



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

**Actualización del control electrónico del
espectrógrafo integral de
campo "PUMA"**

TESIS

Que para obtener el título de
Ingeniero Mecatrónico

P R E S E N T A

MARSILLI ESTRADA RUBÉN ZIMRI

DIRECTOR DE TESIS

Dra. Margarita Eugenia del Socorro Rosado Solís



Ciudad Universitaria, Cd. Mx., 2016



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

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

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

ACTUALIZACIÓN DEL CONTROL ELECTRÓNICO DEL ESPECTRÓGRAFO INTEGRAL DE CAMPO "PUMA"

RUBEN ZIMRI MARSILLI ESTRADA



instituto de astronomía

UNAM

Actualización del control electrónico del espectrógrafo integral de campo PUMA

TESIS

EN CUMPLIMIENTO DE LOS REQUISITOS PARA OBTENER EL TÍTULO
DE
INGENIERO EN MECATRÓNICA

por

RUBEN ZIMRI MARSILLI ESTRADA

Directora de tesis:

Dra, Margarita Eugenia del Socorro Rosado Solís



**Universidad Nacional Autónoma de México
Facultad de Ingeniería**

Ciudad Universitaria

Diciembre 2016

México



Departamento de Ingeniería Mecatrónica
División de Ingeniería Mecánica e Industrial
Facultad de Ingeniería
Universidad Nacional Autónoma de México
Ciudad Universitaria, México

Instrumentación Astronómica
Instituto de Astronomía
Universidad Nacional Autónoma de México
Ciudad Universitaria, México

Tesis de licenciatura (Ingeniero en Mecatrónica)
por Zimri Marsilli, 2016
email zimri.marsilli@aol.com
Número de matrícula: 410061695



Actualización del sistema electrónico de control del espectrógrafo integral de campo "PUMA"

Breve resumen:

El instrumento PUMA fue diseñado y construido por astrónomos e ingenieros del Instituto de Astronomía de la Universidad Nacional Autónoma de México, operando desde 1994 en el telescopio de 2.1 m en el Observatorio Astronómico Nacional, en San Pedro Mártir Baja California. El instrumento permite realizar observaciones comprendiendo imágenes directas e interferogramas de Fabry-Pérot de barrido en longitudes de onda de las bandas del óptico.

El control del instrumento se realizaba por medio de un microcontrolador marca Octagon, modelo 5080 μ PC, del cual sólo hay un repuesto, motivando la actualización de su control electrónico. El principal objetivo de este trabajo fue analizar, modelar, diseñar e implantar un sistema automatizado para interactuar armónicamente con los subsistemas del instrumento "PUMA", controlado por la monoplaca reducida marca Rabbit serie BL4S200.

Palabras clave: Espectrometría, PUMA, Interferometría Fabry-Pérot de barrido, Rabbit, BL4S200, OAN-SPM, IA-UNAM

"Basado en trabajos realizados en el Instituto de Astronomía de la Universidad Nacional Autónoma de México y en el Observatorio Astronómico Nacional en la Sierra de San Pedro Mártir (OAN-SPM), Baja California, México."

*Para aquellos que, como un cometa, salen de la nube de Oort,
empezando un viaje inesperado
y de un recorrido tan largo de muchos años luz.*

ÍNDICE GENERAL

| | |
|------------------------|------------|
| Prefacio | VII |
| Agradecimientos | IX |
| Resumen | XI |
| I Tesis | 1 |

INTRODUCCIÓN _____ **PÁGINA 3**

| | |
|---|----|
| 1.1 Interferometría óptica | 6 |
| 1.1.1 El inicio de una tecnología | 6 |
| 1.1.2 Teoría de la interferometría | 10 |
| 1.1.3 Espectrometría de Fabry-Pérot | 17 |
| 1.1.4 Técnicas de instrumentación | 19 |
| 1.1.4.1 El instrumento PUMA | 21 |

OBJETIVOS Y ALCANCES _____ **PÁGINA 23**

| | |
|--|----|
| 2.1 Problema y motivación | 24 |
| 2.2 Objetivo general | 24 |
| 2.3 Objetivos particulares | 25 |
| 2.4 Alcances | 25 |
| 2.5 Contribución de esta tesis | 26 |

DESCRIBIENDO AL SISTEMA _____ **PÁGINA 27**

| | |
|--|----|
| 3.1 Diseño optomecánico | 28 |
| 3.1.1 Interferómetro Fabry-Pérot | 30 |
| 3.1.2 Rueda de filtros | 32 |
| 3.1.3 Unidad de calibración | 33 |

| | | |
|-------|--|----|
| 3.1.4 | Brazo de prismas | 35 |
| 3.1.5 | Sistema de enfoque | 35 |
| 3.2 | Sistemas de control | 36 |
| 3.2.1 | Controlador CS100 | 36 |
| 3.2.2 | Sistema de control electrónico | 38 |
| 3.2.3 | Detector de imágenes | 39 |
| 3.3 | Interfaz de usuario | 40 |

MICROCONTROLADOR DE PLACA REDUCIDA BL4S200 _____ PÁGINA 43

| | | |
|---------|--|----|
| 4.1 | Controlador embebido | 44 |
| 4.2 | Características | 46 |
| 4.3 | Subsistemas | 48 |
| 4.3.1 | Entradas y salidas digitales | 50 |
| 4.3.2 | Interrupciones | 51 |
| 4.3.3 | Modulación de ancho de pulso | 51 |
| 4.3.3.1 | Entradas y salidas digitales de alta corriente | 52 |
| 4.3.4 | Entradas del convertidor analógico digital | 53 |
| 4.3.5 | Distribución de E/S para el control del instrumento PUMA | 53 |

EL ENTORNO DE DESARROLLO _____ PÁGINA 57

| | | |
|---------|---|----|
| 5.1 | El software Dynamic C | 58 |
| 5.2 | Protocolo TCP/IP | 60 |
| 5.3 | Modelo esencial | 62 |
| 5.3.1 | Modelo ambiental | 63 |
| 5.3.1.1 | Propósito del sistema | 63 |
| 5.3.1.2 | Diagrama de contexto | 63 |
| 5.3.1.3 | Lista de eventos | 64 |
| 5.3.2 | Modelo de comportamiento | 66 |
| 5.3.2.1 | Diagrama de flujo de datos | 67 |
| 5.3.2.2 | Diagrama de transición de estados | 70 |
| 5.3.2.3 | Diccionario de datos | 70 |
| 5.3.2.4 | Especificación de procesos | 73 |

SIMULACIÓN EN LABORATORIO **PÁGINA 95**

| | |
|--|-----|
| 6.1 Pruebas de desarrollo | 96 |
| 6.1.1 Pruebas de factibilidad y desarrollo | 97 |
| 6.1.2 Pruebas de red | 98 |
| 6.1.3 Salidas de alta corriente | 102 |

MÓDULO DE CONTROL ELECTRÓNICO **PÁGINA 115**

| | |
|--|-----|
| 7.1 Diseño del circuito impreso | 116 |
| 7.2 Diagrama de bloques a nivel de placa | 117 |
| 7.3 Componentes del hardware | 117 |
| 7.3.1 Etapa de optoacoplamiento | 119 |
| 7.3.2 Etapa de control de lámparas | 120 |
| 7.3.3 Etapa de filtrado | 121 |
| 7.4 Esquemática de enrutamiento | 121 |
| 7.4.1 Preparación para fabricación del PCB | 123 |
| 7.5 Verificación de enrutamiento | 123 |
| 7.6 Pruebas | 123 |

PRUEBAS EN EL TELESCOPIO **PÁGINA 127**

| | |
|---|-----|
| 8.1 Integración | 128 |
| 8.2 Verificación | 130 |
| 8.3 Operación | 132 |
| 8.4 Validación | 133 |
| 8.5 En curso de mantenimiento | 134 |

DISCUSIÓN Y CONCLUSIÓN **PÁGINA 141**

| | |
|--|-----|
| 9.1 Interpretación de los resultados | 142 |
| 9.2 Problemas encontrados | 145 |
| 9.3 Lecciones aprendidas | 146 |
| 9.4 Mejora para un trabajo futuro | 146 |

PROGRAMA DE CONTROL EN DYNAMIC C **PÁGINA 151**

CÓDIGO DEL CLIENTE EN LENGUAJE VISUAL STUDIO C++ **PÁGINA 167**

GRÁFICAS **PÁGINA 171**

DATOS OBTENIDOS DE LAS PRUEBAS DEL PUENTE H **PÁGINA 175**

HOJAS DE DATOS DE LOS DISPOSITIVOS ELECTRÓNICOS **PÁGINA 181**

PCB DEL MÓDULO ELECTRÓNICO **PÁGINA 189**

PLANOS MECÁNICOS **PÁGINA 193**

DATOS Y ESQUEMA DE CONEXIONES **PÁGINA 199**

TERMISTOR NTC **PÁGINA 205**

BIBLIOGRAFÍA **PÁGINA 209**

ÍNDICE DE CUADROS **PÁGINA 213**

ÍNDICE DE FIGURAS **PÁGINA 215**

PREFACIO

Esta tesis será presentada a la Facultad de Ingeniería de la Universidad Nacional Autónoma de México, como cumplimiento a los requerimientos para obtener el título de Ingeniero en Mecatrónica y es el resultado de un proyecto bajo la continua supervisión, durante todo el transcurso del trabajo, de la Doctora Margarita Rosado junto con el Físico Abel Bernal; con el financiamiento y apoyo del Instituto de Astronomía de la misma universidad.

La investigación, simulación, diseño, pruebas, análisis y fabricación del dispositivo se desarrolló en el Laboratorio de Electrónica y Detectores y en el Taller Mecánico del Instituto de Astronomía y se consideró una semana para su instalación, operación y pruebas en el telescopio de 2.1 m del Observatorio Astronómico Nacional, en San Pedro Mártir Baja California.

Las ideas presentadas aquí son el resultado directo de fructíferas discusiones y sugerencias con los asesores de esta tesis; la cual abarca campos de la ingeniería electrónica, de programación, de control, mecánica y de diseño; además de conocimientos de la Física en el área de la Óptica y un panorama general de las observaciones en la Astronomía. La exposición integral de estos conocimientos, se pretende, sea de una manera lógica y continua.

ESTRUCTURA DE LA TESIS

Este trabajo está dividido en dos partes. La Parte I de la tesis provee información fundamental para conocer el instrumento PUMA y el análisis y desarrollo del proyecto. La parte II muestra información complementaria, que fundamenta y ayuda a la comprensión de la Parte I.

La primera parte se distribuye en nueve capítulos de forma secuencial, introduciendo al conocimiento adquirido para el entendimiento del proyecto, siguiendo con el desarrollo y construcción del sistema.

En el Capítulo 1 se encuentra una introducción a la teoría óptica que rige el espectrógrafo integral de campo; mientras que el valor de la realización del nuevo sistema de control, la definición clara de los objetivos, la justificación y el alcance de la tesis se establece en el Capítulo 2. El Capítulo 3 trata de la descripción de los subsistemas que integran el instrumento PUMA, al igual que el análisis de los datos. La variedad de información contenida en este trabajo fue gracias a las numerables referencias que del tema del interferómetro Fabry-Pérot existen y se encuentran disponibles hoy en día y, que de una investigación exhaustiva, se trató de cohesionar en esta tesis.

Las características del funcionamiento y hardware del microcontrolador de placa reducida utilizado, se especifican en el Capítulo 4 y en el Capítulo 5 el desarrollo de la programación.

La realización de las pruebas del software de control y el registro de datos, al igual que el uso de la interfaz de usuario, se encuentran en el Capítulo 6.

El proceso de construcción del módulo de control, así como el diseño de la placa electrónica auxiliar, se incluyen en el Capítulo 7.

Parte de esta tesis trata de las pruebas realizadas en las observaciones en el telescopio de 2.1 m, el cual es operado por el equipo del OAN-SPM. Por lo que se describen las pruebas en el telescopio en el Capítulo 8.

Finalmente se hace una discusión de los resultados obtenidos en el Capítulo 9.

En la segunda parte, los capítulos se complementan con una serie de apéndices sobre material que soporta el desarrollo técnico del proyecto. Estos proporcionan algunos planos mecánicos de las piezas que constituyen el módulo de control del instrumento (Apéndice G), el código fuente definitivo del programa para la BL4S200 en lenguaje Dynamic C (Apéndice A), así como el código fuente que se utilizó para las pruebas previas del software de control (Apéndice B); los datos (Apéndice D) y algunas gráficas (Apéndice C) de comportamiento de las salidas de alta corriente de dichas pruebas, para el movimiento de los motores que activan los carros hallados dentro del instrumento; se muestran también los archivos de diseño del circuito impreso (Apéndice H) y sus archivos generados (Apéndice F). Se adjuntan algunos documentos (Apéndice I) que son para mejorar en un futuro el instrumento PUMA.

AGRADECIMIENTOS

Es un placer para mí expresar mi gratitud a muchas personas, quienes sin su ayuda, no se habría podido llevar a cabo este proyecto y a quienes han sido parte importante de mi vida en los últimos seis años.

Quisiera expresar mi sincero agradecimiento y respeto a mi directora de tesis, la Doctora Margarita Rosado Solís, investigadora del Instituto de Astronomía, por el compromiso y responsabilidad que me ha confiado durante todo este tiempo, por introducirme al fascinante mundo de la Astronomía y por apoyarme en todo momento, desde el comienzo de este proyecto, hasta perseguir el camino para conseguir mi grado de ingeniero. Su amigable carácter, disciplina, ética de trabajo y espíritu generoso es inspirador. Sin su paciencia infinita en el último año, yo no habría sido capaz de completar este trabajo; su inestimable experiencia e intelecto han formado muchos aspectos de esta tesis.

Estoy igualmente agradecido con mi co-asesor el Físico Abel Bernal Bejarle, por ofrecerme la oportunidad de trabajar en tan interesante proyecto, por la plena libertad de perseguir mis ideas para desarrollar mi curiosidad y creatividad, guiarme en la dirección correcta, por las agradables discusiones de ayuda y quien fuera tan generoso en darme su tiempo, pericia y perspectivas, por la cercana colaboración que hemos tenido y por supervisar mi investigación de tesis. Su conocimiento profundo del arte de la electrónica (de un valor incalculable), que compartió conmigo fue muy valioso, sus sugerencias útiles y su humor indispensable.

Quisiera agradecerles el apoyo durante todo este proyecto y su ayuda al corregir mi tesis. Nunca me hubiera imaginado haber trabajado en un buen proyecto con tan excelentes asesores, agradezco a ellos por pasar un sin número de horas enseñándome cómo ser un buen investigador, lo que permitirá convertirme en un mejor profesionalista.

Estoy particularmente agradecido con el Dr. Luis Artemio Martínez Vázquez. Con su ayuda nuestro proyecto puede comunicarse con la estación de trabajo, en la cual los astrónomos realizan sus observaciones. Gracias por el tiempo gastado ayudando con las pruebas del software y conversión de datos. También por aportar al proyecto, desde las pruebas hasta su instalación en el observatorio.

Al Dr. Leonel Gutiérrez quien en los días de instalación su ayuda fue indispensable, en las correcciones del programa, en la instalación mecánica y en el conocimiento general del OAN-SPM. Su gran entusiasmo fue parte importante para la excelente culminación del proyecto.

A numerosos investigadores quienes con sus comentarios personales y perspicacia científica durante la formación y ejecución de este trabajo, compartieron su experiencia. A lo largo de la misma línea de todas las personas importantes involucradas en el proyecto, agradezco al Ing. Gerardo Lara Lucario, quien en varias ocasiones asistió para ayudarme con la elaboración de los planos mecánicos y con los desafíos que encontré en el último año; al Ing. Leonardo Garcés Medina por la revisión de los archivos de diseño electrónico para su fabricación; al Fís. Alan Morgan Watson Forster, por su oportuna atención al proyecto y a mis trámites de servicio social, y al departamento de mecánica y manufactura por la fabricación de las piezas de aluminio que necesitábamos.

Realmente cada persona que he llegado a conocer a través de este proyecto ha sido excepcional, su entusiasmo y profundo conocimiento de áreas importantes de la investigación, enriquecieron mi pensamiento de la ingeniería, lo cual contribuyó a la calidad de este trabajo. Gracias por la atención que han puesto en este proyecto

Un agradecimiento a todas las personas agradables que he conocido en el Instituto de Astronomía. Incluso durante ese poco tiempo aprendí muchísimo de ellos. Estoy agradecido por haber tenido la oportunidad de estar rodeado de increíbles personas, por sus entusiastas colaboraciones, interesantes discusiones y sabio consejo.

Adicionalmente agradezco al instituto de Astronomía por darme un espacio de trabajo durante el curso del desarrollo del proyecto y por concederme viajar, arreglar mi itinerario y permitir mi estadía en San Pedro Mártir en Baja California en Octubre de 2015.

A Mónica Sánchez Cruces y a Nelli Cárdenas Martínez por introducirme a las observaciones astronómicas y quiénes me ayudaron a cerrar las brechas de su comprensión, lo que me llevó a realizar las conclusiones de esta tesis. A Ilse Plauchu Frayn, por las singulares fotos que tomamos en el observatorio, durante mi estancia, sirvieron de portada para este trabajo.

Agradezco al grupo de miembros del Observatorio Astronómico Nacional, por los buenos comentarios y discusiones que hemos tenido. Gracias a las personas quienes innovan y mantienen los telescopios del OAN SPM, hacen de ellos maravillosas aventuras para la investigación. Gracias por la oportunidad y las fantásticas experiencias que tuve en el observatorio, fue un aprendizaje formativo para mí e hizo de este trabajo de estudio uno de los más emocionantes y agradables.

Doy mi profunda gratitud al Consejo Nacional de Ciencia y Tecnología de México (CONACyT) (programa con folio CY-253085) y a la Dirección General Asuntos del Personal Académico (DGA-PA) de la UNAM, por medio del Programa de Apoyo a Proyectos de Investigación e Innovación Tecnológica (PAPIIT) (programa con folio IN103116), por financiar este ambicioso proyecto.

A mis amigos de la universidad, a quienes admiro por sus habilidades y talentos. Estoy muy feliz por la amistad que construimos durante estos años, por sus consejos que me ayudaron en todo momento, por apoyarme a través de este esfuerzo y por ser mi segunda familia en mi segunda casa: la UNAM. Siempre les estaré agradecido por los buenos tiempos y constante amistad. Fue un gran privilegio compartir estos últimos años con ustedes y espero pasemos muchos más años juntos.

Finalmente, nunca les podré agradecer lo suficiente a mis padres por todo su amor y apoyo, por ser personas muy inteligentes y con grandes valores éticos, siempre esforzándose más de la cuenta por el éxito de sus hijos. Durante mi vida ellos han creído en mí y siempre han sido una fuente de fuerza, de motivación, alegría, inspiración y reflexión, en el presente y serán en el futuro. No podría estar más agradecido por todo lo que han hecho por mí. Les estoy muy agradecido por sus contribuciones en mi educación y en mi vida.

Esperando que esta tesis cuya diligente y entrañable preparación, sirva como una fuente de realización personal a quien la lea.

Zimri Marsilli

Ciudad Universitaria, México, Junio 2016

RESUMEN

Un área de investigación astronómica en el Instituto de Astronomía es dirigida hacia el campo de la espectroscopía. Con ayuda de tecnología de vanguardia (desarrollada en el mismo instituto), se estudia la dinámica del medio interestelar en galaxias (sus campos de velocidades, curvas de rotación, distribución de masa, forma estructural, etc.), interacción entre dos o más galaxias y sus efectos, cinemática de nebulosas y el desarrollo de instrumentación astronómica más sofisticada.

Esta tesis describe el análisis, modelo, diseño, desarrollo e implementación de un sistema automatizado para controlar un instrumento espectroscópico, instalado en uno de los tres telescopios del Observatorio Astronómico Nacional en San Pedro Mártir.

El sistema se analizó y diseñó como un sistema de tiempo real automatizado, conformado de otros sistemas, capaz de interactuar y ser controlado por un microcontrolador de placa reducida, conformado por hardware y un software de control, para llevar a cabo las funciones deseadas por la estación de observación.

La placa BL4S200 es un sistema embebido que fue programado para controlar el instrumento PUMA e interactuar con la estación de trabajo del telescopio de 2.1 m, con soporte de comunicación tipo Ethernet y a través de la red del observatorio. La placa reducida Rabbit utilizada cuenta con 40 líneas que pueden ser programadas como salidas y/o entradas, algunas analógicas y las restantes digitales, justificando la tecnología implementada en el sistema. También se agregó hardware auxiliar para manipular la placa BL4S200, que controlará el movimiento de los elementos que conforman el instrumento PUMA, comunicándose con el controlador CS100 y la estación de observación.

El sistema se consideró como un sistema de control de operación, que interactúa en un ambiente autónomo, manteniendo el control entre las señales de entrada y salida, es decir, el sistema cumple con una real exigencia de las operaciones en tiempo real. El sistema de control recibe las señales externas, las procesa y devuelve las respuestas casi instantáneamente (como réplica a las señales de entrada), para influir en los movimientos de su ambiente.

La programación se realizó de manera estructurada, a partir de cada función que debía realizar cada dispositivo que integra el instrumento; cada módulo de la programación define las acciones a conmutaciones en finales de carrera y a detecciones en los sensores de proximidad, ambos casos aplican en los carros móviles, a interrupciones y a señales de pulsos que envía y recibe por parte del controlador CS100.

Las pruebas que se realizaron en el Laboratorio de Electrónica y Detectores del Instituto de Astronomía, consistieron en la prueba del código del programa de control, en la fiabilidad de la comunicación ETHERNET y del protocolo TCP/IP, y en garantizar la capacidad de las salidas de alta corriente de la placa Rabbit para manejar los actuadores, para el movimiento de los elementos ópticos del instrumento PUMA.

Se describe el comportamiento del sistema en conjunto con el nuevo sistema de control, con pruebas en operación y especificando las mejoras para un trabajo a futuro.

El nuevo sistema de control electrónico está actualmente en uso en el Observatorio Astronómico Nacional en la Sierra de San Pedro Mártir en el estado de Baja California. Se logró construir un sistema escalable, entendible y mantenible, con capacidad adaptable a las necesidades del instrumento y de interactuar armónicamente con el sistema físico

Parte I

TESIS

CAPÍTULO

1

Introducción

Este primer capítulo presenta una introducción a los fundamentos del interferómetro de Fabry-Pérot, proporcionando su historia, algunos de sus desarrollos y un análisis teórico del equipo PU-MA, que se utiliza para la investigación científica en el área de la astrofísica.

La intención de presentar la siguiente información es dar al lector el conocimiento necesario para entender las bases del funcionamiento del interferómetro Fabry-Pérot de barrido y la técnica de espectroscopía 3D.

Índice

| | |
|---|----------|
| 1.1 Interferometría óptica | 6 |
|---|----------|



Figura 1.1: Observatorio de la Sierra de San Pedro Mártir en el cual se encuentra el telescopio de 2.1 m.
Fotografía tomada por Zimri Marsilli.

Galaxias espirales, nebulosas galácticas, cúmulos estelares y planetas circundantes, son algunos de los objetos distantes que estudia la Astronomía; sin embargo, para descubrir y contemplar estos cuerpos celestes, se necesita de la instrumentación astronómica, en donde se gestione la investigación aplicada y se estudien los conceptos teóricos de la óptica física y de la astronomía, para crear y realizar proyectos, sistemas y procesos que sean más precisos, inteligentes y seguros que sus predecesores, gracias a la implementación de nuevas tecnologías (Rosado et al., 1995), (Vázquez Pérez, 2015), (Fuentes Carrera, 2001), (Sánchez Cruces, 2013).

Una posible innovación se produce con tecnología de vanguardia, mejor adecuada a las necesidades de las observaciones astronómicas y a un mejor control de los instrumentos, que para estas actividades se requieran. No obstante cierta tecnología tiene que desarrollarse, puesto que algunos problemas requieren exigencias específicas.

La espectrometría integral de campo es una técnica de la instrumentación astronómica, que permite la observación de fuentes extendidas y difusas (caracterizadas por su tamaño angular, subestructura y características espectrales (Duxbury, 2012), e.g. una nube de gas, estrellas densas o una galaxia completa). Es posible obtener de una región del espacio que se esté observando, su espectro, a partir de la técnica de la espectroscopía tridimensional. La espectroscopía 3D utiliza los datos astronómicos en un arreglo tridimensional, llamado cubo, que incluye una determinada cantidad de imágenes de alto contraste de la región del cielo observada, cada una a diferente longitud de onda (University of Wisconsin-Madison, 2016).

Astrónomos, físicos ópticos e ingenieros del Instituto de Astronomía de la UNAM, a cargo de la Dra. Margarita Rosado y el Fís. Abel Bernal, han construido un espectrógrafo Integral de campo llamado *PUMA*, para observar e investigar, en el rango visible de la luz, los campos de velocidades de objetos en el espacio y otras propiedades. El instrumento se instaló en el año de 1994, está optimizado para funcionar en la parte óptica del espectro y tiene integrado como elemento óptico dispersivo de precisión, un interferómetro Fabry-Pérot de barrido, cuya información se usa en la investigación espectroscópica (Rosado et al., 1995).

Capítulo 1. Introducción

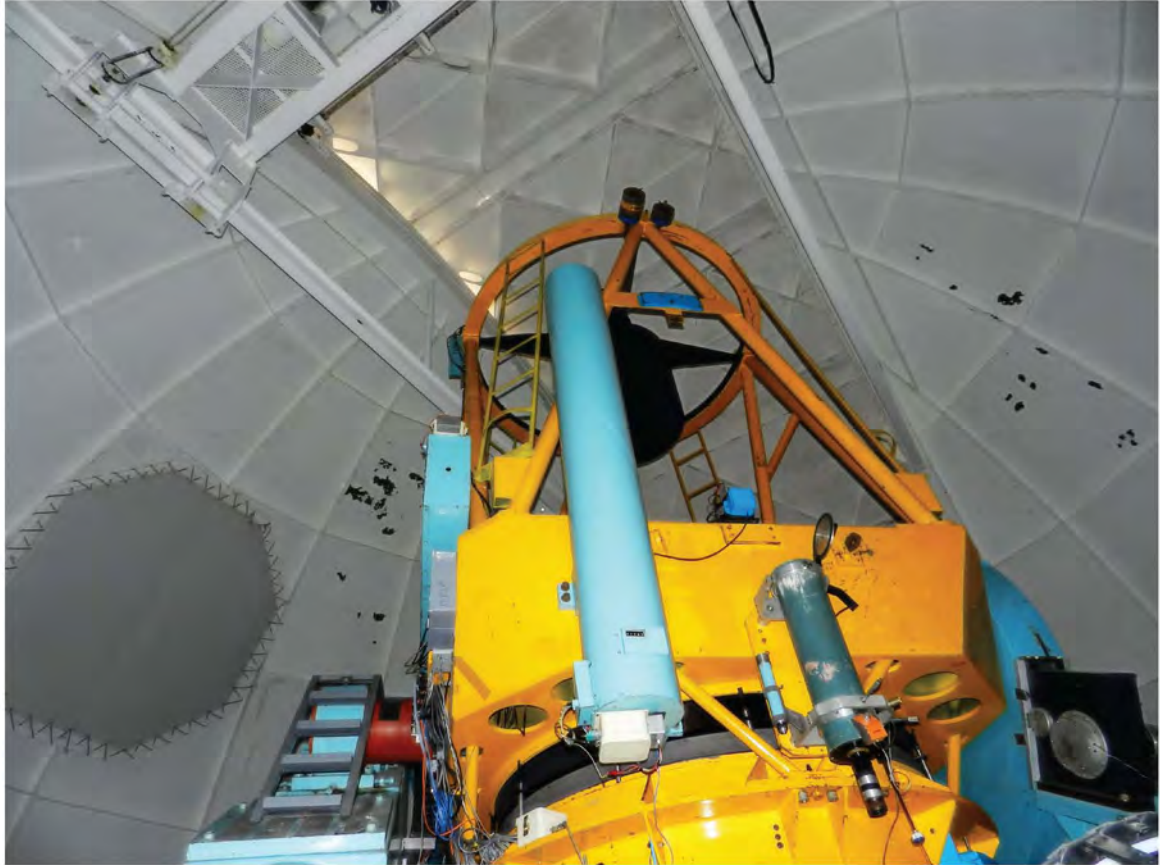


Figura 1.2: Telescopio de 2.1 m en el cual se instala el instrumento PUMA. *Fotografía tomada por Zimri Marsilli*

La técnica de exploración de Fabry-Pérot, para la espectroscopía integral de campo, tiene la capacidad de grabar un espectro de cada parte del objeto extendido, permitiendo obtener mucha más información del objeto en una sola exposición, de modo tal que se obtiene un cubo de datos (x, y, λ) , con dimensiones espaciales x e y y de la longitud de onda λ , que debe ser construido a través de una serie de exposiciones a partir de una longitud de onda deseada, haciendo mejor uso de la luz incidente.

Durante la reducción de datos se calcula la longitud de onda de cada canal (exposición), y el cubo de datos se crea y se reorganiza, de tal manera que el eje z corresponda a la longitud de onda. Se reconoce un objeto a una alta resolución espacial y a una muy alta resolución espectral, mediante la producción de una franja de transmisión, que es extremadamente estrecha en anchura de la banda espectral (Bershady, 2010).

El instrumento *PUMA* está en funcionamiento colocado en el foco Cassegrain, en la configuración de $f/7.5$ del telescopio de 2.12 m (Figura 8.9), del Observatorio Astronómico Nacional (Figura 1.1), laboratorio nacional de investigación científica y dependencia del Instituto de Astronomía de la Universidad Nacional Autónoma de México, ubicado en la sierra de San Pedro Mártir (OAN-SPM) en Baja California, al Noroeste de México a una altitud de 2830 m sobre el nivel del mar y localizado a una longitud de $115^{\circ}27'49''$ Oeste y a una latitud de $31^{\circ}02'39''$ Norte, en un lugar ideal para la observación astronómica (IA-UNAM, 2015).

1.1 Interferometría óptica

El interferómetro Fabry-Pérot es una importante herramienta en el estudio de los espectros y tiene principal beneficio en las áreas de la física cuántica, la óptica, la espectroscopía, la astronomía y la astrofísica. La configuración del instrumento es muy sencilla; pero con este dispositivo se pueden observar directamente las estructuras finas de las líneas espectrales y se pueden comparar fácilmente las longitudes de onda.

A más de un siglo de su construcción y planteamiento teórico, el interferómetro de Fabry y Pérot sigue proporcionando información invaluable a las investigaciones astronómicas. Como describe Mulligan (Mulligan, 1998) la cita de Charles Fabry: "*todo en el universo está ligado, y no hay por que separar el estudio astronómico de la ciencia física...*", así Fabry y Pérot descubrieron una forma refinada de poder entender al universo y los fenómenos cuánticos.

1.1.1 El inicio de una tecnología

Charles Fabry y Alfred Pérot, físicos franceses pertenecientes a la *Université de Marseille*, publicaron en el año de 1897 el artículo "*Sur les franges des lames minces argentées et leur application à la mesures de petites épaisseurs d'air*" en *Annales de Chimie et de Physique*, en el cual presentaban un nuevo dispositivo de interferencia (Vaughan, 1989), con fundamento teórico en la interferencia de haces múltiples de luz que inciden sobre dos placas (ligeramente platinadas y colocadas lo más paralelamente posible), haciendo posible la aparición de patrones de interferencia en forma de anillos concéntricos. Sobre estos es posible realizar mediciones y comparaciones de las longitudes de onda con gran precisión, fiabilidad y facilidad, provenientes de lámparas de comparación y de sus espectros de emisión.

Años antes en 1831 George Biddel Airy (físico inglés de la Universidad de Cambridge), explicó el análisis teórico del fenómeno de interferencia, con el enfoque de la naturaleza ondulatoria de la luz, diciendo que cuando dos ondas con la misma longitud coexistían se producía interferencia y encontraban la intensidad de combinación. Airy lo ejemplificó con dos placas de vidrio separadas por un espaciamiento muy pequeño (Figura 1.3), calculando la intensidad refractada, el retraso del haz entre las placas de vidrio y el brillo de la luz reflejada. Observó el carácter complementario de la luz reflejada y transmitida, así como la diferencia de la banda de colores con respecto a otros fenómenos de interferencia; sin embargo en sus escritos no mencionó la formación de algún patrón de anillos circulares, debido a los haces de luz que interferían.

El fenómeno de los anillos de interferencia lo describió Haidinger en 1855 (Vaughan, 1989), cuando hizo incidir un haz de luz en una placa de mica, transmitiendo la luz y reflejando parte de esta al ojo. Lord Rayleigh diferenció este patrón de anillos concéntricos descubiertos por Haidinger de los anillos de Newton, a causa de la dependencia del espaciamiento entre las placas que los producían. En 1893 Boulouch, fue el primero en obtener líneas del espectro del sodio, en forma de anillos de interferencia, realizando sus cálculos con los fundamentos de Airy para la luz transmitida y reflejada. Con los mismos principios de construcción, comprobó que los anillos de transmisión son complementarios de los anillos reflejados.

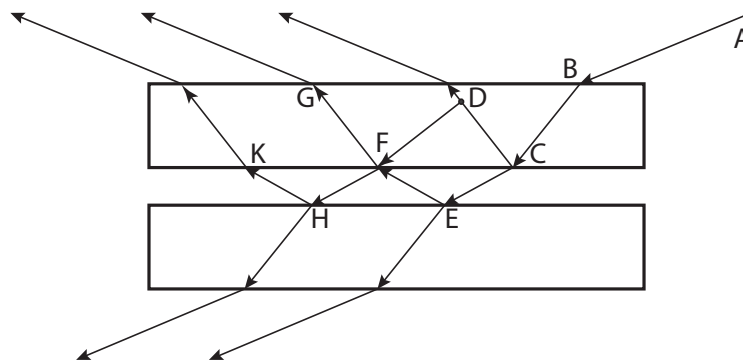


Figura 1.3: Representación de Airy del fenómeno de interferencia, el rayo AB es el incidente, el cual deriva las reflexiones y refracciones CE, EF, FH y HK. Referencia (Vaughan, 1989).

En 1899 Fabry y Pérot describieron un dispositivo para realizar mediciones precisas de distancias muy pequeñas y la aplicación de un nuevo método de espectroscopía de interferencia, para la comparación de longitudes de onda. Señalando claramente una relación de nitidez y contraste, y derivando una expresión de Airy para la transmisión de la luz (Vaughan, 1989):

$$I = \frac{\theta^2}{(1-f)^2} \frac{1}{1 + \left[\frac{4f}{(1-f)^2} \right] \sin^2 \frac{\pi\Delta}{\lambda}} \quad (1.1)$$

Siendo θ y f los coeficientes de transmisión y reflexión respectivamente, Δ el retraso en el camino óptico y λ la longitud de onda. Ellos se refirieron al término del denominador de la ecuación 1.1, como la nitidez de los anillos, que puede ser considerado como la fineza (ecuación 1.2). Para un caso ideal el máximo de la intensidad, sería igual a la luz incidente.

$$\frac{4f}{(1-f)^2} \quad (1.2)$$

Fabry y Pérot también describieron que la formación de los anillos de interferencia estaban localizados en el infinito, en dirección normal a las placas y que este patrón de interferencia no se podía formar, a menos de que las superficies de estas placas estuvieran perfectamente planas y paralelas. También mencionaron que a medida que el espaciamiento entre las placas aumentaba, el diámetro de los anillos disminuía.

Para sus primeras mediciones con fuentes luminosas, utilizaron descargas de arco de metal y construyeron un dispositivo óptico que llamaron *étalon*, con el cual pudieron medir desplazamientos de aproximadamente 200 partes de la longitud de onda incidente. En él separaban las placas de vidrio con una cuña y proyectaban los anillos de interferencia y sus órdenes de varias líneas espectrales en una pantalla. Posteriormente diseñaron otro instrumento con una de sus placas en movimiento, siendo el primer *interferómetro de Fabry-Pérot de barrido*.

Fabry y Pérot describieron el equipo y lo llamaron espectroscopio de interferencia. Este dispositivo lo colocaron en un telescopio, considerando que los rayos incidentes del telescopio son paralelos, comentando la importancia del mantener estable el paralelismo de las placas de vidrio platinadas, aún con su desplazamiento micrométrico a lo largo de un carro con movimiento guiado, sin algún otro tipo de fuerza o movimiento que lo afectara. Los reajustes de paralelismo

de las placas y el movimiento azimutal del dispositivo, lo realizaban por medio de tornillos micrométricos para controlar desplazamientos finos, cada vuelta de tornillo correspondía a $\sim 3 \mu\text{m}$; con un escala unida al carro y leída con un microscopio fijo, obteniendo una medida aproximada del desplazamiento del carro y, en consecuencia, de los desplazamientos de las placas. El dispositivo estaba protegido a vibraciones externas. En el año 1901 un nuevo modelo fue construido por M. Jobin (Figura 1.4), en éste la resolución era de $0.15 \mu\text{m}$ para los desplazamientos.

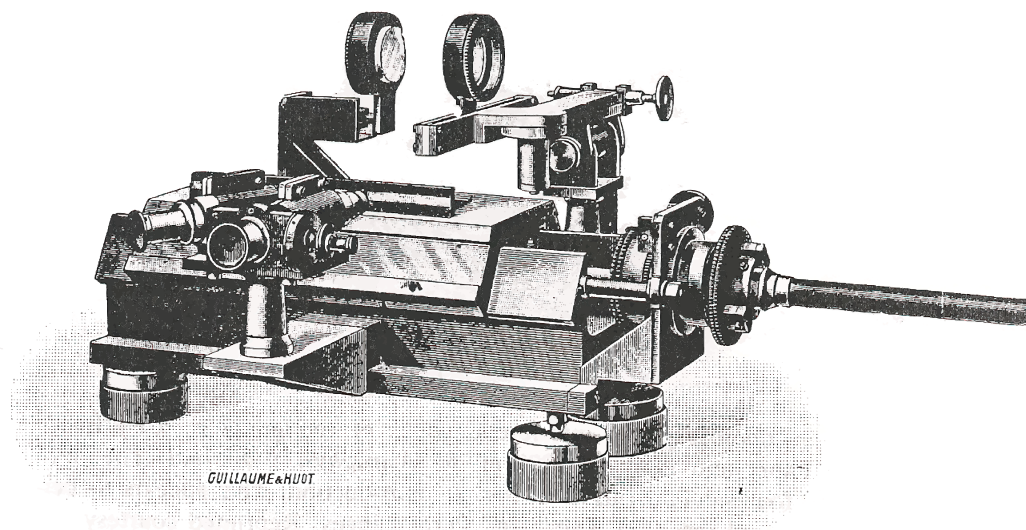


Figura 1.4: El interferómetro de alta ingeniería construido por M Jobin para Fabry y Pérot, descrito en 1899 y construido en 1901. Referencia (Vaughan, 1989).

Para el análisis de los espectros de las fuentes luminosas, Fabry y Pérot crearon el método de las coincidencias, en el que se podían estudiar fácilmente los cambios de longitud de onda de una línea dada; sin embargo tenían que conocer con precisión las proporciones de las longitudes de onda empleadas. Ellos eligieron como estándar la línea roja del cadmio, mediante una separación de placas uniforme concluyeron que el poder de dispersión era mejor que en otros dispositivos espectroscópicos. Continuaron con el estudio de las líneas de absorción que emitía el espectro del Sol e investigaron con otros tipos de fuentes de luz. El dispositivo era suficiente para mostrar los detalles finos más difíciles de distinguir.

Posteriormente implementaron el método de diámetros, con separación fija e iluminación sucesiva, el cual consistía en obtener, para cada longitud de onda, un orden en el centro del patrón de interferencia. Los diámetros angulares se midieron con un telescopio, provisto de un ocular micrométrico, para obtener una mejor exactitud en sus mediciones, además de observar que el efecto Doppler tenía consecuencias en sus mediciones, mostrando una expansión o contracción en los anillos de interferencia. Fabry de una sola exposición fotográfica obtuvo los espectros de interferencia y el registro de varias líneas espectrales (Figura 1.5), en donde cada anillo era un espectro de la fuente.

Las diferencias con otro tipos de instrumentos de interferencia, es que con el interferómetro de Fabry-Pérot, no hay gran cantidad de luz perdida debido a la reflexión del platinado de las placas, de modo que se transmite sólo un pequeño porcentaje y que tiene una construcción más sencilla y más precisa.

Fabry y Buisson se basaron en lámparas de descarga de arco metálico para la comparación y medición de líneas espectrales, proporcionando nuevos detalles experimentales, procedimientos,

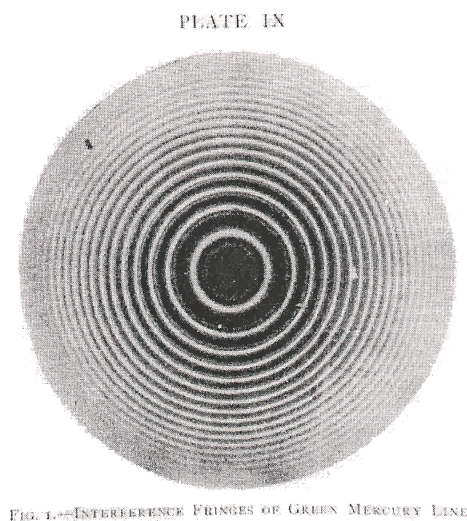


Figura 1.5: Primera fotografía de anillos de interferencia de la línea de mercurio verde en $\lambda = 5460,7424\text{\AA}$, dada por un arco de mercurio en el vacío. Referencia (Vaughan, 1989).

calibraciones, ajustes de alineación y registros fotográficos. En 1905, ellos entre otros físicos, en un congreso en Oxford, decidieron establecer longitudes de onda de un cierto número de líneas de arco como estándares espectroscópicos. Entre los acuerdos la línea de rojo del cadmio fue el patrón primario, designándola como 6438.4696\AA . También Fabry y Buisson contribuyeron a la definición de la medida Ångström aproximándola a 10^{-10} m .

Eversheim en 1907, tomó en consideración la temperatura como factor para la estabilidad de las mediciones de longitud de onda en un interferómetro, concluyendo que una décima de grado afectaba el diámetro de los anillos de interferencia. Pfund en 1908, mejoró la reflectividad de las placas con un tratamiento químico y la planicidad de sus superficies puliéndolas, con esto mantenía el diámetro de los anillos en cada observación, conservando la temperatura con circulación de agua y reguladores de temperatura (Vaughan, 1989).

Fabry y Buisson realizaron las primeras observaciones astronómicas con los métodos de interferometría estudiando nebulosas. Ellos colocaron el dispositivo en el foco de un telescopio de 80 cm de diámetro (del observatorio de Marsella), puesto sobre una nebulosa y observando en el mismo plano la imagen real del objeto y la imagen de los anillos de interferencia. Calcularon, de las líneas de hidrógeno, la velocidad radial del objeto por las diferencias de longitud de onda de un punto a otro, la temperatura por el tamaño de las mismas líneas, mientras que las líneas de origen les proporcionaron información del peso atómico del gas que lo conformaba.

En el nuevo instrumento que utilizaron colocaron filtros de absorción para asilar diferentes líneas y minimizar las pérdidas de luz, logrando el estudio de fuentes de luz débiles. En 1912 Buisson y Fabry midieron el ancho de las líneas espectrales del helio, neón y kriptón, empezando así un estudio de espectroscopía atómica y astronómica. Observaron que ciertos cambios de presión cambiaban el desplazamiento de las líneas solares, tomando como explicación algunas teorías de Einstein, concluyeron que fenómenos adicionales pueden provocar una falsa posición de las líneas espectrales.

A partir de la segunda mitad del siglo XX, el interferómetro de Fabry-Pérot se comenzó a perfeccionar con nuevas formas de manufactura de las placas, obteniendo superficies más planas y

mejores coeficientes de reflexión, además de tener más precisión de paralelismo en el espaciamiento entre las placas. El interferómetro de Fabry-Pérot de barrido sigue siendo, por lo tanto, un instrumento de espectroscopía sofisticado hoy en día.

1.1.2 Teoría de la interferometría

Para llegar a las ecuaciones derivadas de las fórmulas de Airy, con las que Fabry y Pérot describieron el funcionamiento fundamental de su interferómetro (para la luz transmitida y la luz reflejada), se debe comprender el estudio de la interferencia de haces múltiples (Born and Wolf, 2005), para dos placas paralelas parcialmente reflejantes. En este esquema las reflexiones múltiples refuerzan las áreas donde los efectos constructivos y destructivos ocurren, construyendo el patrón de interferencia resultante mucho más definido, lo que ayuda a tener mayor precisión de las medidas de longitud de onda y de rango espectral libre.

Los conceptos y técnicas de espectroscopía óptica del interferómetro de Fabry-Pérot, parten del concepto de campo electromagnético $E(r, t)$ como principal manifestación de la onda de luz que se detecta y se somete a análisis. Este haz contiene propiedades de coherencia espacial y temporal y, para fenómenos de interferometría de Fabry-Pérot, sufre división de amplitud de onda por las placas dispuestas en paralelo, causando un retardo del campo electromagnético entre las placas; por consiguiente y como resultado, produce interferencia constructiva a cierta longitud de onda y ángulo de incidencia particular. Mediante la dispersión de varias longitudes de onda a ciertas frecuencias, se podrá determinar una medición de un espectro de alguna fuente extendida, caracterizando el campo de luz incidente.

La interferometría sigue siendo una guía precisa para la medición de longitudes de onda. La longitud de onda se define como $\lambda_0 = c/\nu$, donde ν es la frecuencia de radicación y c la velocidad de la luz.

Para una fuente extensa los perfiles circulares de interferencia solamente ocurren cuando las placas que los producen están colocadas en paralelo y cambian parcialmente la distancia entre ellas, por lo que el patrón de interferencia se moverá debido al cambio de longitudes de interferencia constructiva y destructiva.

El análisis geométrico del rayo permite obtener parámetros importantes del fenómeno de la interferometría como es la *finesse*, el poder de resolución y el rango espectral libre, además de que este tratamiento de la luz resulta satisfactorio.

Como se muestra en la Figura 1.6, un rayo i que incide a un ángulo θ_s en una de las placas paralelas, con un espaciamiento físico de d_s y con amplitud unitaria, representará la dirección de una onda plana. La longitud de la onda de luz λ_0 (a vacío), producirá sucesivas reflexiones $a, b, c, f, g \dots$ y transmisiones $\alpha, \beta, \gamma, \xi \dots$, con índice de refracción efectivo de η_s .

El vector de campo eléctrico de la onda electromagnética incidente, tendrá amplitud unitaria y estará polarizada perpendicularmente al plano de incidencia. Los rayos reflejados se considerarán positivos de derecha a izquierda y los rayos transmitidos se considerarán positivos de izquierda a derecha.

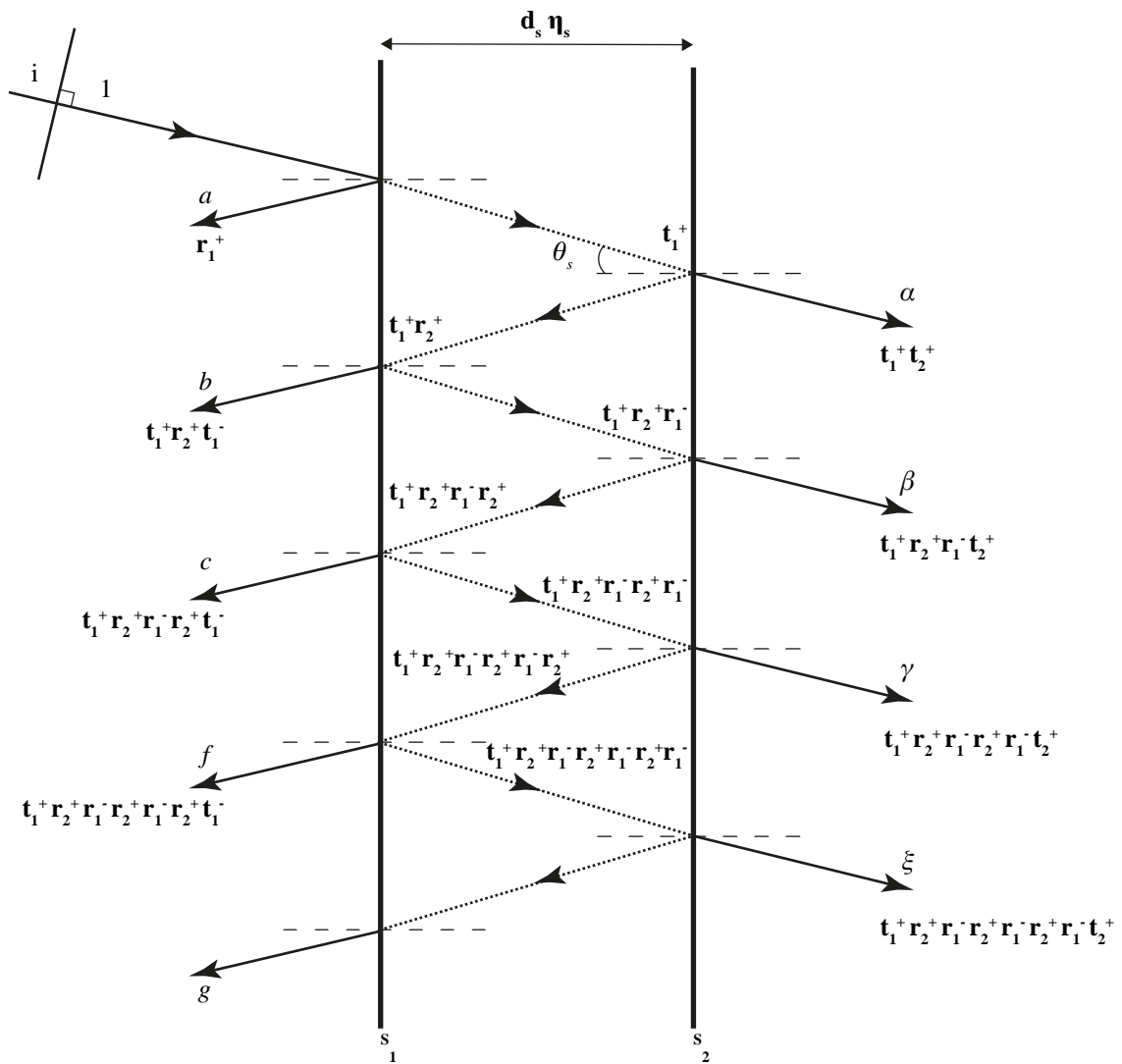


Figura 1.6: Las ondas planas se reflejan sucesivamente entre dos superficies paralelas planas. *Referencia (Vaughan, 1989).*

La diferencia de fase δ entre dos ondas sucesivamente transmitidas, debida al retraso ocasionado por la diferencia de camino en las múltiples reflexiones, está dada por:

$$\delta = \frac{2\pi (2\eta_s d_s \cos\theta_s)}{\lambda_0} \tag{1.3}$$

Una onda plana es la más simple representación de una onda en movimiento, tiene una amplitud uniforme y está en fase en cualquier plano perpendicular a su dirección de propagación (Schmidt, 2010). Más generalmente un campo de una onda plana está dado por:

$$E(r) = Ae^{ikr} \tag{1.4}$$

Si el primer orden m de rayos transmitidos son superpuestos, la amplitud del campo $E_t(m)$ de la

resultante del vector eléctrico de la luz transmitida es:

$$E_t(m) = (t_1^+ t_2^+) + (t_1^+ r_2^+ r_1^- t_2^+) e^{i\delta} + (t_1^+ r_2^+ r_1^- r_2^+ r_1^- t_2^+) e^{i2\delta} + (t_1^+ r_2^+ r_1^- r_2^+ r_1^- r_2^+ r_1^- t_2^+) e^{i3\delta} + \dots \quad (1.5)$$

Así con la excepción de la primera reflexión a , ambas series de rayos forman progresiones geométricas (Born and Wolf, 2005), con el mismo factor $(r_2^+ r_1^-) e^{i\delta}$:

$$E_t(m) = (t_1^+ t_2^+) + [t_1^+ (r_2^+ r_1^-) t_2^+] e^{i\delta} + [t_1^+ (r_2^+ r_1^-)^2 t_2^+] e^{i2\delta} + [t_1^+ (r_2^+ r_1^-)^3 t_2^+] e^{i3\delta} + \dots \quad (1.6)$$

En función de m :

$$E_t(m) = (t_1^+ t_2^+) \left[1 + (r_2^+ r_1^-) e^{i\delta} + (r_2^+ r_1^-)^2 e^{i2\delta} + (r_2^+ r_1^-)^3 e^{i3\delta} + \dots + (r_2^+ r_1^-)^{(m-1)} e^{i(m-1)\delta} \right] \quad (1.7)$$

La serie geométrica converge mientras $|(r_2^+ r_1^-) e^{i\delta}| < 1$ (Hecht and Zajac, 2002), y el número de términos se acerque a infinito, entonces:

$$E_t(m) = (t_1^+ t_2^+) \left[\frac{1 - (r_2^+ r_1^-)^m e^{im\delta}}{1 - (r_2^+ r_1^-) e^{i\delta}} \right] \quad (1.8)$$

Para un número infinito de rayos de luz, $m \rightarrow \infty$

$$E_t = E_t(m \rightarrow \infty) = \frac{t_1^+ t_2^+}{1 - (r_2^+ r_1^-) e^{i\delta}} \quad (1.9)$$

El promedio de la intensidad transmitida para un campo instantáneo, está dado por su módulo $E_t E_t^*$ (Vaughan, 1989):

$$I_t = \frac{|t_1^+ t_2^+|^2}{1 + |r_2^+ r_1^-|^2 - 2|r_2^+ r_1^-| \cos \delta} \quad (1.10)$$

Las superficies S_1 y S_2 son idénticas, por lo que los coeficientes r y t son reales y están en términos de los ángulos e índices de refracción (Macleod, 2001); por tanto las ecuaciones derivadas de las relaciones de Stokes del principio de reversibilidad (Hecht and Zajac, 2002) son:

$$t^+ t^- + r^+ r^- = 1 \quad y \quad r^+ t^+ + t^+ r^- = 0 \quad de \quad ahí \quad t^+ t^- = 1 - r^{+2} \quad y \quad r^+ = -r^- \quad (1.11)$$

Como los rayos transmitidos $t_1^+ = t_2^+$ y reflejados $r_1^+ = r_2^+$ y como no hay absorción en las placas dieléctricas (Macleod, 2001) entonces

$$R + T = 1 \quad donde \quad t^+ t^- = T \quad r^{+2} = r^{-2} = R \quad (1.12)$$

R y T son los coeficientes de reflexión y transmisión respectivamente, sustituyéndolos en la ecuación 1.10:

$$I_T = \frac{T^2}{1 + R^2 - 2R \cos \delta} \quad (1.13)$$

Con la relación trigonométrica para función par:

$$\sin^2 \frac{\delta}{2} = \frac{1 - \cos 2\delta}{2} \quad (1.14)$$

la ecuación es:

$$I_T = \frac{T^2}{(1 - R)^2 + 4R \sin^2 \left(\frac{\delta}{2} \right)} \quad (1.15)$$

Factorizando $\frac{T^2}{(1-R)^2}$ de 1.15:

$$I_T = \frac{T^2}{(1 - R)^2} \left(\frac{1}{1 + \left[\frac{4R}{(1-R)^2} \right] \sin^2 \left(\frac{\delta}{2} \right)} \right) \quad (1.16)$$

Por: $1 - R = T$

$$I_T = \frac{1}{1 + \left[\frac{4R}{(1-R)^2} \right] \sin^2 \left(\frac{\delta}{2} \right)} \quad (1.17)$$

Si el primer orden m de rayos reflejados son superpuestos, la amplitud del campo $E_r(m)$ de la resultante del vector eléctrico de la luz reflejada es:

$$E_r(m) = (r_1^+) + (t_1^+ r_2^+ t_1^-) e^{i\delta} + (t_1^+ r_2^+ r_1^- r_2^+ t_1^-) e^{i2\delta} + (t_1^+ r_2^+ r_1^- r_2^+ r_1^- r_2^+ t_1^-) e^{i3\delta} + \dots \quad (1.18)$$

A excepción de la primera reflexión, factorizando el término $(t_1^+ r_2^+ t_1^-) e^{i\delta}$ de la ecuación:

$$E_r(m) = (r_1^+) + (t_1^+ r_2^+ t_1^-) e^{i\delta} \left[1 + (r_1^- r_2^+) e^{i\delta} + (r_1^- r_2^+ r_1^- r_2^+) e^{i2\delta} + \dots + (r_1^- r_2^+)^{(m-2)} e^{i(m-2)\delta} \right] \quad (1.19)$$

La serie geométrica converge mientras $|(r_1^- r_2^+) e^{i\delta}| < 1$, y el número de términos se acerque a infinito, entonces:

$$E_r(m) = r_1^+ + \frac{(t_1^+ r_2^+ t_1^-) e^{i\delta} [(1 - r_1^- r_2^+) e^{i(m-1)\delta}]}{1 - (r_1^- r_2^+) e^{i\delta}} \quad (1.20)$$

Para un número infinito de rayos de luz, $m \rightarrow \infty$

$$E_r = r_1^+ + \frac{(t_1^+ r_2^+ t_1^-) e^{i\delta}}{1 - (r_1^- r_2^+) e^{i\delta}} \quad (1.21)$$

Por las ecuaciones derivadas de las relaciones de Stokes del principio de reversibilidad (Hecht and Zajac, 2002):

$$E_r = r_1^+ - \frac{(t_1^+ r_1^+ t_1^-) e^{i\delta}}{(1 - r_1^{-2} e^{i\delta})} \quad (1.22)$$

$$E_r = r_1^+ - \frac{\sqrt{R}(1 - Re^{i\delta} - Te^{i\delta})}{(1 - Re^{i\delta})} \quad (1.23)$$

$$E_r = r_1^+ - \frac{\sqrt{R}(1 - e^{i\delta})}{(1 - Re^{i\delta})} \quad (1.24)$$

El promedio de la intensidad transmitida, para un campo instantáneo está dado por su módulo (Vaughan, 1989) $E_r E_r^*$

$$I_R = \frac{R(2 - 2\cos\delta)}{(1 + R^2 - 2R\cos\delta)} \quad (1.25)$$

Con la relación trigonométrica, para función par:

$$I_R = \frac{4R\sin^2\left(\frac{\delta}{2}\right)}{\left[(1 - R)^2 + 4R\sin^2\left(\frac{\delta}{2}\right)\right]} \quad (1.26)$$

Se observa que al igual que $T + R = 1$ lo es $I_R + I_T = 1$, es decir, no hay pérdidas por absorción y las fórmulas resultan complementarias.

