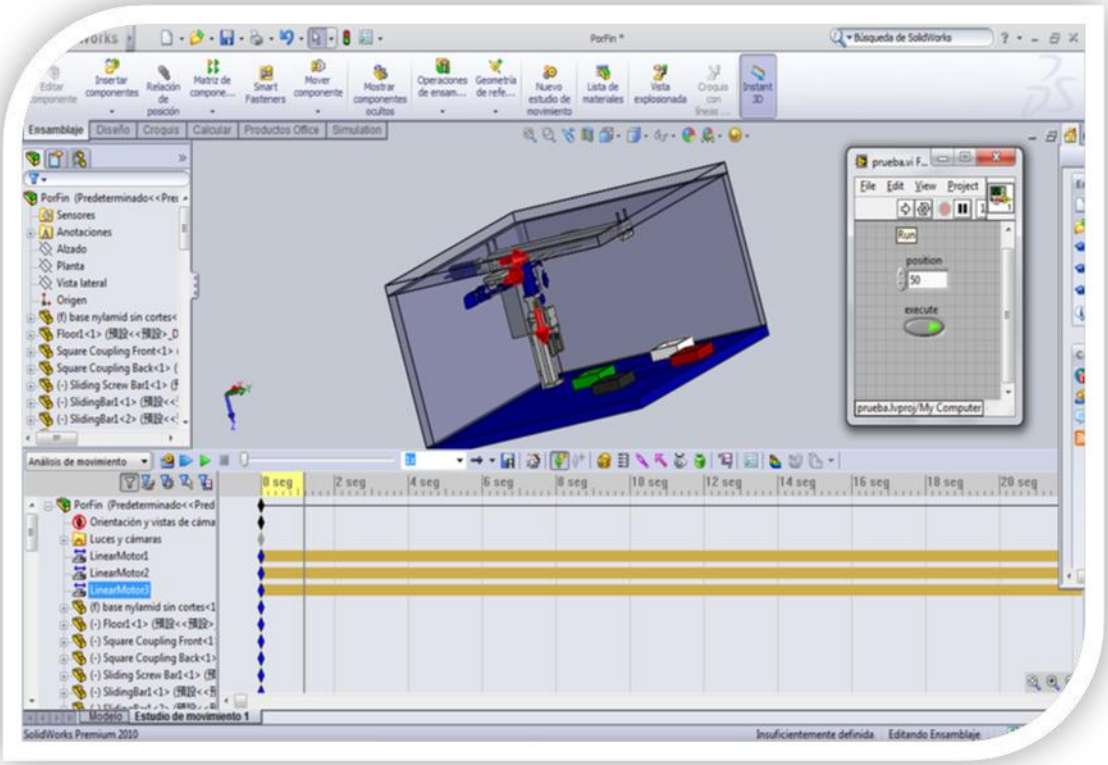


Figura 4.30 Prueba de la programación de un eje en *LabVIEW*, el ensamble de SW al fondo.

Listo ahora ya se puede dar clic en “RUN” en el VI.

Escribimos un número no muy grande en “position”, por ejemplo 50 y se da clic en “execute”, si todo salió bien se podrá ver como el eje recorre 50 mm.

La programación de los otros dos ejes se realiza de manera similar y queda libre para el lector.

En la Figura 4.31 se puede observar el diagrama de bloques de la programación de la mesa de coordenadas en éste proyecto.

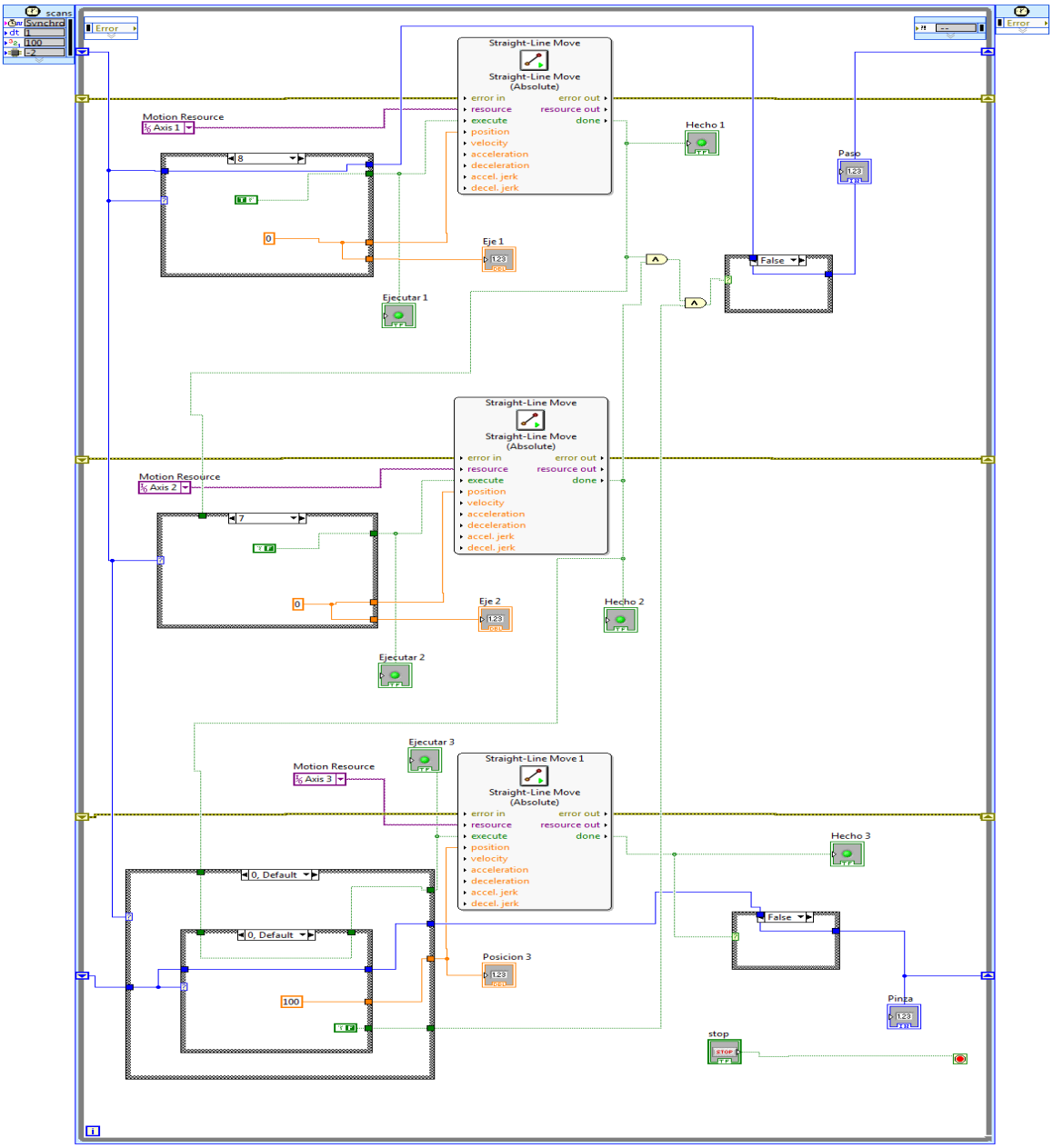


Figura 4.31 Diagrama de bloques de la programación de la mesa de coordenadas en *LabVIEW*.

4.3 Integración del software de visión

Por último se aprenderá a integrar el software *Vision Builder* al proyecto en *LabVIEW*.

4.3.1 ¿Qué es *Vision Builder AI*?

NI *Vision Builder* for Automated Inspection (AI) de *National Instruments* es un entorno configurable de desarrollo de visión artificial que no requiere programación. *Vision Builder AI* ayuda a

resolver la mayoría de los retos de aplicaciones de visión artificial sin el uso de un lenguaje de programación o herramientas complicadas para personalización. *Vision Builder AI* incluye el software NI Vision Acquisition, un juego de controladores y utilidades para adquirir, mostrar y guardar imágenes desde cualquier tarjeta de adquisición de imágenes, cámara GigE Vision o cámara IEEE 1394. El software nos permite:

- Adquirir y procesar imágenes con cualquier tarjeta de adquisición de imágenes, GigE Vision o cámaras IEEE 1394 o el Sistema Compact Vision.
- Construir, evaluar y desplegar aplicaciones de visión artificial completas sin programación.
- Configurar más de 100 potentes herramientas de visión artificial incluyendo igualación geométrica, OCR y análisis de partículas.
- Comunicar resultados de disparo e inspección directamente a dispositivos industriales de E/S digital, serial o protocolos Ethernet. (14)

4.3.2 Programación de *Vision Builder AI*

Abrimos Visión Builder y se da clic en “Configure Inspection” (ver la Figura 4.32)

Figura 4.32 Cuadro de Inicio de *Vision Builder* for Automated Inpection.

Para que *Vision Builder AI* funcione debe de obtener imágenes, ya sea de alguna cámara conectada al equipo o tarjeta de adquisición o para éste caso puede obtenerlas de un archivo.

Pueden utilizar las mismas imágenes que se usaron en el proyecto (ver las siguientes tres imágenes en la Figura 4.33), guardándolas en una carpeta especial para ellas, por ejemplo, “Imágenes Fichas”.

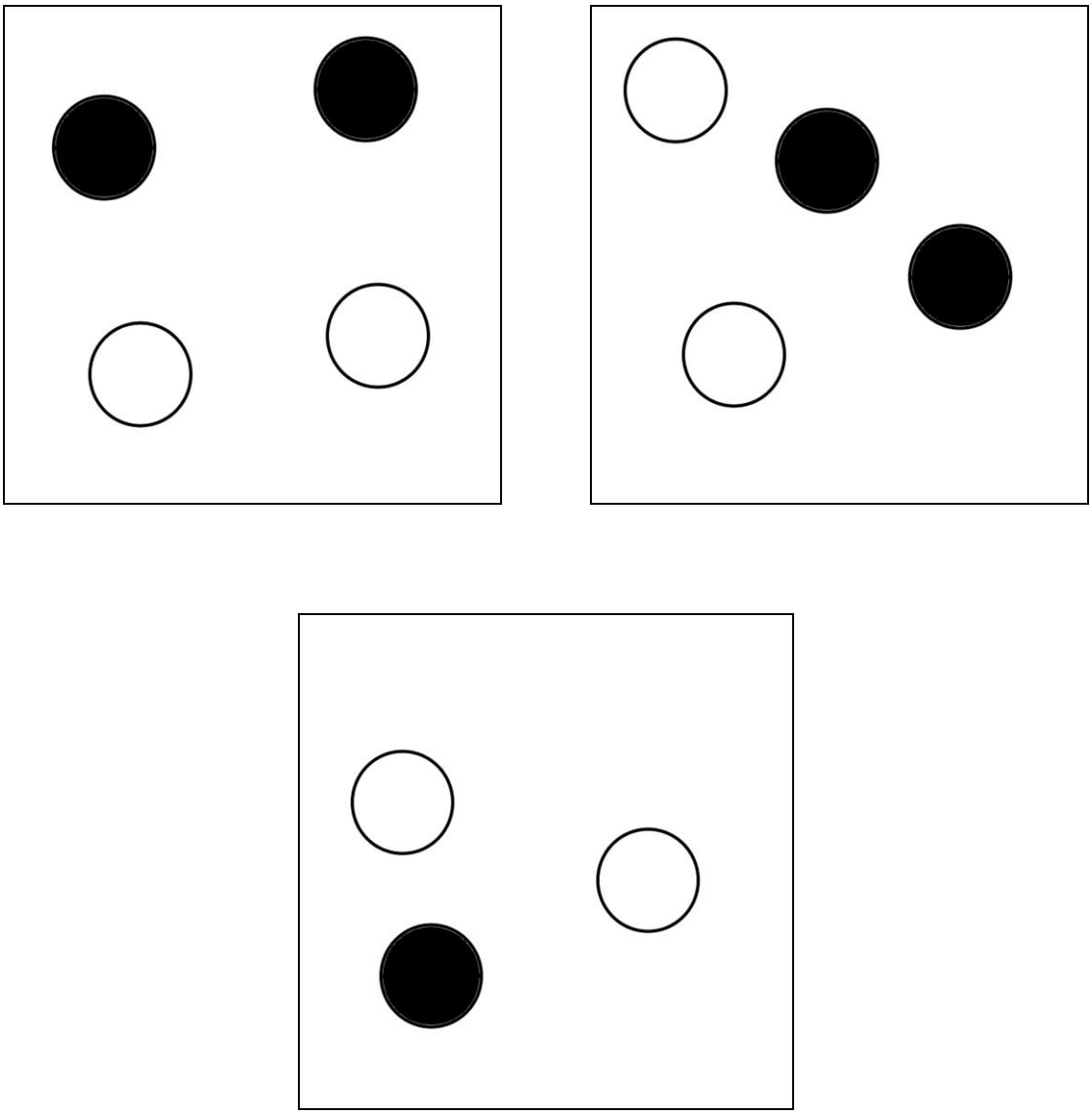


Figura 4.33 Las tres imágenes que usamos en *el* proyecto para simular fichas blancas y negras sobre una hoja blanca.

Una vez guardadas las imágenes se podrá empezar con el primer paso, la adquisición de imágenes.

Se da clic en “Simulate Acquisition” dentro de la pestaña “Acquire Images” de los Inspection Steps. (Ver la Figura 4.34).

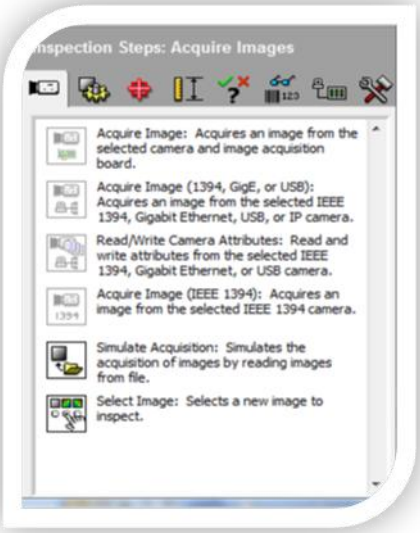


Figura 4.34 Pestaña "Acquire Images" dentro de Inspections Steps en *Vision Builder AI.*

El cuadro de diálogo cambiará a "Simulate Acquisition Setup" en donde se dará un nombre al primer paso de adquisición de imágenes, por ejemplo, "Adquirir Imágenes". También se le dirá la dirección de la carpeta con nuestras imágenes dando clic en "Browse" o escribiéndola directamente si es que se sabe. Por último se dejará activada la casilla "Cycle Through Folder Images", esto es para que adquiera todas las imágenes dentro de nuestra carpeta y aplique todos los pasos de Inspección a ellas, simulando ser una cámara y diferentes productos o cosas a inspeccionar en tiempo real. Se da clic en "OK". (Ver la Figura 4.35).

Figura 4.35 Cuadro de diálogo *Simulate Acquisition Setup*.

El siguiente paso es calibrar la imagen para que el programa pueda calcular distancias en unidades reales, para este ejemplo la hoja de papel puede medir lo que se desee, pero para casos más específicos se podrá tomar una fotografía del objeto bajo supervisión junto a una regla o algún instrumento de medición como un vernier.

Se cambia a la siguiente pestaña de los Pasos de Inspección (Inspection Steps), llamada, "Enhance Image". Ahí se da clic en el botón "Calibrate Images". (Ver la Figura 4.36).

Figura 4.36 Pestaña "Enhance Images" de Inspection Steps.

Al dar clic se abre el cuadro de diálogo "Calibrate Image Setup", ahí se da un nombre al paso de inspección en la casilla "Step Name" por ejemplo, "Calibración". Se da clic en el botón "New Calibration…" y aparecerá en cuadro de diálogo "Calibration Wizard". (Ver la Figura 4.37).

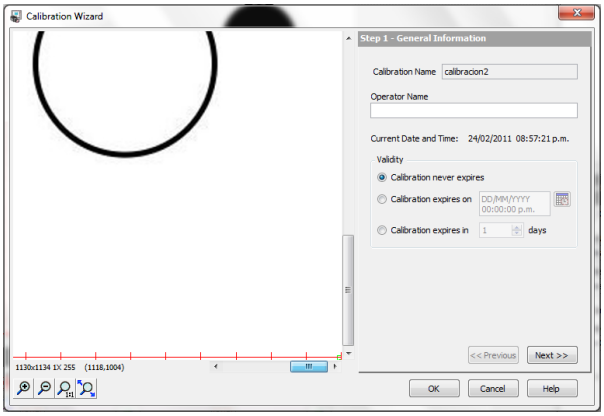


Figura 4.37 Cuadro de diálogo "Calibration Wizard".

