ESPE

UNIVERSIDAD DE LAS FUERZAS ARMADAS

INNOVACIÓN PARA LA EXCELENCIA

DEPARTAMENTO DE CIENCIAS DE LA ENERGÍA Y MECÁNICA

CARRERA DE INGENIERÍA MECATRÓNICA

TRABAJO DE TITULACIÓN, PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO MECATRÓNICO

TEMA: CONTROL DE UN BANCO DE PRUEBAS ESTÁTICAS Y DINÁMICAS PARA NEUMÁTICOS

AUTOR: SUÁREZ FREILE DANIEL ALBERTO

DIRECTOR: GÓMEZ REYES ALEJANDRO PAÚL

CODIRECTOR: OERTEL CHRISTIAN

SANGOLQUÍ

2016

CERTIFICADO DEL DIRECTOR DEL TRABAJO DE TITULACIÓN

DEPARTAMENTO DE CIENCIAS DE LA ENERGÍA Y MECÁNICA

CARRERA DE INGENIERÍA MECATRÓNICA

CERTIFICACIÓN

Certifico que el trabajo de titulación, ***"CONTROL DE UN BANCO DE PRUEBAS ESTÁTICAS Y DINÁMICAS PARA NEUMÁTICOS"*** realizado por el señor ***DANIEL ALBERTO SUÁREZ FREILE***, ha sido revisado en su totalidad y analizado por el software anti-plagio. Así, el mismo cumple con los requisitos teóricos, científicos, técnicos, metodológicos y legales establecidos por la Universidad de las Fuerzas Armadas – ESPE. Por lo tanto, me permito acreditarlo y autorizar al señor ***DANIEL ALBERTO SUÁREZ FREILE*** que lo sustente públicamente.

Sangolquí, 21 de diciembre de 2016

ALEJANDRO PAÚL GÓMEZ REYES

DIRECTOR

AUTORÍA DE RESPONSABILIDAD

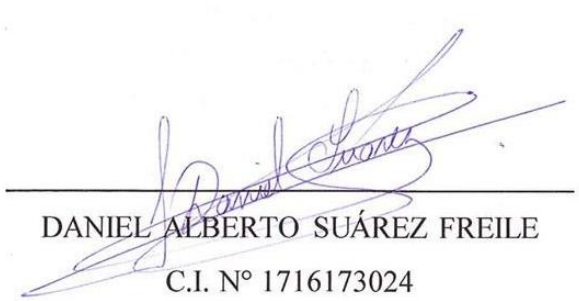
DEPARTAMENTO DE CIENCIAS DE LA ENERGÍA Y MECÁNICA
CARRERA DE INGENIERÍA MECATRÓNICA

AUTORÍA DE RESPONSABILIDAD

Yo, ***DANIEL ALBERTO SUÁREZ FREILE***, con cédula de identidad N° 1716173024, declaro que este trabajo de titulación, ***"CONTROL DE UN BANCO DE PRUEBAS ESTÁTICAS Y DINÁMICAS PARA NEUMÁTICOS"*** ha sido desarrollado considerando los métodos de investigación existentes, así como también ha respetado los derechos intelectuales de terceros, considerándolos en las citas bibliográficas.

Consecuentemente, declaro que este trabajo es de mi autoría. En virtud de ello, me declaro responsable del contenido, veracidad y alcance de la investigación mencionada.

Sangolquí, 21 de diciembre de 2016



DANIEL ALBERTO SUÁREZ FREILE
C.I. N° 1716173024

AUTORIZACIÓN (PUBLICACIÓN BIBLIOTECA VIRTUAL)

DEPARTAMENTO DE CIENCIAS DE LA ENERGÍA Y MECÁNICA

CARRERA DE INGENIERÍA MECATRÓNICA

AUTORIZACIÓN

Yo, ***DANIEL ALBERTO SUÁREZ FREILE***, autorizo a la Universidad de las Fuerzas Armadas – ESPE publicar en la Biblioteca Virtual de la institución el presente trabajo de titulación, ***"CONTROL DE UN BANCO DE PRUEBAS ESTÁTICAS Y DINÁMICAS PARA NEUMÁTICOS"*** cuyo contenido, ideas y criterios son de mi autoría y responsabilidad.

Sangolquí, 21 de diciembre de 2016

DANIEL ALBERTO SUÁREZ FREILE

C.I. N° 1716173024

DEDICATORIA

Dedico el presente trabajo, en primer lugar, a mis padres, Alberto y Janeth, quienes siempre han estado ahí para mí y me han brindado su apoyo incondicional a lo largo de mi vida. Ellos son el pilar fundamental de lo que he sido, soy y seré. A mis dos hermanas, Salomé y Sofía, quienes han sido mis mejores amigas desde que tengo memoria y han compartido conmigo muchas de las mejores experiencias de mi vida. A toda mi familia, pues sé que ellos jamás han dejado de creer en mí y de ayudarme. A mis amigos, quienes me han acompañado a lo largo de mi trayectoria profesional, hasta por los rincones más remotos del mundo, compartiendo alegría y éxito. Y a mí querida Catarina, por demostrarme que el amor es el tesoro más valioso que hay en el mundo, por ser esa mujer excepcional que nunca me ha dejado solo, y por acompañarme todos los días de mi vida, en lo más profundo de mí…

Daniel Suárez

2016-09-19

AGRADECIMIENTO

Agradezco, primeramente, a mis padres, a mi familia, a mis amigos y a mi enamorada, por todo su apoyo, esfuerzo y dedicación a lo largo del desarrollo del presente trabajo.

A la Universidad de las Fuerzas Armadas – ESPE y la Universidad de Ciencias Aplicadas de Brandemburgo, por todo el conocimiento impartido y por la oportunidad de realizar un proyecto de investigación en el exterior.

A la Unidad de Relaciones de Cooperación Interinstitucional – ESPE, a la Unidad de Bienestar Estudiantil - ESPE, a la Mg. Sara Durán y a la Lic. María de los Ángeles Batson, por todo su empeño a lo largo del proceso del presente trabajo.

Al Mg. Alejandro Gómez, al Prof. Dr.-Ing. Christian Oertel, y al Ing. Jan Hempel por toda la ayuda y enseñanzas necesarias para culminar con éxito el presente proyecto.

A Mina Sedra, por toda su colaboración a lo largo de toda la estancia de investigación.

Y a cada una de las personas que de alguna forma, siempre han estado ahí para mí y me han brindado una mano, mi más sincero agradecimiento y consideración de estima…

Daniel Suárez

2016-09-19

ÍNDICE

ÍNDICE DE TABLAS

ÍNDICE DE FIGURAS

RESUMEN

El presente proyecto detalla el control implementado en el banco de pruebas estáticas y dinámicas para neumáticos, en desarrollo por el Departamento de Ingeniería de la Universidad de Ciencias Aplicadas de Brandemburgo, en Brandemburgo del Havel - Alemania. Dicho control o automatización consta de una Interfaz Gráfica de Usuario o GUI (sigla en inglés de Graphical User Interface) y de un sistema de instrumentación mecatrónica, para cada una de las dos estaciones del banco de pruebas. Las mencionadas estaciones son independientes y a su vez forman parte de un proyecto de mayor escala, llamado Laboratorio de Propiedades de Neumáticos o TPL (Tire Property Lab). El objetivo principal del TPL es generar un modelo paramétrico de cualquier neumático disponible en el mercado mediante la medición y el procesamiento de las propiedades más significativas del mismo. El modelo paramétrico permite modelar computacionalmente un neumático, estudiarlo y correr pruebas sobre el mismo con el fin de optimizar y desarrollar nuevos neumáticos que puedan beneficiar a la industria automovilística. Ahora bien, para generar dicho modelo paramétrico, la primera estación, llamada Mesa 3D o 3DTisch extrae y modela la sección transversal de un corte de tal característica de cualquier neumático. Consiguientemente, dicho modelo puede ser sujeto a pruebas de Análisis de Elementos Finitos dado el estudio del material. La segunda estación, propiamente la Estación de Pruebas o TTR (Tire Test Rig) ejecuta las pruebas encargadas de medir las propiedades más importantes de los neumáticos. Así, los resultados obtenidos de ambas estaciones contribuyen al modelo paramétrico del neumático sujeto a prueba.

PALABRAS CLAVE:

- **CONTROL.**
- **INTERFAZ GRÁFICA DE USUARIO.**
- **MODELO PARAMÉTRICO.**
- **PRUEBAS ESTÁTICAS Y DINÁMICAS PARA NEUMÁTICOS.**

ABSTRACT

The following project explains in detail the implemented control of the Tire Test Rig (TTR) used for static and dynamic tests, currently in development under the supervision of the Department of Engineering at the Brandenburg University of Applied Sciences, in Brandenburg an der Havel – Germany. The mentioned control or automation is managed through a Graphical User Interface (GUI) and a mechatronic instrumentation system, for each of the two workstations composing the TTR. Both workstations are independent and together, form a greater project, called Tire Property Lab (TPL.) The main goal of the TPL is to generate a parametric model of any tire available on the market, by measuring and processing most of its important properties. The parametric model can be used to model, study and test the same or a different tire with similar characteristics in a specific computer software. The results could then lead to new models, studies and further development of not-yet-existing tires that could benefit the car industry and in which, tire manufactures are very interested. In order to create the already said parametric model of a tire, the first workstation called 3DTable or 3DTisch is in charge of modeling its tire cross-section. As a result, this model can be utilized for Finite Element Analysis (FEA) and future research. The second workstation is the TTR itself and it allows to test, measure and obtain the most significant properties of the tire. Then, the acquired results from both workstations contribute to the parametric model of the test subject.

KEYWORDS:

- **CONTROL.**
- **GRAPHICAL USER INTERFACE.**
- **PARAMETRIC MODEL.**
- **STATIC AND DYNAMIC TESTS FOR TIRES.**

CAPÍTULO 1
GENERALIDADES Y DESCRIPCIÓN DEL PROBLEMA

El presente capítulo se centra en brevemente definir los principales aspectos del proyecto, incluyendo la situación que permitió la realización del mismo, su importancia e impacto en la industria, el alcance en cuanto a la utilidad del proyecto y una concisa descripción de los principales componentes del mismo. Así, este capítulo pretende familiarizar al lector con el proyecto y dar a entender las características más significativas del trabajo realizado.

1.1. ANTECEDENTES

La Universidad de Ciencias Aplicadas de Brandemburgo fue fundada en abril de 1992, en la ciudad de Brandemburgo del Havel, en Alemania. Las instalaciones que hoy ocupa la universidad fueron en el pasado el Cuartel Militar del distrito de Brandemburgo, y en la actualidad cuenta con un sinnúmero de edificios e instalaciones renovadas. La universidad ha estado desde entonces comprometida con la educación y el desarrollo de la sociedad alemana, concentrándose efusivamente en el campo tecnológico e ingenieril.

El Departamento de Ingeniería de la Universidad de Ciencias Aplicadas de Brandemburgo está vinculado con la investigación y el desarrollo de nuevas tecnologías que supongan un desafío para sus estudiantes tanto nacionales como internacionales. En la actualidad, cuenta con proyectos de investigación en los cuales intervienen los principales campos de la ingeniería, como la mecatrónica, la ingeniería de control y la informática.

Hoy en día, la Universidad de las Fuerzas Armadas “ESPE” y la Universidad de Ciencias Aplicadas de Brandemburgo cuentan con una estrecha relación que permite el desarrollo de proyectos de investigación en la institución alemana, para las ramas de electrónica y mecatrónica. Fue el caso del presente proyecto, que se centró en el control de un banco de pruebas estáticas y dinámicas para neumáticos con el fin de determinar las principales características de los mismos en cuanto a su diseño y manufactura.

1.2. JUSTIFICACIÓN E IMPORTANCIA

La tendencia actual de la industria es la automatización de sus procesos. Esto ha permitido facilitar el manejo y la instrumentación en líneas de producción y manufactura, así como mejorar el control de calidad y promover la investigación de nuevas tecnologías. Muchas empresas se dedican a estudiar y optimizar sus productos, y lo hacen a través de bancos de prueba, en ambientes controlados, con el fin de determinar el comportamiento y las principales características de los mismos.

Teniendo en cuenta que los neumáticos son productos de utilización diaria en un sinnúmero de sistemas mecánicos y electromecánicos, como son los autos, los aviones, y los sistemas de transporte terrestre en general, es menester de equipos de desarrollo y testeo realizar pruebas de los mismos. Dichas pruebas ayudan a determinar las propiedades fundamentales de los neumáticos como: el labrado necesario para los distintos tipos de terreno, las velocidades óptimas de funcionamiento, la distribución de presión, las dimensiones, la carga máxima recomendada, los coeficientes de rozamiento y desgaste, la rigidez vertical, etc.

Así, la Universidad de Ciencias Aplicadas de Brandemburgo, en Brandemburgo – Alemania, ha desarrollado un banco de pruebas estáticas y dinámicas para neumáticos, centrándose en la gama de neumáticos livianos, con el fin de extraer algunas de las características antes mencionadas de los mismos y generar un modelo paramétrico que permita estudios posteriores. El presente proyecto detalla la instrumentación de la máquina así como su control manual y automático mediante una Interfaz Gráfica de Usuario.

El desarrollo del proyecto es entonces fundamental, pues contribuye a la investigación del desarrollo de neumáticos, mediante la automatización de una máquina encargada de testearlos, en un mundo que básicamente depende de ellos en cuanto a transporte terrestre, y más aún en el entorno industrial europeo y el creciente latinoamericano.

1.3. OBJETIVOS

1.3.1. Objetivo general

- Controlar un banco de pruebas estáticas y dinámicas para neumáticos mediante algoritmos de control, instrumentación y despliegue de información en una Interfaz Gráfica de Usuario desarrollada en Visual C++.

1.3.2. Objetivos específicos

- Contribuir al desarrollo del banco de pruebas, mediante la implementación de sistemas eléctricos y electrónicos.
- Instrumentar los procesos de prueba para las variables de estudio con base en la plataforma de programación gráfica de LabVIEW y sus módulos de medición.
- Diseñar un sistema de control en relación a las variables de estudio, para las diferentes estaciones del banco de pruebas.
- Programar una Interfaz Gráfica de Usuario encargada de monitorear y controlar los procesos y pruebas de cada estación de trabajo del banco de pruebas.
- Elaborar un manual de usuario que detalle el funcionamiento y los procesos de cada una de las estaciones de trabajo del banco de pruebas.

1.4. ALCANCE DEL PROYECTO

El presente proyecto tiene como meta desarrollar un sistema de monitoreo, supervisión y control para un banco de pruebas estáticas y dinámicas para neumáticos, y a través del mismo, automatizar dicho banco con el fin de facilitar las diferentes pruebas que buscan realizarse y obtener resultados, a partir de las variables de estudio.

Así, el banco de pruebas será capaz de medir y determinar las principales características y propiedades de cualquier neumático disponible en el mercado. Cabe señalar que ambas estaciones de trabajo forman parte de un proyecto de mayor alcance llamado Laboratorio de Propiedades de Neumáticos, el cual busca parametrizar el neumático sujeto a prueba y modelarlo en software, con el fin de realizar estudios y análisis posteriores, que buscan dar lugar a una biblioteca virtual de neumáticos.

Entre las pruebas que busca realizar el banco destacan:

- Prueba de rigidez estática.
- Prueba de rigidez dinámica.
- Prueba de uniformidad de corte.

1.5. DESCRIPCIÓN GENERAL DEL PROYECTO

El desarrollo del sistema de monitoreo, supervisión y control para el banco de pruebas comprende en un inicio analizar el funcionamiento mecánico de cada una de las estaciones que lo comprenden. Después, se adicionará la instrumentación, encargada de censar y retroalimentar los sistemas. Posteriormente, a través del software de control se desarrollaran los algoritmos pertinentes así como la Interfaz Gráfica de Usuario, dependiendo de las pruebas, para que manejen el comportamiento de las variables de estudio.

1.5.1. Sistema mecánico

El sistema mecánico del banco de pruebas está compuesto por dos máquinas principales, una por estación. La primera es una máquina de Control Numérico Computarizado y la segunda la estación de pruebas en sí, conformada por algunos elementos mecánicos de sujeción y trabajo, como ejes, rodamientos y válvulas electro-neumáticas. A continuación se presenta un modelado de esta última:

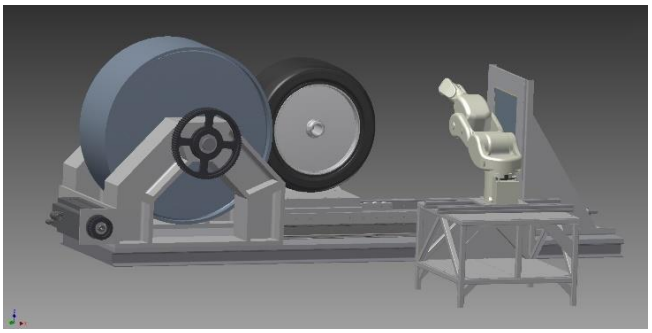


Figura 1. Modelado computacional de la estación de pruebas para neumáticos.

1.5.2. Sistema eléctrico/electrónico

Los elementos eléctricos que conforman el banco de pruebas son principalmente de control industrial como pulsadores, switches y luces indicadoras; y actuadores como motores eléctricos.

En cuanto a los elementos electrónicos, el banco de pruebas consta con sensores y sistemas de adquisición de datos. Dichos componentes constituyen el hardware del sistema de instrumentación.

1.5.3. Sistema de control

Para el presente proyecto, tres son los elementos que definen su desarrollo. Así:

Variables de estudio:

- La fuerza aplicada al neumático durante la prueba de rigidez estática y dinámica, en la estación de pruebas.
- La velocidad del motor encargado indirectamente de aplicar la fuerza al neumático mediante un mecanismo de tornillo sin fin, durante las pruebas de rigidez en la estación de pruebas.
- El paso de los motores de pasos de la máquina de CNC, en la Mesa 3D.
- La distancia entre la sección transversal de un corte de un neumático y el Punto Central de la Herramienta o TCP (sigla en inglés de Tool Center Point) de la máquina CNC.

Controlador:

Puesto que la primera estación de trabajo es una máquina de CNC, comprende un controlador integrado, con el cual la GUI se comunica directamente, proporcionando los comandos necesarios.

La estación de pruebas cuenta con dos controladores para la prueba de rigidez vertical tanto estática como dinámica. Un controlador clásico Proporcional-Integral-Derivativo o PID y un controlador por intervalos.

Interfaz Gráfica de Usuario:

Ambas estaciones cuentan con una GUI independiente y fueron programadas en Visual C++. La primera se comunica directamente con el controlador de la máquina CNC y con el sensor de distancia, a través de un sistema de adquisición de datos vinculado a LabVIEW.

La segunda GUI se encuentra interconectada con el software de instrumentación mediante Bibliotecas de Enlace Dinámico o DLL's (sigla en inglés de Dynamic-Link Libraries), el cual a su vez se conecta con el sistema físico de instrumentación, compuesto por los sensores y actuadores.

1.6. INTRODUCCIÓN

El presente trabajo detalla entonces el proceso antes mencionado. En un inicio, se presenta una breve síntesis de los fundamentos teóricos utilizados a lo largo del desarrollo del sistema de instrumentación, de control y de monitoreo para cada una de las estaciones de trabajo.

Posteriormente, se describe concisamente la implementación de dichos sistemas en la primera estación de trabajo, denominada Mesa3D o 3DTisch, así como sus principales elementos constitutivos tanto mecánicos como electrónicos, el algoritmo de control, la Interfaz Gráfica de Usuario y los resultados obtenidos.

La segunda estación de trabajo o propiamente estación de pruebas es una máquina de mayor complejidad, y para facilidad de comprensión, el desarrollo del control de la misma ha sido dividido en cuatro capítulos.

El primer capítulo de los cuatro, analiza a la estación en términos de capacidad al momento de realizar pruebas mecánicas, y propone un diseño preliminar de la Interfaz Gráfica de Usuario que controlará el sistema.

El siguiente capítulo describe la fase preliminar del sistema de instrumentación de la TTR diseñado para comunicar los sensores y actuadores con la unidad de control y con el código fuente.

El tercer capítulo referente a la estación de pruebas explica la implementación de las clases finales que definen el programa de control, así como la elaboración de un manual de usuario en cuanto a normas de seguridad y funcionalidad de la TTR.

El último capítulo relacionado con la TTR detalla la implementación final de los Instrumentos Virtuales necesarios para establecer la comunicación entre los sensores y actuadores de la estación de pruebas y la Interfaz Gráfica de Usuario.

Las conclusiones y recomendaciones, fundamentales para determinar la validez del proyecto, se presentan en la última sección del trabajo, así como los anejos.

CAPÍTULO 2
FUNDAMENTO TEÓRICO Y ESTADO DEL ARTE

El presente capítulo sustenta teóricamente al proyecto y define los principales conceptos del mismo. Entre aquellos destacan: las Interfaces Gráficas de Usuario, encargadas de ser el vínculo entre el usuario y las estaciones de trabajo; los controladores, fundamentales en cuanto al monitoreo, supervisión y operación de las mismas; y las plataformas computacionales utilizadas para generar los algoritmos de control.

2.1. INTERFAZ GRÁFICA DE USUARIO

2.1.1. Definición de una Interfaz Gráfica de Usuario

Una Interfaz Gráfica de Usuario o GUI (sigla en inglés de Graphical User Interface) es un tipo de interfaz que permite al usuario interactuar con dispositivos electrónicos a través de íconos gráficos e indicadores visuales, contrario a las interfaces basadas en texto y/o etiquetas de comando. (Shelly, 2012)

2.1.2. Elementos de una GUI en control

Las GUI están compuestas por elementos comúnmente conocidos como Widgets. Los Widgets son elementos de interacción como botones o barras de navegación. Estos últimos son componentes del software que el usuario manipula directamente con el fin de leer o editar cualquier tipo de información acerca de una aplicación o de un proceso industrial. (Microsoft, 2016)

Las GUI engloban un sinnúmero de Widgets. No obstante, destacan los siguientes:

Botones

Un botón es un Widget que puede ser presionado con el fin de realizar una acción o ejecutar un evento. Además, son equivalentes a los botones normalmente utilizados en instrumentos mecánicos y electrónicos.

Existen diferentes tipos de botones, por ejemplo:

- **Pulsador o push button:** Es un componente gráfico que implementa un pulsador.