Se observa que cuando $R \rightarrow 1$, la luz transmitida se convierte en estrechas franjas brillantes en un fondo oscuro y la luz reflejada se convierte en estrechas franjas oscuras en un fondo brillante.

$$\frac{4R}{(1 - R)^2} \quad (1.27)$$

El término de la ecuación 1.27 se conoce como fineza, entonces las ecuaciones de intensidad en transmisión (Figura 1.7) y reflexión (Figura 1.8), son de la siguiente forma:

$$I_R = \frac{F\sin^2\left(\frac{\delta}{2}\right)}{\left[1 + F\sin^2\left(\frac{\delta}{2}\right)\right]} \quad (1.28)$$

$$I_T = \frac{1}{1 + F\sin^2\left(\frac{\delta}{2}\right)} \quad (1.29)$$

Las ecuaciones anteriores son conocidas como la fórmula de Airy. Al observar la gráfica de transmisión T contra el retraso óptico δ , los máximos de transmisión son franjas cuando el valor de F incrementa y el ancho de las franjas se vuelve estrecho. La relación de la separación de las franjas

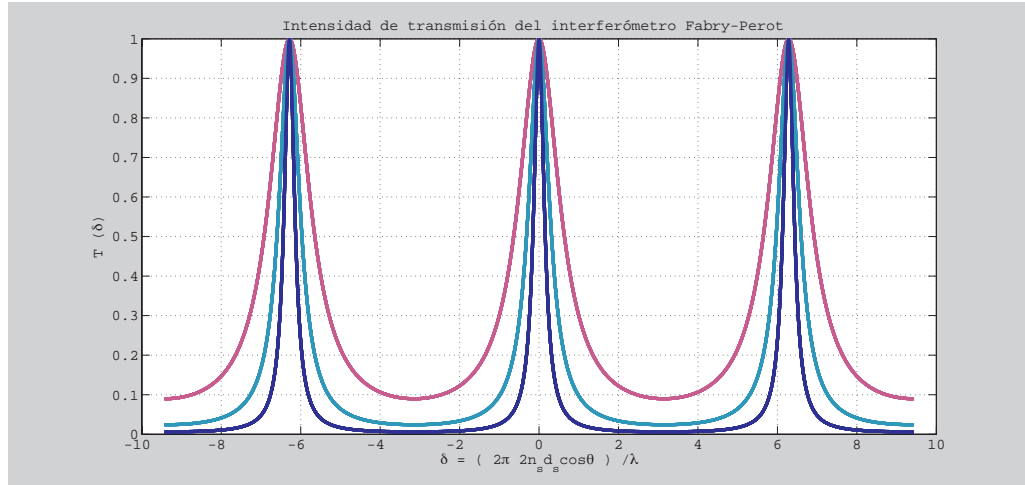


Figura 1.7: Intensidad de transmisión del interferómetro Fabry-Pérot. Gráfica realizada por Zimri Marsilli.

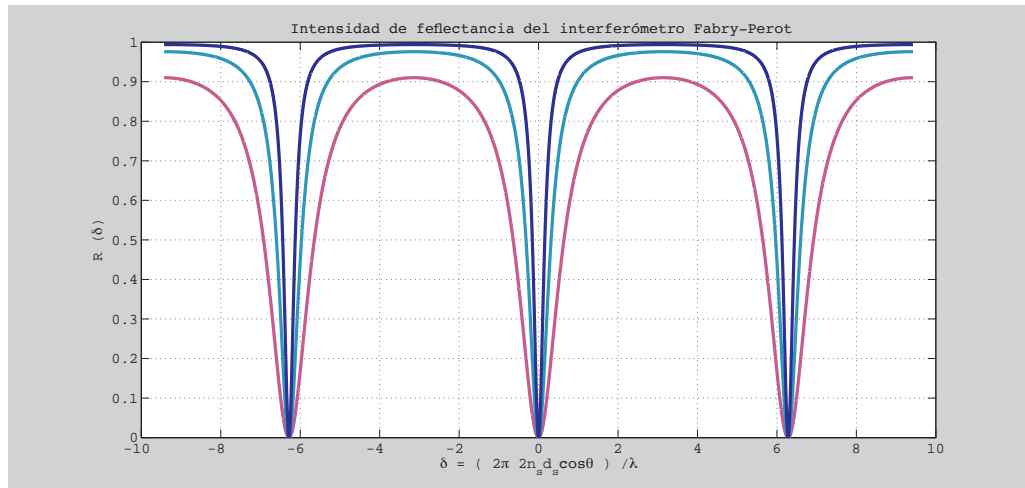


Figura 1.8: Intensidad de reflectancia del interferómetro de FP. Gráfica realizada por Zimri Marsilli.

adyacentes con respecto a la mitad del ancho de la franja del máximo de intensidad, es llamado finesse \mathcal{F} , entonces:

$$\frac{1}{2} = \frac{1}{1 + F \sin^2\left(\frac{\delta}{2}\right)} \tag{1.30}$$

Pero si el valor de F es grande, entonces δ es pequeño y se puede aproximar:

$$\sin^2\left(\frac{\delta}{2}\right) \approx \left(\frac{\delta}{2}\right)^2 \tag{1.31}$$

Se observa que el valor de T es máximo para cuando existe una interferencia constructiva (Sears et al., 2008), es decir cuando el camino óptico $2\eta_s d_s \cos\theta_s$ es igual a múltiplos enteros de la lon-

gitud de onda $m\lambda$, cuando $m = 0, \pm 1, \pm 2, \dots$, entonces:

$$\frac{\delta}{2} = 2\eta_s d_s \cos \theta_s = m\pi \quad (1.32)$$

$$\delta = 2m\pi \quad (1.33)$$

Para la intensidad a la mitad del máximo:

$$\frac{\delta}{2} = m\pi \pm \frac{\epsilon}{2} \quad (1.34)$$

Donde ϵ es el la mitad del ancho del máximo de intensidad para la transmisión de la luz, entonces:

$$\frac{1}{2} = \frac{1}{1 + F\left(\frac{\epsilon}{4}\right)^2} \quad (1.35)$$

es decir:

$$\epsilon = \frac{4}{F^{\frac{1}{2}}} \quad (1.36)$$

La separación entre los valores de ϵ representa las franjas sucesivas de 2π , entonces, por definición de la *finesse* \mathcal{F} :

$$\mathcal{F} = \frac{2\pi}{\epsilon} \quad (1.37)$$

$$\mathcal{F} = \frac{\pi F^{\frac{1}{2}}}{2} \quad (1.38)$$

o por la definición de F :

$$\mathcal{F} = \frac{\pi R^{\frac{1}{2}}}{(1-R)} \quad (1.39)$$

Donde la *finesse* depende de la reflectividad de las placas paralelas del interferómetro. Se observa en las Figuras 1.9 y 1.10, la relación de la *finesse* y del factor de contraste con respecto a la reflectividad de las placas, que tienen un comportamiento similar.

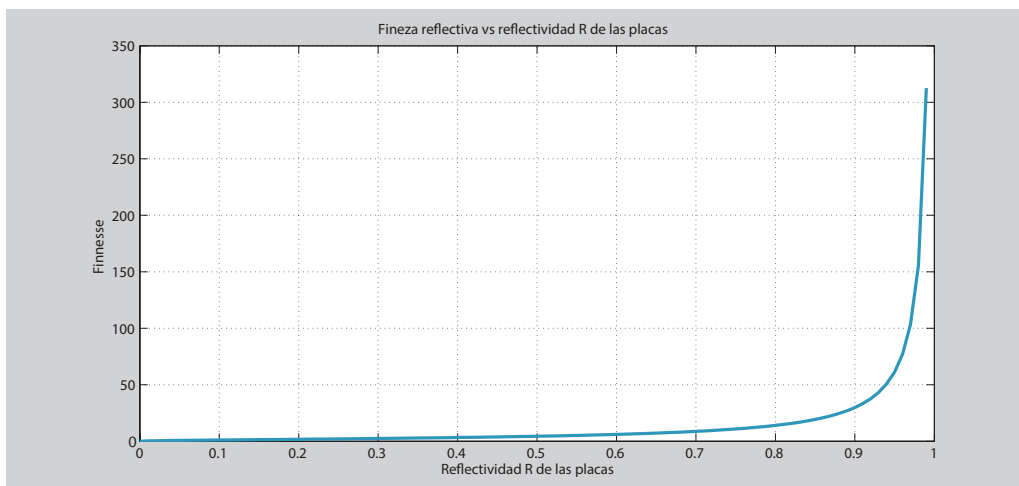


Figura 1.9: Finesse vs reflectividad de las placas. Gráfica realizada por Zimri Marsilli.

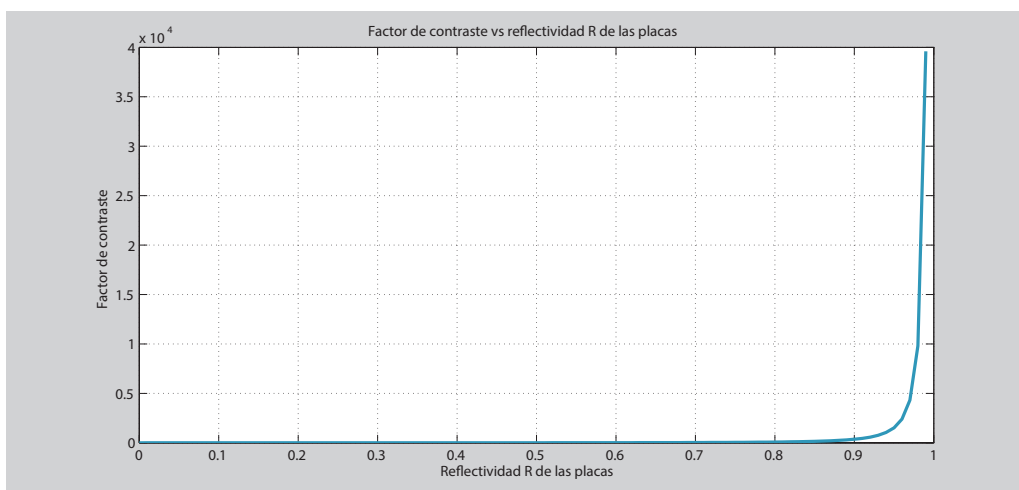


Figura 1.10: Factor contraste vs reflectividad de las placas. Gráfica realizada por Zimri Marsilli.

1.1.3 Espectrometría de Fabry-Pérot

El interferómetro de Fabry-Pérot es de importancia práctica, fundamental para aplicaciones modernas, usado principalmente para examinar la estructura fina de las líneas espectrales. Al utilizar la grabación fotoeléctrica directa de la luz, para detectar los anillos de interferencia que produce un interferómetro Fabry-Pérot, instantáneamente se puede utilizar para la espectrometría. Esta técnica fue introducida por Jacquinot y Dufour en 1948 (Vaughan, 1989). Mediante la variación de la separación de las placas, se escanean los anillos de interferencia a través de una abertura fija de un detector, el cual debe estar colocado en el centro del patrón de interferencia, en el plano focal del sistema óptico. Como número experimental de referencia, es conveniente calcular el diámetro del primer anillo de interferencia, para una separación de las placas dada. Los componentes auxiliares al interferómetro Fabry-Pérot deben ser de muy alta calidad ya que los rayos, a través del área de las placas del FP, deben ser llevados a un enfoque preciso.

El instrumento PUMA tiene un arreglo de lentes para colimar la luz del telescopio y formar los patrones de interferencia, variando el espacio entre sus placas. En el patrón de interferencia, como función de los parámetros del dispositivo, se pueden medir valores de longitudes de onda muy precisos. Las limitaciones de alcanzar una *finesse*, son debido a las imperfecciones de sus placas, producto de su manufactura. Por la variación entre la planicidad de las placas, aumenta las variaciones de separación entre placas y por lo tanto del retraso óptico inducido.

La detección de luz del equipo debe ser capturada por algún mecanismo de registro (e.g. las cámaras CCD o placas fotográficas). Estos medios detectan la interacción de los fotones con el medio de registro, con el objeto de tener un modo de almacenamiento de la información en un ordenador para su utilización, procesamiento y su análisis. Las imágenes de la lente de la cámara, son moduladas por los anillos de interferencia de Fabry-Pérot, sobre un detector de área que es operado como una matriz de píxeles individuales.

El poder de captación de la luz del detector (del sistema óptico) se caracteriza por el *étendue*, que es el producto del área de la zona de origen con el ángulo de luz recogida, la cual es proporcional al brillo de la fuente. Sin embargo en el caso de los espectros atómicos pueden ocurrir fenómenos que alteren la captación de luz, como la auto-absorción, alteraciones en el campo eléctrico y el ensanchamiento Doppler. El dispositivo de carga acoplada (CCD), es un detector de óxido de silicio en estado sólido, de muy alta sensibilidad al color rojo y cerca del infrarrojo. Está conformado por una matriz bidimensional de cientos o miles de fotodiodos. El enfriamiento disminuye las cargas inducidas del CCD y mejora la detección de imágenes muy débiles.

El poder de resolución determina la capacidad de un instrumento óptico para distinguir entre dos estructuras espectrales muy próximas y es un método de comparación entre los diferentes instrumentos espectroscópicos. El poder de resolución se puede derivar de acuerdo con el tratamiento del rayo. Las pérdidas de energía son pérdidas debidas a la reflexión de los espejos. El poder de resolución óptica del instrumento Fabry-Pérot cuando dos líneas intensas son resueltas es:

$$\frac{\lambda}{\Delta\lambda} = 0,97m\mathcal{F} \quad (1.40)$$

Para alcanzar un gran poder de resolución, es necesario que las superficies de los interferómetros tengan alta calidad en sus superficies planas y tengan alta precisión del paralelismo de las placas entre sí. El poder de resolución básicamente dice que, para un valor dado de la separación de las placas, puede incrementarse mientras la reflectancia se acerque a la unidad. Teniendo como dependencia un alto coeficiente de reflectividad y el brillo de la fuente. Por lo que las placas son trabajadas con materiales ópticos de alta calidad y las superficies reflectantes deben estar recubiertas uniformemente con películas de plata, aluminio u oro.

La calidad de las superficies reflectantes es de vital importancia en la interferometría de Fabry-Pérot, por lo que películas dieléctricas han sido empleadas, incluyendo mejoras en el acabado de la superficie. En 1950 Kuhn y Wilson con la técnica por evaporación en un alto vacío, acondicionaron películas de diferente espesor (Vaughan, 1989). Actualmente se pueden fabricar películas dieléctricas de capas múltiples de muy alta reflectividad, sobre placas de gran diámetro y con bajas pérdidas de luz. Las especificaciones importantes para la elección de placas reflejantes y los filtros deben ser: la reflectividad en diferentes longitudes de onda, las pérdidas por absorción, la uniformidad y planicidad de las superficies, la humedad, la temperatura, etc.

El intervalo espectral libre, es la relación entre la diferencia del número de onda entre dos líneas espectrales, que se superponen en los órdenes sucesivos para la separación de las placas.

El interferómetro de Fabry-Pérot tiene una alta eficiencia como un filtro dependiente del ángulo, siendo muy sensible y se puede ajustar variando la separación entre sus placas. Se puede colocar una lente de colimación en el plano focal, para la fuente de luz, para presentarse como un haz paralelo ante el interferómetro, presentándose al final en el plano de la imagen la formación del patrón de interferencia en forma de anillos brillantes circulares, que corresponde a la simetría cilíndrica de las dos superficies paralelas y de los elementos sobre el eje óptico, obtenidos por la superposición y adición coherente de los haces reflejados sucesivamente. Se coloca una lente de colimación que reúne los haces de transmisión en el plano focal, modulada por la interferencia. Si la fuente cruza con el eje óptico, determinado por la lente de colimación y las superficies ópticas del interferómetro, se formarán los anillos concéntricos de forma automática (Figura 1.11).

Se puede evitar el primer lente que funciona como colimador, ya que si de la fuente de luz que ilumina al interferómetro encuentra a dos o múltiples rayos con el mismo ángulo de incidencia, para una región del plano focal, y cumplen con la condición de interferencia constructiva, resulta una suma de las contribuciones de toda la fuente y el interferómetro hace paralela su propia luz, esto se puede considerar para todos los puntos de la fuente. La función de las lentes de colimación servirá entonces, para aumentar la iluminación del resultante patrón de interferencia. A falta de la primer lente de colimación, las fuentes débiles y pequeñas, tendrán como resultado una imagen desenfocada, pérdida de iluminación y, por consecuencia, reducción considerable de la intensidad de los anillos.

Como se estudió anteriormente, los anillos se forman de la transmisión y su nitidez aumenta con el aumento de la reflectividad de las placas, expresándose como la relación entre la distancia de anillos sucesivos y la anchura a media intensidad del anillo. Para una longitud de onda ligeramente diferente a un haz de luz cuasimonocromático de longitud de onda λ_0 , se puede formar un sistema similar de anillos de interferencia.

Para el procedimiento de calibración, se deben llevar a cabo una serie de exposiciones a diferentes separaciones del Fabry-Pérot (FP). En el caso del instrumento PUMA, se utilizan lámparas de helio, neón e hidrógeno, teniendo las líneas espectrales estrechas, simétricas y aproximadamente del mismo brillo. Las exposiciones deberían ser suficientes para que, con las grabaciones de los anillos, se puedan medir las longitudes de onda y compararse con las líneas de los objetos astronómicos y se puedan obtener sus velocidades, debidas al corrimiento Doppler.

1.1.4 Técnicas de instrumentación

El uso del interferómetro de Fabry-Pérot ha contribuido en la ampliación de los conocimientos de la física moderna y en la comprensión de la naturaleza de la luz. Las primeras aportaciones con el instrumento, en la investigación astrofísica, las realizaron evidentemente Fabry y Pérot. En donde llevaron a cabo un trabajo muy largo sobre medidas de longitud de onda absoluta, con el estudio de las líneas espectrales del sol y más tarde, en 1911 Fabry y Buisson, al estudiar la nebulosa de Orión (Vaughan, 1989), iniciando la exploración del estudio físico de los astros con el distinguido dispositivo.

Con un buen ajuste óptico (ya que la fuente primaria y el carácter espectral no tienen control del observador), el interferómetro de FP tiene una alta resolución, buen contraste y proporciona un gran factor de rango espectral libre, por lo que el interferómetro es capaz de recibir toda la luz

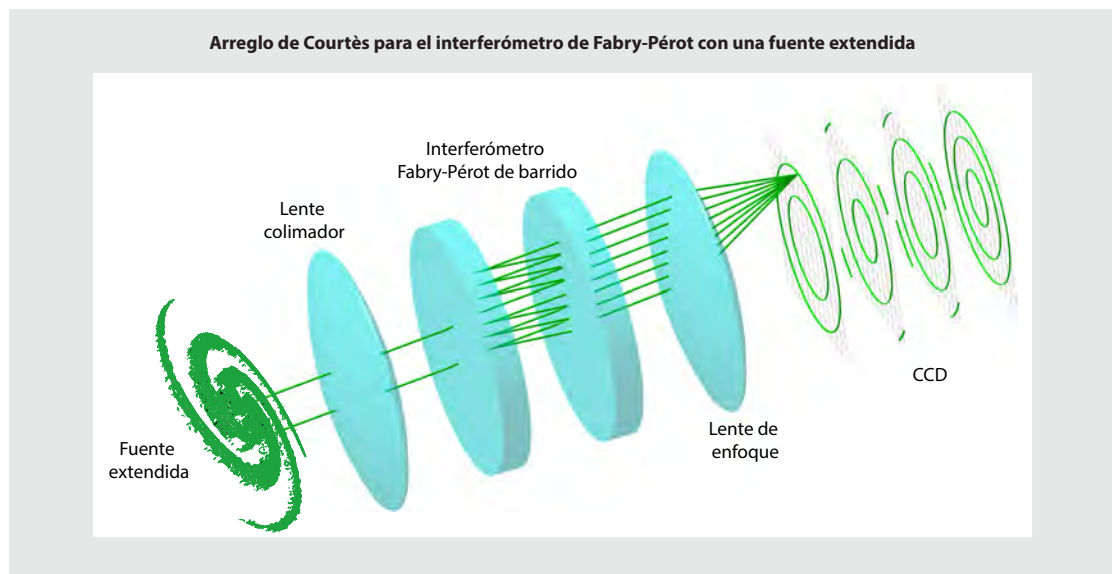


Figura 1.11: Arreglo del interferómetro de Fabry-Pérot con una fuente extendida y la transmisión y reflexión de los haces de luz. *Imagen realizada por Zimri Marsilli.* Este arreglo es conocido como la caja de herramientas de Courtyès (Courtyès et al., 1995).

de un telescopio de grandes dimensiones. La resolución espacial y espectral de alta frecuencia se requieren para el mapeo de las velocidades de los objetos que emiten las líneas astronómicas.

La importancia del interferómetro de barrido Fabry-Pérot, para la astronomía y la astrofísica, es su aplicación en el estudio de las fuentes fuera del sistema solar, como nebulosas y galaxias. Para lograr este análisis, se requiere la formación de imágenes con luz filtrada y una estabilidad térmica en el dispositivo. Los filtros utilizados son diseñados para pasar sólo un determinado ancho de banda. Los resultados a diferentes bandas de frecuencias revelan las diferentes condiciones cinemáticas y dinámicas de los diferentes objetos estudiados.

La medición de una línea de emisión (o alguna parte de ella), se puede realizar mediante el escaneo con el interferómetro de Fabry-Pérot y, para obtener una mayor resolución, se puede disponer de una configuración de varios interferómetros Fabry-Pérot como lo hizo Ramsay (1970). Loughhead (1978) también describió un instrumento para su uso en la espectroscopía solar, el cual consistía en un interferómetro FP controlado por ordenador para varios propósitos. Las mediciones precisas de posición de las líneas espectrales pueden proporcionar desplazamientos Doppler y, por tanto, las componentes de velocidad; la interpretación de anchura y perfil de la línea pueden dar la temperatura, la presión y otras condiciones sobre la atmósfera solar.

El arreglo del interferómetro Fabry-Pérot para su uso en el telescopio, incluye una disposición adecuada de instrumentos colocados en el foco *Cassegrain* y, por consiguiente, en el plano focal. Varias técnicas se han desarrollado para las mediciones extraterrestres de banda ancha y la espectroscopía de las líneas de absorción, con ayuda de instrumentos auxiliares dispersantes (prismas, grismas y rejillas echelle, con alta dispersión angular), para la forma de espectros de dispersión. El interferómetro puede tener un barrido piezoeléctrico, en un rango espectral, ajustando la velocidad de barrido de longitud de onda del interferómetro y sintonizándolo con los otros elementos ópticos. El interferómetro se puede establecer para la observación de perfiles sucesivos, de una fuente de alguna línea estrecha o la observación del continuo. Los filtros de banda angosta son suficientes para eliminar la superposición de órdenes.

1.1.4.1 El instrumento PUMA

La particularidad especial del instrumento PUMA es el uso de un interferómetro Fabry-Pérot de barrido, el cual realiza exposiciones en todo el rango espectral libre y con alta fineza (Rosado et al., 1995), para el estudio de fuentes extendidas como el estudio de la nube de magallanes (Le Coarer et al., 1993). Además del FP el instrumento cuenta con un sistema mecánico con diferentes instrumentos ópticos incluidos, entre ellos el sistema de control de la distancia entre las placas del interferómetro, un detector CCD, un software de adquisición de datos y un sistema de control y comunicación con todas los sistemas del PUMA (este sistema es el estudio principal de este trabajo).

El interferómetro que está instalado en el PUMA es fabricado por Queensgate Instruments, Ltd. El barrido del FP se realiza por medio de pequeños desplazamientos de piezoeléctricos, colocados entre las placas paralelas del Fabry-Pérot, controlados por un CS100 debido a diferencias de voltaje en sus puentes capacitivos. Con este sistema de control se mantiene la separación de las placas del FP, con una resolución de una longitud de onda por cada cien partes (Bland and Tully, 1989).

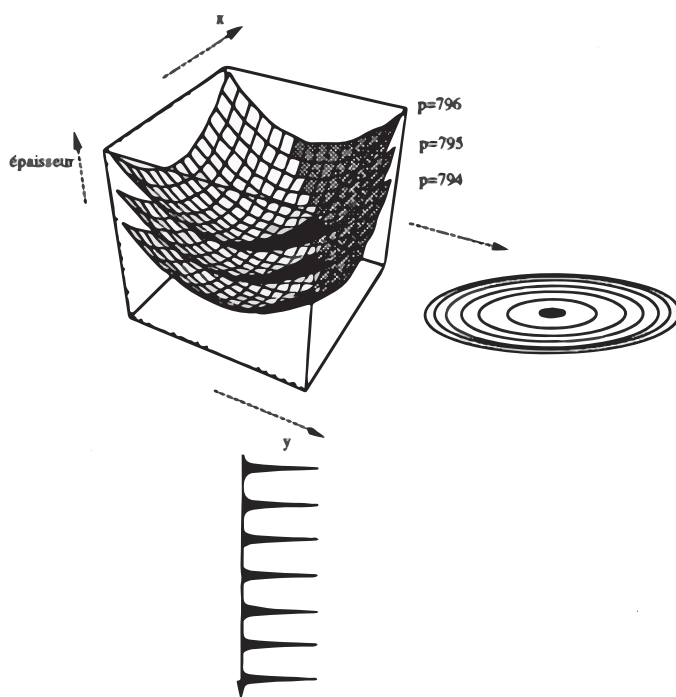


Figura 1.12: Representación de los anillos de interferencia, producidos por los máximos de transmisión representados a través de superficies de corrección de fase constante. El cubo donde se encuentran estas superficies tiene dimensiones de las coordenadas en el plano x e y por el espaciamiento λ del interferómetro de Fabry-Pérot de barrido. Referencia (Le Coarer, 1992).

Se realiza la corrección de longitud de onda (a determinado orden y ángulo del rayo incidente en las placas paralelas), es decir la corrección de fase para el interferómetro Fabry-Pérot. Los puntos máximos de transmisión, resultantes de los canales de barrido, son dependientes de la distancia

de los espaciamientos que realiza el FP a lo largo de su camino óptico. Esta calibración se realiza a partir de cubos de superficies hiperbólicas de fase constante (Figura 1.12). Las ecuaciones modelo para la corrección las presentan Bland y Tully (Bland and Tully, 1989), en donde se determina una relación que sólo depende de las características instrumentales del sistema.

El interferómetro de Fabry-Pérot del instrumento PUMA obtiene dos tipos de cubos de datos, uno se construye de los interferogramas del objeto y el otro de una fuente difusa monocromática. De cada interferograma observado y obtenido se podrán calcular las velocidades radiales, por medio de los movimientos en los anillos de interferencia. Es decir que cada posición de píxel intersectado por un anillo, es una velocidad que se podrá medir.

El software de adquisición y reducción de datos del instrumento PUMA, recolecta los canales del barrido del FP y obtiene el cubo de canales para su posterior análisis. Este software permite ajustar los tiempos de exposición y el procesamiento de las imágenes a partir de un intervalo definido de píxeles. Por medio de algoritmos estadísticos se analizan los parámetros de dispersiones lineales, velocidades radiales e intensidad del continuo.

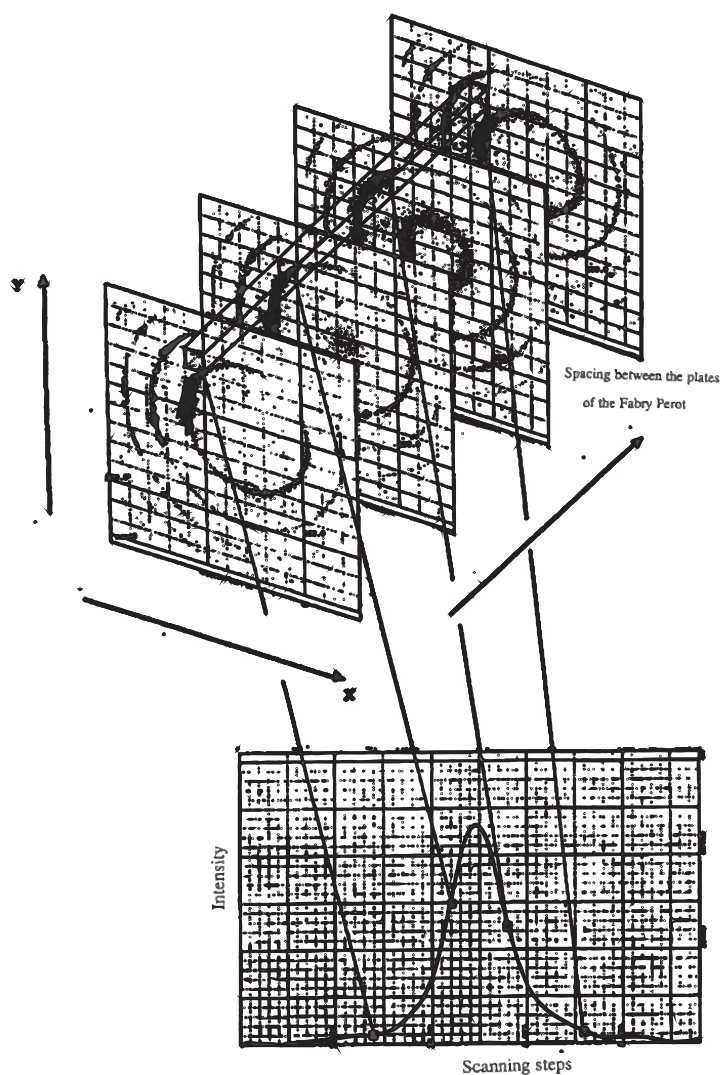


Figura 1.13: Análisis de los píxeles de los anillos de los máximos de transmisión, obtenidos del cubo de interferogramas. Referencia (Le Coarer, 1992).

CAPÍTULO

2

Objetivos y alcances

Este capítulo detalla el objetivo general y los objetivos específicos del proyecto, así como sus alcances y justificación.

Índice

| | |
|--|----|
| 2.1 Problema y motivación | 24 |
| 2.2 Objetivo general | 24 |
| 2.3 Objetivos particulares | 25 |
| 2.4 Alcances | 25 |
| 2.5 Contribución de esta tesis | 26 |



Figura 2.1: Instituto de Astronomía sede en Ensenada. *Fotografía tomada por Zimri Marsilli.*

2.1 Problema y motivación

El instrumento PUMA es un instrumento astronómico que inició su operación en el año 1994, en el telescopio de 2.12 m del OAN-SPM. El instrumento cuenta con un sistema de control conformado por un microcontrolador marca *Octagon*, modelo 5080 μ PC (Figura 2.2), y su hardware adicional.

Dicho sistema tiene alrededor de 20 años de funcionamiento y debido a la obsolescencia de los sistemas electrónicos, se volvió necesario su modernización.

2.2 Objetivo general

El principal objetivo del proyecto es la modernización del sistema de control electrónico del espectrógrafo integral de campo PUMA, que está en funcionamiento en el Observatorio Astronómico Nacional en San Pedro Mártir Baja California (OAN-SPM), con sede en Ensenada (Figura 2.1).

Es decir, el proyecto puntea el desarrollo de un sistema de control, capaz de manipular los dispositivos dentro del instrumento y mejorar el desempeño del sistema original.

2.3 Objetivos particulares

Los objetivos específicos para implementar el nuevo sistema de control y mejorar el funcionamiento del espectrógrafo integral de campo PUMA, del Observatorio Astronómico Nacional son:

Implementar el sistema de control, utilizando el microcontrolador de placa reducida modelo BL4S200.

Realizar la programación de las funciones para la manipulación de actuadores y la detección de señales provenientes de los sensores del instrumento PUMA. Los actuadores a controlar son: dos carros de movimiento, una rueda de filtros y un brazo de prismas.

Realizar la programación de las funciones para la manipulación de las señales del controlador CS100. Las señales a controlar son: los ejes del interferómetro Fabry-Pérot de barrido; así como la detección de lazo abierto del puente de capacitancia y cerrar el lazo.

Realizar la programación de las funciones para el encendido de las lámparas de calibración del instrumento PUMA.

Implementar comunicación Ethernet a través de un protocolo TCP/IP. Establecer una red local (LAN), para el desarrollo de pruebas de la programación e instalación del hardware del proyecto, e implementarlo en la red del observatorio para la comunicación entre el instrumento y el cuarto de control.

Instalación y pruebas. Realizar la instalación y las pruebas necesarias en el telescopio de 2.12 m del OAN-SPM.

2.4 Alcances

El proyecto planea el desarrollo de un sistema de control electrónico, capaz de sustituir la electrónica existente en el instrumento PUMA. Se espera que este tipo de tecnología en microcontroladores y módulos electrónicos, mejore el desempeño del instrumento y proporcione más confiabilidad en su uso para la obtención de datos astronómicos.

Las mejoras consistirán en modernizar la integración, simplificación y comunicación remota del sistema. También se pretende mejorar el movimiento de los elementos móviles, garantizar su correcto posicionamiento y obtener un mejor control sobre el FP y su sistema de barrido.

2.5 Contribución de esta tesis

El trabajo desarrollado en este proyecto y descrito en esta tesis, presenta una manera de alta integración de componentes, para un sistema de control aplicado, mediante el uso de un microcontrolador de placa reducida. Esta tecnología se aplicará a la instrumentación astronómica, para la comprensión y estudio de los cuerpos celestes.

También presenta una aplicación de la ingeniería mecatrónica hacia el campo de la instrumentación astronómica, donde se utilizan simultáneamente los conocimientos y desarrollos de la electrónica, mecánica, óptica y de programación.

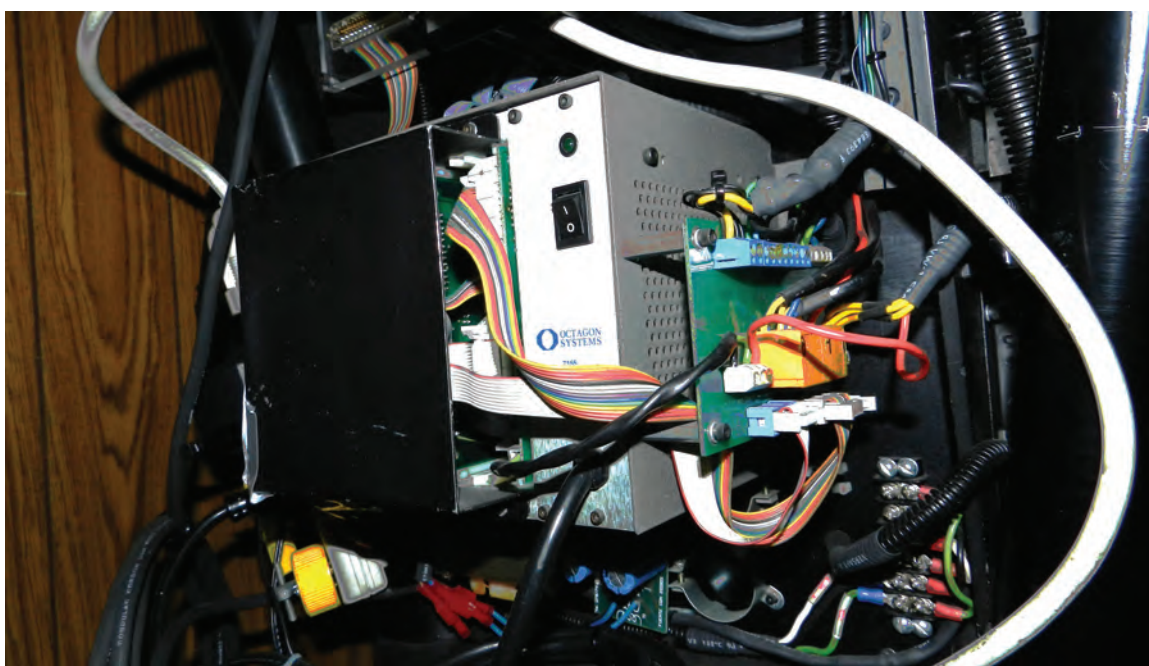


Figura 2.2: Microcontrolador Octagon utilizado en el primer sistema de control del instrumento PUMA.
Fotografía tomada por Zimri Marsilli.

CAPÍTULO

3

Describiendo al sistema

Este capítulo detalla la descripción del sistema, la descripción de sus subsistemas, así como la interfaz de usuario. Estos elementos son fundamentales para la concepción del nuevo diseño del sistema de control.

Índice

| | | |
|------------|----------------------------|-----------|
| 3.1 | Diseño optomecánico | 28 |
| 3.2 | Sistemas de control | 36 |
| 3.3 | Interfaz de usuario | 40 |

El instrumento PUMA es un sistema compuesto de varios subsistemas ópticos, mecánicos y electrónicos, integrados en conjunto para realizar las observaciones astronómicas. El instrumento se coloca en el foco Cassegrain del telescopio de 2.12 m, en su configuración de $f/7,5$.

El uso y operación del instrumento se realiza desde una computadora remota, localizada en el cuarto de observación del telescopio. Con esta computadora es posible controlar los dispositivos del instrumento para realizar las observaciones astronómicas.

En lo que sigue del capítulo se describen los diferentes subsistemas del instrumento PUMA, que tienen que ser considerados para el nuevo diseño del sistema electrónico de control, objetivo principal de este trabajo.

3.1 Diseño optomecánico

El instrumento PUMA (Figura 3.1), cumple las consideraciones de Kasunic (Kasunic, 2015), para ser un sistema de ingeniería optomecánica en su diseño y ensamble, con prevención a flexiones de su estructura, de su cinemática de sus componentes y de sus componentes térmicos; específicamente es la integración de varios elementos mecánicos y ópticos que interactúan con la luz. El sistema que controla la interacción de los dispositivos es una computadora Octagon, que sincroniza partes automatizadas y el elemento de paralelización del interferómetro Fabry-Pérot es el controlador CS100.

La trayectoria de la luz proveniente del telescopio debe pasar por una etapa de filtrado, conformado por una rueda de filtros mecanizada, capaz de cambiar hasta ocho filtros optimizando la observación del objeto y aislando la región espectral de interés. El haz de luz es desviado 60° por medio de un colimador (teniendo como referencia la dirección de la trayectoria); el haz colimado es direccionado normalmente a las placas del interferómetro Fabry-Pérot y finalmente las imágenes son enfocadas por una cámara, que actúa como reductor focal para la detección de los canales espectrales con el dispositivo de carga acoplada (CCD).

El interferómetro de Fabry-Pérot puede ser retirado del camino óptico, para que la trayectoria del haz conserve el libre paso de la luz y se obtengan imágenes directas del objeto observado. Alguno de los filtros acomodado en la rueda de filtros puede ser fácilmente evitado del sistema, sin herramientas ni algún movimiento del telescopio, ya que de la rueda por lo menos se extrae un filtro y se cede un espacio vacío en su lugar.

Exponiendo la calibración de las imágenes y la corrección de valores del controlador CS100, se debe ubicar la unidad de calibración móvil (un espejo inclinado a 45°), para interponer la luz proveniente del telescopio y transportar un haz de luz telecéntrico, derivado de un arreglo de lentes y un difusor, cuya fuente puede ser alguna de los tres tubos de descargas de dos gases raros, helio y neón o de hidrógeno, el haz se dirige al colimador y del colimador al Fabry-Pérot. Ulteriormente llega a un arreglo de cuatro prismas (mediante un brazo también automatizado), obteniendo cuatro patrones de interferencia de los espectros atómicos.

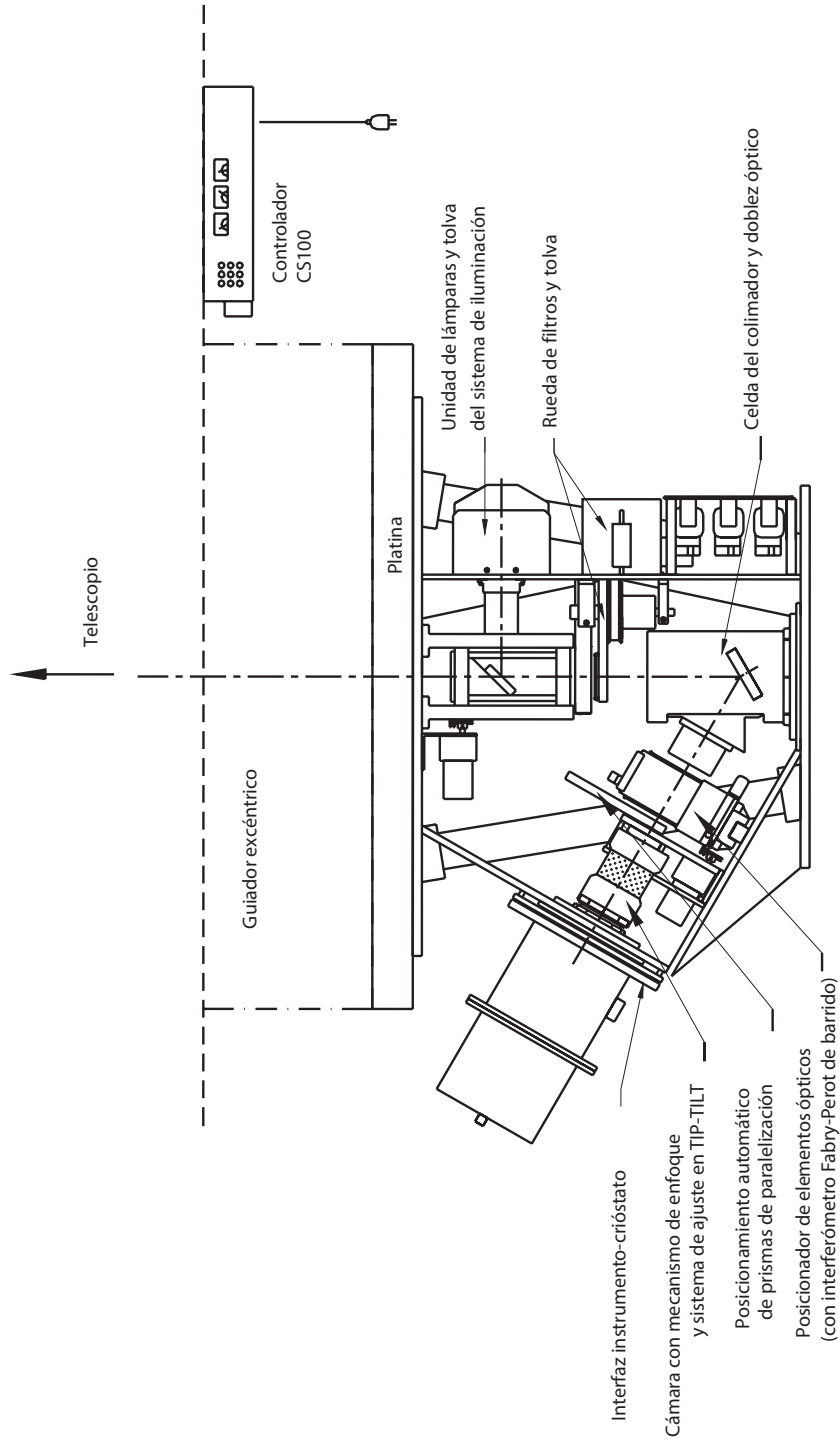


Figura 3.1: Sistema optomecánico del instrumento PUMA. Referencia (Langarica and Tinoco, 2005).

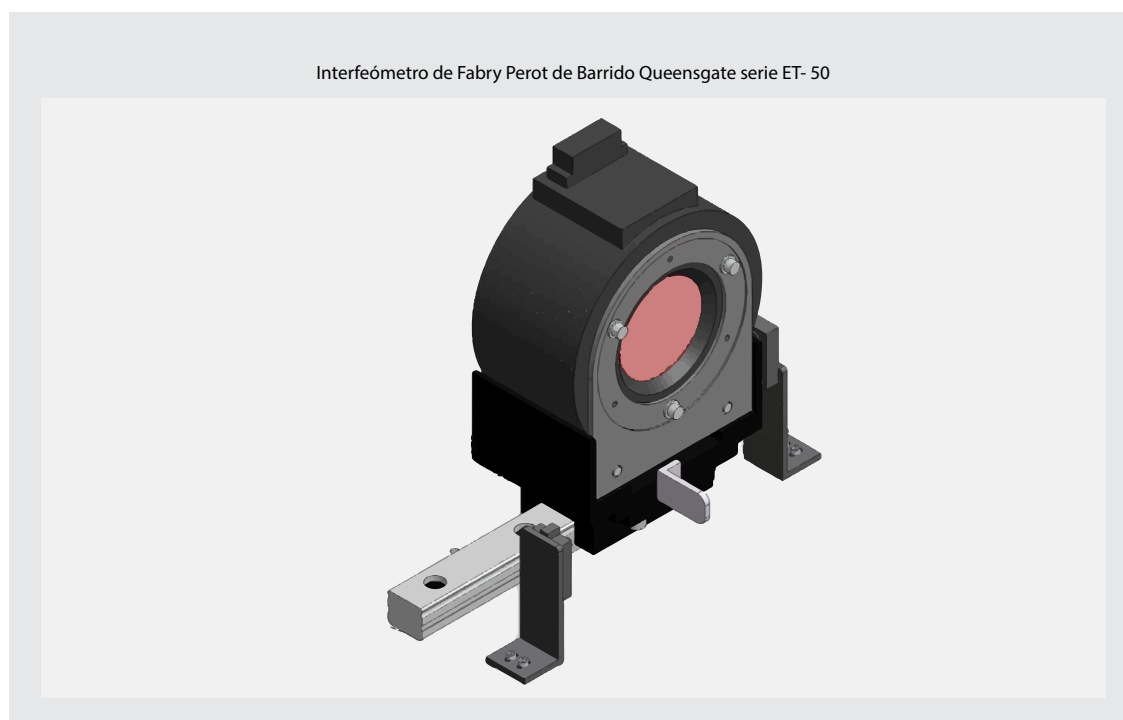
3.1.1 Interferómetro Fabry-Pérot

Figura 3.2: Interferómetro Fabry-Pérot serie ET. Referencia (Langarica and Tinoco, 2005).

El interferómetro Fabry-Pérot utilizado en el instrumento PUMA es de la serie ET-50 de interferómetros, el cual fue desarrollado por Queensgate Instruments Limited (Figura 3.2), es controlado por el controlador CS100 para el estudio y la resolución de problemas espectrométricos, así como para la detección de fuentes débiles extragalácticas de interés astronómico (Queensgate-Instruments, 1981).

El interferómetro Fabry-Pérot trabaja con sensores de desplazamiento capacitivo (con micrometros de resolución), los cuales detectan cambios en el paralelismo, corrigiendo estas variaciones con la disposición de transductores piezolétricos y así, por medio de pequeños desplazamientos, mantienen la estabilización de las placas del Fabry-Pérot. Los cinco condensadores están formados por evaporación de oro y se encuentran sobre la parte interna de las placas del interferómetro FP (Figura 3.3).

El interferómetro de Fabry-Pérot se localiza en la placa estructural de posicionamiento para elementos ópticos; está colocado sobre un carro de posicionamiento lineal, con transmisión de cadena, catarinas y un motor de 12 VDC con caja de reducción de 65.5:1. Por medio de sensores de proximidad de final de carrera, se detectan las posiciones extremas de la unidad.

El teorema de Nyquist-Shannon (Petersen and Middleton, 1962) fundamenta una condición de muestreo para obtener información íntegra de una señal continua a partir de una señal discreta, en un tiempo de ancho de banda definido. Dado que una señal discreta puede corresponder a varias señales continuas, se establece que para muestrear fielmente una señal se requiere una frecuencia de muestreo igual o mayor al doble de la frecuencia de la señal original.

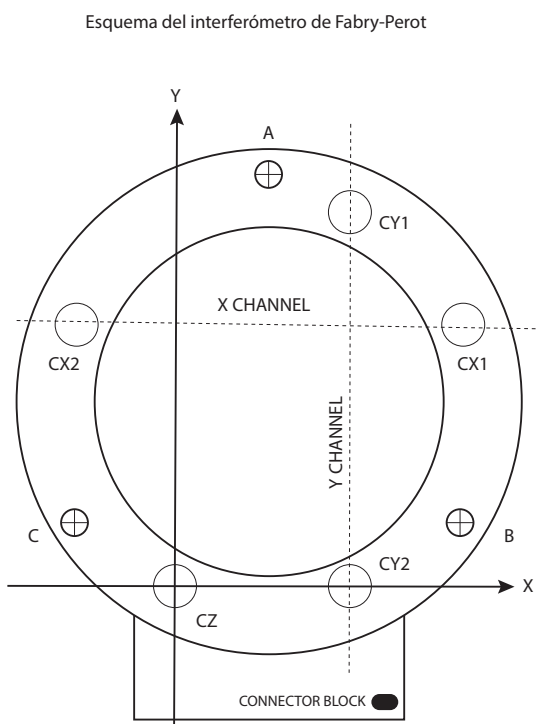


Figura 3.3: Entre las dos placas los transductores piezoeléctricos son los indicados por las letras A, B y C; y los sensores capacitivos son los indicados por CX1, CX2, CY1, CY2 y CZ. Referencia (Queensgate-Instruments, 1981).

El teorema es aplicable a funciones dependientes del tiempo, pero tiene uso extendido al muestreo multi-variable, como son los canales obtenidos del interferómetro Fabry-Pérot en el detector CCD, en el cual cada imagen bidimensional obtenida, se puede representar como una matriz compuesta por píxeles. Por lo que la imagen puede ser indistinguible si la resolución de muestreo (o densidad de píxeles) es inadecuado. Cumpliendo con el teorema del muestreo, el instrumento PUMA tiene un número de pasos de barrido de canales para un muestreo del rango espectral, del doble valor de la finesse (Cuadro 3.1).

Cuadro 3.1: Para las longitudes de H alpha y [OIII] el interferómetro Fabry-Pérot recibe estas características, nótese que el número de pasos es el doble del valor de la finesse. Referencia(Langarica and Tinoco, 2005).

| Características | H α | [OIII] |
|---------------------------|------------|-----------|
| Orden de interferencia | 330 | 432 |
| Intervalo espectral libre | 2.86 [Å] | 11.57 [Å] |
| Resolución espectral | 0.43 [Å] | 0.58 [Å] |
| Finesse de reflexión | 24 | 10 |
| Barrido típico | 48 pasos | 20 pasos |

Puede haber variaciones en el paralelismo de las placas del Fabry-Pérot por cambios de temperatura, presión y humedad, los cuales provocan una alteración en la constante dieléctrica. El FP utiliza un flujo de gas nitrógeno seco, para evacuar la humedad. La eficiencia del interferómetro Fabry-Pérot, para la resolución de diversos problemas espectrométricos, está directamente relacionada con la perfecta paralelización entre sus placas.

3.1.2 Rueda de filtros

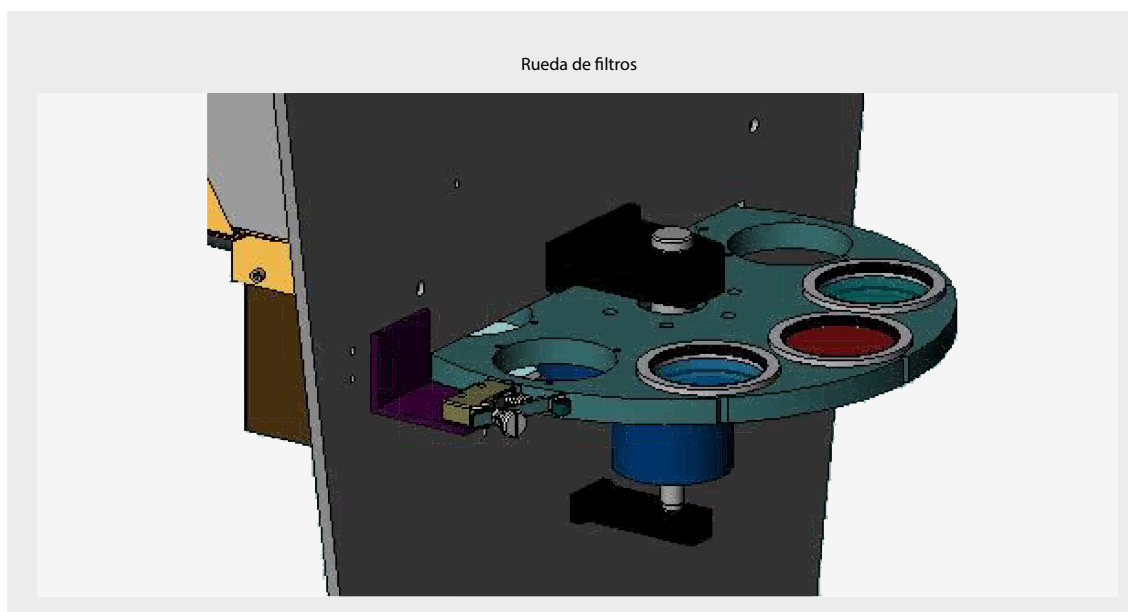


Figura 3.4: Rueda de ocho filtros instalada en el instrumento PUMA. Referencia (Langarica and Tinoco, 2005).

Los filtros ópticos son para seleccionar el rango del espectro en el que se van a realizar las observaciones y formar imágenes espectroscópicas, es decir, solamente transmiten la luz de cierta longitud de onda dependiente del filtro.

Para la técnica de interferometría Fabry-Pérot se requieren filtros de banda estrecha. Teniendo en cuenta una región de longitud de onda de interés, la longitud de onda central del filtro (λ_{cw}) y el ancho de banda deseada del filtro ($\Delta\lambda$) (Cuadro 3.2).

El instrumento PUMA posee una rueda de filtros. Esta rueda es un ensamble de movimiento giretorio indexado, con transmisión por banda dentada, poleas y motor a 12 V DC con reducción (65.5:1). La rueda tiene ocho filtros intercambiables con sus respectivos portafiltros; con dimensiones de dos pulgadas de diámetro y 6 mm de espesor cada uno. La posición de los filtros se encuentra en el foco del telescopio. Los rayos que pasan a través de ellos deben proceder de un sistema normal al filtro, esto es para rechazar las longitudes de onda correspondientes a otras diferentes antes de entrar en el tren óptico. Por otra parte, una rueda de filtros no es un dispositivo pequeño y puede ser voluminoso, por lo tanto requiere de optimización al colocarse en el camino óptico (Figura 3.4).

En el sistema se deja una posición vacía, la cual es para la imagen directa (sin filtro) para permitir el libre paso de la luz; dos posiciones opcionales están disponibles. El mecanismo de giro de la rueda es capaz de girar en sentido horario o en sentido antihorario para el posicionamiento preciso del filtro a utilizar. La rueda de filtros tiene cortes en forma de cuña en su perímetro, que actúan como estaciones de indexación mecánicas y están diseñadas para proporcionar una alineación precisa con alta repetibilidad. Cada montura de filtros está diseñada para adaptarse a cualquiera de los ocho filtros disponibles, capaz de girar para colocar cada filtro de forma independiente. Los filtros están numerados del cero al siete.