Se da un nombre a la nueva calibración, por ejemplo “Calibración de Ficha”, se da clic en “Next”. El siguiente paso es seleccionar el tipo de calibración, para éste caso la calibración Simple (Simple Calibration), se da clic en “Next”.

Ahora se escogerá la imagen que será nuestra referencia, se usará la primera de las imágenes que es la que se está usando en este momento, así que se dejará seleccionada la casilla “Use Current Image”, se da clic en “Next”. El siguiente paso es seleccionar el tipo de Pixel de nuestra cámara, se deja seleccionado “Square” ya que nuestra imagen es una imagen digital de pixeles cuadrados, se da clic en “Next”.

El siguiente paso es seleccionar los puntos de referencia, para éste caso se dará clic en la esquina inferior izquierda como primer punto y en la esquina inferior derecha como segundo punto, en la casilla “Correspondence Image – Real World” se escribirá 250 (o el número que les convenga para su propio proyecto) y en “Unit” se escoge milímetros (o la unidad que a ustedes les convenga), se da clic en “Next”.

El siguiente paso es escoger el origen del eje de calibración, se puede dejarlo en (0,0) o cambiarlo para que las unidades no les salgan negativas, en éste caso se cambiará el sistema para que el eje de las “Y” sea positivo hacia arriba, se da clic en “OK”, se da clic de nuevo en “OK” para que la calibración quede agregada en el diagrama de Pasos de Inspección (Ver la Figura 4.38).

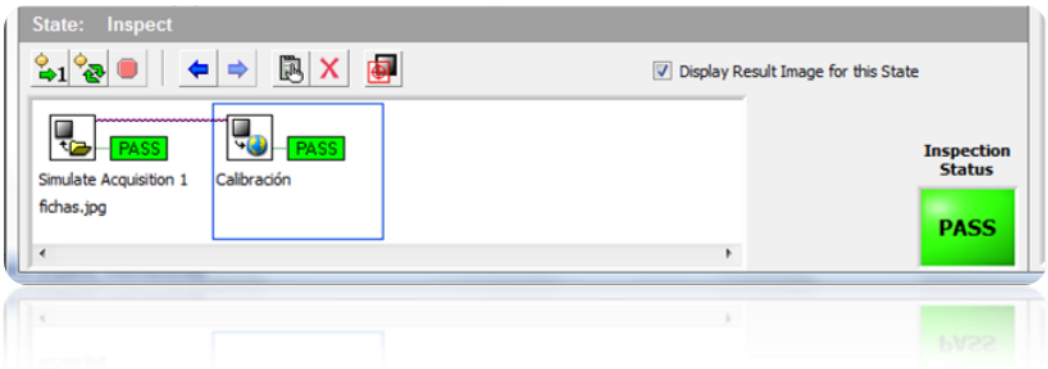


Figura 4.38 Diagrama de Pasos de Inspección.

El siguiente paso es detectar los objetos de la imagen. Se cambia a la siguiente pestaña de los Pasos de Inspección “Locate Features” y se da clic en el botón “Detect Objects”, aparecerá el cuadro de diálogo “Detect Objects Setup” y un cuadro azul encima de la imagen (ver la Figura 4.39).

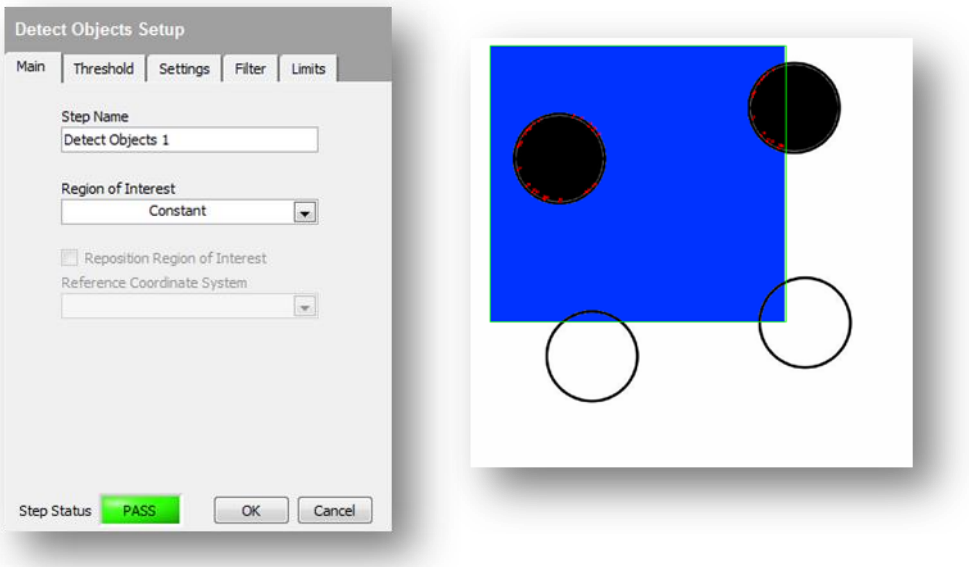


Figura 4.39 Detect Objects.

En este paso se detectará a las fichas blancas, por lo que se podrá nombrar a este paso como “Detectar fichas blancas”. En la sección “Region of interest” se escoge la opción “Full Image”, esto es para que el programa busque en toda la imagen los parámetros que se le digan, en este caso buscará fichas blancas en toda la imagen.

Hay varias maneras de hacer que el programa encuentre las fichas, puede ser por el color, por la forma, por el tamaño, etc., en este caso se guiará por el área de color de la ficha, ya que es una de las formas más fáciles de hacerlo.

Se da clic en la siguiente pestaña del cuadro diálogo, “Threshold”, en la sección “Look for” se escoge “Dark Objects”, esto hará que el programa busque colores obscuros para después poder decirle que mida su área. En el campo “Method” se escoge “Auto Threshold: Clustering”, hasta ahora con estos pasos el programa ya debió haber encontrado cuatro objetos, los remarcará con un cuadro rojo.

Se avanza a la siguiente pestaña, “Settings”, se dejan activadas sólo las casillas “Minimum object size” y “Maximum object size”, para éste caso se escribirá 100 y 300 [mm^2] respectivamente, pero para su programa pueden variar las medidas. Una vez hecho esto el programa debe de detectar sólo 4 objetos, las fichas blancas (ver la Figura 4.40). Se da clic en “OK”.

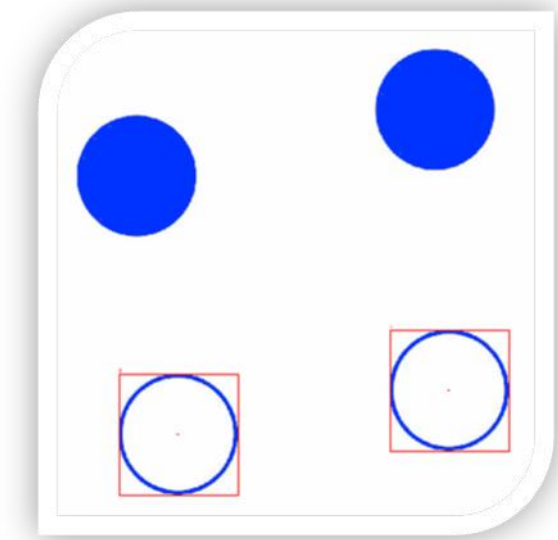


Figura 4.40 Detectando fichas blancas.

Para detectar las fichas negras se puede copiar el mismo paso anterior del diagrama de Pasos de Inspección, dando clic derecho en él y escogiendo “Copy” del menú desplegable, se da clic derecho en un área libre y se escoge “*Paste*”. Ahora se da doble clic en el nuevo paso para modificarlo y que detecte las fichas negras. Se cambia el “*Step Name*” por “Detectar fichas negras” o el nombre de su gusto y lo único que se tiene que hacer para detectar las fichas negras es modificar el mínimo y el máximo tamaño del objeto en la pestaña “Settings”, en éste caso se escribirá 1000 [mm^2] para el mínimo y 3000 [mm^2] para el máximo, el programa tiene que estar detectando dos objetos, en este caso, el número de fichas negras.

Ahora el programa es capaz de detectar el número de fichas blancas y negras que hay, el último paso es hacer que el programa en *Vision Builder AI* nos “transfiera” o comparta el número de fichas y su ubicación en el plano coordenado, al programa en *LabVIEW*, para que la Mesa de Coordenadas Virtual localice cada ficha, la encuentre y la

lleve a su depósito correspondiente. Para esto es necesario que aprense da a crear variables compartidas. (15)

4.3.3 Creación de Variables compartidas

Lo primero que se debe hacer es crear variables para cada ficha. Para hacer esto abrimos el proyecto de *LabVIEW*, se da clic derecho en "*My Computer*", se posiciona en "*New*" y se escoge "*Variable*" (ver la Figura 4.41).

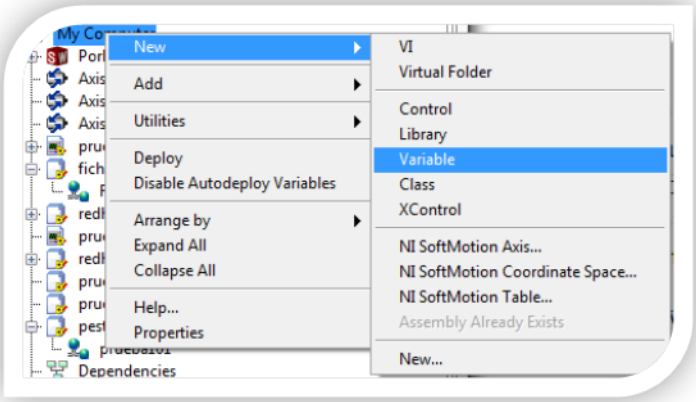


Figura 4.41 Agregar nueva Variable al proyecto en *LabVIEW*.

Se abrirá el cuadro de diálogo "*Shared Variable Properties*" en donde en el campo "*Name*" se dará un nombre a la variable, por ejemplo, "FichaNegra1_X" y se da clic en "OK" (ver la Figura 4.42).

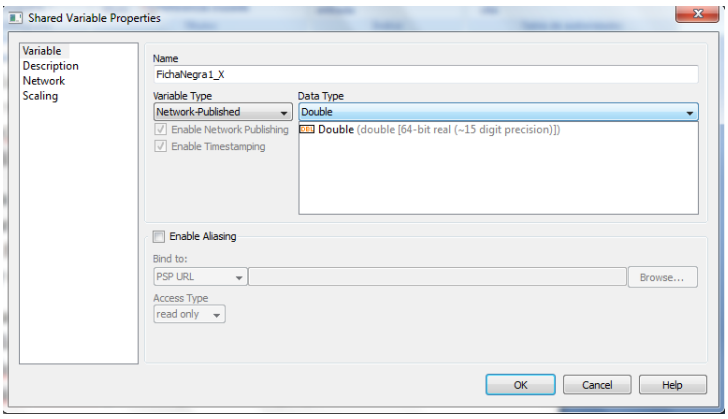


Figura 4.42Cuadro de diálogo Shared Variable Properties.

La nueva variable aparecerá en el proyecto dentro de una nueva biblioteca llamada “Untitled Library 1”, para que esta biblioteca tenga un nombre en especifico es necesario guardar el proyecto, automáticamente *LabVIEW* preguntará con qué nombre se quiere guardar la biblioteca, por ejemplo, “Fichas Negras”.

Para que la nueva variable sea accesible al programa *Vision Builder* para leerla y escribirla, es necesario implementarla, es decir, ponerla en funcionamiento, para esto se da clic derecho sobre la biblioteca y se escoge “Deploy”.

Se abrirá un cuadro de diálogo mostrándonos el progreso de la implementación llamado “Deployment Progress”, una vez completado se da clic en “Close” y listo, la nueva variable estará lista para ser usada por *Vision Builder* (ver la Figura 4.43).

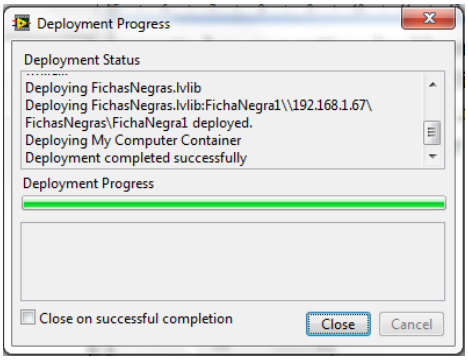


Figura 4.43 Cuadro de diálogo Deployment Progress.

Se debe crear una variable para cada eje coordenado de las fichas y una variable para el número de fichas de cada color, se pueden crear las variables dentro de la biblioteca dando clic derecho en ella y escogiendo “Add” y luego “Variable”, o se pueden crear nuevas bibliotecas siguiendo el procedimiento anterior.

En éste caso, se tiene un máximo de dos fichas negras y dos fichas blancas, por lo que el número de variables usadas serían diez. En la

Tabla 2 se puede ver un ejemplo de las variables a usar.

No. De Variable	Posible Nombre	Tarea de la Variable
1	FichasNegras	Guardar el número de fichas negras que el programa de visión detecte.
2	FichasBlancas	Guardar el número de fichas blancas que el programa de visión detecte.
3	FichaNegra1_X	Guardar el valor de la posición en el eje X de la primera ficha negra que el programa de visión detecte.
4	FichaNegra1_Y	Guardar el valor de la posición en el eje Y de la primera ficha negra que el programa de visión detecte.
5	FichaNegra2_X	Guardar el valor de la posición en el eje X de la segunda ficha negra que el programa de visión detecte.
6	FichaNegra2_Y	Guardar el valor de la posición en el eje Y de la segunda ficha negra que el programa de visión detecte.
7	FichaBlanca1_X	Guardar el valor de la posición en el eje X de la primera ficha blanca que el programa de visión detecte.
8	FichaBlanca1_Y	Guardar el valor de la posición en el eje Y de la primera ficha blanca que el programa de visión detecte.
9	FichaBlanca2_X	Guardar el valor de la posición en el eje X de la segunda ficha blanca que el programa de visión detecte.
10	FichaBlanca2_Y	Guardar el valor de la posición en el eje Y de la segunda ficha blanca que el programa de visión detecte.

Tabla 2. Ejemplo de las variables compartidas a usar en *el* proyecto.

Una vez generadas todas las variables e implementadas en el proyecto (Hacer "Deploy" en cada biblioteca) se puede regresar a *Vision Builder*.

4.4 Exportar datos de *Vision Builder* a *LabVIEW*

Se deben agregar las variables compartidas al catálogo de variables de *Vision Builder*. Se da clic en el menú "Tools" y se escoge "Variable Manager", se abrirá un cuadro de diálogo del mismo nombre, se da clic en la pestaña "Network Variables" y de nuevo clic en el botón "Add" (ver la Figura 4.44).

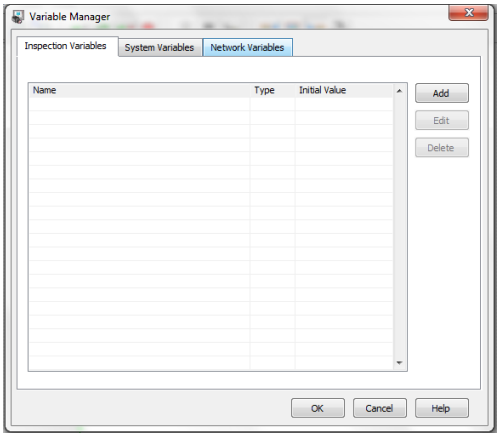


Figura 4.44 Cuadro de diálogo Variable Manager.

Se abrirá el cuadro de diálogo "Add Network Variable", se da clic en "Select Source Item"  , se abrirá de nuevo un cuadro de diálogo del mismo nombre. Se da clic en el icono de nuestro equipo  para expandir los elementos que contiene, dentro estarán las bibliotecas que contienen a las variables compartidas que se crearon con anterioridad, se escoge una y se da clic en "OK" (ver la Figura 4.45).

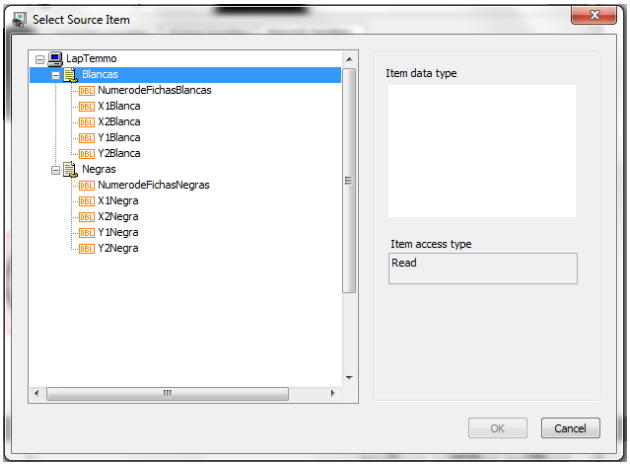


Figura 4.45 Cuadro de diálogo Select Source Item.