Figura 2. Pulsador.

- **Botón de activación o toggle button:** Es un botón de dos estados que al ser activado, selecciona uno de estos.

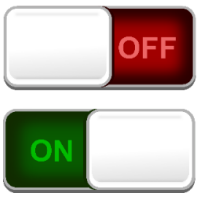


Figura 3. Botón de activación.

- **Deslizador o slider:** Es un control gráfico que permite ajustar un valor específico.

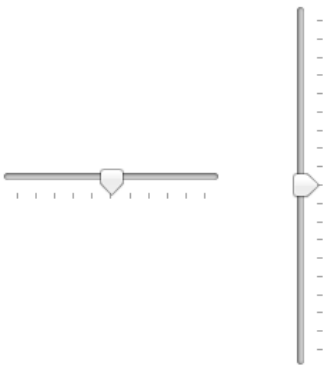


Figura 4. Deslizadores.

Diálogos

Un cuadro de diálogo es un elemento gráfico de control que comunica cierta información al usuario y le solicita una respuesta o acción. (Microsoft, 2016)

Figura 5. Cuadro de diálogo de Microsoft Windows.

Existen dos tipos de cuadros de diálogo:

- **Cuadros de diálogo modales:** Un cuadro de diálogo modal genera un caso donde la ventana principal o activa no puede ser usada. Dicho de otro modo, éstos impiden que el usuario interactúe con la GUI al menos que este atienda a la solicitud del cuadro de diálogo modal.
- **Cuadros de diálogo no modales:** Como su nombre indica, son exactamente el opuesto de los cuadros de diálogo modales, Así, al aparecer, estos no bloquean la interacción del usuario con la GUI.

2.1.3. Medidas de seguridad en GUI

La ingeniería en seguridad y fiabilidad de sistemas se ha convertido en un peldaño fundamental dentro del control industrial. La meta principal de diseñar un proceso seguro es reducir drásticamente el riesgo tanto para la maquinaria como para los usuarios de la misma. Comúnmente, las normas de seguridad se relacionan directamente con otras disciplinas afines como la disponibilidad y el mantenimiento. Consecuentemente, la seguridad industrial garantiza también la calidad de un producto o en sí, de un proceso dentro de la industria.

Las GUI contienen una variedad de elementos que promueven medidas de seguridad, tales como alertas visuales y advertencias. A su vez, éstas pueden manipular señales de entrada y de salida con el fin de mostrar una posible amenaza y automáticamente generar una respuesta que reduzca o elimine el riesgo, por ejemplo una parada de emergencia automática.

La manipulación de señales se relaciona significativamente con las medidas de seguridad ya que las GUI dependen de dichas señales para funcionar correctamente y garantizar una precisa adquisición de datos y posterior procesamiento.

A continuación se presenta un ejemplo bastante demostrativo de lo que sería un excelente sistema de seguridad. El tablero de instrumentos de un automóvil contiene un sinnúmero de elementos como actuadores, símbolos visuales y alertas de sonido. Éstos son muy útiles para el conductor puesto que permiten al mismo conocer la condición y el comportamiento de muchos de los componentes del automóvil.

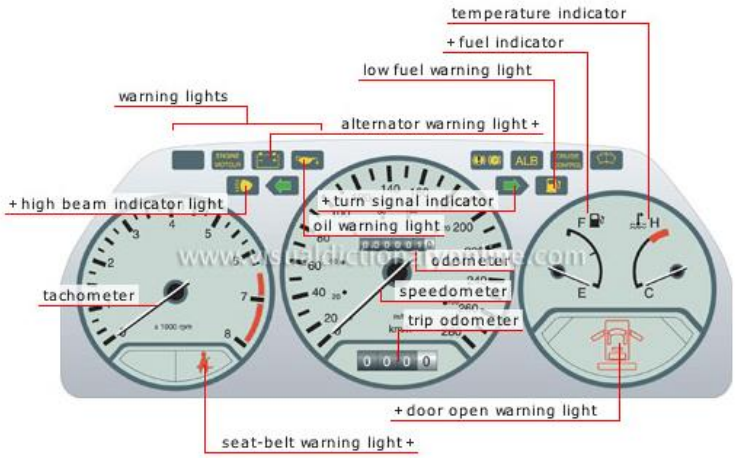


Figura 6. Ejemplo ilustrativo de un tablero de instrumentos de un automóvil.

2.1.4. Estado del arte

Con botones de activación y pulsadores, deslizadores y perillas, los paneles de control industrial eran parques mecánicos. Puesto que dichos tableros eran completamente mecánicos, tenían el potencial de ser una fuente de problemas, desde articulaciones soldadas hasta mecanismos móviles. Además, al ser construidos para una tarea específica de control, los tableros eran difíciles, si no imposibles, de

reconfigurar para cualquier otra aplicación. Conjuntamente, su utilización era compleja y su utilización era complicada.

En la última década, las GUI han mejorado totalmente y en la actualidad, son ampliamente usadas en procesos industriales. El objetivo principal de una GUI es permitir al usuario de la misma fácilmente alcanzar su deseada interacción en el menor tiempo posible y utilizando la mínima cantidad de recursos. Existen muchos tipos de GUI y muchas maneras de diseñarlas, por ejemplo GUI implementadas en pantallas táctiles. Éstas se han vuelto muy comunes y ahora son muy utilizadas en la industria debido a que son amigables al usuario, convierten a los procesos en aplicaciones más sencillas y reducen la complejidad de los mismos.

Así que, en lugar de paneles de control mecánico, las instalaciones en la actualidad se hallan principalmente compuestas de elementos de estado sólido, controladas por una GUI, generalmente desplegada en una pantalla. Al interactuar con la GUI, esta cambia puesto que está vinculada visualmente al proceso y éste consta de diferentes funciones. Esto significa que, además de ser reconfigurable, el sistema puede guiar al usuario a través de las operaciones.

Los paneles táctiles poseen muchas aplicaciones. Frecuentemente, son utilizados en máquinas industriales como sistemas de Control Numérico Computarizado o CNC. Gracias a su simplicidad en cuanto al uso, las GUI desplegadas en pantallas táctiles permiten al usuario seleccionar funciones de cualquier operación, como por ejemplo la herramienta a utilizarse y la velocidad de rotación y avance. Adicionalmente, en algunos casos, el usuario puede observar una simulación del producto antes y durante el maquinado.

La interrogante que se presentó durante la realización del presente proyecto fue si es que aplicar directamente una GUI desplegada en una pantalla táctil era viable y significativo. En realidad, sí, ya que muchos de los parámetros podían ser controlados a través de la misma. No obstante, el sistema de control para ambas estaciones de trabajo necesitaba un sistema de adquisición y procesamiento de datos fuerte, y un panel táctil podía resultar caro, por lo que se optó por uno tradicional, diseñado en Visual C++ y capaz de comunicarse fácilmente con los actuadores y sensores.

Figura 7. Ejemplo de una GUI desplegada en una pantalla táctil.

2.2. CONTROLADORES

2.2.1. Definición de un controlador

Un controlador es un dispositivo que monitorea y físicamente altera las condiciones de operación de un sistema dinámico dado. Entre las aplicaciones más comunes para los controladores están las de mantener parámetros como por ejemplo de presión, temperatura, flujo y velocidad.

Los controladores son de vital importancia en procesos industriales porque permiten a los mismos ser más precisos, al medir sus parámetros de entrada y salida, y relacionarlos al valor deseado con el fin de generar una respuesta adecuada.

En la teoría de control electrónica, los controladores pueden diseñarse tanto en hardware como en software. Para los de diseño en hardware, los elementos más usados son los circuitos integrados. En el caso de los diseñados en software, se generan códigos fuente basados en algoritmos que más tarde se implementan en micro-controladores o en sistemas computacionales.

2.2.2. Tipos de control

Control retroalimentado o feedback

Un control retroalimentado se caracteriza porque su entrada es lo mismo que se intenta controlar. Como su nombre mismo lo indica, la variable de control es retroalimentada al controlador. Un ejemplo ilustrativo es el termostato de una casa. El

controlador depende de la medición de la variable de control, en este caso la temperatura de la casa, y del ajuste de la salida. Sin embargo, el control retroalimentado usualmente resulta en intervalos de tiempo donde la variable de control no está sobre el "set-point" deseado.

Control feed-forward

La idea de un control feed-forward es actuar mucho más rápido que un control retroalimentado, y se basa en la anticipación y medición de las perturbaciones antes de que tengan tiempo de afectar directamente al sistema. La dificultad del presente controlador radica en que el efecto de las perturbaciones debe ser precisamente predichas y no deben existir perturbaciones desconocidas.

Es posible implementar ambos tipos de control y aprovechar ambos beneficios. Por un lado, el control retroalimentado permite controlar perturbaciones desconocidas sin necesariamente saber exactamente cómo el sistema responde a las mismas. Por otro lado, el control feed-forward autoriza al controlador a responder ante perturbaciones antes de que puedan afectar al sistema.

2.2.3. Clases de controladores

Controladores clásicos

- **Controlador sí/no u on/off:** El controlador sí/no o también conocido como controlador de histéresis, es un controlador retroalimentado que cambia abruptamente entre dos estados.
- **Controlador Proporcional-Integral-Derivativo o PID:** El controlador PID es un controlador retroalimentado, comúnmente usado en sistemas de control industrial, cuya característica principales es calcular continuamente la señal de error, dada por la diferencia entre el "set-point" deseado y la medición correspondiente a la variable de control, y aplicar un factor de corrección con base en los términos proporcional, integral y derivativo de la misma. Normalmente, si el error es significativo, su acción de control también lo será y

viceversa. El controlador también admite las siguientes variaciones: un controlador P, un controlador PI o un controlador PD.

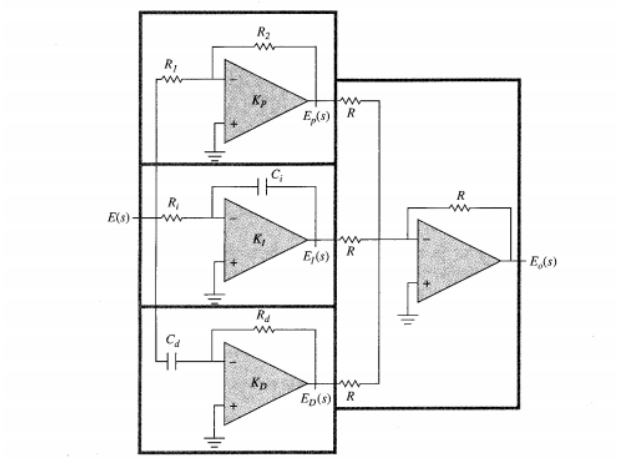


Figura 8. Ejemplo de un controlador PID basado en hardware.

Controladores inteligentes

- Un controlador inteligente es un sistema que pretende controlar a otro con base en una técnica de inteligencia artificial y no en una técnica matemáticamente determinista.
- Es pertinente aplicar un control inteligente cuando la planta a controlarse se encuentra en ambientes variables e inciertos, puesto que los controladores inteligentes poseen capacidad de adaptación y aprendizaje.
- Entre los más comunes, destacan:
  - Control difuso.
  - Control neuronal.
  - Control neuro-difuso.
  - Anfis (usa algoritmos genéticos).

2.2.4. Implementación de controladores

La teoría de control es una rama de la ingeniería que se concentra en el comportamiento de los sistemas dinámicos, y como éstos responde a una cierta retroalimentación. Por lo tanto, el objetivo principal de la teoría de control es controlar

una planta para que su salida siga una señal deseada de control o referencia, la cual puede ser variable o constante. Para cumplir dicho objetivo, es necesario implementar un controlador que cumpla la función de monitorear la salida y compararla con la referencia.

Así, las técnicas de análisis y diseño de sistemas de control se categorizan como:

- **Representación en dominio del tiempo:** En este tipo de representación, los valores de las variables de estado están representadas como funciones dependientes del tiempo. Consecuentemente, el sistema analizado está compuesto de una o más ecuaciones diferenciales. El dominio del tiempo es ampliamente utilizado para analizar sistemas no lineales.
- **Representación en dominio de la frecuencia:** En este tipo de ilustración, los valores de las variables de estado están representadas como funciones de la frecuencia. Para obtener dichas funciones, tanto las variables de estado como la función de transferencia del sistema son convertidas de funciones dependientes del tiempo a funciones de la frecuencia a través de transformaciones matemáticas. El domino de la frecuencia simplifica el cálculo matemático. No obstante, están limitados sólo a sistemas lineales. (Wikibooks, 2012)

A pesar de que existe un sinnúmero de transformaciones matemáticas que son utilizadas para analizar funciones en el dominio del tiempo, solamente dos están relacionadas a la teoría de control. La transformada de Laplace, la cual define los sistemas analógicos, y la transformada Z, la cual describe a los sistemas discretos.

Existe una importante distinción entre los sistemas analógicos y digitales. Así:

- **Sistemas analógicos:**
  - **Tiempo continuo:** Una señal es de tiempo continuo si se encuentra definida para todo tiempo *t*. Por tanto, un sistema en tiempo continuo toma una señal de entrada en tiempo continuo e imprime una señal de salida en tiempo continuo. La siguiente figura ilustra la presente definición:

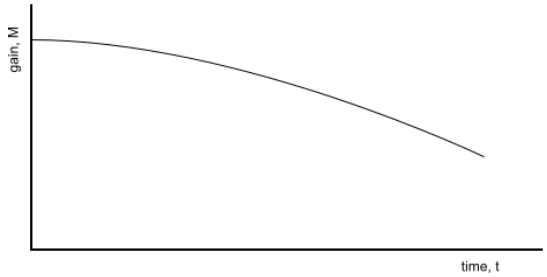


Figura 9. Ejemplo de una forma de onda analógica.

- Por definición, se considera a una señal analógica si ésta se halla definida para todo tiempo *t* y puede tomar cualquier valor real dentro de su rango. En otras palabras, un sistema analógico es un sistema que es continuo tanto en tiempo como en magnitud.

- **Sistemas discretos:**
  - **Tiempo discreto:** Una señal es de tiempo discreto si y sólo si está definida para puntos particulares del tiempo. En consecuencia, un sistema es de tiempo discreto si toma como entrada una señal en tiempo discreto y produce una señal de salida en tiempo discreto. La figura consiguiente describe la diferencia entre una forma de onda analógica y su muestra equivalente en tiempo discreto.

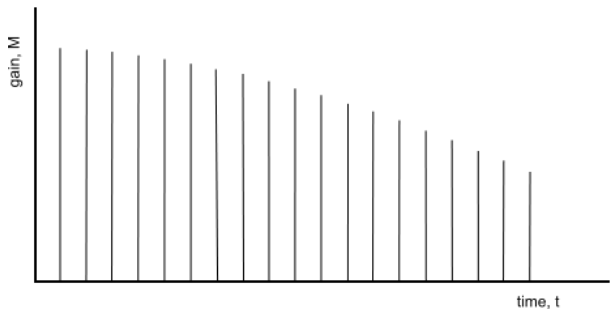


Figura 10. Ejemplo de una forma de onda digital.

- Por definición, se considera a una señal digital si ésta se halla representada y cuantificada en tiempo discreto. Una señal está cuantificada cuando sólo puede tomar ciertos valores. La siguiente figura denota una señal discreta en magnitud, pero continua en tiempo, donde la forma de onda toma solamente ciertos valores, dándole una apariencia de escalera. (Wikibooks, 2012)

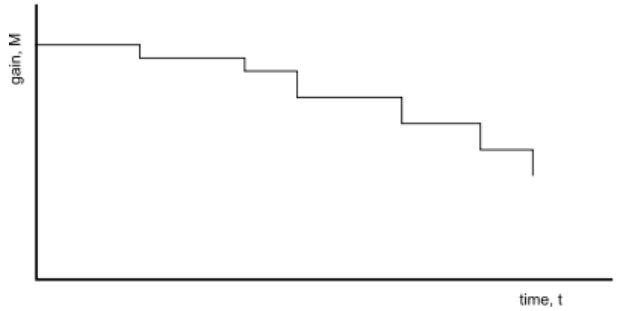


Figura 11. Ejemplo de una señal discreta en magnitud, pero continua en tiempo.

2.2.5. Estado del arte

En la actualidad, es imposible pensar que la aplicación de la teoría de control y la implementación de controladores no estén relacionadas con la industria. Éstas han permitido mejorar significativamente los procesos industrias, optimizar los actuadores de numerosas máquinas y generar productos de una calidad más elevada.

Como se menciona anteriormente, el objetivo principal de un controlador es monitorear alterar físicamente las condiciones de operación de un sistema dinámico dado. Para ello, se han desarrollado una serie de estrategias y diseños, tomando en cuenta la lógica e innumerables artificios matemáticos que han dado lugar a algoritmos muy estrictos.

Hoy en día se habla de controladores inteligentes, así como de control robusto. Los primeros se basan técnicas no determinísticas y de inteligencia artificial, y han aparecido debido al avance en la complejidad de sistemas industriales. Tal es el caso de los controladores difusos y las redes neuronales, las mismas que permiten a la planta anticipar y aprender una mejora manera de funcionar de acuerdo a los parámetros propios de ésta, muchas veces sin necesidad de modelarla matemáticamente.

Dentro de la teoría de control, el control robusto se enfoca en el diseño de controladores que manejen explícitamente la incertidumbre. Así, los métodos robustos buscan alcanzar un rendimiento sólido y estable en presencia tanto de perturbaciones como de posibles errores de modelamiento. Normalmente, un sistema de control robusto incorpora topologías que incluyen lazos de retroalimentación múltiples y de feed-forward, y sus algoritmos de control trabajan con funciones de transferencia de orden superior.

Para el presente proyecto, la implementación de las técnicas de control mencionadas probaba ser innecesaria pues los sistemas mecatrónicos que conformaban ambas estaciones de trabajo no tenían un grado de complejidad elevado.

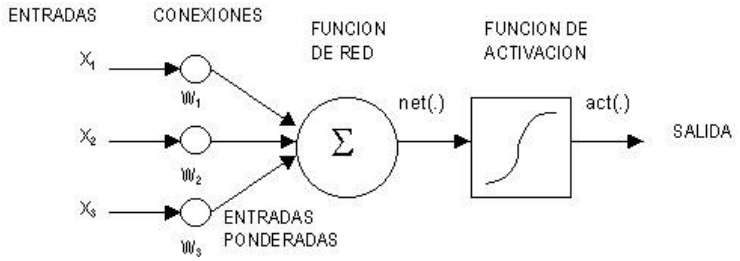


Figura 12. Esquema general de una red neuronal.

2.3. PLATAFORMAS COMPUTACIONALES DE INSTRUMENTACIÓN Y CONTROL

2.3.1. Microsoft Visual Studio Ultimate 2012

Microsoft Visual Studio es un Entorno de Desarrollo Integrado o IDE (sigla en inglés de Integrated Development Environment) edificado por Microsoft. (Microsoft, 2016) Primordialmente, es usado en el desarrollo de programas de computación para Microsoft Windows, así como páginas web, aplicaciones web y servicios web. Visual Studio utiliza las principales plataformas de desarrollo de software de Microsoft como Windows API, Windows Forms, Windows Presentation Foundation, Windows Store y Microsoft Silverlight. En adición, Visual Studio trabaja tanto con código fuente como con código gestionado.

Visual Studio admite algunos lenguajes de programación y permite al programador y al depurador trabajar con éstos dado que el servicio específico de tal lenguaje exista. Entre los lenguajes de programación incluidos se encuentran C, C++, C++/CLI, VB.NET, C# y F#.

Figura 13. Logotipo de Microsoft Visual Studio.

Marco .NET o .NET framework

El marco .NET es una estructura de software, desarrollada por Microsoft, que incluye una enorme librería denominada FCL (sigla en inglés de Framework Class Library). Esta librería se encarga de proveer un sinnúmero de funciones tales como: interfaces de usuario, acceso de información, conectividad entre bases de datos, algoritmos numéricos, comunicación entre redes, etc. Además, .NET proporciona interoperabilidad entre algunos lenguajes de programación, lo que significa que cada lenguaje de programación puede utilizar código escrito en otros lenguajes. (Bogotobogo, 2016)

Los programas escritos utilizando el marco .NET se ejecutan en un ambiente de software conocido como CLR (Common Language Runtime). El CLR es una máquina virtual de aplicación que brinda ciertos servicios como manejo de memoria, manejo de excepciones y seguridad. Tanto la FCL como el CLR conforman el marco .NET.

Figura 14. Logo del marco .NET.

Lenguaje de programación C++

C++ es un lenguaje de programación de propósito general basado en el lenguaje de programación C. Posee aplicaciones de orientación a objetos y programación genérica así como provee servicios de manipulación de memoria de bajo nivel. (ACSES, 2015)

Fue diseñado con una inclinación hacia la programación de sistemas tradicionales y embebidos, junto con un buen rendimiento, eficiencia y flexibilidad en cuanto a su estructura. Sin embargo, no ha dejado de ser muy útil en otros contextos ya que tiene elementos clave relacionados con la infraestructura de software y desarrollo de aplicaciones.

Por último, C++ es un lenguaje compilado que puede ser implementado en algunas plataformas, y es provisto por varias organizaciones como Microsoft.

Figura 15. Logo de Microsoft Visual C++.

Estructura general de los programas basados en .NET

El marco .NET dio un giro inesperado al mundo de la programación. Antes de su lanzamiento, cualquier aplicación diseñada por un programador funcionaba únicamente dentro de un sistema operativo específico. Hoy en día, gracias a su aparición, los programadores son capaces de ejecutar sus aplicaciones en cualquier sistema operativo.

Para cumplir la tarea antes mencionada, .NET trasforma cualquier programa codificado en lenguaje de alto nivel a un programa en lenguaje intermedio llamado CIL (sigla en inglés de Common Intermediate Language) y más tarde el CLR convierte a dicho lenguaje en uno de bajo nivel, capaz de ser interpretado por el sistema operativo.

Los principales lenguajes de programación que utilizan .NET son C++, C#, Visual Basic y J#, siendo C++ el lenguaje principal del presente proyecto.