Cuadro 3.2: Cada filtro tiene características de longitud de onda central (CWL) y ancho de banda (BW) para un ángulo de incidencia (θ). Referencia (Langarica and Tinoco, 2005).

| Posición | Identificación de la rueda | Filtro | CWL [Å] | BW [Å] | θ |
|----------|----------------------------|------------|---------|--------|----------|
| 0 | 0 | Sin filtro | - | - | - |
| 1 | 5007 | [OIII] | 5007 | 10 | 3.6 |
| 2 | 5374 | Sin filtro | - | - | - |
| 3 | 6563 | H α | 6570 | 20 | 4.0 |
| 4 | 6584 | [NII] | 6584 | 10 | 3.6 |
| 5 | 6717 | [SII] | 6717 | 10 | 4.0 |
| 6 | 6731 | [SII] | 6731 | 10 | 3.8 |
| 7 | Sin identificación | [SII] | 6720 | 20 | 3.8 |

3.1.3 Unidad de calibración

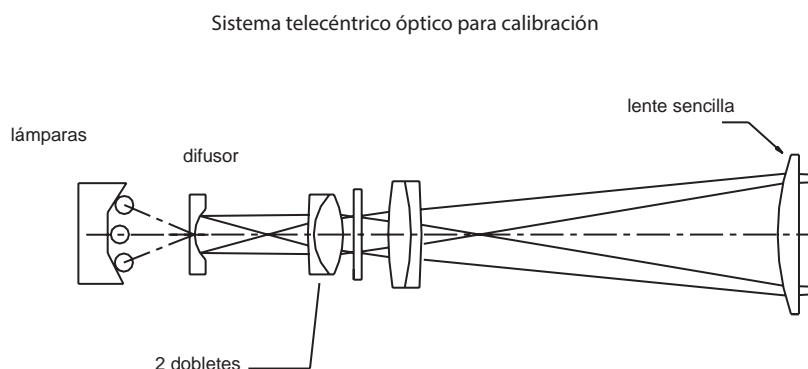


Figura 3.5: Esquema telecéntrico del sistema de calibración. Referencia (Langarica and Tinoco, 2005).

Durante las observaciones se requieren hacer calibraciones con el sistema, para tener un patrón de anillos de referencia y compararlos con el espectro real observado. La emisión lumínica de

las lámparas consta de líneas espectrales discretas de cierta longitud de onda definida, anchura espectral e intensidad relativa.

El instrumento PUMA cuenta con tres lámparas (Figura 3.6): una de hidrógeno (H), una de helio (He) y una de neón (Ne), junto con un sistema telecéntrico y un espejo en la unidad móvil, para llevar la luz de las lámparas al detector. La luz de cada una de las tres lámparas, tiene su propia línea de emisión y ciertas líneas asociadas. Las lámparas espectrales para la calibración producen una iluminación uniforme (Cuadro 3.3). No hay ningún cambio sistemático en las velocidades debido a los componentes telecéntricos utilizados para la calibración de la longitud de onda.

El espejo móvil debe ser colocado para desviar la luz de las lámparas al camino óptico del telescopio dentro del instrumento. Éste es otro mecanismo de posicionamiento lineal con transmisión de cadena, catarinas y motor con reducción similar al que posiciona el FP (Figura 3.5). Esta unidad de calibración motorizada puede ser controlada para introducirla o sacarla del camino óptico del instrumento, al momento del inicio de la calibración o al comenzar la observación.

Cuadro 3.3: Descripción de las lámparas de calibración. Referencia (Langarica and Tinoco, 2005).

| Lámpara | Línea de emisión | Línea asociada |
|-----------|------------------|---|
| Hidrógeno | 6562.78 [Å] | H α (λ 6563) Å [NII] (λ 6584) Å] |
| Helio | 5015.678 [Å] | [OIII] (λ 50007)]Å |
| Neón | 6717.048 [Å] | [SII] (λ 6717y λ 6731)] Å |

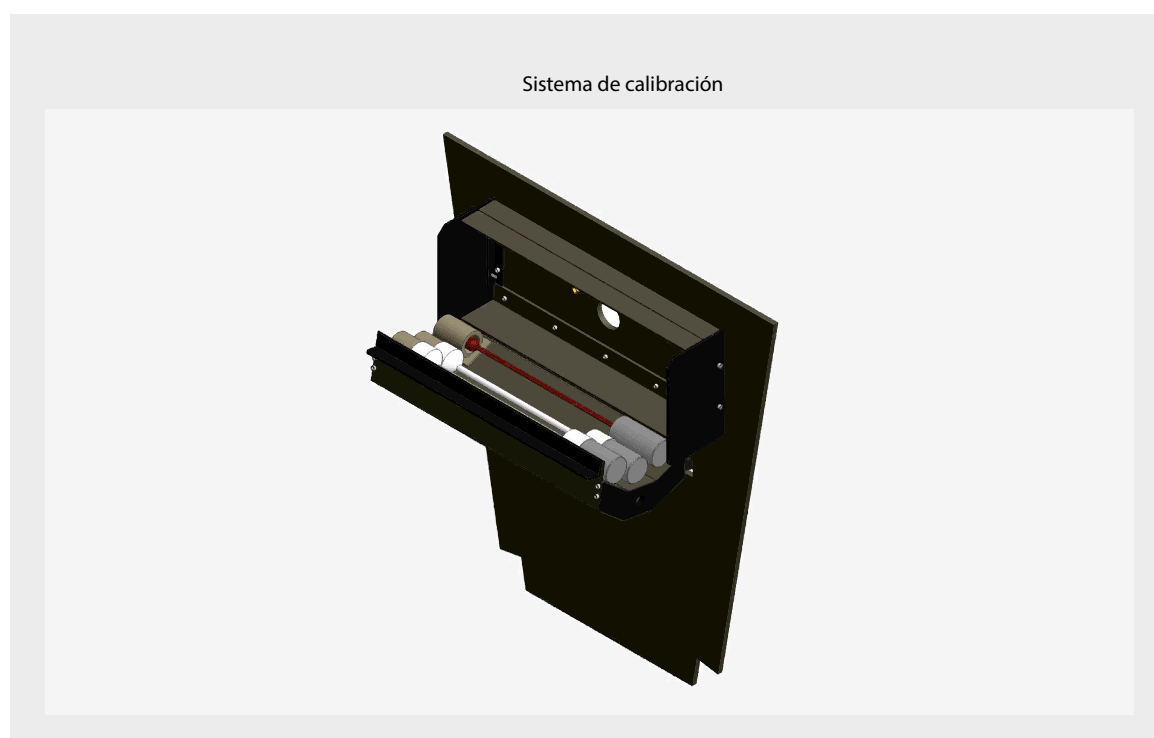


Figura 3.6: Modelo del ensamble de las lámparas de calibración. Referencia (Langarica and Tinoco, 2005).

3.1.4 Brazo de prismas

El brazo de prismas (Figura 3.7) también tiene la capacidad de moverse dentro o fuera del camino óptico del instrumento. Se mueve 90° hacia adentro para introducir cuatro prismas, que producen cuatro patrones circulares de anillos de interferencia que se utilizan para paralelizar el FP, mediante un software que indica las correcciones a realizar en los valores x y y del paralelismo. Una vez calibrado el brazo se puede mover 90° fuera del camino óptico y realizar las observaciones.

Cada prisma en forma de cuña tiene una pulgada de diámetro y un ángulo aproximado de 3.85° , ubicados en el interior de las celdas dispuestas en el brazo (Langarica and Tinoco, 2005). Este sistema permite la automatización del proceso de paralelismo de las placas del interferómetro FP.

El mecanismo es un brazo con movimiento angular de 90° y transmisión de engranes acoplados a un motor de 12 V DC y reducción de 218.4:1.

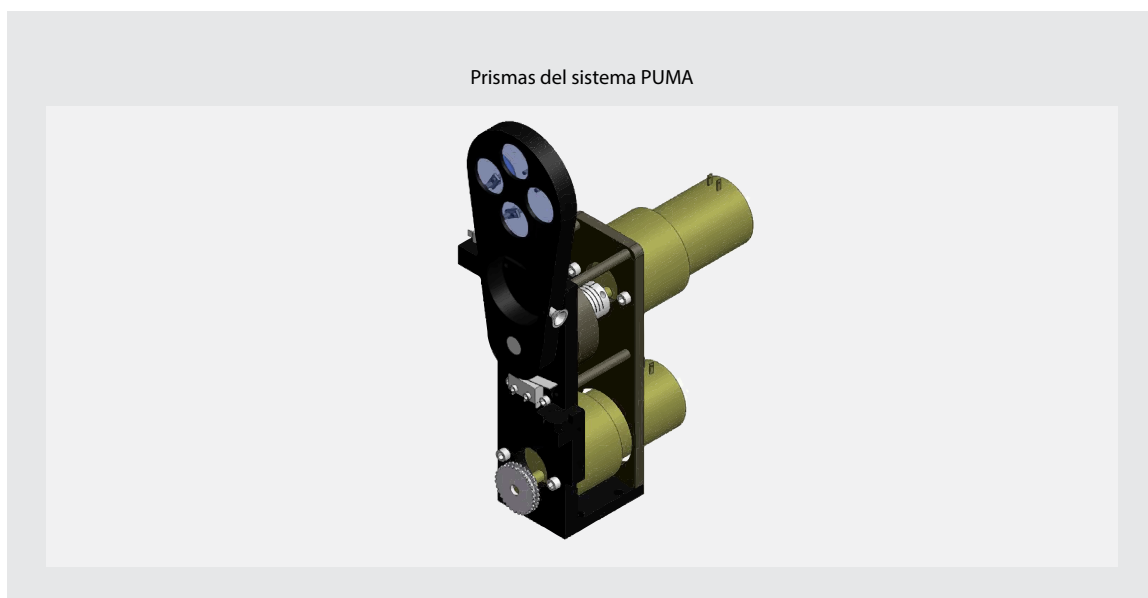


Figura 3.7: Brazo de prismas. *Referencia (Langarica and Tinoco, 2005).*

3.1.5 Sistema de enfoque

Para la correcta detección de imágenes se requiere un enfoque preciso, para obtener el detalle fino en las imágenes. Sin embargo, cuando el objeto es tan débil, requiere de exposiciones largas con un enfoque perfecto. El instrumento PUMA, tiene una cámara compuesta de lentes especiales, con su propio obturador capaz de enfocar los canales de interferencia hacia el detector CCD. El sistema se muestra en la Figura 3.8.

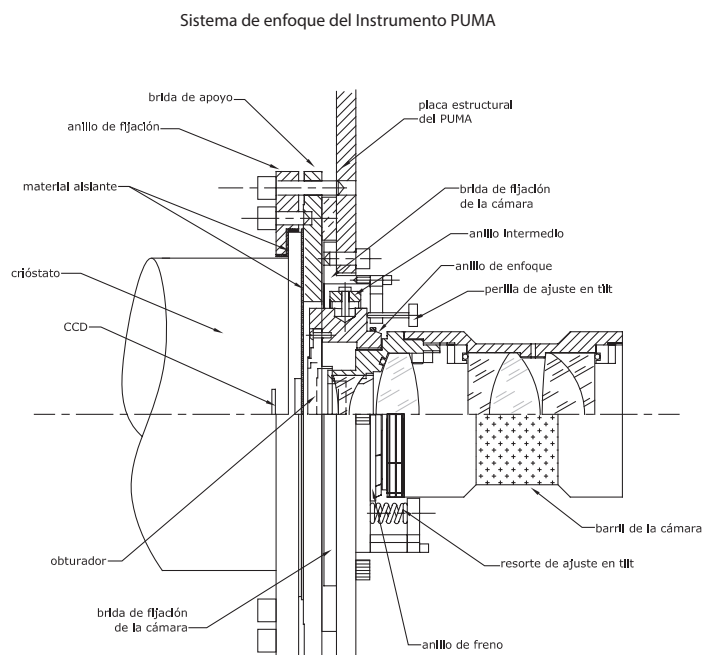


Figura 3.8: Cámara para enfoque del sistema PUMA. Referencia (Langarica and Tinoco, 2005).

3.2 Sistemas de control

3.2.1 Controlador CS100

El controlador CS100 de Queensgate Instruments Limited, fue desarrollado por un grupo de astrónomos del Imperial College de Londres (Queensgate-Instruments, 1981). El controlador tiene como finalidad realizar el barrido del interferómetro Fabry-Pérot y lograr la paralelización de las placas que lo componen, controlando el cambio de su distancia con exactitud. El CS100 proporciona el control requerido con retroalimentación activa y sensada de la posición de las placas del interferómetro. El CS100 corregirá en promedio los errores de la posición de las placas del interferómetro, en tiempos del orden de milisegundos.

El objetivo del controlador es dar desplazamientos al interferómetro Fabry-Pérot. Los ajustes de espaciado se pueden realizar por medio del control manual localizado en su panel frontal o remotamente por medio de la interfaz lógica compuesta de un ducto de comunicación paralela. Los datos de las resoluciones de estos desplazamientos y paralelismos se encuentran en los cuadros 3.4 y 3.5. La interfaz lógica maneja las funciones de control y es compatible con tecnología tipo TTL. El controlador CS100 se puede adaptar al suministro de línea de AC de 105 a 125 V ó 210 a 250 V, a una frecuencia de 50 a 60 Hz.

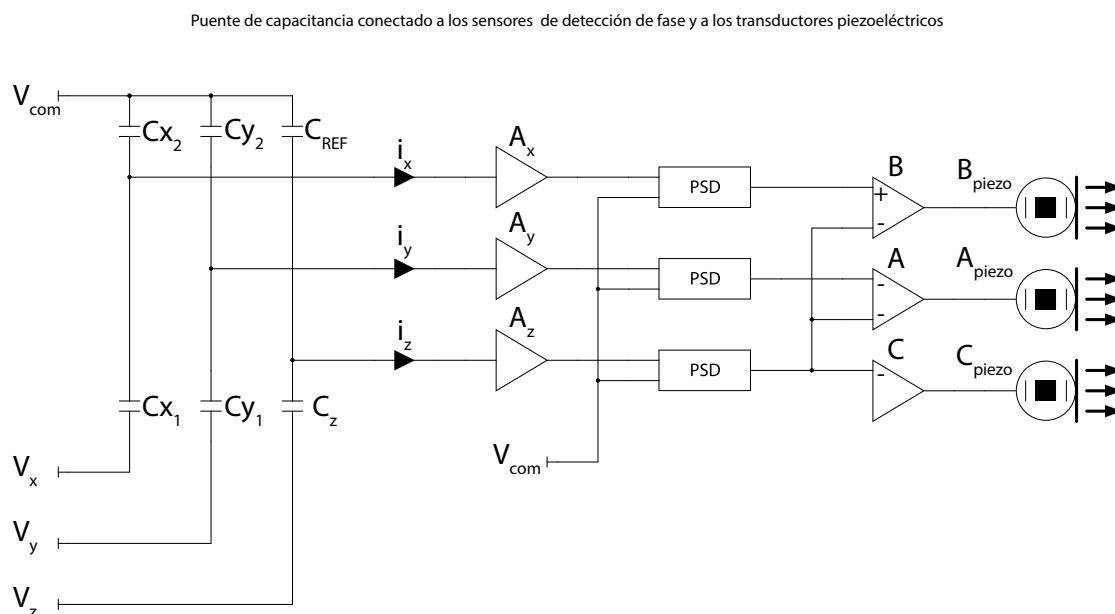


Figura 3.9: Puente de capacitancia del controlador CS100. Referencia (Queensgate-Instruments, 1981).

Cuadro 3.4: Resoluciones del controlador CS100. Referencia (Queensgate-Instruments, 1981)

| Parámetro | Valor |
|--|----------------------------------|
| Capacitancia micrométrica | Normalmente 22 [pF] |
| Espaciamiento del camino del capacitor | Normalmente 50 [μm] |
| Sensibilidad del transductor piezoeléctrico | 2.4 [nm/V] |
| Tiempo de respuesta para el Fabry-Pérot | Típico 1 [ms] |
| Tensión de excitación del puente de capacitancia | Nominal 10 [V] p-p |
| Frecuencia | 15.625 [kHz] |

El controlador CS100 trabaja con las señales del Fabry-Pérot de sus sensores de desplazamiento capacitivo, con resoluciones micrométricas, los cuales detectan cambios en el paralelismo. Para corregir estas variaciones se utilizan los transductores piezoeléctricos y así, mediante pequeños desplazamientos, se mantiene la estabilización de las placas del FP.

El principio de funcionamiento del CS100 (ilustrado en la Figura 3.9) es mediante la comparación de dos sensores capacitivos que corresponden a cada canal del Fabry-Pérot, es decir, para el canal x se comparan las señales de C_{x1} con C_{x2} , y de esta manera para el canal y ; en el caso del canal z , C_z se compara con un condensador muy estable de referencia montado en el FP, C_{REF} . Con las señales de error transferidas a los trasductores se forman tres lazos cerrados de control.

Los tres puentes de capacitancia C_x , C_y y C_z , son controlados por los voltajes V_x , V_y y V_z , teniendo como comparación al voltaje V_{com} . Los voltajes son de AC de la misma amplitud; sin embargo V_{com} está en fase opuesta a los demas voltajes. Si el voltaje de comparación es igual con alguno de los otros tres voltajes y el valor del par de las capacitancias de cada eje es el mismo, no habrá corriente que se transfiera a cada uno de los amplificadores destinados a cada canal A_x , A_y , y A_z ,

esto significa que el puente está balanceado.

Si existe un cambio entre el par de las capacitancias de un mismo eje, existirá una corriente que circulará hacia su amplificador correspondiente. Si los primeros canales son mayores a los segundos, V_{com} tendrá un retraso en fase de 90° y viceversa. El detector sensible de fase utiliza de referencia a V_{com} , por lo tanto genera un voltaje proporcional al error de paralelización en magnitud y sentido, transfiriendo la señal a un amplificador de alto voltaje, que manejará al transductor piezoeléctrico ligado formando el lazo cerrado.

Un error en el canal x provoca que el canal y se corrija, pero sin corregir el eje x nuevamente; aunque un cambio en el eje z provoca una corrección en los ejes x e y . Se pueden aplicar pequeñas tensiones controladas (dV_x , dV_y , dV_z) en los ejes x e y , para ajustar el paralelismo de las placas del interferómetro y el eje z para adecuar su espaciamiento. Los controles del panel frontal del CS100, del ajuste de los canales, se utilizan para la configuración inicial y las salidas de la interfaz de bus de control del CS100, se destinan para supervisar a control remoto los barridos del FP, su paralelismo y su condición de funcionamiento.

En el panel frontal del CS100 también se encuentran los controles de ganancia y de constante de tiempo de lazo abierto, que regulan su rendimiento. Estos controles determinan qué tan rápido se requiere que el sistema responda a un error de desplazamiento o tener una respuesta a resonancias mecánicas durante el barrido del interferómetro. El sistema CS100 puede responder muy rápido al barrido y corregir las alteraciones mecánicas.

Cuadro 3.5: Paralelismo y espaciamientos del controlador CS100. Referencia (Queensgate-Instruments, 1981).

| Parámetro | Rango | Resolución |
|-----------------------|------------------------------------|---------------------|
| Paralelismo manual | $\pm 0\text{-}5$ [μm] | 0.1 [nm] |
| Espaciamiento manual | $\pm 5\text{-}0$ [μm] | 1.0 [nm] |
| Paralelismo externo | $\pm 0\text{-}5$ [μm] | 0.25 [nm] (12 bits) |
| Espaciamiento externo | $\pm 1\text{-}0$ [μm] | 0.50 [nm] (12 bits) |

3.2.2 Sistema de control electrónico

El sistema de control original del instrumento utiliza un microcontrolador marca Octagon, modelo micro PC 5080 (μ PC) (Figura 3.10). Este microcontrolador se comunica con una estación de trabajo por medio de su puerto serie y un convertidor serie a ETHERNET. El control del interferómetro Fabry-Pérot y las señales de error del controlador CS100 se logran por medio de una de las dos interfaces periféricas programables con las que cuenta, con un total de 46 entradas y salidas digitales. La otra unidad se encarga de las señales de entrada de los detectores de proximidad, de los interruptores límite y de las señales de salida del control de los motores y del encendido o apagado de las lámparas.

Como sistema embebido, la Octagon es compatible con código del microprocesador Zilog Z80 y para su programación y desarrollo cuenta con un compilador de lenguaje BASIC, optimizado

Diagrama de bloques del Microcontrolador Octagon

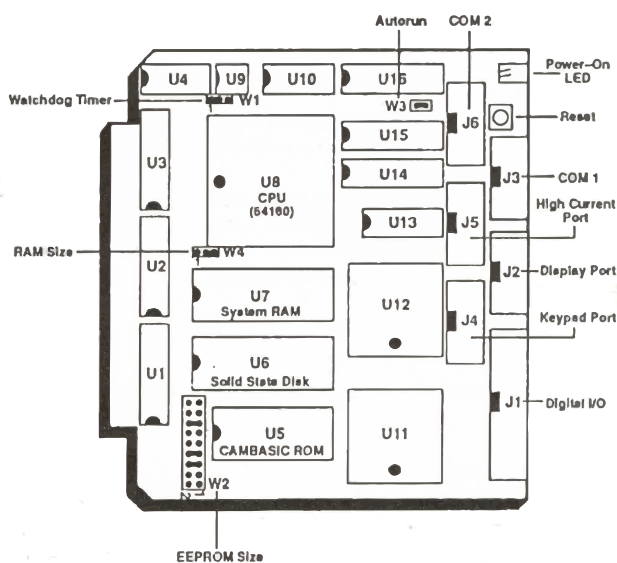


Figura 3.10: Diagrama de bloques de la computadora Octagon para el control del instrumento PUMA. Referencia (Bernal and Martínez, 2010).

para aplicaciones de control denominado: CAMBASIC IV (Bernal and Martínez, 2010). Las características de la μ PC son: 27 kB de memoria ROM, 128 kB de memoria RAM y 32 kB de memoria EEPROM, trabajando a 9.216 MHz.

Paras las conexiones entre el controlador CS100 y la μ PC, se dispone un arreglo de optoacopladores para aislar señales y eliminar ruido. Para comunicarse con la computadora Octagon, una interfaz de usuario fue diseñada para su interacción.

3.2.3 Detector de imágenes

El detector de imágenes utilizado es una cámara tipo CCD, que se coloca en el centro del patrón de interferencia, con el cual se detectan los anillos de interferencia como van pasando. El detector entrega como resultado una imagen digital como respuesta a la imagen observada. Para obtener la proporción de luz necesaria, en un área específica, los dispositivos de carga acoplada tienen una superficie parcialmente de gran tamaño.

En las observaciones astronómicas los CCD's tienen una gran ventaja en eficiencia cuántica, en comparación con los contadores de fotones (Moiseev et al., 2002); porque son más sensibles y responden más linealmente. Hay que tener en cuenta una dependencia del tiempo de un brillo del cielo y de una atmósfera; además con el CCD se debe conseguir el enfoque fino y perfecto del tren óptico (es decir del reductor focal, los filtros, etc).

El instrumento PUMA cuenta con un CCD, enfriado criogénicamente (para reducir el ruido electrónico y aumentar el tiempo de exposición), compuesto por una matriz de 2048 x 2048 píxeles, los cuales tienen un tamaño de 13 μm cada uno. Los píxeles son elementos sensibles a la luz y son grabados en una superficie de silicio, estos corresponden a un condensador capaz de transferir su carga eléctrica a los condensadores vecinos. Esto pasa cuando los fotones inciden sobre la placa y generan fotoelectrones que se guardan en un pozo potencial (pixel polarizado). Sin embargo estos píxeles en la lectura, representan un número, que en conjunto son procesados por medio de una computadora. El funcionamiento del CCD está basado en el principio fotoeléctrico.

3.3 Interfaz de usuario

La adquisición de las imágenes y los datos astronómicos se realizan a través de una interfaz de usuario, localizada en una de las estaciones de trabajo (SONAJA), del cuarto de observación del telescopio de 2.1 m del OAN-SPM. Esta interfaz se comunica con el instrumento y el CCD a través de distintos módulos tanto de software como de hardware.

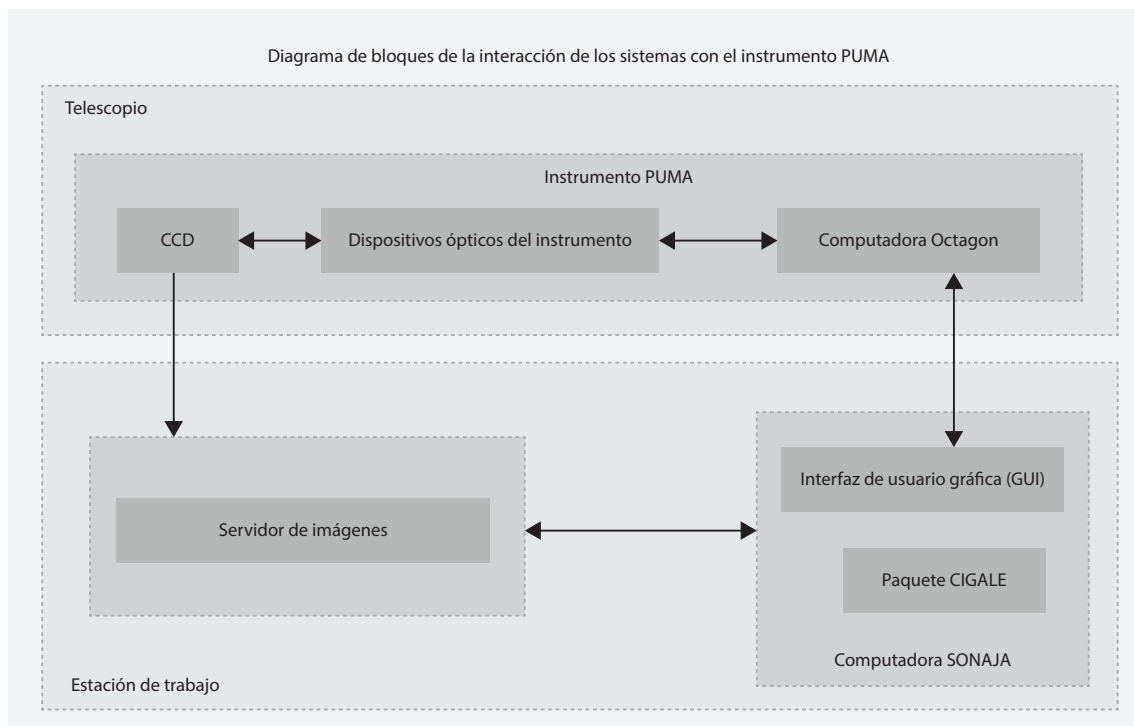
La interfaz gráfica del usuario (GUI) controla al instrumento PUMA, es decir que se comunica con el microcontrolador (μ PC) y manda comandos requeridos para el movimiento de la rueda de filtros, del brazo de prismas, de la unidad móvil de calibración y de la posición del FP, además de seleccionar y encender o apagar las lámparas. También permite enviar una señal de reinicio al controlador CS100, si sucediera que el lazo de control del puente capacitivo se abriera.

El programa CIGALE se encarga de la reducción de los datos ensamblados en forma de cubo (x, y, λ) , obtenidos de la observación. CIGALE tiene memoria compartida con la GUI en SONAJA, por lo que se pueden apreciar y analizar los canales del cubo de datos en tiempo real.

El flujo de información se realiza de la siguiente manera. La GUI se comunica con el sistema de control, se envían los comandos de selección y posicionamiento. El sistema de control envía el estado del instrumento a la GUI, que corre en la estación de trabajo SONAJA. La GUI configura el controlador del CCD, envía los comandos de adquisición y recibe los archivos con las imágenes. El sistema de control configura y ejecuta el barrido del FP de acuerdo a los canales y los pasos requeridos.

La GUI favorece al astrónomo al permitirle controlar todo el instrumento PUMA. La GUI permite adquirir una imagen o adquirir todo un barrido de canales (en formato FITS), para la obtención de un cubo de datos, tanto del objeto celeste como el de las lámparas de calibración. Para cualquier caso se debe configurar el instrumento PUMA, posteriormente se deben capturar los parámetros de adquisición del CCD y del Fabry-Pérot. Después de la adquisición se procede al análisis del cubo de datos (sí es que se decidió obtener el cubo de datos). En este caso la IGU se comunica con parte del programa de control de la Octagon, en el cual automatiza el barrido del Fabry-Pérot. Las imágenes se guardan en el disco duro de la estación de trabajo.

De la interfaz gráfica de usuario (Figura 3.11) se puede seleccionar la configuración del CCD, también se pueden modificar diversos parámetros para la adquisición de uno o varios canales con características determinadas, entre ellas el tamaño de la imagen (el máximo valor es el tamaño del CCD, de 2048 x 2048), las dimensiones de la imagen pueden ser menores y si no se especifican las coordenadas de localización de la imagen, se tomará en cuenta por defecto el centro



Cuadro 3.6: Interacción entre el instrumento PUMA, la computadora Octagon y la estación de trabajo.

de la matriz de píxeles. También se permite la combinación de píxeles adyacentes por medio de la función binning; el tiempo de exposición de cada canal o imagen; la ganancia y el tipo de imagen.

Para las observaciones con el interferómetro Fabry-Pérot se tiene que adquirir un cubo de canales de la calibración y otro cubo de datos de los interferogramas del objeto, puesto que las calibraciones son referencia para los mapas de fase y deducción de las velocidades radiales con los interferogramas del objeto.

La configuración de los parámetros del Fabry-Pérot se pueden establecer en la ventana con dicho nombre, donde se debe capturar la longitud de onda de la lámpara seleccionada para la calibración, del objeto a observar y del barrido con el que se escaneará el objeto; la velocidad del objeto, que se corregirá más tarde con los datos procesados por el programa CIGALE; el número de canales o imágenes expuestos y el paso y origen del barrido que se generarán al seleccionar el cálculo de los mismos.

Al final de la configuración del instrumento se puede proceder a la exposición de los cubos de datos o de la imagen directa, donde se podrá visualizar cada canal adquirido. El cubo de datos se puede guardar en formato para el programa CIGALE o en el formato FITS. Los nombres con los que se guardan los datos llevan los nombres cio/cie. La extensión .cie será el cubo de interferogramas obtenido de las calibraciones y la extensión .cio serán los interferogramas del objeto observado.

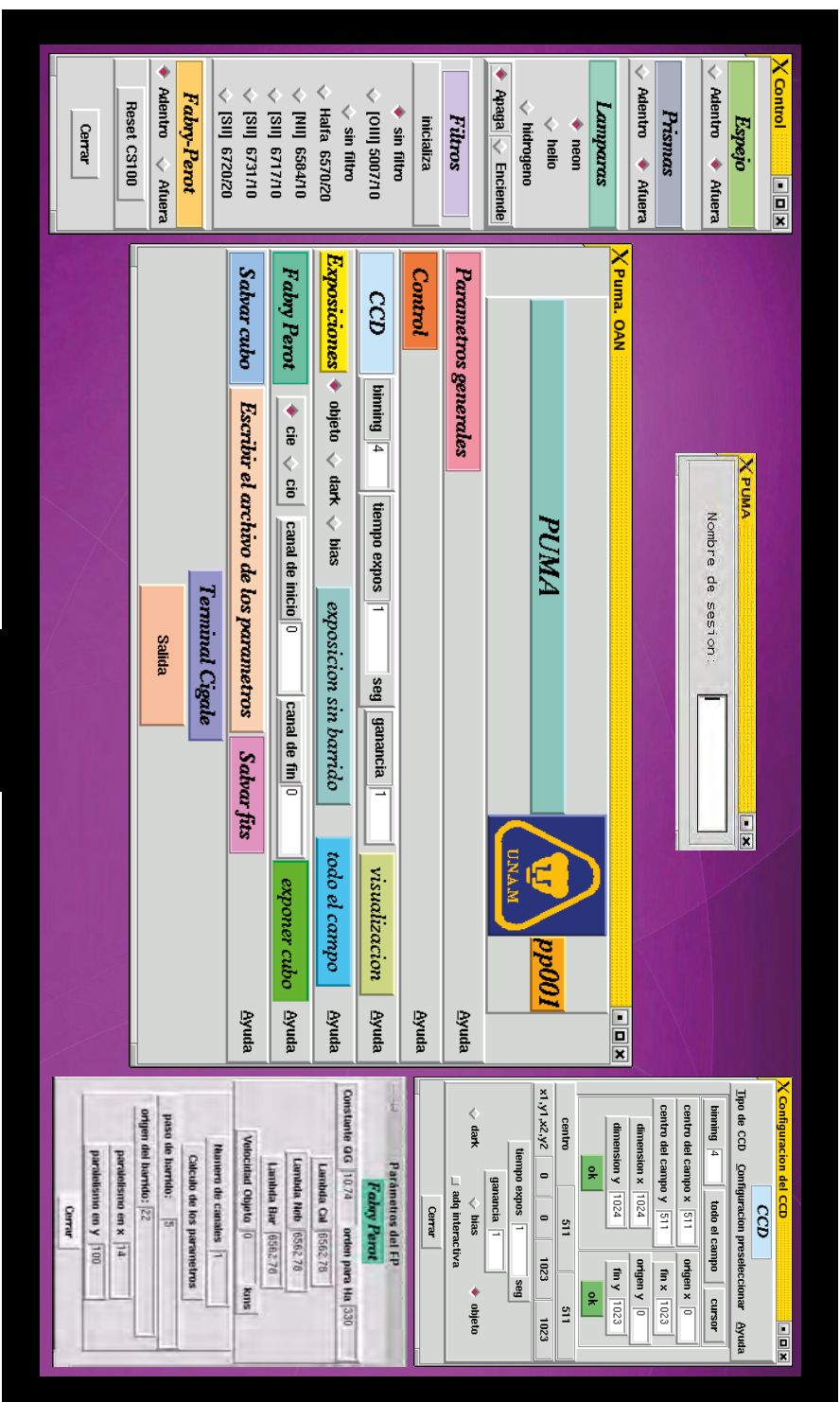


Figura 3.11: Interfaz gráfica del usuario del instrumento PUMA. Se muestran las ventanas de control del CCD, del Fabry-Pérot y del control de los componentes móviles del PUMA. El barrido del Fabry-Pérot se realiza con el botón exponer cubo y se debe ejecutar al final del establecimiento de todos los parámetros del instrumento. *Referencia (Martínez et al., 2007)*

CAPÍTULO

4

Microcontrolador de placa reducida BL4S200

En este capítulo se describen las características del controlador para el nuevo sistema electrónico

Índice

| | |
|------------------------------------|----|
| 4.1 Controlador embebido | 44 |
| 4.2 Características | 46 |
| 4.3 Subsistemas | 48 |

4.1 Controlador embebido



Figura 4.1: Controlador del nuevo sistema electrónico, el microcontrolador de mono-placa reducida Rabbit BL4S200. Referencia (Digi-Semiconductor, 2013a).

Este proyecto de ingeniería hace frente al control y automatización de los procesos del sistema y a su construcción. El instrumento PUMA comparte atributos comunes de las aplicaciones para un control embebido, incluyendo sensores para controlar actuadores, en donde se requieren comportamientos deseados y, en donde también, existen interfaces entre hombre y máquina. Por lo que el uso de un microcontrolador embebido es necesario para operar detrás de todos estos procesos y manipular al sistema en conjunto.

El control que necesita el sistema PUMA, es un ejemplo de una aplicación embebida que requiere funciones de control particulares, pero con numerosas características de supervisión y detección; sin embargo el rendimiento de este tipo de sistema también es fijado por el diseño optomecánico.

Los controladores embebidos se pueden dividir en cuatro categorías: controladores compactos, controladores de nivel de placa, módulos básicos y dispositivos de nivel de chip (Hyder and Perrin, 2005). Los controladores a nivel de placa requieren un integrador de sistemas para proporcionar protección física de los componentes electrónicos. La solución a nivel de placa tiene una vasta selección y características de entradas y salidas, de tipos de procesador y de capacidad de memoria. Estos tipos de controladores a menudo pueden ser utilizados para proporcionar una solución de control completa para aplicaciones de bajo volumen.

El instrumento PUMA fue concebido bajo la filosofía anterior y debido al paso del tiempo, se llegó a la necesidad de modernizar el sistema de control. Las razones fueron la obsolescencia de los componentes electrónicos y la necesidad de mejorar el desempeño general del instrumento.

El sistema de control anterior, utilizaba un microcontrolador Octagon, que se sustituyó con un nuevo diseño de control, utilizando el microcontrolador de placa reducida Rabbit BL4S200. Esta tecnología de microprocesadores nació de RABBIT Semiconductors, fundada por el CEO de Z-World, Norm Rogers (Hyder and Perrin, 2005), al ver la necesidad de mejorar los microprocesadores en el mercado integrado y llevar a cabo la innovación en la industria del controlador embebido a nivel de chip con los módulos principales del Rabbit, enfocándose en los controladores de funciones completas.

Los principios de RABBIT Semiconductors inician cuando Norm y su compañero de trabajo Paul Mennen fundaron Decmation, en el año de 1981 en Silicon Valley, cuyo primer producto original era una tarjeta enchufable que permitía ejecutar un software en un determinado tipo de mini-ordenadores de Digital Equipment Corporation (DEC). En 1985 con investigación y desarrollo por parte de Norm, fabricaron un producto en etapa de crecimiento llamado Blue Lightning, que permitía a los usuarios de IBM ejecutar sus aplicaciones heredadas de CP/M (un sistema operativo desplazado por el MS-DOS de IBM).

Sin embargo Norm empezó a desarrollar herramientas para sistemas embebidos basados en Z80 (microprocesador de 8 bits lanzado en 1976 por la compañía Zilog). Por el cambio de giro y reclamos por parte de DEC, Norm decidió cambiar el nombre de la empresa de Decmation a Z-World (la "Z" fue por el microprocesador Z80), cuyo primer éxito fue el Little Giant en 1990, el cual integró al microprocesador Z80 con entradas y salidas que permitirían el control y supervisión de los sistemas; así como interactuar con señales externas y cuya herramienta de desarrollo fue Dynamic C, un compilador de lenguaje C escrito por el equipo de Norm. Con ajustes y relaciones entre el hardware y el firmware de ejecución y el desarrollo de nuevas herramientas de software de compilación, Norm empezó a construir un nuevo procesador.

En 1997 (Hyder and Perrin, 2005), Norm, Pedram (estudiante de ingeniería eléctrica en la Universidad Estatal de Sacramento) y Monte (arquitecto y diseñador de productos Zilog) comenzaron con el desarrollo del Rabbit 2000. Cuyos costos de desarrollo eran mucho menores que el de otros microprocesadores en ese tiempo. Con gran aceptación en la industria, siendo Rabbit Semiconductors (comprado en 2006 por Digi International), lanzaron el Rabbit 2000 en 1999. Posteriormente en 2001, lanzaron el Rabbit 3000 y recientemente las serie de Rabbit 4000. Todos estos productos comparten herramientas de desarrollo común, esto hace que las migraciones de diseño a una nueva variante sean rápidos, fáciles, de bajo costo y de bajo riesgo.

La influencia de factores de selección como el costo, tamaño, disponibilidad, herramientas de desarrollo, sostenibilidad del producto y requisitos computacionales en condiciones de trabajo, el monitoreo de su rendimiento y de su funcionamiento en tiempo real, así como buenas características de documentación y depuración para la solución y corrección de problemas recurrentes hacen de este microcontrolador de placa reducida (SBC), un controlador embebido ideal para el proyecto.

El microcontrolador Rabbit también tiene la capacidad de realizar multitareas, realizando actividades síncronas y asíncronas. Las entradas y/o salidas digitales de la placa reducida están protegidas contra altos voltajes, pero en caso de que fallaran, el microcontrolador seguiría funcionando realizando el trabajo requerido durante su ejecución, esperando posteriormente al reemplazo del transistor o componente dañado.

En un sistema embebido cabe resaltar la importancia de la dependencia entre el software y el hardware del sistema, cuyo objetivo es resolver los problemas de diseño específicos para el desarrollo de sistemas integrados y su control dedicado, como un conjunto de dispositivos gobernados por un microprocesador. En este caso los sensores deben ser monitoreados, requiriendo la

comunicación y sincronización con otros dispositivos, en donde los actuadores deben ser energizados. En este proyecto de instrumentación astronómica existen aplicaciones que requieren exigencias para la detección de sensores, al igual que otros proyectos industriales. En el instrumento PUMA, el control se reduce a la vigilancia de sensores y dispositivos de accionamiento.

En el proyecto se seleccionaron los componentes de apoyo adecuados y se interconectaron estos componentes a la SBC, diseñando la sincronización y la lógica para hacer que todos los sistemas trabajen juntos. El microcontrolador de placa reducida permite integrar los sistemas embebidos y centrarse en los problemas de interfaz y en el código de aplicación, para ser finalizado y probado, tomando en cuenta el tiempo en el diseño, la creación del prototipo y la prueba de un sistema de este tipo.

4.2 Características

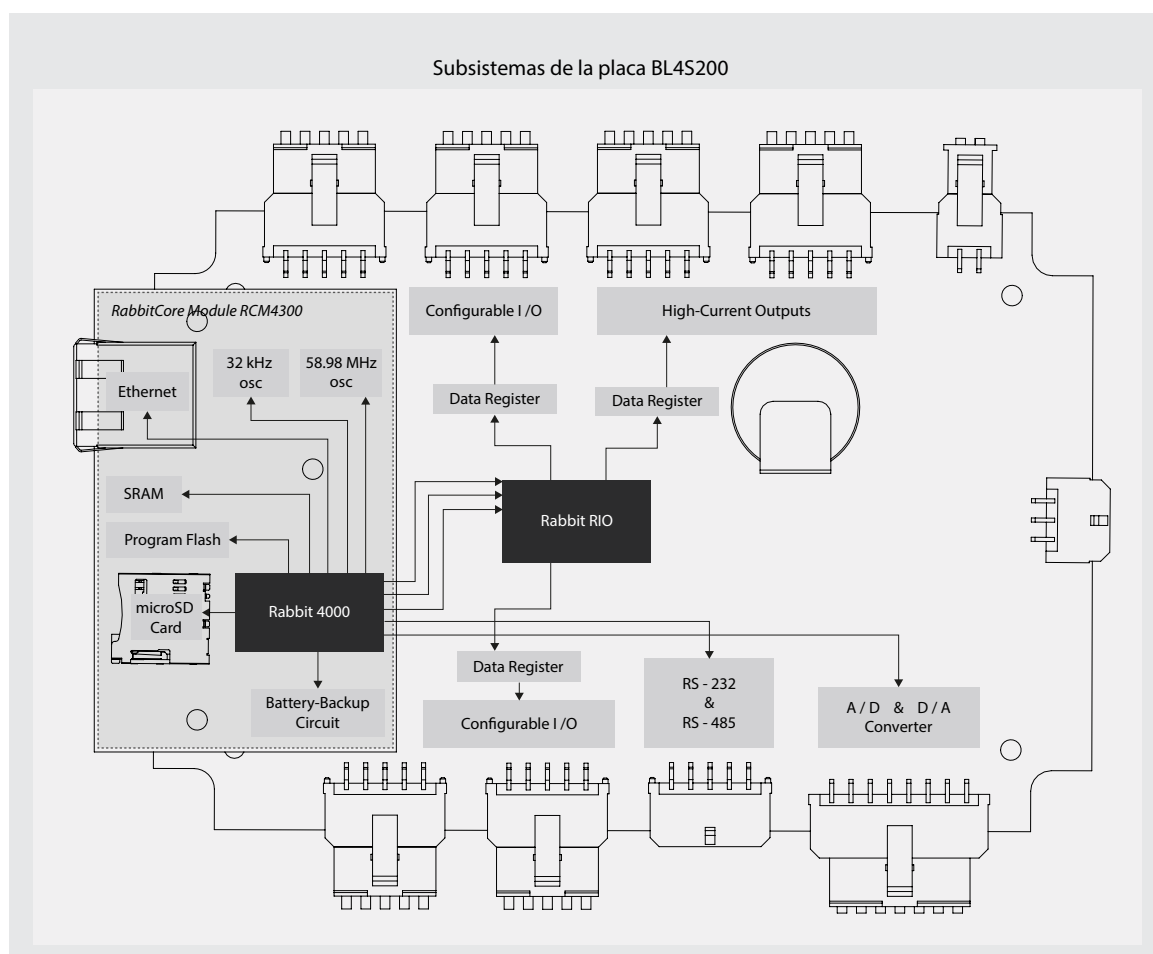


Figura 4.2: Diagrama descriptivo de los principales componentes y subsistemas de hardware del RCM4300 y de la placa BL4S200. Referencia (Digi-Semiconductor, 2013a).

Los microcontroladores de monoplaca reducida BL4S200 (Figura 4.1) son de alto rendimiento (Digi-Semiconductor, 2013a), su entorno de desarrollo y depuración de programas es el compilador Dynamic C (para todo el hardware Rabbit Semiconductor) y tienen unas excelentes características de funcionamiento a un rango de temperaturas de -20°C a $+85^{\circ}\text{C}$ y con 5% a 95% de humedad sin condensación. Son ideales para las aplicaciones en los procesos de control y consisten de un módulo principal RabbitCore RCM4300, con microprocesador Rabbit 4000, el cual procesa rápidamente los datos con su instrumentación asociada, capaz de interactuar con interfaces de hardware, permitiendo aplicaciones de control de sistemas embebidos.

El Rabbit 4000 es un microprocesador de alto rendimiento con baja interferencia electromagnética (EMI clase B), y está diseñado específicamente para el control incorporado, las comunicaciones y la conectividad Ethernet. Gran variedad de funciones integradas y una arquitectura inferior facilitan un rápido diseño de hardware, mientras que un conjunto de instrucciones de programación accesible en C promueve el desarrollo eficiente en las aplicaciones más complejas.

En el corazón del Rabbit 4000 está el bloque de la CPU, el cual tiene un bus interno que se comunica con los periféricos internos; este bus se compone de señales de dirección de control con cinco puertos paralelos de 8 bits, dando un total de 40 entradas o salidas digitales y de direcciones internas (Figura 4.2). El Rabbit 4000 cuenta con bloques de contadores y temporizadores para su uso distribuidos en el Rabbit RIO; esto permite la creación de señales PWM y señales de cuadratura.

El Rabbit 4000 ofrece muchos periféricos especializados: una interfaz interna de 10/100 Base-T Ethernet, que soporta ocho canales de DMA en la memoria externa; dos canales de decodificadores de cuadratura (cada uno de dos entradas); así como un contador ascendente o descendente; cuatro salidas independientes de amplitud de modulador de pulso (PWM), cada uno basado en una estructura de 1024-pulsos, conducidos por la salida de un temporizador interno programable; dos canales de entrada de captura, cada uno con un contador de 16 bits, sincronizado por la salida de un temporizador interno; cuatro niveles de prioridad de interrupción con una respuesta rápida a eventos en tiempo real, un convertidor analógico-digital de 8 canales, dos relojes (oscilador principal y el del tiempo normal) y el supervisor de *watchdog*.

El Rabbit 4000 corre a velocidades de hasta 58,98 MHz, el cual se deriva de un cristal de 29.49 MHz, gozando del duplicador del reloj interno del microprocesador, permitiendo el uso de cristales de mediana frecuencia para reducir las emisiones radiadas, operando con un núcleo a 3.3 V. También tiene varias características de diseño, que prácticamente eliminan problemas de EMI. La amplitud de cualquier radiación electromagnética se reduce por el esparcidor de espectro interno, por los relojes cerrados (que impiden la sincronización innecesaria de registros no utilizados), por los planos de alimentación separados por el núcleo del procesador y los pines de entrada y salida. Sus instrucciones compactas establecen altas velocidades de reloj, con un rendimiento lógico-matemático rápido.

El módulo RabbitCore RCM4300 (Digi-Semiconductor, 2013e) utilizado en la placa BL4S200, tiene una memoria RAM estática (SRAM) de datos de 512 kB, respaldada por batería y una SRAM de 512 kB para el programa en ejecución, una memoria flash serie de 512 kB, que contiene el bloque de usuario y almacena el programa de aplicación. El módulo soporta también una memoria microSD de 128 MB a 1 GB.

El módulo RabbitCore, en el modelo BL4S200, es compatible con una tarjeta microSD extraíble de hasta 1 GB para almacenar datos y estará dedicada en las aplicaciones en las que se tengan que guardar datos de forma masiva, pero es generalmente inadecuada para la ejecución directa del programa. A diferencia de otros dispositivos flash, la tarjeta micro SD lee, escribe y maneja

cualquier pre-borrado necesario internamente. Estos dispositivos, por ser tecnología de estado sólido, son ideales para su uso en sistemas embebidos.

Una de las razones de que se haya utilizado el sistema embebido Rabbit, es que Rabbit Semiconductors, a partir de Z-World, se ha comprometido a la disponibilidad y desarrollo a largo plazo de sus circuitos integrados y de sus controladores embebidos.

4.3 Subsistemas

El BL4S200 es un equipo avanzado de monoplaca reducida que incorpora el potente microprocesador Rabbit 4000, las opciones de memoria flash, memoria RAM estática, puertos de entradas y salidas digitales, entradas para el convertidor analógico-digital y las salidas del convertidor digital analógico, puertos de comunicación serial RS-232 / RS-485 y conectividad de red Ethernet con conector RJ-45 (Figura 4.3).

La captura de entrada de doble canal es una de las peculiaridades más distinguidas del módulo Rabbit. Esta parte inteligente de la ingeniería, permite al Rabbit 4000 medir con precisión el tiempo entre eventos digitales externos, con capacidades de entrada avanzadas, incluyendo la cuenta de eventos, captura de eventos y decodificadores de cuadratura, la medición de ancho de pulso y la frecuencia, que pueden configurarse en la mayoría de los pines de entradas y salidas digitales.

La placa BL4S200 dispone de un chip Rabbit RIO, como dispositivo periférico, para la distribución de las entradas y salidas, diseñado para las aplicaciones en las que cambien constantemente los tiempos y contadores. Los circuitos integrados de los Rabbit RIO se conforman de ocho bloques tipo contador/temporizador. En la placa BL4S200 se encuentran tres chips RIO, lo que da un total de 24 bloques, cada bloque se define por el número de RIO y su número de bloque.

El Rabbit RIO puede comunicarse con el microprocesador Rabbit 4000, ya sea en paralelo o en un modo serie (Digi-Semiconductor, 2013d). En el modo paralelo se comunica a través de un bus paralelo de ocho bits de datos, cuatro bits de dirección y cuatro bits de control. En el modo de serie las líneas de datos en paralelo están disponibles para ser utilizadas como de uso general, ya sea de entradas o salidas.

Este diseño de los bloques de las entradas y salidas del Rabbit RIO permite que cualquiera de los ocho bloques idénticos, cada uno con cuatro bits o pines de entradas o salidas, puedan ser programados para realizar cualquier número de funciones diferentes, incluyendo un modulador de ancho de pulso, contadores de eventos, decodificadores de cuadratura; así como entradas y salidas. También cuenta con varios puertos de salida con capacidad de manejo de corriente, a través de transistores tipo MOSFET.

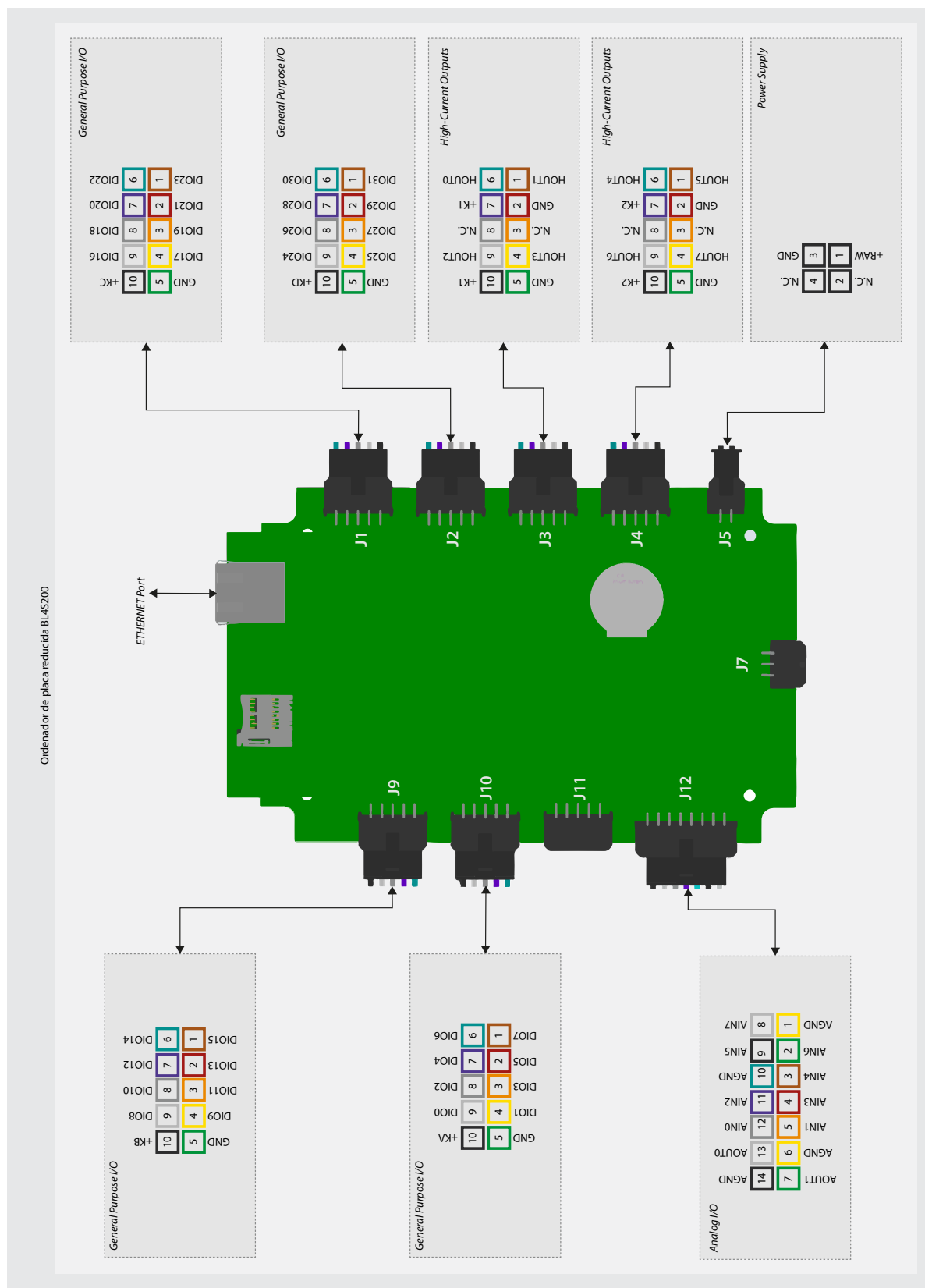


Figura 4.3: Disposición de las entradas y salidas de la monoplaca Rabbit BL4S200. Referencia (Digi-Semiconductor, 2013a).

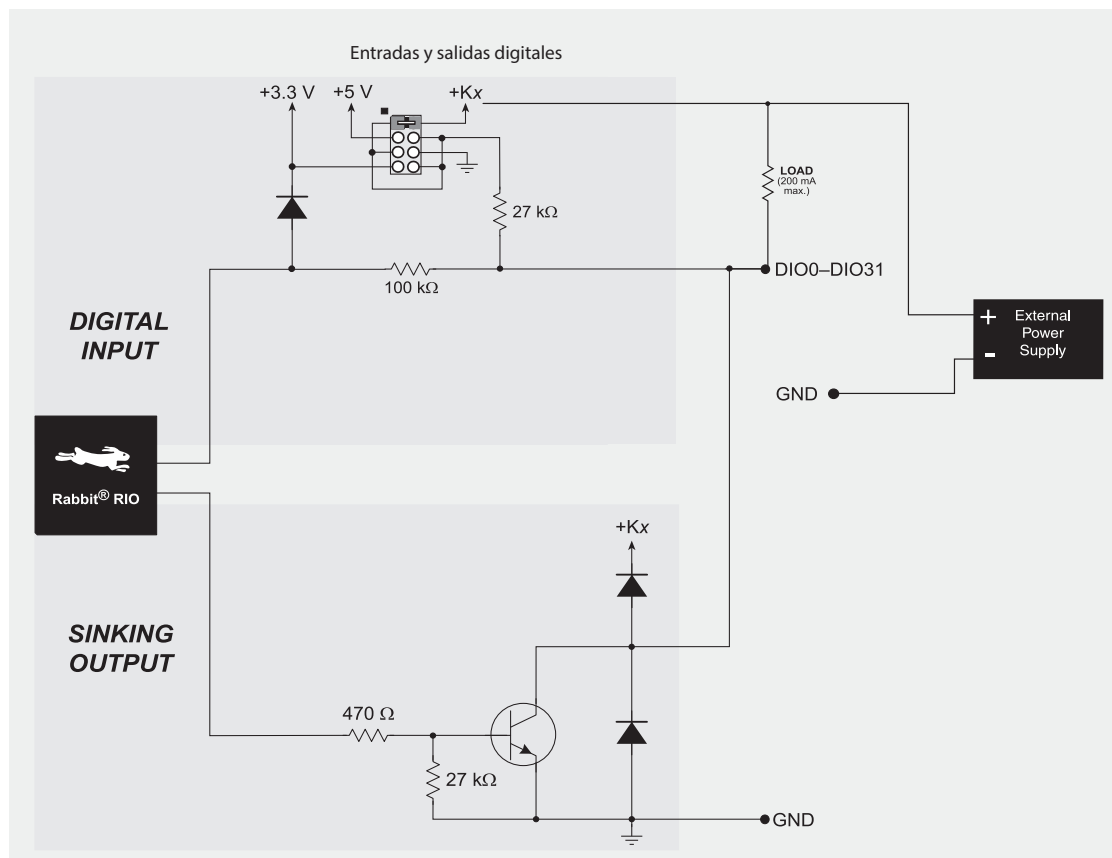


Figura 4.4: Configuración de las entradas y salidas digitales de la placa BL4S200. Referencia (Digi-Semiconductor, 2013a).