Se agregará la ruta de la variable en el cuadro de diálogo "Add Network Variable", se da clic en "OK", listo, la variable se agregará al catálogo de variables en *Vision Builder*. Se deben repetir estos pasos para agregar todas las demás variables.

En la configuración de *Vision Builder* nos quedamos en el reconocimiento de fichas, ahora lo que se hará será grabar los datos que genere *Vision Builder* para ocuparlos en *LabVIEW*. Se selecciona el paso tres del "diagrama de inspección" (ver la Figura 4.46) para que el nuevo paso de inspección quede después de "detectar fichas blancas".

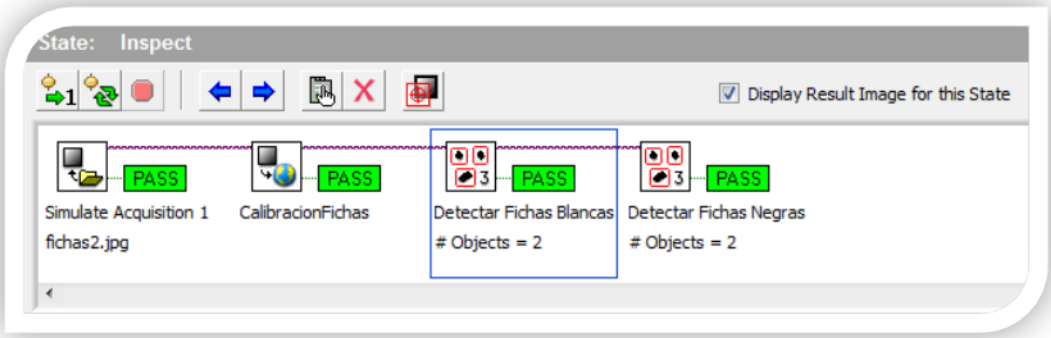


Figura 4.46 Diagrama de Inspección en *Vision Builder AI*.

Se selecciona "Use Additional Tools" en la sección "Inspection Steps" y se da clic en "Set Variable" .

Se abrirá un cuadro de diálogo en donde se le dará un nombre al Inspection Step, por ejemplo, "Guardar variables 1". Más abajo se veran las variables del catálogo de *Vision Builder*, ahí deben de estar las variables compartidas que se agregaron anteriormente.

Se da clic en la variable que se usará para guardar el número de fichas blancas, en éste caso es "NumerodeFichasBlancas", más abajo en la sección "Operation" se podrá escoger lo que la variable guardará, se da clic en "Set to Measurement", esto significa que se asignará uno de los valores que el programa ha medido o calculado de la inspección.

A la derecha se activará una lista en la que se escogerá el valor que se requiera, en éste caso se escogerá "Detectar Fichas Blancas - # Objects", esto asignará a la variable el número de fichas blancas que la inspección ha detectado (Ver la Figura 4.47).

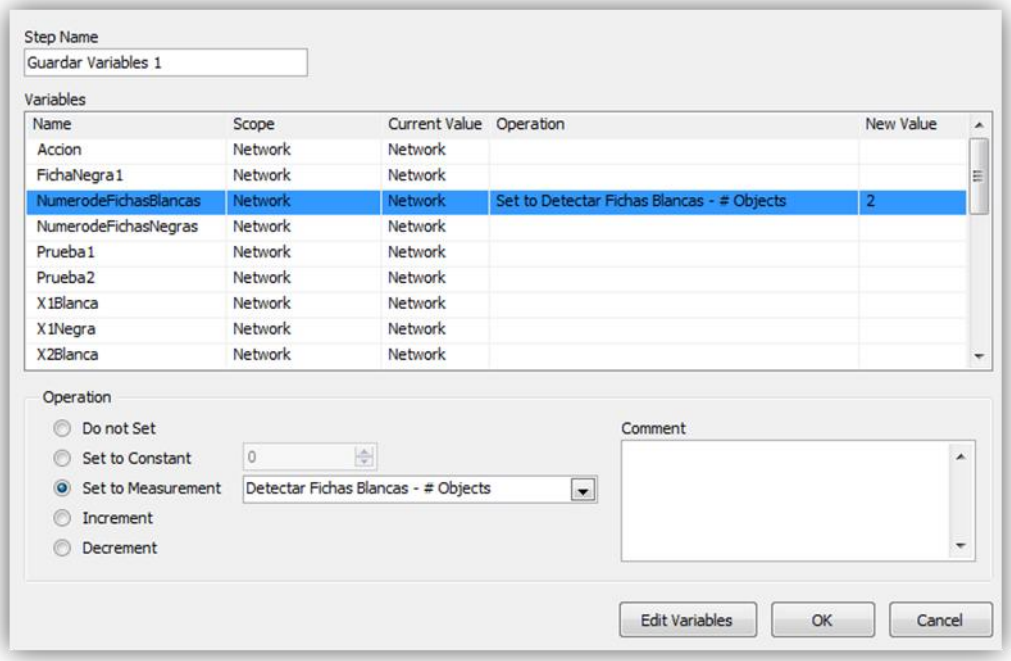


Figura 4.47 Cuadro de diálogo Set Variable.

A continuación elegimos la variable en donde se guardará la coordenada X de la primera ficha blanca, en éste caso se escogerá "X1Blanca", en "Operation" se escoge "Set to Measurement" y de la lista se elige "Detectar Fichas Blancas – Object [1].X Position (Calibrated)", se escoge de igual forma la variable que guardará la coordenada X de la segunda ficha, en éste caso "X2Blanca", se elige ahora "Detectar Fichas Blancas – Object [2].X Position (Calibrated)", se hace lo mismo para las otras variables que guardarán las coordenadas Y, se da clic en "OK".

Ahora se agregará otro paso para guardar la información de las fichas negras. Esta vez se selecciona el quinto y último paso del diagrama de inspección y se da clic en el botón "Set Variable" del menú de "Inspection Steps" como se había hecho anteriormente. (Ver la Figura 4.48)

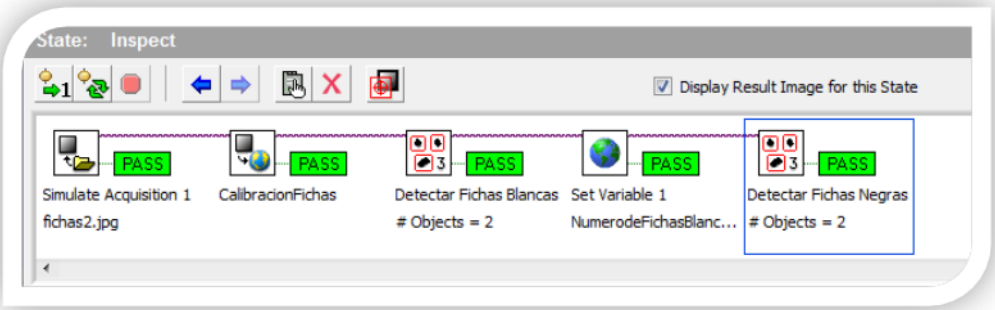


Figura 4.48 Diagrama de Inspección en *Vision Builder AI*.

La configuración del nuevo paso "Set Variable" debe verse más o menos como en la Figura 4.49. Se da clic en "OK".

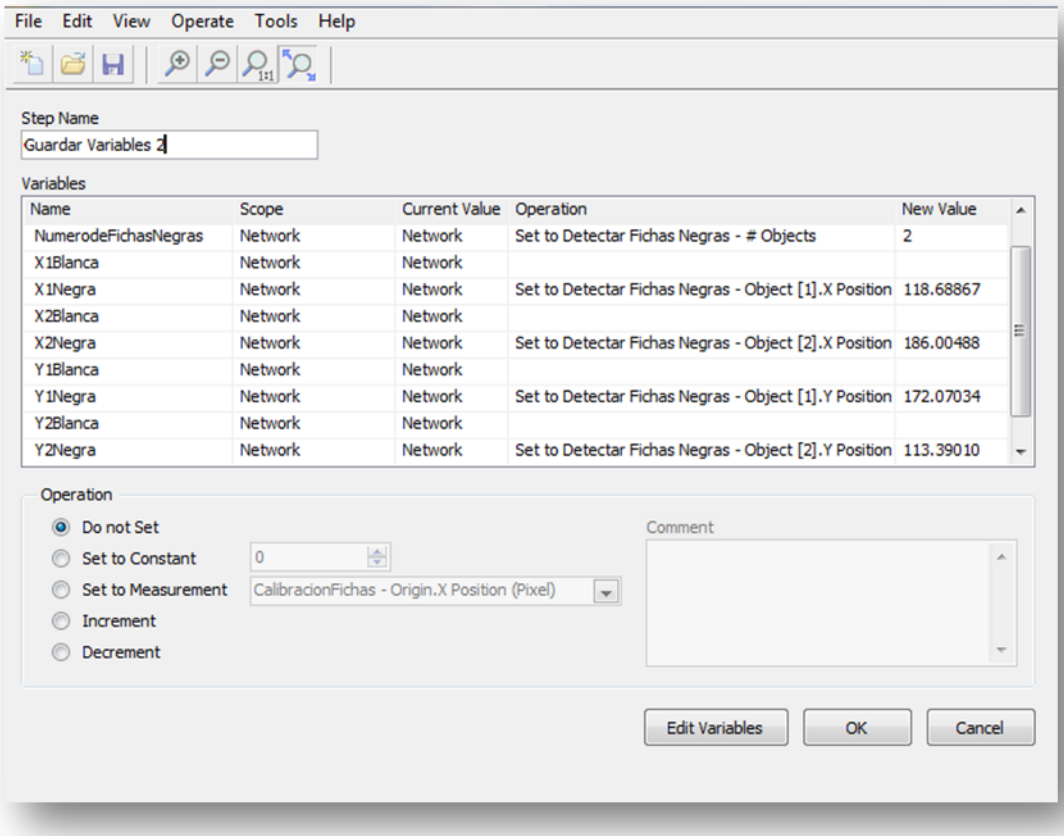


Figura 4.49 Cuadro de diálogo Set Variable en *Vision Builder AI*.

Listo, ahora se puede ejecutar la inspección y exportar los datos a *LabVIEW*. Para hacerlo se da clic en el botón "Run State Once"  , esto

ejecutará la inspección una vez y al darle clic de nuevo cambiará automáticamente a la siguiente imagen y la inspeccionará proporcionándonos los datos que se necesitan.

Para usar los datos contenidos en las variables compartidas abrimos el proyecto en *LabVIEW* y se da doble clic en el VI guardado anteriormente, se arrastra la variable desde el proyecto hasta el espacio libre del "diagrama de bloques" del VI. Ahora cada vez que se ejecute la inspección en *Vision Builder AI* los datos de las variables se actualizarán y se podrá trabajar con las coordenadas actualizadas de cada ficha para que la mesa de coordenadas pueda encontrarlas y organizarlas en *SolidWorks*.

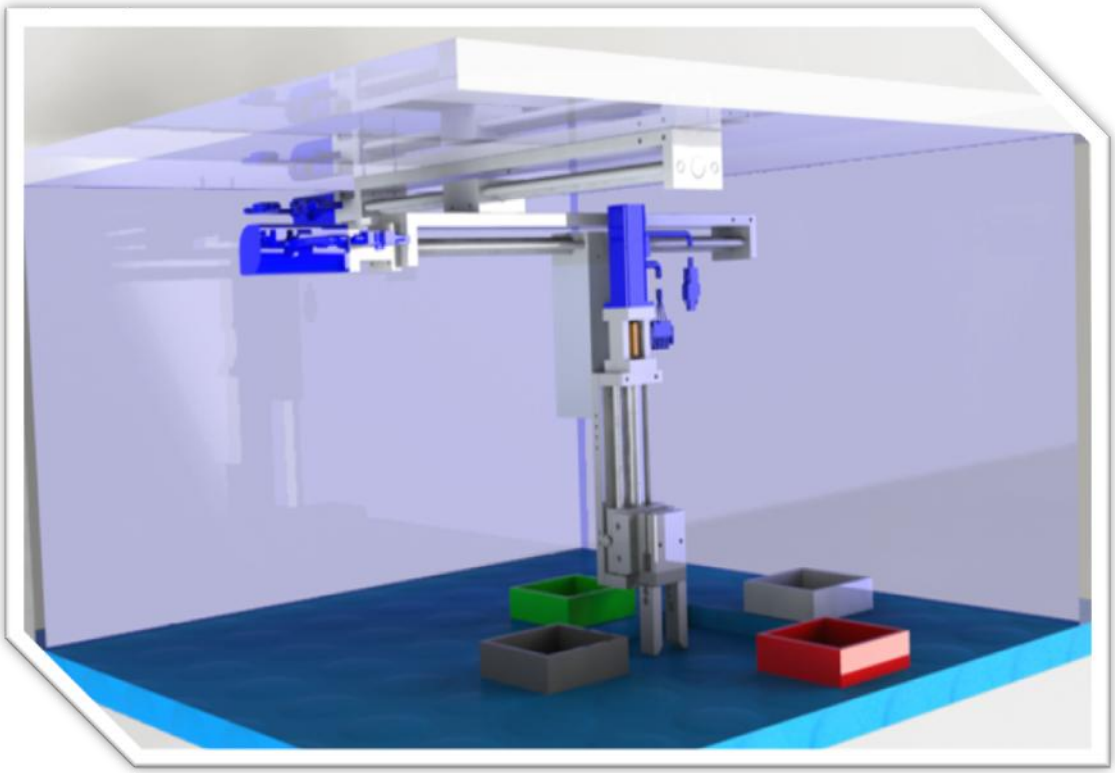


Figura 4.50 Render de la mesa de coordenadas realizado en PhotoView 360 de *SolidWorks*.

CAPÍTULO 5

Pruebas y resultados

En este capítulo se presentarán las pruebas realizadas a la mesa de coordenadas virtual y a su programación a lo largo del desarrollo del proyecto.

5.1 Prueba de la programación de un eje en *NI LabVIEW* para su animación en *SolidWorks* utilizando *NI SoftMotion*

Esta pequeña prueba fue la que más problemas presentó, ya que hay muy poca documentación de parte de *National Instruments* para la puesta en marcha de la comunicación entre *SolidWorks* y *NI SoftMotion*, en los ejemplos que *National Instruments* aporta en su página de internet todo funcionó a la perfección, pero al momento de implementar el sistema a uno de los ejes, éste no se movía. Se contactó varias veces con ingenieros de NI mandando videos, ensambles y configuraciones para que se nos proporcionara ayuda pero sólo nos repetían lo mismo que dice su documentación. Después de intentar varias cosas todo se solucionó con una simple actualización del software *SoftMotion* de NI.

En la prueba el eje tiene que moverse 20 cm y regresar a su punto de origen.

Como se puede ver en el video "Prueba sobre un eje"[1] la prueba fue satisfactoria, el eje se desplazó 20 cm y regresó al punto de origen.

[1] Prueba documentada en video en el canal de Youtube llamado "MCVUNAM" http://www.youtube.com/user/MCVUNAM

5.2 Prueba de la programación de tres ejes en *NI LabVIEW* para su animación en *SolidWorks* utilizando *NI SoftMotion*

Esta prueba es muy parecida a la anterior, pero ahora son tres ejes los que se mueven al mismo tiempo. Esta prueba depende mucho del hardware ya que consume muchos recursos del equipo y es muy probable que el programa *SolidWorks* deje de funcionar durante la prueba.

La documentación de *SolidWorks* recomienda no dar instrucciones a más de un eje al mismo tiempo, sino esperar a que un eje termine su tarea para dar la siguiente instrucción de movimiento.

En la prueba, cada eje debe de moverse 20 cm y regresar al punto de origen.

Como se puede ver en el video "Prueba sobre tres ejes"[2] la prueba fue satisfactoria, cada eje se desplazó 20 cm y regresó al punto de origen.

5.3 Prueba del guardado de los valores generados en la Inspección en *Vision Builder AI* en las Variables Compartidas

En esta prueba simple sólo se comprueba la fiabilidad del guardado y la actualización de cada una de las variables compartidas realizada por *Vision Builder AI*.