A continuación se detallan brevemente los elementos más importantes dentro de un programa basado en .NET:

- **Archivos de cabecera y fuente o header y source files:** Los archivos de cabecera y archivos fuente son los elementos principales de un programa en C++. Se denomina archivo de cabecera al archivo en forma de código fuente que el compilador incluye al procesar algún otro archivo fuente. Así, un archivo fuente es aquel ejecuta una secuencia de comandos y basa su codificación en los archivos de cabecera incluidos en el mismo.
- **Constructores y destructores:** Los constructores y destructores son funciones de miembro especial. Por un lado, un constructor es un tipo de función que inicializa una estancia de su propia clase. Por otro lado, un destructor es la función inversa de un constructor. Son llamadas cuando algún objeto es eliminado.
- **Variables:** Una variable es un espacio reservado de memoria que requiere una dirección y un tamaño específico. Además, una variable debe definir un tipo de dato como: valores enteros, valores flotantes, valores tipo caracter, valores booleanos, etc.
- **Punteros:** Un puntero es una variable que almacena la dirección de memoria de otra variable.
- **Código gestionado o managed code:** El código gestionado es un tipo de código diseñado para utilizar los servicios y características de un entorno de programación, tal como el CLR del marco .NET. El uso de dicho código evita que el programador cometa errores de programación típicos relacionados con aplicaciones inestables y vacíos de seguridad. Además, el código en sí se encarga de completar tareas tales como un manejo óptimo de la memoria y la destrucción de objetos inutilizados.
- **Recolector de basura o garbage colector:** El recolector de basura de .NET maneja la asignación y la liberación de memoria de una aplicación.

2.3.2. National Instruments LabVIEW 2013

LabVIEW (sigla en inglés de Laboratory Virtual Instrument Engineering Workbench) es una plataforma de diseño y entorno de desarrollo con base en un lenguaje visual de programación elaborado por National Instruments. El lenguaje gráfico que el programa utiliza se denomina G. (National Instruments, 2016)

LabVIEW ofrece un sinnúmero de aplicaciones de control, procesamiento y adquisición de datos. Entre ellas destacan:

- Integración de hardware.
- Matemática.
- Procesamiento de señales y control.
- Lectura, escritura y comunicación de datos.
- Integración de código y software.

Figura 16. Logotipo de LabVIEW.

Descripción concisa de LabVIEW

Los programas en LabVIEW se llaman Instrumentos Virtuales o VI (Virtual Instruments) ya que su apariencia y operación se asemeja a instrumentos físicos de medición y control. En adición, LabVIEW contiene un conjunto significativo de herramientas para adquirir, analizar, desplegar y guardar información. (National Instruments, 2016)

La plataforma de LabVIEW se basa en dos ventanas. La ventana de Panel Frontal o Front Panel window y la ventana de Diagramas de Bloque o Block Diagram window. Básicamente, la ventana de Panel Frontal es la Interfaz Gráfica de Usuario del VI y contiene típicamente botones, deslizadores, perillas, gráficas, luces indicadores, etc; mientras que la ventana de Diagramas de Bloque contiene el código fuente gráfico que controla cada uno de los elementos de la ventana de Panel Frontal. La figura a continuación ilustra lo mencionado.

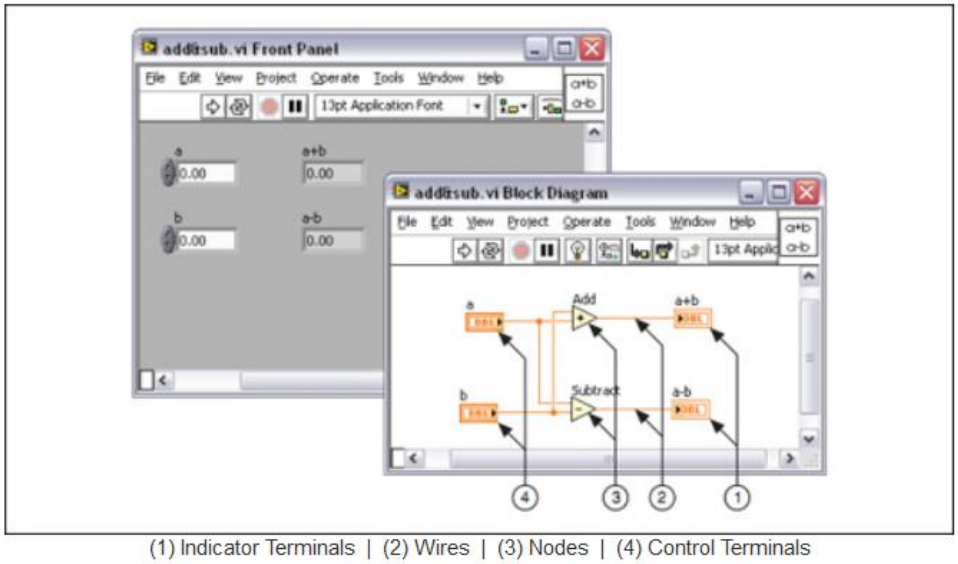


Figura 17. Ejemplo de un VI en LabVIEW.

Brevemente, LabVIEW permite a los diseñadores construir aplicaciones versátiles a través de Interfaces Gráficas de Usuario y codificación gráfica.

2.3.3. Gnuplot 5.0

Gnuplot es una aplicación portátil basada en comandos escritos diseñada para Linux, MS Windows, OSX y muchas otras plataformas computacionales. Originalmente, fue diseñada para que investigadores puedan visualizar funciones matemáticas y datos de una forma interactiva, mas hoy en día se ha convertido en una herramienta incluso para desarrolladores web. Adicionalmente, se ha convertido en una plataforma de trazado para un sinnúmero de otras aplicaciones. Gnuplot está bajo desarrollo activo desde 1986.

2.3.4. Estado del arte

Hoy en día, uno de los campos que más ha destacado en cuanto a su avance tecnológico ha sido la informática. Sin duda, es una ciencia que ha permitido el desarrollo de numerosas aplicaciones gracias a los sistemas computacionales, tal es el

caso muchas nuevas plataformas de programación, algunas con su propio lenguaje. La computación ha hecho del manejo de información una tarea muy sencilla y cada vez más poderosa.

La informática ha despegado en numerosos campos, y uno de los más importantes es la industria, pues ha habido un significativo crecimiento en el área gracias a la mecatrónica y a la ingeniería de control. Muchos de los procesos industriales actualmente son supervisados por algoritmos previamente programados, de acuerdo a los requerimientos del proceso, y han superado notablemente a los antiguos sistemas de control debido a su flexibilidad, adaptabilidad y facilidad de implementación.

Para diseñar los algoritmos de control, hoy en día existe una vasta cantidad de software específicamente edificado con el fin de recolectar datos, procesarlos y proporcionar resultados óptimos durante y después de la construcción del código. Tal es el caso de LabVIEW, que año tras año ha ido incorporando un sinnúmero de herramientas que lo vuelven muy versátil en cuanto al direccionamiento de información.

Adicionalmente, la tendencia tecnológica hoy se dirige a los sistemas móviles, y a la programación de aplicaciones para los mismos. Las plataformas son cada vez más sencillas y accesibles, y requieren de un conocimiento muy básico. Además, el campo de la comunicación avanza conjuntamente y permite interconectar prácticamente cualquier dispositivo con otro. El desarrollo de software no requiere mayor inversión y es muy fácil de implementar y probar.

El presente proyecto requirió de un sistema de adquisición de datos sólido. Por tanto, se utilizaron muchas de las herramientas de LabVIEW en cuanto a comunicación, adquisición, procesamiento de datos y comunicación. Sin embargo, al tratarse de estaciones de trabajo industriales, no ameritaba programar para dispositivos móviles.

Figura 18. Tendencia tecnológica de las ciencias informáticas.

CAPÍTULO 3
ESTACIÓN DE TRABAJO MESA 3D O 3DTISCH

La estación de trabajo Mesa 3D o 3DTisch es una de las estaciones que conforma el Laboratorio de Propiedades de Neumáticos o TPL (sigla en inglés de Tire Property Lab). Ésta se encarga de analizar, escanear y generar un modelo computacional de la sección transversal de un corte de tal característica de cualquier neumático. Dicho modelo es después llevado a un software desarrollado por el Departamento de Ingeniería de la Universidad de Ciencias aplicadas de Brandemburgo, llamado RMOD-K 7, que se encarga de analizar y determinar los parámetros que darán lugar al modelo paramétrico del neumático sujeto a prueba.

Principalmente, la 3DTisch está compuesta por una máquina de Control Numérico Computarizado, un módulo de adquisición de datos y otro de procesamiento de información. La estación es monitoreada y controlada por el usuario a través del módulo de procesamiento, el cual también permite al usuario obtener y visualizar los resultados.

Cabe señalar que la consideración más significativa del diseño fue la duración de la prueba, y dicho parámetro puede ser modificado a través de la interfaz mediante el cambio de la precisión de los resultados.

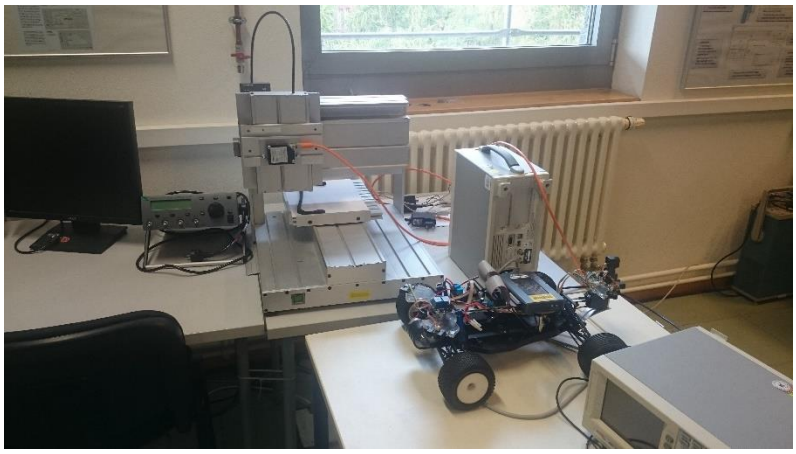


Figura 19. Estación de trabajo 3DTisch.

3.1. CORTE DE SECCIÓN TRANSVERSAL DE UN NEUMÁTICO

Realizar un corte de sección transversal a un neumático es un método común de representar el arreglo interno del mismo en dos dimensiones. Dichas secciones son estudiadas con el fin de determinar ciertos parámetros que influyen el rendimiento de un neumático durante rutinas estáticas y dinámicas. Uno de estos parámetros es el ancho de la sección y es fundamental puesto que determina el ancho del rin sobre el cual el neumático es montado.

El ancho de la sección transversal de un neumático es la medida entre la pared lateral interna y la pared lateral externa, excluyendo cualquier elemento de protección, decoración y labrado, en el punto más ancho del neumático. Esta medición es realizada sin carga sobre el neumático y después de que el mismo haya sido correctamente montado sobre el rin para el cual el neumático fue industrialmente diseñado. (Tire Rack, 2016)

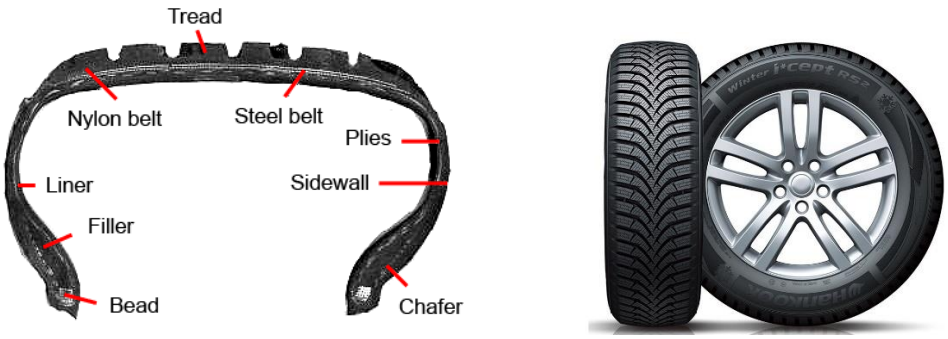


Figura 20. Corte de sección transversal de un neumático.

3.2. HARDWARE CONSTITUTIVO DE LA ESTACIÓN

3.2.1. Máquina CNC EMC KOSY2-MCS

La EMC KOSY2-MCS es una máquina de Control Numérico Computarizado (CNC) de multiprocesador diseñada por Elektronik & Mechanik GmbH. El Punto Central de la Herramienta o TCP (sigla en inglés de Tool Center Point) se mueve a lo largo de tres ejes: el eje X, el eje Y y el eje Z, es decir, se desplaza con base en un

sistema de coordenadas cartesianas. Por otra parte, el área efectiva del TCP es 250 mm × 330 mm × 108 mm respectivamente.

La base de la máquina está hecha con perfiles de aluminio, sujetados entre sí. En consecuencia, la máquina es bastante flexible y adaptable a muchas aplicaciones que requieran de CNC. Igualmente, la máquina está impulsada por motores de pasos de muy alto rendimiento, los cuales le proporcionan a la máquina una resolución y reproducibilidad muy precisa.

Figura 21. Máquina CNC KOSY2-MCS.

3.2.2. Transductor analógico láser Waycon LAS-T5-250-10V

El transductor analógico de desplazamiento láser Waycon LAS-T5-250-10V es un sensor láser que cubre rangos de medición desde 50 a 300 mm y entrega una señal de salida certera, proporcional a la distancia detectada.

Figura 22. Sensor laser LAS-T5.

3.2.3. Módulo de entrada analógica National Instruments NI 9215

El módulo NI 9215 es un módulo de entrada analógica simultánea compatible con los sistemas NI CompactDAQ y CompactRIO. Incluye cuatro canales de entrada analógica muestreados simultáneamente y convertidores analógico-digital (ADCs) de 16 bits de registro sucesivo de aproximación (SAR). (National Instruments, 2016)

Los módulos de la serie NI 100 C son usados comúnmente para aplicaciones de control, medición y comunicación. Además, los módulos de la serie C pueden ser conectados a cualquier sensor o bus, adquiriendo mediciones de alta precisión y cumpliendo los requerimientos de aplicaciones avanzadas de control y adquisición de datos.

Figura 23. Módulo de entrada analógica simultánea NI 9215.

3.2.4. Chasis de ranura USB National Instruments NI cDAQ-9171

El chasis NI cDAQ-9171 es un chasis USB CompactDAQ de una ranura alimentado por bus y diseñado para sistemas sensoriales de medición pequeños y portables. Combinado con cualquiera de los módulos de la serie NI C, el ensamble puede convertirse en un sistema de entradas y salidas analógicas, entradas y salidas digitales, o en un sistema contador/temporizador.

Los módulos están disponibles para una variedad de medidas de sensores incluyendo termopares, RTDs, galgas extensiométricas, transductores de presión y carga, celdas de torsión, acelerómetros, medidores de flujo, codificadores y micrófonos. Los sistemas NI CompactDAQ combinan medidas de sensores con señales de voltaje, corriente y digitales para crear sistemas personalizados de señales con un solo cable USB a la PC o laptop. (National Instruments, 2016)

Figura 24. Chasis de Ranura USB NI CompactDAQ.

3.2.5. Computadora personal con Windows 8.1 ×64 bits

Para la presente estación, se utilizó como módulo de procesamiento, monitoreo y control una computadora con Windows 8.1 ×64 bits. Dicha computadora posee el software necesario para el desarrollo del sistema de control de la estación 3DTisch, explicado en el capítulo anterior del presente proyecto.

3.3. CONTROL Y SUPERVISIÓN DE LA ESTACIÓN

3.3.1. Adquisición de datos y procesamiento de la señal del sensor

Algoritmo de adquisición y procesamiento

Para recibir la señal emitida por el sensor láser, el mismo fue ubicado a lo largo del eje Z de la máquina CNC, siendo de este modo capaz de medir el cambio de altura debido a la presencia del corte de sección transversal del neumático de prueba sobre la mesa de trabajo. Así también, la señal de salida del sensor fue conectada al módulo

de entrada analógica y luego al chasis de ranura USB con el fin de que la computadora sea capaz leerla y procesarla.

A continuación se muestra el algoritmo desarrollado en LabVIEW para la adquisición y el procesamiento de la señal de salida proveniente del sensor.

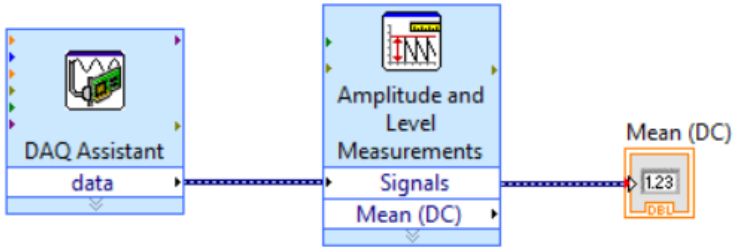


Figura 25. Algoritmo de procesamiento y adquisición de la señal sensorial.

El Asistente DAQ permite al programador establecer las configuraciones de medición de la señal de entrada, mediante la selección de escalas y rangos esta última. Estos valores son cruciales, pues determinan el rendimiento de la señal de entrada durante toda la ejecución del programa principal de control.

La herramienta de Amplitud y Niveles de Medición calcula el valor medio de la señal de entrada, provista por el sensor láser. Considerando que el láser emitido por el sensor no determina una coordenada exacta sobre la mesa de trabajo de la máquina CNC, la mencionada herramienta computa el valor de medición relacionada con el área efectiva del punto láser.

Biblioteca de Enlace Dinámico o DLL (Sigla en inglés de Dynamic-Link Library)

El control principal de la estación de trabajo 3DTisch fue desarrollado en Visual Studio y codificado en C++. Por lo tanto, para transmitir los datos desde una aplicación de LabVIEW hacia un archivo C++ se creó una DLL. Este recurso permite a un programa desarrollado en C++ obtener información desde cualquier otro programa, y en el presente caso, desde una interfaz virtual de LabVIEW, como un función sencilla, la cual puede ser llamada en cualquier lugar durante la ejecución del programa.

3.3.2. Interfaz Gráfica de Usuario

Para poder controlar y supervisar cada uno de los aspectos de la estación de trabajo, se desarrolló una Interfaz Gráfica de Usuario en Visual Studio. Dicha GUI es sencilla, clara y amigable al usuario.

La GUI se comunica directamente con el controlador incorporado en la máquina CNC, a través de un protocolo serial. Así, el controlador recibe los comandos (caracteres en sistema hexadecimal) enviados a través del disparo de los diferentes eventos del programa y los ejecuta. Una vez terminados, el controlador envía una señal de contestación.

El manual de comandos fue proporcionado por la empresa fabricante y detalla cada una de las acciones que la máquina CNC puede completar. Dicho manual está adjunto al presente documento en la sección de anexos.

Así, la siguiente figura muestra la GUI diseñada para la estación de trabajo 3DTisch. La misma se halla compuesta de elementos comunes para monitorear y controlar los componentes de la estación de trabajo. Entre ellos destacan los mencionados en el capítulo anterior, así como los cuadros de texto, barras de progreso, cuadros de lista, etiquetas, etc.

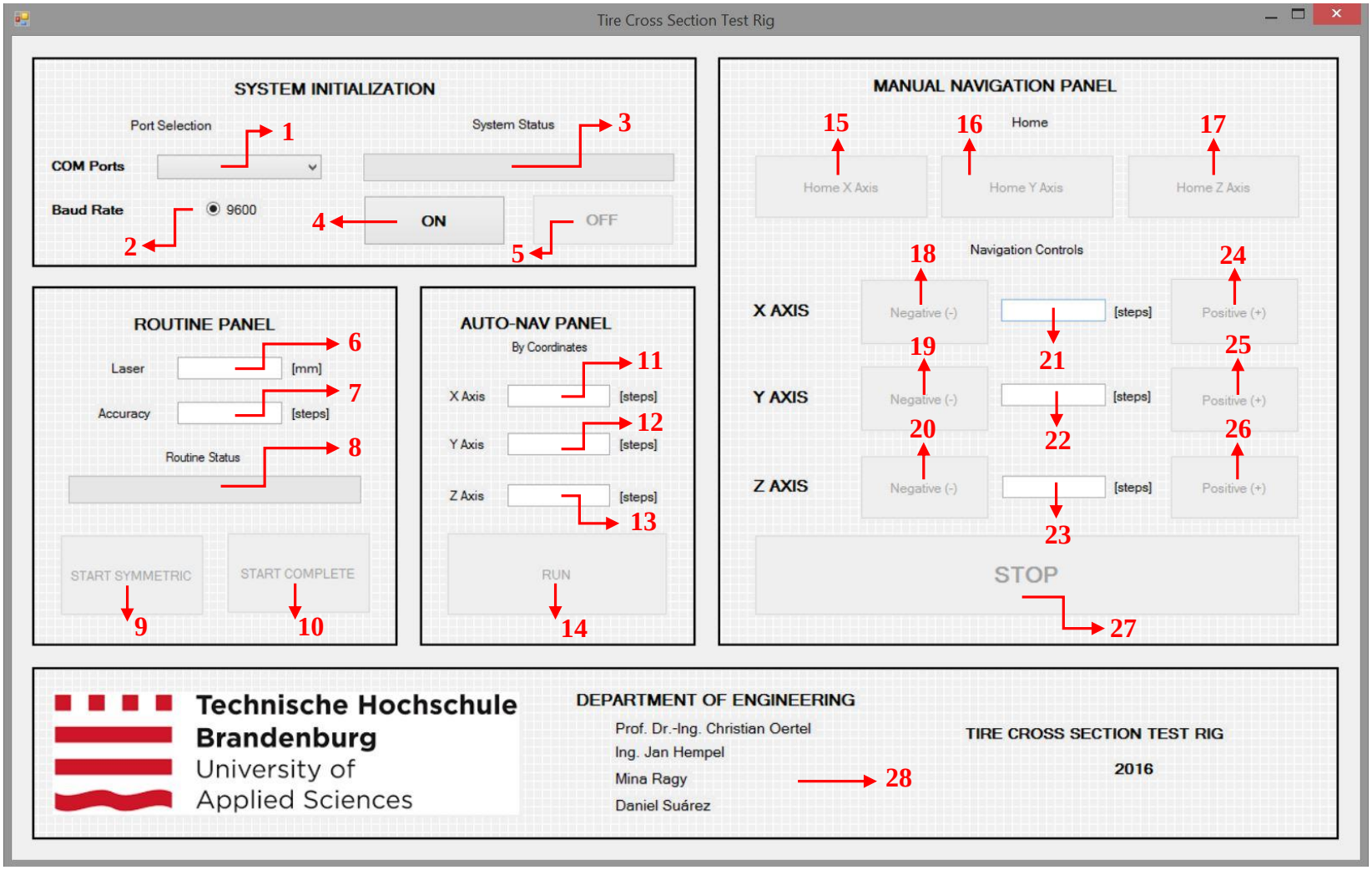


Figura 26. GUI diseñada para la estación de trabajo 3DTisch.