4.3.1 Entradas y salidas digitales

El ambiente del hardware trabaja por medio de puertos de entradas y salidas digitales protegidas (elementos clave en el diseño de todos los sistemas integrados), cuya lógica depende del sistema a controlar. El conjunto de las funciones de entradas/salidas del Rabbit hace que este sea más identificable con aplicaciones de control, que como un procesador numérico. La placa está suministrada con fuentes de alimentación, de entradas y de salidas adecuadas para las aplicaciones embebidas (Figura 4.4).

La placa BL4S200 cuenta con 32 canales de entradas/salidas digitales, configurables individualmente por medio de software. El voltaje de entrada se puede configurar de 0 a 36 V DC y las salidas de tipo sumidero, tienen capacidad de manejar hasta 40 V y 200 mA de corriente cada una.

El Rabbit 4000 permite al diseño del sistema, la opción de mantener el bus de memoria de la CPU separado del bus de las entradas y salidas. Cuenta con cuatro puertos de entradas y salidas digitales de ocho bits. Los pines de entradas/salidas comparten funciones con otros periféricos del chip y tienen varias características deseables. Una de las mayores preocupaciones prácticas es la capacidad de interactuar con la lógica de cinco voltios. El Rabbit 4000 está diseñado para funcionar de 1,8 a 3,6 V de alimentación. Sin embargo, todas las entradas del Rabbit (con la excepción

de los pines del oscilador y de potencia) son tolerantes hasta 36 V. Los pines de entradas/salidas del Rabbit, son alimentados desde un conjunto separado de la alimentación al núcleo de la CPU.

4.3.2 Interrupciones

El poder de un sistema embebido es su capacidad para responder a los eventos del mundo real; un sistema embebido puede controlar una máquina y tomar las decisiones adecuadas. Otra de las razones para el uso interrupciones en los sistemas controlados por microprocesador, se debe a que a menudo, es la única manera para lograr un alto rendimiento de las entradas y salidas en un diseño. Los sistemas embebidos se refieren a sistemas basados en eventos, debido a su capacidad para responder a eventos del mundo real en tiempo real. También es importante señalar que los eventos en tiempo real son completamente asíncronos a la sincronización de la CPU, la cual tiene que responder a estos eventos, independientemente de donde se encuentre el flujo del programa.

A los dispositivos de entrada se les puede interrogar sobre la generación de un evento, entonces el sistema puede ser diseñado de modo que el evento genere una interrupción. En el caso de la entrada del usuario, desde un puerto serie o un estado crítico identificado por un sensor, el sistema puede hacer una pausa de lo que está haciendo actualmente, responder al evento externo y tomar decisiones apropiadas. La ejecución del código de la interrupción muestrea sobre el estado específico de la entrada en cualquier momento; el evento de interrupción suspenderá la aplicación y las tareas que se están ejecutando, obligando al microprocesador a ejecutar una parte especial del código llamando a una rutina de servicio. Esta procesa la interrupción y, al finalizar el código, reanuda la aplicación a partir de donde se produjo la interrupción.

En el sistema PUMA el manejo por interrupciones es para las aplicaciones que requieren hacer un servicio, a partir de una señal de un dispositivo de entrada; generando una excepción para que los vectores de la CPU den servicio a la ISR. Por lo tanto, la interrupción se opera como una señal que indica que ha ocurrido un evento externo, tal como el aviso de un sensor o conmutador.

La capacidad del módulo para generar interrupciones permite al diseño del sistema añadirle hasta cuatro interrupciones externas al sistema basado en el Rabbit, incluso si las capacidades de evento de sincronización del módulo no se utilizan; las interrupciones dependen de cómo se escribe el código; sin embargo el módulo posee fuentes de interrupción externas e internas, con prioridad unas sobre otras y pueden ser activadas por el control del programa o estar siempre habilitadas. La rutina de servicio de interrupción (ISR) es responsable de determinar qué temporizador tropezó con la interrupción.

4.3.3 Modulación de ancho de pulso

El término modulación de ancho de pulso se deriva de una suposición que fija el período de un pulso. En este caso el pulso se pensó en la parte alta de la forma de onda de la Figura 4.5. El ciclo de trabajo (D) es equivalente a la anchura de pulso cuando se refiere al PWM. Una salida de PWM

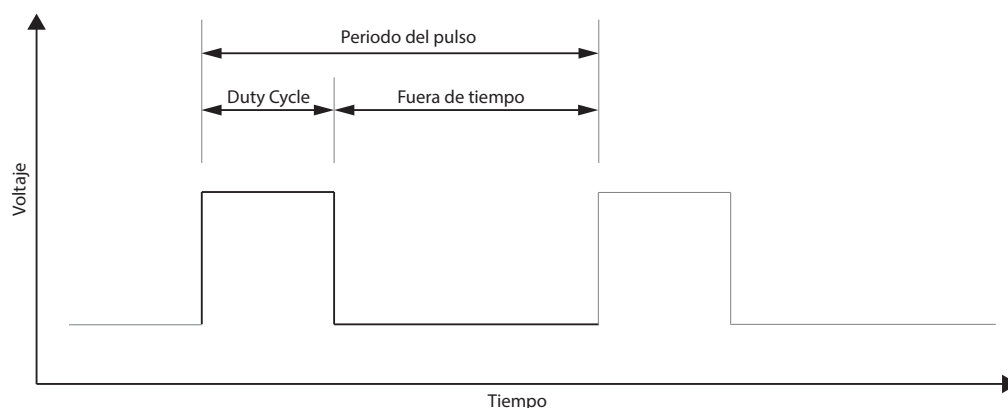


Figura 4.5: Representación de una modulación de ancho de pulso. Referencia (Hyder and Perrin, 2005).

se describe como no invertida cuando comienza en alto, permanece en alto durante un ciclo de trabajo (que es una fracción del periodo) y, a continuación, pasa a nivel bajo durante el resto del período. El porcentaje de ciclo de trabajo se define como la relación entre la duración cuando la función esta en nivel alto (τ), entre el periodo de la función (T).

$$\%D = \left(\frac{\tau}{T} \right) \cdot 100\% \quad (4.1)$$

Cada bloque de entrada/salida puede generar hasta cuatro señales PWM sincronizadas, que se elevan en el mismo módulo del contador, pero que tienen diferentes ciclos de trabajo. Una salida PWM se establece cuando es activada por un evento y la señal puede ser configurada en forma ascendente o descendente. El inicio de la señal de PWM es independiente en la mayoría de los pines de entrada/salida y en todas las salidas de alta corriente; sin embargo, si las señales de entrada/salida pertenecen a un mismo bloque Rabbit RIO, mantienen la misma fase.

4.3.3.1 Entradas y salidas digitales de alta corriente

La inclusión de un controlador incorporado en el instrumento PUMA permite la implementación de algoritmos, para sincronizar gradualmente los movimientos y las velocidades de los motores de las partes móviles y operarlos para aplicar pares específicos, en tiempos específicos y en posiciones específicas. Las salidas digitales, por lo general, necesitan sistemas integrados para el control de actuadores con circuitos digitales. La corriente limitada disponible en una salida de CMOS, raramente es suficiente para encender más allá de un LED. Para la aplicación de salidas de alta corriente se requiere de transistores de unión bipolar (BJTs), pares Darlington, MOSFETs, relés electromecánicos o relés de estado sólido (SSR).

La SBC BL4S200 junto con su configuración de puerto en su configuración de puente H, proporciona suficiente corriente para proporcionar suficiente par a los motores y permitir su movimiento con precisión, ya que cuenta con ocho salidas digitales de alta corriente configurables por software, de tipo sumidero o fuente, con voltajes hasta de +40 V DC y 2 A como máximo por cada canal.

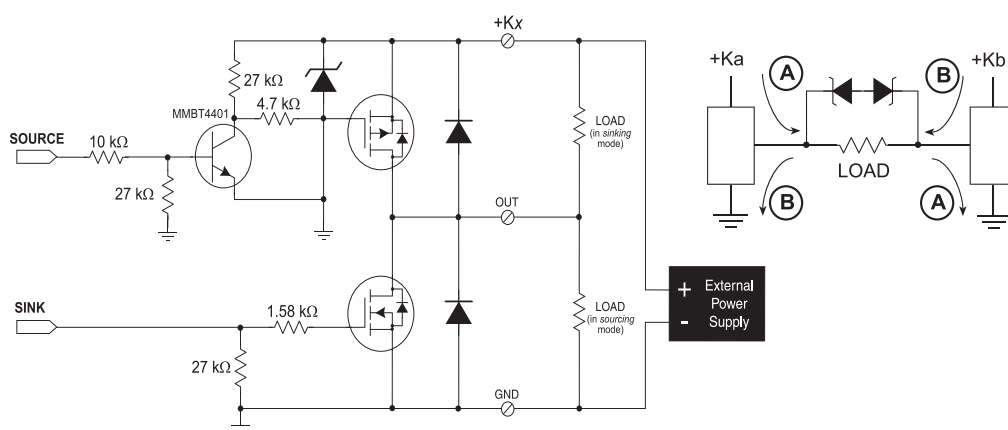


Figura 4.6: Para el puente de H, Ka y Kb deben ser los mismos. Esto se logra más fácilmente mediante el uso de las salidas del mismo banco en un conector. Referencia (Digi-Semiconductor, 2013a).

El control del motor es implementado con la modulación por ancho de pulsos (PWM). EL Rabbit tiene un módulo de PWM que proporciona ocho salidas independientes de alta corriente con control PWM (Figura 4.6). EL PWM en el Rabbit se define y opera de la siguiente manera: suponiendo que la energía en un pulso, está contenida completamente en la porción del ciclo de trabajo de la forma de onda, entonces este será directamente proporcional a la energía.

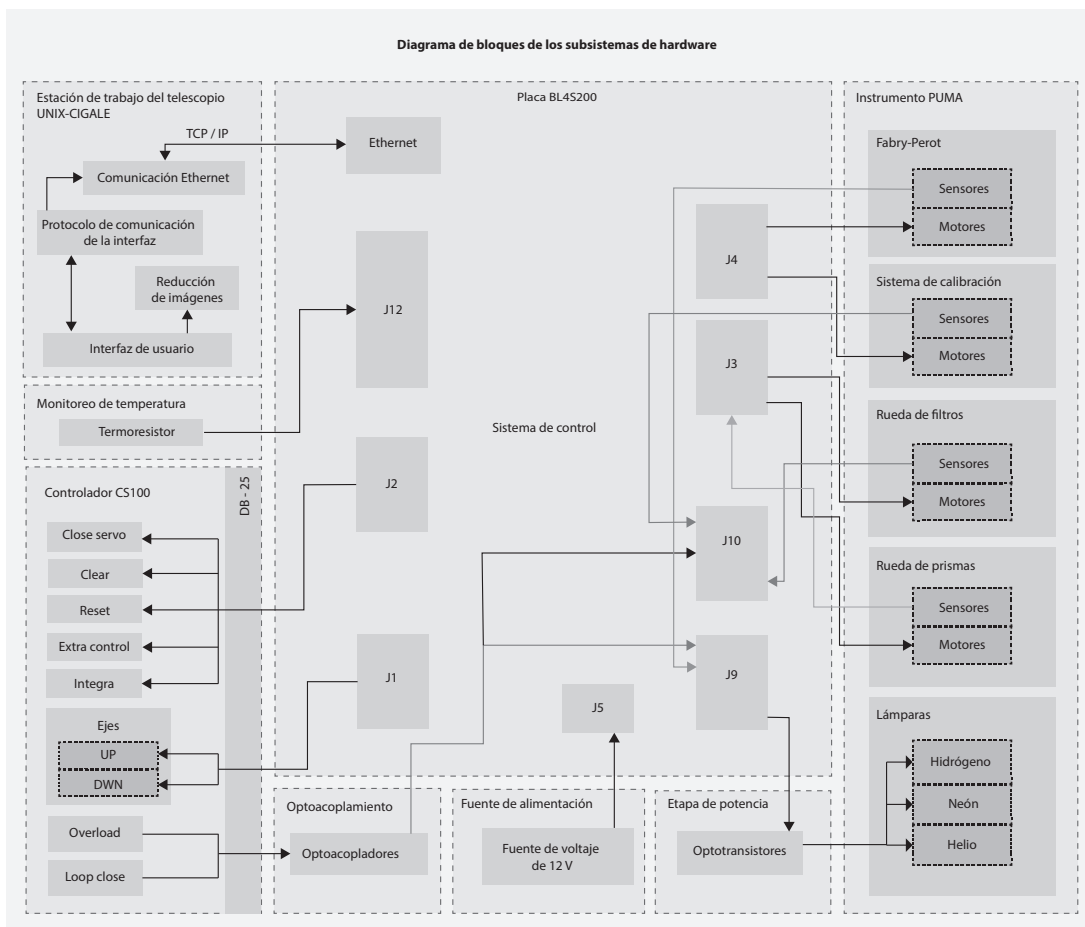
Todas las salidas digitales de alta corriente son utilizadas como un amplificador en configuración de puente H. Cuando el BL4S200 se enciende por primera vez o se reinicia, todas las salidas están desactivadas, es decir, están en un estado de alta impedancia. Cada banco de cuatro salidas de alta corriente tiene su propio suministro de voltaje a +12 V.

4.3.4 Entradas del convertidor analógico digital

La BL4S200 también cuenta con un chip convertidor analógico digital, con una resolución de 12 bits (11 bits para el valor y un bit para la polaridad). El chip convertidor analógico digital tiene un amplificador programable de entrada, que acondiciona la señal y proporciona un filtrado. Las ocho entradas externas se escalan y se filtran para proporcionar una buena impedancia de entrada.

4.3.5 Distribución de E/S para el control del instrumento PUMA

La disposición final de las entradas y salidas del microcontrolador de placa reducida BL4S200 (Cuadro 4.1), se pensaron de forma que cubrieran los requerimientos del instrumento PUMA.

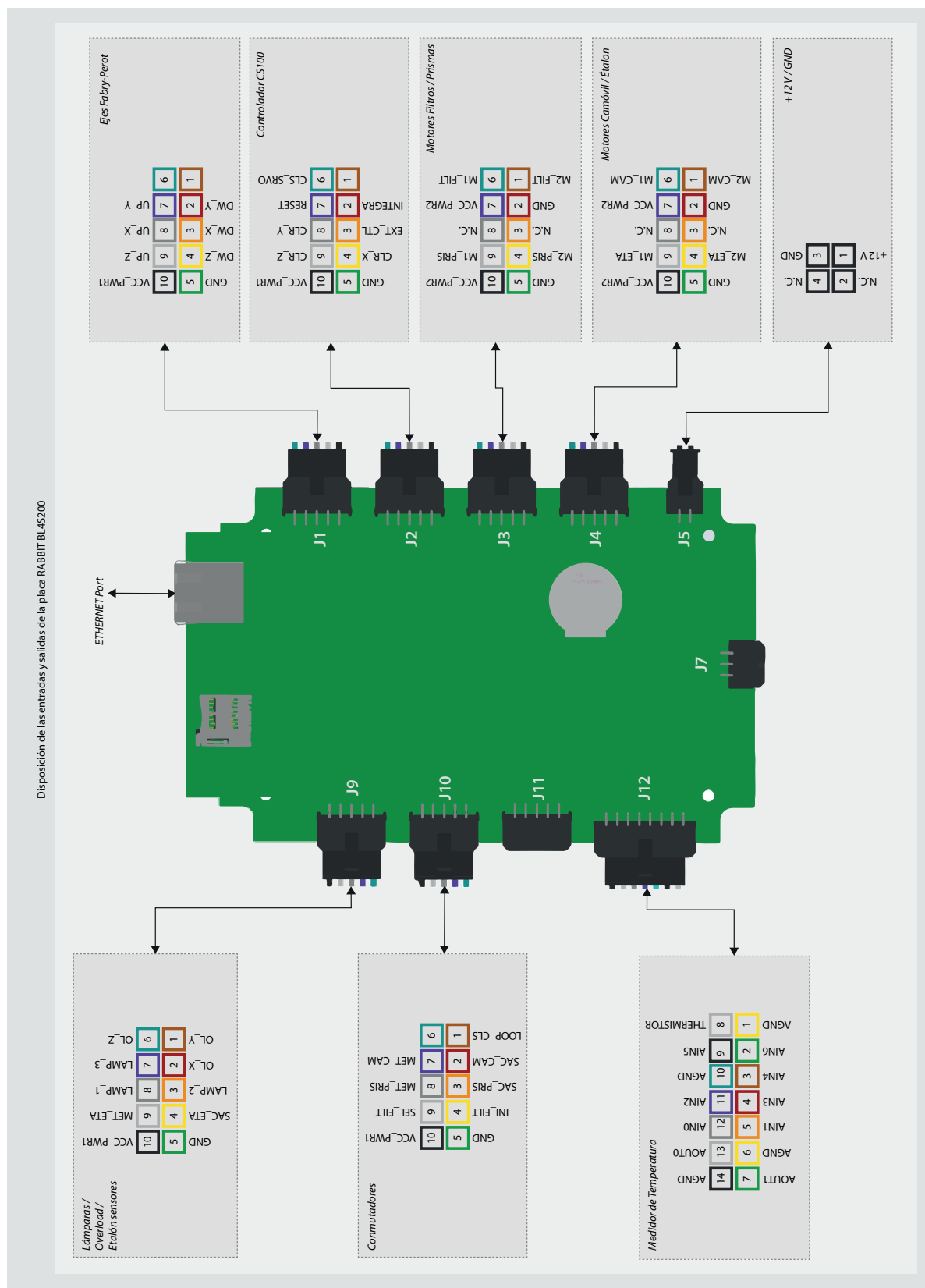


Cuadro 4.1: Distribución de conexiones del sistema de control a los demás subsistemas. *Imagen realizada por Zimri Marsilli.*

Considerando además, los aspectos prácticos del posicionamiento de sensores y actuadores para el control digital.

El conector *J4* (ver Figura 4.7), como uno de los dos conectores de alta corriente, se designó a los motores que controlan a la unidad móvil de calibración y al posicionamiento del interferómetro Fabry-Pérot dentro del camino óptico; el otro conector, *J3* de salidas de alta corriente, se designó para la rueda de filtros y para controlar el movimiento del brazo de prismas. El conector *J2* se usó como salidas de propósito general, para el envío de señales al controlador CS100, mientras que el conector *J1* fue utilizado para controlar la paralelización del interferómetro Fabry-Pérot. El conector *J10* se centra en los sensores de final de carrera y conmutadores para determinar el inicio y fin del movimiento de todos los carros motorizados. Finalmente el conector *J9* controla el encendido de las lámparas de calibración.

El conector RJ-45 es esencial para la comunicación TCP/IP tipo Ethernet, con la estación de trabajo del observatorio. Igualmente se hace uso del conector de alimentación a +12 V de la placa BL4S200.



CAPÍTULO

5

El entorno de desarrollo

En este capítulo se describe brevemente el entorno de desarrollo del Rabbit y se cubren los aspectos del entorno Dynamic C mediante el análisis, diseño y metodología del ambiente de programación del nuevo sistema de control del instrumento PUMA.

Índice

| | |
|-------------------------------------|----|
| 5.1 El software Dynamic C | 58 |
| 5.2 Protocolo TCP/IP | 60 |
| 5.3 Modelo esencial | 62 |

5.1 El software Dynamic C

El comportamiento de un sistema embebido debe ser capaz de responder a las entradas del usuario, tomar una decisión y luego realizar alguna acción en respuesta a un evento; el cual puede tener su origen en el mundo externo o puede ser generado dentro del propio sistema, por lo que al diseñar aplicaciones con sistemas embebidos, se tienen muchas opciones que considerar al seleccionar un controlador, como es el tiempo y la cantidad de código escrito (cargado en el sistema de destino).

Es favorable que el desarrollo del software se reduzca (al igual que la cantidad de hardware implementado), en lugar de gastar tiempo en escribir controladores y código de interfaz de periféricos de bajo nivel y centrarse en el desarrollo de la aplicación de nivel superior, como es la creación de algoritmos del código específico y simplemente realizar las llamadas correctas al conjunto de subrutinas, funciones y procedimientos para el proyecto. El lenguaje C es una buena opción, debido a la disponibilidad del conocimiento del software y a la gran variedad de bibliotecas disponibles.

Las herramientas y desarrollo de software han avanzado, por lo que el mantenimiento del software (también en hardware), puede exigir constantes cambios durante la evolución del proyecto, además de la detección y corrección de errores del software. Es así que un sistema de desarrollo de fácil comprensión en un futuro podrá mantenerse y repararse rápidamente manteniendo la estructura del código base.

Z-World resolvió las exigencias particulares de rendimiento de los sistemas embebidos, empleando diferentes puntos de vista y tecnologías establecidos en su época, como las extensiones y la interpretación creativa de la sintaxis del lenguaje de programación C, compatible con ANSI/ISO C 90, el cual mantiene un estándar para el lenguaje de programación C para el microprocesador Rabbit (con sus respectivas diferencias), satisfaciendo las demandas de los controladores embebidos.

Norm (fundador de Rabbit Semiconductors) durante su establecimiento en California, contrató para Z-World a un estudiante de la Universidad de Davis, Greg Young, quien desarrollaba de manera independiente herramientas en Z80; posteriormente Greg escribió el primer entorno de desarrollo integrado para un nuevo entorno de programación: Dynamic C. Greg empezó como gestor e ingeniero de diseño del grupo técnico y actualmente funge como consultor de la compañía (Hyder and Perrin, 2005).

Dynamic C ha estado en uso desde 1989 (Digi-Semiconductor, 2013b) y es un sistema de desarrollo integrado, que ofrece una herramienta sólida y digna de consideración para cualquier proyecto, dando solución al diseño y escritura de software embebido, optimizando su diseño para la programación C aplicado en los circuitos integrados, placas de control y módulos básicos compactos, basados en los microprocesadores de Rabbit Semiconductors.

El controlador Rabbit mantiene un equilibrio entre la esencia del lenguaje primitivo C y las nuevas tecnologías en el desarrollo del software para nuevos diseños; sin embargo la arquitectura de estos sistemas se apoya en Zilog Z180, pero sin ser compatible a nivel binario con las partes Zilog (Hyder and Perrin, 2005). El conjunto de instrucciones de alto rendimiento ofrece una mayor eficiencia y velocidad de ejecución de código C generado por el compilador.

El sistema de desarrollo Dynamic C está especialmente diseñado para los sistemas de programación embebidos y posee características poderosas de compilación rápida y depuración, también incluye las funciones de edición, enlace y carga durante el desarrollo del programa; proporcionando extensiones al lenguaje C, además de su propio conjunto de librerías de funciones en forma de código fuente, para adaptar controladores estándar a aplicaciones personalizadas y dar soporte a la programación en tiempo real de las entradas y salidas; así mismo puede ser capaz de escribir y compilar en C, en código ensamblador o en herramientas de terceros con un compilador ANSI C, sin salir del entorno de desarrollo de Dynamic C.

La BL4S200 se programa utilizando Dynamic C (Digi-Semiconductor, 2013c), que incluye al sistema operativo en tiempo real del microcontrolador, el protocolo de punto a punto, el sistema de archivos FAT, RabbitWeb, el paquete de seguridad incorporado Rabbit y una biblioteca específica con un cifrado avanzado estándar. Dynamic C se ejecuta en computadoras personales bajo una plataforma Windows Microsoft. Mientras se codifica la aplicación, Dynamic C descargará la imagen ejecutable en el hardware de destino, que son los sistemas Rabbit (sin necesidad de gastar recursos para un emulador), a través de una comunicación con los procesadores Rabbit, por medio de una interfaz de cable a un puerto de programación serie, para descargar el código al módulo principal y ejecutar el código desde la memoria flash conectada al bus de memoria de la CPU.

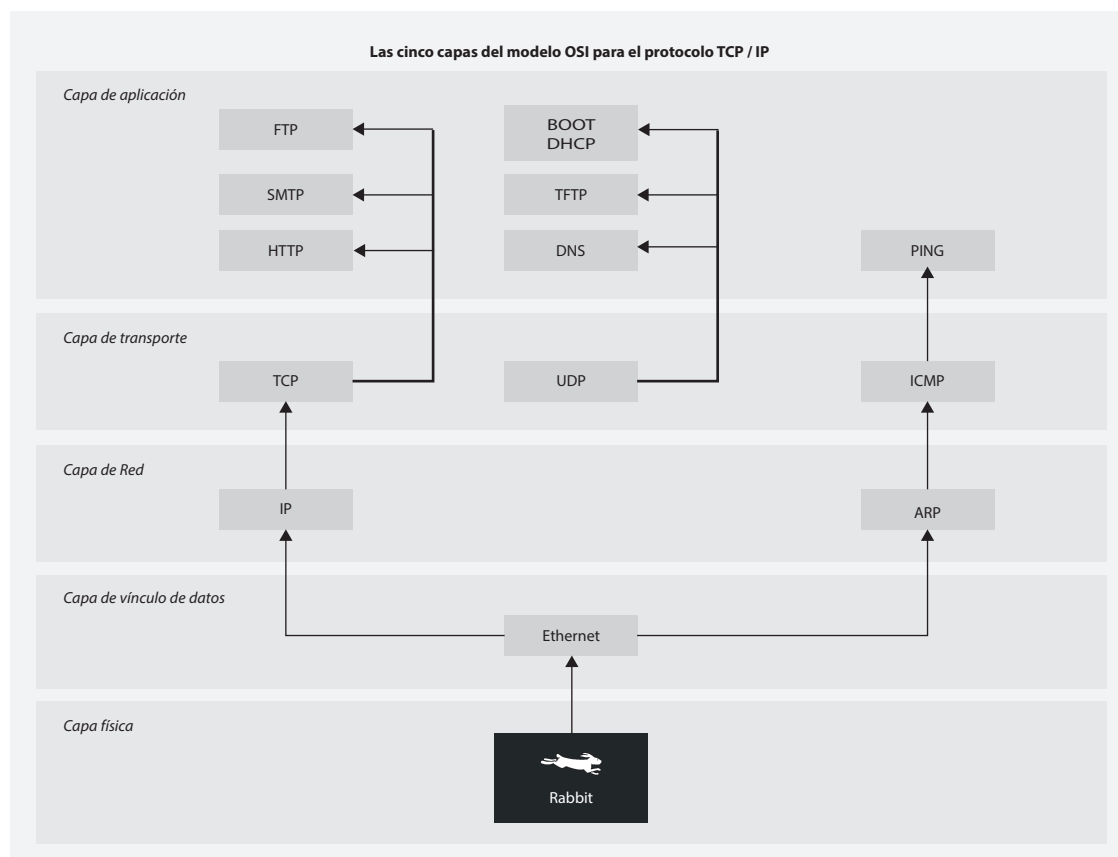
El código final siempre se debe almacenar en la memoria flash para un funcionamiento fiable, también se puede incluir una cantidad significativa en la memoria RAM. Las ventajas de trabajar en la memoria RAM es salvar el desgaste de la memoria flash, que se limita a alrededor de 100,000 ciclos de escritura, la carga del programa será muy rápida, se pueden establecer puntos de interrupción (incluso en lenguaje ensamblador) y puede ser corregido sobre la marcha para probar las rutas de código o de prueba para la detección de errores simples, siendo el código del programa que se está depurando el mismo que se desplegará, por lo que no hay cambios en él o necesidades de recompilación; sin embargo el código y los datos pueden no corresponder al espacio disponible de la RAM.

La depuración en la memoria flash, asigna a la memoria el mismo espacio del programa del código final, así toda la memoria RAM del controlador está disponible; pero el programa tardará más en cargar, porque la flash debe ser borrada y programada en cada carga, además existirán algunos inconvenientes con los puntos de interrupción para depurar el programa y el código adicional necesitará ser compilado en el programa.

El desarrollo de software del módulo RCM4300 debe ser cargado en la memoria SRAM estática, es decir, la aplicación debe ser compilada directamente en la SRAM de datos. Utilizar el código del compilador en la memoria RAM durante la depuración, puede hacer los tiempos de descarga más rápidos, pero se debe volver a compilar y correr desde la SRAM después de que el cable de programación en serie sea desconectado. La depuración en la memoria RAM puede ser utilizada para realizar pruebas de partes de código de la aplicación.

En el nivel de diseño de sistemas se consideran los aspectos que involucran el acceso, rendimiento, fiabilidad, disponibilidad y seguridad. En el programa para el nuevo sistema de control del PUMA se caracterizó al software según el movimiento de las partes mecánicas y el diseño de la interfaz de usuario; así como a la programación del servidor de red.

5.2 Protocolo TCP/IP



Cuadro 5.1: El modelo de interconexión de sistemas abiertos (OSI) tiene siete capas, pero para la aplicación del protocolo TCP/IP se omiten las capas de presentación y sesión, quedando así las capas (empezando por la más baja en la jerarquía y siguiendo hacia la más alta): física, de vínculo de datos, de red, de transporte y de aplicación. *Referencia (Digi-Semiconductor, 2013g).*

Una de las características de Dynamic C (Digi-Semiconductor, 2013g) es que se puede trabajar con comunicación tipo Ethernet y con el protocolo de conexión punto a punto más común: el TCP. Este protocolo se utilizará como un entorno de desarrollo de red en base al Rabbit 4000. El módulo RCM4300 proporciona una conexión directa a una red Ethernet, en un chip de interfaz 10/100 Base-T.

El protocolo de control de transmisión es un conjunto de protocolos independientes, con gran cantidad de servicios de redes, cuyo objetivo principal es proporcionar transferencia de datos orientados a la conexión fiable de extremo a extremo. El TCP es muy eficaz para las aplicaciones, ya que utiliza varios métodos para el control de flujo, de secuenciación y de detección de errores para cumplir su misión, cuando se ordene algo al controlador y deba responder con precisión de tiempo ejecutando las instrucciones solicitadas.

El modelo de interconexión de sistemas abiertos (OSI) de siete capas (Cuadro 5.1), se reduce a cinco capas para su aplicación al protocolo TCP/IP: la capa física (capa 1) especifica el hardware de la red básica; la capa de red (capa 2) especifica la organización, marco y transmisión de datos; la capa de Internet (capa 3) especifica el formato de paquetes y enrutamiento; la capa de trans-

porte (capa 4) especifica como garantizar un transporte fiable de datos y la capa de aplicación (capa5) especifica las aplicaciones particulares utilizadas en una red.

Dynamic C trabaja en la capa física y de enrutamiento, pero su mayor contribución se encuentra en el transporte y la aplicación. Los dos protocolos de transporte principales son el TCP y el UDP. El TCP proporciona una transferencia de datos fiable, mientras que el UDP proporciona un servicio de mejor esfuerzo, pero sin garantía en la recepción y envío de los datos (Digi-Semiconductor, 2013f). Por otra parte el TCP requiere recursos de red más altos que el UDP, es decir que requiere un protocolo de enlace de inicialización entre el emisor y el receptor para mantener una conexión entre dos dispositivos.

El tiempo de llegada de paquetes que se transportan en la red son diferentes debido al enrutamiento y la congestión, lo que provoca que no lleguen en la misma secuencia en que se enviaron. Es por esto que el TCP ordena los datos que se envían, garantizando los paquetes recibidos durante una conexión sincronizada y estable, a través de la conectividad de red. Dependiendo de la aplicación que se desarrolle, la fiabilidad del TCP puede significar una mayor sobrecarga de la aplicación en términos de complejidad de código, tiempo de procesamiento, conexión adicional y sobrecarga de recursos; sin embargo a través de TCP se puede asegurar que el paquete llegue al receptor.

El protocolo TCP garantiza la entrega de los datos en el orden correcto, permitiendo que dos *hosts* puedan establecer una conexión con intercambio de flujo de datos (tratados en bytes). TCP puede detectar errores o pérdida de información, desencadenando la retransmisión hasta que se reciban los datos completos y sin errores.

Dynamic C igualmente proporciona soporte para el protocolo de Internet (IP), proporcionando la comunicación entre *hosts* de diferentes tipos de redes. En este caso la red utiliza direcciones IP para describir sus fuentes y destinos. Se trata de una conexión de servicio de entrega de datos independiente.

El IP define un esquema de direccionamiento que es independiente de la dirección física. Se especifica con un número único de 32 bits para cada *host* en una red, este número se conoce como la dirección de protocolo de Internet, la dirección IP o la dirección de Internet. Cada paquete enviado a través de Internet contiene una dirección IP de origen y una dirección IP de destino. Para el encaminamiento de la eficiencia, la dirección IP se considera en dos partes: el prefijo que identifica la red física y el sufijo que identifica un ordenador de la red. Se necesita un prefijo único para cada dirección IP en la red.

Se necesita una dirección IP para el dispositivo de red del módulo Rabbit, que sea consistente con la red de comunicación. Si esto se hace de forma incorrecta diversos elementos de red podrían no reconocer al módulo principal. Se utilizan dos métodos para configurar una dirección IP de los dispositivos de red, estáticamente y dinámicamente. En el direccionamiento estático las direcciones IP no cambian, a menos que se configuren manualmente. Si el módulo de red del Rabbit se direcciona de esta manera sólo podrá trabajar en un segmento de red específico.

El protocolo de configuración dinámica de *host* (DHCP), permite la asignación dinámica de direcciones IP a los dispositivos. En el contexto de sistemas embebidos, significa que el sistema puede arrancar sin una dirección y posteriormente podrá negociar con un servidor DHCP para recibir una dirección IP. La asignación dinámica de direcciones IP es común, ya que alivia la carga de los administradores de red para asignar y gestionar direcciones IP de forma estática. El direccionamiento dinámico utiliza una entidad externa, un servidor DHCP que maneja un mecanismo de petición y respuesta para asignar direcciones IP de forma dinámica.

La perfecta combinación de hardware y software del Rabbit, junto con su entorno de desarrollo Dynamic C, permitirá al proyecto de esta envergadura que se instale de forma rápida, sencilla y compacta, construyendo una aplicación con alto grado de fiabilidad en la comunicación del control de flujo para evitar la congestión, con detección de errores y retransmisión usando TCP.

5.3 Modelo esencial

En el sistema PUMA el hardware será entonces el microcontrolador de placa reducida BL4S200; sin embargo el software, en este caso, será su programa de control. Para su respectivo análisis se considerará como un sistema computacional de tiempo real, el cual controlará y verificará procesos recibiendo datos de forma remota (desde una terminal), procesándolos y respondiendo con suficiente rapidez, interactuando de forma autónoma con sus subsistemas.

El proyecto simultáneamente puede llevar a cabo procesos de muchas actividades, por lo que es necesario asignarles prioridades diferentes, teniendo en cuenta a las que requieran un servicio inmediato y a las que se les pueden aplazar periodos razonables, interrumpiendo una tarea antes de concluirla para comenzar con una de mayor prioridad, existiendo comunicación entre las tareas de cada proceso sincronizado.

Los principios para la creación del sistema automatizado PUMA son que será optimizado y será capaz de adaptarse a circunstancias nuevas y diferentes, pudiendo transportar el hardware y software a otras distintas situaciones de funcionamiento; diseñando un modelo compacto, para disminuir recursos, mantenimiento, tener seguridad y documentación; tomando en consideración que, como todo sistema, este tiene la naturaleza de crecer con la adición de nuevo código, sensores y actuadores (no hay posibilidad de considerar un sistema como estático). Este proyecto se enfocará en el sistema de control electrónico del sistema PUMA, que funciona en un sistema mayor que es el telescopio y que puede ser modificado con su simple reprogramación.

El modelo esencial del sistema es un modelo lógico de lo que el sistema debe hacer, para satisfacer los requerimientos, aunque la computadora se encontrará en un medio hostil, responderá y no quedará fuera de control, aun así, como es un proyecto con alta exigencia del procesamiento en tiempo real, es importante que no se pierdan los datos de entrada y sus respuestas a los estímulos externos durante el paso del tiempo.

Conociendo las características esenciales y las limitaciones de implementación del nuevo sistema de control y el ambiente del usuario, se analizará el desarrollo del proyecto con herramientas de documentación del modelado esencial del sistema, mediante el análisis estructurado de Yourdon (Yourdon, 1993), para diseñar la programación y construir el sistema.

Las herramientas del análisis estructurado son para describir las funciones del sistema, para el comportamiento de un sistema dependiente del tiempo con un entendimiento y visión general, enfatizando ciertas propiedades críticas respecto de otras y para elaborar un diseño superior y tecnológicamente posible. Como resultado se obtiene un modelo estructural abstracto de un programa de control (y de hardware), que beneficie al usuario al simplificar los cambios y correcciones con riesgo mínimo. Las herramientas consistirán en métodos gráficos adicionales, para facilitar el análisis y mostrar los componentes principales del modelo; así como las conexiones entre los componentes, además se agregan herramientas textuales que definen con precisión el modelo.

5.3.1 Modelo ambiental

En esta sección se definirán las fronteras e interfaces entre el sistema y el resto del entorno, es decir que se modela el ambiente en el que se encuentra el sistema, identificando los acontecimientos que ocurren; así como la información externa que entra y la información de salida que produce. Tomando en consideración solamente aquellas acciones a las que debe responder el sistema, lo que se encuentre de la frontera será lo que se tendrá que automatizar. Este modelo consiste en un diagrama de contexto, en una lista de acontecimientos y en una descripción breve del propósito del sistema.

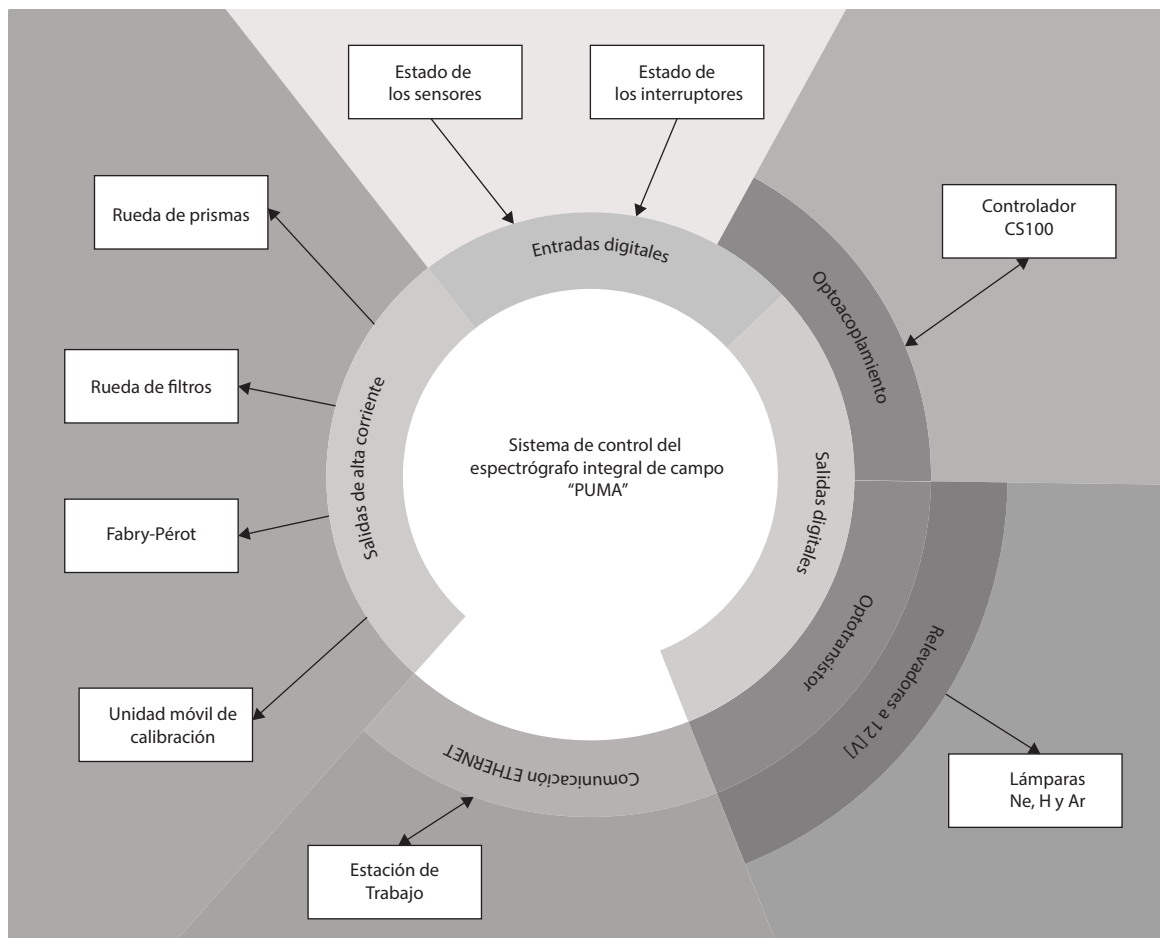
5.3.1.1 Propósito del sistema

El propósito de este nuevo sistema de control es volverlo más eficiente, productivo, corregible, portable y mantenible a la hora de procesar los movimientos de los carros móviles del instrumento espectroscópico PUMA y controlar el paralelismo de su interferómetro Fabry-Pérot de barrido, por medio de su controlador CS100. Todo esto por medio de electrónica digital y a través de una interfaz de usuario, que se opera en una estación remota de trabajo dentro de la red del observatorio, para el procesamiento de imágenes astronómicas.

5.3.1.2 Diagrama de contexto

El sistema de control del instrumento PUMA se representa como un solo proceso (Cuadro 5.2), y se definen las fronteras e interfaces entre el sistema y el resto de los componentes externos, como son los elementos motorizados (brazo de prismas, rueda de filtros, interferómetro), el controlador de paralelización (controlador CS100), el conjunto de sensores finales de carrera y de proximidad, la unidad móvil de calibración (lámparas de calibración y el espejo móvil) y la retroalimentación con la interfaz gráfica del usuario.

Los datos que el sistema recibe del exterior y que deben procesarse de alguna forma son las señales de los sensores e interruptores de los carros móviles, los cuales verifican el estado de su posición, también se debe verificar el estado del controlador CS100 y la constante comunicación con la interfaz (operada en la estación de trabajo). Los datos que el sistema produce y que se envían al exterior son las señales para activar los motores de los carros móviles y las señales para encender las lámparas.



Cuadro 5.2: En este diagrama de contexto una burbuja en el centro representa a todo el sistema y lo delimita de su entorno. Definiendo los elementos de control y a controlar que interactúan con el sistema, así como los flujos de datos que fluyen entre sí, como entradas o como salidas.

5.3.1.3 Lista de eventos

Para describir los acontecimientos de control vistos desde el punto de vista del entorno, los eventos serán representados por mnemónicos, los cuales son necesarios para el entendimiento de la comunicación vía red entre la interfaz gráfica del usuario y el instrumento (Cuadro 5.3).

El control del CS100 se realiza por medio del bus de control, que permite el control remoto o control computarizado del sistema (el primer ajuste y calibración se realizan por medio de los controles del panel frontal). Los ejes *z*, *x* e *y* (sensores piezoeléctricos) del interferómetro Fabry-Pérot, se balancean con señales enviadas al CS100, seguido de un número entre -2047 y +2048, que indicará el número de pasos que las placas paralelas del interferómetro avanzará o retrocederá respecto de una posición previa. Cuando se capture un nuevo cubo de datos o se reinicie el controlador CS100, se tendrán que limpiar los valores de cada eje, para esto bastará la instrucción *c* e indicando el eje a vaciar, iniciando el nuevo conteo a partir del valor cero.

El reinicio se utiliza principalmente cuando se cambian los controles de constante de tiempo en el modo de lazo cerrado y se vuelven a cerrar los lazos cuando se han abierto por el circuito de

| Lista de eventos | | | | | |
|----------------------|-------------------------------------|-------------------------------------|---------|------------------------------------|---------|
| Evento | Proceso | Entrada | Tipo | Salida | Tipo |
| Instrucción z s#### | Aumenta z Disminuye z | NA | NA | Escritura DIO16 Escritura DIO17 | Externo |
| Instrucción x #### | Aumenta x Disminuye x | NA | NA | Escritura DIO18 Escritura DIO19 | Externo |
| Instrucción y s#### | Aumenta y Disminuye y | NA | NA | Escritura DIO20 Escritura DIO21 | Externo |
| Instrucción c(eje) | Limpia z Limpia x / Limpia y | NA | NA | Escritura DIO24 / DIO25 / DIO26 | Externo |
| Instrucción r | Reset del CS100 | NA | NA | Escritura DIO28 | Externo |
| Instrucción f# | Seleccionar filtro | Lectura de interrupción DIO0 | Externo | Escritura HOUT0 Escritura HOUT1 | Externo |
| Instrucción i | Iniciar rueda de filtros | Lectura de interrupción DIO1 | Externo | Escritura HOUT0 | Externo |
| Instrucción e# | Sacar etalón Meter etalón | Lectura de DIO9 Lectura de DIO8 | Externo | Escritura HOUT6 Escritura HOUT7 | Externo |
| Instrucción p## | Sacar prismas Meter prismas | Lectura de DIO2 Lectura de DIO3 | Externo | Escritura HOUT2 Escritura HOUT3 | Externo |
| Instrucción l## | Prender lámparas Apagar lámparas | NA | NA | Escritura DIO10 / DIO11 / DIO12 | Externo |
| Instrucción v## | Sacar camóvil Meter camóvil | Lectura de DIO4 Lectura de DIO5 | Externo | Escritura HOUT4 / HOUT5 | Externo |
| instrucción s | Ver estado del sistema | NA | NA | Cadena enviada | Interno |
| Instrucción FIN | Salir del sistema | Cerrar comunicación ETHERNET | interno | Cierre del sistema | Interno |
| Instrucción CLS_SRV | Cerrar Lazo | Lectura de interrupción DIO7 | Interno | Escritura DIO7 | Externo |
| Instrucción OL_(eje) | Apagar interruptores del servo | Lectura de DIO13 / DIO14 / DIO15 | Externo | Escritura DIO13 / DIO14 / DIO15 | Externo |
| Instrucción Integra | Escritura DIO29 | NA | NA | Escritura DIO29 | Externo |
| Instrucción EXT_CTRL | Usar control externo | NA | NA | Escritura DIO27 | Externo |

Cuadro 5.3: En la tabla se describen los acontecimientos a los cuales debe responder el sistema. En donde evento indica que produce un suceso, proceso indica el proceso que se va a desencadenar en el sistema al llegar a ese evento, entrada indica los datos que van a llegar al proceso, salida indica los datos que van a salir del proceso como respuesta del sistema al evento y tipo indica si es de tipo externo (E) o de tipo interno (I).

protección, es decir, por problemas de comunicación o retroalimentación del controlador CS100. Esta función inicializa de nuevo el sistema (independientemente de los ajustes del panel frontal), y descarga los condensadores de los detectores sensibles a la fase y del circuito integrador.

Para el proceso de calibración y captura de interferogramas, es necesario que el sistema de control conozca el estado del instrumento. Se debe inicializar la rueda de filtros con la función *i*. Si se desea seleccionar alguno de los filtros dispuestos se deberá llamar la función *f* seguido del número de la posición. La selección y encendido de alguna de las tres lámparas de calibración se realiza mediante la instrucción *l* seguida de dos bits, los cuales indicarán el estado y el tipo de lámpara.

Los demás dispositivos: el interferómetro (e), la rueda de prismas (p) y la unidad de calibración móvil (v), son las instrucciones encargadas de accionar los elementos y colocarlos o moverlos fuera o dentro del tren óptico, dependiendo del objetivo de la observación.

Para proceder la toma del cubo de datos se requiere verificar el estado del instrumento y la posición de cada uno de sus elementos motorizados mediante la instrucción s . Al encenderse el instrumento se prepara con la instrucción inicializar para su posterior configuración. Además se agrega una función de finalizar para mejorar la confiabilidad de operación.

En caso de que el puente de capacitancia del CS100 se encuentre acoplado y en retroalimentación con el interferómetro, el lazo de regulación estará cerrado, para ello también se encuentra la función *LOOP CLS*. Cuando se activa esta función, inmediatamente se acciona un reinicio para inicializar el sistema después de un retraso de alrededor de medio segundo, el suministro de alta tensión a los amplificadores de almacenamiento del transductor estará activado y después de medio segundo más, el reinicio se liberará para permitir el funcionamiento de lazo cerrado correcto.

La función integrar modifica la función de transferencia de lazo para eliminar cualquier error estático residual, debido a perturbaciones mecánicas a bajas ganancias, es un interruptor dentro del diagrama del circuito del amplificador e integrador. La función integrar a altas ganancias no afecta al sistema notablemente.

Durante el funcionamiento normal de lazo cerrado, las señales en las salidas de los amplificadores de corriente alterna serán pequeñas (<1 V p-p). Las funciones de sobrecarga se pueden desencadenar por una potencia superior a 15 V p-p. Dichas señales se generarían por la oscilación violenta del sistema, debido a un incorrecto ajuste de constantes de tiempo y ganancia o por un gran desequilibrio resistivo de configuración o condiciones generadoras de tales fallos, como una conexión del Fabry-Pérot en circuito abierto o señales en corto.

En la instrucción de cerrar servo, si el circuito de protección detecta las oscilaciones debido a la configuración incorrecta de los controles de ganancia y de constantes de tiempo, abre el lazo de servo de forma automática, evitando posibles daños al FP.

5.3.2 Modelo de comportamiento

El modelo de comportamiento describe lo que el sistema de tiempo real requiere para que interactúe de manera exitosa. Se trata del interior del sistema e incluye los aspectos críticos a modelar: un diagrama de flujo de datos (que describirá las funciones), un diccionario de datos (que describirá los datos), la especificación de procesos (que describirá los procedimientos) y un diagrama de transición de estados (que describirá los tiempos).

Para que el análisis y diseño del sistema sea completo, correcto y consistente; siendo su aplicación mantenible, cuando se necesiten cambiar o agregar más funciones de los dispositivos que actualmente coexisten en el instrumento y evolucione junto a los requerimientos que él mismo exija, durante el transcurso de tiempo de su operación.

5.3.2.1 Diagrama de flujo de datos

En la comunicación vía red TCP usada por el sistema operacional, las funciones que debe desempeñar el sistema, su interacción, su transformación y el destino de sus resultados son de gran importancia. Con la intención de permitir una considerable flexibilidad para elegir mejor el estudio y la implementación de las exigencias directas de las partes del sistema, siendo este más eficiente y más confiable.

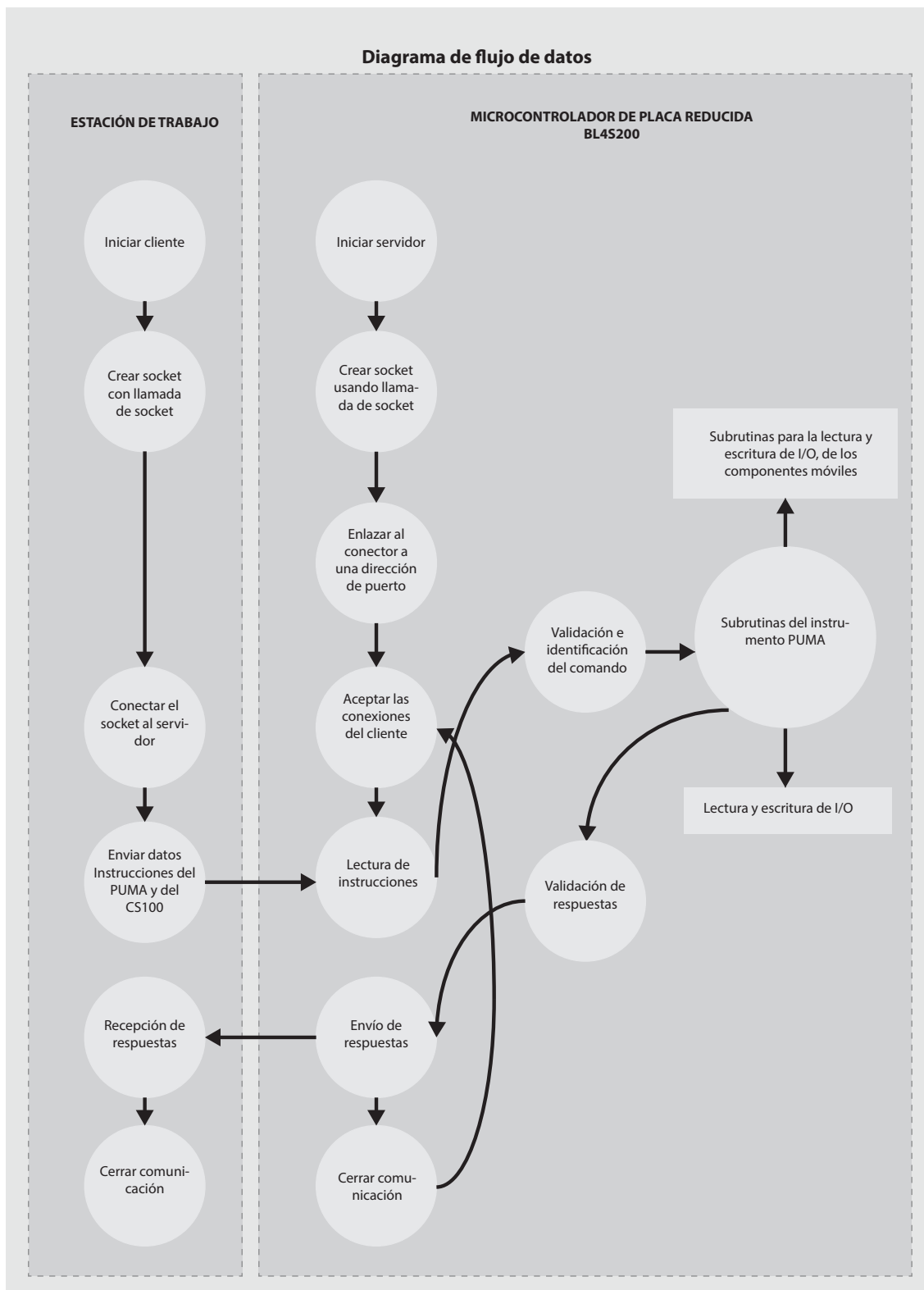
En el diagrama de flujo (Cuadro 5.4) se representan esquemáticamente las decisiones en secuencia de actividades, para llevar a cabo ciertos procedimientos e ilustrar las funciones que el sistema deberá realizar para transformar las entradas en salidas. Utilizar esta herramienta permitirá entender y analizar la correlación entre los procesos que están internamente comunicados, comprendiendo de una mejor manera a todos los componentes funcionales del sistema.

El modelo gráfico-teórico se conformará de procesos de control, que representarán las diversas funciones individuales que el sistema llevará a cabo; se traza como una burbuja, cuyo objetivo es comunicarse y organizarse con otras funciones dentro del diagrama de flujo. Los flujos de control serán las señales de control y representarán la información que dichos procesos requerirán como entrada o la información que generarán como salida, describirán el movimiento de los datos (bits, caracteres, mensajes o números en punto flotante) de una parte del sistema a otra. Los flujos pueden traducirse como la tecnología que se usará para implantar el sistema. Los terminadores mostrarán las entidades externas con las que el sistema se comunicará y representarán entidades con las cuales el sistema se comunicará fuera del control del sistema, estando fuera del dominio del cambio (puede ser otro sistema computacional con el cual se comunica éste mediante una interfaz entre él y el mundo externo).

Para el establecimiento de la conexión entre la estación de trabajo y la placa BL4S200, se establece un protocolo de comunicación cliente-servidor. El cliente será la estación de trabajo; se encargará de la comunicación en general e iniciará la conexión solicitando servicios cuando sea necesario y el servidor TCP, en este caso, será el microcontrolador de placa reducida, responderá estando a la escucha por más solicitudes entrantes para comenzar la operación. Aunque estos dos sistemas envían y reciben los mismos datos, se tiene que especificar cómo el cliente y el servidor se comunicarán, de manera que los bytes de datos no consigan intercambiarse y perderse de forma inesperada.

Un cliente TCP exige un orden de secuencia para que se pueda establecer una conexión con un servidor, además desde una perspectiva de sistemas embebidos, se debe ser claro acerca de otros aspectos de la comunicación. Ambas partes tienen que utilizar los mismos protocolos y tienen que saber qué conexiones disponer para saber con quien comunicarse, quien inicia la comunicación, qué datos necesitan ser intercambiados, el formato de los datos, la forma en la petición, cómo se estructurará la respuesta y cuáles son los tiempos de espera. Lo anterior es fundamental para evitar condiciones en las que el cliente y el servidor estén indefinidamente a la espera de una respuesta de la otra parte.

Antes de que un cliente o servidor puedan hablar uno con el otro, se necesita crear un *socket*. Los *socket* de flujo se utilizan para las comunicaciones TCP, identificando los dispositivos por su nombre o dirección IP y el puerto asignado para establecer la conexión. Es decir, un *socket* es una combinación de una dirección IP y un número de puerto. Los números de puerto asignados por



Cuadro 5.4: Se muestra el diagrama de flujo de datos del programa para el control del instrumento PUMA y el controlador CS100 incluyendo la conexión a red.

el dispositivo de hardware encargado de la comunicación, pueden ser puertos específicos, conocidos y utilizados por las aplicaciones de red o números de puerto no utilizados que se pueden

asignar dinámicamente por la aplicación.