La comprobación se hizo visualmente desde el programa LabVIEW en donde se observan las diez variables listadas en la

---

[2] Prueba documentada en video en el canal de Youtube llamado "MCVUNAM" http://www.youtube.com/user/MCVUNAM

Tabla 2.

Como se puede ver en el video "Prueba actualización de variables compartidas"[3] la prueba fue satisfactoria, cada una de las variables fue guardada y actualizada correctamente en cada inspección.

5.4 Prueba del funcionamiento del sistema completo

Esta prueba consiste en poner en marcha el sistema completo, se empieza por ejecutar una primera inspección en *Vision Builder AI* para que *LabVIEW* obtenga las coordenadas del primer grupo de fichas. *LabVIEW* manda las ordenes a cada uno de los ejes (uno tras de otro, nunca simultáneamente, de acuerdo a las recomendaciones de la documentación de *SolidWorks* para evitar fallos). *SolidWorks* recibe las instrucciones y comienza la animación, el primer eje se mueve y se posiciona en la coordenada X recibida de la primera ficha negra detectada, el eje 2 se mueve a continuación a la coordenada Y, una vez ubicadas las coordenadas de la primera ficha el eje 3 se mueve para colocarse en posición de agarre y que el gripper o pinzas puedan tomar la ficha, el eje 3 regresa a su posición inicial simulando que tiene la ficha sujeta, el eje 2 se mueve a las coordenada Y del depósito de fichas negras y a continuación el eje 1 se mueve a la coordenada X del mismo depósito. Una vez posicionado el eje 3 desciende de nuevo simulando la colocación de la ficha en el depósito y por último regresa a su posición inicial para quedar en espera de las coordenadas de la siguiente ficha, estos pasos deben de repetirse correctamente para cada una de las fichas en cada una de las tres imágenes.

Como se puede ver en el video "Prueba del sistema completo"[4] la prueba fue satisfactoria, la simulación del traslado de cada una de las fichas en su depósito correspondiente fue correcta.

[3] Prueba documentada en video en el canal de Youtube llamado "MCVUNAM" http://www.youtube.com/user/MCVUNAM

[4] Prueba documentada en video en el canal de Youtube llamado "MCVUNAM" http://www.youtube.com/user/MCVUNAM

CAPÍTULO 6

Conclusiones y trabajo a futuro

No cabe duda que los modelos virtuales rápidamente reemplazarán a los modelos reales, ya que representan grandes ventajas a los diseñadores, a las empresas, al usuario y al medio ambiente.

En lo que respecta al futuro de este proyecto, no me es difícil imaginar que muy pronto los alumnos puedan estar haciendo prácticas de laboratorio con modelos virtuales para probar desde su diseño mecánico hasta su control por computadora.

Con respecto a las mejoras que se le pueden hacer al proyecto, la lista es bastante extensa, ya que depende de varios factores, por ejemplo el enfoque que se le quiera dar a las prácticas o la posible construcción del modelo físico, además del claro y obvio avance del software usado y las nuevas propuestas que se presentarán en los próximos años.

Las mejoras que se pueden poner en práctica inmediatamente son:

- Mejorar el diseño mecánico
- Mejorar el diseño de detalle
- Adentrarse más en las funciones disponibles en *NI SoftMotion*
- Mejorar la programación en *LabVIEW* para hacerla más fluida y que pueda ordenar más de un tablero de fichas sin detenerse
- Para poder usar más de un eje al mismo tiempo en la simulación será necesario trabajar en una computadora de última generación con un buen procesador y una gran cantidad de memoria RAM, ya que el software usado en el proyecto consume gran cantidad de recursos

En caso de que se quiera construir el modelo físico se recomienda:

- Agregar al diseño mecánico una cámara especializada para capturar imágenes en tiempo real. Puede estar unida al tercer eje o fija en cualquier parte de la estructura en donde pueda obtener imágenes de las fichas o los objetos a ordenar.
- La programación debe mudarse a un FPGA para realizar operaciones de control en tiempo real.

Se espera que este proyecto sea de utilidad para maestros y alumnos y que puedan utilizar esta tesis como un buen medio de introducción para el modelado virtual de sus dispositivos y máquinas.

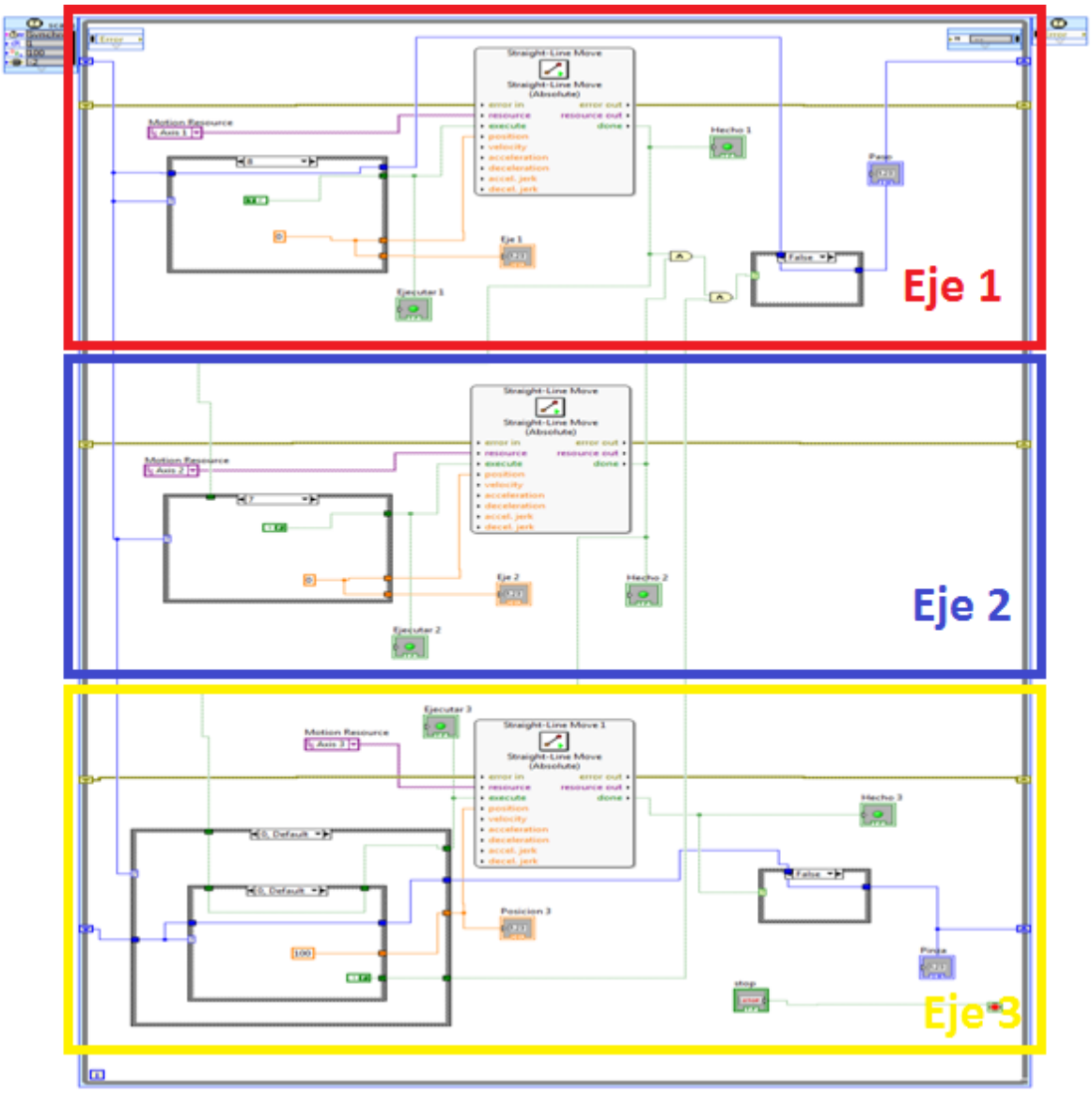
En unos años el modelo virtual responderá a factores externos perfectamente igual a como respondería un modelo real, arrojando datos interesantes y muy útiles para la construcción de máquinas, dispositivos y productos en general, en menor tiempo, con mayor calidad y utilizando un mínimo de recursos.

Los diseños serán construidos de manera más eficiente y precisa, con cada mm de material como se había planeado, saliendo prácticamente de una computadora a las manos del usuario.

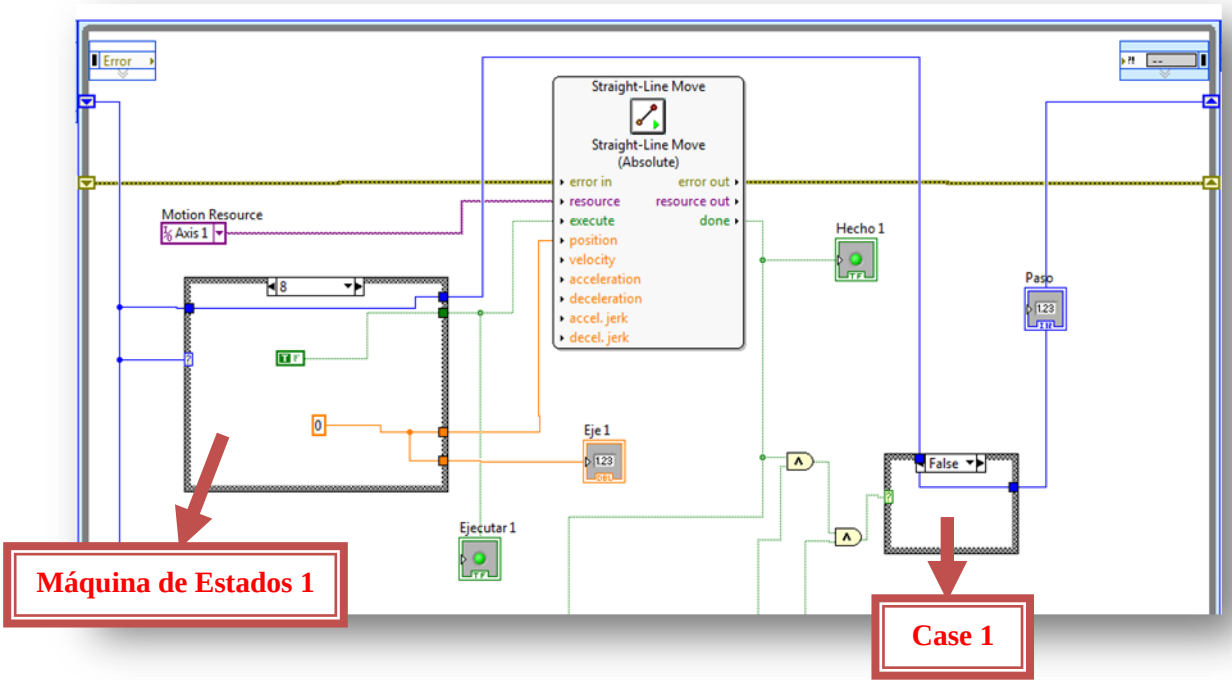
APÉNDICE A

Programación gráfica de la mesa de coordenadas

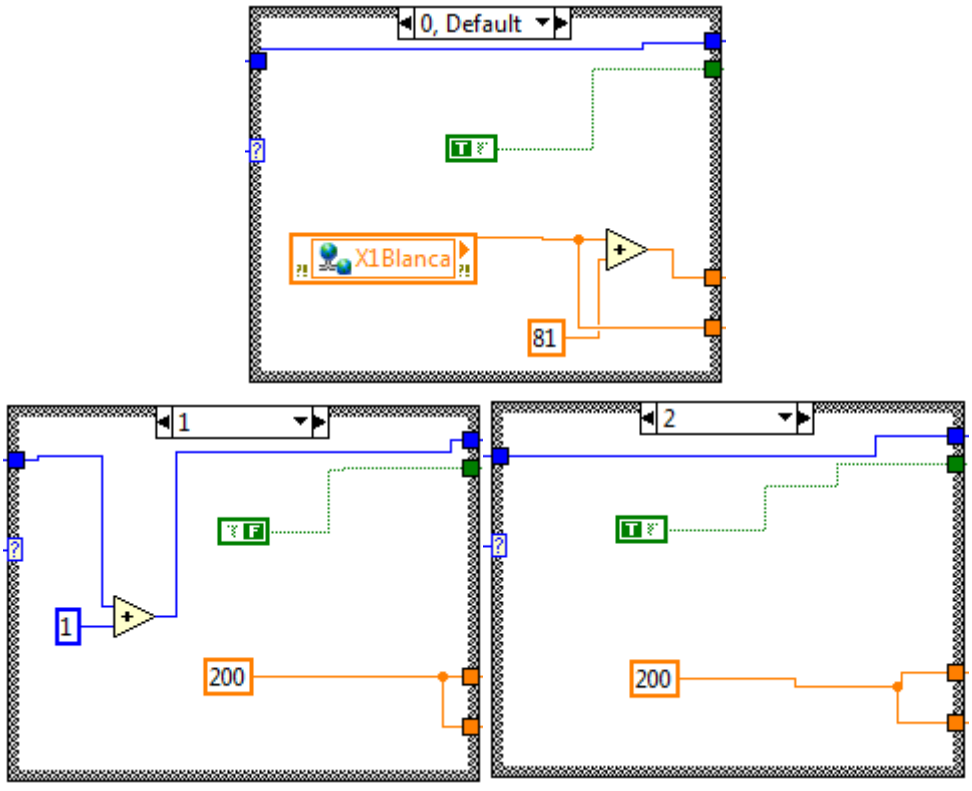
A continuación se presenta la programación que se realizó en *NI LabVIEW*. Se muestran imágenes para detallar cada uno de los casos en las máquinas de estados de cada eje.