Descripción de los elementos de la GUI

1. Cuadro de lista COM Ports para seleccionar el puerto de comunicación disponible.
2. Botón de radio para elegir la Tasa de Baudios o Baud Rate, la cual determina el número de unidad de señales por segundo.
3. Barra de progreso para mostrar visualmente al usuario que el puerto serial elegido se encuentra habilitado y que la máquina está ubicada en su posición de inicio o home.
4. Botón de encendido ON, el cual inicia el sistema, habilita el puerto serial elegido, sitúa a la máquina en la posición home y permite al usuario interactuar con los botones de operación deshabilitados por defecto.
5. Botón de apagado OFF, encargado de deshabilitar el puerto serial de comunicación así como los botones de operación de la GUI.
6. Cuadro de texto Laser delegado de desplegar la medida obtenida del láser en milímetros.
7. Cuadro de texto de precisión o Accuracy, el cual permite al usuario ingresar el valor deseado en pasos de precisión para las rutinas de escaneo.
8. Barra de progreso para definir el estado de las rutinas de escaneo de los cortes de sección transversal de los neumáticos sujetos a prueba.
9. Botón de inicio de la rutina Symmetric.
10. Botón de inicio de la rutina Complete.
11. Cuadro de texto para ingresar una coordenada en pasos para el eje X.
12. Cuadro de texto para ingresar una coordenada en pasos para el eje Y.
13. Cuadro de texto para ingresar una coordenada en pasos para el eje Z.
14. Botón de ejecución RUN, encargado de desplazar el TCP de la máquina a la coordenada previamente escrita en 11, 12 y 13.
15. Botón de posición home para el eje X.
16. Botón de posición home para el eje Y.
17. Botón de posición home para el eje Z.
18. Botón de desplazamiento manual negativo sobre el eje X. La máquina no se detiene al menos que se presione el botón STOP.

19. Botón de desplazamiento manual negativo sobre el eje Y. La máquina no se detiene al menos que se presione el botón STOP.
20. Botón de desplazamiento manual negativo sobre el eje Z. La máquina no se detiene al menos que se presione el botón STOP.
21. Cuadro de texto encargado de mostrar en tiempo real y en todo momento la coordenada del TCP sobre el eje X.
22. Cuadro de texto encargado de mostrar en tiempo real y en todo momento la coordenada del TCP sobre el eje Y.
23. Cuadro de texto encargado de mostrar en tiempo real y en todo momento la coordenada del TCP sobre el eje Z.
24. Botón de desplazamiento manual positivo sobre el eje X. La máquina no se detiene al menos que se presione el botón STOP.
25. Botón de desplazamiento manual positivo sobre el eje Y. La máquina no se detiene al menos que se presione el botón STOP.
26. Botón de desplazamiento manual positivo sobre el eje Z. La máquina no se detiene al menos que se presione el botón STOP.
27. Botón STOP, encargado de detener cualquier movimiento manual de la máquina.
28. Membrete de información.

3.3.3. Funcionamiento de la estación 3DTisch

Inicialización del sistema

Para que la estación de trabajo pueda usarse correctamente, el controlador necesita ser inicializado. Para hacerlo, el controlador interno de la máquina requiere una serie de comandos específicos, indicados en el manual de la máquina. Dichos comandos corresponden a valores hexadecimales, enviados a través de un protocolo serial. Cuatro son los comandos que configuran inicialmente la máquina y no son susceptibles a cambio por el usuario. Así, existe un comando para inicializar la comunicación, para configurar la posición home del TCP, para establecer el área de trabajo y para inicializar los podómetros de cada motor de pasos.

Cabe señalar que las etiquetas de la GUI otorgan al usuario medidas en pasos. Esto fue una consideración especial dada por el Departamento de Ingeniería, para futuras aplicaciones. Sin embargo, el manual de la máquina señala que 100 pasos son equivalentes a 1 mm.

Navegación manual

La GUI está dividida en una serie de paneles. El panel de Navegación Manual o Manual Navigation, como su nombre lo indica, es responsable del movimiento manual de la máquina. Es posible desplazar el TCP sobre cualquier eje y cualquier dirección seleccionada por el usuario, uno a la vez. Así también, la máquina sólo se detendrá si el usuario detiene manualmente el proceso, al presionar el botón STOP.

Además, el panel de Navegación Manual contiene tres cuadros de texto, uno para cada eje. Éstos son los encargados de mostrar en tiempo real y en todo momento las coordenadas cartesianas equivalentes a la posición del TCP.

Navegación automática

El panel de Navegación Automática o Automatic Navigation se encarga de desplazar el TCP a una posición predeterminada por el usuario. Dentro del mismo, al igual que en el panel de Navegación Manual, existen tres cuadros de texto sobre los cuales el usuario puede ingresar un valor para cada coordenada del sistema. Al presionar el botón de Ejecución RUN, el TCP se moverá a dicho punto, desplazándose simultáneamente en las tres sentidos.

Rutinas

Para escanear y modelar el corte de sección transversal de un neumático, dos rutinas fueron desarrolladas. La primera escanea todo el elemento y se denomina COMPLETE. La segunda asume que el elemento es totalmente simétrico (teóricamente lo es) y escanea media sección. Ésta rutina se denomina SYMMETRIC.

Vale mencionar una vez más que el tiempo de la prueba era el factor determinante de la misma. Antes de iniciar cualquier rutina, el usuario debe ingresar un valor de precisión en pasos y comprobar que la máquina esté en la posición home. El tiempo de prueba dependerá de este, así como la resolución final del trazado. El valor de precisión fue necesario puesto que la máquina sólo conoce su posición en cuando se detiene. Así, el desplazamiento de cualquier eje de la máquina se convirtió en discreto.

La última consideración del sistema fue establecer por defecto el ancho del corte del neumático a 1000 pasos (equivalente a 10 mm) ya que justamente por consideraciones de la prueba, las medidas sobre el eje Z eran despreciables. En otras palabras, el TCP tiene sólo dos movimientos, sobre el eje X e Y, y permanece estático sobre el eje Z.

El algoritmo desarrollado para ambas rutinas es muy similar. Por tanto, se describirá la rutina COMPLETE. Lógicamente, la rutina SYMMETRIC fue elaborada para reducir el tiempo de prueba. El corte de sección transversal debe ser ubicado sobre la mesa de trabajo de la estación, previo a la ejecución de la prueba.

Una vez ingresado el valor de precisión y presionado el botón de ejecución START COMPLETE, la máquina comprueba que esté en la posición home (en caso de no estarlo, la máquina ejecuta automáticamente una secuencia home) y arranca. Así:

1. El primer paso es ubicar el TCP (punto láser) sobre aproximadamente la mitad del corte, con un movimiento positivo sobre el eje X.
2. Una vez alcanzado dicho punto, el TCP se desplaza únicamente sobre el eje Y positivo, hasta encontrar cualquier punto del corte de sección transversal.
3. A continuación, la máquina desplaza al TCP el número de pasos establecido por el usuario, bordeando el corte. Para hacerlo, el algoritmo supervisa, cada vez que existe un desplazamiento, el valor del láser, para establecer sí está el objeto de prueba presente o no. Si el algoritmo detecta que el corte está entre el láser y la mesa de trabajo, entonces sigue los bordes del corte. Caso contrario, se moverá unidireccionalmente hasta hallar de nuevo al corte y seguir de nuevo su borde.

Mientras el programa ejecuta el algoritmo, y el TCP siga los bordes del corte de sección transversal, el programa imprime cada coordenada en un archivo. Al finalizar el programa, el archivo de extensión .DAT puede ser ejecutado por el software de trazado Gnuplot, el cual se encarga de mostrar las figuras finales. Así mismo, dicho

archivo puede ser llevado al software RMOD-K 7 con el fin de continuar con la parametrización del neumático sujeto a prueba.

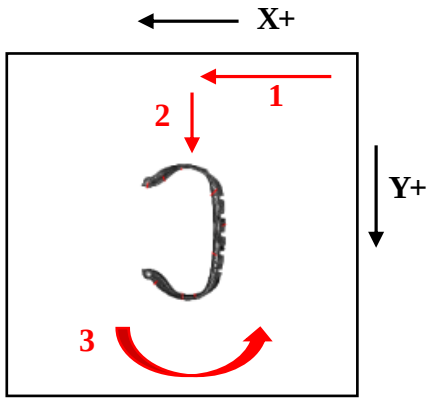


Figura 27. Esquema del algoritmo de escaneo COMPLETE.

3.4. RESULTADOS

Como se había dicho antes, los resultados varían de acuerdo a la precisión ingresada por el usuario, así como el tiempo de la prueba. Por ejemplo, si el usuario ingresa como precisión 50 pasos, el resultado de la prueba será obtenido más rápido que si el usuario hubiese ingresado como precisión 20 pasos. Sin embargo, el resultado sufrirá una penalización en cuanto a resolución y exactitud. De hecho, el sistema está condicionado para trabajar entre 10 y 100 pasos. Cualquier valor fuera del rango provocará sea un tiempo muy elevado o una falla en el algoritmo. Cabe señalar una vez más que 100 pasos equivalen a 1 mm en el desplazamiento de la máquina.

A continuación se presentan los trazados obtenidos de cuatro pruebas realizadas.

3.4.1. Rutina COMPLETE con precisión de 50 pasos

Dicha prueba tuvo una duración de 12 minutos con 27 segundos.

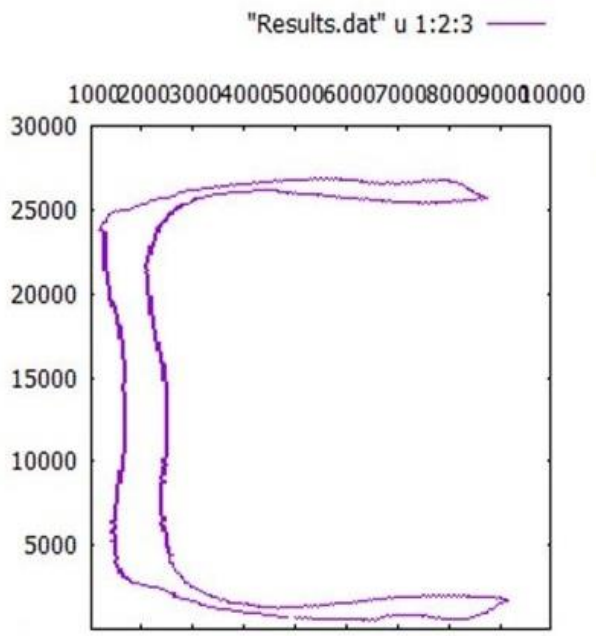


Figura 28. Resultado de la prueba de rutina completa con 50 pasos.

3.4.2. Rutina SYMMETRIC con precisión de 50 pasos

Dicha prueba tuvo una duración de 7 minutos con 43 segundos.

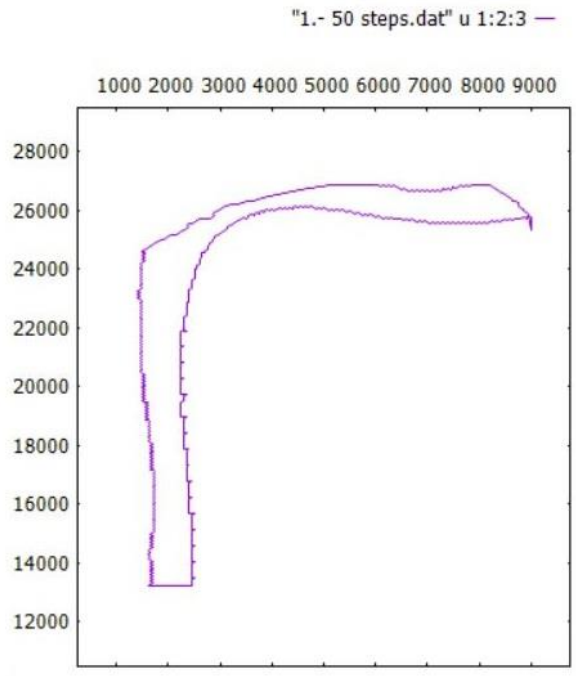


Figura 29. Resultado de la prueba de rutina simétrica con 50 pasos.

3.4.3. Rutina SYMMETRIC con precisión de 20 pasos

Dicha prueba tuvo una duración de 11 minutos con 56 segundos.

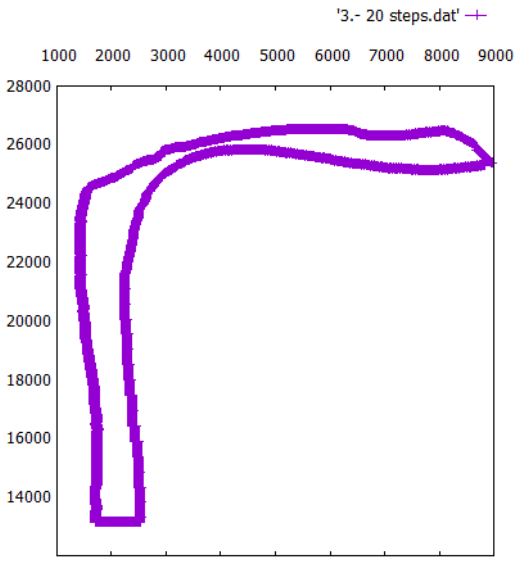


Figura 30. Resultado de la prueba de rutina simétrica con 20 pasos.

3.4.4. Rutina SYMMETRIC con precisión de 100 pasos

Dicha prueba tuvo una duración de 5 minutos con 46 segundos.

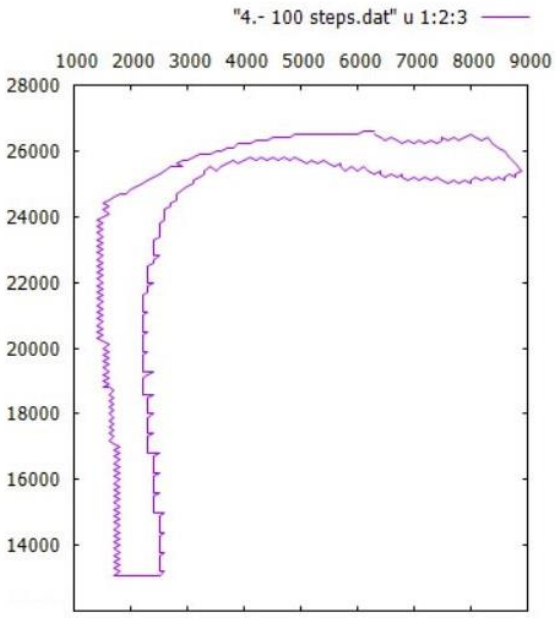


Figura 31. Resultado de la prueba de rutina simétrica con 100 pasos.

3.4.5. Análisis de resultados

Se tiene entonces la siguiente tabla:

Tabla 1. Tabulación de datos referente a las pruebas de escaneo.

Rutina	Precisión (pasos)	Tiempo de ejecución (s)
Simétrica	20	716
Simétrica	50	463
Simétrica	100	346
Completa	50	747

Al observar los resultados, se puede decir que la hipótesis previamente planteada es verdadera. El resultado de la prueba depende de la precisión ingresada por el usuario en cuanto a exactitud y resolución. Así también, el tiempo es inversamente proporcional al valor de precisión.

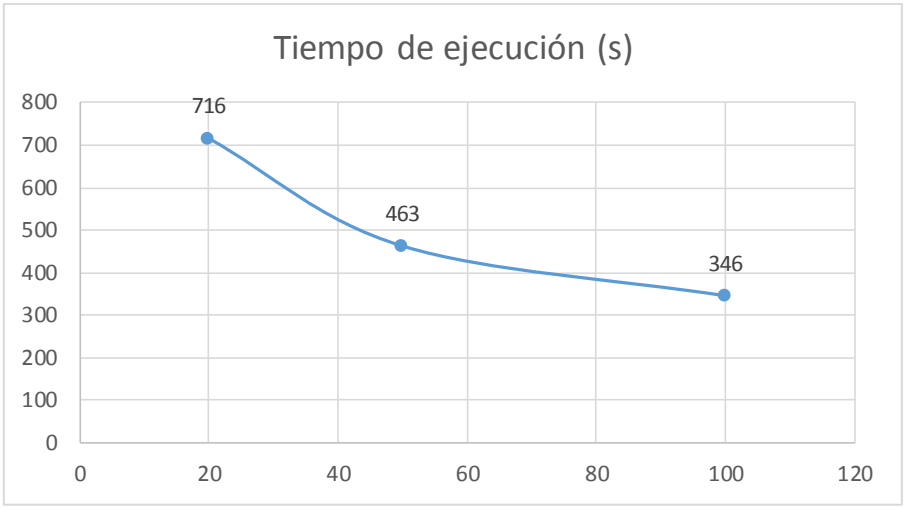


Figura 32. Tiempo de ejecución de la rutina simétrica en función de la precisión

Después de analizar los resultados, se definió como valor de precisión para la prueba por defecto 50 pasos, es decir 0,5 mm de exactitud.

CAPÍTULO 4
ANÁLISIS PRELIMINAR DE LA ESTACIÓN DE PRUEBAS

Como se ha mencionado en secciones anteriores, el banco de pruebas estáticas y dinámicas para neumáticos está compuesto por dos estaciones de trabajo. La segunda, propiamente la estación de pruebas o TTR (sigla en inglés de Tire Test Rig) juega un papel muy importante puesto que es capaz de realizar algunas de las pruebas mecánicas requeridas para extraer las propiedades necesarias para el modelamiento paramétrico del cual se encarga el Laboratorio de Propiedades de Neumáticos.

La TTR lleva tres años de construcción dentro del Departamento de Ingeniería de la Universidad de Ciencias Aplicadas de Brandemburgo (UCAB) y día a día, es objetivo de investigación puesto que permite la incorporación de nuevos módulos, que permitan incrementar el alcance del sistema.

El presente capítulo tiene como objetivo describir brevemente la composición y las principales características de la segunda estación de trabajo, denominada TTR, en términos de capacidad y versatilidad. Así también, el presente introduce el diseño preliminar de la Interfaz Gráfica de Usuario (GUI) que monitoreará y controlará cada una de las pruebas que la estación puede realizar.

Figura 33. Estación de pruebas de la UCAB.

4.1. CAPACIDAD DE LA TTR EN APLICACIONES DE MEDICIÓN

La TTR ha sido desarrollada con el fin de medir las principales propiedades de los neumáticos a través de la ejecución de diferentes pruebas. Para el presente proyecto, se concluyó con éxito las pruebas de rigidez vertical estática y dinámica para neumáticos. Cabe señalar que la TTR consta con bastantes componentes para la realización de otras pruebas. Sin embargo, muchas funcionan con otros sistemas de control, ajenos al propuesto en este proyecto.

4.1.1. Componentes de la estación de pruebas

- **Computadora personal con Windows 7 ×32 bits:** Para la presente estación, se utilizó como módulo de procesamiento, monitoreo y control una computadora con Windows 7 ×32 bits y el software necesario para la programación de los módulos de medición y la GUI.
- **Placa de zona de contacto:** Es una placa metálica cuya función es soportar la carga ejercida sobre el neumático durante la prueba de rigidez vertical estática. Así también, cumple la función de sostener la matriz sensorial de 128 × 128 para la prueba de zona de contacto, encargada de determinar la huella del neumático. Dicha prueba compete a otro sistema de control.

Figura 34. Placa de zona de contacto de la TTR.

- **Huso:** El huso consta de un mecanismo de tornillo sin fin acoplado al soporte del neumático, el cual lo aproxima (y por ende el neumático) hacia la placa de zona de contacto o hacia el tambor. Indirectamente, es el mecanismo encargado de someter al neumático a una carga y desplazarlo a lo largo de la estación.

Figura 35. Huso de la TTR.

El mecanismo de tornillo sin fin que acciona el huso se encuentra acoplado a un motor eléctrico, cuyo sentido de rotación determina la dirección de movimiento lineal del huso. La velocidad de dicho motor es controlada a través de una señal analógica de voltaje comprendida entre 0 y 10V. Sin embargo, antes de enviar la señal analógica, es necesario transmitir dos señales digitales antes. Una habilita el motor y la otra define el sentido de rotación.

Figura 36. Motor eléctrico de accionamiento del uso.

Del otro extremo del mecanismo de tornillo sin fin, se halla ensamblado un sensor incremental angular, encargado de medir el desplazamiento angular del tornillo, en grados sexagesimales o en radianes. Este sensor es crucial, pues indirectamente es capaz de medir el desplazamiento lineal que corresponde al valor de deformación del neumático durante cualquiera de las pruebas de rigidez vertical.

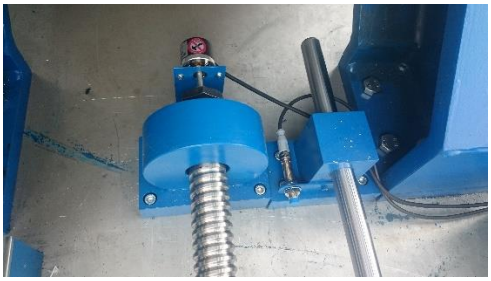


Figura 37. Sensor incremental de desplazamiento angular.

Como última consideración, el huso está ensamblado también a un sistema de sujeción neumático. Cuando el motor se habilita (recibe la señal digital), el sistema desacopla el freno y permite el desplazamiento del huso y viceversa. Dicho sistema fue incorporado puesto que cuando el neumático esté sujeto a cargas significativas, se requiere que el sistema esté completamente estático y no permita retroceso alguno.

Figura 38. Sistema de sujeción neumático para el huso.

- **Tambor:** El tambor es un cilindro metálico encargado de rotar al neumático durante la prueba de rigidez vertical dinámica. Su rotación es generada a partir de un mecanismo de transmisión de bandas, acoplado a un motor eléctrico de alta potencia.

Figura 39. Tambor de la TTR.

El motor eléctrico del sistema tiene su propio controlador, y para accionarlo, requiere únicamente de una señal digital.