El TCP es un protocolo orientado a la conexión, una conexión que debe ser establecida antes de que un *socket* de flujo pueda enviar o recibir datos. Una vez que la conversación básica ha tenido lugar, ambas partes pueden continuar con la comunicación de dos vías, tanto desde el lado del servidor como del lado del cliente.

La conexión TCP/*socket* es de forma pasiva cuando el proceso de conexión TCP se realiza mediante un acuerdo entre un cliente y un servidor. El cliente solicita una conexión mediante el envío de un segmento TCP, el servidor responde con su propio segmento, el cliente reconoce el segmento del servidor y a continuación, se establece la conexión identificada de forma única por cuatro números, llamada *socket*: la dirección IP de destino, el número de puerto de destino, la dirección IP de origen y el número de puerto de origen para el establecimiento y transcurso de la conexión.

En el sistema de control del PUMA, se establece una comunicación ETHERNET del tipo TCP. La estación de trabajo corre los clientes necesarios para comunicarse con el microcontrolador, que está programado como servidor. El servidor tiene creado un *socket*, con una dirección y una dirección IP, que entra en estado activo cuando el cliente solicita una conexión. Una vez conectados, se lleva a cabo una comunicación de dos vías y el cliente puede enviar al servidor los comandos que activarán las distintas subrutinas en el programa de control del instrumento.

En esta comunicación de dos vías es importante definir cómo el cliente será la señal de inicio de la comunicación y cómo éste enviará la instrucción al servidor de conexión a través de un par de estados, esto se hace mediante el reconocimiento de una cadena de caracteres. Ambas partes deben ponerse de acuerdo en cuanto termine la comunicación. En este caso se considera suficiente suponer que la comunicación se termina en el momento de detección de una respuesta, compuesta por una determinada secuencia de bytes, enviados desde una parte a otra y que indica que se ha terminado la operación deseada y que procede a desconectar la comunicación. De esta manera el sistema se puede recuperar de una desconexión esperada e inesperada.

Durante la configuración de red la dirección IP se configura manualmente; sin embargo cuatro piezas de información son necesarios para cualquier *host* en una red. La dirección IP de la máquina (MY_IP_ADDRESS). La parte que distingue a la máquina en la red local de otras máquinas y de otras redes se llama máscara de sub-red (MY_NETMASK), la cual es una combinación de bytes para delimitar el ámbito en una red de computadoras, su función es indicar qué partes de la red identificadas en la dirección IP son accesibles al dispositivo. La puerta de enlace predeterminada (MY_GATEWAY) es la dirección de la puerta de enlace en la red local y la dirección del servidor asignada por un dominio (MY_NAMESERVER).

Los números de puerto van de 0 a 65.535; no obstante se tiene que asegurar que la aplicación utilice el número de puerto que está permitido por el sistema con el que se trabaja. A continuación se muestran los cuatro elementos del programa de control del instrumento PUMA, para el establecimiento de la red local.

```
1 #define TCPCONFIG 1
3 #define _PRIMARY_STATIC_IP "192.168.0.19"
  #define _PRIMARY_NETMASK "255.255.255.0"
5 #define MY_NAMESERVER "132.248.10.2"
  // #define MY_NAMESERVER "10.0.4.2"
7 #define MY_GATEWAY "192.168.0.254"
9 #define PORT 12007
```

5.3.2.2 Diagrama de transición de estados

El diagrama de transición de estados (Cuadro 5.5), es para construir un sistema con tecnología ideal. Un estado representa algún comportamiento del sistema y perdurará durante algún periodo finito de tiempo, durante el cual el sistema puede sufrir un cambio de estado o transiciones de un estado a otro, por una o más condiciones (sucesos o circunstancias que propicien el cambio de estado), asociadas a una, más de una o tal vez ninguna acción, es decir, respuestas, salidas o actividades que se llevan a cabo como parte del cambio de estado.

Con intención de mejorar el control de procesos del sistema construido y mantener el control sobre los dispositivos del instrumento, el sistema maneja fuentes externas de datos a altas velocidades, además de proporcionar respuestas y datos de salida de manera eficiente y rápida para manejar el ambiente externo.

Este sistema de tiempo real en la ingeniería, tiene complejas relaciones en el tiempo que deben configurarse tan cuidadosamente como las funciones, ya que debe responder dentro de un tiempo breve, de tan sólo unos milisegundos, a ciertas entradas provenientes del ambiente exterior y debe estar preparado para recibir diversas combinaciones y secuencias de entradas a las cuales debe responder adecuadamente.

En el programa de control, el comportamiento dependiente del tiempo del sistema y la secuencia con la que se ejecutarán las funciones y actividades, dominarán sobre cualquier otro factor del sistema. Es por esto que se analizará con el diagrama de transición de estados, una herramienta de alto nivel, que sirve como especificación de los procesos para todo el sistema. Se representa una especificación de proceso para una burbuja de control del diagrama de flujo (Yourdon, 1993).

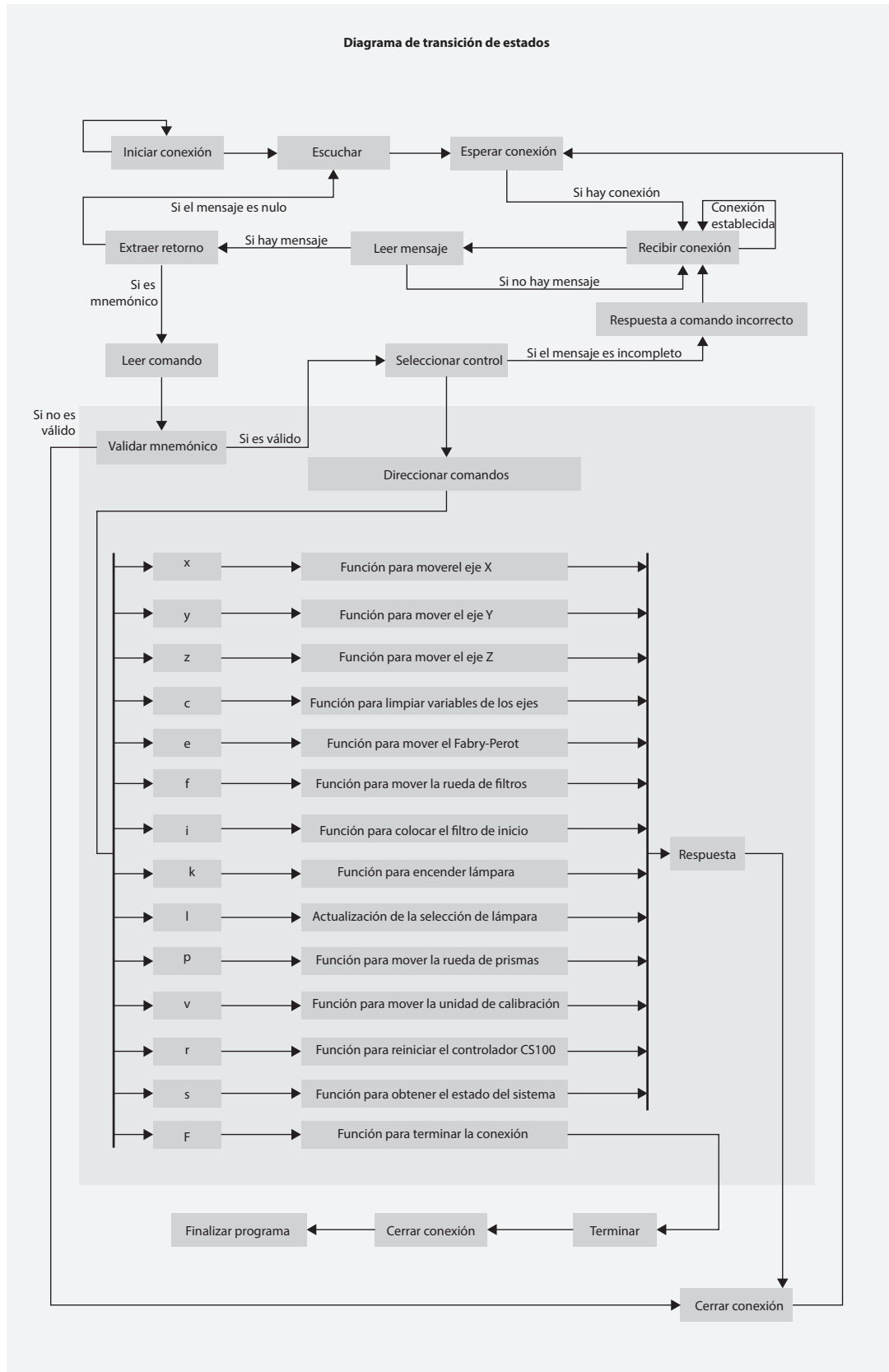
Después de la detección de comandos la instrucción será recibida por parte de la interfaz gráfica del usuario, reconociendo el tipo de menmónico válido para ejecutar la función que le corresponda, para finalmente cerrar conexión y volver a comunicarse con la estación de control.

5.3.2.3 Diccionario de datos

Para el control de las variables (con las cuales se representará de manera abstracta a las funciones y elementos del sistema), y para formalizar el inicio de la escritura de código, se hará un listado estructurado de todos los datos adecuados al sistema, llamado diccionario de datos; con definiciones específicas y detalladas, para tener una comprensión de las entradas, salidas y cálculos intermedios durante la codificación del programa de control.

El diccionario de datos (Cuadro 5.6) y la especificación de procesos (como herramientas complementarias de modelado de tipo textual), puntualizan como los datos se modifican durante el transcurso del programa de control.

Para la configuración de variables, el término estática significa que los datos se encuentran en un lugar fijo permanente para la vida del programa. El término auto, por lo tanto se refiere a las variables que se colocan en la pila del sistema para la vida de una llamada de función.



Cuadro 5.5: Diagrama de transición de estados

| Diccionario de datos | | | | | | |
|----------------------|---------------|--------------|--------|-----------------------------|---|--------|
| Posición | Atributo | Tipo de dato | Tamaño | Valor inicial | Descripción | Acceso |
| 1 | leeComando | int | | const char *control | Apuntador mutable a la cadena inmutable | global |
| 2 | selector | int | | | Parámetro de referencia de tipo entero | global |
| 3 | mSdelay | int | | | Parámetro de referencia de tipo entero | global |
| 4 | posicion_X | int | | | Parámetro de referencia de tipo entero | global |
| 5 | posicion_Y | int | | | Parámetro de referencia de tipo entero | global |
| 6 | posicion_Z | int | | | Parámetro de referencia de tipo entero | global |
| 7 | Filtro | int | | | Parámetro de referencia de tipo entero | global |
| 8 | lampara | int | | | Parámetro de referencia de tipo entero | global |
| 9 | estado | int | | | Parámetro de referencia de tipo entero | global |
| 10 | disp | char | | | Parámetro de referencia de tipo entero | global |
| 11 | modo | in | | | Parámetro de referencia de tipo entero | global |
| 12 | bufSaca | char | [2] | | Variable de tipo caracter | global |
| 13 | buffComandFP | char | [20] | | Variable de tipo caracter | global |
| 14 | respuesta | static char | [100] | "" | Parámetro de referencia de tipo entero | global |
| 15 | comando | int | | 0 | Parámetro de referencia de tipo entero | global |
| 16 | termina | int | | 1 | Parámetro de referencia de tipo entero | global |
| 17 | eje | int | | 0 | Parámetro de referencia de tipo entero | global |
| 18 | Z | static int | | 0 | Variable ubicada estáticamente | global |
| 19 | Z_ant | static int | | 0 | Variable ubicada estáticamente | global |
| 20 | X | static int | | 0 | Variable ubicada estáticamente | global |
| 21 | X_ant | static int | | 0 | Variable ubicada estáticamente | global |
| 22 | int Y | static | | 0 | Variable ubicada estáticamente | global |
| 23 | Y_ant | static int | | 0 | Variable ubicada estáticamente | global |
| 24 | C | static int | | 1 | AVariable ubicada estáticamente | global |
| 25 | CX | static int | | 0 | Variable ubicada estáticamente | global |
| 26 | CY | static int | | 0 | Variable ubicada estáticamente | global |
| 27 | CZ | static int | | 0 | Variable ubicada estáticamente | global |
| 28 | j | static int | | | Variable ubicada estáticamente | global |
| 29 | Filtro | int | | 1 | Variable ubicada estáticamente | global |
| 30 | Lamp | int | | 1 | Variable ubicada estáticamente | global |
| 31 | lampara | int | | 0 | Variable ubicada estáticamente | global |
| 32 | Edo | int | | 0 | Variable ubicada estáticamente | global |
| 33 | Etalon | int | | 1 | Variable ubicada estáticamente | global |
| 34 | Prismas | int | | 1 | Variable ubicada estáticamente | global |
| 35 | Cmovil | int | | 1 | Variable ubicada estáticamente | global |
| 36 | Lazo | int | | 1 | Variable ubicada estáticamente | global |
| 37 | duty | static int | | [NUM_PWM] | Apuntador mutable a la cadena inmutable | global |
| 38 | bp_handle0 | static int | | | Variable ubicada estáticamente | global |
| 39 | bp_handle1 | static int | | | Variable ubicada estáticamente | global |
| 40 | bp_handle6 | static int | | | Variable ubicada estáticamente | global |
| 41 | bp_handle7 | static int | | | Variable ubicada estáticamente | global |
| 42 | button_event0 | static char | | | Variable ubicada estáticamente | global |
| 43 | button_event1 | static char | | | Variable ubicada estáticamente | global |
| 44 | button_event6 | static char | | | Variable ubicada estáticamente | global |
| 45 | button_event7 | static char | | | Variable ubicada estáticamente | global |
| 46 | bytes_read | int | | | Apuntador mutable a la cadena inmutable | local |
| 47 | cntl | int | | | Apuntador mutable a la cadena inmutable | local |
| 48 | result | int | | | Apuntador mutable a la cadena inmutable | local |
| 49 | buffer | char | [80] | | Apuntador mutable a la cadena inmutable | local |
| 50 | extraebuff | char * | | | Apuntador mutable a la cadena inmutable | local |
| 51 | i | int | | | Apuntador mutable a la cadena inmutable | local |
| 52 | Valida | char | [71] | x,y,z,c,e,f,i,k,l,p,v,r,s,F | Apuntador mutable a la cadena inmutable | local |
| 53 | Buff1 | char | [4] | | Apuntador mutable a la cadena inmutable | local |
| 54 | n | int | | | Apuntador mutable a la cadena inmutable | local |

Cuadro 5.6: Diccionario de datos

```

1 tcp_Socket socket;

3 ////////////////buffers requeridos
char bufSaca [2]; //buffer para obtener valores posicion
5 char buffComandFP [20]; //buffer para valor de comando del Fabry-Perot

7 static char respuesta [100]=""; //Respuesta enviada al socket

9 int comando=0;
int termina=1;
11 int eje=0;

13 ////////////////Varibles de posicion de todos los componentes//
static int Z=0;
15 static int Z_ant=0;
static int X=0;
17 static int X_ant=0;
static int Y=0;
19 static int Y_ant=0;
static int C=0;
21 static int CX=0;
static int CY=0;
23 static int CZ=0;
static int j;
25 static int Filtro=0;
static int Lamp=1;
27 static int Edo=0;
static int Etalon=1;
29 static int Prismas=1;
static int Cmovil=1;
31 static int Lazo=0;

33 ////////////////Varibles de posicion para interrupciones//
static int bp_handle0;
35 static int bp_handle1;
static int bp_handle6;
37 static int bp_handle7;

39 static char button_event0;
static char button_event1;
41 static char button_event6;
static char button_event7;
43
static char_buffer [80]=""; //Buffer para lectura de Socket

```

5.3.2.4 Especificación de procesos

Para construir detalladamente el modelo del sistema se hará uso de la herramienta de especificación de procesos, es decir, el uso de módulos ordenados (en forma de subrutinas o funciones) para modelar y estructurar la codificación. Simultáneamente al análisis de especificación de procesos, se empezará a escribir el código del programa de control. El programa completo se encuentra en el Apéndice A

La programación estructurada de la lógica del programa (las decisiones y ciclos), se organiza en módulos pequeños, con sencillez de estilo en combinaciones anidadas de construcción *if*, *else*, *do* y *while*. El código de programación reduce su complejidad al dividirse en trozos más pequeños. Un modelo de estratificación ayuda a esta división y proporciona la base conceptual para la comprensión de cómo los protocolos de software (junto con los dispositivos de hardware), se tienen que integrar.

Durante la primera configuración se llama a la librería *dcrtcp.lib*, que maneja las funciones de red y la librería para el uso del hardware de la placa BL4S200.

```
#class auto
2
#memmap xmem
4 #use "dcrtcp.lib"
#use "BLxS2xx.lib"
```

Se definen los prototipos de las funciones empleadas.

```
1 int leeComando(const char *control);
void selecciona(int selector);
3 void Delay1ms(int mSdelay);
void Eje_X(int posicion_X);
5 void Eje_Y(int posicion_Y);
void Eje_Z(int posicion_Z);
7 void IniciaFiltros(void);
void RuedaFiltros(int Filtro);
9 void Lamparas(void);
void EnciendeLamp(int estado);
11 void Dispositivo(char disp, int modo);
void Clear(void);
13 void Reset_CS100(void);
void Status(void);
15 void Inicializando(void);
void LoopCls();
17 void Butea(void);
void Finalizar(void);
19 void Overload(void);
```

Se configuran los servicios de interrupción, se generan sus banderas y se limpian para trabajar con ellas.

```
1 root void button_press0()
{
3   button_event0 = 1;
   RSB_CLEAR_ALL_IRQ(bp_handle0);
5 }

7 root void button_press1()
{
9   button_event1 = 1;
   RSB_CLEAR_ALL_IRQ(bp_handle1);
11 }

13 root void button_press6()
{
15   button_event6 = 1;
   RSB_CLEAR_ALL_IRQ(bp_handle6);
```

```

17 }
19 root void button_press7()
20 {
21     button_event7 = 1;
22     RSB_CLEAR_ALL_IRQ(bp_handle7);
23 }

```

Programa principal. En el programa principal (Cuadro 5.7), la función `main()` inicializa por primera vez la pila DCRTCP.LIB TCP/IP con una llamada a `sock_init()`, la cual inicializará las estructuras de datos internos y activará el chip Ethernet.

La función `brdInit` se llama en el comienzo del programa de control, para configurar el sistema de puertos de las entradas y salidas del microcontrolador BL4S200. Se prosigue con la configuración de los canales de salida (0-31), como salidas digitales inicializándolas con lógica 0, es decir, en este estado se comienza conectando la carga a GND.

La llamada a `addISR`, añade un manejador de interrupciones especificadas en el parámetro IER para el bloque RIO dado, que aloja el pin de entradas y salidas configurables dados. La rutina de servicio de interrupción (ISR) siempre se desactiva cuando se crea y es responsable de la limpieza de la interrupciones dentro del alojamiento en el bloque RIO cuando se le llama. Se configura para un bloque de entrada el reconocimiento del evento `button_press` para las canales 0,1,6 y 7.

Las interrupciones externas, sobre los pines configurables de las entradas y salidas de la placa BL4S200, se configuran mediante la llamada a la función `setExtInterrupt()`. La interrupción se puede configurar para que ocurra en un flanco de subida, un flanco de bajada o por estado. En este caso, se utilizan los parámetros de el evento de interrupción en el flanco ascendente `BL_IRQ_RISE` y el evento de interrupción en el flanco descendente `BL_IRQ_FALL`. La función `enableISR` está configurada en un inicio (0), para desactivar el manejador de la rutina de interrupción.

La llamada a la función `digOutTriStateConfig_H()`, configura las salidas de alta corriente en modo de alta impedancia (*tri-state*), para los puentes H.

Es fundamental llamar a la función `sock_init()` antes de proceder a la creación del *socket*, o llamar a cualquiera de las funciones que se relacionan con las funciones relativas a la comunicación ETHERNET. Por otra parte, el código debe comprobarse para asegurarse de que la llamada a `sock_init()` ha sido un éxito; un valor de retorno de 0 indica que la llamada a `sock_init()` se ha realizado correctamente.

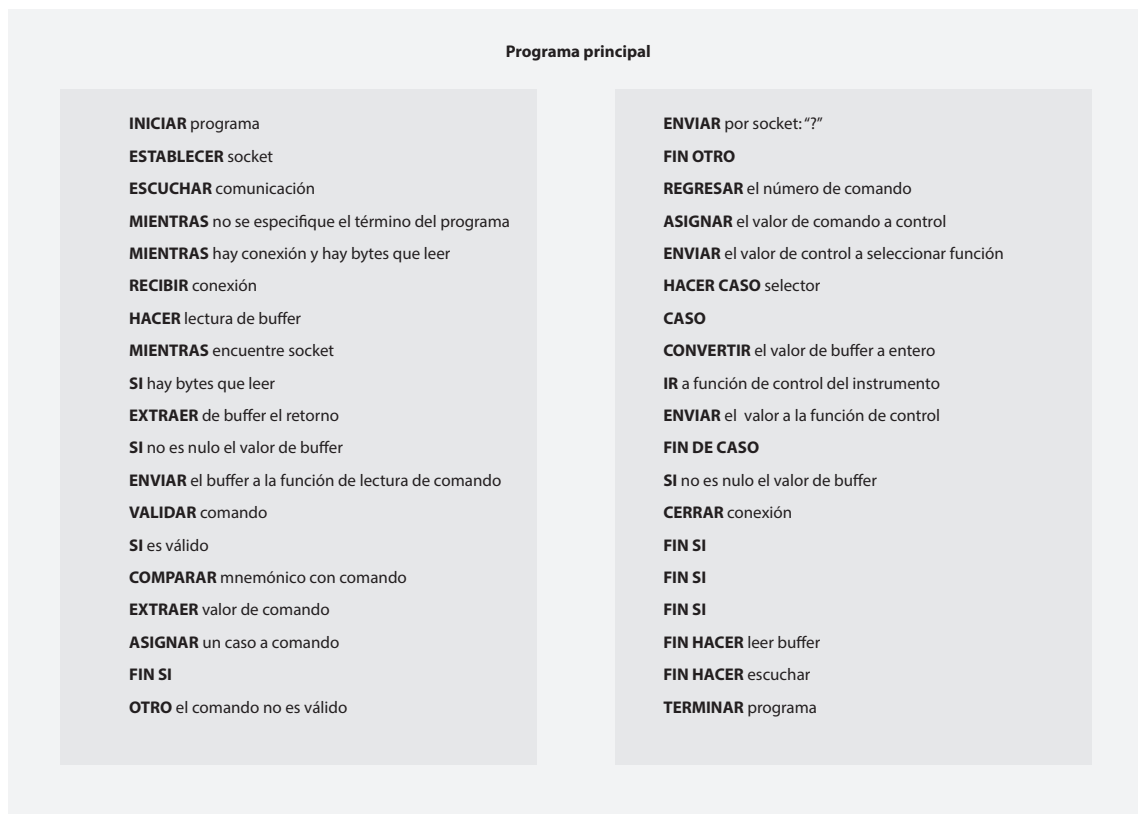
Con la función `tcp_tick()` se revisa el estado de la pila TCP y permite la actualización de los arreglos de caracteres, provenientes del hardware del controlador de red al *socket* creado.

A continuación se muestra la parte del código del programa de control, correspondiente al programa principal.

```

1 void main()
2 {
3     //Inicia el Instrumento
4     Inicializando();
5
6     //Inicia la red y espera a que aparezca la interfaz (o salga del error).
7     sock_init();
8     memset(buffer,0,sizeof(buffer));

```

Cuadro 5.7: Especificación de procesos del programa principal

```

9  while(termina==1) {
11     i = tcp_listen(&socket,PORT,0,0,NULL,0);
12     printf("\nWaiting for connection...\n");
13     while(!sock_established(&socket) && sock_bytesready(&socket) <=0)
14         tcp_tick(NULL);
15     printf("Connection received...\n");
16     do {
17         bytes_read=sock_fastread(&socket,buffer,sizeof(buffer));
18         if(bytes_read>0) {
19             buffer[bytes_read]=0;
20             printf("comando= %s\n",buffer);
21             extraebuff=strchr(buffer,'\r');
22             if(extraebuff!=NULL){
23                 cntl=leeComando(buffer);
24                 selecciona(cntl);
25             }
26         }
27         buffer[0]='\0';
28     } while(tcp_tick(&socket));
29 }
30 printf("Programa Terminado\n");
31 }

```

Seleccionar caso. Una vez que se establece la conexión TCP/IP, el programa de control puede recibir un comando que consiste de una cadena de caracteres, para realizar una función determinada. Esta parte del programa interpreta los caracteres y ejecuta una función específica. Al

terminar regresa al programa principal para dar fin a un ciclo de comunicación. A continuación se muestra parte del código del programa de control correspondiente a seleccionar un caso.

```

1  int leeComando(const char *control)
   {
3  int i;
   char Valida [71]="x,y,z,c,e,f,i,k,l,p,v,r,s,F,o,n";
5  char Buff1 [4];

7  Buff1 [0]=control [0];
   Buff1 [1]='\0';
9  memset(buffComandFP, '\0', sizeof(buffComandFP));
   if(strstr(Valida, Buff1) !=NULL){
11
       if(control [0]=='x'){
13         for(i=1;i<7;i++){
             buffComandFP [i-1]=control [i];
15         }
         comando=1;
17     }
       //FPX
19     if(control [0]=='y'){
       for(i=1;i<7;i++){
21         buffComandFP [i-1]=control [i];
       }
23     comando=2;
       }
25     //FPY
       if(control [0]=='z'){
27         for(i=2;i<7;i++){
             buffComandFP [i-2]=control [i];
29         }
       //printf("valor de z%s\n",buffComandFP);
31     comando=3;
       }
33     //FPZ

35     .
37     .
39     }
   }
41 else{
       sock_write(&socket, "?:", strlen("?:"));
43     sock_flush(&socket);
       }
45 return comando;
   }//leeComando.
47
void selecciona(int selector)
49 {
51 switch(selector){
   case 1:
53     eje=atoi(buffComandFP);
       Eje_X(eje);
55     break;
   case 2:
57     eje=atoi(buffComandFP);
       Eje_Y(eje);
59     break;

```

```

case 3:
61     eje=atoi(buffComandFP);
        Eje_Z(eje);
63     break;
65     .
        .
67     .
69     }
        comando=0;
71 }

```

Configuración de la función de retraso. Se realiza una función para seleccionar libremente el tiempo de retraso, en unidades de milisegundos. A continuación se muestra la parte del código del programa de control correspondiente a la función de retraso.

```

1  /*****
   *
3  * NOMBRE DE LA FUNCIÓN: Delay1ms
   *
5  * ARGUMENTOS:
   *
7  * ARGUMENTO TIPO E/S DESCRIPCIÓN
   * -----
9  * mSdelay int E Tiempo requerido en milisegundos
   *
11 * RETORNO: void
   *
13 *****/

```

```

1 void Delay1ms(int mSdelay)
  {
3     auto unsigned long t0;
        t0= MS_TIMER;
5     while (MS_TIMER < (t0 + mSdelay));
  }

```

Función de inicialización de la rueda de filtros. Para la inicialización de la rueda de filtros (Cuadro 5.8), un sensor de proximidad (RF-16 marca Turck) que se encuentra por debajo de la rueda, detecta un inserto ferroso sobre ésta, el cual indica la posición de filtro número cero. Siempre que se inicialice la rueda el inserto coincidirá con el sensor de proximidad. A continuación se muestra la parte del código del programa de control, correspondiente a la función de inicialización de la rueda de filtros.

```

/*****
2 *
   * NOMBRE DE LA FUNCIÓN: IniciaFiltros
4 *
   * ARGUMENTOS:
6 *
   * ARGUMENTO TIPO E/S DESCRIPCIÓN
8 * -----
   * void - - Sin parámetros
   */

```

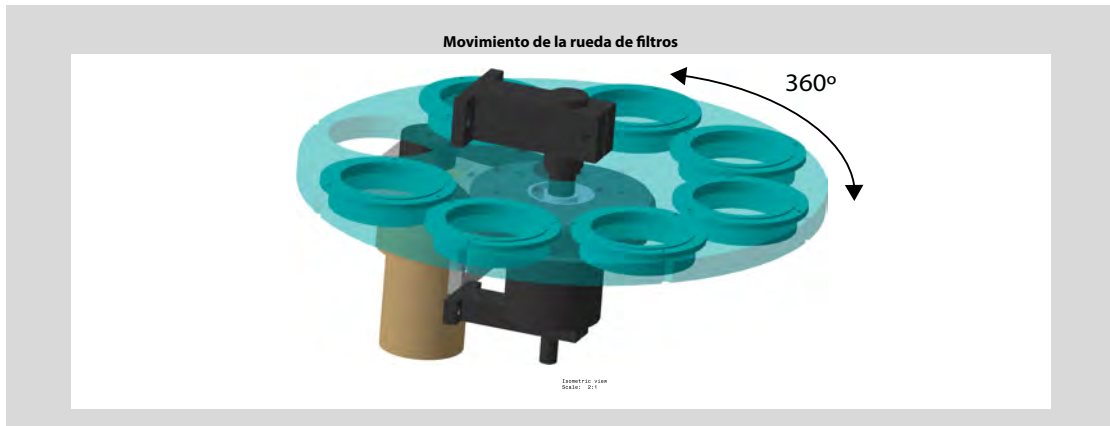
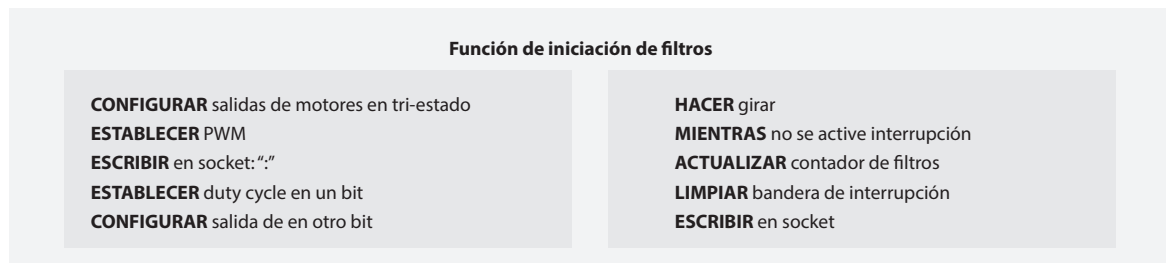


Figura 5.1: Movimiento de giro la rueda de filtros



Cuadro 5.8: Especificación del proceso selección de filtro de origen

```

10 *
11 * RETORNO: void
12 *
13 ***** /
    
```

```

1 void IniciaFiltros(void)
2 {
3     digOutTriState_H(0,1);
4     setPWM_H(1, 50.0, 25.0,HCPWM_TRI_LOW, 0);
5     sock_write(&socket, ":", strlen(":"));
6     sock_flush(&socket);
7     do{
8         while(button_event1==0);
9         digOutTriState_H(0,2);
10        digOutTriState_H(1,2);
11        Delay1ms(500);
12        RuedaFiltros(1);
13        Delay1ms(500);
14        RuedaFiltros(0);
15        digOutTriState_H(0,2);
16        digOutTriState_H(1,2);
17        Filtro=0;
18        C=0;
19        button_event1=0;
20        sock_write(&socket, "#", strlen("#"));
21        sock_flush(&socket);
22    }
23    return;
    
```

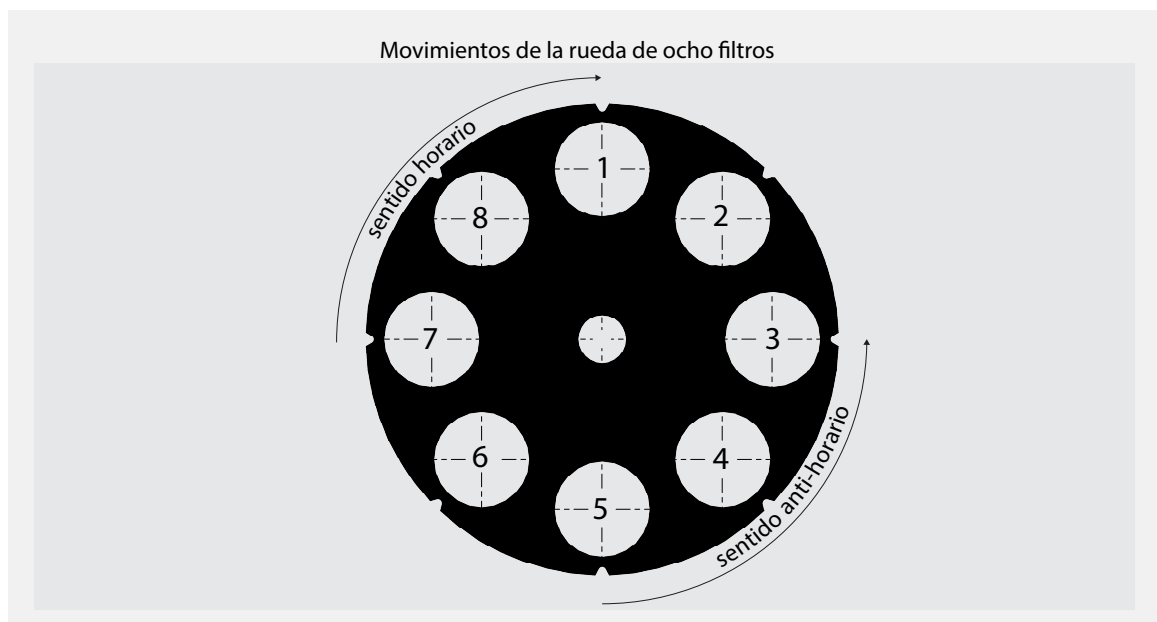


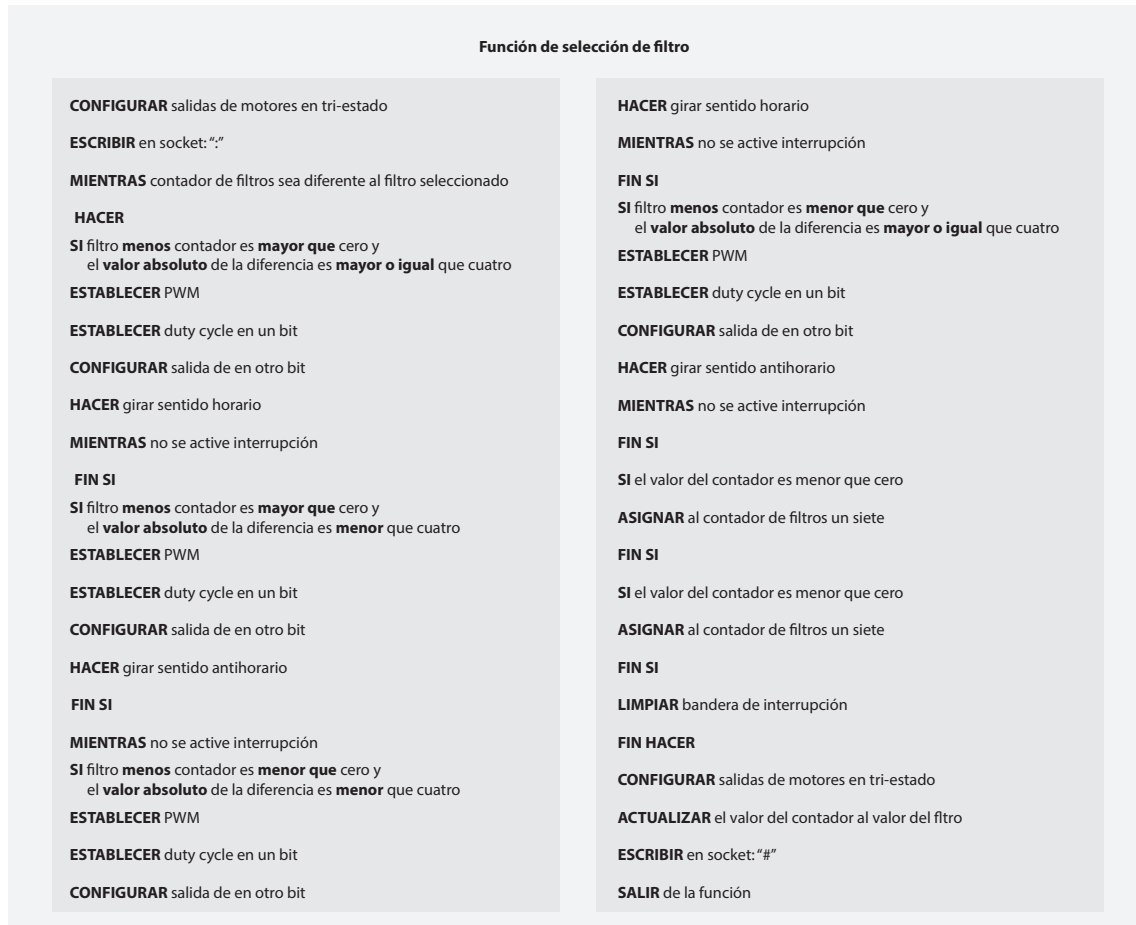
Figura 5.2: Indexaciones de la rueda de filtros. La rueda de filtros puede girar en sentido horario o en sentido antihorario. Para el programa de control se contarán los filtros del 1 a 8 (en sentido horario).

Diferencias entre la posición actual y la siguiente de la rueda de filtros

| F - C | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|-------|---|----|----|----|----|----|----|----|
| 1 | 0 | -1 | -2 | -3 | -4 | -3 | -2 | -1 |
| 2 | 1 | 0 | -1 | -2 | -3 | -4 | -3 | -2 |
| 3 | 2 | 1 | 0 | -1 | -2 | -3 | -4 | -3 |
| 4 | 3 | 2 | 1 | 0 | -1 | -2 | -3 | -4 |
| 5 | 4 | 3 | 2 | 1 | 0 | -1 | -2 | -3 |
| 6 | 3 | 4 | 3 | 2 | 1 | 0 | -1 | -2 |
| 7 | 2 | 3 | 4 | 3 | 2 | 1 | 0 | -1 |
| 8 | 1 | 2 | 3 | 4 | 3 | 2 | 1 | 0 |

Cuadro 5.9: Análisis del movimiento de la rueda de filtros. C toma el valor de la posición de filtro a la que se quiere posicionar la rueda y F es la posición actual en la que se encuentra la rueda de filtros; la diferencia determinará cuantos lugares tendrá que girar y el signo determinará su sentido de giro.

Función para seleccionar filtro. Para pasar de un filtro al siguiente simplemente se gira el disco de la rueda a las posiciones requeridas (Figura 5.2). Cada posición (1 - 8) está claramente marcada en el dial; en el posicionamiento de la rueda se seleccionan sus posiciones absolutas



Cuadro 5.10: Especificación del proceso de la rueda de filtros

de forma remota a través de la interfaz de usuario en la estación de trabajo. El dispositivo de la rueda de filtros es muy preciso y repetible. La rueda de filtros inteligente trae un nuevo nivel de equipamiento sofisticado. El uso del filtro incorrecto puede estropear fácilmente una (por lo demás exitosa), noche de observación. En el programa se ajustan las posiciones para mandar el número de filtro con rango de 0-7. Por las indexaciones, las posiciones de la rueda de filtros son detectadas por un interruptor mecánico.

El Cuadro 6.18 muestra el análisis que se llevó a cabo para que al seleccionar el filtro deseado, la rueda se desplace las mínimas posiciones posibles y se pueda colocar en el filtro correcto. El parámetro que se tomó en cuenta fue la diferencia entre la posición del filtro actual (F) y la posición del filtro deseado o contador (C). El valor absoluto de cada número de la tabla significa el número de posiciones por las que la rueda de filtros tiene que girar. Se observa que el máximo número de posiciones de filtro que pudiera recorrer la rueda, es de cuatro y el mínimo de cero. Por ejemplo, si la rueda va de la posición tres a la tres la diferencia es cero, es decir que no se movería. Los números positivos de la tabla indican un giro anti-horario y los números negativos un giro en sentido horario (Figura 5.10).

A continuación se muestra la parte del código del programa de control correspondiente a la función de selección de filtro.

```

1 /*****
  *
3 * NOMBRE DE LA FUNCIÓN: RuedaFiltros
  *

```

```

*
5 * ARGUMENTOS:
*
7 * ARGUMENTO TIPO E/S DESCRIPCIÓN
* -----
9 * Filtro int E Número de filtro seleccionado
*
11 * RETORNO: void
*
13 *****/

1 void RuedaFiltros(int Filtro)
{
3   button_event0=0;
   sock_write(&socket, ":", strlen(":"));
5   sock_flush(&socket);
   while(C!=Filtro){
7     if((Filtro-C)>0 && (abs(Filtro-C)>=4)){
         C=C-1;
9         digOutTriState_H(1,1);
         setPWM_H(0, 50.0, 25.0, HCPWM_TRI_LOW, 0);
11        Delay1ms(500);
         button_event0=0;
13        do{}
         while(button_event0==0);
15    }
     if((Filtro-C)>0 && (abs(Filtro-C)<4)){
17        C=C+1;
         digOutTriState_H(0,1);
19        setPWM_H(1, 50.0, 25.0, HCPWM_TRI_LOW, 0);
         Delay1ms(500);
21        button_event0=0;
         do{}
23        while(button_event0==0);
     }
25    if((Filtro-C)<0 && (abs(Filtro-C)<4)){
         C=C-1;
27        digOutTriState_H(1, 1);
         setPWM_H(0, 50.0, 25.0, HCPWM_TRI_LOW, 0);
29        Delay1ms(500);
         button_event0=0;
31        do{}
         while(button_event0==0);
33    }
     if((Filtro-C)<0 && (abs(Filtro-C)>=4)){
35        C=C+1;
         digOutTriState_H(0,1);
37        setPWM_H(1, 50.0, 25.0, HCPWM_TRI_LOW, 0);
         Delay1ms(500);
39        button_event0=0;
         do{}
41        while(button_event0==0);
     }
43    if((C)<0){
         C=7;
45    }
     if((C)>7){
47        C=0;
     }
49    button_event0=0;

```

```

    }
51     digOutTriState_H(0, 2);
    digOutTriState_H(1, 2);
53     C=Filtro;
    sock_write(&socket, "#", strlen("#"));
55     sock_flush(&socket);
    return;
57 }

```

Carros móviles. Para la función de movimiento de los carros móviles y el brazo de prismas (Cuadro 5.11), se toman en cuenta sus posiciones extremas. Los movimientos del posicionador de elementos ópticos (Figura 5.3) y de la unidad de calibración móvil (Figura 5.4) son lineales, es decir sólo tienen movimiento en una sola dirección y sus posiciones extremas son detectadas por medio de sensores de proximidad. El movimiento del brazo de prismas (Figura 5.5) es circular y limitado a 90°.

A continuación se muestra la parte del código del programa de control, correspondiente a la función de movimiento del carro móvil del Fabry-Pérot, la programación para el resto de los actuadores tiene la misma estructura.

```

1  /*****
   *
3  * NOMBRE DE LA FUNCIÓN: Dispositivo
   *
5  * ARGUMENTOS:
   *
7  * ARGUMENTO TIPO E/S DESCRIPCIÓN
   * -----
9  * disp      char  E   Tipo de dispositivo seleccionado
   * modo      int   E   Posición en el camino óptico
11 *
13 * RETORNO: void
   *
   *****/

```

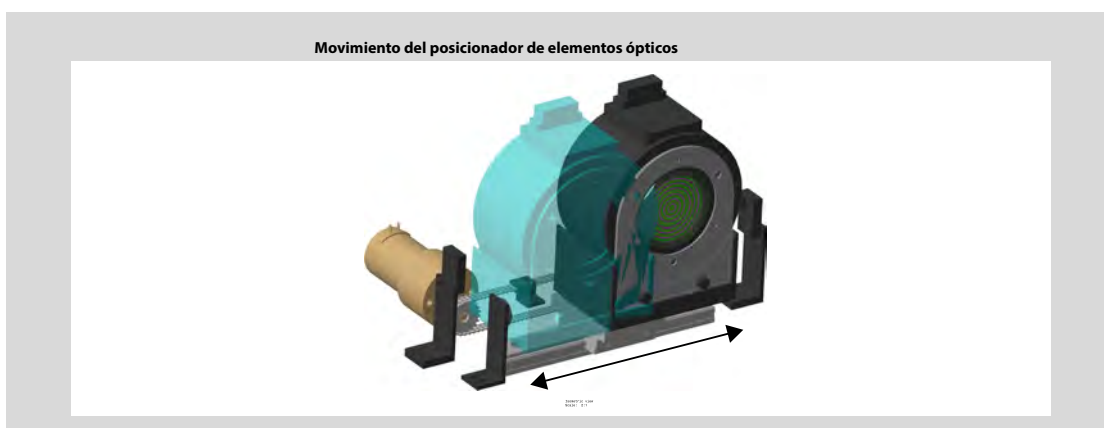


Figura 5.3: Movimiento del posicionador de elementos ópticos.

```

void Dispositivo(char disp, int modo)
2 {

```

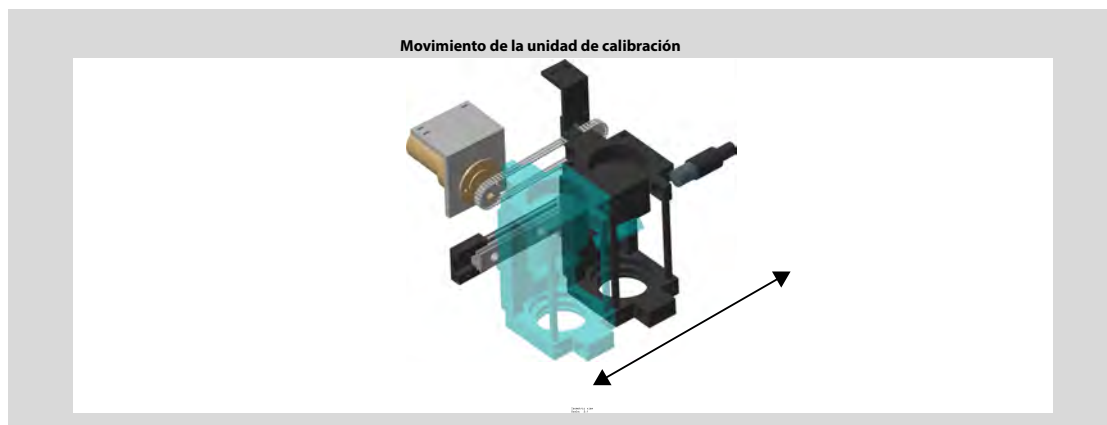



Figura 5.4: Movimiento de la unidad de calibración.

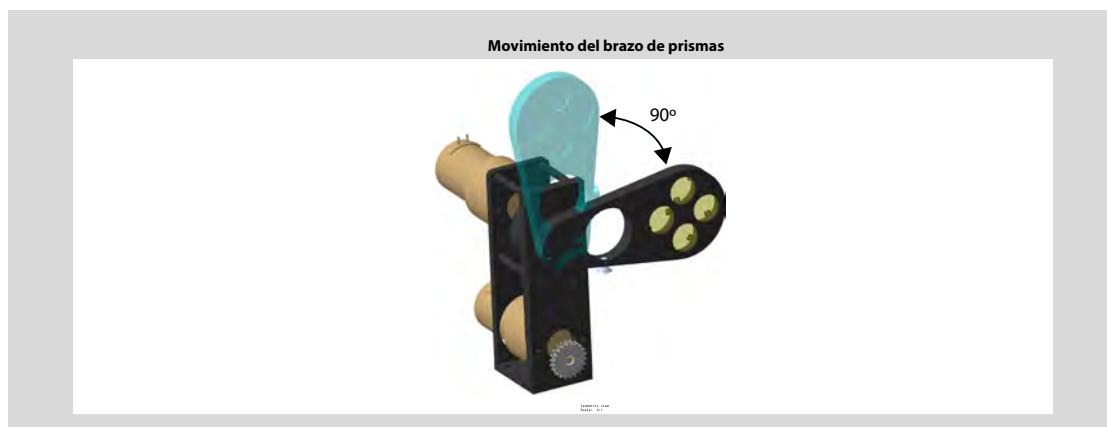
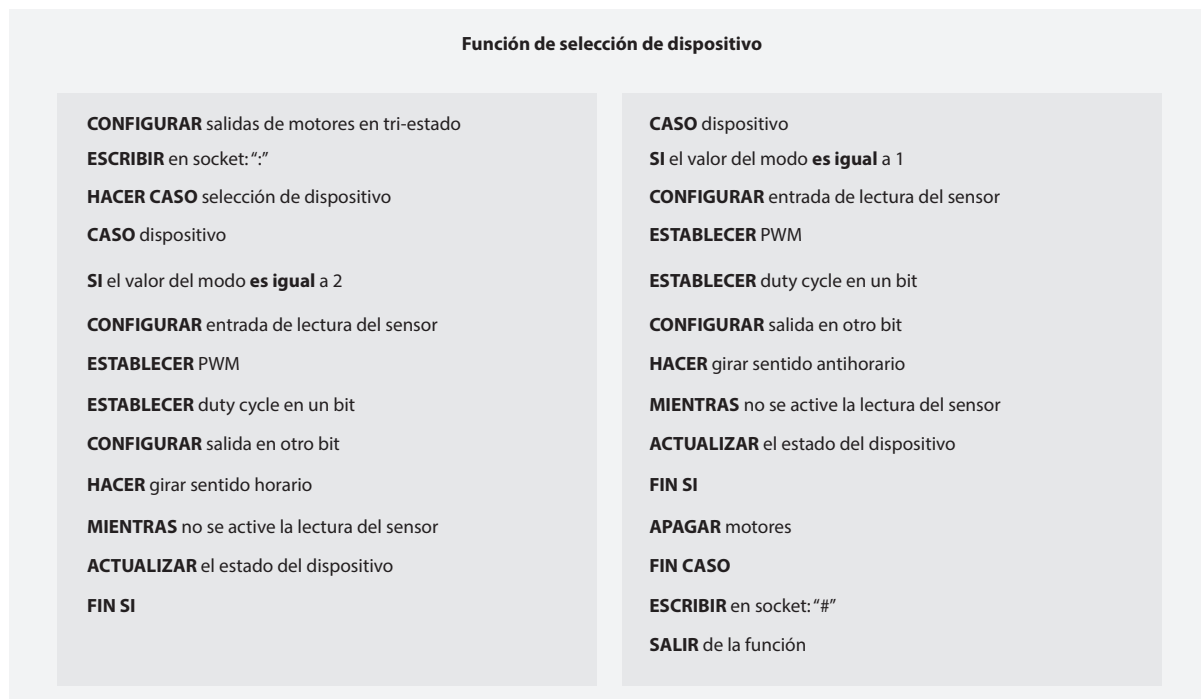


Figura 5.5: Movimiento del brazo de prismas.

```

4 sock_write(&socket, ":", strlen(":"));
5 sock_flush(&socket);
6 switch(dispatch){
7     case 'e':{//subrutina de etalon
8         if(modos==2){ //Adentro
9             setDigIn(8);
10            setDigIn(9);
11            digOutTriState_H(7,1);
12            setPWM_H(6, 50.0, 50.0,HCPWM_TRI_LOW, 0);
13            do{}
14            while(digIn(8)==1);
15            Etalon=2;
16        }
17        if(modos==1){
18            digOutTriState_H(6,1);
19            setPWM_H(7, 50.0, 50.0,HCPWM_TRI_LOW, 0);
20            do{}
21            while(digIn(9)==1);
22            Etalon=1;
23        }
24        digOutTriState_H(7,2);
25        digOutTriState_H(6,2);
26        break;
27    }

```



Cuadro 5.11: Especificación del proceso de los carros móviles.

```

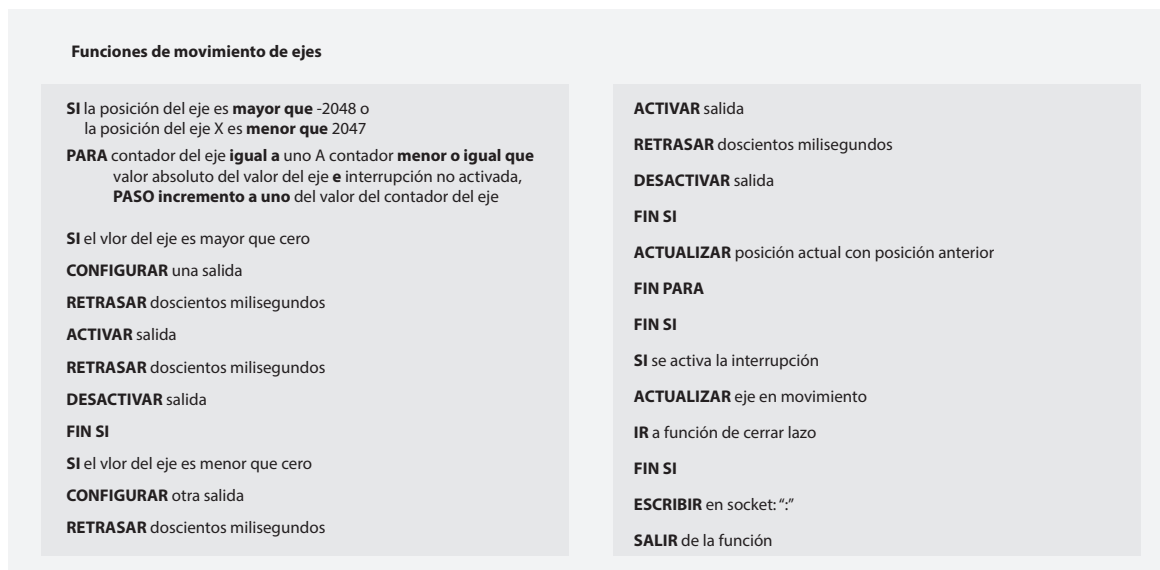
28     .
30     .
32     }
    sock_write(&socket, "#", 2);
34     sock_flush(&socket);
    return;
36 }
    
```

Función para el movimiento de los ejes del interferómetro Fabry-Pérot. Para la actualización de la posición de los ejes del Fabry-Pérot (Cuadro 5.12), se toma en cuenta la posición actual y, por medio de un contador, la posición deseada. Se aumentan o disminuyen los pasos según la diferencia entre la posición actual y el contador. Un paso para cada uno de los ejes (x , y ó z), será una señal cuadrada con ancho de pulso de 5 ms.