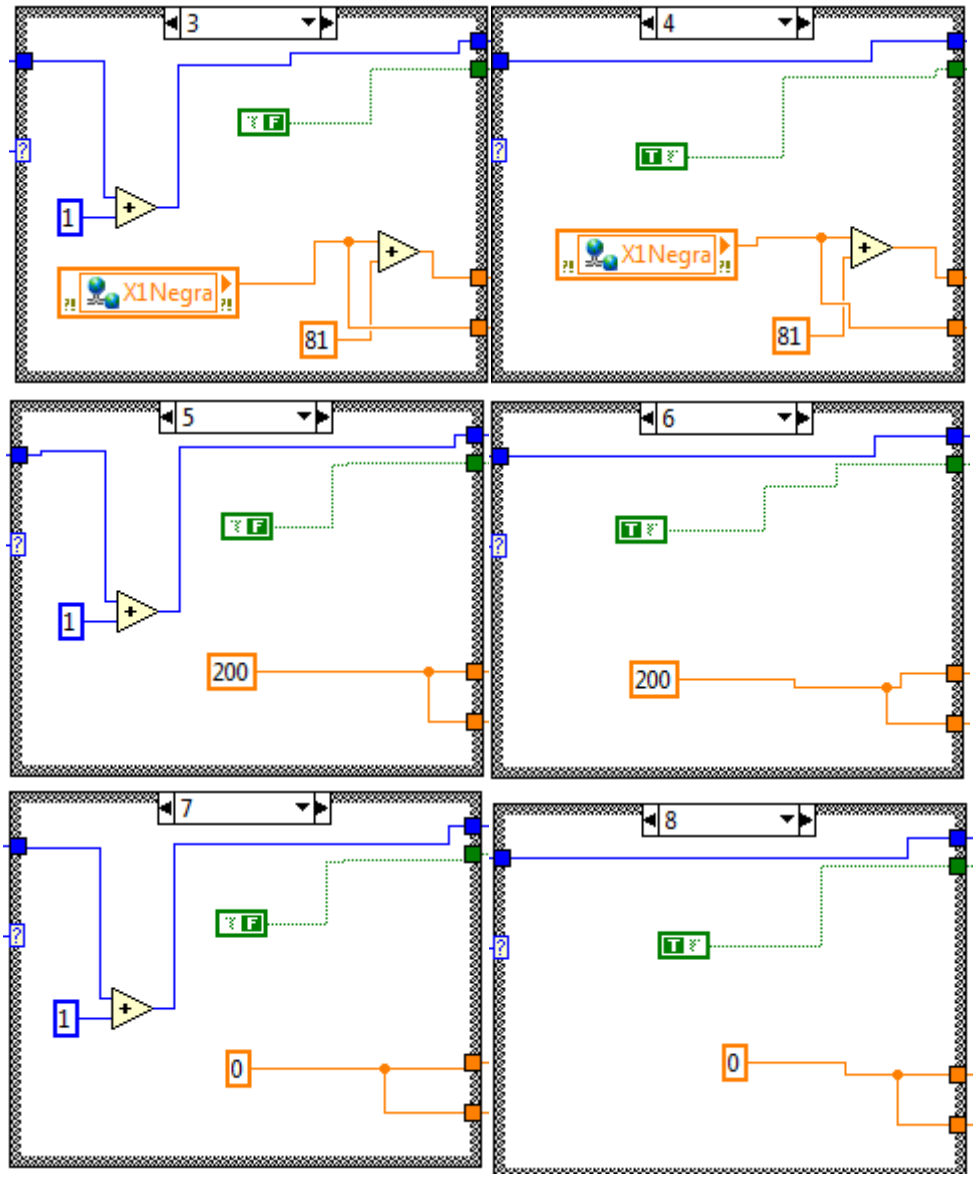


Eje 1

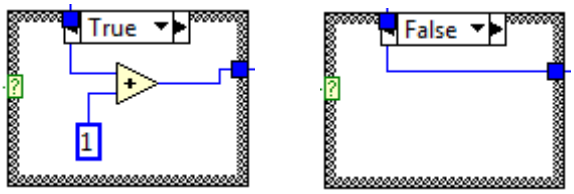


Máquina de Estados 1

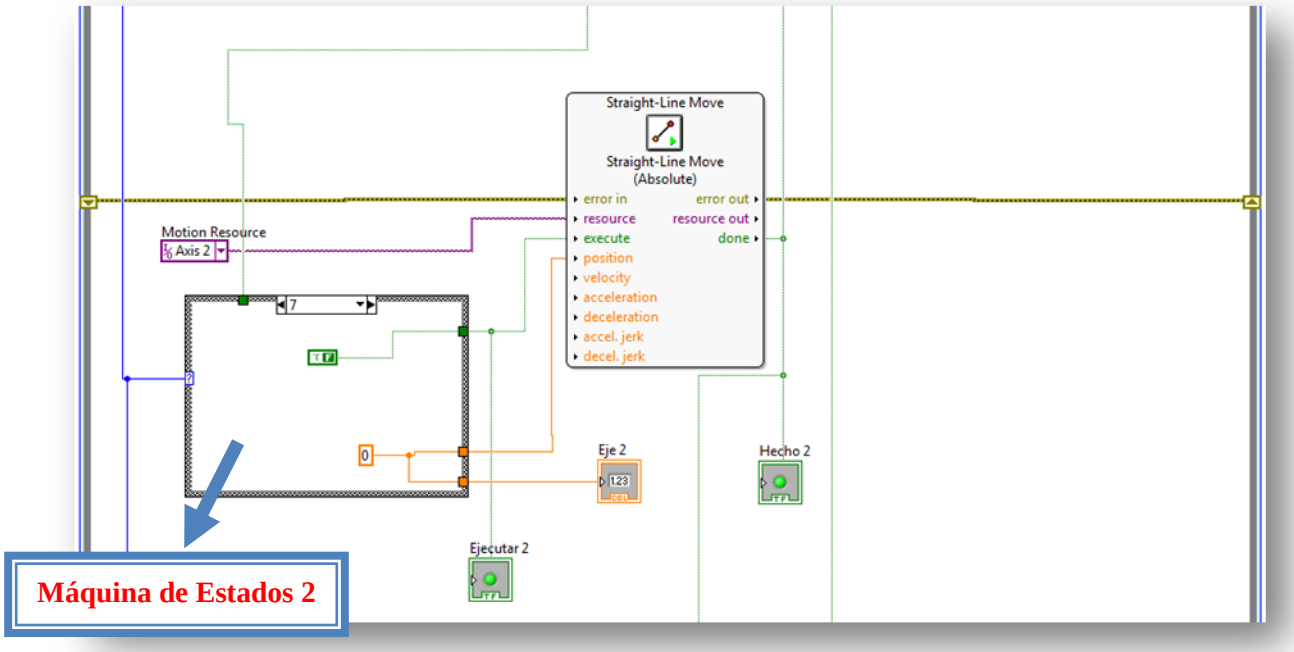



3
F
1
X1Negra
81
4
T
X1Negra
81
5
F
1
200
6
T
200
7
F
1
0
8
T
0

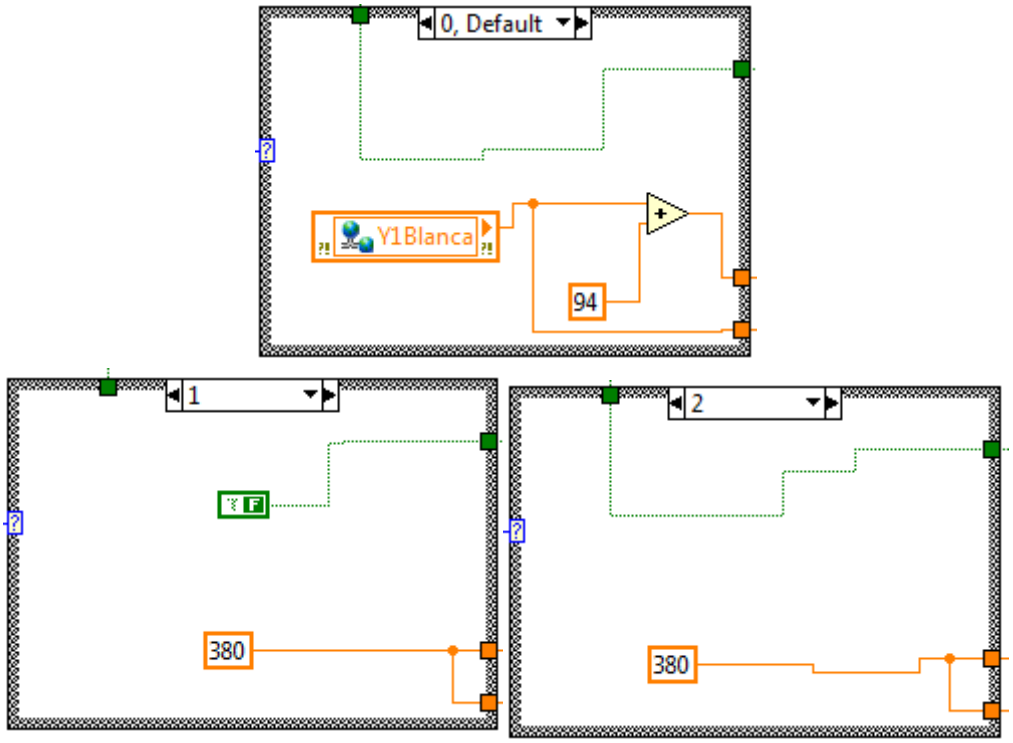
Case 1

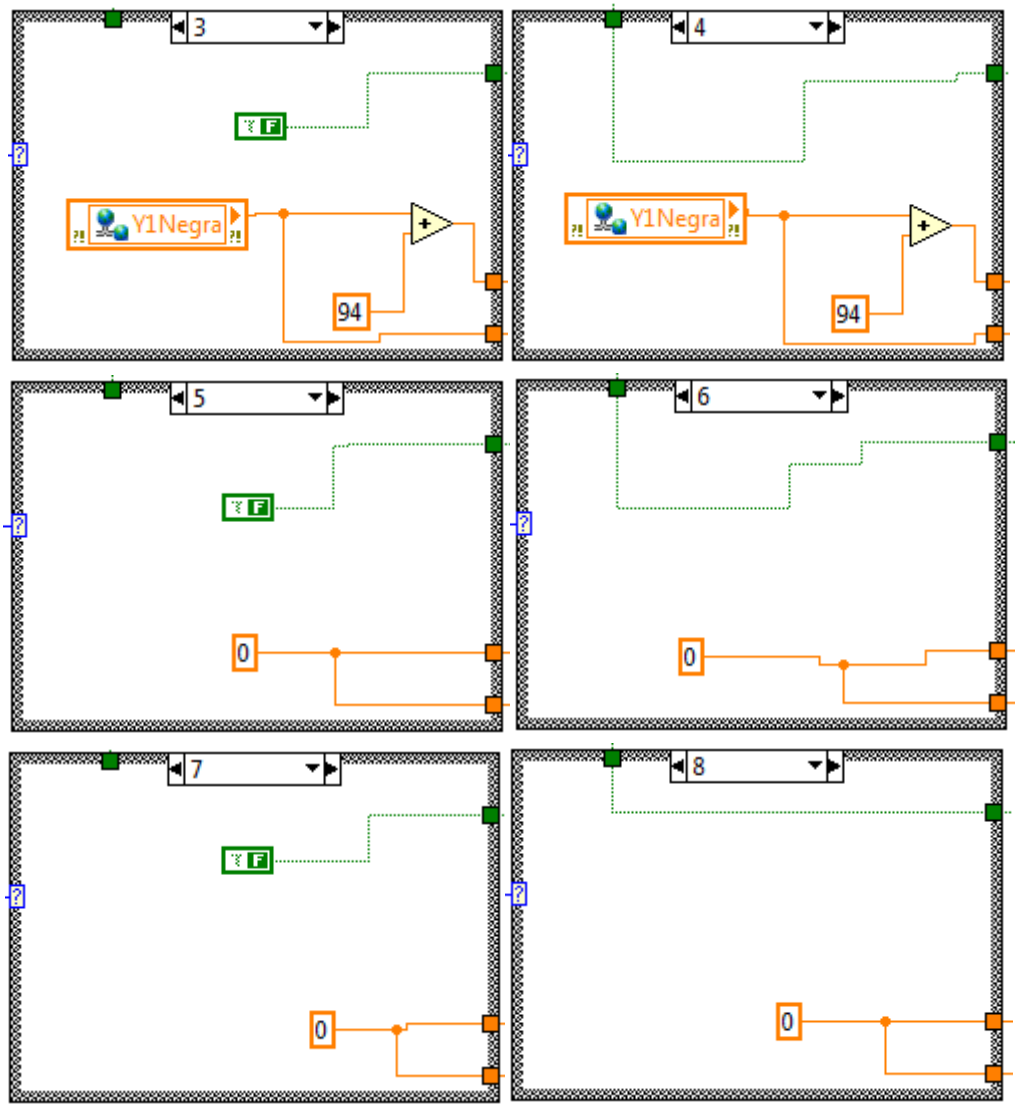

True
1
False

Eje 2

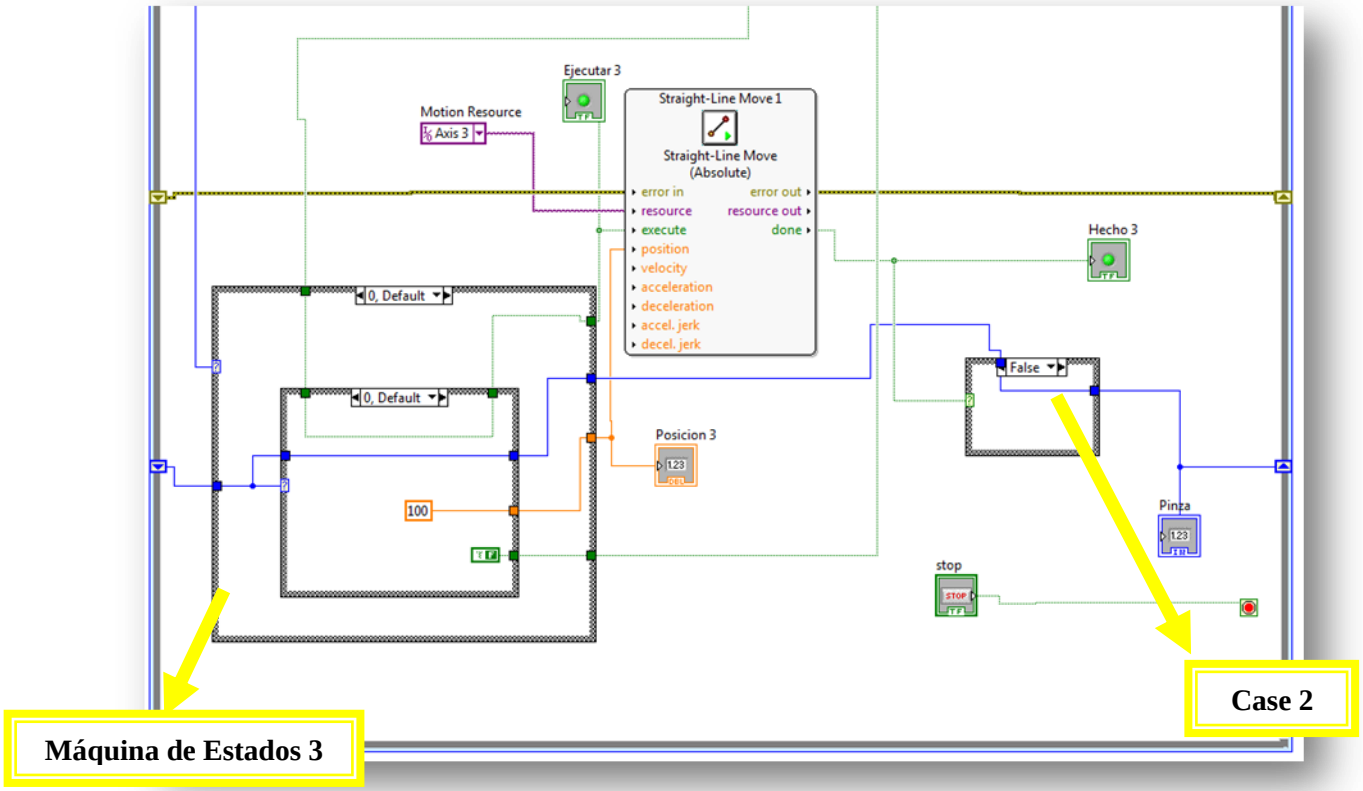


Máquina de Estados 2

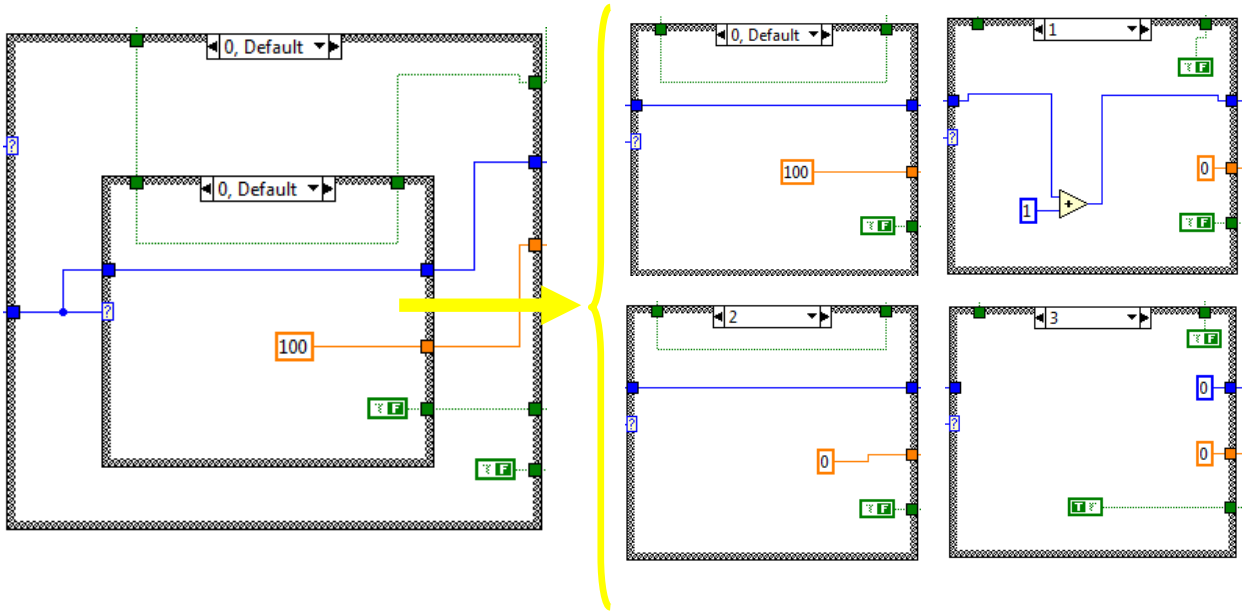


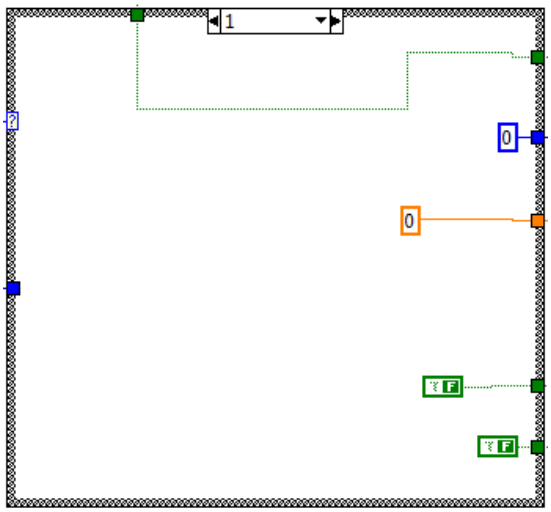

3
Y1Negra
94
4
Y1Negra
94
5
0
6
0
7
0
8
0

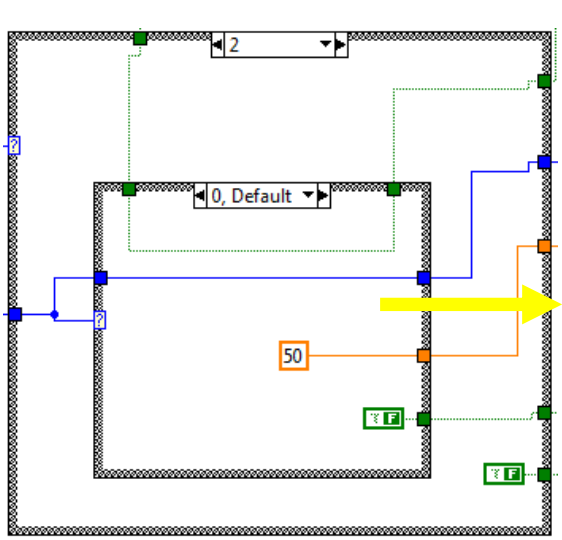
Eje 3

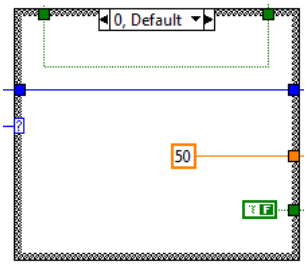


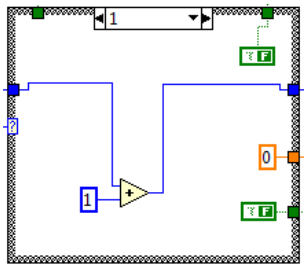
Máquina de Estados 3

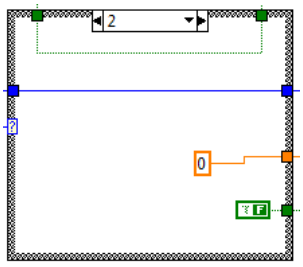


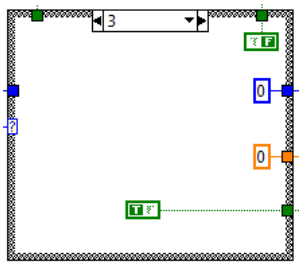

1
0
0

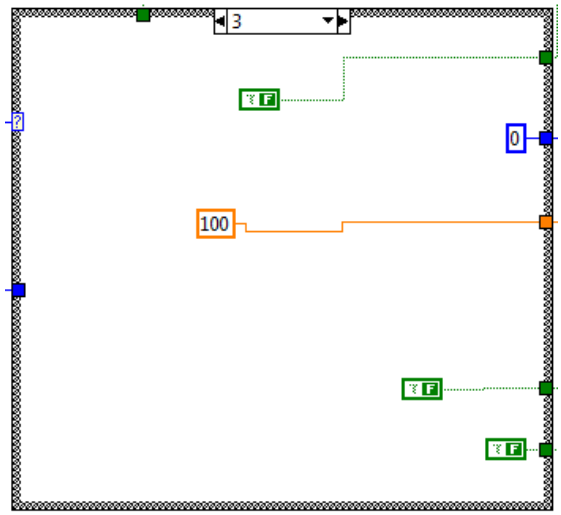

2
0, Default
50

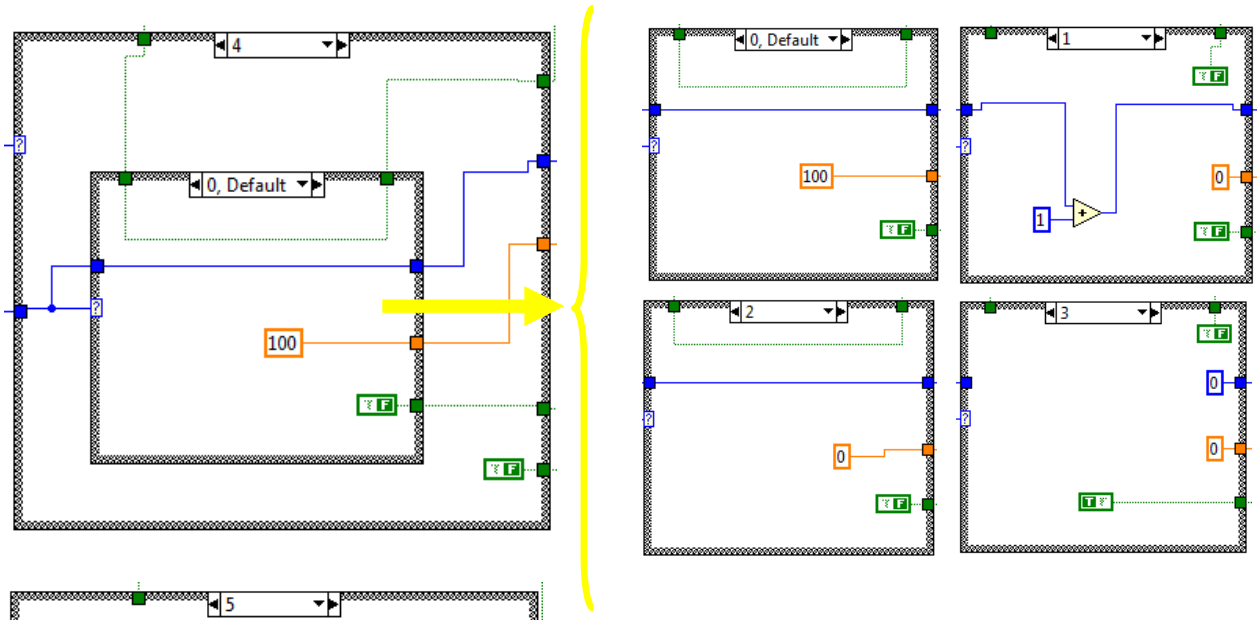

0, Default
50

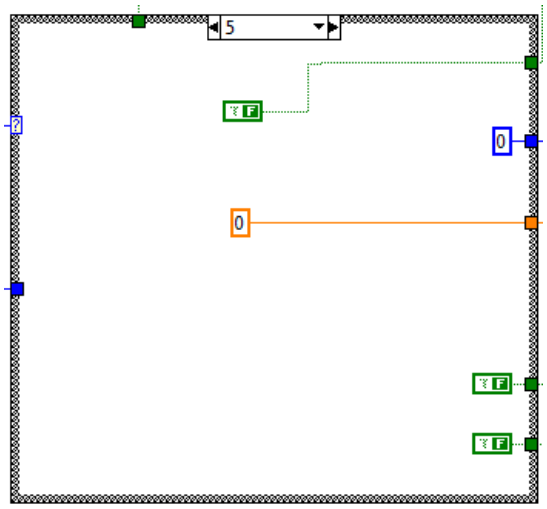

1
0
1

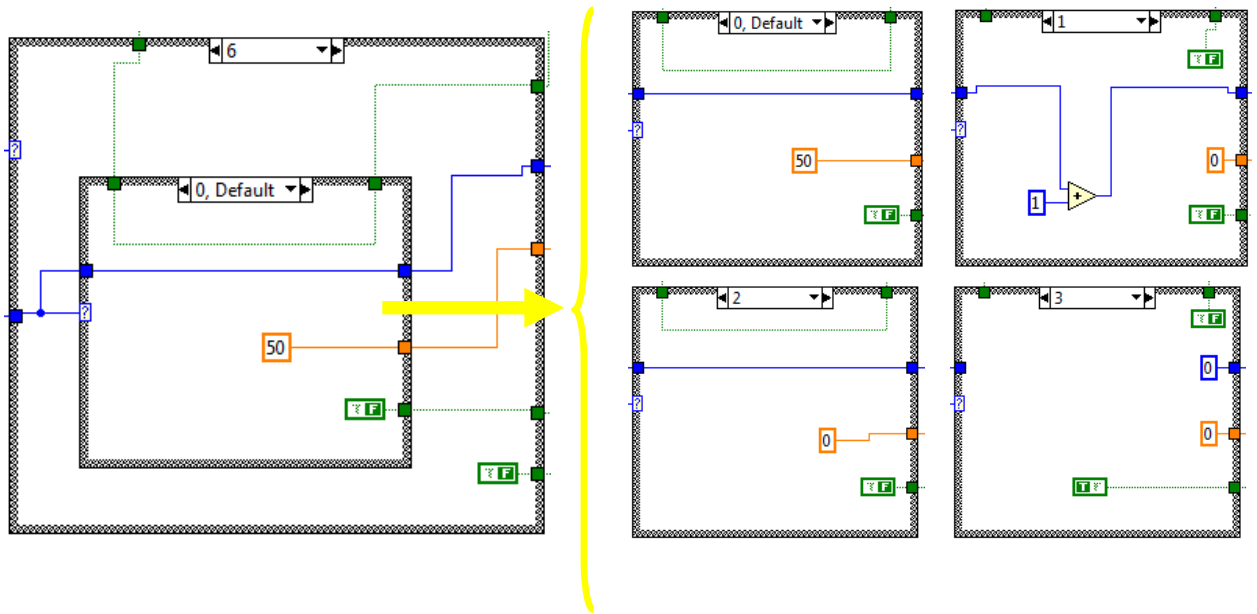

2
0

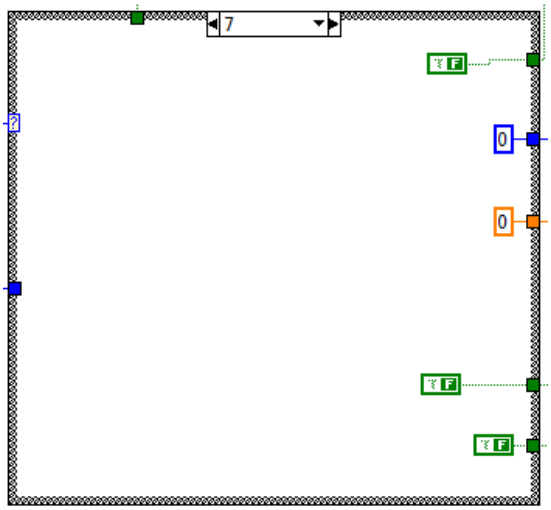

3
0
0

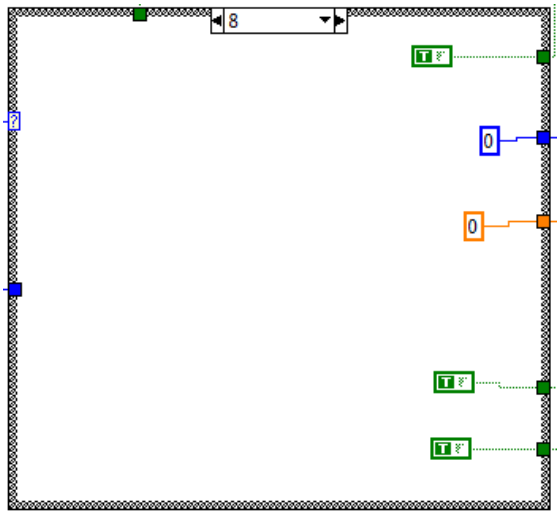

3
0
100


4
0, Default
100
0, Default
100
1
0
1
2
0
3
0
0

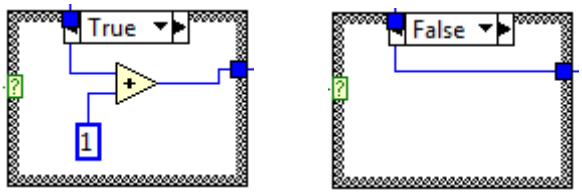

5
0
0


6
0, Default
50
0, Default
50
1
0
1
2
0
3
0
0





Case 2



APÉNDICE B

Hojas de especificaciones

Servomotor Yaskawa SGMAH-01B

Common Features			
Time Rating:	Continuous	Ambient Humidity:	20 to 80% (non-condensing)
Insulation:	Class B	Rated Speed:	3000rpm
Vibration:	15μm or less	Maximum Speed:	5000rpm