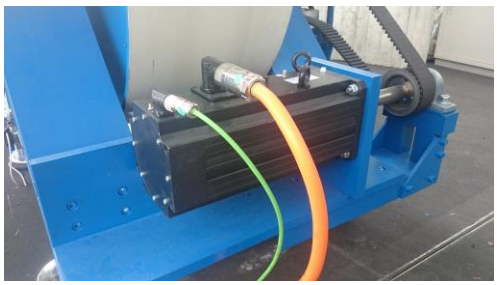


Figura 40. Motor eléctrico de accionamiento del tambor.

Como en el caso del huso, el tambor también tiene un sistema de frenado neumático que se habilita o deshabilita de acuerdo al caso necesario.

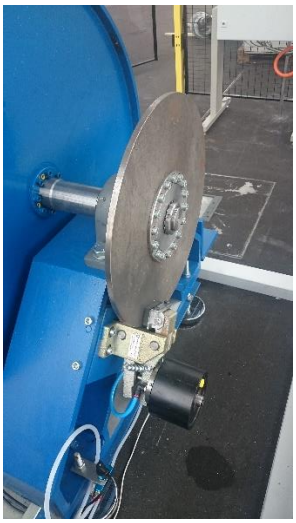


Figura 41. Freno neumático para el tambor de la TTR.

- **Soporte del neumático:** El soporte del neumático se encuentra acoplado al huso y cumple la función de sujetar al neumático sujeto a prueba. Consta de un tambor al cual se ensambla el neumático directamente, como si fuera el tambor de un auto, y de dos elementos adicionales que intervienen en las pruebas.

  El primer elemento es el sensor de fuerza de seis componentes (ver Figura 42). Dicho sensor está acoplado al tambor del neumático y será el encargado de medir las tres componentes de fuerza lineal y los tres momentos a los cuales estará sujeto el neumático durante las pruebas de rigidez vertical.

Figura 42. Soporte del neumático.

El segundo elemento es un motor de pasos, encargado de rotar el neumático a través de un mecanismo de cadena. Dicho elemento tiene su propio controlador, conectado al sistema de control principal. La prueba que pretende realizar junto con el robot (siguiente sección) es la de uniformidad completa, y no se consideró en el presente proyecto, pese a que la GUI fue desarrollada.

Figura 43. Motor de pasos del soporte del neumático.

- **Robot:** El robot Kawasaki RSOSN ubicado frente a la TTR tiene como principal objetivo realizar la medición de la sección exterior de un neumático durante la prueba de uniformidad completa. Trabaja conjuntamente con el motor de pasos ubicado en el soporte del neumático. En su TCP, se encuentra ubicado un sensor de desplazamiento láser.

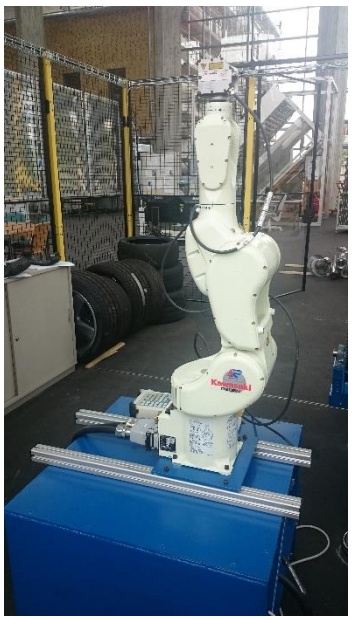


Figura 44. Robot Kawasaki RSOSN de la TTR.

4.2. PRUEBAS ESTÁTICAS Y DINÁMICAS DE LA TTR

Para medir los distintos parámetros de un neumático, se debe ejecutar diferentes pruebas con procedimientos específicos. Para el presente proyecto se desarrollaron dos pruebas de rigidez vertical, así como la prueba de uniformidad de corte realizada por la 3DTisch (Ver Capítulo 3). Pese a que se desarrolló la sección de la GUI para la prueba de uniformidad completa, ésta no pudo ser concluida debido a un percance con el controlador del robot.

Brevemente, la prueba de uniformidad consiste en medir el perfil exterior transversal de un neumático cuando esté está incorporado a su rin y debidamente inflado. Para ello, el TCP del robot bordearía el neumático y el sistema de control guardaría los datos obtenidos. Posteriormente, el motor de pasos rotaría un cierto ángulo al neumático y la prueba continuaría. Se parece mucho a la prueba realizada por la estación 3DTisch, a diferencia de que ésta tiene todo el ensamble del neumático.

Para empezar, la rigidez es la característica que presenta un objeto al resistirse a la deformación, producto de una fuerza impresa sobre éste. Así, la rigidez es igual a la carga ejercida sobre el objeto dividida para su deformación. Entonces, la rigidez vertical de un neumático es equivalente a la carga sometida sobre el mismo dividida para su deformación (o deflexión), correspondiente a dicha carga.

El procedimiento específico de la prueba de rigidez vertical estática es:

1. El usuario ingresa la carga que desea aplicar sobre el neumático, y así determinar la rigidez vertical correspondiente a la carga.
2. El neumático se desplazará hacia la placa de zona de contacto, y se detendrá al entrar en contacto con la misma.
3. A partir de ese instante, se disparará el algoritmo de control, capaz de someter al neumático a la fuerza establecida por el usuario. El algoritmo de control se explicará en capítulos posteriores.
4. Una vez que el neumático alcance la fuerza objetivo, el sistema se detendrá, y determinará los valores finales de la rigidez vertical.

La única diferencia existente entre la prueba estática y dinámica es que en prueba de rigidez vertical dinámica, la rigidez es calculada mientras el tambor rota el neumático. Consecuentemente, el neumático se desplaza hacia el tambor en lugar de hacia la placa de zona de contacto.

Para finalizar, se desarrolló también un control manual para la prueba, donde el usuario tiene la capacidad de manipular el huso y el tambor, y aplicar cualquier fuerza mientras los valores se despliegan en tiempo real.

4.3. DISEÑO Y CONTROL PRELIMINAR DE LA GUI DE LA TTR

Con el fin de controlar la estación de pruebas o Tire Test Rig (TTR), se desarrolló en un inicio un concepto de Interfaz Gráfica de Usuario en la plataforma de Visual Studio. La idea general de la aplicación se elaboró de la siguiente manera.

1. Al iniciar el programa, el neumático debe estar montado, el sistema neumático habilitado y el usuario se encontrará frente a una página de inicio o home. A través de ésta, el usuario será capaz de manipular manualmente cada uno de los actuadores de la TTR. Así también, el usuario puede seleccionar una prueba de la lista de pruebas, la que desee realizar sobre el neumático.
2. Al seleccionar cualquiera de las pruebas, se cerrará la ventana home y se desplegará la ventana correspondiente a la prueba, con su respectivo control.

4.3.1. Ventana preliminar de inicio o home

La figura mostrada a continuación describe la página home elaborada en un inicio, como base de la GUI.

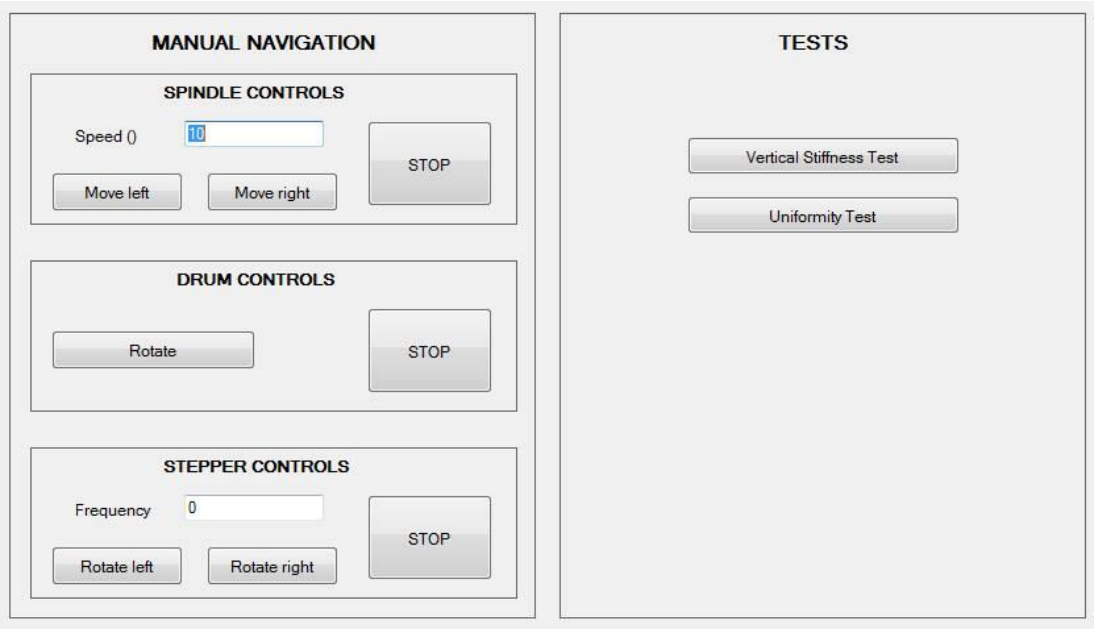


Figura 45. Diseño preliminar de la página home de la TTR.

A continuación se detallan los elementos que la conforman:

- **Botón Move left:** La ejecución de este botón moverá el neumático hacia el tambor.
- **Botón Move right:** La ejecución de este botón moverá el neumático hacia la placa de zona de contacto.
- **Cuadro de texto Speed:** El usuario ingresa una determinada velocidad como un valor porcentual entre 0 y 100% para el motor eléctrico del huso. Dicho valor es proporcional al rango de la señal analógica que varía entre 0 y 10V.
- **Botón STOP (SPINDLE CONTROLS):** La ejecución de este botón detendrá al motor eléctrico del huso, es decir detendrá al soporte del neumático.
- **Botón Rotate:** El tambor empezará a rotar después de la ejecución de este botón. Cabe mencionar que la velocidad y sentido de giro del tambor están preestablecidas dentro de su controlador interno.
- **Botón STOP (DRUM CONTROLS):** La ejecución de este botón detendrá la rotación del motor del tambor.
- **Cuadro de texto Frequency:** El usuario ingresa un determinado valor de frecuencia que se relaciona directamente con el paso del motor de pasos encargado de rotar el neumático.

- **Botón Rotate left:** La ejecución de este botón rotará en sentido anti horario al neumático.
- **Botón Rotate right:** La ejecución de este botón rotará en sentido horario al neumático.
- **STOP (STEPPER CONTROLS):** La ejecución del presente botón detendrá la rotación del neumático. Cabe señalar que todos los controles del panel STEPPER CONTROLS están programados en base, pero carecen de funciones debido a un inconveniente con el controlador del motor de pasos. Su implementación será objetivo de futuras aplicaciones.
- **Botón Vertical Stiffness Test:** La ejecución de este botón cerrará la ventana de inicio y desplegará la venta de control para la sección de pruebas de rigidez vertical.
- **Botón Uniformity Test:** La ejecución de este botón cerrará la ventana de inicio y desplegará la venta de control para la sección de prueba de uniformidad completa. Dicha ventana se desarrolló pero no se implementó en el sistema.

4.3.2. Ventana preliminar de pruebas de rigidez vertical

Para acceder a ella, el usuario debe haber presionado el botón correspondiente en la página de inicio. La ventada de pruebas de rigidez vertical se caracteriza por tener tres pestañas. Una para la prueba estática, otra para la dinámica y la tercera para la prueba de rigidez vertical manual, a pesar de que esta última puede considerarse también como un control manual más personalizado.

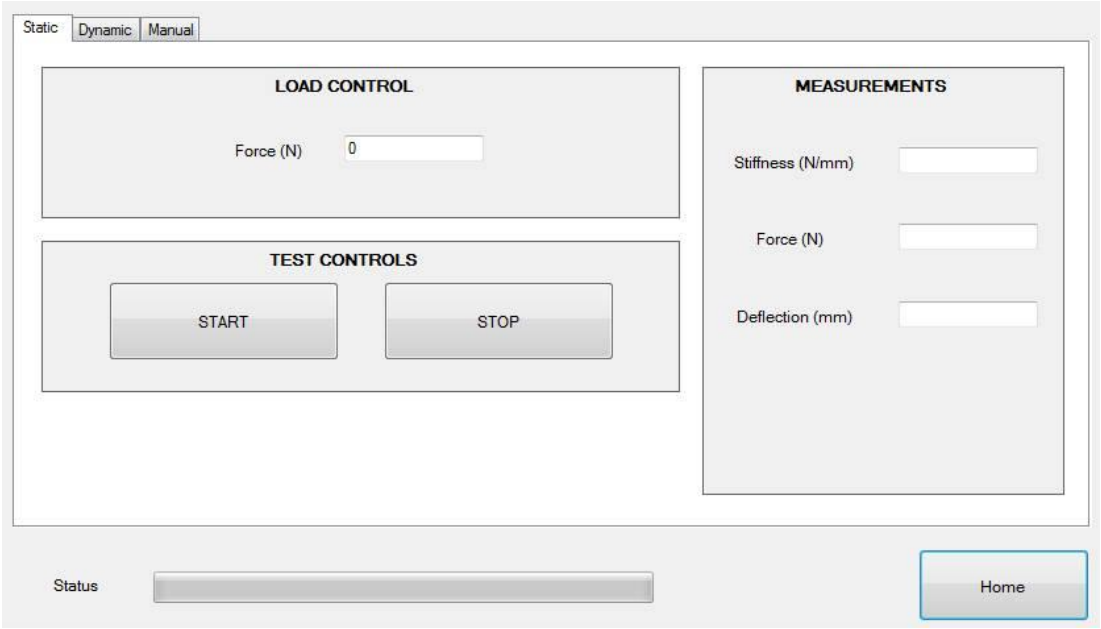


Figura 46. Pestaña preliminar de rigidez vertical estática.

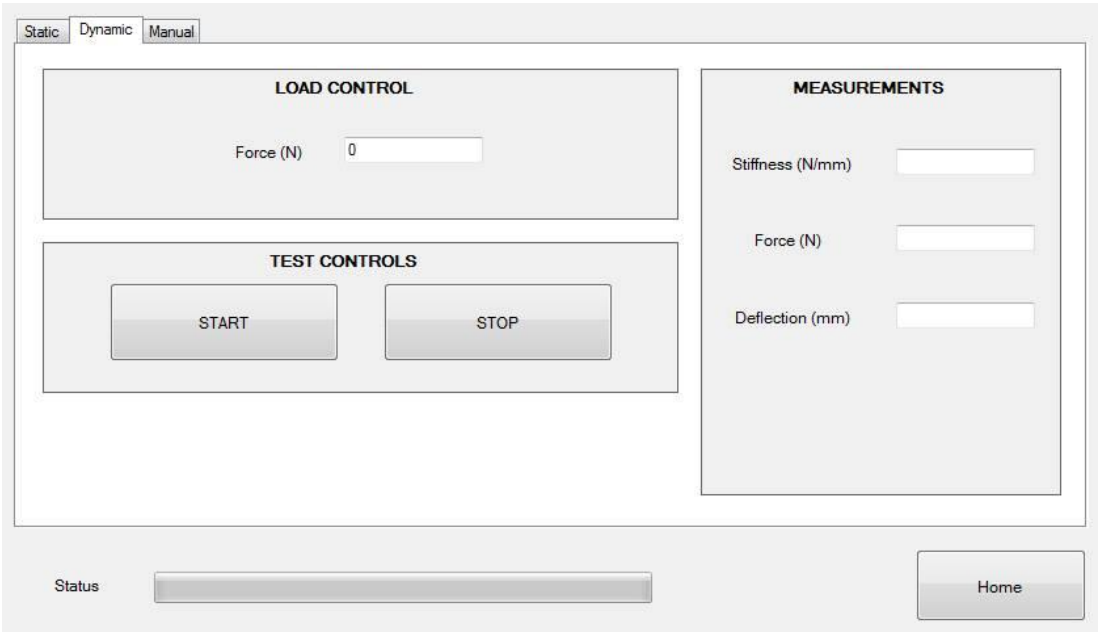


Figura 47. Pestaña preliminar de rigidez vertical dinámica.

Como se puede observar, ambas pestañas son idénticas. La única diferencia se encuentra en la programación de la prueba, pues cuando se inicia la prueba de rigidez vertical dinámica, el tambor empieza a girar y lógicamente, el neumático se aproxima hacia él.

Seguido, se detalla brevemente cada uno de los componentes de las pestañas:

- **Cuadro de texto Force (LOAD CONTROL):** El usuario ingresa una determinada carga entre 200N y 10000N. Dicha cifra será el valor objetivo del controlador, y se determinará la rigidez vertical del neumático sujeto a la misma.
- **Cuadro de texto Force (MEASUREMENTS):** Este cuadro de texto mostrará cada cierto tiempo la carga actual en Newton a la que está sometido el neumático.
- **Cuadro de texto Stiffness:** Este cuadro de texto mostrará cada cierto tiempo la relación entre la carga y la deflexión del neumático, es decir el valor de la rigidez vertical.
- **Cuadro de texto Deflection:** Este cuadro de texto mostrará cada cierto tiempo la deflexión del neumático, al entrar en contacto con la placa de zona de contacto o con el tambor.
- **Botón START:** Este botón inicia la prueba de rigidez vertical, montado ya el neumático.
- **Botón STOP:** La prueba se detiene inmediatamente después de la ejecución de este botón.
- **Botón Home:** El usuario puede regresar a la página de inicio al presionar este botón. Como consideración, este botón permanece bloqueado mientras se esté realizando una prueba.
- **Barra de progreso de estado:** Se encarga de mostrar al usuario el estado de la prueba que esté ejecutándose.

4.3.3. Ventana preliminar de la prueba de rigidez vertical manual

La siguiente figura detalla la pestaña correspondiente a la prueba de rigidez vertical manual.

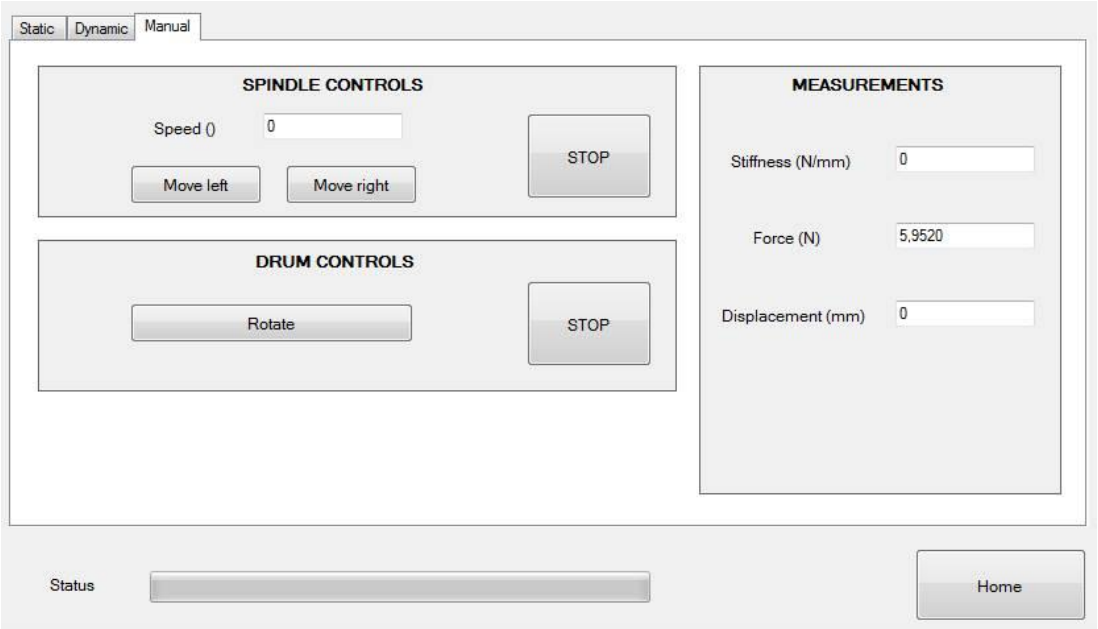


Figura 48. Pestaña preliminar de rigidez vertical manual.

A continuación se presenta el detalle de cada uno de los componentes de la pestaña:

- **Botón Move left:** La ejecución de este botón moverá el neumático hacia el tambor.
- **Botón Move right:** La ejecución de este botón moverá el neumático hacia la placa de zona de contacto.
- **Cuadro de texto Speed:** El usuario ingresa la velocidad a la que desea que se desplace el uso, como un valor porcentual.
- **Botón STOP (SPINDLE CONTROLS):** La ejecución de este botón detendrá el desplazamiento del soporte del neumático.
- **Botón Rotate:** El tambor empezará a rotar después de la ejecución de este botón.
- **Botón STOP (DRUM CONTROLS):** La ejecución de este botón detendrá la rotación del motor del tambor.
- **Cuadro de texto Force (MEASUREMENTS):** Este cuadro de texto mostrará cada cierto tiempo la carga actual en Newton a la que está sometido el neumático.
- **Cuadro de texto Stiffness:** Este cuadro de texto mostrará cada cierto tiempo el valor de la rigidez vertical.
- **Cuadro de texto Deflection:** Este cuadro de texto mostrará cada cierto tiempo la deflexión del neumático durante la prueba.

- **Botón Home:** El usuario puede regresar a la página de inicio al presionar este botón.
- **Barra de progreso de estado:** Se encarga de mostrar al usuario el estado de la prueba que esté ejecutándose. Para la presente prueba se encuentra deshabilitada.

4.4. FUNCIONES DE CONTROL DE LA TTR

Para controlar la estación de pruebas y ejecutarlas a través del código fuente desarrollado en C++, se diseñaron un sinnúmero de funciones. Cada vez que el usuario presiona un botón dentro de una ventana específica, una función o un conjunto de ellas se dispara con el fin realizar la tarea correspondiente al evento de ejecución del botón. En la siguiente sección se explican las funciones preliminares de cada ventana.