A continuación se muestra la parte del código del programa de control correspondiente a la función de movimiento del eje del interferómetro de barrido Fabry-Pérot, el código de programación de los ejes restantes tiene la misma estructura.

```

/*****
2  *
  * NOMBRE DE LAS FUNCIONES: Eje_X, Eje_Y, Eje_Z
4  *
  * ARGUMENTOS :
6  *
  * ARGUMENTO TIPO E/S DESCRIPCIÓN
8  *
  * posicion_X char E Número de pasos en el eje X
  *
  *****/
    
```



Cuadro 5.12: Especificación del proceso de los ejes del Fabry-Pérot.

```

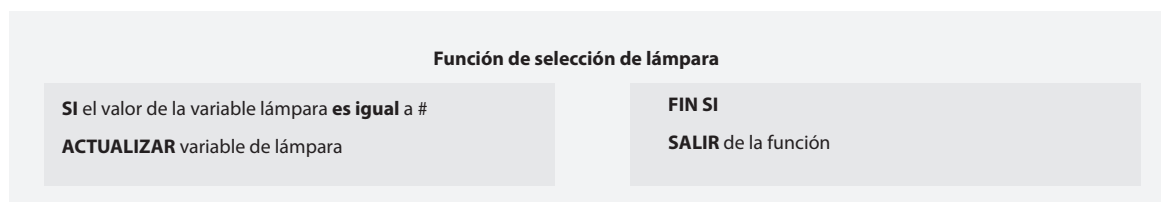
10 * posicion_Y char E Número de pasos en el eje Y *
11 * posicion_Z char E Número de pasos en el eje Z *
12 * * *
13 * RETORNO: void *
14 * * *
***** /

```

```

1 void Eje_X(int posicion_X)
2 {
3     if(posicion_X > -2048 && posicion_X < 2047){
4         X=posicion_X-X_ant;
5         for(CX=1;CX<=abs(X)&&button_event7==0;CX++){
6             if(X>0){
7                 Delay1ms(5);
8                 digOut(18,1);
9                 Delay1ms(5);
10                digOut(18,0);
11            }
12            if(X<0){
13                Delay1ms(5);
14                digOut(19,1);
15                Delay1ms(5);
16                digOut(19,0);
17            }
18        }
19        X_ant=posicion_X;
20    }
21    if(button_event7!=0){
22        LoopCls();
23    }
24    button_event7=0;
25    sock_write(&socket,":",strlen(":"));
26    sock_flush(&socket);
27    return;
28 }

```



Cuadro 5.13: Especificación del proceso de la selección de las lámparas de calibración

Función para selección de lámpara. Esta función determina que lámpara, de las tres disponibles, será activada (Cuadro 5.13). A continuación se muestra la parte del código del programa de control, correspondiente a la función de selección de lámpara.

```

/*****
2  *
3  * NOMBRE DE LA FUNCIÓN: Lamparas
4  *
5  * ARGUMENTOS:
6  *
7  * ARGUMENTO TIPO E/S DESCRIPCIÓN
8  * -----
9  * void - - Sin parámetros
10 *
11 * RETORNO: void
12 *
*****/
    
```

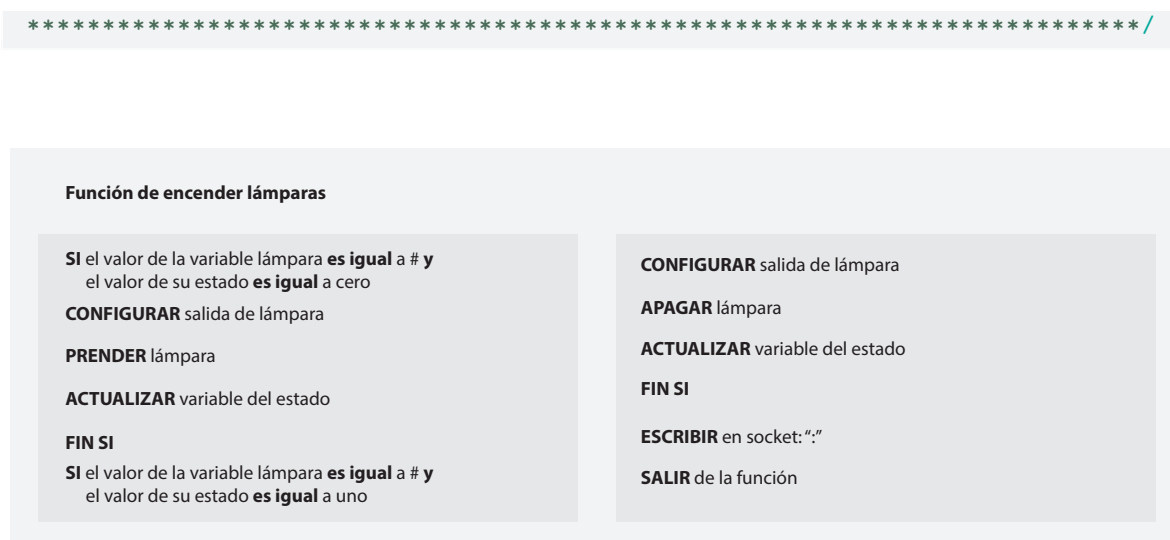
```

1 void Lamparas(void)
2 {
3     sock_write(&socket, ":", strlen(":"));
4     sock_flush(&socket);
5     return;
6 }
    
```

Función para encender lámpara. En esta función alguna de las tres lámparas será activada (Cuadro 5.14). A continuación se muestra la parte del código del programa de control correspondiente a la función de encendido o apagado de alguna lámpara uno de calibración, el código de programación para las lámparas restantes tiene la misma estructura.

```

/*****
2  *
3  * NOMBRE DE LA FUNCIÓN: EnciendeLamp
4  *
5  * ARGUMENTOS:
6  *
7  * ARGUMENTO TIPO E/S DESCRIPCIÓN
8  * -----
9  * Edo int E Seleccionar el estado de la lámpara
10 *
11 * RETORNO: void
12 *
*****/
    
```



Cuadro 5.14: Especificación del proceso de activación de lámpara

```

1 void EnciendeLamp(int Edo)
  {
3   if(Lamp==1 && Edo==0){
4       setDigOut(10,1);
5       digOut(10,1);
6   }
7   if(Lamp==1 && Edo==1){
8       setDigOut(10,1);
9       digOut(10,0);
10  }
11
12  .
13  .
14  .
15  }
16
17 sock_write(&socket,":",strlen(":"));
18 sock_flush(&socket);
19 return;
20 }
    
```

Función para limpiar el valor de algún eje del Fabry-Pérot. Para el reinicio de las secuencias de barrido del CS100, es necesario limpiar los valores de las posiciones de los ejes del interferómetro Fabry-Pérot, que quedaron guardados en el programa de control. A continuación se muestra la parte del código del programa de control correspondiente a la función para limpiar el eje *x* del interferómetro Fabry-Pérot de barrido, el código de programación de los ejes restantes tiene la misma estructura.

```

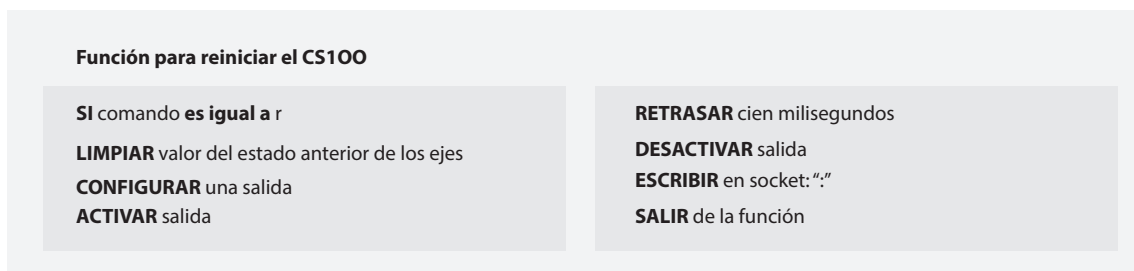
/*****
2  *
3  * NOMBRE DE LA FUNCIÓN: Clear
4  *
5  * ARGUMENTOS:
6  *
7  * ARGUMENTO TIPO E/S DESCRIPCIÓN
8  * -----
9  * void - - Sin parámetros
10 *
11 *
12 *
13 *
14 *
15 *
16 *
17 *
18 *
19 *
20 *
21 *
22 *
23 *
24 *
25 *
26 *
27 *
28 *
29 *
30 *
31 *
32 *
33 *
34 *
35 *
36 *
37 *
38 *
39 *
40 *
41 *
42 *
43 *
44 *
45 *
46 *
47 *
48 *
49 *
50 *
51 *
52 *
53 *
54 *
55 *
56 *
57 *
58 *
59 *
60 *
61 *
62 *
63 *
64 *
65 *
66 *
67 *
68 *
69 *
70 *
71 *
72 *
73 *
74 *
75 *
76 *
77 *
78 *
79 *
80 *
81 *
82 *
83 *
84 *
85 *
86 *
87 *
88 *
89 *
90 *
91 *
92 *
93 *
94 *
95 *
96 *
97 *
98 *
99 *
100*
    
```

```

10 *
11 * RETORNO: void
12 *
13 *****/

1 void Clear()
2 {
3     if(bufSaca[0]=='x'){
4         X_ant=0;
5         digOut(25,0);
6         Delay1ms(1000);
7         digOut(25,1);
8     }
9
10    .
11    .
12    .
13 }
14
15 sock_write(&socket,":",strlen(":"));
16 sock_flush(&socket);
17 return;
18 }
    
```

Función para el restablecimiento del controlador CS100. Para el restablecimiento del controlador CS100 (Cuadro 5.15), ya sea por un fallo o si se requiere para iniciar una nueva observación, se crea una función en donde se activa el control externo, se limpian los valores anteriores de los ejes del interferómetro FP y se activa el restablecimiento.



Cuadro 5.15: Especificación del proceso del reinicio del controlador CS100

A continuación se muestra la parte del código del programa de control correspondiente a la función de reinicio del CS100.

```

/*****
2 *
3 * NOMBRE DE LA FUNCIÓN: Reset_CS100
4 *
5 * ARGUMENTOS:
6 *
7 * ARGUMENTO TIPO E/S DESCRIPCIÓN
8 * -----
9 * void - - Sin parámetros
10 *
11 * RETORNO: void
12 *
13 *****/
    
```



```

8 * ARGUMENTO TIPO E/S DESCRIPCIÓN *
* ----- ---- - ---- *
* void - - Sin parámetros *
10 * *
* RETORNO: void *
12 * *
*****/

```

```

1 void Status(void)
{
3 int n;
n= sprintf(respuesta,"F= %d L= %d X,Y,Z= %d , %d , %d ETA= %d CMVIL= %d
PRIS= %d LAMP %d LAZO %d:",Filtro+1,Lamp,X_ant,Y_ant,Z_ant,Etalon,Cmovil,
Prismas,Edo,Lazo);
5 sock_write(&socket,respuesta,strlen(respuesta));
sock_flush(&socket);
7 respuesta==" ";
return;
9 }

```

Función para inicializar el instrumento. Al inicializar el instrumento PUMA los dispositivos deben encontrarse en una posición predeterminada. La función inicializar establece que la rueda de filtros se debe encontrar en la posición uno (filtro vacío), se debe colocar el posicionador de elementos ópticos fuera del camino óptico; así como la unidad móvil de calibración y el brazo de prismas. Se debe seleccionar la lámpara uno, activar el bit de control externo y cerrar el lazo de control del CS100 y volver al programa principal, a escuchar la posible llegada de comandos.

A continuación se muestra la parte del código del programa de control correspondiente a la función para inicializar el instrumento PUMA.

```

1 /*****
*
3 * NOMBRE DE LA FUNCIÓN: Inicializando *
* *
5 * ARGUMENTOS: *
* *
7 * ARGUMENTO TIPO E/S DESCRIPCIÓN *
* ----- ---- - ---- *
9 * void - - Sin parámetros *
* *
11 * RETORNO: void *
* *
13 *****/

```

```

1 void Inicializando(void)
{
3 int n;

5 IniciaFiltros();
Dispositivo('e',1);
7 Dispositivo('v',1);
Dispositivo('p',1);
9 Lamparas();
EnciendeLamp(0);

```



```

11     digitalWrite(25,0);
13     Delay1ms(1000);
14     digitalWrite(25,1);
15
16     digitalWrite(26,0);
17     Delay1ms(1000);
18     digitalWrite(26,1);
19
20     digitalWrite(24,0);
21     Delay1ms(1000);
22     digitalWrite(24,1);
23
24     return;
25 }

```

Función para inicializar el controlador CS100. El lazo de control del controlador CS100, para el inicio de las observaciones y calibraciones con el FP, debe asegurarse que esté cerrado. A continuación se muestra la parte del código del programa de control correspondiente a la función para cerrar el lazo del CS100.

```

1  /*****
2  *
3  * NOMBRE DE LA FUNCIÓN: LoopCls
4  *
5  * ARGUMENTOS:
6  *
7  * ARGUMENTO TIPO E/S DESCRIPCIÓN
8  * -----
9  * void - - Sin parámetros
10 *
11 * RETORNO: void
12 *
13 *****/

```

```

1 void LoopCls()
2 {
3     Lazo=0;
4     return;
5 }

```

Función para finalizar el instrumento. Esta función garantiza que los motores de los dispositivos permanezcan desenergizados, al igual que las señales de los ejes del FP y del controlador CS100 al terminar el programa de control. A continuación se muestra la parte del código del programa de control correspondiente a la función para finalizar el instrumento PUMA.

```

1  /*****
2  *
3  * NOMBRE DE LA FUNCIÓN: Finalizar
4  *
5  * ARGUMENTOS:
6  *
7  * ARGUMENTO TIPO E/S DESCRIPCIÓN
8  * -----

```

```

9  * void      -      Sin parámetros      *
  *                                          *
11 * RETORNO: void                        *
  *                                          *
13 *****/

```

```

1 void Finalizar()
  {
3 char Bank_2;
  char Bank_3;
5 int canal;
  //Banco de motores
7 for(canal=0; canal<= 7; ++canal){
  digOutTriState_H(canal, 2);
9 }
  //Banco de Ejes 22&23 N.C.
11 digOutBank(2,Bank_2);
  //Banco de funciones del CS100
13 digOutBank(3,Bank_3);
  return;
15 }

```

Función para detectar sobrecarga. Si ocurre algún problema con el controlador CS100 la función de sobrecarga permite diagnosticar que eje del controlador puede tener problemas. A continuación se muestra la parte del código del programa de control correspondiente a la función para detectar fallo en algún eje de movimiento del interferómetro Fabry-Pérot.

```

1 /*****
  *
3 * NOMBRE DE LA FUNCIÓN: Overload      *
  *
5 * ARGUMENTOS:                        *
  *
7 * ARGUMENTO  TIPO  E/S  DESCRIPCIÓN  *
  * -----  ---  ---  -----  *
9 * void      -      Sin parámetros    *
  *
11 * RETORNO: void                      *
  *
13 *****/

```

```

1 void Overload()
  {
3
5 setDigIn(13);
  digIn(13)==1;

7 setDigIn(14);
  digIn(14)==1;

9
11 setDigIn(15);
  digIn(15)==1;
  }

```


CAPÍTULO

6

Simulación en laboratorio

En este capítulo se describen las pruebas realizadas con el hardware del microcontrolador y su adaptación con el programa de control.

Índice

| | |
|-------------------------------------|----|
| 6.1 Pruebas de desarrollo | 96 |
|-------------------------------------|----|

6.1 Pruebas de desarrollo

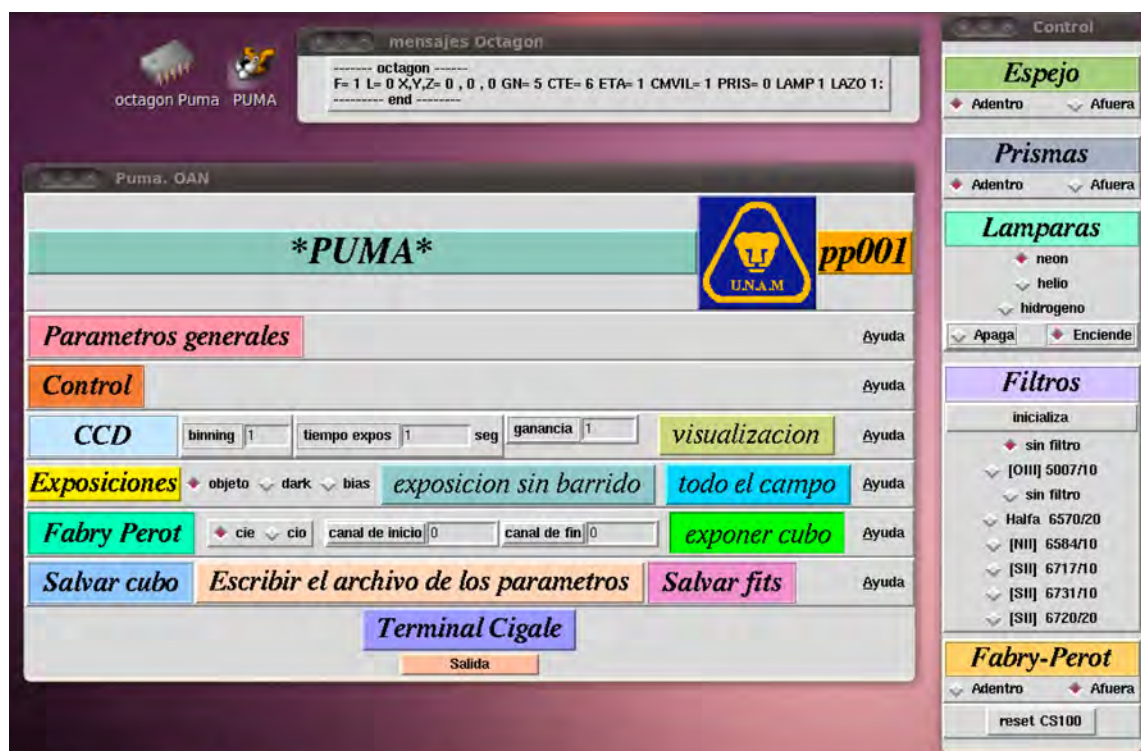


Figura 6.1: Interfaz de usuario simulada del instrumento PUMA.

En el Laboratorio de Electrónica y Detectores del Instituto de Astronomía se realizaron tres pruebas de unidades separadas o modulares, para integrar los resultados a los subsistemas del instrumento PUMA y posteriormente realizar procesos de pruebas en el telescopio y eliminar fallas durante su operación.

Este capítulo trata sobre las pruebas que se realizaron para el desarrollo del sistema. Se describen organizadamente las actividades y el propósito de las pruebas, es decir, cuál fue el objetivo de la prueba y qué parte del sistema se está probando, localización y fecha de la prueba, la descripción de la prueba, la descripción de las entradas que se proporcionarán al sistema y las salidas que se anticipan. En el caso de las pruebas de la programación las descripciones de las entradas están en el formato del diccionario de datos.

En los procedimientos de prueba se encuentra la descripción de cómo se deben planificar y mostrar los datos del sistema y cómo capturar los resultados de salida, cómo analizar los resultados de las pruebas y cualesquiera otros procedimientos operacionales que se deban observar. Aunque un plan es en parte suficiente, los problemas sólo se resuelven mediante la investigación y la experimentación, para evitar obstáculos tecnológicos que puedan complicar el proyecto.

La placa de prototipos Rabbit primero se someterá a pruebas electrónicas y de control en cada una de sus subrutinas y luego se colocará en un ensamble terminado, formando parte del nuevo controlador del instrumento PUMA. Es por ello que como un controlador que hace uso de

los servicios de red, el código tiene que ser depurado en condiciones simuladas de operación. Además es posible la generación de código de diagnóstico que sea fácil de instalar y utilizar posteriormente.

Las pruebas y para el programa de control, se realizaron por medio del sistema de desarrollo Dynamic C, que corre en el sistema operativo Windows y con un simulador de la interfaz de usuario (Figura 6.1).

6.1.1 Pruebas de factibilidad y desarrollo

Objetivo de la prueba: Depurar y corregir el código del programa de control que se compilará en la placa BL4S200.

Subsistema a probar: Código de control.

Localización y fecha de la prueba: Laboratorio de Electrónica y Detectores del Instituto de Astronomía. Del 8 de enero de 2015 al 27 de Julio de 2015.

Salidas anticipadas: Salidas y entradas digitales correspondientes a los sensores y detectores del instrumento PUMA.

Procedimiento de prueba:

Se realizó código de ensayo para simular partes del programa completo de la aplicación. Con estos programas de prueba se aislaron componentes individuales de hardware y se verificó su funcionamiento para revisar la factibilidad del diseño. Si los sensores requerían ser identificados y los motores calibrados, el código de prueba también podía realizar esta tarea. Después de que el hardware del instrumento fuera calificado por el código de prueba, el código de la aplicación se cargó a la placa.

Los microcontroladores de una sola placa BL4S200 se programan en un puerto USB del PC estándar, a través de un cable de programación; por lo que Dynamic C requiere un PC con sistema operativo Windows con al menos un puerto USB. Para preparar la placa de prototipo BL4S200, para su desarrollo, se colocó el módulo RCM4300 en la junta de la placa y se conectó el cable de programación (JTAG), el cual se utiliza para conectar el puerto de programación del RCM4300 a un puerto USB del PC.

El RCM4300 está automáticamente en el modo de programación cuando se conecta el cable de programación y está automáticamente en modo de funcionamiento cuando no está conectado el cable de programación. Un programa “corre” en cualquiera de los modos, pero sólo se puede descargar y depurar cuando el RCM4300 está en el modo de programación. Finalmente la placa de prototipos se conecta a la fuente de alimentación.

6.1.2 Pruebas de red

Objetivo de la prueba: Tener una red de comunicación funcional con un sistema de dispositivos de hardware, para interconectar y controlar los medios de transmisión; así como para detectar y corregir problemas, con el propósito de la comunicación y el intercambio de recursos.

Subsistema a probar: Protocolo y código de transmisión de datos TCP.

Localización y fecha de la prueba: Laboratorio de Electrónica y Detectores del Instituto de Astronomía. Del 26 de enero de 2015 al 27 de Julio de 2015.

Salidas anticipadas: Fiabilidad en el envío y recepción de comandos de control entre la estación de trabajo y la placa BLAS200.

Procedimiento de prueba:

Mientras que algunas decisiones de control deben realizarse a nivel local, otras decisiones se toman de forma remota por los operadores o por el sistema de control del CS100. Los comandos y protocolos de comunicación deben ser iguales entre la estación de trabajo y el instrumento PUMA.

El controlador automatizado se comunica utilizando las características del protocolo Ethernet TCP/IP para redes de sistemas embebidos, cuyo cliente y servidor entre una PC y el módulo central del Rabbit transfieren y adquieren datos de un sistema a otro con códigos de red. La estación de trabajo del observatorio es elegida como la plataforma de control del instrumento PUMA; por lo que puede ser operado a través de una red IP (protocolo de Internet) desde una ubicación remota. Lo anterior permite al operador no estar en el telescopio para controlar el sistema. Los paquetes de información TCP/IP se transportan fácilmente a través de la red de Internet del Observatorio. El control del PUMA posee un puerto Ethernet para la comunicación entre el instrumento instalado en el telescopio y la estación de trabajo.

El software de cliente será un programa de aplicación arbitrario cuando se desee solicitar un servicio remoto e iniciará activamente el contacto, invocado por un usuario y ejecutándose durante una sesión. El servidor es capaz de proporcionar múltiples servicios, según sea necesario, pero activamente tendrá contacto con sólo un cliente remoto a la vez. Entretanto el software del servidor tiene un propósito especial y privilegia al programa dedicado a proporcionar un servicio. La aplicación de servidor esperará pasivamente a la aplicación cliente para iniciar el contacto. El tipo de servicios que proporciona el servidor puede requerir un hardware potente y un sistema operativo sofisticado, dependiendo del número de clientes a los que sirva.

El código en Dynamic C se ejecutará como un servidor para escuchar las peticiones entrantes en un puerto específico. Una vez que el cliente establece la conexión, el cliente y el servidor entran en un bucle en el que el servidor mantiene la escucha de solicitud del cliente y devuelve una respuesta basada en el canal solicitado. Durante estas pruebas se desarrollaron dos códigos de programa cliente, uno para Visual C de Windows y otro en Linux (UNIX). El código de cliente se muestra en el Apéndice B.

Cualquier dispositivo que se coloca en una red de protocolo de Internet (IP) basada en Ethernet debe tener su propia dirección IP. Las direcciones IP se expresan como números de 4 decimales separados por puntos, cada número decimal debe estar entre 0 y 255. La dirección IP total es un número de 32 bits que consiste en los 4 byte (es decir que se divide en cuatro campos de 8 bits),

el cual identifica de forma única a un dispositivo. Una red local utiliza un grupo de direcciones IP adyacentes. Algunos rangos de direcciones IP están reservados para uso en redes internas y se pueden asignar libremente, siempre que no haya dos *hosts* internos que tengan la misma dirección IP. Estas direcciones IP internas no se enrutan a Internet y cualquier *host* interno puede utilizar una de estas direcciones IP reservadas; sin embargo para comunicarse externamente a través de Internet se necesita estar conectado a un *host* que tenga una dirección IP válida de Internet.

La máscara de red (también llamada máscara de subred) determina el número de direcciones IP que pertenecen a la red local. La máscara de red es también una dirección de 32 bits expresada en la misma forma que la dirección IP e indica a la pila TCP/IP qué parte de la dirección es la dirección de red y qué parte es el identificador del *host*. Esto se realiza mediante una operación lógica bit a bit tipo *AND* de la dirección IP y la máscara de red.

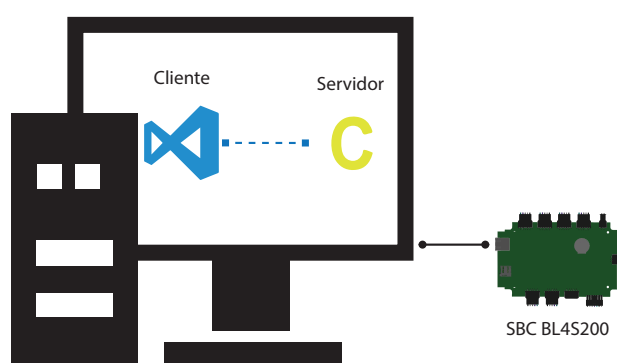


Figura 6.2: Prueba de conexión de red 1.

Primer prueba (Figura 6.2). Se utilizó una conexión directa, que es una alternativa más sencilla para el desarrollo de escritorio. Se conectó el puerto Ethernet del módulo RCM4300 directamente a la tarjeta de interfaz de red del PC mediante un cable cruzado RJ-45 CAT 5/6, ya que es necesario por que los dos equipos están conectados directamente. Un cable cruzado es un cable especial que cambia algunas conexiones entre los dos conectores y permite la conexión directa de dos sistemas (un cable de red estándar RJ-45 no funcionará para este fin). Se conectó directamente la placa Rabbit con la PC, transmitiendo comandos mediante un cliente-servidor, programado en lenguaje Visual C.

Las siguientes líneas contienen los parámetros de dirección IP, máscara de red, servidores de nombres, y la puerta de enlace, con los que se conectó con un servidor TCP.

```

1 #define _PRIMARY_STATIC_IP "10.10.6.100"
2 #define _PRIMARY_NETMASK "255.255.255.0"
3 #define MY_NAMESERVER "10.10.6.1"
4 #define MY_GATEWAY "10.10.6.1"
Port: 23

```

Después de estar satisfechos de que el servidor TCP del Rabbit funcionó como se esperaba, se utilizó el código Unix de cliente en la plataforma PC para comunicarse con el servidor TCP del Rabbit.

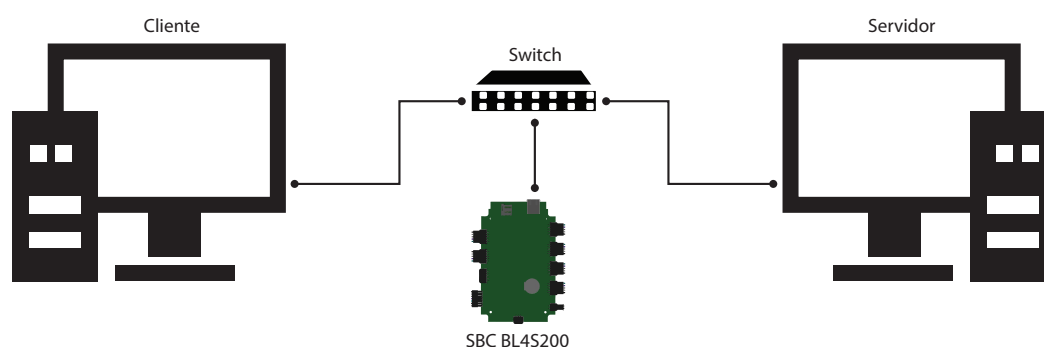


Figura 6.3: Prueba de conexión de red 2.

Segunda prueba (Figura 6.3). En la segunda prueba los programas requerían que la conexión de la PC y la placa BL4S200 compartieran la misma red. Durante la fase de desarrollo, un módulo básico de Rabbit es probable que pueda conectarse a una VLAN (Red Virtual de Área Local) de prueba interna para las aplicaciones en red. Se estableció una red local privada (preferida para la experimentación inicial y depuración). Junto con dos cables de conexión directa CAT 5/6 Ethernet para conectores RJ-45 y un *switch* (una alternativa a la conexión directa mediante un cable cruzado, es una conexión directa utilizando un concentrador). El *switch* retransmite los paquetes recibidos en cualquier puerto a todos los puertos en el concentrador. El RCM4300 utiliza 100 Mbps Ethernet, por lo que el adaptador concentrador de Ethernet puede ser una unidad de acceso que proporcione conectividad a 10 Mbps o una unidad de 10/100 Mbps. Mediante el uso de *switches* y *routers*, una red puede incluir un gran número de equipos.

Las aplicaciones de cliente y servidor no tienen que ser escritos en el mismo lenguaje o plataforma de desarrollo. Por ejemplo, en este caso se utilizó el mismo programa en Dynamic C, para trabajar con servidores entre la plataforma Linux y su ejecución en una estación de trabajo en Windows, ambos en la misma VLAN, para probar la conectividad con el módulo principal del Rabbit. El uso de un modo reconocido de protocolos de interfaz de red hace posible la plataforma y el lenguaje de la abstracción.

Para ser controlable a distancia, el sistema estará diseñado con una interfaz Ethernet. Una red Ethernet utiliza direcciones de adaptador de Ethernet, en las conexiones de hardware reales a través de todas las interfaces Ethernet se tiene una dirección única de 48 bits, que se suministra y se programa por el fabricante además de ser única para cada adaptador. Se llama la dirección MAC (Media Access Control) o dirección de hardware. Las placas Rabbit habilitadas para Ethernet almacenan este valor en la memoria flash (EEPROM). Con el fin de enviar un paquete a otro equipo, dada su dirección IP, primero se determina si el paquete debe ser enviado directamente al otro equipo.

Cada módulo RCM4300 RabbitCore tiene su propia dirección MAC única, que consiste en el prefijo 0090C2 seguido por un código que es único para cada módulo RCM4300. Por ejemplo, en este caso la dirección de MAC es:

```
1 00:90:c2:e8:52:91
```

Las siguientes líneas contienen los parámetros de dirección IP, máscara de red, servidores de nombres y la puerta de enlace, con los que se conectó con un servidor TCP para la prueba 2.

```

1 #define _PRIMARY_STATIC_IP "10.0.4.16"
  #define _PRIMARY_NETMASK "255.255.0.0"
3 #define MY_NAMESERVER "132.248.10.2"
  #define MY_GATEWAY "10.0.0.1"
5 Port: 12007

```

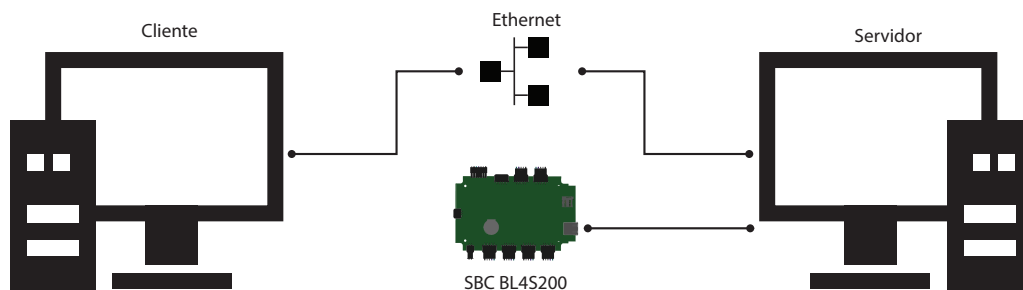


Figura 6.4: Prueba de conexión de red 3

Tercera prueba (Figura 6.4). En la tercera prueba de red, la más importante de todas, se debe tener acceso total a una red Ethernet. Esta opción requiere más cuidado en la selección y en las pruebas durante las acciones de dirección, ya que pueden producirse conflictos con otros usuarios, servidores y sistemas. Se establece una Red de Área Local (LAN) para conectar el puerto Ethernet del módulo RCM4300 a una LAN existente, preferiblemente una a la que el PC de desarrollo esté ya conectado. Se tendrá que obtener información de direccionamiento IP del administrador de red. La red de área local es el tipo que más ampliamente se despliega de la red, la red LAN se ha diseñado como una alternativa a la transmisión de más recursos durante una conexión punto a punto.

En el OAN-SPM se tiene una topología particular de redes locales. El control de los telescopios y sus instrumentos se realiza en un segmento de red local, llamado «red 0» que se encuentra aislado de los demás segmentos para enviar interdependencia con la operación y el desarrollo de las observaciones astronómicas. La administración de la red del OAN-SPM asigna una dirección IP estática a cada dispositivo conectado a la red local.

Si se cuenta con un cable Ethernet conectado al administrador del sistema junto con su dirección IP, la máscara de red y la dirección de puerta de enlace, entonces se puede ejecutar el programa de control sin tener que configurar una conexión directa entre el ordenador y el RCM4300. También se necesitará la dirección IP del servidor de nombres, el nombre o la dirección IP del servidor y el dominio.

Las siguientes líneas contienen los parámetros de dirección IP, máscara de red, servidores de nombres y la puerta de enlace, con los que se conectó con un servidor TCP para la prueba 3.

```

1 #define _PRIMARY_STATIC_IP "192.168.0.19"
  #define _PRIMARY_NETMASK "255.255.255.0"
3 #define MY_NAMESERVER "132.248.10.2"
  #define MY_GATEWAY "192.168.0.254"
5 Puerto: 12007

```


Subsistema a probar: Salidas de alta corriente de la placa BL4S200.

Localización y fecha de la prueba: Laboratorio de Electrónica y Detectores del Instituto de Astronomía. Del 16 abril de 2015 al 3 de mayo de 2015.

Salidas anticipadas: Torque proporcionado por los puentes H de la placa BL4S200 iguales o mayores a los requeridos.

Procedimiento de prueba:

Se utilizó un equipo confiable de diseño de hardware, el cual se controlaba por simples bits del código de control para verificar que los motores fueran correctamente manejados bajo el control del software. Los osciloscopios de almacenamiento digital se utilizaron para observar los niveles de ruido y verificar que los transitorios se encontraban dentro de las tolerancias de diseño. Además de verificar la generación de la modulación de ancho de pulso, también se definieron los rangos de resolución de las características de la corriente de excitación, frecuencia y ciclo de trabajo.

Para las pruebas se utilizaron dos motores de corriente directa (DC), conectados de tal manera que tuvieran pares en sentidos opuestos y unidos mediante un cople. Cada motor tiene las características de los cuadros 6.1, 6.2, 6.3 y 6.4; se hizo uso de una fuente de voltaje de laboratorio, para producir un torque al *Motor 500, 1000 B* que representaba la carga para los motores de los carros móviles, a este motor en todas las pruebas se le suministraron 10 V. Se utilizó un tacómetro digital **DT-2234C⁺**, para medir las revoluciones por minuto del par total. Por último para medir los cambios de voltaje y corriente, se hizo uso de un multímetro digital marca **Fluke modelo 87**, que proporcionaba valores mínimos, máximos y promedio para cada medición en el sistema de pruebas, con un tiempo de muestreo de 100 ms.

Cuadro 6.1: Motor que representa a los carros móviles:

| Globe Motors | | | | | |
|--------------|----------------------|--------------------|--------------------------|-----------------------|--|
| Voltaje [V] | Multiplicador de par | Número de pre-fijo | Par máximo continuo [Nm] | Relación de reducción | |
| 24 | 10 | 409A6015-3 | 0.247154 | 16 | |

Cuadro 6.2: Motor del brazo de prismas

| Pittman Motors | | | | | | |
|----------------|---------------------------|--------------------|--------------------------|-----------------------|--|--|
| Voltaje [V] | Velocidad sin carga (rpm) | Número de pre-fijo | Par máximo continuo [Nm] | Relación de reducción | | |
| 12 | 13 | GM9413-4 | 1.2 | 218.4 | | |

Cuadro 6.3: Motor de los carros móviles, para el *étalon*, la unidad móvil de calibración y la rueda de filtros.

| Pittman Motors | | | | | | |
|----------------|---------------------------|--|--------------------|--------------------------|--|-----------------------|
| Voltaje [V] | Velocidad sin carga (rpm) | | Número de pre-fijo | Par máximo continuo [Nm] | | Relación de reducción |
| 12 | 13 | | GM9413-3 | 1.2 | | 65.5 |

Cuadro 6.4: Motor que representa la carga que tienen los subsistemas del instrumento PUMA:

| Galil Motors | | | | | | |
|--------------|---------------------------|--|--------------------|--------------------------|--|-----------------------|
| Voltaje [V] | Velocidad sin carga (rpm) | | Número de pre-fijo | Par máximo continuo [Nm] | | Relación de reducción |
| 60 | 3750 | | Motor 500/1000 B | 1.09454 | | NA |

Mediciones de voltaje. La variación de tensión se midió colocando las puntas del multimetro digital en paralelo, entre las conexiones del puente H al motor 409A6015–3. Como lo muestra la Figura 6.5.

Mediciones de corriente. La variación de corriente se midió colocando las puntas del multimetro digital en serie, entre una de las conexiones del puente H al motor 409A6015–3. Como lo muestra la Figura 6.6.

Para la caracterización de la salida de alta corriente y la verificación de su funcionamiento junto con los motores del PUMA, se realizaron mediciones en varias condiciones de funcionamiento. La tabla de mediciones y datos de comparación de voltaje V , de corriente I y de velocidad angular ω respecto a diferentes porcentajes de ciclo de trabajo (PWM), de 40 a 90%, para diferentes rangos de frecuencias (de 50 Hz a 50 kHz que permite la placa), se encuentran en el apartado del Apéndice D. Para determinar los diferentes torques a las diferentes entradas y cambios de ciclo de trabajo y frecuencias, se evaluaron mediante la siguiente ecuación:

$$T = \frac{V \cdot I \cdot 60}{2\pi \cdot \omega} \quad (6.1)$$

En la que el torque es proporcional a la potencia en la velocidad angular de operación (Fitzgerald et al., 1981). Las unidades dimensionales del par T están en $[N \cdot m]$, las de voltaje $V = [V]$, corriente I en $[A]$ y las de velocidad angular ω en $[rpm]$. Como resultado se obtuvieron familias de curvas que representan el comportamiento de las salidas de alta corriente de la placa BL4S200.

En la gráfica de la Figura 6.7, se muestra la relación entre la velocidad que proporciona la salidas de corriente al motor sin carga a diferentes ciclos de trabajo, es casi lineal de los 5 kHz a los 50 kHz y del 30% al 80% del ciclo de trabajo.

En la gráfica de la Figura 6.8, se observa que a altas frecuencias a partir de 5000 kHz, se tiene un comportamiento lineal y constante para cada curva de comportamiento, esta prueba se realizó aplicando carga y sin carga; esto comprueba que no afecta su comportamiento en este rango de frecuencias. La curva que corresponde a 1 kHz muestra la no linealidad del proceso para frecuencias bajas del ciclo de trabajo.

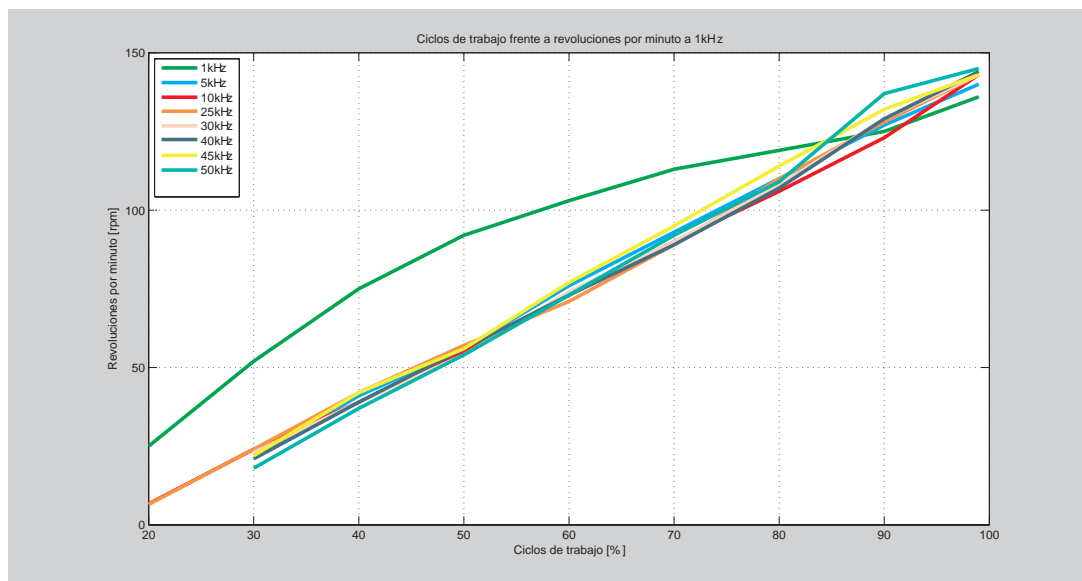


Figura 6.7: Comparación de las velocidades que alcanza un motor respecto a diferentes porcentajes de ciclo de trabajo a diferentes frecuencias.

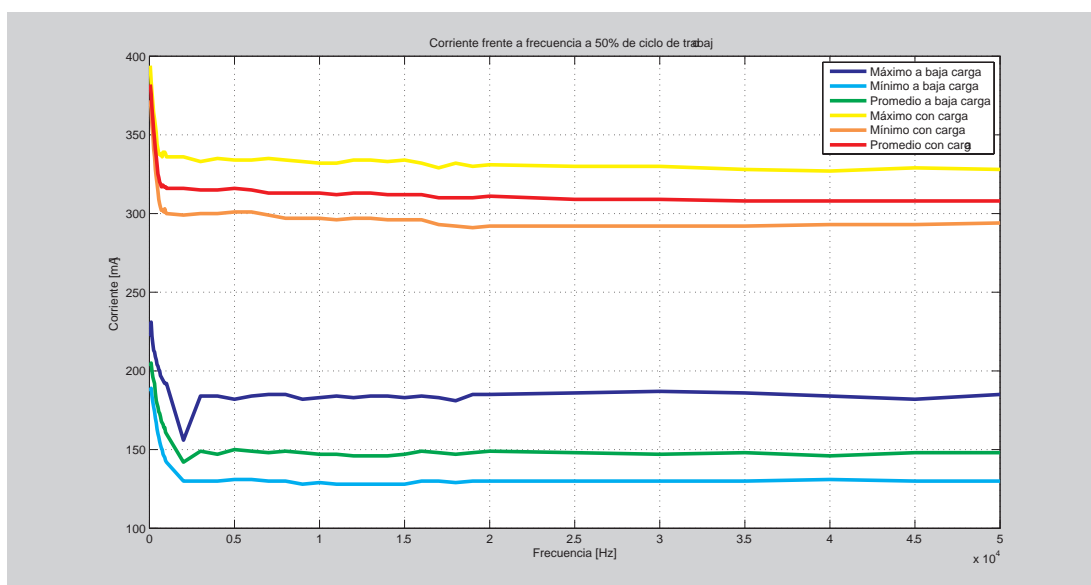


Figura 6.8: Comparación de los cambios de frecuencia frente a variaciones altas y bajas de corriente a diferentes porcentajes de ciclo de trabajo.

En la gráfica de la Figura 6.9, se observan las líneas de comportamiento de potencia ($V \cdot I$), con relación a la corriente, cuya suma de cada una de las curvas de comportamiento muestra una relación constante, en donde el ciclo de trabajo es proporcional a la potencia de salida.

En la gráfica de la Figura 6.10, se muestra un comportamiento similar al de la Figura 6.8, en el que a altas frecuencias la potencia se comporta como una constante para cada línea de comportamiento. Esta conducta se observa a partir de 500 Hz, aumentando la potencia conforme aumenta el ciclo de trabajo.

En la gráfica de la Figura 6.11, la tendencia constante de cada una de las líneas de comportamien-

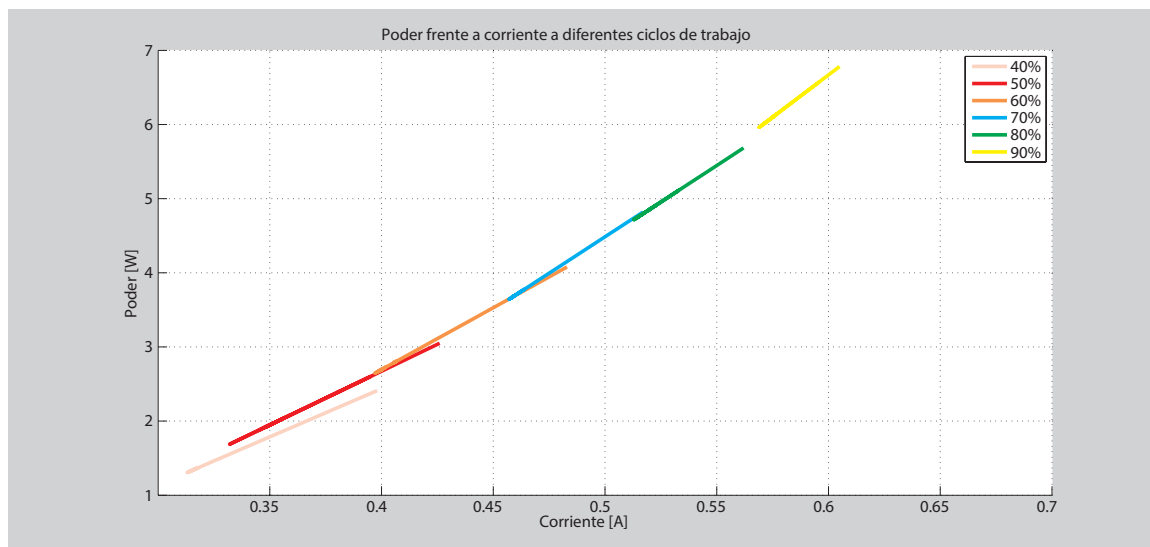


Figura 6.9: Comparación de los valores de potencia en watts respecto a diferentes variaciones de corriente a diferentes ciclos de trabajo. Las curvas de comportamiento se ajustaron a una función potencial del tipo $y = c_1 \cdot x^1 + c_0 \cdot x^0$.

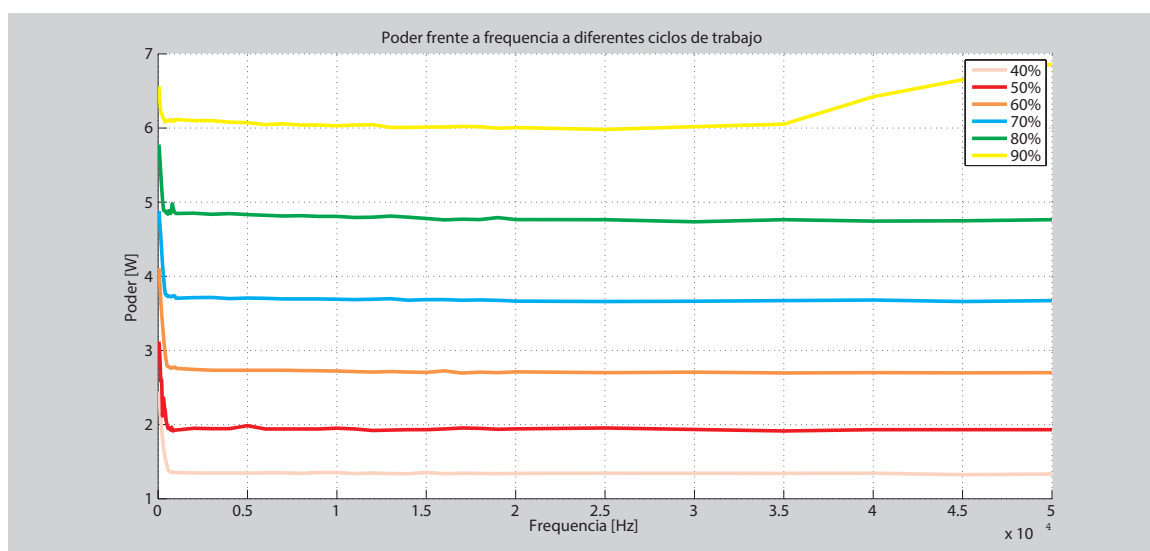


Figura 6.10: Comparación de los valores de potencia en watts respecto a diferentes valores de frecuencias a diferentes ciclos de trabajo.

to es perceptible en la relación de velocidad con respecto a la frecuencia, a partir de los 2000 Hz. El comportamiento para el ciclo de trabajo a 70% es un poco transitorio, el ciclo de trabajo a 90% alcanza la mayor velocidad cerca de 70 rpm, pero para mover los carros es suficiente una velocidad de 20 rpm a 30 rpm, ya que a una velocidad mayor podría ser mayor el impacto al detenerse y causaría daños en los instrumentos.

La gráfica de la Figura 6.12, es la más representativa ya que muestra la relación entre el torque de salida que producen las salidas de alta corriente al motor. Entre las curvas de comportamiento se observan tres que se comportan de manera diferente, siendo constantes a partir de los 5000 Hz a las otras casi lineales desde los 0 Hz. El mayor torque se produce con un ciclo de trabajo a 90%, llegando a $0.8 \text{ N} \cdot \text{m}$, con la misma tendencia se observa a 80%, de ciclo de trabajo con

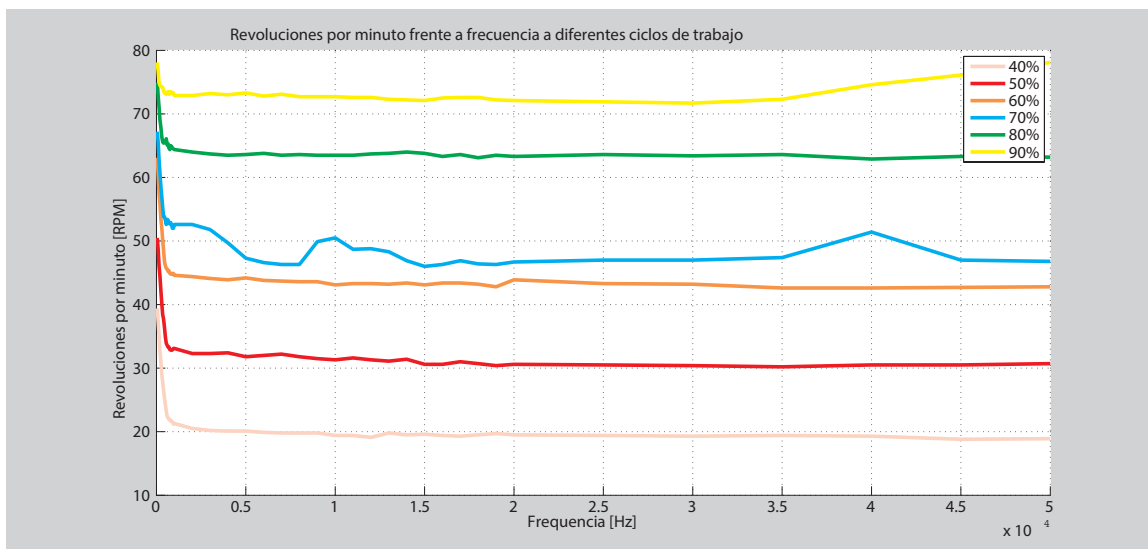


Figura 6.11: Comparación de los valores de velocidad angular respecto a diferentes valores de frecuencias a diferentes ciclos de trabajo.

aproximadamente $0.72 \text{ N} \cdot \text{m}$ y a 60%, de ciclo de trabajo con $0.6 \text{ N} \cdot \text{m}$. Las tres curvas restantes aumentan su valor llegando la de 70%, de ciclo de trabajo a $0.75 \text{ N} \cdot \text{m}$, la de 40%, de ciclo de trabajo a aproximadamente $0.68 \text{ N} \cdot \text{m}$ y la de 50%, de ciclo de trabajo a $0.61 \text{ N} \cdot \text{m}$.

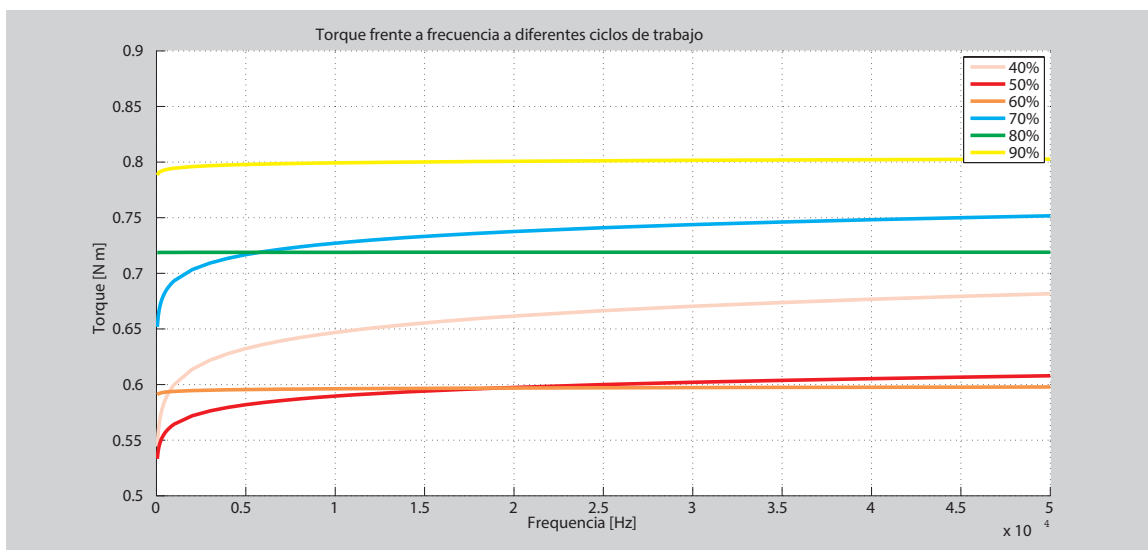


Figura 6.12: Comparación de los valores de torque respecto a diferentes valores de frecuencias a diferentes ciclos de trabajo. Las curvas de comportamiento se ajustaron a una función potencial del tipo $y = c \cdot x^a$.

En la gráfica de la Figura 6.13, se muestran las líneas de comportamiento para la relación torque y la velocidad angular. Se muestra que el máximo valor es para un ciclo de trabajo del 90% con un torque de $0.8 \text{ N} \cdot \text{m}$ aproximadamente 71 rpm y el menor de $0.75 \text{ N} \cdot \text{m}$ a 40 rpm.

La gráfica de la Figura 6.14, muestra una relación lineal entre el voltaje V y la corriente I de las mediciones, aumentando sus valores al aumentar sus ciclos de trabajo.

La gráfica de la Figura 6.15, muestra relaciones casi lineales de relación de voltaje respecto a la

frecuencia, al aumentar el valor de los ciclos de trabajo.

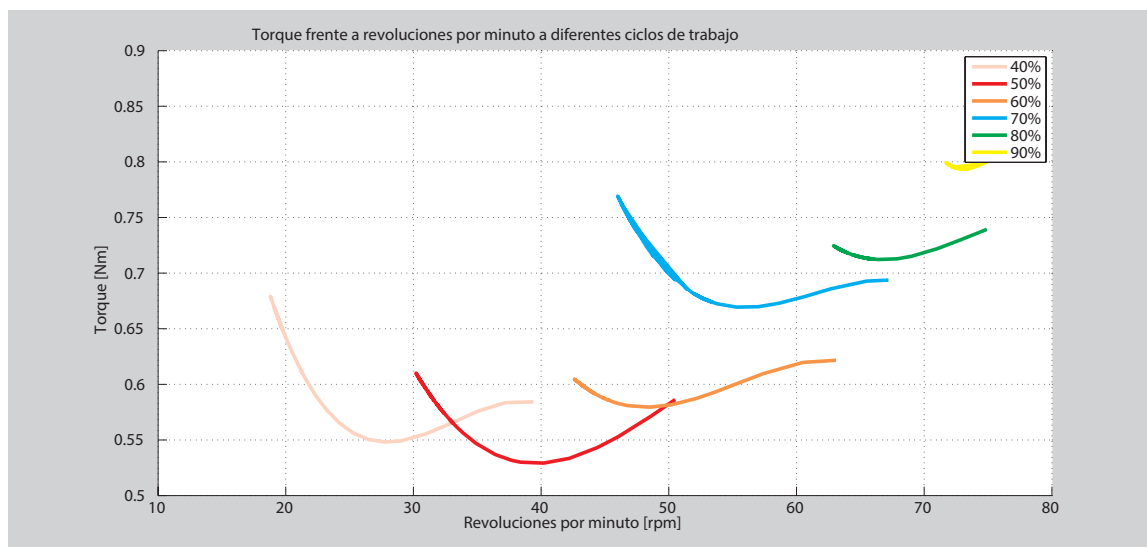


Figura 6.13: Comparación de valores de torque respecto a diferentes velocidades que alcanza un motor a diferentes porcentajes de ciclo de trabajo. Las curvas de comportamiento se ajustaron a una función potencial del tipo $y = c_3 \cdot x^3 + c_2 \cdot x^2 + c_1 \cdot x^1 + c_0 \cdot x^0$.

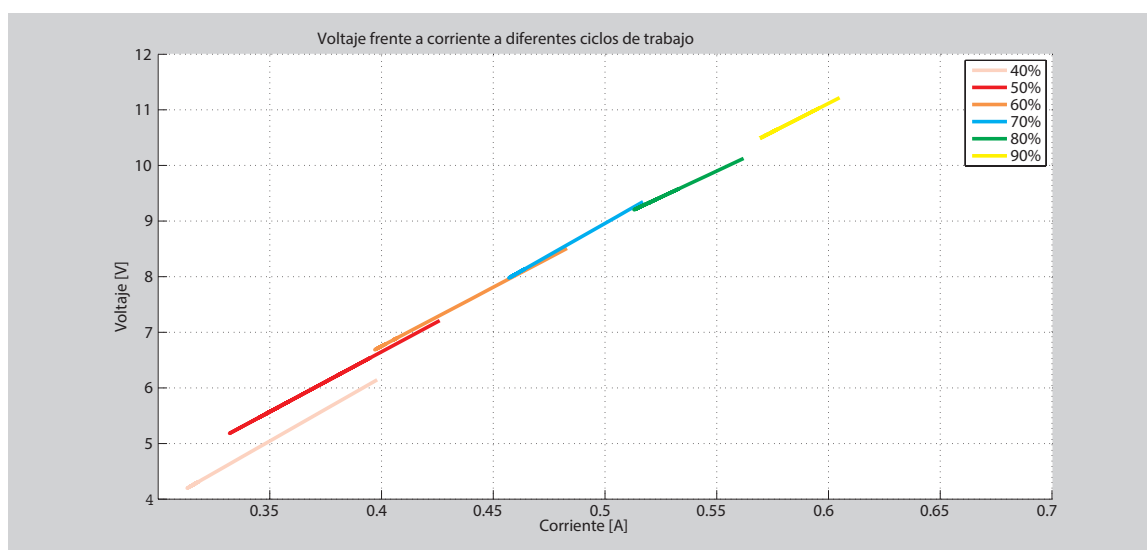


Figura 6.14: Comparación de los valores de voltaje respecto a valores de corriente a diferentes porcentajes de ciclo de trabajo. Las curvas de comportamiento se ajustaron a una función potencial del tipo $y = c_1 \cdot x^1 + c_0 \cdot x^0$.