Withstand Voltage:	$1500V_{AC}$	Excitation:	Permanent magnet
Insulation Resistance:	10MΩ minimum at $500V_{DC}$	Drive Method:	Direct drive
Enclosure:	Totally-enclosed, self-cooled IP55 (except for shaft opening)	Mounting:	Flange-mounted
Ambient Temperature:	0 to 40ºC	Applicable Encoder:	13-bit incremental or 16-bit absolute encoder

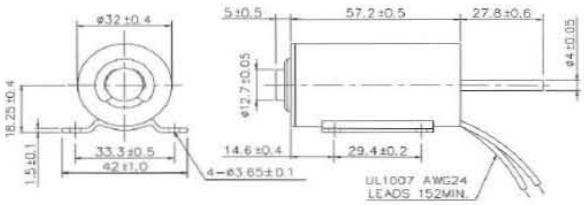
System Voltage	Model Number	Rated Output	Rated Torque[1]		Instantaneous Peak Torque[1]		Continuous Rated Current	Maximum Peak Current	Rated Angular Acceleration
		W (hp)	oz • in	N • m	oz • in	N • m	A_{RMS}	A_{RMS}	rad/s^2
100V AC	**SGMAH-01B**	100 (0.13)	45.1	0.318	135	0.96	2.4	7.2	87400

System Voltage	Model Number	Moment of Inertia				Holding Brake (at 20°C)			
		Motor without Brake		Motor with Brake		Capacity	Torque	Coil Resistance	Rated Current
		oz • in • s^2 x 10^{-3}	kg • m^2 x 10^{-4}	oz•in• s^2 x 10^{-3}	kg • m^2 x 10^{-4}	W	N • m	Ω	A
100V AC	**SGMAH-01B**	0.515	0.0364	0.635	0.0449	6	3.5	96	0.25

Solenoide SMT-3257S

SUN MAGNET®

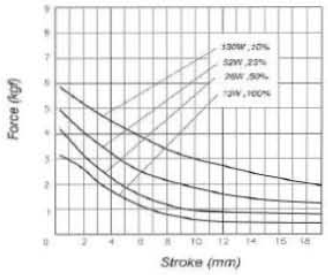
SMT-3257S Tubular Solenoids



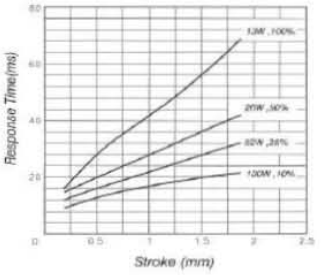
Plunger Weight 可動鐵心重量：70g
Total Weight 總重量：295g

Shown Energized 通電狀態 UNIT 單位：mm

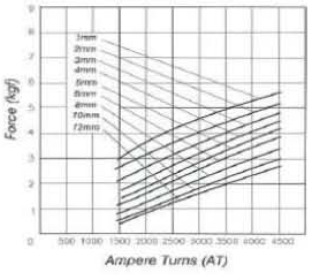
STROKE Vs FORCE
行 程 ／ 吸引力



STROKE Vs RESPONSE TIME
行 程 ／ 反應時間



AMPERE-TURNS Vs FORCE
安－匝 ／ 吸引力



COIL DATA 線圈資料

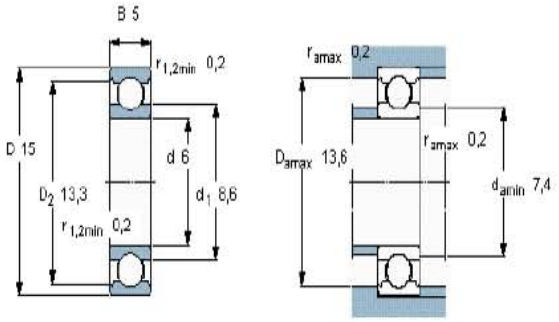
Heat Sink: 130×130×3mm (alumimum)
散熱片：130 × 130 × 3mm (鋁)