4.4.1. Funciones de la ventana de inicio

Las primeras tres funciones codificadas dentro de la clase de esta ventana son las responsables del movimiento del neumático. Cuando el usuario presiona sea el botón *Move left* o sea el botón *Move right*, el evento llama a tres funciones y a una secuencia de instrucciones programadas dentro del propio evento. Así:

1. La primera función define el sentido de giro del motor mediante una señal digital, determinando un desplazamiento hacia la izquierda (hacia el tambor) o hacia la derecha (hacia la placa de zona de contacto).
2. La segunda función es la encargada de habilitar o deshabilitar el motor mediante otra señal digital. Dicha señal también deshabilita o habilita el sistema de freno neumático mencionado anteriormente.
3. La última función emite la señal analógica de 0 a 10V correspondiente a la velocidad ingresada por el usuario.

La siguiente función elaborada se encarga de detener el movimiento del neumático. Por tanto, utiliza dos de las previas funciones: la función de señal digital para deshabilitar y la función de señal analógica para establecer como velocidad 0.

```
//This function allows the user to move the spidle to the left with a desired speed
//-------------------------------------------------------------------------------------------------------------------------------------------------------
    private: System::Void SpLeft_Click(System::Object^  sender, System::EventArgs^  e)
//-------------------------------------------------------------------------------------------------------------------------------------------------------
            {
                //Setting direction and disabling the neumatic brake
                SDirection(1);//1 towards drum
                SEnable(1);

                //Disabling other buttons for safety reasons
                this->SpRight->Enabled=false;

                if(Convert::ToInt32(this->textBoxSpeed->Text)<=0)
                {
                    MessageBox::Show("Please enter a value larger than zero.", "Warning", MessageBoxButtons::OK, MessageBoxIcon::Warning);
                    this->SpRight->Enabled=true;
                    this->VSTest->Enabled=true;
                    this->UTest->Enabled=true;
                    SEnable(0);
                }
                try
                {
                    Spindle_Speed(Convert::ToInt32(this->textBoxSpeed->Text)/10);
                }
                catch(...)
                {
                    //Pop-up message telling the user that the speed ought to be set
                    MessageBox::Show("Please include the desired speed of the spindle.", "Warning", MessageBoxButtons::OK, MessageBoxIcon::Warning);
                }
            }
```

Figura 49. Codificación del evento *Move left*.

Con el fin de rotar el tambor, se creó una simple función de señal digital encargada de habilitar el motor acoplado al tambor. Cabe mencionar una vez más que tanto el sentido de giro como la velocidad de rotación del tambor fueron preestablecidas dentro de su propio controlador, por razones de seguridad. El usuario sólo lo habilita.

La última función de la ventana es muy sencilla y se encarga de detener la rotación del tambor.

4.4.2. Funciones de la ventana de rigidez vertical

Para el control manual de la prueba de rigidez vertical, las funciones de desplazamiento utilizadas son las mismas que aquellas de la ventana de inicio. Sin embargo, se incorporaron tres adicionales, a través de programación por hilos en secuencia, con el fin de desplegar en los cuadros de texto cada cierto intervalo de tiempo (50 ms), los valores correspondientes de fuerza, deflexión y rigidez vertical.

```
//This thread displays the force during the manual stiffness test
//-----------------------------------------------------------------------------------------------------------
    private: System::Void timer6_Tick(System::Object^  sender, System::EventArgs^  e)
//-----------------------------------------------------------------------------------------------------------
             {
                 //Updating and showing the values while moving manually
                 if(this->textBoxDeflection->Text !="0,0000")
                     this->textBoxStiffness->Text=(Force(xforce_calibration)/deflection).ToString("F4");
                 else
                     this->textBoxStiffness->Text="0,0000";

                 if(Force(xforce_calibration)>=0)
                     this->textBoxForce->Text=Force(xforce_calibration).ToString("F4");
                 else
                     this->textBoxForce->Text=(-Force(xforce_calibration)).ToString("F4");

                 //if (j==0 && (Force(xforce_calibration)>25 || Force(xforce_calibration)<-25))
                 //{


                     timer->Interval = 50;
                     timer->Enabled = true;
                     timer->Start();

                     ThreadStart ^myThreadDelegate = gcnew ThreadStart(this,&Vertical_stiffness::repeat);
                     trd = gcnew Thread(myThreadDelegate);
                     trd->IsBackground = true;
                     trd->Start();
```

Figura 50. Codificación del hilo encargado de desplegar la fuerza.

Las funciones utilizadas para ejecutar las pruebas de rigidez vertical estática y dinámica son las mismas, con la distinción de la dirección de desplazamiento y accionamiento del tambor. Al iniciar la prueba, automáticamente se dispara una secuencia de hilos, primero acercado el neumático hacia la superficie de prueba con una velocidad preestablecida y luego definiendo un controlador digital PID con un tiempo de muestreo de 50 ms, encargado de controlar la fuerza aplicada sobre el neumático a través de la velocidad del motor del huso. Mientras la carga sea baja, la fuerza de acción de la velocidad es alta y viceversa, llegando a estabilizarse con un error despreciable en la fuerza ingresada por el usuario.

```
gain_kp=kp;
gain_ki=ki*kp*Convert::ToInt32(this->textBoxStepTime->Text)/1000;// ki=1/Ti
gain_kd=(kd*kp)/(Convert::ToInt32(this->textBoxStepTime->Text)*0.001);// kd=Td

//Trying to reach the setpoint value of the force
if(setpoint>output)
{
    setpoint=setforcevalue;
    output=readforcevalue;
    error=setpoint-output;
    proportional=gain_kp*error;
    integral=gain_ki*error+int_0;
    derivative=gain_kd*(error-err_0);
    precontrol=proportional+integral+derivative;

    control=precontrol;

    //Boundary conditions
    if(control>max_speed)
        control=max_speed;
    if(control<min_speed)
        control=min_speed;

    this->textBoxStaticSpeed->Text=(control*10).ToString("F4");

    int_0=integral;
    err_0=error;
    return control;
}
```

Figura 51. Extracto de la codificación del controlador PID digital para la fuerza.

La sintonización del controlador fue realizada con prueba y error. Sin embargo, la implementación del mismo se vio afectada por un percance del que se comentará en capítulos posteriores, y por tanto, no fue el algoritmo final de control para las pruebas de rigidez vertical.

Todas las funciones mencionadas fueron automáticamente creadas por la herramienta de DLL de LabVIEW. Las funciones creadas a través de esta herramienta son aquellas que se comunican directamente con el controlador físico de National Instruments y envían directamente las señales a los actuadores, así como reciben las señales de los sensores. Éstas serán descritas a profundidad en capítulos posteriores.

CAPÍTULO 5
INSTRUMENTOS VIRTUALES DE MEDICIÓN Y DLL

El objetivo principal de la estación de pruebas o TTR (Tire Test Rig) es contribuir a la medición de las principales propiedades de cualquier neumático disponible en el mercado, realizando una serie de pruebas, para generar un modelo paramétrico del mismo. Para ello, la TTR requiere de un sistema de instrumentación que permita medir dichas características.

El software de medición que se utilizó para el presente proyecto fue LabVIEW, debido a su flexibilidad y versatilidad. Por ende, el hardware de medición óptimo para el sistema de instrumentación necesitaba ser compatible con la mencionada plataforma. Así, los módulos de medición permiten al usuario obtener los valores de las variables que manejan e influencian la ejecución de las pruebas que la TTR puede realizar.

Consecuentemente, se programaron algunos Instrumentos Virtuales (VI) conjuntamente con Bibliotecas de Enlace Dinámico (DLL) con el fin de obtener los valores entregados por los sensores, procesar las señales, comunicarlas con el código fuente del sistema de control y generar una respuesta hacia los actuadores. Así, LabVIEW se convierte en el puente entre los sensores y actuadores, y el código fuente, limitado simplemente a adquirir y transmitir señales entre los componentes.

El desarrollo del programa de control responsable de la TTR fue realizado en tres fases:

1. La primera fase consistió en analizar la estación de pruebas y generar una GUI preliminar teniendo en cuenta los conceptos de clase, las funciones necesarias, los métodos más importantes, las estructuras de datos, y las diferentes tareas que debía cumplir la TTR.
2. La segunda fase se detalla en el presente capítulo y consistió en diseñar un VI para cada tarea de la TTR con el fin de resolver la transferencia de datos entre la unidad de control y los sensores y actuadores. Cada uno de estos VI se probaron con data sintética (modo offline). Dicho de otro modo, todas las señales requeridas para el control de la TTR fueron simuladas y generadas a través de hardware de adquisición de datos (DAQ) inherente a LabVIEW.

3. La tercera fase consiste en la implementación de la Interfaz Gráfica de Usuario conjuntamente con el sistema de instrumentación, así como la realización de pruebas, optimización del código y refinamiento de procesos.

5.1. HARDWARE CONSTITUTIVO DE LOS VI DE MEDICIÓN

5.1.1. Bloque de conexión blindado NI BNC-2110 para la Serie X y M

El NI BNC-2110 es un bloque de conexión blindado con conectores BNC. Puede ser utilizado con dispositivos de adquisición de datos (DAQ) multifunción y dispositivos de salida analógica. El bloque conector BNC-2110 simplifica la conexión de señales analógicas, algunas señales digitales y dos conexiones definidas por el usuario al dispositivo DAQ y mantiene la integridad de sus medidas con una cubierta blindada. (National Instruments, 2016)

Figura 52. Bloque conector blindado NI BNC-2110.

5.1.2. Bloque de conexión blindado NI BNC-2120 para la Serie X y M

El NI BNC-2110 es un bloque de conexión blindado con conectores BNC. Puede ser utilizado con dispositivos de adquisición de datos (DAQ) multifunción de entrada y salida, así como con dispositivos de salida analógica. El bloque conector BNC-2110 simplifica la conexión de señales analógicas, algunas señales digitales y dos

conexiones definidas por el usuario al dispositivo DAQ y mantiene la integridad de sus medidas con una cubierta blindada. (National Instruments, 2016)

Figura 53. Bloque conector blindado NI BNC-2120.

5.1.3. Tarjeta de adquisición de datos National Instruments PCI-6259

La PCI-6259 de National Instruments es una tarjeta de adquisición de datos (DAQ) multifunción de alta velocidad de la Serie M optimizada para una precisión superior a velocidades de muestreo más altas.

Las tarjetas de alta velocidad de la Serie M ofrecen características avanzadas como el controlador de sistema NI-STC 2, el amplificador programable NI-PGIA 2, y la tecnología de calibración NI-MCal para mejorar el rendimiento y la precisión. Las tarjetas de alta velocidad de la Serie M poseen un amplificador NI-PGIA 2 integrado diseñado para asentamiento rápido a altas velocidades de muestreo, garantizando precisión de 16 bits incluso cuando mide todos los canales a máxima velocidad. (National Instruments, 2016)

Figura 54. Tarjeta de Adquisición de Datos (DAQ) NI PCI-6259.

5.2. VARIABLES DE ADQUISICIÓN Y CONTROL

5.2.1. Prueba de rigidez vertical estática y dinámica

La meta principal de esta prueba es determinar el valor de la rigidez vertical de un neumático mientras rota y mientras no. Entonces, la prueba requiere el control de las siguientes variables:

- La velocidad angular del huso.
- El sentido de giro del huso.
- La velocidad angular del tambor (preestablecida y únicamente habilitada para la prueba de rigidez vertical dinámica.
- El ángulo incremental del huso.
- La carga aplicada sobre el neumático, producida por la rotación del huso.

5.2.2. Prueba de uniformidad completa

La meta principal de esta prueba es definir el perfil exterior de un neumático ensamblado con su respectivo rin y debidamente inflado. Para ello, la prueba requiere la ayuda de un robot, equipado con un sensor láser de desplazamiento. Entonces, la presente prueba necesita el control de las siguientes variables:

- La velocidad angular del huso.
- El sentido de giro del huso.
- El ángulo incremental del huso.

- El número de pasos por giro para la posición angular del neumático.
- La distancia calculada por el sensor láser.

Desafortunadamente, la prueba de uniformidad completa no pudo ser concluida en la realización del presente proyecto, debido a una falla presente en el controlador del robot. No obstante, se programó la ventana base de la misma, para futuras aplicaciones.

Todas las variables anteriores, a excepción del número de pasos por giro para la posición angular del neumático y la distancia calculada por el sensor láser fueron simuladas con dispositivos de adquisición de datos.

5.3. INSTRUMENTOS VIRTUALES DE MEDICIÓN

5.3.1. VI generador de señal analógica

Los VI generadores de señal analógica fueron creados específicamente para trabajar con datos sintéticos. Estos VI fueron utilizados solamente para pruebas y sus señales fueron generadas por las salidas analógicas pertenecientes a los dispositivos DAQ. En consecuencia, las señales son leídas por el mismo dispositivo DAQ que las produce y embebidas dentro del código fuente desarrollado en C++ a través de una Biblioteca de Enlace Dinámico (DLL). Esto permitió analizar el rendimiento de las funciones más significativas del programa mientras se adquiría y generaba señales.

Aproximadamente, todos los VI generadores de señal analógica contienen la misma estructura. Un deslizador responsable de manejar la amplitud de la señal y un dispositivo DAQ, utilizado como generador de señal como muestra el ejemplo:

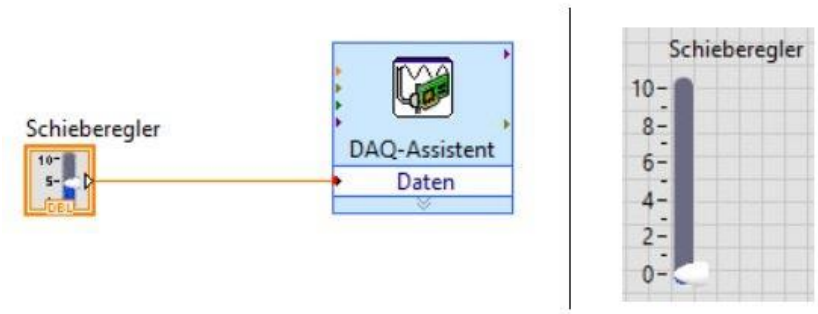


Figura 55. VI generador de señal analógica.

5.3.2. VI de adquisición de señal analógica.

Los VI de adquisición de señal analógica fueron construidos con el fin de leer los valores enviados por los sensores y actuadores de la estación de pruebas. Estos VI reciben las señales a través de las entradas analógicas de los dispositivos DAQ. En consecuencia, las señales son también procesadas por el mismo dispositivo DAQ que las produce y embebidas dentro del código fuente desarrollado en C++ a través de una DLL.

Normalmente, todos los VI de adquisición de señal analógica contienen la misma estructura. Un dispositivo DAQ, usado como el adquirente de la señal, y un indicador, comúnmente un valor, gráfico o medidor visual. El ejemplo ilustra la posible adquisición de la señal de la fuerza generada por el sensor de fuerza incorporado en la TTR.

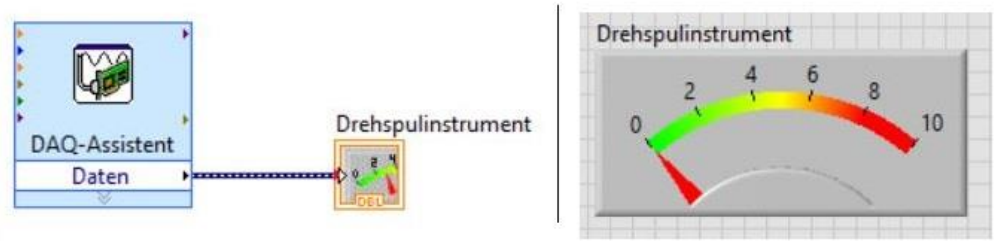


Figura 56. VI adquirente de señal analógica.

5.3.3. VI generador de señal digital

Los VI generadores de señal digital fueron programados para enviar señales HIGH o LOW requeridas para establecer ciertos parámetros de los actuadores de la TTR. Estos VI generan señales a través de los puertos de salida digitales de los dispositivos DAQ, los cuales incluyen un indicador LED. En consecuencia, el indicador LED muestra visualmente el estado de la señal digital. Las señales digitales producidas también se encuentran embebidas en el código fuente a través de una DLL. Esto hizo posible el análisis de las señales necesarias para habilitar o deshabilitar los actuadores en cuanto a las funciones del programa de control.

Comúnmente, los VI generadores de señal digital tienen la misma estructura. Un dispositivo DAQ, utilizado como el generador de señal y un control Booleano, usualmente un botón de dos estados. El ejemplo ilustra un generador de señal digital.

Figura 57. VI generador de señal digital.

5.4. BIBLIOTECAS DE ENLACE DINÁMICO

Una Biblioteca de Enlace Dinámico o DLL (sigla en inglés de Dynamic-Link Library) es la implementación, de Microsoft, del concepto de una librería compartida dentro del sistema operativo Windows. (Microsoft, 2016) Una biblioteca compartida es un archivo que pretende compartir otros archivos de caracter ejecutable. Así, ciertos módulos utilizados por un programa son cargados en la memoria a partir de objetos individuales compartidos durante el tiempo de ejecución, en lugar de ser copiados.

LabVIEW es capaz de crear tales bibliotecas compartidas con el fin de construir funciones que permitan la transferencia de información entre plataformas basadas en C y LabVIEW. En definitiva, las DLL permiten tanto al programado como al usuario manipular las señales generadas o adquiridas por un dispositivo DAQ, inherente a LabVIEW, a través de una Interfaz Gráfica de Usuario programada en C++.

El ejemplo que se muestra a continuación detalla la definición del prototipo de una DLL para la adquisición de una señal. La función Prototype indica *Force(void)*, lo que implica que la función tendrá como retorno un valor tipo *double* perteneciente a la lectura del sensor de fuerza de la TTR. Por lo tanto, si dicha función fuera a ser codificada en C++, retornaría el valor de la fuerza en el entorno C++.

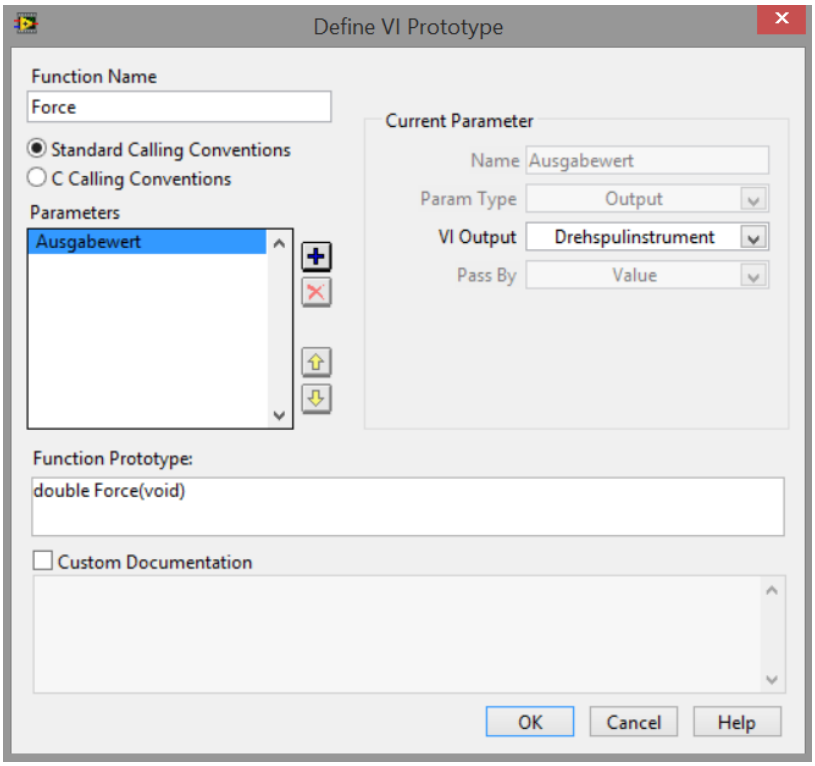


Figura 58. Definición de la función prototipo para el VI de adquisición de fuerza.

Puesto que el programa de control está escrito en C++, todos los VI fueron convertidos en DLL, con el fin de que todas señales puedan ser monitoreadas por una plataforma única, siendo este el objetivo principal de implementa Bibliotecas de Enlace Dinámico.

CAPÍTULO 6
IMPLEMENTACIÓN DE LAS CLASES DE LA GUI

El presente capítulo describe la implementación final de las clases, programadas en Visual C++, que conforman la Interfaz Gráfica de Usuario de la estación de pruebas o TTR. Así, el programa está conformado por tres clases:

- La clase MyForm, la cual se encarga de gestionar todas las funciones de la ventana principal o de inicio del programa.
- La clase Vertical_stiffness, la cual ejecuta la prueba de rigidez vertical tanto estática como dinámica.
- La clase Uniformity_test, responsable de operar la TTR en cuanto a la prueba de uniformidad completa. Dicha clase quedó programada en base, mas la implementación de sus funciones no fue realizada en este proyecto.

Adicionalmente, este capítulo detalla la elaboración de un manual de usuario que explica las principales características de funcionalidad, así como las medidas de seguridad de la estación, en cuanto a hardware y software. Dicho manual se encuentra en la sección de anexos del presente trabajo.

6.1. CLASE MYFORM

Como se explica en la sección introductoria, la clase MyForm ejecuta la ventana principal o home del programa de control. La función principal de esta ventana es dar la bienvenida al usuario, operar manualmente los principales actuadores de la estación y acceder a cada uno de los menús de pruebas. En adición, la ventana home permite también al usuario ingresar ciertas propiedades del neumático que va a ser sujeto a prueba.