Cálculos de par de los carros móviles. Se calcularán los torques mínimos requeridos con detalle, considerando la rueda de filtros con un movimiento rotatorio, los demás movimientos son lineales, movimientos automatizados determinados por finales de carrera y sensores de proximidad inductivos. Los valores dimensionales de masa, centro de gravedad etc., de los componentes del instrumento PUMA se calcularon con el software de diseño asistido por computadora CATIA V5, para las siguientes evaluaciones.

Actuador de cadena para el étalon (Figura 6.16). La masa total de la carga y de la cadena transportadora es de $m=3.041$ kg, La carga es la suma de la cadena más el interferómetro. La masa de

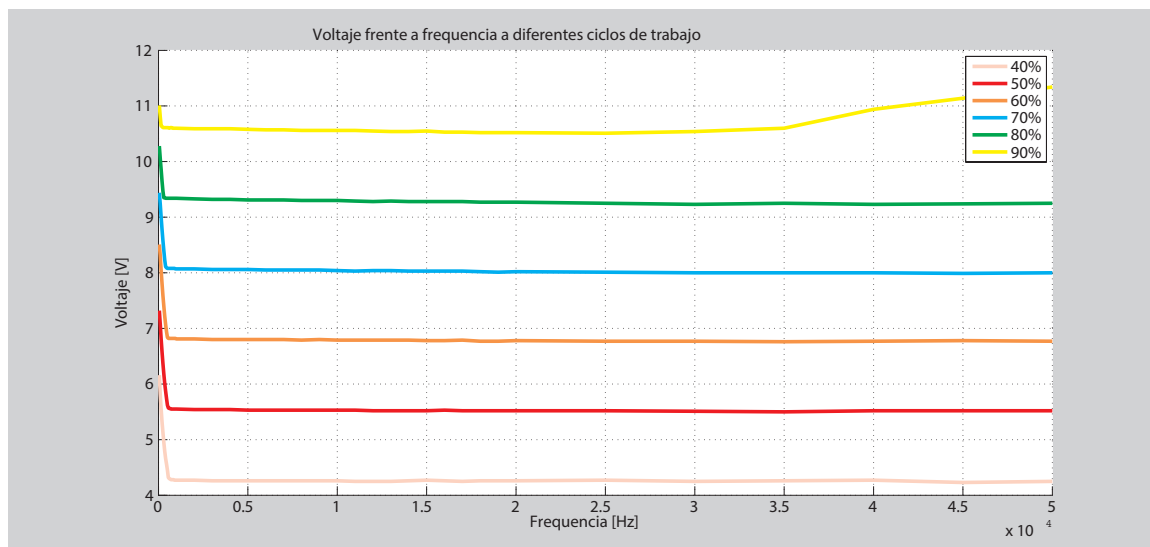


Figura 6.15: Comparación de los valores de voltaje respecto a variaciones de frecuencia a diferentes porcentajes de ciclo de trabajo.

la cadena se calculó con la densidad del acero, las partes estructurales del FP se consideraron de aluminio y los componentes de paralelización del etalón del material del espejo. De acuerdo a (Bearings, 2016), el máximo valor de coeficiente de fricción para los mecanismos de transmisión es de 0.05.

Las especificaciones de la única catarina de transmisión $n=1$ pc, se consideraron para el material acero, $D_p=24.86$ mm. La masa de la catarina es de $m_p=0.008$ kg y se requiere una eficiencia del $\eta=100\%$. No existe una fuerza externa adicional al sistema $F_A=0$ N y no hay otras catarinas, poleas o engranes de transmisión, puesto que se utiliza una estructura de acoplamiento directo. Adicionalmente es necesario sostener la carga después de que el motor se pare y es necesario mantenerse en posición después de que la fuente de alimentación se apague. Se considerará una velocidad fija de operación de $V_1=30$ mm/s. La precisión de detención es de una diferencia de $\Delta l=0$ mm y se tomará como factor de seguridad a $SF=1.5$.

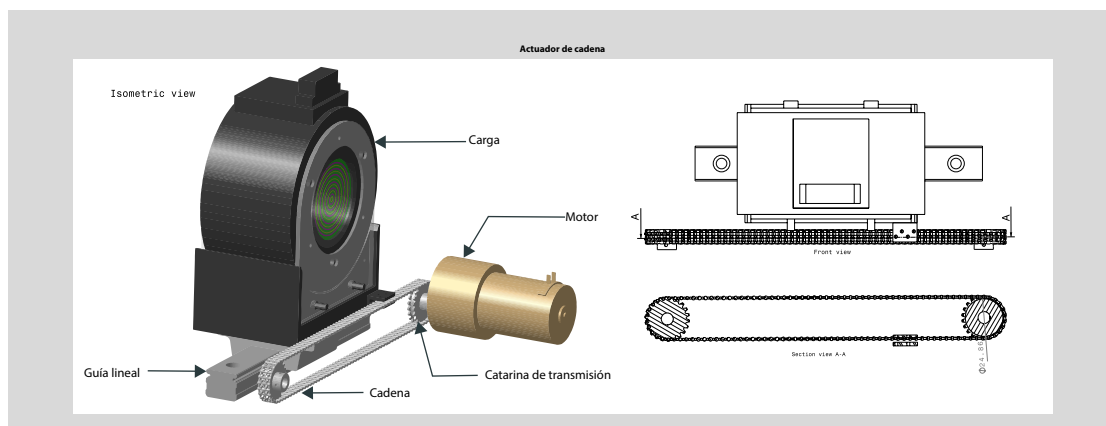


Figura 6.16: Análisis dimensional del interferómetro.

La inercia de carga se calcula como sigue.

$$J_m = m \cdot \left(\frac{D_p \cdot 10^{-3}}{2} \right)^2 = 3,041 \cdot \left(\frac{24,86 \cdot 10^{-3}}{2} \right)^2 = 4,698e-4 [kg \cdot m^2] \quad (6.2)$$

$$J_{DP} = \frac{1}{8} \cdot m_p (D_p \cdot 10^{-3})^2 \cdot n = \frac{1}{8} \cdot 0,008 (24,86 \cdot 10^{-3})^2 \cdot 1 = 6,180e-7 [kg \cdot m^2] \quad (6.3)$$

$$J_L = J_m + J_{DP} = 4,698e-4 + 6,180e-7 = 4,705 [kg \cdot m^2] \quad (6.4)$$

La velocidad requerida será:

$$V_m = V_1 \cdot \left(\frac{60}{\pi D_p} \right) = 30 \cdot \left(\frac{60}{\pi 24,86} \right) = 23,06 [rmin] \quad (6.5)$$

Par de carga

$$F = F_A + (m \cdot 9,8) \cdot (\sin \alpha + \mu \cos \alpha) = 0 + (3,041 \cdot 9,8) \cdot (\sin 0 + 0,05 \cdot \cos 0) = 1,490 [N] \quad (6.6)$$

$$T_L = \frac{F \cdot (D_p \cdot 10^{-3})}{2 \cdot \eta \cdot 0,001} = \frac{1,490 \cdot (24,86 \cdot 10^{-3})}{2 \cdot 100 \cdot 0,001} = 1,852e-2 [N \cdot m] \quad (6.7)$$

Par requerido

$$T = (T_a + T_L) \cdot SafetyFactor = (0 + 1,852e-2) \cdot 1,5 = 2,778e-2 [N \cdot m] \quad (6.8)$$

Aplicación de brazo para prismas (Figura 6.17). Entre las dimensiones del brazo, la longitud total del brazo es de A=156.5 mm, el ancho se considerará el diámetro mayor de la figura B=35.5 mm y la variación longitudinal de r=12 mm, la masa total del brazo es de m=0.168 kg, en donde los prismas se consideraron de plastiglas y la estructura del brazo, así como los soportes para los prismas se consideraron de aluminio.

Para las condiciones del mecanismo, este operará de manera vertical y tendrá una precisión de parada prescrita de $\Delta\theta = 90^\circ$, teniendo el sistema una eficiencia de $\eta = 100\%$ y no existen catarinas, poleas o engranes de transmisión, puesto que se utiliza una estructura de acoplamiento directo. Adicionalmente es necesario sostener la carga después de que el motor se pare y es necesario mantenerse en posición después de que la fuente de alimentación se apague. Se considerará una velocidad fija de operación de $V_1=30$ mm/s, y se tomará como factor de seguridad a SF=1.5.

La inercia de carga se calcula como sigue.

$$J_m = \frac{1}{12} \cdot m \cdot \left((A \cdot 10^{-3})^2 + (B \cdot 10^{-3})^2 + 12 \cdot (r \cdot 10^{-3}) \right) \quad (6.9)$$

$$J_m = \frac{1}{12} \cdot 0,168 \cdot \left((156,5 \cdot 10^{-3})^2 + (35,5 \cdot 10^{-3})^2 + 12 \cdot (12 \cdot 10^{-3}) \right) = 3,847e-4 [kg \cdot m^2] \quad (6.10)$$

$$J_L = J_m = 3,847e-4 [kg \cdot m^2] \quad (6.11)$$

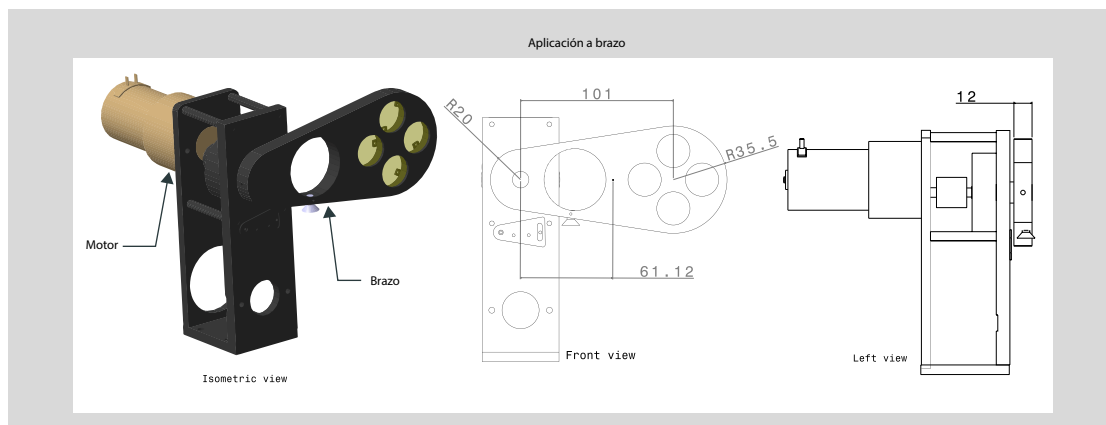


Figura 6.17: Análisis dimensional del brazo de prismas.

La velocidad requerida

$$V_m = V_1 = 30[r \text{ mín}] \quad (6.12)$$

Par de carga

$$F = 9,8 \cdot m = 9,8 \cdot 1,680e - 1 = 1,646[N] \quad (6.13)$$

$$T_L = F(r \cdot 10^{-3}) \left(\frac{1}{\eta \cdot 0,01} \right) = 1,646(12 \cdot 10^{-3}) \left(\frac{1}{100 \cdot 0,01} \right) = 1,976e - 2[N \cdot m] \quad (6.14)$$

Par requerido

$$T = (T_a + T_L) \cdot SafetyFactor = (0 + 1,976e - 2) \cdot 1,5 = 2,964e - 2[N \cdot m] \quad (6.15)$$

Aplicación de rueda indexada para filtros (Figura 6.18). Para el análisis dimensional se consideraron elementos estructurales de aluminio, entre los cuales la rueda de filtros se evaluó como una superficie redonda de diámetro de $D=250$ mm y masa de $m=0.94$ kg, y el eje de transmisión con diámetro de $D_2 = 8.5$ mm y masa de $m_2 = 0.06$ kg.

La carga se considerará dimensionalmente de forma cilíndrica con un diámetro de $D_l = 45.6$ mm, y con una distancia del centro de la rueda al centro de la carga de $r=90$ mm, con un número de cargas o filtros de $n=7$ pc, y una masa de $m_l = 0.046$ kg por carga. Entre la rueda y el mecanismo de soporte existirá un coeficiente de fricción de $\mu = 0.05$, con una distancia de contacto entre la rueda y el mecanismo de $l=16$ mm, se requiere una eficiencia del η 100%. El sistema de filtros para su movimiento tiene dos poleas dentadas, la polea primaria tiene un diámetro de $D_{p1}=25$ mm y una masa de $m_{p1} = 0.019$ kg; la segunda polea es de un diámetro de $D_{p2} = 86$ mm y masa $m_{p2} = 0.231$ kg, ambas consideradas de aluminio.

El mecanismo tendrá una condición horizontal de operación, adicionalmente es necesario sostener la carga después de que el motor se pare y es necesario mantenerse en posición después de que la fuente de alimentación se apague. Funcionando a una velocidad fija de operación de $V_1 = 30$ mm/s, con una precisión de detención a $\Delta\theta = 45^\circ$ y un factor de seguridad de $SF = 1.5$.

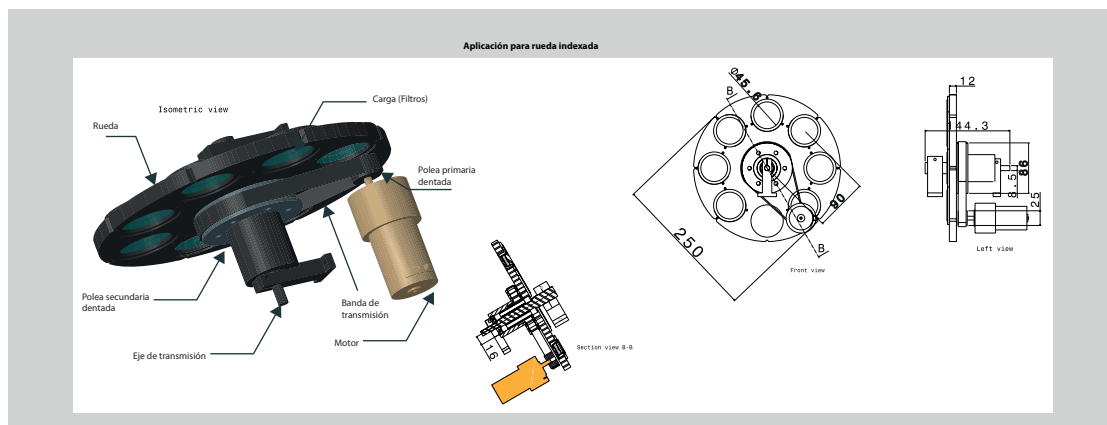


Figura 6.18: Análisis dimensional de la rueda de filtros

La inercia de carga se calcula como sigue.

$$J_t = \frac{1}{8} \cdot m \cdot (D \cdot 10^{-3})^2 = \frac{1}{8} \cdot 0,94 \cdot (250 \cdot 10^{-3})^2 = 7,344e-3 [kg \cdot m^2] \quad (6.16)$$

$$J_s = \frac{1}{8} \cdot m_2 \cdot (D_2 \cdot 10^{-3})^2 = \frac{1}{8} \cdot 0,06 \cdot (8,5 \cdot 10^{-3})^2 = 5,419e-7 [kg \cdot m^2] \quad (6.17)$$

$$J_l = \left(\frac{1}{8} \cdot m_l \cdot (D_l \cdot 10^{-3})^2 + m_l \cdot (r \cdot 10^{-3})^2 \right) \cdot n \quad (6.18)$$

$$J_l = \left(\frac{1}{8} \cdot 0,046 \cdot (45,6 \cdot 10^{-3})^2 + 0,046 \cdot (90 \cdot 10^{-3})^2 \right) \cdot 7 = 2,692e-3 [kg \cdot m^2] \quad (6.19)$$

$$J_{Dp1} = \frac{1}{8} \cdot m_{p1} \cdot (D_{p1} \cdot 10^{-3})^2 = \frac{1}{8} \cdot 0,019 \cdot (25 \cdot 10^{-3})^2 = 1,484e-6 [kg \cdot m^2] \quad (6.20)$$

$$J_{Dp2} = \frac{1}{8} \cdot m_{p2} \cdot (D_{p2} \cdot 10^{-3})^2 = \frac{1}{8} \cdot 0,231 \cdot (86 \cdot 10^{-3})^2 = 2,136e-4 [kg \cdot m^2] \quad (6.21)$$

$$J_L = (J_t + J_s + J_l + J_{Dp2}) \cdot \left(\frac{D_{p1}}{D_{p2}} \right)^2 + J_{Dp1} \quad (6.22)$$

$$J_L = (7,344e-3 + 5,419e-7 + 2,692e-3 + 2,136e-4) \cdot \left(\frac{25}{86} \right)^2 + 1,484e-6 = 8,676e-4 [kg \cdot m^2] \quad (6.23)$$

La velocidad requerida será

$$V_m = V_1 \cdot \left(\frac{D_{p2}}{D_{p1}} \right) = 30 \cdot \left(\frac{86}{25} \right) = 103 [r/min] \quad (6.24)$$

Par de carga

$$T_L = 9,8 \cdot (m + m_l \cdot n) \cdot \mu \cdot (l \cdot 10^{-3}) \left(\frac{1}{\eta \cdot 0,01} \right) \left(\frac{D_{p1}}{D_{p2}} \right) \quad (6.25)$$

$$T_L = 9,8 \cdot (9,4e-1 + 4,6e-2 \cdot 7) \cdot 0,05 \cdot (16 \cdot 10^{-3}) \left(\frac{1}{100 \cdot 0,01} \right) \left(\frac{25}{86} \right) = 2,876e-3 [N \cdot m] \quad (6.26)$$

Par requerido

$$T = (T_a + T_L) \cdot SafetyFactor = (0 + 2,876e-3) \cdot 15 = 4,314e-3 [N \cdot m] \quad (6.27)$$

Precisión de detención requerida

$$\Delta\theta = \Delta\theta \left(\frac{D_{p2}}{D_{p1}} \right) = 45 \cdot \left(\frac{86}{25} \right) = 154,8 [deg] \quad (6.28)$$

Actuador de cadena para camóvil (Figura 6.19). La masa total de la carga y de la cadena transportadora es de $m=1.748$ kg, La carga es la suma de la cadena más el sistema de calibración. La masa de la cadena se calculó con la densidad del acero, las partes estructurales del sistema de calibración se consideraron de aluminio más la masa del espejo. De acuerdo a (Bearings, 2016), el máximo valor de coeficiente de fricción para los mecanismos de transmisión es de 0.05.

Las especificaciones de la única catarina de transmisión $n=1$ pc, se consideraron para el material acero $D_p = 24.86$ mm. La masa de la catarina es de $m_p = 0.008$ kg y se requiere una eficiencia del $\eta=100\%$. No existe una fuerza externa adicional al sistema $F_A=0$ N y no hay otras catarinas, poleas o engranes de transmisión, puesto que se utiliza una estructura de acoplamiento directo. Adicionalmente es necesario sostener la carga después de que el motor se pare y es necesario mantenerse en posición después de que la fuente de alimentación se apague. Se considerará una velocidad fija de operación de $V_1 = 30$ mm/s. La precisión de detención es de una diferencia de $\Delta l=0$ mm y se tomará como factor de seguridad a $SF=1.5$.

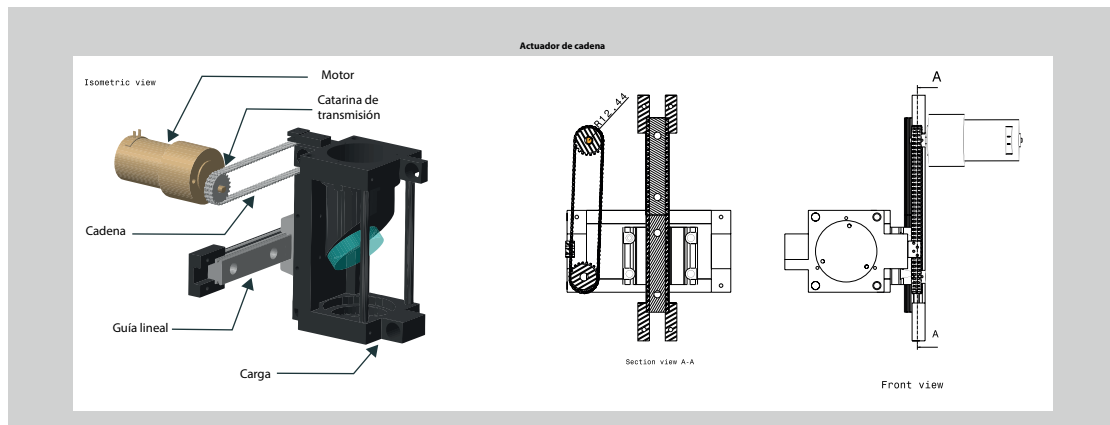


Figura 6.19: Análisis dimensional de la unidad de calibración móvil.

La inercia de carga se calcula como sigue.

$$J_m = m \cdot \left(\frac{D_p \cdot 10^{-3}}{2} \right)^2 = 1,748 \cdot \left(\frac{24,86 \cdot 10^{-3}}{2} \right)^2 = 2,701e-4 [\text{kg} \cdot \text{m}^2] \quad (6.29)$$

$$J_{DP} = \frac{1}{8} \cdot m_p (D_p \cdot 10^{-3})^2 \cdot n = \frac{1}{8} \cdot 0,008 (24,86 \cdot 10^{-3})^2 \cdot 1 = 6,180e-7 [\text{kg} \cdot \text{m}^2] \quad (6.30)$$

$$J_L = J_m + J_{DP} = 2,701e-4 + 6,180e-7 = 2,707e-4 [\text{kg} \cdot \text{m}^2] \quad (6.31)$$

La velocidad requerida será:

$$V_m = V_1 \cdot \left(\frac{60}{\pi D_p} \right) = 30 \cdot \left(\frac{60}{\pi 24,86} \right) = 23,06 [\text{rpm}] \quad (6.32)$$

Par de carga

$$F = F_A + (m \cdot 9,8) \cdot (\sin \alpha + \mu \cos \alpha) = 0 + (1,748 \cdot 9,8) \cdot (\sin 0 + 0,05 \cdot \cos 0) = 8,565e-1 [\text{N}] \quad (6.33)$$

$$T_L = \frac{F \cdot (D_p \cdot 10^{-3})}{2 \cdot \eta \cdot 0,001} = \frac{8,565e-1 \cdot (24,86 \cdot 10^{-3})}{2 \cdot 100 \cdot 0,001} = 1,065e-2 [\text{N} \cdot \text{m}] \quad (6.34)$$

Par requerido

$$T = (T_a + T_L) \cdot \text{SafetyFactor} = (0 + 1,065e-2) \cdot 1,5 = 1,597e-2 [\text{N} \cdot \text{m}] \quad (6.35)$$

Se observa que los pares generados en las pruebas son más que suficientes para producir los movimientos de los carros de los subsistemas del instrumento PUMA, según los pares calculados. Ya que el valor de par máximo que se puede lograr, según la gráfica de torque respecto a frecuencia, es de $0.8 \text{ N} \cdot \text{m}$, y el máximo par requerido según los cálculos previos es de $2.964e-2 \text{ N} \cdot \text{m}$ que sería para el brazo de prismas, que sin problema puede funcionar por que además tiene una relación de reducción mayor que el motor que se utilizó de prueba. Más gráficas se muestran en el Apéndice C.

Cuadro 6.5: Resultados de los cálculos teóricos para el análisis mecánico del movimiento de los carros móviles.

| Subsistema | Inercia de carga [$\text{kg} \cdot \text{m}^2$] | Velocidad requerida [rpm] | Par de carga [$\text{N} \cdot \text{m}$] | Par requerido [$\text{N} \cdot \text{m}$] |
|-----------------------------------|---|---------------------------|--|---|
| Posicionador de elementos ópticos | 4.705 | 23.06 | $1,852 \cdot 10^{-2}$ | $2,778 \cdot 10^{-2}$ |
| Brazo para prismas | $3,847 \cdot 10^{-4}$ | 30 | $1,976 \cdot 10^{-2}$ | $2,964 \cdot 10^{-2}$ |
| Rueda indexada para filtros | $8,676 \cdot 10^{-4}$ | 103 | $2,876 \cdot 10^{-3}$ | $4,314 \cdot 10^{-3}$ |
| Unidad de calibración móvil | $2,707 \cdot 10^{-4}$ | 23.06 | $1,065 \cdot 10^{-2}$ | $1,597 \cdot 10^{-2}$ |

CAPÍTULO

7

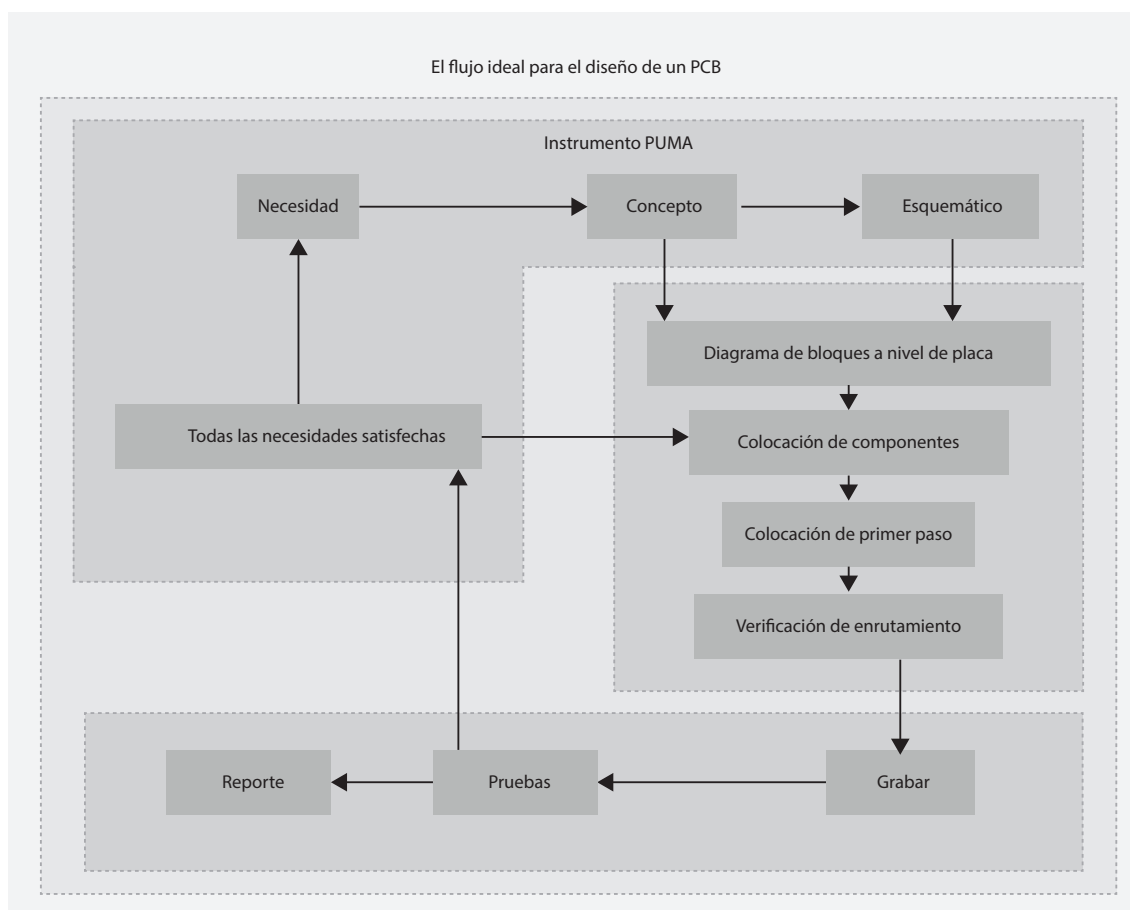
Módulo de control electrónico

En este capítulo se describe la electrónica diseñada para la interconexión del sistema.

Índice

| | | |
|-----|--|-----|
| 7.1 | Diseño del circuito impreso | 116 |
| 7.2 | Diagrama de bloques a nivel de placa | 117 |
| 7.3 | Componentes del hardware | 117 |
| 7.4 | Esquemática de enrutamiento | 121 |
| 7.5 | Verificación de enrutamiento | 123 |
| 7.6 | Pruebas | 123 |

7.1 Diseño del circuito impreso



Cuadro 7.1: Flujo ideal para el diseño de un PCB. Referencia (Smith, 2013).

El proceso del diseño básico e ideal de una tarjeta de circuito impreso (PCB), comienza con la necesidad de elaborar un PCB y continúa a través de las etapas finales de fabricación (Cuadro 7.1). Después de determinar por qué se necesita el PCB, el concepto final del circuito para la placa de prototipo debe ser decidido. El concepto incluye las características del diseño, las funciones de la tarjeta que debe tener y que debe llevar a cabo, la interconexión con otros circuitos, la colocación y las dimensiones finales aproximadas.

El diseño de una placa auxiliar se realizó para proporcionar a los componentes una interfaz simple y una distribución de alimentación. La tarjeta de circuito impreso (PCB), junto con los circuitos integrados, son generalmente herramientas usadas para un diseño a bajo costo, con capacidad para modelar el circuito y para probar el diseño metódicamente.

Como un sistema con múltiples entradas y salidas, utilizadas puramente para funciones lógicas, el PCB organiza los mazos de cables que llevan las señales digitales hacia unos conectores tipo borneras de tornillo. A los circuitos integrados se les acompaña de una resistencia en serie (tole-

rada y calculada), en las salidas y en las entradas. Para los bajos requisitos de entrada de corriente se asegura que la caída de tensión, en una resistencia en serie, sea mínima.

Sin embargo para las aplicaciones que requieren mayor conducción de corriente se añaden dos circuitos integrados que contienen dos canales de conductores Darlington, cada uno. Ambos dispositivos tienen diodos de supresión de retorno integrales. En particular, las corrientes de carga superiores se pueden conmutar, con corrientes de excitación más bajas. Como se conducen relés electromecánicos, los diodos de supresión de retorno protegen los transistores de la fuerza contra-electromotriz generada por la bobina del relevador. Cuando una aplicación llama a un interruptor con una baja resistencia de contacto, un relevador electromecánico es a veces la mejor opción, es por ello que así se manejan las señales para las lámparas.

El módulo a nivel controlador y en forma de tarjeta PCB, estará diseñado para proporcionar un método pragmático que acomoda y fija las entradas y salidas del microcontrolador a los dispositivos y sensores que componen la aplicación de control. Un módulo principal estará construido sobre una placa de circuito impreso (similar a muchos de los pequeños controladores a nivel de placa), será físicamente pequeño, de bajo costo y de fácil conexión a la placa portadora específica de la aplicación, el microcontrolador Rabbit.

El módulo tendrá los sistemas electrónicos ya preparados, compuestos por dispositivos de montaje superficial (SMT). Los componentes estarán en una tarjeta compacta con fácil acceso a los conectores de tornillo. El módulo básico tendrá juntas de desarrollo disponibles. El diseño del módulo puede ser copiado libremente a un PCB, personalizado para la producción si es necesario. Por lo tanto, un técnico puede tener el módulo como una sola tarjeta totalmente funcional e inmediatamente enchufarlo en la nueva aplicación en el diseño embebido.

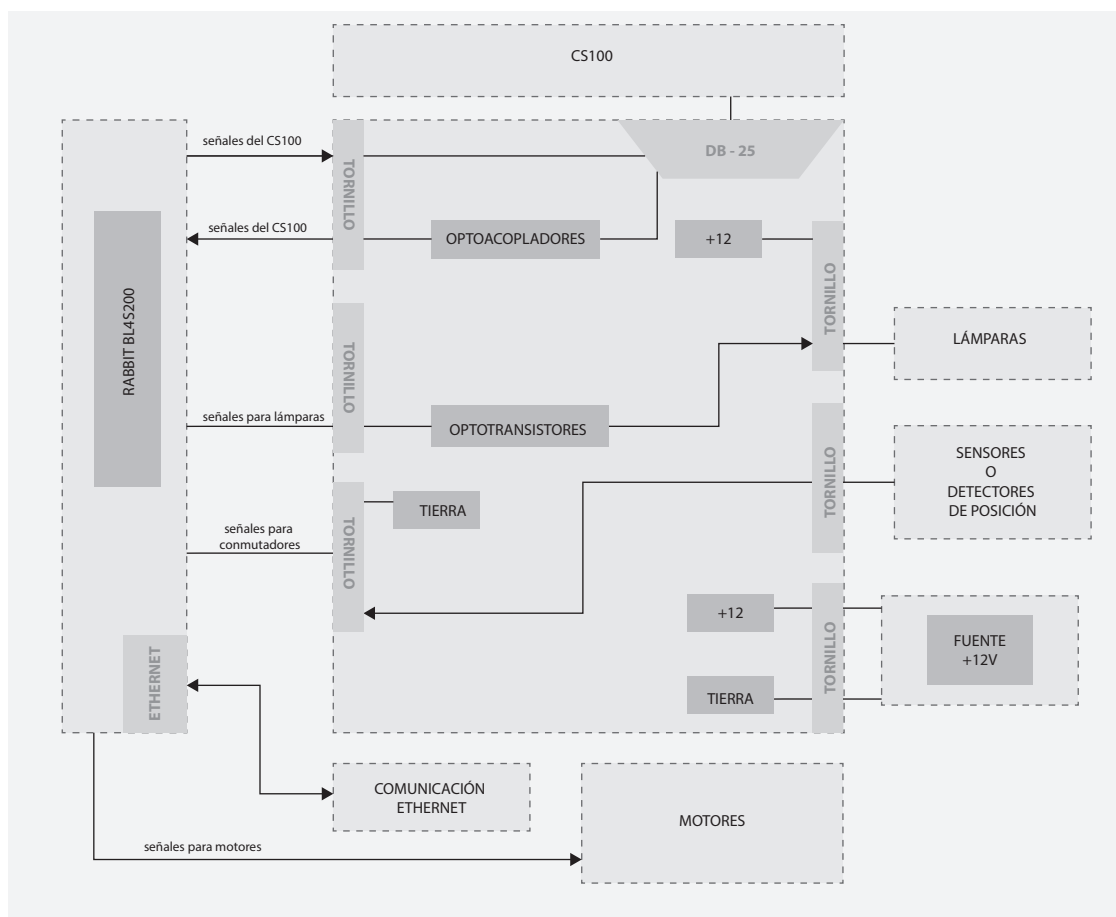
7.2 Diagrama de bloques a nivel de placa

El módulo de control y distribución electrónica tendrá interconexión con la placa Rabbit, el controlador CS100, el sistema de encendido de las lámparas y los sensores e interruptores de posición del instrumento PUMA; así como la fuente de voltaje que alimenta al sistema electrónico y de control del instrumento. El diagrama de bloques se muestra en el Cuadro 7.2.

7.3 Componentes del hardware

Para la placa de circuito impreso (PCB), se utilizan dispositivos integrados de montaje superficial (SMD). El diseño de la placa incluye partes y algunas conexiones de montaje superficial de paso fino.

Los conectores de terminales tipo bornera de tornillo, que se fijan a la placa controladora, requieren que primero se monte el controlador y luego se atornillen los cables en el PCB. Un técnico



Cuadro 7.2: Diagrama de bloques para el diseño del módulo electrónico.

puede querer reemplazar un nuevo controlador, con una buena unidad conocida. El tener que desenroscar un gran número de borneras de tornillo y volver a conectarlos en el lugar correcto de la placa puede ser un problema, pero se aseguró este procedimiento con un código de colores para los cables y su reinserción adecuada.

No obstante para las 20 conexiones con el controlador CS100, volver a conectar todos los cables a un nuevo controlador consume tiempo y es potencialmente propenso a errores. Una alternativa atractiva a las terminales de tornillo que se fijan a una placa del controlador es una clase de conectores enchufables. Este sistema permite que el mazo de cables se reemplacen con un zócalo de terminales, que se conecta a una cabecera fija en la placa PCB. Esta disposición permite el montaje rápido de los sistemas empotrados, así como con el reemplazo rápido y libre de errores del PCB, sin importar demasiado que la compensación por esta facilidad de conexión sea un pequeño aumento en el costo.

En el archivo Adjunto E, se muestran las hojas de datos de los componentes electrónicos utilizados en este diseño de PCB. El uso de tecnología de montaje superficial (SMT), es por que los dispositivos reducen el tamaño de la PCB y el número de sus pistas, además de que no atraviesan la tarjeta.

7.3.1 Etapa de optoacoplamiento

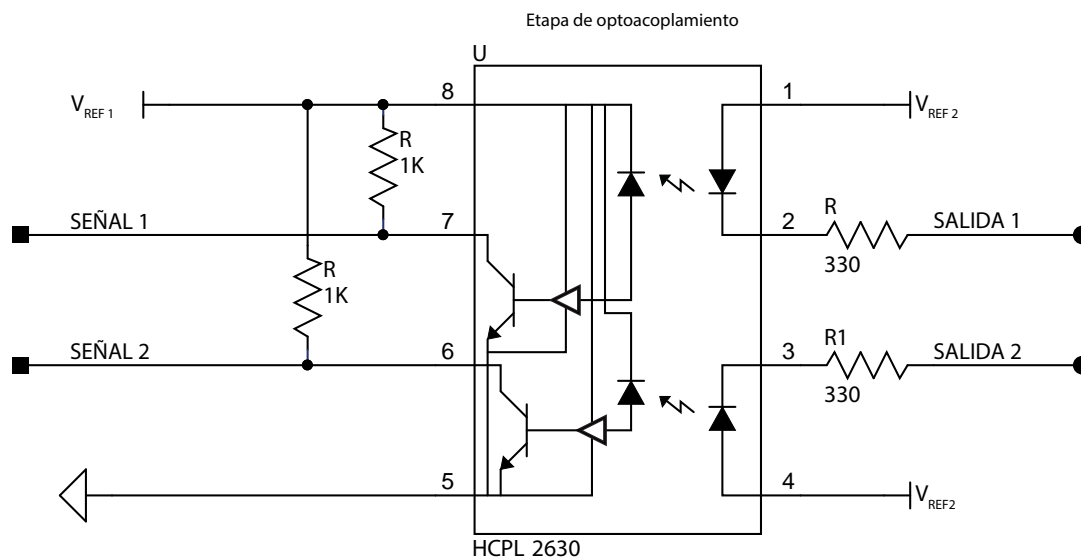


Figura 7.1: Diagrama electrónico para el optoacoplador HCPL2630SV.

Esta etapa está destinada a la interfaz entre las señales de salida del CS100 y el microcontrolador Rabbit. Se utilizaron optoacopladores HCPL2630SV (Figura 7.1), con tecnología de montaje superficial, para aislar distintas fuentes del sistema y no producir lazos anidados de corriente con probabilidad de cortos.

En el cálculo de las resistencias, para la integración de los optoacopladores con el sistema, se consideró la corriente $I_F=10\text{ mA}$ y el voltaje del LED de 1.8 V.

$$0 = 5[V] - 1,8[V] - I_F \cdot R_F \tag{7.1}$$

$$5[V] = 1,8[V] + I_F \cdot R_F \tag{7.2}$$

$$R_F = \frac{3,2[V]}{10[mA]} = 320[\Omega] \tag{7.3}$$

Se utilizó una resistencia de valor nominal a 330 Ω . Como la corriente del colector es de $I_c = 15\text{ mA}$, la resistencia R para tener este valor, cuando el transistor está polarizado, es:

$$I_c = 15[mA] \tag{7.4}$$

$$12[V] = R \cdot I_c \tag{7.5}$$

$$R = \frac{12[V]}{15[mA]} = 800[\Omega] \tag{7.6}$$

Se obtuvo un valor de 800Ω como mínimo, por lo que se escoge un valor de $1 k\Omega$. La potencia del resistor sería:

$$P_R = R \cdot I_c^2 \quad (7.7)$$

$$P_R = (1000) (0,015)^2 = 0,225 [W] \quad (7.8)$$

7.3.2 Etapa de control de lámparas

Dado que las lámparas funcionan con corriente alterna, se requiere una etapa de acoplamiento entre el microcontrolador y los relevadores que manejan las corrientes alternas de las lámparas. El dispositivo que se usó en la etapa de potencia es un optotransistor TLP127 (Figura 7.2), con tecnología de montaje superficial, para el control de los relevadores que conmutan el encendido/apagado de las lámparas. El transistor que tiene integrado es de tipo Darlington en configuración NPN.

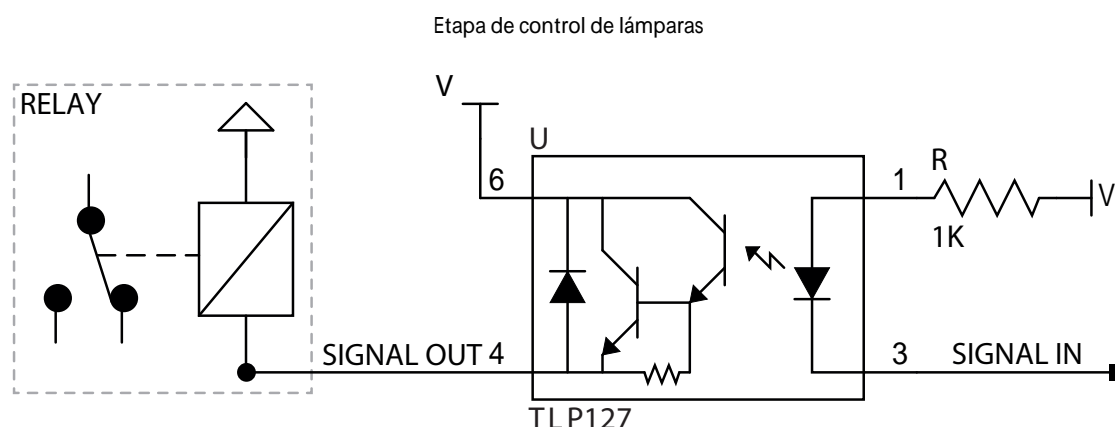


Figura 7.2: Diagrama electrónico para el optotransistor TLP127.

En el cálculo de las resistencias, para la integración de los optoacopladores con el sistema, se consideró la corriente $I_F=10 \text{ mA}$ y el voltaje del LED de 1.3 V .

$$0 = 12[V] - 1,3[V] - I_F \cdot R_F \quad (7.9)$$

$$12[V] = 1,3[V] + I_F \cdot R_F \quad (7.10)$$

$$R_F = \frac{10,7[V]}{10[mA]} = 1070[\Omega] \quad (7.11)$$

Se utilizó una resistencia de valor nominal a 1 kΩ. Para analizar si la corriente que proporciona el optotransistor es necesaria para activar el relevador (Schrack MR201012), se debe considerar el voltaje de la bobina $V_R = 12\text{ V}$ y la resistencia de la bobina $R_R = 110\ \Omega$.

$$I_R = \frac{12[V]}{110[\Omega]} = 110[mA] \tag{7.12}$$

El valor máximo de corriente en el colector es $I_c = 150\text{ mA}$, así que se considera suficiente para garantizar que los relevadores de las lámparas, funcionen correctamente.

7.3.3 Etapa de filtrado

La electrónica en la tarjeta de control y distribución necesita ser inmune al ruido, por lo que se colocó un arreglo de capacitores de *bypass*. La configuración para filtrar el ruido se muestra en la (Figura 7.3).

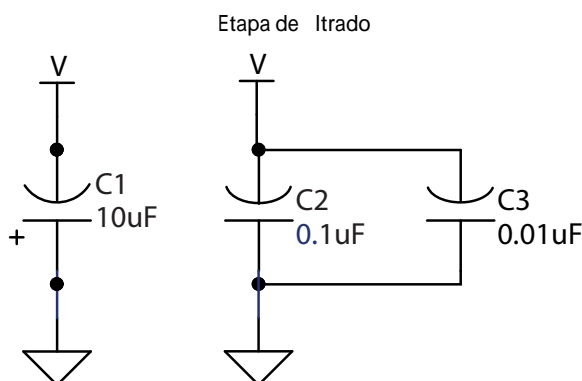


Figura 7.3: Diagrama electrónico para la etapa de filtrado.

7.4 Esquemática de enrutamiento

El esquema de las conexiones del diseño de la tarjeta de control e interconexión se muestra en la (Figura 7.4), y el esquema de las conexiones con la placa Rabbit se encuentra en el Apéndice H.

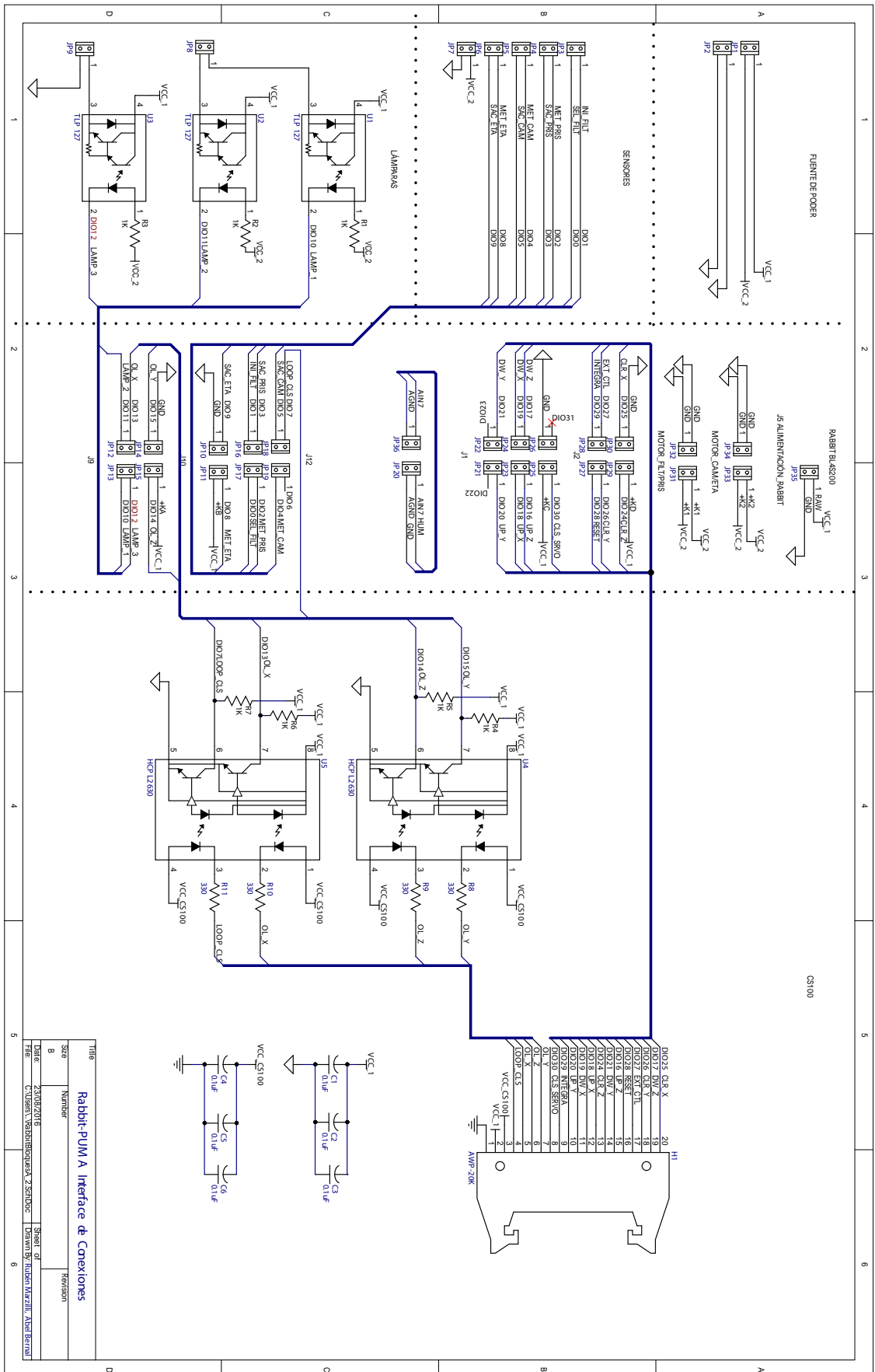


Figura 7.4: Diseño esquemático de las conexiones de la tarjeta de control electrónico.

7.4.1 Preparación para fabricación del PCB

Como el circuito tiene solamente un par de conectores en un pequeño PCB, el enrutamiento se realizó mediante un programa de diseño electrónico, que puede producir archivos utilizables PCB Gerber y un archivo de perforación. Los archivos Gerber son archivos CAM (fabricación asistida por ordenador), que describen dónde poner las pistas de cobre en un PCB. Un archivo de perforación, a veces llamado un archivo de perforación NC (para control numérico), es un archivo de perforación, que describe dónde, qué agujeros y de qué tamaño se colocarán en el PCB. El programa que se utilizó fue *Altium15 Designer* y se mandó a fabricar a *AG Electrónica S.A.*

La documentación de la PCB (archivos NC, Gerber, etc.), se encuentra en el Apéndice F.

7.5 Verificación de enrutamiento

Para la verificación de las señales se consideraron las salidas y entradas, en donde se localizarían los conectores de tornillo (Figura 7.5), a partir de esto se verificó la continuidad de las pistas de la PCB según el esquema de conexiones.

Se distribuyeron 72 señales en la PCB (Cuadro 7.3), las cuales tendrían interfaz con los sistemas del instrumento PUMA.

7.6 Pruebas

Después del grabado se soldaron los componentes de la PCB y, mediante el ensamble final (Figura 7.7), se probó su funcionamiento de las salidas de alta corriente y de los optoacopladores.

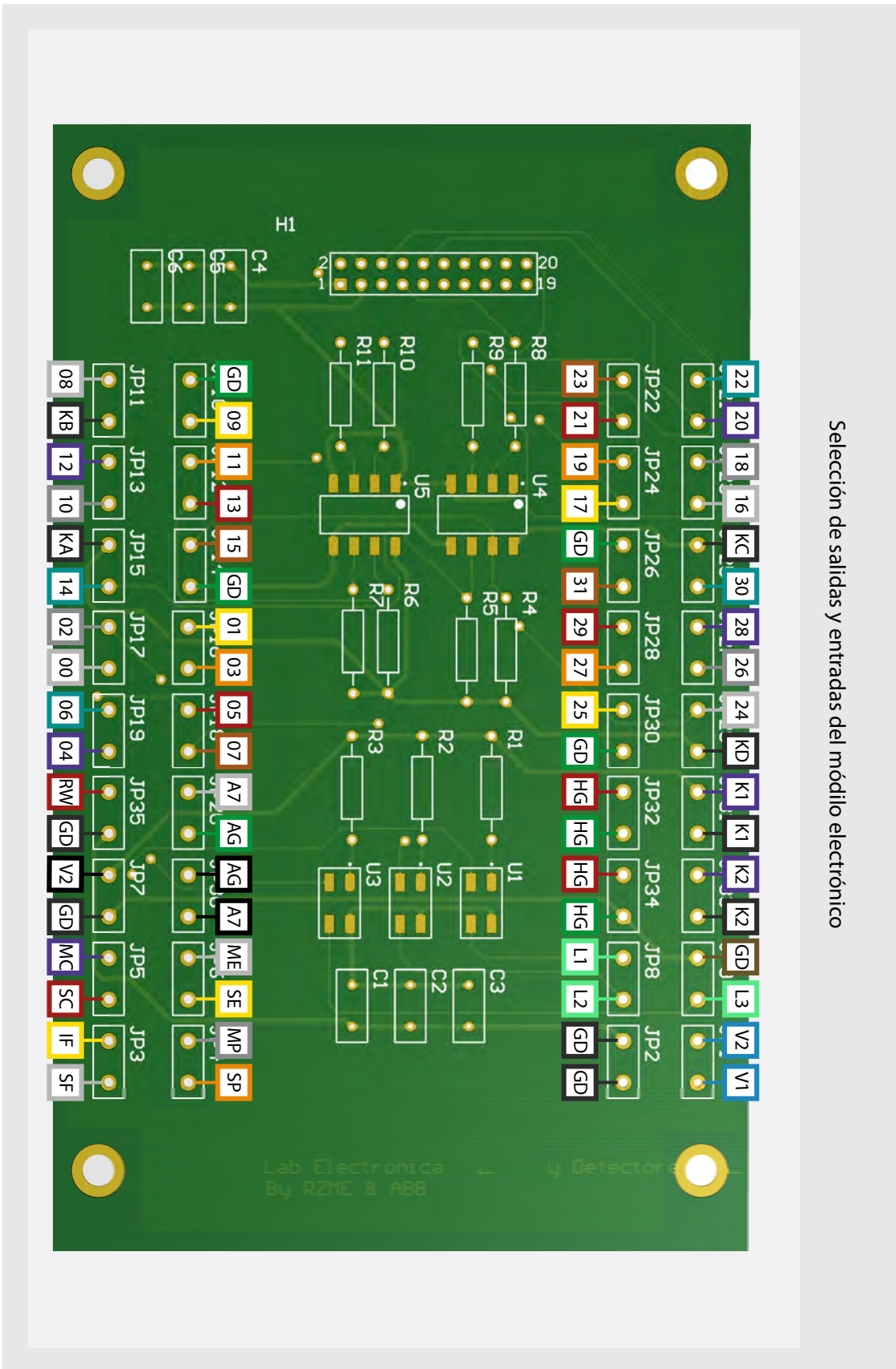


Figura 7.5: Especificación de las conexiones del PCB. El código de colores mostrado en la figura es el mismo al código de colores instalados en el nuevo sistema de control.

Especificación de salidas y entradas del módulo electrónico

| | | | | | | | | | | | |
|----|-------|---------|----|-------|---------|----|-------|----------|----|-------|----------|
| 23 | DIO23 | N.C. | V1 | IN | VCC1 | SP | IN | SAC_PRIS | 08 | DIO08 | MET_ETA |
| 21 | DIO21 | DWN_Y | V2 | IN | VCC2 | MP | IN | MET_PRIS | KB | +KB | VCC_1 |
| 19 | DIO19 | DWN_X | L3 | IN | I LAMP3 | SE | IN | SAC_ETA | 12 | DIO12 | LAMP_3 |
| 17 | DIO17 | DWN_Z | GD | IN | GND LM | ME | IN | MET_ETA | 10 | DIO10 | SAC_ETA |
| GD | GND | GND | K2 | +K2 | VCC_2 | A7 | AIN7 | HUM | KA | +KA | VCC_1 |
| 31 | DIO31 | N.C. | K2 | +K2 | VCC_2 | AG | AGND | GND | 14 | DIO14 | OL_Z |
| 29 | DIO29 | INTEGRA | K1 | +K1 | VCC_2 | AG | AGND | IN | 02 | DIO02 | MET_PRIS |
| 27 | DIO27 | EXT_CTL | K1 | +K1 | VCC_2 | A7 | AIN7 | IN | 00 | DIO00 | SEL_FILT |
| 25 | DIO25 | CLR_X | KD | +KD | VCC_1 | 07 | DIO07 | LP_CLS | 06 | DIO06 | N.C. |
| GD | GND | GND | 24 | DIO24 | CLR_Z | 05 | DIO05 | SAC_CM | 04 | DIO04 | MET_CAM |
| HG | GND | GND | 26 | DIO26 | CLR_Y | 03 | DIO03 | SAC_PRIS | RW | IN | VCC1 |
| HG | GND | GND | 28 | DIO28 | RESET | 01 | DIO01 | INI_FILT | GD | IN | GND |
| HG | GND | GND | 30 | DIO30 | CLR_SR | GD | GND | GND | V2 | IN | VCC_2 |
| HG | GND | GND | KC | +KC | VCC_1 | 15 | DIO15 | OL_Y | GD | IN | GND |
| L1 | IN | I LAMP1 | 16 | DIO16 | UP_Z | 13 | DIO13 | OL_X | MC | IN | MET_CM |
| L2 | IN | I LAMP2 | 18 | DIO18 | UP_X | 11 | DIO11 | LAMP_2 | SC | IN | SAC_CM |
| GD | GND | GND'S | 20 | DIO20 | UP_Y | 09 | DIO09 | SAC_ETA | IF | IN | INI_FILT |
| GD | GND | GND'S | 22 | DIO22 | N.C. | GD | GND | GND | SF | IN | SEL_FILT |

Cuadro 7.3: Descripción de las conexiones del PCB.