Duty cycle (%) = "ON" time / "ON" time + "OFF" time X 100% 作動週期 (%) = ON 時間 / ON 時間 + OFF 時間 X 100%			Continuous (100%) 連續 (100%)	Or less (50%) 間斷 (50%)	Or less (25%) 間斷 (25%)	Or less (10%) 間斷 (10%)
MAX. "on" time in seconds 最大 ON 時間 (秒)			∞	390	60	18
Watts at 20°C 瓦特數 (W) (在 20°C)			13	26	52	130
Ampere-turns at 20°C AT 匝 (在 20°C)			1500	2121	3000	4743
Type no. 型 號	Resistance (20°C) Ω ± 10% 阻值 (Ω) ± 10%	No. turns 繞線圈數	Volts DC 電 壓			
SMT-3257S06AA	2.77	692	6	8.5	12	19
SMT-3257S12AA	11.07	1385	12	17	24	38
SMT-3257S24AA	44.3	2768	24	34	48	76
SMT-3257S48AA	177.2	5535	48	68	96	152

Rodamiento skf 619/6

Rodamientos rígidos de bolas, de una hilera, no están obturados

Dimensiones principales			Capacidades de carga		Carga límite de fatiga	Velocidades		Masa	Designación
			dinámica	estática		Velocidad de referencia	Velocidad límite		
d	D	B	C	C_0	P_u				
mm			kN		kN	rpm		kg	-
6	15	5	1,24	0,475	0,02	100000	63000	0,0039	619/6

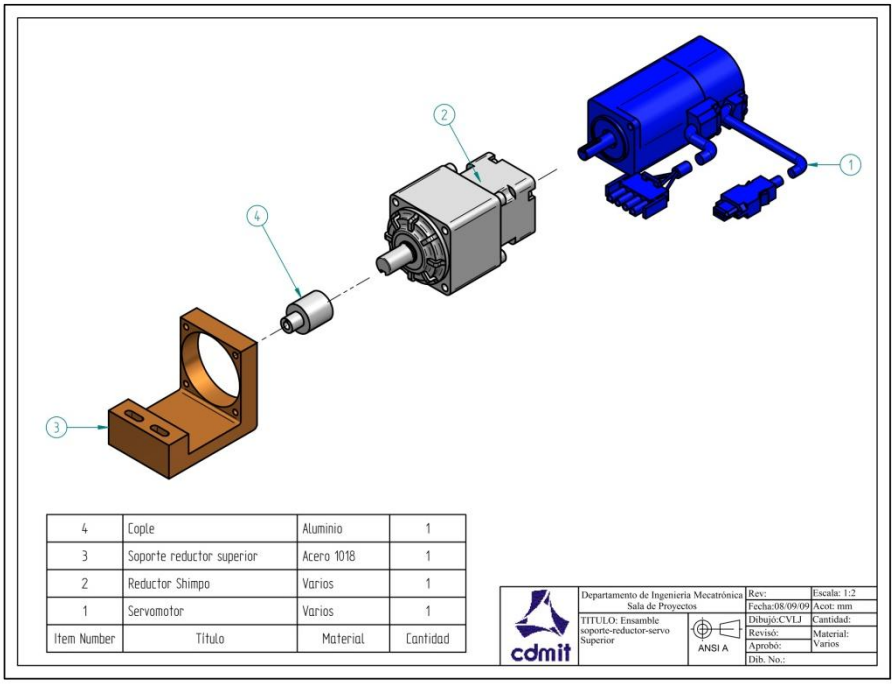


Factores de cálculo

k_r 0,02
f_0 10

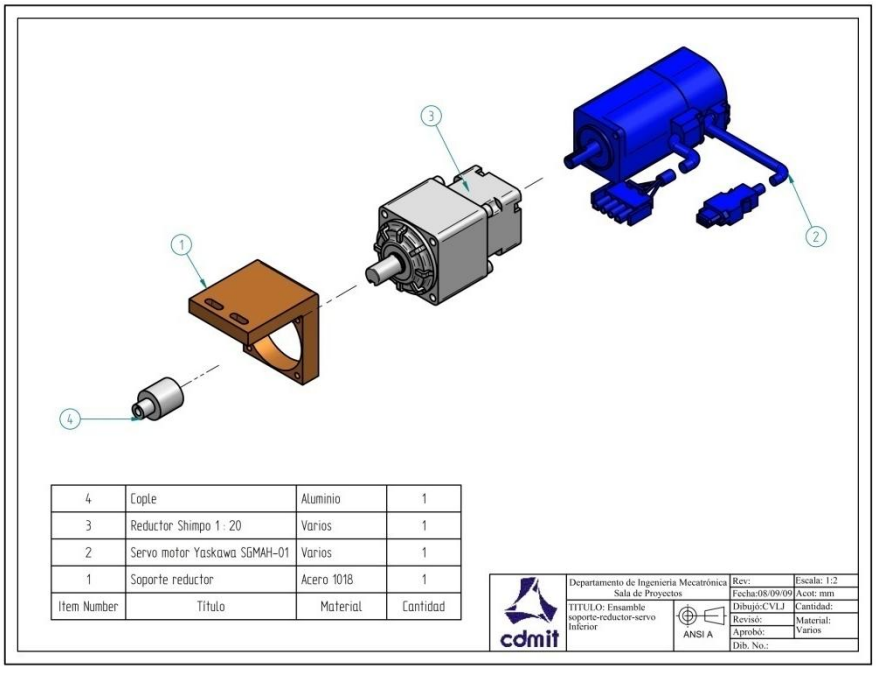
APÉNDICE C

Planos



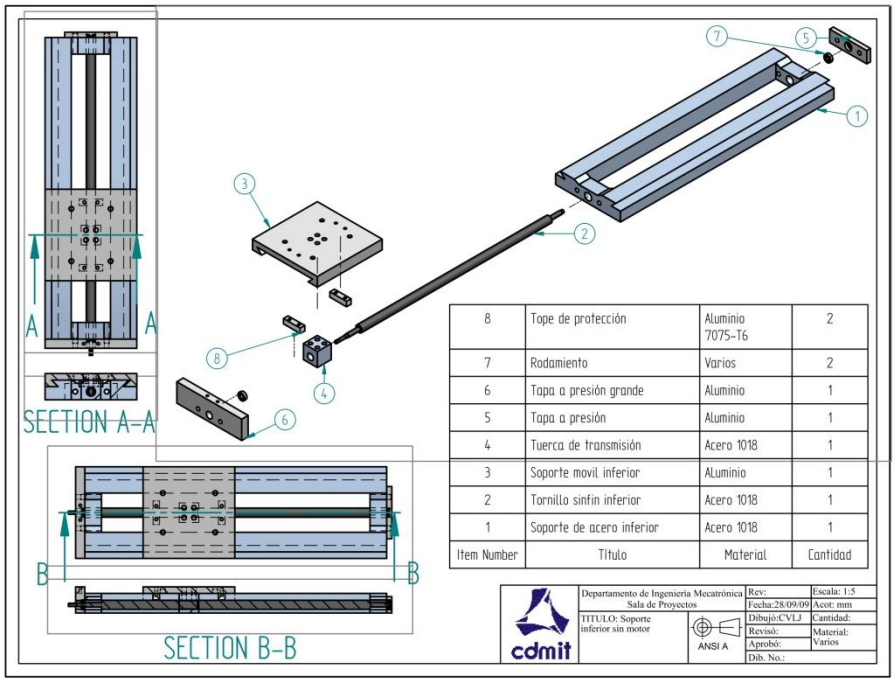
4	Cople	Aluminio	1
3	Soporte reductor superior	Acero 1018	1
2	Reductor Shimpo	Varios	1
1	Servomotor	Varios	1
Item Number	Título	Material	Cantidad

cdmit	Departamento de Ingeniería Mecatrónica Sala de Proyectos	Rev:	Escala: 1:2
		Fecha:08/09/09	Acot: mm
	TITULO: Ensamble soporte-reductor-servo Superior	Dibujó:CVLJ	Cantidad:
	ANSI A	Revisó:	Material: Varios
		Aprobó:	
		Dib. No.:	



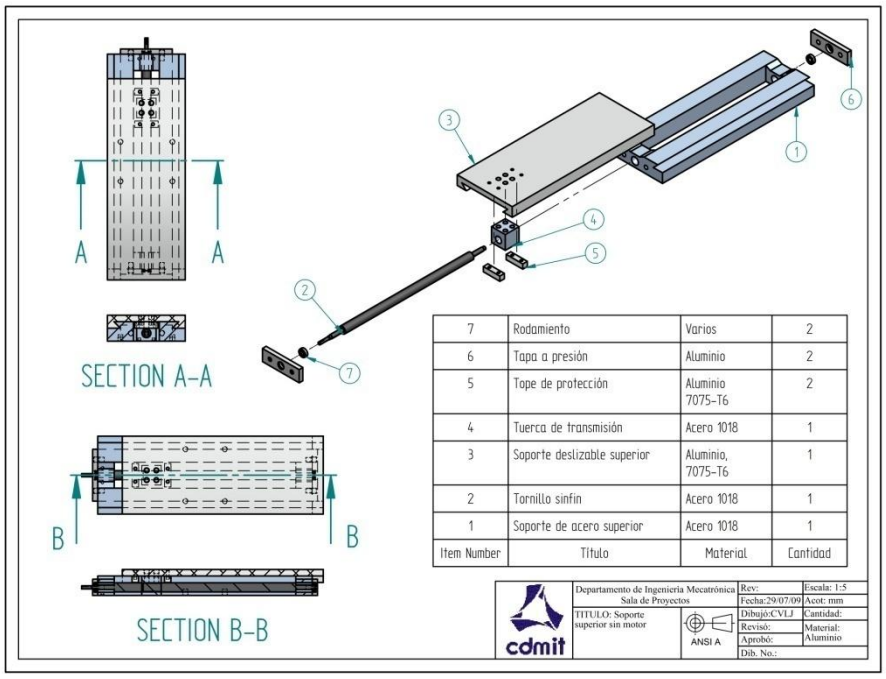
4	Cople	Aluminio	1
3	Reductor Shimpo 1 : 20	Varios	1
2	Servo motor Yaskawa SGMAH-01	Varios	1
1	Soporte reductor	Acero 1018	1
Item Number	Título	Material	Cantidad

cdmit	Departamento de Ingeniería Mecatrónica Sala de Proyectos	Rev:	Escala: 1:2
		Fecha:08/09/09	Acot: mm
	TITULO: Ensamble soporte-reductor-servo Inferior	Dibujó:CVLJ	Cantidad:
	ANSI A	Revisó:	Material: Varios
		Aprobó:	
		Dib. No.:	



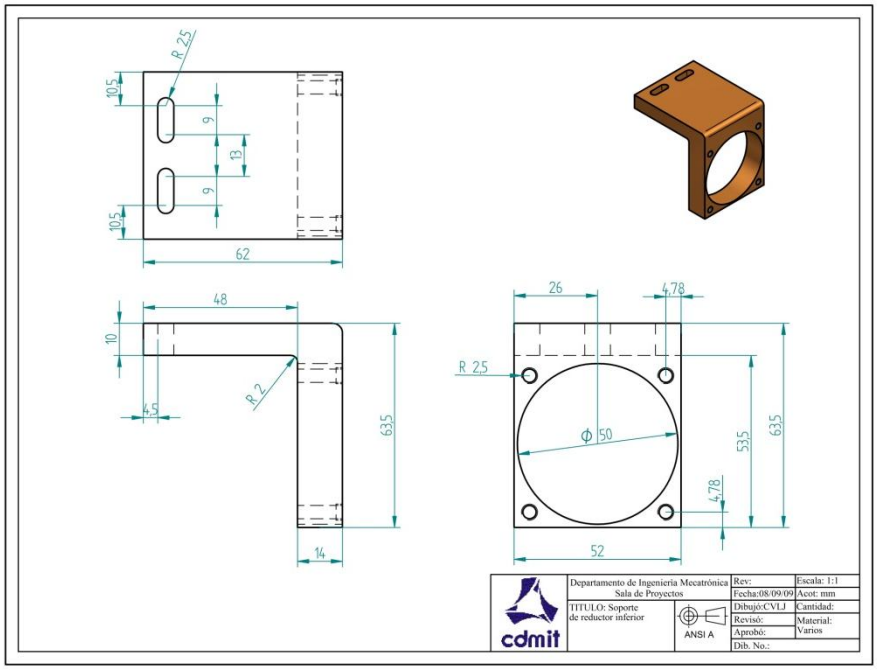
Item Number	Título	Material	Cantidad
8	Tope de protección	Aluminio 7075-T6	2
7	Rodamiento	Varios	2
6	Tapa a presión grande	Aluminio	1
5	Tapa a presión	Aluminio	1
4	Tuerca de transmisión	Acero 1018	1
3	Soporte movil inferior	ALuminio	1
2	Tornillo sinfin inferior	Acero 1018	1
1	Soporte de acero inferior	Acero 1018	1

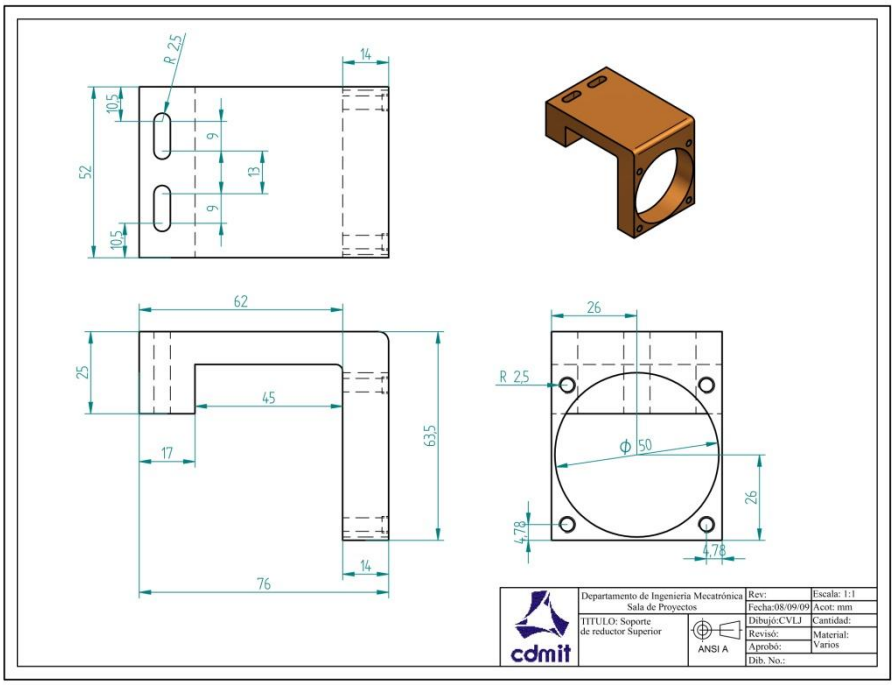
cdmit	Departamento de Ingenieria Mecatrónica Sala de Proyectos	Rev:	Escala: 1:5
		Fecha:28/09/09	Acot: mm
	TITULO: Soporte inferior sin motor	Dibujó:CVLJ	Cantidad:
	ANSI A	Revisó:	Material: Varios
		Aprobó:	
		Dib. No.:	



Item Number	Título	Material	Cantidad
7	Rodamiento	Varios	2
6	Tapa a presión	Aluminio	2
5	Tope de protección	Aluminio 7075-T6	2
4	Tuerca de transmisión	Acero 1018	1
3	Soporte deslizable superior	Aluminio, 7075-T6	1
2	Tornillo sinfin	Acero 1018	1
1	Soporte de acero superior	Acero 1018	1

cdmit	Departamento de Ingenieria Mecatrónica Sala de Proyectos	Rev:	Escala: 1:5
		Fecha:29/07/09	Acot: mm
	TITULO: Soporte superior sin motor	Dibujó:CVLJ	Cantidad:
	ANSI A	Revisó:	Material: Aluminio
		Aprobó:	
		Dib. No.:	


R 25
10.5
9
13
9
10.5
62
48
10
R 2
4.5
635
14
26
4.78
R 25
ϕ 50
535
635
4.78
52
Departamento de Ingeniería Mecatrónica
Sala de Proyectos
TITULO: Soporte
de reductor inferior
ANSI A
Rev:
Escala: 1:1
Fecha:08/09/09
Acot: mm
Dibujó:CVLJ
Cantidad:
Revisó:
Material:
Varios
Aprobó:
Dib. No.:
cdmit


R 25
14
10.5
9
13
9
10.5
52
62
25
45
17
635
14
76
26
R 25
ϕ 50
26
4.78
4.78
Departamento de Ingeniería Mecatrónica
Sala de Proyectos
TITULO: Soporte
de reductor Superior
ANSI A
Rev:
Escala: 1:1
Fecha:08/09/09
Acot: mm
Dibujó:CVLJ
Cantidad:
Revisó:
Material:
Varios
Aprobó:
Dib. No.:
cdmit

GLOSARIO

Ambiguo.- adj. Dicho especialmente del lenguaje: Que puede entenderse de varios modos o admitir distintas interpretaciones y dar, por consiguiente, motivo a dudas, incertidumbre o confusión.