El diseño final de la clase MyForm correspondiente a la ventana principal se detalla a continuación:

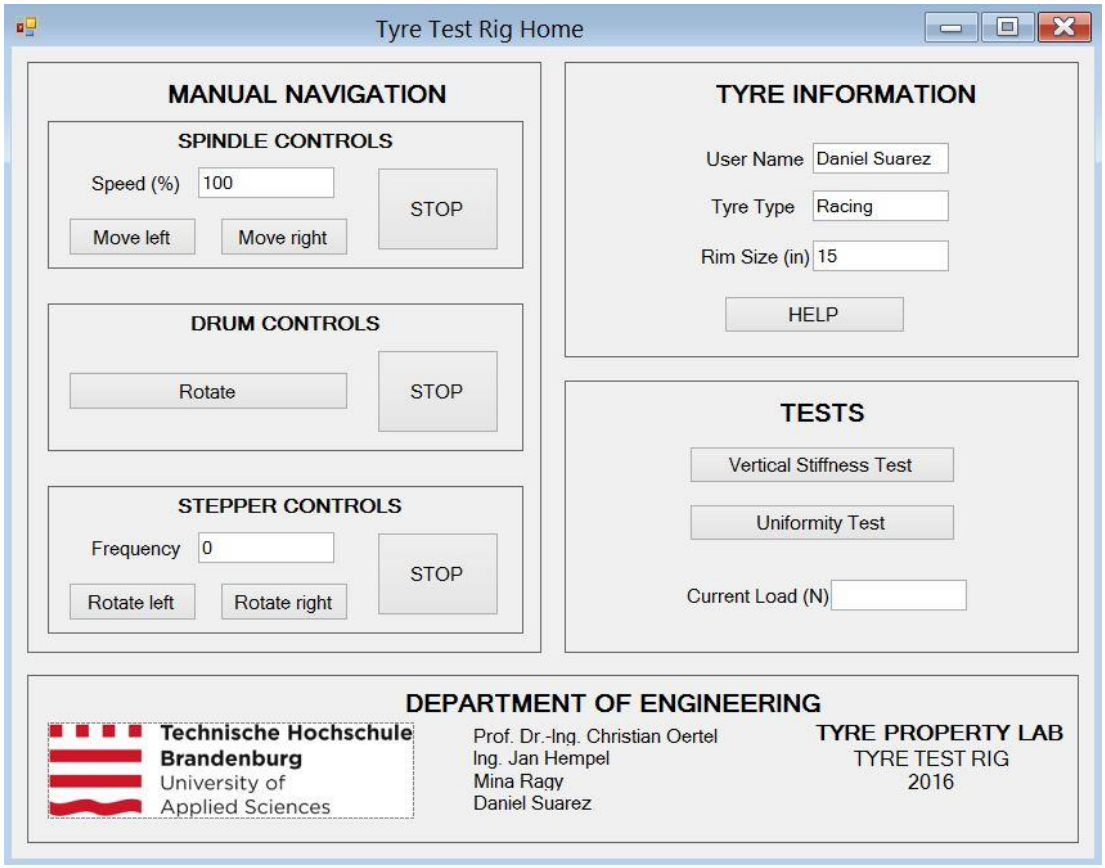


Figura 59. Ventana de inicio final de la TTR.

Como se puede observar, no existe una gran diferencia con el diseño preliminar de la misma (Ver Figura 45). Los elementos que se han adicionado se citan a continuación:

- **Panel de información del neumático:** Utilizado por usuario para ingresar datos iniciales del neumático antes de la prueba. Dicho panel ha sido implementado para futuras aplicaciones.
- **Botón HELP:** La ejecución de este botón despliega en una ventana adicional el manual de usuario edificado de la estación de pruebas.
- **Cuadro de texto Current Load:** Este cuadro de texto se encarga de desplegar el valor inicial de carga en Newton que se encuentre sobre el neumático. Es una medida de seguridad que indica al usuario, cuando inicia el programa, si el neumático sujeto a prueba está sometido ya a alguna carga. En el caso de estarlo, el acceso a las ventanas de prueba está bloqueado.
- **Membrete de información.**

6.2. CLASE VERTICAL_STIFFNESS

La clase Vertical_stiffness fue diseñada para operar y monitorear las señales de entrada y salida requeridas para realizar una prueba de rigidez vertical. La ventana generada por la clase contiene tres pestañas, las cuales dividen la funcionalidad de la misma:

- La pestaña Manual permite al usuario manipular los actuadores con el fin de realizar una prueba de rigidez vertical manual.
- La pestaña Static, la cual ejecuta una prueba de rigidez vertical estática.
- La pestaña Dynamic, la cual se encarga de controlar el proceso de una prueba de rigidez vertical dinámica.

Para el diseño final, tanto la pestaña Static como la Dynamic comparten el mismo diseño y estructura. No obstante, ejecutan pruebas completamente diferentes. Además, la prueba de rigidez vertical dinámica incorpora un actuador adicional, el tambor.

A continuación, se ilustra el diseño final de ambas pestañas:

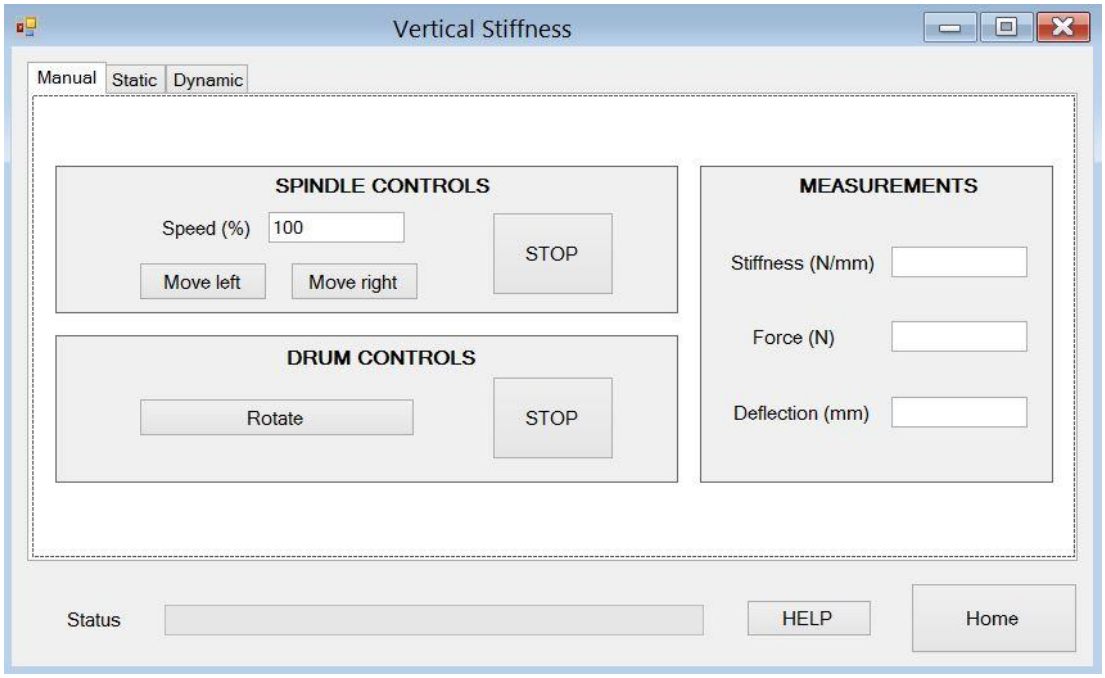


Figura 60. Pestaña final de la prueba de rigidez vertical manual.

La única diferencia que presenta esta pestaña con su antecesora (Ver Figura 48) es la incorporación del botón HELP, el cual despliega la ventana que contiene el manual de usuario.

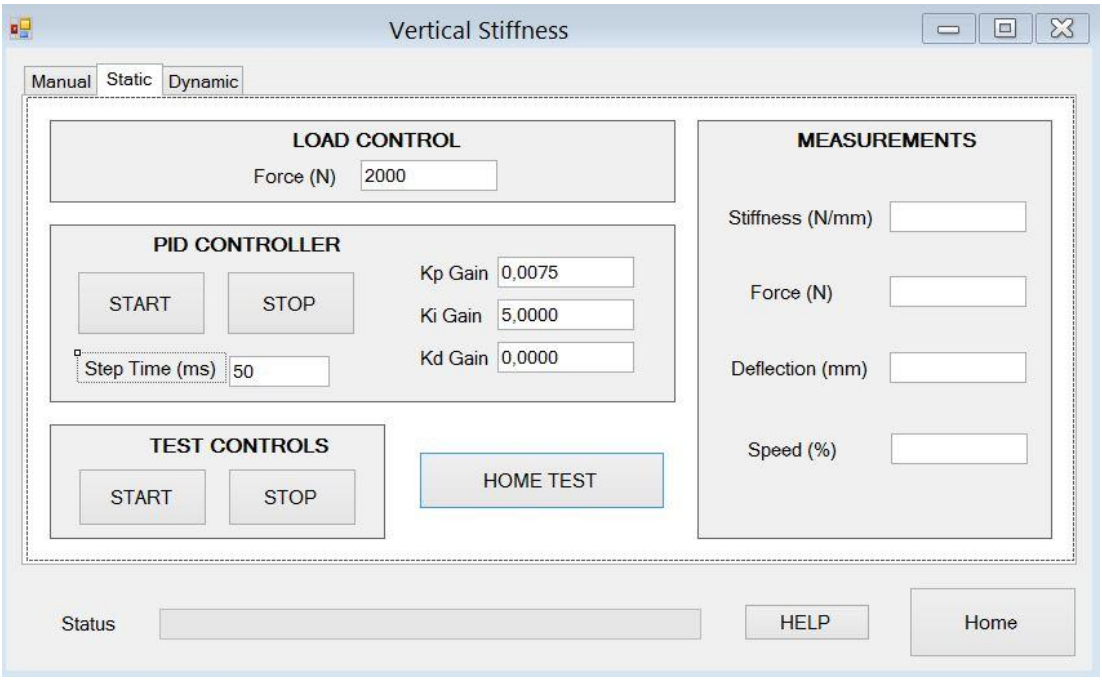


Figura 61. Pestaña final de la prueba de rigidez vertical estática.

En esta implementación final, se puede observar algunos cambios con respecto al diseño preliminar (Figura 46). Entre ellos destacan:

- **Panel PID CONTROLLER:** El presente panel fue incorporado como solución inicial de control para la aplicación de carga sobre el neumático. Su programación fue bastante abierta y su previa sintonización se realizó a través de prueba y error, pues se presentaron complicaciones el momento de modelar la planta. Como es posible observar, todos sus parámetros son modificables por el usuario, ingresando los valores deseados en los respectivos cuadros de texto. El botón START inicia la prueba y el botón STOP la detiene.

  En pruebas, el controlador funcionó parcialmente correcto, sometiendo al neumático a la carga ingresada por el usuario con muy poco error. Sin embargo, después de la incorporación de la medida de deflexión, el controlador presentó problemas en cuanto a la respuesta de retroalimentación, debido al algoritmo de comunicación entre el sensor, LabVIEW y el código fuente de C++. La respuesta

era muy lenta y limitaba la acción del controlador. Dicho problema se presentó al intentar modelar la planta de la TTR.

Pese a lo mencionado, el panel no se retiró con el fin de buscar una solución en futuras aplicaciones e investigación.

- **Panel TEST CONTROLS:** Debido al inconveniente mencionado en el ítem anterior, se diseñó otro algoritmo de control para el panel TEST CONTROLS. Dicho control se basa en los controladores todo/nada y consiste en aplicar la carga sobre el neumático gradualmente.

  La ejecución del botón START inicia una secuencia de hilos que trabaja de la siguiente manera:

  - El neumático se acerca con un 80% de la velocidad máxima del huso hacia cualquiera de las dos superficies. El momento de entrar en contacto se dispara el siguiente hilo.
  - El neumático se somete a carga con un 40% de la velocidad máxima del huso hasta alcanzar una diferencia entre el valor iniciado de la carga y el valor actual de carga de 500N. Al cumplirse dicha condición, se dispara el siguiente hilo.
  - El neumático continúa sometiéndose a carga con un 20% de la velocidad máxima del huso cada 500 ms, hasta alcanzar una diferencia entre el valor iniciado de la carga y el valor actual de carga de 100N. Al satisfacerse la condición, se dispara el último hilo.
  - El neumático se somete a carga con un 10% de la velocidad máxima del uso cada 250 ms hasta llegar al valor deseado por el usuario.

  Cabe señalar que debido a las perturbaciones así como el manejo de la señal de fuerza (Ver Capítulo 7), existía error de estado estable, considerado despreciable pues las cargas a las cuales se sometían los neumáticos de prueba superaban los 3000N.

  El Botón STOP detiene la prueba en su totalidad.

- **Botón HOME TEST:** Al finalizar con éxito sea la prueba con el controlador PID o sea la prueba con el control todo/nada, la ejecución de este botón libera al neumático de cualquier carga y lo ubica a una posición segura de cualquiera de las superficies de contacto.

- **Cuadro de texto Speed:** Muestra al usuario la velocidad actual del huso durante todo el desarrollo de la prueba.
- **Botón HELP:** La ejecución de este botón despliega una ventana adicional que contiene el manual de usuario edificado de la TTR.

6.3. CLASE UNIFORMITY_TEST

La clase Uniformity_test tiene la responsabilidad de ejecutar y controlar el desarrollo de la prueba de uniformidad completa. La prueba de uniformidad completa se divide en dos subrutinas. Por esta razón, la ventana correspondiente a la clase Uniformity_test consta de tres pestañas también:

- La pestaña Manual, que permite al usuario manipular los actuadores relacionados con la prueba de uniformidad completa manualmente.
- La pestaña Outer shape, que realiza la prueba de uniformidad completa con el neumático estático.
- La pestaña Dynamic non-uniformity, la cual realiza la prueba de uniformidad completa incluyendo la rotación gradual del neumático, a través del motor de pasos.

La ventana fue implementada como base para futuras aplicaciones, puesto que su dentro de su realización, se tuvo un problema con el controlador del motor de pasos, así como el del robot.

La pestaña Outer shape y la pestaña Dynamic non-uniformity comparten la misma estructura, como se presenta a continuación:

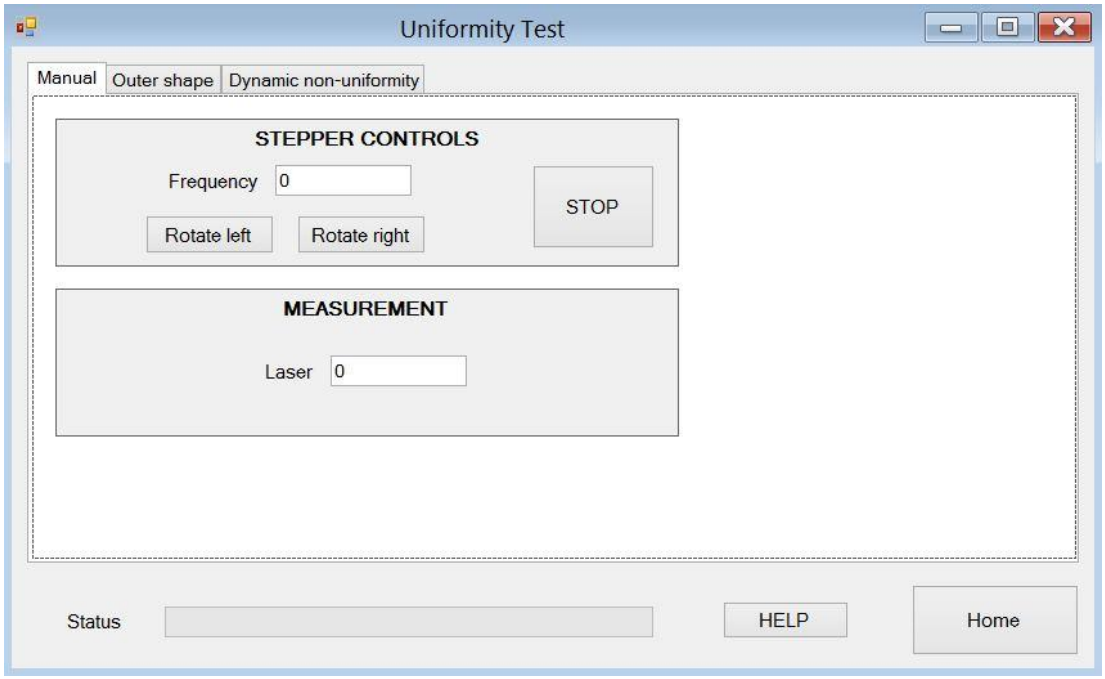


Figura 62. Pestaña final de la prueba de uniformidad completa manual.

La pestaña final de la prueba de uniformidad completa manual contiene dos paneles. El panel STEPPER CONTROLS cuya funcionalidad es muy similar a la del control del motor de pasos de la ventana principal y el panel MEASUREMENT que tiene como objetivo mostrar el valor de medición del láser cada cierto intervalo de tiempo.

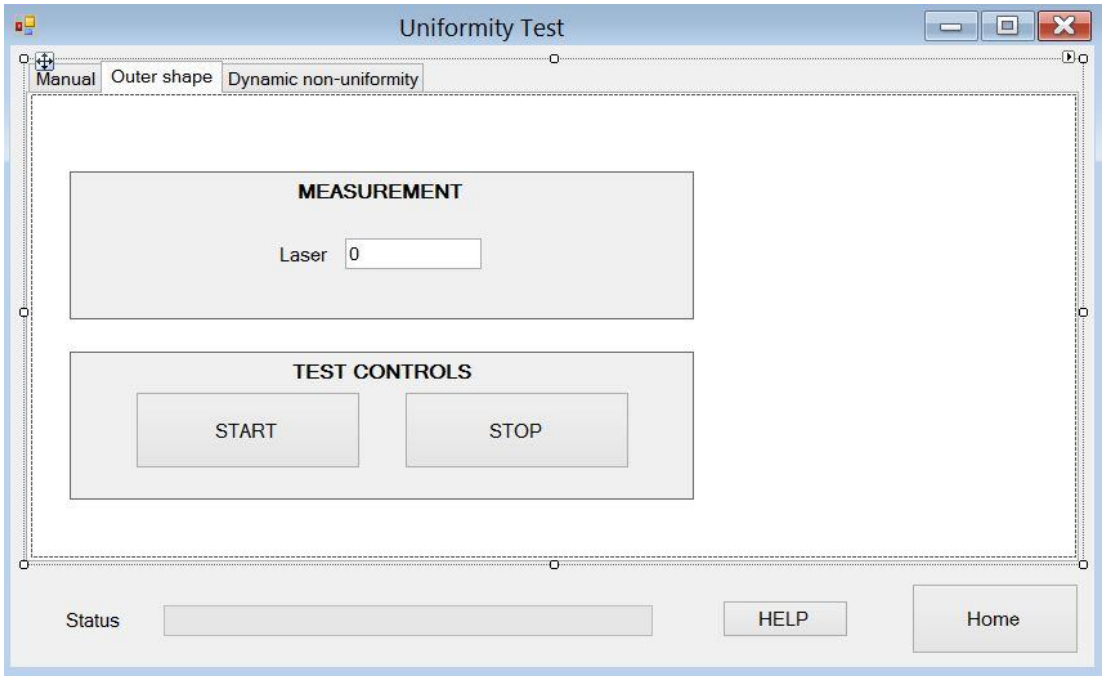


Figura 63. Pestaña final de la prueba de uniformidad completa estática.

La pestaña de la prueba de uniformidad completa estática está compuesta igualmente por dos paneles. El Panel MEASUREMENT que muestra el valor de medición del láser cada cierto tiempo y el panel TEST CONTROLS, el cual se encarga de iniciar o detener la ejecución del algoritmo de prueba, junto con el robot. Dicho algoritmo no fue desarrollado y se propone como una futura aplicación del sistema.

6.4. MANUAL DE USUARIO

Para la presente estación de trabajo, se redactó un manual de usuario a través del programa HelpMaker, una herramienta de autorización de ayuda para generar archivos tipo WinHelp, HTML-Helo, HTML-Website y RTF.

El manual describe la funcionalidad de cada ventana del programa de control durante el proceso de cada prueba así como la manera correcta de operar cada elemento de la TTR a través del software. La siguiente figura ilustra el diseño final del manual.

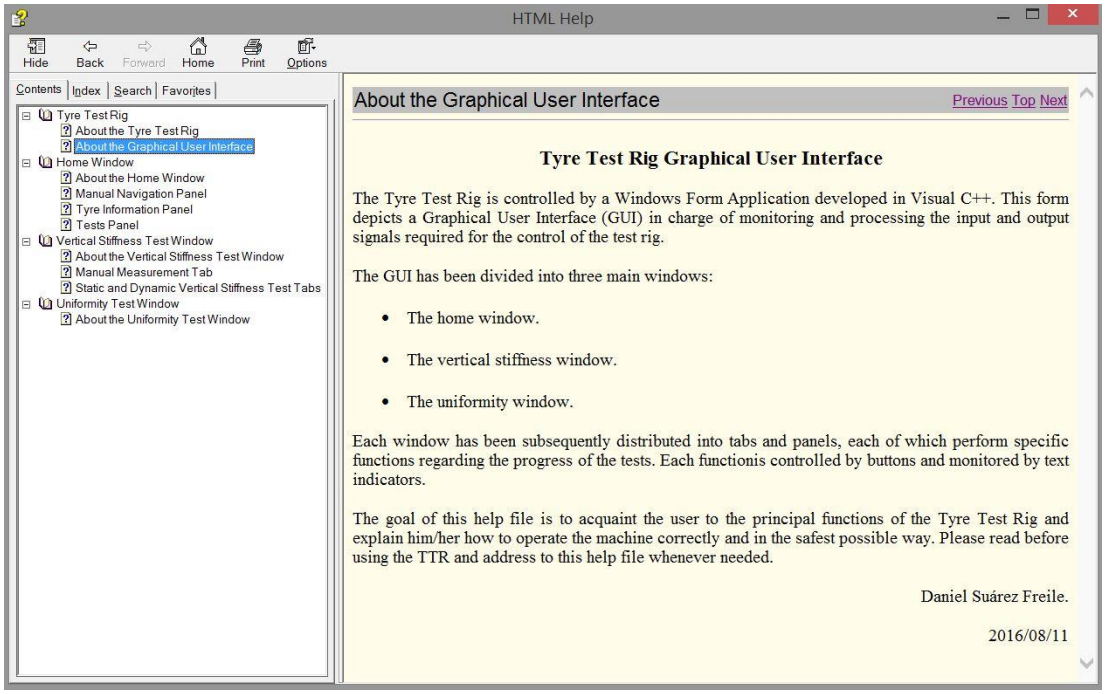


Figura 64. Manual de usuario de la GUI de la estación de pruebas o TTR.

El presente manual se encuentra como anexo, en la parte final de este documento.

CAPÍTULO 7
IMPLEMENTACIÓN DE LOS INSTRUMENTOS VIRTUALES

En el capítulo 5, se presentó brevemente los Instrumentos Virtuales (VI) de medición, inherentes a LabVIEW, desarrollados para establecer la comunicación entre la estación de pruebas (TTR) y la Interfaz Gráfica de Usuario (GUI). El presente capítulo detalla la implementación final de los VI de medición en el programa de control principal. Éstos son:

- El VI generador de señal analógica para la velocidad angular del huso.
- El VI generador de señal digital para habilitar el huso y establecer su sentido de giro.
- El VI de adquisición de señal analógica para tanto la medición de la fuerza como la deflexión del neumático durante las pruebas de rigidez vertical.