Análisis de elementos finitos.- El análisis por elementos finitos (FEA por sus siglas en inglés para: *Finite Element Analysis*) es una técnica de simulación por computador usada en ingeniería. Usa una técnica numérica llamada Método de los elementos finitos (FEM).

Blu-ray.- También conocido como Blu-ray Disc o BD, es un formato de disco óptico de nueva generación de 12 cm de diámetro (igual que el CD y el DVD) para vídeo de gran definición y almacenamiento de datos de alta densidad

Booleano.- El tipo de dato lógico o *booleano* es en computación aquel que puede representar valores de lógica binaria, esto es 2 valores, valores que normalmente representan *falso* o *verdadero*.

CAD.- El diseño asistido por computadora, más conocido por sus siglas inglesas CAD (*computer-aided design*), es el uso de un amplio rango de herramientas computacionales que asisten a ingenieros, arquitectos y a otros profesionales del diseño en sus respectivas actividades

Calibrar.- Graduar exactamente un aparato o instrumento según una unidad de medida

Capitalismo.- El capitalismo es el orden social que resulta de la libertad económica en la disposición y usufructo de la propiedad privada sobre el capital como herramienta de producción.

Consumismo.- El consumismo puede referirse tanto a la acumulación, compra o consumo de bienes y servicios considerados no esenciales, como al sistema político y económico que promueve la adquisición competitiva de riqueza como signo de status y prestigio dentro de un grupo social. El consumo a gran escala en la sociedad contemporánea

compromete seriamente los recursos naturales y el equilibrio ecológico.

Control PID.- Un PID (Proporcional Integral Derivativo) es un mecanismo de control por realimentación que calcula la desviación o error entre un valor medido y el valor que se quiere obtener, para aplicar una acción correctora que ajuste el proceso.

Deploy.- Palabra inglesa que significa implementar, emplear o utilizar.

Eficiencia.- Es la capacidad de lograr un efecto o resultado en cuestión con el mínimo de recursos posibles viable

Embebido.- tr. Dicho de una cosa inmaterial: Incorporar, incluir dentro de sí a otra.

Encoder.- Un codificador rotatorio, también llamado codificador del eje o generador de pulsos, suele ser un dispositivo electromecánico usado para convertir la posición angular de un eje a un código digital, lo que lo convierte en una clase de transductor.

Exactitud.- En ingeniería, ciencia, industria y estadística, se denomina exactitud a la capacidad de un instrumento de medir un valor cercano al valor de la magnitud real.

FPGA.- Un FPGA (del inglés *Field Programmable Gate Array*) es un dispositivo semiconductor que contiene bloques de lógica cuya interconexión y funcionalidad puede ser configurada *'in situ'* mediante un lenguaje de descripción especializado.

Gripper.- Palabra inglesa que significa “agarradera” o “pinza”.

Hoja de especificaciones.- Un datasheet u hoja de especificaciones es un documento que resume el funcionamiento y otras características de un componente (por ejemplo, un componente electrónico) o subsistema (por ejemplo, una fuente de alimentación) con el suficiente detalle para ser utilizado por un ingeniero de diseño y diseñar el componente en un sistema.

Hipervínculo.- Un hipervínculo (también llamado enlace, vínculo, o hiperenlace) es un elemento de un documento electrónico que hace

referencia a otro recurso, por ejemplo, otro documento o un punto específico del mismo o de otro documento.

Ítem.- adv. c. U. para hacer distinción de artículos o capítulos en una escritura u otro instrumento, o como señal de adición.

Máquina de estados.- Se denomina máquina de estados a un modelo de comportamiento de un sistema con entradas y salidas, en donde las salidas dependen no sólo de las señales de entradas actuales sino también de las anteriores.

Memoria RAM.- RAM son las siglas de *random access memory*, un tipo de memoria de ordenador a la que se puede acceder aleatoriamente; es decir, se puede acceder a cualquier byte de memoria sin acceder a los bytes precedentes.

Metodología.- La Metodología, (del griego μετη *metà* "más allá", οδως *odòs* "camino" y λογος *logos* "estudio"), hace referencia al conjunto de procedimientos basados en principios lógicos, utilizados para alcanzar una gama de objetivos que rigen en una investigación científica o en una exposición doctrinal.

Precisión.- En ingeniería, ciencia, industria y estadística, se denomina precisión a la capacidad de un instrumento de dar el mismo resultado en mediciones diferentes realizadas en las mismas condiciones.

Prototipo.- Un prototipo también se refiere a cualquier tipo de máquina en pruebas, o un objeto diseñado para una demostración de cualquier tipo o también como el primer ejemplar de alguna cosa o máquina que servirá como modelo para crear otros de la misma clase.

Reductor de velocidad.- Dispositivo que sirve para que la velocidad de un motor se adapte a la velocidad necesaria para el buen funcionamiento una máquina. Además de esta adaptación de velocidad, se deben contemplar otros factores como la potencia mecánica a transmitir, la potencia térmica, rendimientos mecánicos (estáticos y dinámicos).

Render.- Renderizado (*render* en inglés) es un término usado en jerga informática para referirse al proceso de generar una imagen desde un

modelo 3D o dibujo CAD. Este término técnico es utilizado por los animadores o productores audiovisuales y en programas de diseño en 3D.

Robot.- Un robot es una entidad virtual o mecánica artificial. En la práctica, esto es por lo general un sistema electromecánico que, por su apariencia o sus movimientos, ofrece la sensación de tener un propósito propio

Servomotor.- Un servomotor (también llamado servo) es un dispositivo similar a un motor de corriente continua que tiene la capacidad de ubicarse en cualquier posición dentro de su rango de operación, y mantenerse estable en dicha posición.

Shift Register.- Un registro electrónico (*shift register* en inglés) es un dispositivo lógico secuencial capaz de almacenar varios bits de información.

Simulación.- Técnica numérica para conducir experimentos en una computadora digital. Estos experimentos comprenden ciertos tipos de relaciones matemáticas y lógicas, las cuales son necesarias para describir el comportamiento y la estructura de sistemas complejos del mundo real a través de largos períodos.

Software.- Se conoce como software *al equipamiento lógico o soporte lógico* de una computadora digital; comprende el conjunto de los componentes lógicos necesarios que hacen posible la realización de tareas específicas, en contraposición a los componentes físicos, que son llamados hardware.

Solenoide.- Un solenoide es cualquier dispositivo físico capaz de crear una zona de campo magnético uniforme con un fin en específico, por ejemplo mover un dispositivo mecánico.

Solver.- Un solucionador (*solver* en inglés) es un término genérico que indica una parte de un software matemático, posiblemente en forma de un programa independiente o como una biblioteca de software, que "resuelve" un problema matemático.

Tarjeta de adquisición.- Tarjeta o modulo que se conecta a una computadora u otro dispositivo que permite la adquisición de datos desde diversos sensores.

Virtual.- Algo que tiene existencia aparente y no real. Es un término de frecuente utilización en el mundo de las Tecnologías de la Información y de las Comunicaciones para designar dispositivos o funciones simuladas.

BIBLIOGRAFÍA

1. **Janocha, Hartmut.** *Actuators: Basics and Applications.* s.l. : Springer, 2004.

2. **Bishop, Robert H.** *The Mechatronics Handbook.* s.l. : CRC Press, 2002.

3. **Silva, Clarence W. De.** *Mechatronic Systems.* s.l. : CRC Press, 2007.

4. iRobot. [En línea] 2010. www.irobot.com.

5. **Husqvarna.** [En línea] 2010. www.husqvarna.com.

6. *Definition of a manipulating industrial robot.* **International Organization for Standardization.** 1996, ISO 8373.

7. **Shimpo Drives, Inc.** [En línea] 2010. www.shimpodrives.com.

8. **Escuela Superior de Ingeniería de Gijon.** [En línea] 2010. http://www.unioviedo.es/DCIF/IMecanica/Motion/ayuda.pdf.

9. **Gómez González, Sergio.** *SolidWorks.* 1era Edición. México : Marcombo, 2008.

10. **National Instruments.** Qué es LabVIEW. [En línea] 2010. http://www.ni.com/labview/whatis/esa/.

11. **Molina Martínez, Jose Miguel.** *Programación gráfica para ingenieros.* 1era Edición. Barcelona : Marcombo, 2010.

12. **National Instruments.** Ayuda de NI. [En línea] Junio de 2009. http://zone.ni.com/reference/en-XX/help/371361F-01/lvhowto/creating_lv_projects/.

13. —. NI SoftMotion: Create Your Custom Motion Controller on Any Platform with LabVIEW. [En línea] Abril de 2010. http://zone.ni.com/devzone/cda/tut/p/id/3723.

14. —. Vision Builder for Automated Inspection de National Instruments. [En línea] 2010. http://www.ni.com/vision/esa/vbai.htm.

15. **Klinger, Thomas.** *Image Processing with LabVIEW and IMAQ Vision.* USA : Prentice Hall, 2003.

16. **National Instruments.** Getting Started with NI SoftMotion for SolidWorks. [En línea] Junio de 2009. http://www.ni.com/pdf/manuals/372876a.pdf.

A MODO DE EPÍLOGO

Y a título estrictamente personal.

Quisiera plantear algunas reflexiones que me parecen de suma importancia, no sólo para la ingeniería, sino para la sobrevivencia de nuestra especie y nuestro planeta.

En un futuro nada lejano, la facilidad con la que se creará una máquina será increíble, no dudo que en una década tengamos la capacidad de literalmente poder "imprimir" un robot en cualquiera de nuestras casas, por ejemplo una aspiradora automática como las que se mencionaron en la introducción.

Todo esto suena a un mundo perfecto y casi mágico sacado de una novela de Aldous Huxley, pero no debemos olvidar que, tanto en la novela como en la vida real, nuestras acciones, tanto individuales como colectivas, forjan nuestro destino, por lo que es importante detenerse a reflexionar en nuestro presente y en nuestro futuro y en lo que queremos hacer con nuestro planeta. Lo que quiero decir en palabras mundanas es que "no todo es miel sobre hojuelas" y que al construir una máquina o dispositivo debemos de detenernos a pensar en las consecuencias que puede tener nuestro invento en un mundo controlado por la avaricia y el poder.

Todos sabemos o por lo menos sospechamos, que nuestros medios y métodos de producción actuales están acabando rápidamente con la vida natural en nuestro planeta. Arrasamos bosques, ríos, lagos, destruimos hábitats completos, ecosistemas, extraemos petróleo y lo quemamos desesperadamente para que nuestros vehículos funcionen, fabricamos productos de cuidado personal altamente tóxicos y los vendemos mintiéndole al consumidor sobre sus ingredientes "naturales", construimos plantas nucleares para producir "energía limpia" y barata sin mencionar el alto precio que los residuos tóxicos le harán pagar a nuestro planeta y a sus habitantes, diseñamos productos de consumo con una vida útil tan corta como la garantía que les se da

para que sean desechados y los consumidores tengan que comprar nuevos, implantamos modas y/o estereotipos para que la industria textil pueda diseñar y fabricar ropa que será inservible en un par de meses, fabricamos las máquinas para que las empresas vendan productos alimenticios con altas cantidades de glucosa y las personas engorden y enfermen de diabetes para que después fabriquemos las máquinas que las empresas usarán para fabricar y empaquetar insulina, fabricamos televisiones que le gritan a las personas casi 24 horas al día que no son nadie si no tienen dinero suficiente para comprar uno de los hermosos autos que diseñamos y construimos para que ellas mismas ingresen a un sistema esclavista donde no se necesitan látigos, sólo televisiones LCD. Estamos convirtiendo nuestro hermoso mundo natural en un horrible mundo digital artificial, en donde somos esclavos de nuestros propios dispositivos.

Con todas estas reflexiones uno podría preguntarse ¿realmente vamos en el camino correcto? Y sí, yo me lo pregunto muchas veces. No quiero alarmar a nadie pero creo que éste es un momento crítico para nuestro planeta, estoy cada día más seguro de que si no cambiamos nuestro modo de pensar, de actuar y de diseñar, nuestro mundo sufrirá consecuencias inevitables y la vida como la conocemos se terminará. No dudo que la vida continúe, la vida siempre continúa, siempre evoluciona, pero los que seguramente no van a continuar, somos nosotros, los pequeños habitantes con grandes mentes que, en comparación con el tiempo de vida del planeta no llevamos ni el 0.01% de éste disfrutando de la vida, es decir, que, poniéndolo de otra forma, si la vida completa del planeta, desde que se formó hasta el día de hoy, la imagináramos en siete días, los humanos llevamos viviendo en la tierra apenas los últimos siete segundos de su existencia y lo único que estamos logrando es, nuestra propia extinción.

Así que si queremos seguir gozando de la vida y al mismo tiempo dando vida a nuestras pequeñas creaciones debemos de tomar conciencia de que la tecnología y el capitalismo unidos tienen un costo muy grande. El consumismo está acabando con la vida en este planeta.

La amplia ambición de las instituciones hace que diseñen, construyan, distribuyan y vendan productos desechables, desde ropa y muebles,

hasta televisiones, computadoras y automóviles, generando contaminación, diferenciación económica, presión psicológica, estrés, ansiedad, fanatismo, dependencia y depresión. La recompensa económica es grande si quieres servir al sistema en su amplia ambición de poder y esclavismo.

Pongamos un poco más de conciencia en nuestras acciones, analicemos más las consecuencias o el impacto de un dispositivo o el uso de ciertos materiales en nuestros diseños. Quizás las ganancias sean más bajas pero es mejor para todos nosotros si una computadora funciona por 50 años a que funcione por 5. Si las empresas se decidieran a diseñar computadoras con la facilidad de mejorarlas con sólo cambiar el procesador y la memoria del dispositivo dejaríamos de gastar tanto dinero en ellas y de contaminar valles enteros con chatarra plástica y tóxica, de seguir destruyendo playas, bosques y mares en busca de petróleo. El petróleo debería de dejar de usarse inmediatamente, las energías renovables son lo suficientemente avanzadas para entrar en acción rápidamente, los gobiernos e instituciones tienen la holgura financiera suficiente para costear proyectos de verdad sustentables, los ingenieros de éste y de los demás países son lo suficientemente capaces para desarrollar la tecnología necesaria para dejar de contaminar nuestro planeta.

Debemos de parar la tremenda ambición que tenemos antes de comenzar a diseñar cosas verdaderamente útiles, resistentes y económicas. Un ejemplo claro es la urgente creación de un nuevo concepto de desplazamiento vial, quizás una bicicleta eléctrica con energía solar como combustible y al mismo tiempo combinado con un poco de energía cedida por nuestro propio cuerpo, es decir, pedaleo, ya que también necesitamos ejercitarnos más para evitar la constante demanda de servicios médicos y sobre todo, detener la gran ambición de poder de otra de las empresas más capitalistas del mundo: la farmacéutica.

Sé que mis palabras hasta ahora no van mucho con las bases de la mecatrónica, ya que sin el capitalismo y la revolución industrial no existiría esta ingeniería, pero gracias a ellos también estamos

acabando con nuestra propia existencia, por lo que debemos de enfocar nuestras miradas a un camino nuevo y armónico.

No tengo duda de que podremos lograrlo y podamos transmitir nuestra nueva filosofía de vida a nuestros congéneres, pupilos y descendientes ya que en un futuro no muy lejano esta filosofía no sólo tendremos que tenerla los humanos, sino las máquinas también, ya que por ahora los robots apenas están empezando a “oír”, a “ver”, a “sentir”, a “probar”, a “tocar” y a “decidir”, pero la digitalización y el avance tecnológico un día permitirá a las máquinas crear sus propios diseños y quién sabe, seguramente hasta su propio arte.

La tarea de un diseñador debe de ser hermosa, artística, filosófica y armónica a la vez.

Un ingeniero debe de transformar recursos en bienestar ambiental y animal, porque los seres humanos también somos animales, aunque a veces, lamentablemente, se nos olvide.

Cuauhtémoc Vladimir Luna Jiménez