7.1. VI GENERADOR DE SEÑAL ANALÓGICA

Como se menciona en el capítulo 5, el VI generador de señal analógica fue específicamente diseñado para trabajar con datos sintéticos. Sin embargo, una vez habilitado el motor acoplado al huso, el motor sólo necesitaba una señal analógica de entrada de 0 a 10V. A través del dispositivo DAQ y LabVIEW, es posible suministrar dicha cantidad de voltaje, por lo que el VI diseñado en un inicio para pruebas se convirtió en el VI implementado.

7.2. VI GENERADOR DE SEÑAL DIGITAL

El VI generador de señal digital fue programado para enviar señales LOW o HIGH, requeridas para configurar ciertos parámetros de los actuadores de la TTR. Estas señales están embebidas en el código del programa de control y se encargan de habilitar el motor conectado al uso y establecer su sentido de giro. Los VI implementados contienen la misma estructura que los que se utilizó para las pruebas, y éstos se describen en el capítulo 5.

7.3. VI DE ADQUISICIÓN DE SEÑAL ANALÓGICA

7.3.1. Adquisición de la señal de la fuerza

La estación de pruebas utiliza un sensor de seis componentes para medir la carga aplicada en todo momento al neumático montado en la TTR. El sensor provee seis señales analógicas de tensión. Para obtener el valor de la fuerza, es necesario computar dichos valores con una matriz de transformación, suministrada por el fabricante. La medición de dichos voltajes fuera realizada a través de un dispositivo DAQ y un VI de adquisición de señal analógica diseñada para leer las seis señales y enviarlas directamente al código fuente a través de una DLL, como puntero a valor.

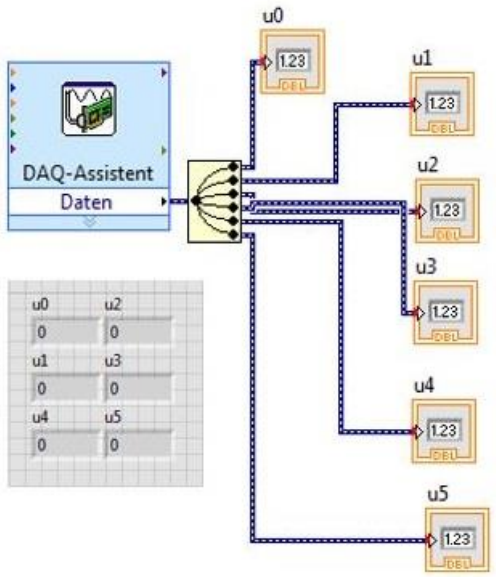


Figura 65. VI de adquisición de la fuerza.

El cómputo de la información fue realizada en el código fuente de la siguiente manera:

```
#include "stdafx.h"
#include "Vertical_stiffness.h"

namespace Test_rig_GUI
{
    //Force computation function
    double Vertical_stiffness::Force(double calib)
    {
        double u1,u2,u3,u4,u5,u6; //Voltages obtained from the force sensor through labview
        double fx;//, fy, fz, mx, my, mz;

        RimForceVoltage(&u1,&u2,&u3,&u4,&u5,&u6);

        fx=4557.6*u1-4589.6*u2+26*u3+4583.5*u4-4622.2*u5+13.1*u6;
        fx=fx-calib;//Calibration

        /*
        fy=-2738.7*u1-2641.8*u2+5202.9*u3-2791.5*u4-2682.1*u5+5262.9*u6;

        fz=6038.4*u1+6090.2*u2+6116.9*u3+6057.7*u4+6274.6*u5+6123.4*u6;

        mx=-547.4*u1-547*u2+392.7*u3+165.7*u4+160.6*u5+381*u6;

        my=-129.3*u1+123.5*u2-408.1*u3-529.2*u4+535*u5+416.7*u6;

        mz=320.1*u1-318.8*u2+320.3*u3-323*u4+317.5*u5-316.7*u6;
        */

        return fx; //Needed for the vertical stiffness tests
    }
```

Figura 66. Código de cómputo de los voltajes de la fuerza.

La función presentada en la figura anterior provee la carga actual que se encuentra aplicada sobre el neumático sujeto a prueba. Es posible observar que ésta requiere de un parámetro de calibración. La siguiente figura y explicación definen a dicho parámetro.

```
//Force computation function
double Vertical_stiffness::ForceX_Calibration()
{
    double u1,u2,u3,u4,u5,u6; //Voltages obtained from the force sensor through labview
    double fxcalib;

    RimForceVoltage(&u1,&u2,&u3,&u4,&u5,&u6);

    fxcalib=4557.6*u1-4589.6*u2+26*u3+4583.5*u4-4622.2*u5+13.1*u6;

    return fxcalib;
}
```

Figura 67. Código de calibración de la fuerza.

La función ForceX_Calibration() se encarga de calcular el valor de calibración necesario para que la medición de la fuerza sea precisa. Para conseguirlo, la función está embebida en un lazo de repetición *for*, el cual obtiene el promedio de 10 valores de medición, mientras el neumático no está sujeto a carga. Luego, dicho valor de

calibración es insertado en la función Force() y así dicha función retorna el valor correcto de la fuerza aplicada al neumático de prueba.

La calibración se realiza solamente una vez, cuando la clase Vertical_stiffness es ejecutada.

7.3.2. Adquisición de la señal analógica del ángulo

Para poder medir el ángulo de rotación del huso mientras éste realiza una prueba de rigidez vertical, se acopló un sensor angular incremental. Dicho sensor requiere de un VI especial de procesamiento de señal y su salida puede ser establecida en grados o en radianes, como ilustran las figuras:

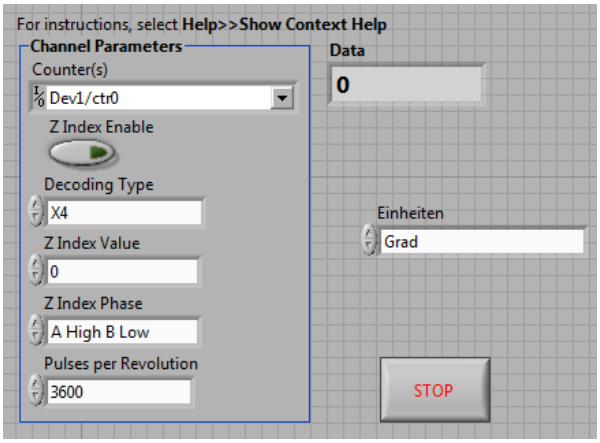


Figura 68. VI correspondiente al sensor angular incremental.

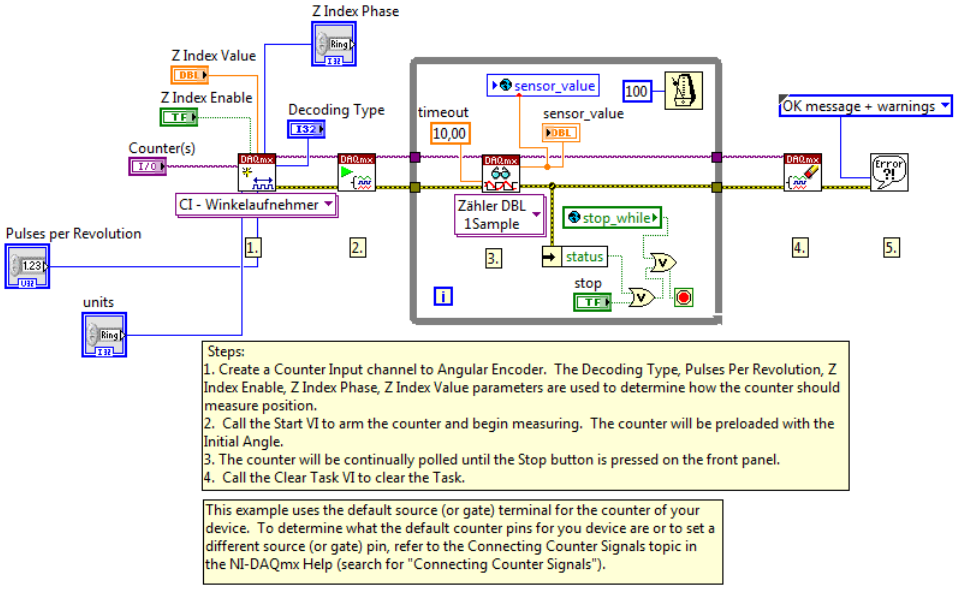


Figura 69. Diagrama de Bloques desarrollado para el sensor angular incremental.

Al intentar embeber dicho VI en una DLL, y más tarde al código fuente, se suscitó un inconveniente. Para medir de manera incremental, el sensor implementado requiere el lazo de repetición *while* descrito en la Figura 69. Para resolver el problema, se diseñó otro VI encargado de medir los valores proporcionados por el VI del sensor a través de variables globales.

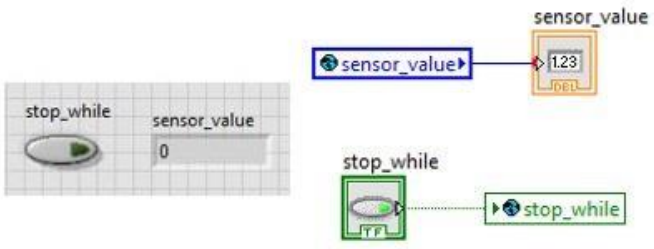


Figura 70. VI utilizado para crear la DLL de medición angular.

Cabe señalar que este VI requiere la señal booleana *stop_while*. Esto implica que para mostrar un valor en el código fuente, el VI de medición angular debe detenerse. Para englobar todos estos aspectos, se diseñó un hilo adicional dentro del código del programa de control.

```
//Shows the deflection
//-------------------------------------------------------------------------------------
    private: System::Void TimerTickEvent(System::Object^ , System::EventArgs^ e)
//-------------------------------------------------------------------------------------
             {
                 double sensor_value;

                 if(Force(xforce_calibration)>25 || Force(xforce_calibration)<-25)
                 {
                     Deflection_angle(&sensor_value,0);
                     deflection = sensor_value/36;

                     if(deflection>=0)
                         this->textBoxDeflection->Text = deflection.ToString("F4");
                     else
                         this->textBoxDeflection->Text = (-deflection).ToString("F4");
                     this->textBoxDeflection->Update();

                     //k=1;
                 }
                 else
                 {
                     this->textBoxDeflection->Text ="0,0000";
                     //timer->Stop();
                     //j=0;
                 }
             }
```

Figura 71. Hilo implementado para mostrar la deflexión durante las pruebas.

Es factible observar que el valor obtenido del sensor se encuentra dividido para 36. Esto se debe a que 1 revolución equivale a 10 mm en desplazamiento lineal. El valor del sensor angular se transmite como puntero a valor. Cabe señalar también que dentro de la función Deflection_angle(), un 0 está siendo enviado. Dicho valor es la señal booleana encargada de detener el lazo de repetición *while* dentro del VI.

CAPÍTULO 8
CONCLUSIONES Y RECOMENDACIONES

El presente capítulo analiza brevemente los principales resultados del proyecto. Consiguientemente, presenta las conclusiones respectivas y futuras recomendaciones que permitan ampliar de manera significativa el proyecto.

8.1. CONCLUSIONES

El trabajo fue implementado exitosamente en dos estaciones de trabajo y comprende dos secciones fundamentales. La primera sección detalla la implementación de un sistema de control y monitoreo de una de las estaciones que conforman el banco de pruebas estáticas y dinámicas para neumáticos; y la segunda describe la elaboración detallada de una Interfaz Gráfica de Usuario y un sistema de instrumentación mecatrónica que tienen como objetivo supervisar y controlar la estación de pruebas o TTR.

El propósito de la estación 3DTisch es escanear la sección de un corte transversal de un neumático, a través de una máquina CNC y un sensor láser de distancia. Esto se consiguió al implementar un algoritmo a través del lenguaje de programación Visual C++ y una GUI capaz de controlar todo el proceso. La comunicación entre el software y el hardware de la estación se realizó a través de dispositivos DAQ inherentes a LabVIEW.

La meta de la segunda parte del trabajo fue elaborar un sistema de control que utilice una GUI y un sistema de instrumentación, y que el mismo sea capaz de monitorear y controlar los actuadores y sensores de la estación de pruebas con el fin de testear neumáticos y extraer sus principales propiedades. Esto se consiguió, especialmente para la prueba de rigidez vertical tanto estática como dinámica ya que los actuadores y sensores que intervenían no presentaron complicaciones mayores, a diferencia de los necesarios para las pruebas de uniformidad.

Adicionalmente, para la segunda estación se diseñó un controlador PID para el control de fuerza durante las pruebas de rigidez vertical. Durante la obtención de la planta y definición de las constantes se determinó que la respuesta del controlador fue excesivamente lenta debido a que la aplicación no era de tiempo real y requería del

procesamiento de mucha información, especialmente la manipulación de datos desde Visual C++ hacia LabVIEW, descartando así la implementación del mismo. Una de las principales causas fue la lectura del sensor angular incremental, debido al manejo de su señal dentro de los Instrumentos Virtuales de LabVIEW.

Pese a lo mencionado, el algoritmo de control y su panel no fueron retirados con el fin de buscar una solución en futuras aplicaciones e investigación.

Debido al inconveniente suscitado para el controlador PID, se diseñó otro algoritmo de control para realizar las pruebas de rigidez vertical. Dicho control se basa en los controladores todo/nada y consiste en aplicar la carga sobre el neumático gradualmente. Básicamente, el controlador se describe en la siguiente sección:

- El neumático se acerca con un 80% de la velocidad máxima del huso hacia cualquiera de las dos superficies. El momento de entrar en contacto se dispara el siguiente hilo.
- El neumático se somete a carga con un 40% de la velocidad máxima del huso hasta alcanzar una diferencia entre el valor iniciado de la carga y el valor actual de carga de 500N. Al cumplirse dicha condición, se dispara el siguiente hilo.
- El neumático continúa sometiéndose a carga con un 20% de la velocidad máxima del huso cada 500 ms, hasta alcanzar una diferencia entre el valor iniciado de la carga y el valor actual de carga de 100N. Al satisfacerse la condición, se dispara el último hilo.
- El neumático se somete a carga con un 10% de la velocidad máxima del uso cada 250 ms hasta llegar al valor deseado por el usuario.

Cabe señalar que debido a las perturbaciones así como el manejo de la señal de fuerza (Ver Capítulo 7), existía error de estado estable, considerado despreciable pues las cargas a las cuales se sometían los neumáticos de prueba superaban los 3000N.

En la Universidad de Ciencias Aplicadas de Brandemburgo, se desarrolló un software capaz de modelar neumáticos de automóviles y realizar Análisis de Elementos Finitos sobre los mismos. Dicho software, conocido como RMOD-K 7, busca que los parámetros de un neumático modelado sean exactamente los mismos que los de un neumático real. Debido a esto, el proyecto pretende alcanzar el mencionado propósito a través de la siguiente estrategia:

1. Las primeras pruebas a realizarse sobre el neumático serían las de uniformidad completa, tanto estática como dinámica. Dichas pruebas no pudieron ser

concluidas en su totalidad debido a los contratiempos encontrados. No obstante, se diseñó la clase correspondiente a estas pruebas y se la implementó en el programa de control principal, dejándola un 60% funcional.

2. El siguiente paso consiste en extraer las propiedades correspondientes a las pruebas de rigidez vertical, y con un análisis breve de las propiedades de los materiales que conforman las capas del neumático, el software puede simular los valores, y compararlos con los de la prueba. El presente ítem fue concluido en un 100%.
3. El último paso del proceso sería realizar un pequeño corte de sección transversal con el fin de obtener el trazado de la misma, y realizar pruebas posteriores dentro del software, también culminado un 100%.

Todo este proceso se encuentra embebido en un lazo repetitivo que lo que busca es hallar los mejores parámetros que describan al neumático sujeto a prueba y que a partir de este, se pueda generar incluso neumáticos aún no existentes en el mercado.

8.2. RECOMENDACIONES

8.2.1. Aplicaciones futuras referentes a la ventana de inicio

Información del neumático

Uno de los paneles de la ventana de inicio tiene el nombre de TYRE INFORMATION y lo que pretende es recibir la información ingresada por el usuario sobre el neumático que se va a sujetar a pruebas. Se recomienda encontrar la manera de generar un archivo con dichas características, una vez que hayan sido ingresadas por el usuario. Así, es posible llevar un registro de todos los neumáticos que alguna vez se hayan sujeto a pruebas.

8.2.2. Aplicaciones futuras referentes a las pruebas de rigidez vertical

Control del huso

Durante la realización de la segunda parte del proyecto, se diseñó parcialmente un control PID digital, dentro del código fuente, con el fin de controlar la velocidad del motor acoplado al huso que desencadena en el desplazamiento lineal encargado de aplicar la carga al neumático. Por lo tanto, el controlador busca monitorear la carga aplicada al sistema. El problema del controlador es su tiempo de respuesta, debido a que mientras el proyecto iba creciendo, el costo computacional también. Así, dicho controlador fue descartado como algoritmo principal durante las pruebas de rigidez vertical y en lugar, se incorporó un controlador de aplicación de carga gradual. El programa, en su etapa final consta de más de doce hilos, encargados de supervisar constantemente todo el proceso. Se recomienda modificar el sensor angular incremental, ya que su procesamiento de señal es muy lento.

Trazado de diagramas

Durante la elaboración de las pruebas de rigidez vertical, se mencionó la incorporación de un sistema de trazado automático capaz de trazar el comportamiento de la deflexión y la fuerza durante una prueba de rigidez vertical. Dichos programas no fueron incluidos en la presentación final debido a un contratiempo con las versiones de OpenGL, la plataforma utilizada. Se recomienda buscar la manera de implementar el sistema ya que facilitaría mucho el análisis de los resultados de una prueba de rigides vertical.

8.3. Aplicaciones referentes a la incorporación de nuevas pruebas

Pruebas de uniformidad completa

Las pruebas de uniformidad completa tanto estática como dinámica pretenden obtener el trazado de la sección de un neumático sujeto a prueba. Se diferencia del proceso de la estación 3DTisch debido a que en la estación de pruebas, el neumático está correctamente inflado y montado sobre su rin correspondiente. Se recomienda implementar tanto el programa del robot, una vez solucionado el contratiempo con su controlador, para poder realizar dicha prueba; como el VI encargado del motor de pasos que contribuye a la rotación del neumático.

Prueba de zona de contacto

La prueba de zona de contacto busca obtener la huella de distribución de la presión o *contact patch* de un neumático sujeto a una determinada carga. La estación de pruebas es capaz de realizarla, a través del uso de la matriz de sensores. Se recomienda incluir dicha prueba al programa de control principal, y monitorearla desde dicha unidad de control.

BIBLIOGRAFÍA

ACSES. (2015). *C++*. Obtenido de ACSES: http://www.wceacses.org/cpp.html

Bahuguna, A. (2008). *Reference Architecture for Enterprise Batch Processing of Information*. Obtenido de IEEE Xplore Digital Library: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4746803&tag=1

Bogotobogo. (2016). *.NET Framework 2016*. Obtenido de Open Source ...: http://www.bogotobogo.com/CSharp/.netframework.php

Cplusplus. (2009). *Pointers*. Obtenido de Cplusplus: http://www.cplusplus.com/doc/tutorial/pointers/

Digi-Key. (29 de 02 de 2012). *Industrial Touch Screens Come of Age*. Obtenido de World's Largest Selection of Electronic Components Available for Immediate Shipment!: http://www.digikey.com/en/articles/techzone/2012/feb/industrial-touch-screens-come-of-age

Disch. (2009). *Headers and Includes: Why and How*. Obtenido de Welcome to Cplusplus: http://www.cplusplus.com/forum/articles/10627/

Fridman, L., Poznyak, A., & Bejarano, F. J. (2014). *Robust Output LQ Optimal Control via Integral Sliding Modes*. New York: Springer Science+Business Media New York.

Gene, F., Powell, D., & Emami-Naeini, A. (2002). *Feedback Control of Dynamic Systems*. New Jersey: Prentice Hall.

Haas Technologies Ltd. (2014). *Controls*. Obtenido de HmGroup: http://hmg-bd.com/haas-technology-controls/

Microsoft. (2016). *Constructors (C++)*. Obtenido de Developer Network: https://msdn.microsoft.com/en-us/library/s16xw1a8.aspx

Microsoft. (2016). *Destructors (C++)*. Obtenido de Developer Network: https://msdn.microsoft.com/en-us/library/6t4fe76c.aspx

Microsoft. (2016). *Dialog Boxes*. Obtenido de Developer Network: https://msdn.microsoft.com/en-us/library/windows/desktop/dn742499%28v=vs.85%29.aspx

Microsoft. (2016). *Graphic Elements*. Obtenido de Developer Network: https://msdn.microsoft.com/en-us/library/windows/desktop/dn742484%28v=vs.85%29.aspx

Microsoft. (2016). *Visual Studio*. Obtenido de Visual Studio: https://www.visualstudio.com/

National Instruments. (Enero de 1998). *User Manual*. Obtenido de http://www.ni.com: http://www.ni.com/pdf/manuals/320999b.pdf

National Instruments. (2016). *LabVIEW Environment Basics*. Obtenido de National Instruments: https://www.ni.com/getting-started/labview-basics/environment

National Instruments. (2016). *NI 9215 Datasheet*. Obtenido de National Instruments: http://www.ni.com/pdf/manuals/373779a_02.pdf

National Instruments. (2016). *What is LabVIEW?* Obtenido de National Instruments: http://www.ni.com/newsletter/51141/en/

Ogata, K. (2010). *Ingeniería de Control Moderna.* México D.F.: Prentice-Hall.

Shelly, G. B. (2012). *Discovering Computers: Fundamentals.* Cram101.

Tire Rack. (2016). *Tire Specs Explained: Section Width*. Obtenido de Tire Rack: http://www.tirerack.com/tires/tiretech/techpage.jsp?techid=200

Wikibooks. (26 de 03 de 2012). *Control Systems/Digital and Analog*. Obtenido de WikiBooks: https://en.wikibooks.org/w/index.php?title=Control_Systems/Digital_and_Analog&oldid=2292718