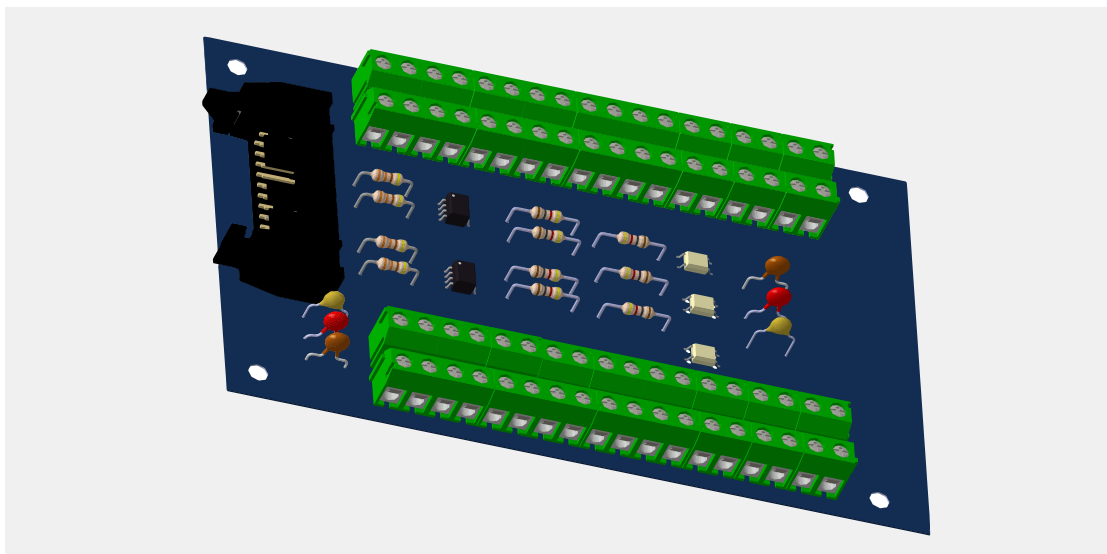


Figura 7.6: Modelo 3D del módulo externo de control electrónico.

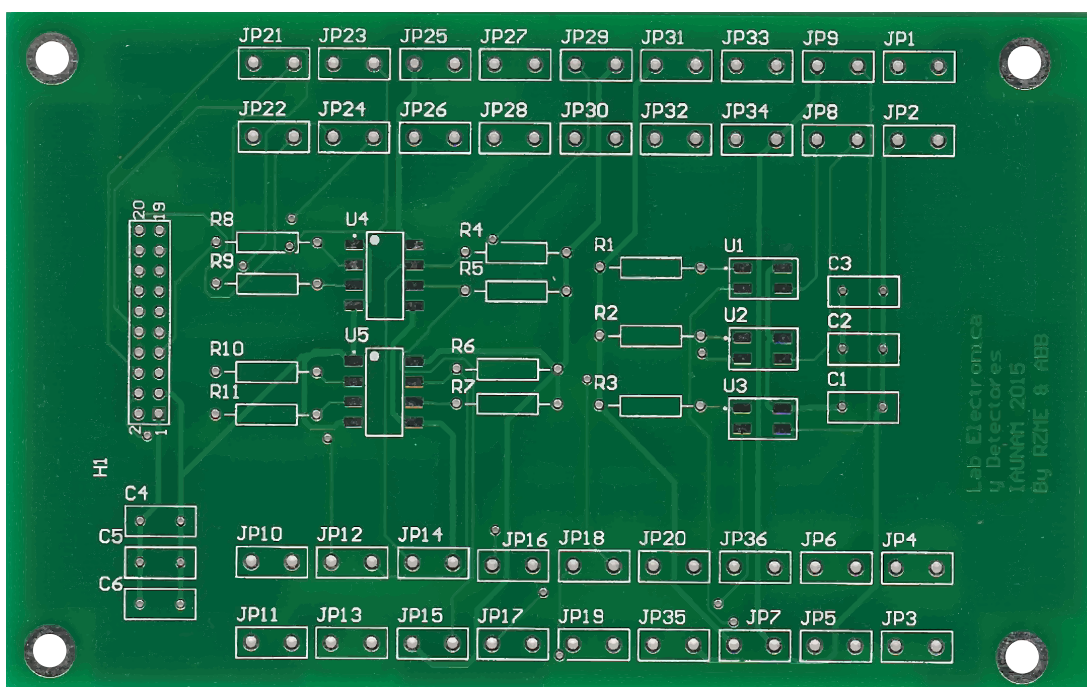


Figura 7.7: Plaqueta de circuito impreso fabricada.

CAPÍTULO

8

Pruebas en el telescopio

Se describe el proceso de instalación del sistema en el instrumento y los pasos para verificar la funcionalidad del instrumento en el telescopio.

Índice

| | |
|--|------------|
| 8.1 Integración | 128 |
| 8.2 Verificación | 130 |
| 8.3 Operación | 132 |
| 8.4 Validación | 133 |
| 8.5 En curso de mantenimiento | 134 |

Este proyecto se desarrolló en el Laboratorio de Electrónica y Detectores, del Instituto de Astronomía ubicado en Ciudad Universitaria, para que fuese instalado en el OAN-SPM localizado en el estado de Baja California, en donde se instaló, probó y se encuentra en funcionamiento el proyecto finalizado.

El sistema de control del instrumento se llevó al telescopio una semana antes del comienzo de la temporada de observación. Allí, junto con el departamento técnico y en colaboración con el personal local del observatorio, nos encargamos de la reintegración. Nuestro trabajo consistió en la re-construcción del sistema de control y llevarlo al mismo nivel de aceptación como lo fue cuando operaba con el microcontralador Octagon, además debió ser instalado y probado en el telescopio. Todo esto tuvo que estar finalizado en un plazo muy estricto de tiempo.

Posteriormente y durante las noches siguientes, el instrumento se puso en marcha, lo que significa que se declara apto para observaciones científicas. En este proceso, el sistema se ajustó para alcanzar sus resultados óptimos en las extremas condiciones existentes en el telescopio. Este tipo de pruebas es de mucha utilidad para los ingenieros y astrónomos del observatorio.

8.1 Integración

Una vez se obtuvieron los componentes, se prosiguió a integrar el sistema para el ensamble final y su integración con el instrumento. En esta etapa para los componentes críticos, en donde sus propiedades y comportamientos deben estar dentro de las especificaciones y en donde se llevaron a cabo las pruebas de aceptación individuales. En este caso se estudió la precisión del movimiento y reproducibilidad de los actuadores que controla la placa Rabbit, cuyas mediciones se realizaron en condiciones operativas finales similares, con la carga adecuada, en una variedad de pruebas y en instalaciones totalmente equipadas.

La integración virtual del sistema es un paso principal en el que todos los componentes y subsistemas se pueden montar juntos para hacer al instrumento funcionar. En general se tuvo la mayor responsabilidad en esta etapa. El sistema de control (Figura 8.1), se ajustó para alcanzar buenos niveles de precisión en los movimientos de los motores de los carros móviles y también consistió en hacer la alineación de las placas del Fabry-Pérot, que siempre es un paso crítico en la fase previa a la observación astronómica. El instrumento PUMA contiene varios componentes ópticos que deben estar alineados con precisión de acuerdo con las especificaciones del diseño. Los criterios de calidad de alineación están en las aberraciones del frente de onda y estabilidad de imagen, corrigiéndose con el mismo interferómetro y el detector CCD con largas iteraciones, es decir, con el instrumento funcionando, a través del software de control del sistema y a través de la electrónica de la unidad de control.

Todos los controladores a nivel de placa requieren un montaje en algún tipo de protección física. La placa BL4S200 (el controlador embebido) y el módulo externo electrónico se distribuyeron sobre un gabinete marca Altech AL2023 – 11, de material de aluminio, con dimensiones de 202 x 230 mm; en este caso no se necesita un sistema de refrigeración, ya que las temperaturas de operación son cercanas a 0°C. Las medidas de la caja se encuentran en el Apéndice G.

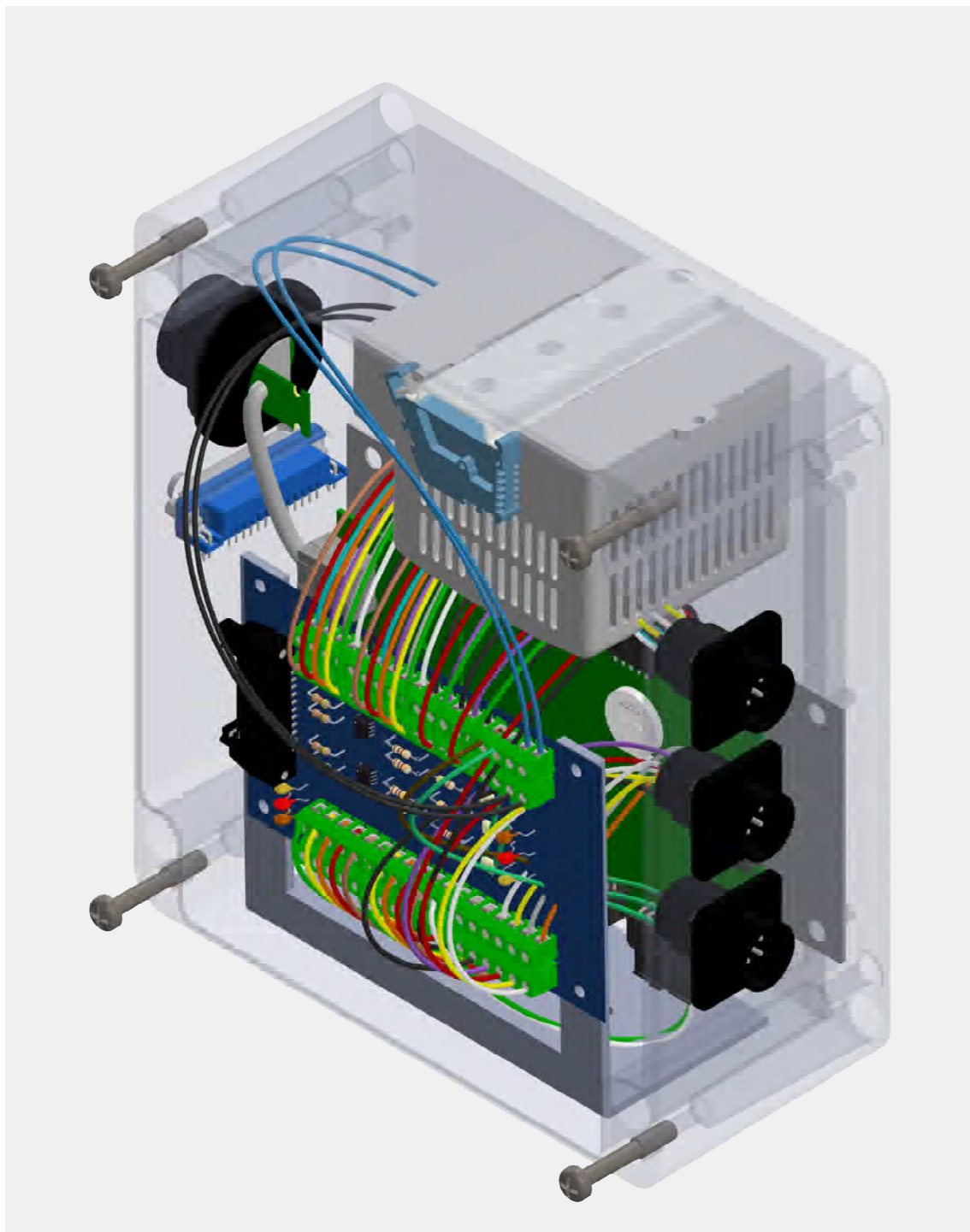


Figura 8.1: Modelo 3D para la integración del nuevo sistema de control. *Modelo 3D realizado por Zimri Marsilli.*

La placa BL4S200 se montó sobre separadores que facilitan su manejo durante el desarrollo, además de proteger la parte inferior de la placa de circuito impreso contra los arañazos o cortocircuitos mientras se está trabajando con el microcontrolador.

Las conexiones de la placa BL4S200 de control al módulo electrónico requieren una cuidadosa atención. Se ocuparon terminales de tornillo enchufables, una opción popular cuando se tiene

un número elevado de terminales, en los que se conectan cables por separado a la placa Rabbit. Para un montaje final mucho más confiable se probaron las salidas de la Rabbit antes de enchufarse en la PCB montada. La utilización de terminales de tornillo fijas como parte del montaje final de la PCB, fueron para evitar ramificaciones de cables a partir de los individuales y evitar que la tarea de conexión pudiera ser propensa de errores. Además, la prueba de los cables es bastante difícil con un manejo de cables sueltos volando alrededor. Esto implica que los cables que se probaron por primera vez se encuentran en el sistema completo. Los técnicos de servicio, abrumadoramente prefieren terminales roscados conectables a terminales fijas con tornillo. Cuando hay que cambiar un par de cables, por ejemplo para reemplazar un sensor, ambos tipos de terminales de tornillo son igualmente adecuados. Sin embargo, si la tarjeta de conexión debe ser cambiada, el tiempo requerido para retirarla y volverla a instalarla con la gran cantidad de cables individuales, puede resultar más laboriosa que la simple eliminación de una terminal de tornillo enchufable. Completadas las conexiones de los cables el sistema puede ser probado en su totalidad fácilmente.

El sistema de control (Figura 8.2), se compondrá de la placa Rabbit conectada a la fuente de voltaje y al módulo electrónico. El módulo se conectará a los tres conectores cpc que realizan la interfaz de conexión con los motores. El conector db25 que conecta con el CS100 y el conector Ethernet para el establecimiento de la comunicación por red. Se integrará a la placa estructural de componentes electrónicos del instrumento PUMA. Una vez montado al PUMA se prosigue con las conexiones a los sensores y actuadores.

8.2 Verificación

Después del ensamble e integración sigue la verificación, en la que se optó por hacer pruebas de funcionamiento de todos los subsistemas producidos, del sistema de control y de los conectados al instrumento PUMA, ya que el instrumento es complejo y requiere la interacción de los diversos subsistemas. Un programa especial de prueba funcional (código de prueba) fue preferible en la plena aplicación, para verificar que los diferentes componentes de hardware estuvieran en funcionamiento (Figura 8.3).

En la prueba de los subsistemas se consideraron módulos individuales, los cuales se combinaron para formar unidades cada vez más grandes que se probaron en masa, es decir, se probó el sistema y todos los componentes se combinaron para probarse en conjunto.

El interferómetro debe estar conectado en la unidad del panel trasero del controlador CS100. Los controles del panel frontal del CS100 se clasifican en tres categorías, los controles para el balance del puente de capacitancia, los controles de ganancia de lazo de control, de constante de tiempo y, por último, los interruptores y conexiones de alimentación. En los controles para el balance del puente de capacitancia se encuentran los controles de los ejes x , y y z para su respectivo paralelismo en lazo cerrado, o calibrar los tres ejes en lazo abierto teniendo un paralelismo no muy exacto. Los incrementos gruesos y exactos tienen una resolución de $1 \mu\text{m}$ y las calibraciones para los desplazamientos finos tienen una resolución de 0.5 nm . Si se establecen parámetros iniciales con la calibración gruesa, los valores de los incrementos finos tendrán superposición sobre los valores gruesos establecidos. El espaciado definido por el canal z definirá el espacio medio entre las placas del FP en modo de lazo cerrado o equilibrará el puente de capacitancia en modo

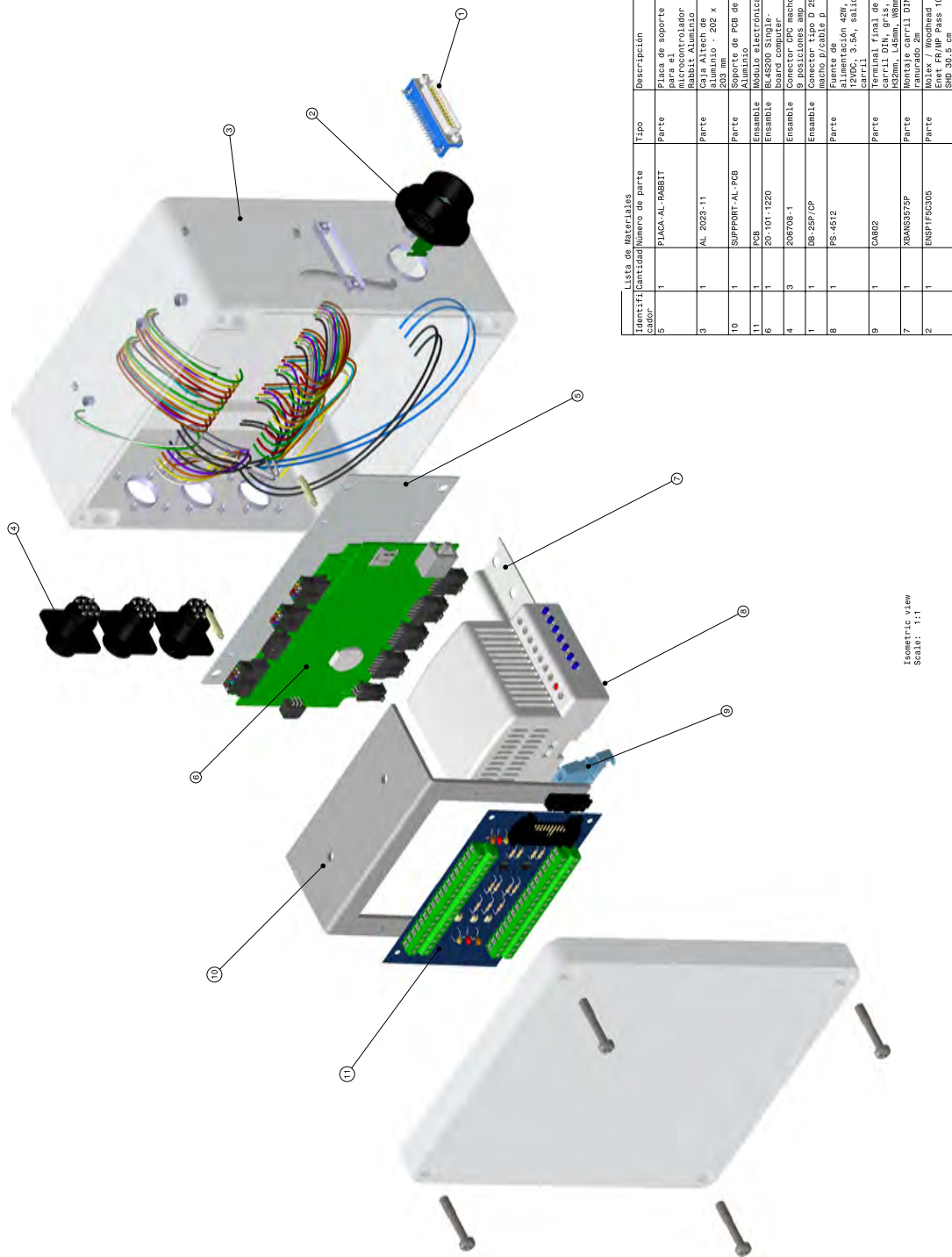


Figura 8.2: Desarme del nuevo sistema de control.

de lazo abierto, la resolución para el ajuste del espaciamiento no fino tiene una variación de $\pm 1 \mu\text{m}$ y con control fino es de 0.5 nm .

La alineación de las placas del interferómetro de Fabry-Pérot se realizó mediante la observación del patrón de anillos de interferencia, que se produjeron mediante una fuente difusa vista a través de este. Como el FP se usa en la región visible, los anillos se pudieron producir a partir de una de las líneas de emisión de una lámpara fluorescente como fuente. Mediante el ojo del observador y con el control de paralelización del controlador a lo largo de los ejes x e y , se pudo ajustar y minimizar la expansión y contracción del patrón de anillos de interferencia que se genera al mover el ojo. El control del eje z se utilizó para hacer aparecer un anillo de interferencia en el centro del campo

Una vez que el interferómetro estuvo correctamente conectado y el interruptor de protección del panel trasero estuvo encendido, se pudo encender el controlador CS100. En este caso se conocían los parámetros del CS100, los valores de paralelismo, del puente de balance, la ganancia y la constante de tiempo, estos se establecieron en el controlador CS100. Se desactivó el interruptor de lazo cerrado y el de la función integrar y finalmente se activó la alimentación. Se conmutó a activo el interruptor de lazo cerrado, moviendo las agujas de los medidores a una diferencia de $\pm 0.5 \text{ V}$ alrededor de 0 V , sin encender los indicadores de sobrecarga, en cambio, después de un retraso de aproximadamente 1 s el indicador de lazo cerrado encendió, indicando su buen funcionamiento. Después se conmutó el interruptor de la componente resistiva y se ajustó a cero las agujas de los medidores con los controles de equilibrio relativo. De esta manera el FP estuvo correctamente paralelizado, siendo comprobado mediante el sistema de calibración del instrumento PUMA posteriormente.

El sistema de control del instrumento PUMA hace uso del bus de control externo del CS100, operando todos los controles del panel frontal del controlador (sin considerar el de la fuente de poder), de forma remota. Las entradas usan lógica negativa compatibles con tecnología TTL (al igual que las salidas), a 5 V y con resistencia de $4 \text{ k}\Omega$.

8.3 Operación

Durante la operación se realizaron pruebas de aceptación, donde se permitió al usuario usar sus propios casos de prueba para verificar que el sistema estuviera trabajando de manera correcta. En la preparación del telescopio para el PUMA, una vez alineadas todas las funcionalidades del sistema, estuvo listo para ser sintonizado y alcanzar excelentes beneficios. Esta etapa no fue de nuestra responsabilidad a pesar de que dimos nuestro apoyo para resolver los posibles problemas que vienen siempre al comenzar a utilizar este tipo de sistemas complejos. Esta etapa duró la noche de observación próxima a la instalación del sistema de control, terminando con la llamada aceptación preliminar. Cuando el instrumento se declaró compatible, estuvo autorizado para ser operado en el telescopio (Figura 8.9).

Se observó un objeto (Figura 8.4) en el espacio con el instrumento PUMA y se tomaron diferentes canales de calibración (Figura 8.5) e interferogramas (Figura 8.6) .

Integración del sistema de control al instrumento PUMA

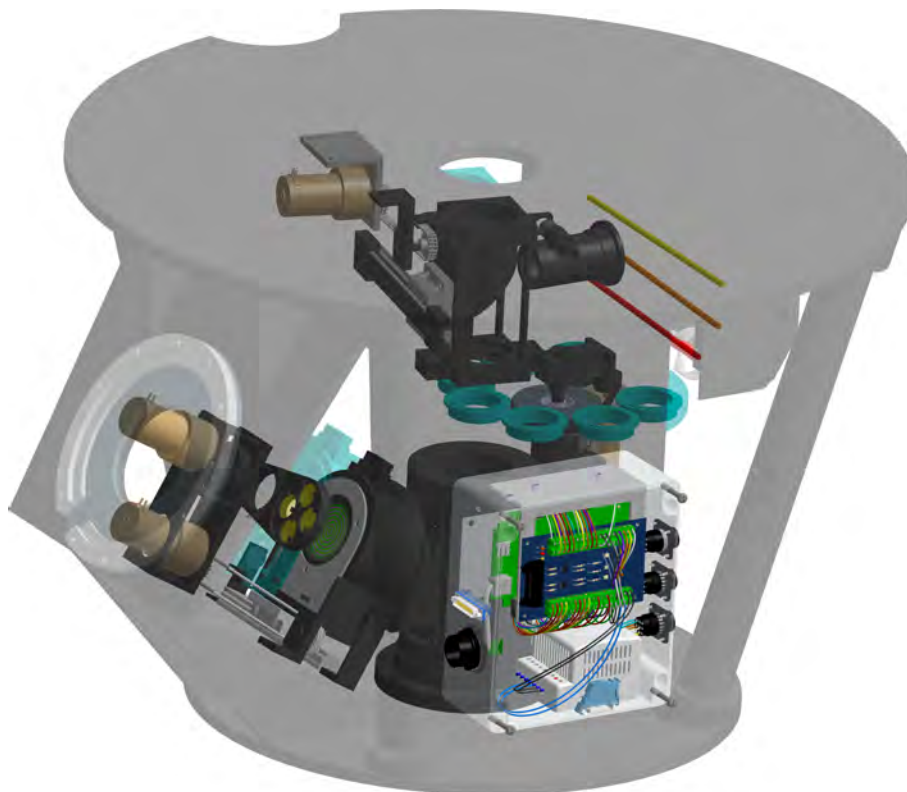


Figura 8.3: Nuevo sistema de control montado en el instrumento PUMA. *Modelo 3D realizado por Zimri Marsilli.*

8.4 Validación

Para la validación del instrumento se realizaron pruebas funcionales. Esta es la forma más común de prueba y su propósito es asegurar que el sistema realice sus funciones normales de manera correcta. Así los casos de prueba se desarrollan y alimentan las salidas del sistema. Las observaciones astronómicas realizadas con el sistema deben ser consistentes y compatibles entre sí, es por ello que se han realizado las siguientes observaciones para validar la operación completa del instrumento PUMA, Cuadro 8.1.

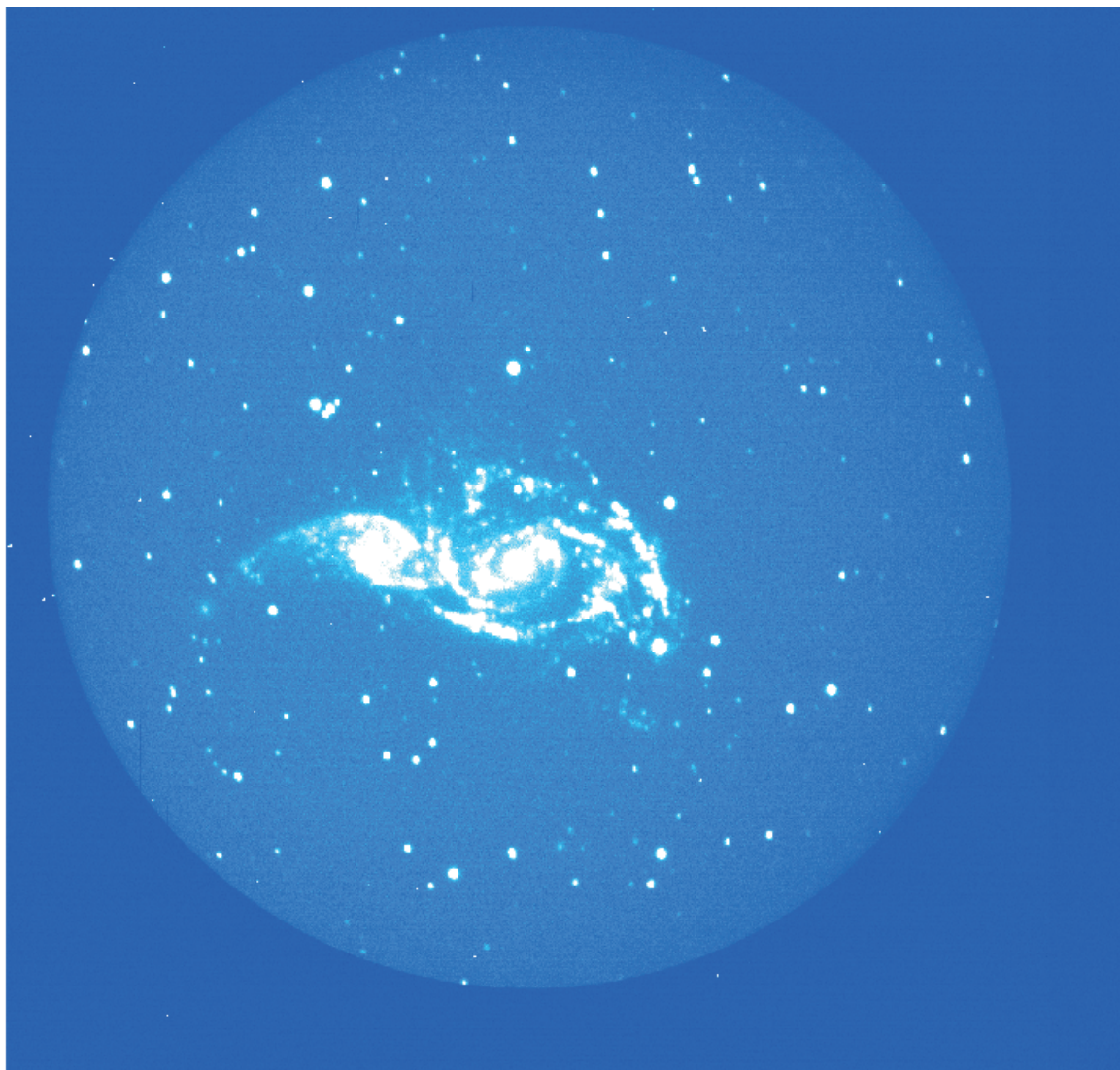


Figura 8.4: NGC 2207 y IC 2163, par de galaxias espirales que se encuentran a 80 millones de años luz de distancia. Imagen directa tomada por el instrumento PUMA con el nuevo sistema de control ya instalado. *Imagen tomada por el Fís. Abel Bernal.*

8.5 En curso de mantenimiento

Se realizó una lista de costos y de materiales para el mantenimiento y seguimiento del sistema de control (Cuadro 8.2).

Durante el transcurso de mantenimiento se realizarán las pruebas de desempeño, cuyo propósito de este tipo de prueba es asegurar que el sistema pueda manejar las entradas y salidas especificadas en el modelo de análisis, además de asegurar que tenga el tiempo de respuesta requerido.

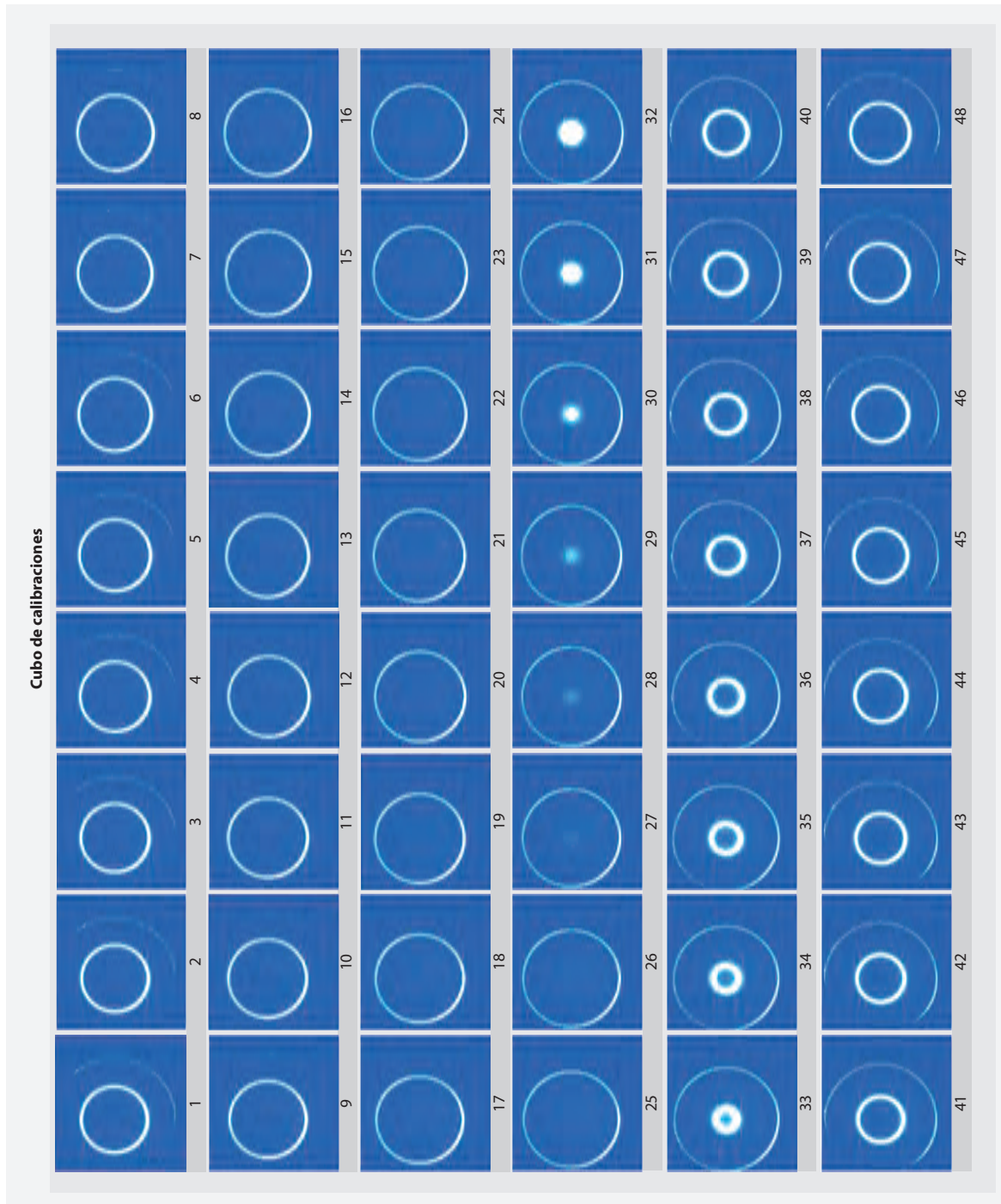


Figura 8.5: Archivos de cie de calibración.

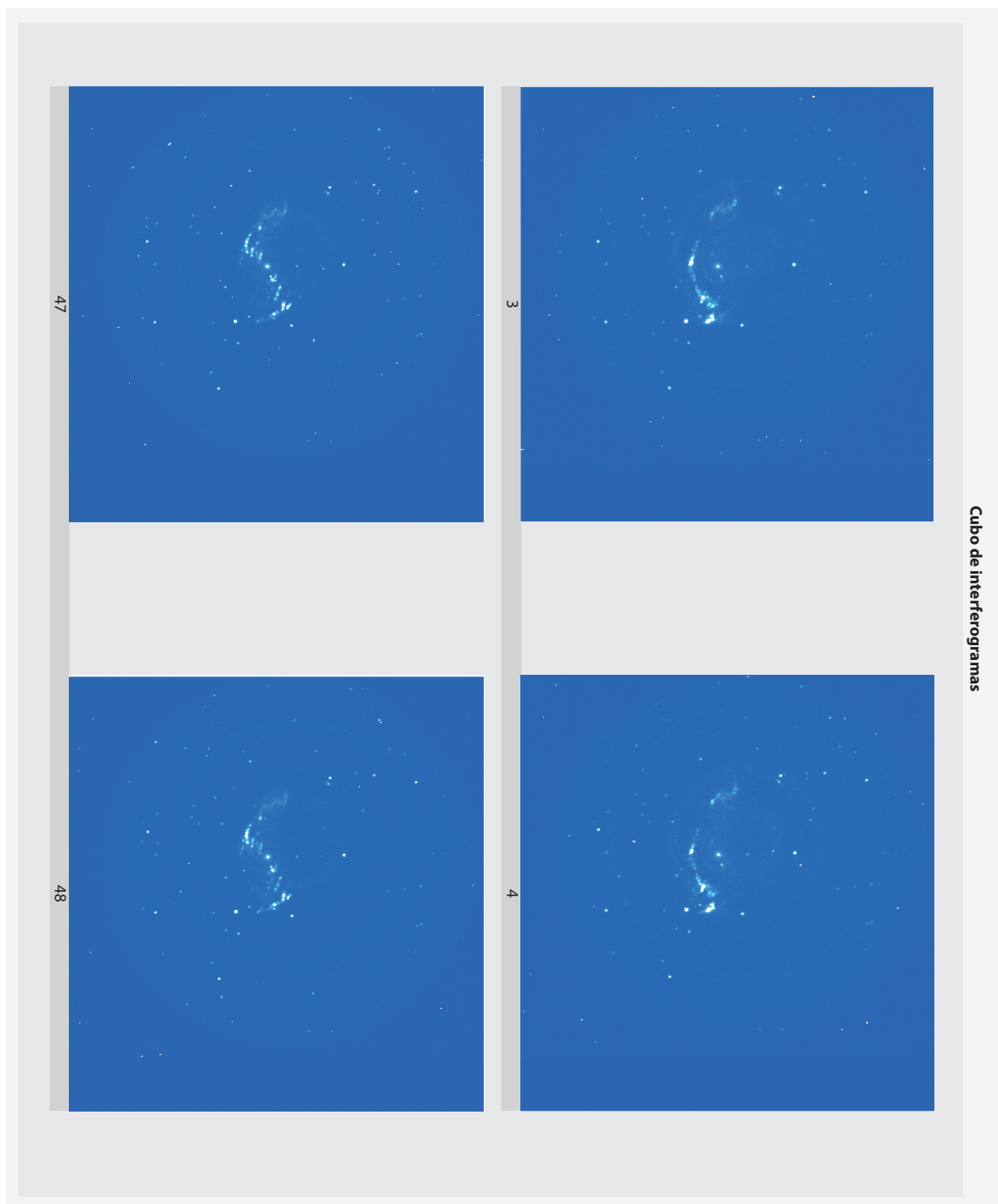


Figura 8.6: Archivos de cío de interferogramas.

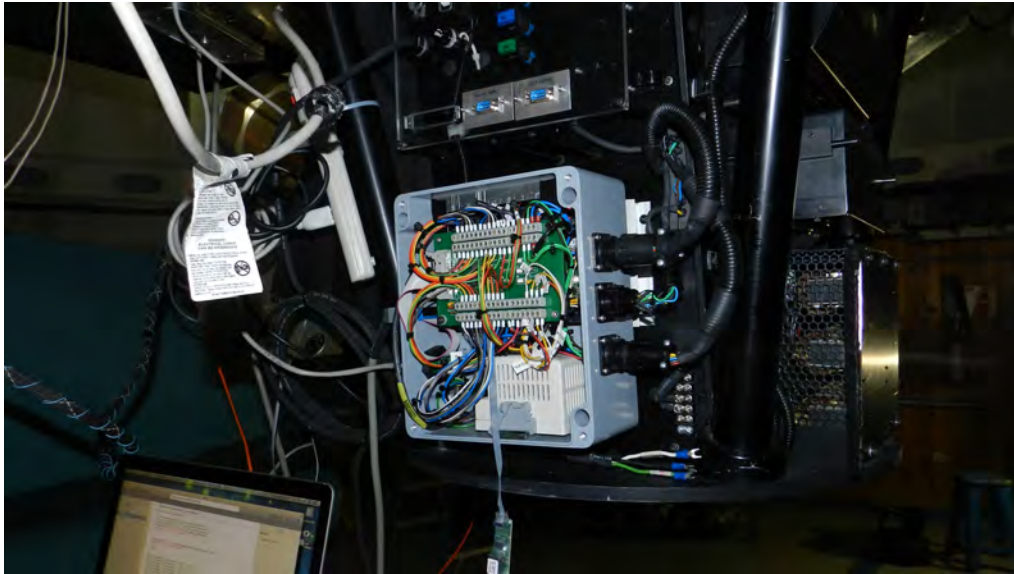


Figura 8.7: Sistema de control durante la etapa de pruebas finales con el instrumento PUMA. *Fotografía tomada por Zimri Marsilli.*

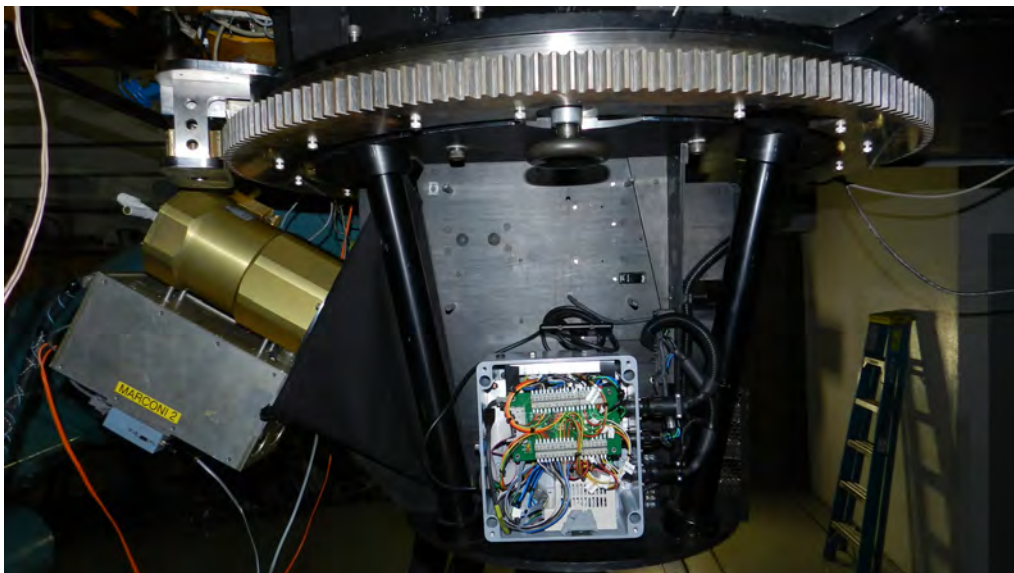


Figura 8.8: Sistema de control instalado en el instrumento PUMA. A la izquierda se muestra el crióstato de la cámara CCD. *Fotografía tomada por Zimri Marsilli.*

Cuadro 8.1: Observaciones realizadas con el instrumento PUMA en el telescopio de 2.1 m. Se observa que se contaron con seis noches de ingeniería y han sido hasta ahora 27 noches de observación con el nuevo sistema de control electrónico ya instalado. En la temporada del mes de Octubre se instaló el sistema. (Departamento de Astronomía Observacional, 2016)

| Tipo de noche | Año | Mes | Número de Noches |
|---------------|------|------------|------------------|
| Ingeniería | 2015 | Octubre | 6 |
| Observacional | 2015 | Diciembre | 5 |
| Observacional | 2016 | Enero | 7 |
| - | 2016 | Febrero | - |
| Observacional | 2016 | Marzo | 6 |
| - | 2016 | Abril | - |
| - | 2016 | Mayo | - |
| Observacional | 2016 | Junio | 9 |
| - | 2016 | Julio | - |
| - | 2016 | Agosto | - |
| - | 2016 | Septiembre | - |

Lista de costos de los materiales del nuevo sistema de control PUMA

| Lista de materiales: ensamble principal | | | | | | | | | |
|---|--------------------------------|----------|---|-----------|-----------------|-----------------|-------|--|--|
| Cantidad | Número de parte | Tipo | Nomenclatura | Revisión | Precio Unitario | Precio total | | | |
| 1.00 | SPL / IA - PM - CE - 120 / 002 | Ensamble | Caja Altech de aluminio - 202 x 203 mm | Comercial | 2752.000 | 2752.000 | ** | | |
| 1.00 | SPL / IA - PM - CE - 120 / 003 | Parte | Placa de soporte para el microcontrolador Rabbit Aluminio | Fabricado | 250.000 | 250.000 | | | |
| 1.00 | Soporte de PCB de Aluminio | Parte | Soporte de PCB de Aluminio | Fabricado | 250.000 | 250.000 | * | | |
| 2.00 | 20-101-1220 | Ensamble | BL45200 Single-board computer | Comercial | 5576.724 | 11153.448 | | | |
| 3.00 | 206705-1 | Ensamble | Conector CPC hembra 9 posiciones, amp | Comercial | 435.000 | 1305.000 | *** | | |
| 3.00 | 206708-1 | Ensamble | Conector CPC macho 9 posiciones amp | Comercial | 442.280 | 442.280 | **** | | |
| 1.00 | PS-4512 | Parte | Fuente de alimentación 42W, 12VDC, 3.5A, Marca Altech | Comercial | 1419.000 | 1419.000 | ***** | | |
| 1.00 | CA802 | Parte | Terminal final de carril DIN, gris, H32mm, L45 mm, W8mm | Comercial | 12.090 | 12.090 | ** | | |
| 1.00 | XBANS3575P | Parte | Montaje carril DIN ranurado 2m | Comercial | 357.040 | 357.040 | ***** | | |
| 1.00 | ENSP1F5C305 | Parte | Molex / Woodhead Enet FR/MP Pass 10T SHD 30.5cm | Comercial | 1078.000 | 1078.000 | ***** | | |
| 1.00 | PCB | Parte | Módulo electrónico | Fabricado | 2083.446 | 2083.446 | | | |
| 1.00 | DB-25P/CP | Parte | Conector tipo D 25P macho p/cable p | Comercial | 12.069 | 12.069 | * | | |
| 1.00 | DB-25S | Parte | Conector Tipo D 25 pins hembra | Comercial | 11.207 | 11.207 | * | | |
| 1.00 | 17250 10 B1 | Parte | Cable de alimentación | Comercial | 110.97 | 110.97 | ***** | | |
| N.D. | M6, 10-24, etc | Parte | Tornillería tipo Allen inoxidable | Comercial | N.D. | 100.00 | * | | |
| Total | | | | | | 21336.55 | | | |

| Lista de materiales: PCB | | | | | | | | | |
|--------------------------|-----------------|-------|--|-----------|-----------------|-----------------|---|--|--|
| Cantidad | Número de parte | Tipo | Nomenclatura | Revisión | Precio Unitario | Precio total | | | |
| 10.00 | Board | Parte | Circuito impreso | Fabricado | 180.73 | 1807.28 | * | | |
| 36.00 | TRT-02 | Parte | Bornera p/circuito impreso 2 termin | Comercial | 3.349 | 120.564 | * | | |
| 7.00 | RC-1K/1/2 | Parte | Resistencia de carbón 1k Ohm 5% 1/2W | Comercial | 0.862 | 6.034 | * | | |
| 4.00 | R330 | Parte | Resistencia de carbón 330 Ohms 5% 1/2W | Comercial | 0.862 | 3.448 | * | | |
| 1.00 | RPS-20RLM | Parte | Conector macho en ángulo recto para PCB 20 pines | Comercial | 20.000 | 20.000 | * | | |
| 1.00 | FCN20 | Parte | Conector 20 pin para cable plano | Comercial | 4.741 | 4.741 | * | | |
| 2.00 | CC-104/500V | Parte | Capacitor cerámico 1uF | Comercial | 2.500 | 5.000 | * | | |
| 2.00 | CC-1/50V-WELSON | Parte | Capacitor cerámico .1uF | Comercial | 1.724 | 3.448 | * | | |
| 2.00 | CC-010/100V | Parte | Capacitor cerámico .010uF= 10nF | Comercial | 0.862 | 1.724 | * | | |
| 3.00 | TLP127 | Parte | Optoacoplador Fototransistor 2500V | Comercial | 14.655 | 43.965 | * | | |
| 2.00 | HCPL26315V | Parte | Optoacoplador de compuerta lógica de doble canal | Comercial | 33.621 | 67.242 | * | | |
| Total | | | | | | 2083.446 | | | |

*Precios de AG electrónica: www.agelectronica.com

**Precio de www.altechcorp.com

*** Precio de www.sucabsa.net/#/conectores/c1wul

****Precio de www.mouser.mx/ProductDetail/Altech/PS-4512/

*****Precio <https://mexico.newark.com/>

lunes, 8 de agosto de 2016 11:53:50

Cuadro 8.2: Costos de componentes del nuevo sistema de control. Cuadro realizado por Zimri Marsilli.



Figura 8.9: El sistema de control instalado en el instrumento PUMA que a su vez está montado en el telescopio de 2.1m. *Fotografía tomada por Zimri Marsilli.*

CAPÍTULO

9

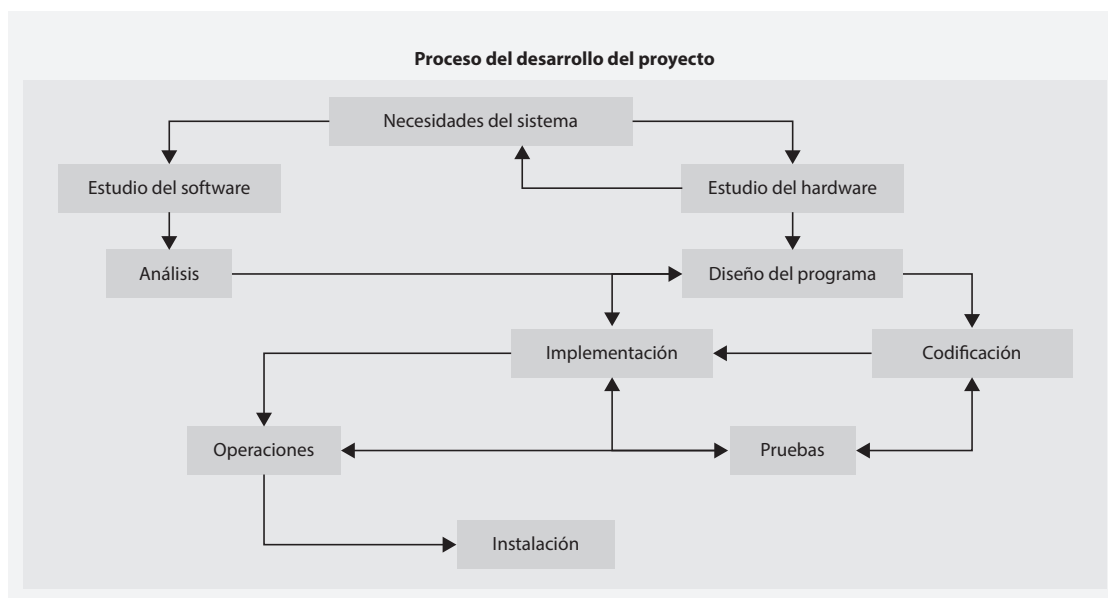
Discusión y conclusión

Este último capítulo concluye el proyecto y discute los resultados de esta tesis.

Índice

| | |
|---|------------|
| 9.1 Interpretación de los resultados | 142 |
| 9.2 Problemas encontrados | 145 |
| 9.3 Lecciones aprendidas | 146 |
| 9.4 Mejora para un trabajo futuro | 146 |

9.1 Interpretación de los resultados



Cuadro 9.1: Metodología ideal en que se basó el desarrollo del proyecto. Referencia (Yourdon, 1993).

El reto fue encontrar una aplicación completamente nueva, con el uso de una nueva tecnología de microcontroladores (diferente con los que usualmente se desarrolla). En este caso el procesador embebido altamente integrado Rabbit, se desempeña como una tecnología notable. Su compatibilidad con el protocolo de comunicación Ethernet, es también muy útil. Los compiladores y depuradores en lenguaje C, y una pila con todas las funciones TCP/IP son las herramientas adecuadas para trabajar y desarrollar instrumentación astronómica.

Pero como en todo desarrollo de sistemas, las herramientas utilizadas para modelar el proyecto son tan importantes como la tecnología que se utiliza.

En la metodología utilizada para el desarrollo del sistema (Cuadro 9.1), el análisis y diseño fueron los principales modelos de estudio, detallados gráficamente para ser simples y fáciles de entender, representando de manera precisa los objetivos del proyecto; requeridos para un sistema nuevo con funciones existentes que debieron reimplantarse. También se necesitó añadir nuevas funciones y criterios de desempeño para el nuevo sistema, siendo el principal la factibilidad, es decir, se solicitó que algunas partes del sistema se automatizaran y controlaran de manera más eficiente y que la implantación del sistema fuera de alta calidad y libre de errores. Esto dependió en gran medida de la naturaleza del diseño, la capacidad de la programación y de su mantenimiento para realizar cambios en el sistema después de haberlo puesto en operación.

El proceso de análisis de las actividades del sistema implicó el desarrollo de un entorno ambiental y el estudio de su comportamiento, representando una descripción formal de lo que el sistema debía de hacer para cubrir los objetivos. Aunque se complementó el enfoque conservador con uno radical, se crearon documentos de manera formal (un conjunto de “modelos de papel”)

al final del proyecto, que deberán permanecer siempre con el sistema a lo largo de su operación y mantenimiento.

Estos modelos integraron métodos, técnicas y herramientas de modelado gráfico para el análisis estructurado durante el desarrollo, diseño, organización, programación, prueba e implantación del sistema. Estos gráficos, particionados y mínimamente redundantes, son sujetos a cambios y extensiones. Como herramientas de modelado y simulación de bajo costo se utilizaron para estudiar este sistema complejo. Cada modelo se enfocó en un número limitado de aspectos del sistema (según las funciones que debía realizar), para poder destacar adecuadamente características importantes, modelar los objetivos y predecir el comportamiento del sistema.

El nuevo programa de control se hizo mediante la programación estructurada (técnica moderna para el desarrollo de sistemas con el enfoque del diseño formal de sistemas). Para generar un conjunto de módulos que ejecutarán los requisitos del usuario. De una especificación clásica se transformó en un diagrama de flujo de datos, en un diccionario de datos y en una especificación de procesos, siempre considerando la retroalimentación entre una herramienta y otra. Este sistema totalmente automatizado que se desarrolló en este caso es de remplazo de uno existente, por lo que se describieron los asuntos relacionados con la codificación y la integración de módulos creciendo progresivamente hasta un sistema final más completo.

El campo de actividad inicial del sistema para el proyecto propuesto implicó el desarrollo de un diagrama de contexto, cuyo principal propósito fue definir cuanto abarcaba el sistema, así como los diversos sistemas con los que interactuaba. Posteriormente se identificaron las deficiencias actuales en el sistema, comprendiendo una nueva lista de funciones, que complementarían esas faltas y que se estuvieran llevando a cabo insatisfactoriamente en el sistema actual. Adicionalmente se estudió el comportamiento dependiente del tiempo del sistema, para mostrar interrupciones, señales, la sincronización y coordinación de distintas labores de proceso (flujos de control, procesos de control y el diagrama de transición de estados). Para este sistema de tiempo real la tecnología fue la adecuada; además el diseño y la programación lograron un nivel de eficiencia aceptable.

La herramienta de desarrollo Dynamic C se ha utilizado a lo largo del proyecto para desarrollar el código de control para la placa Rabbit; sin embargo también se ha utilizado Visual C++ en Windows para comunicarse a través de una PC a la placa de control Rabbit. También se utilizó programación C++ en ambiente Linux (GNU-GCC) durante las pruebas del sistema. Con todas las opciones para el desarrollo con la mono placa reducida Rabbit y con el uso del conocimiento de redes y programación de micro-controladores, se implementó la actualización del sistema de control del espectrógrafo integral de campo PUMA.

Se generaron y comenzaron a producir las actividades como un conjunto de casos de prueba de aceptación. El desarrollo de las pruebas de aceptación sucedieron al mismo tiempo que las actividades de diseño e implantación. Se modeló la implantación actual del sistema con suposiciones sobre las entradas y salidas que son causadas e iniciadas por los terminadores del sistema, ya que la tecnología que representan estos terminadores no se puede modificar para mejorar su confiabilidad.

En la instalación del nuevo sistema, se procedió con la preparación de la sede del controlador (planeado con meses de anticipación). Esto implicó construir un complemento adicional de soporte para la instalación del hardware para conectar la computadora. La instalación del software, involucró cargar el programa de control que se escribió para el nuevo sistema en la placa BL4S200 y prepararla para su operación. Existiendo una sola instalación en una sola sede (el observatorio), fue necesario instalar el sistema por etapas, siendo el cambio al nuevo sistema casi inmediato durante un periodo de cinco días.

La prueba final o prueba de aceptación se realizó con el sistema integrado, producido en la implantación y cuyo propósito de la actividad fue verificar que el sistema tuviera un nivel apropiado de calidad. La validación de red fue con mnemónicos (respuestas TCP /IP), que el usuario podría recordar fácilmente.

En conclusión se siguió un proceso de programación, pruebas, conversión e instalación, como lo especifica el Cuadro 9.1. Sin embargo en este caso, también se utilizó un enfoque radical, ya que se requirió un plan de proyecto con un traslape casi inmediato del análisis, el diseño, la programación y las pruebas.

La elaboración del sistema final es el resultado de una alta calidad, teniendo características de productividad, confiabilidad, mantenibilidad y de oportunidad, con precisión y con suficiente detalle, de lo que el instrumento debe hacer. La eficiencia se traduce a los tiempos de respuesta adecuados para los carros móviles, su potencia y rapidez.

La seguridad es un asunto de importancia para este proyecto (pues estará en red), debido a que se actualizó un bien muy delicado de la comunidad universitaria, por lo que, al igual que el precedente, evita el acceso no autorizado, además de la actualización y la eliminación no autorizada de datos delicados.

También es confiable respecto a un tiempo promedio entre fallas y la disponibilidad del sistema (porcentaje de tiempo en el que está disponible). Es transportable al cambio fácil entre un programa y otro en el modelo de implantación. Tiene cohesión entre los módulos, en los cuales cada instrucción es necesaria para poder llevar a cabo una sola tarea bien definida.

El programa de control también tiene acoplamiento, pues los módulos tienen interfaces sencillas, limpias y que comparten un número mínimo de datos entre ellos, sin modificar la lógica interna o los datos de alguno otro. El tamaño de cada módulo se trató de programar lo suficientemente pequeño, y cuyo alcance de control tiene un módulo administrador que llama a módulos subordinados. Las restricciones políticas y operacionales, se concentran en la configuración específica del hardware en condiciones de trabajo bajo condiciones ambientales de temperatura y humedad.

La especificación del proceso no se puede considerar congelada después de concluida la fase de análisis. El usuario puede decidir si quiere añadir una nueva funcionalidad al sistema actual, reimplantando las aplicaciones existentes de una manera algo más eficiente.

Los datos obtenidos del análisis teórico, de torques requeridos, concuerdan con los datos experimentales, en este caso los motores no necesitan producir demasiado torque debido a la naturaleza mecánica del instrumento PUMA y a la caja de reducción de los motores instalados. El que requiere un mayor torque es el brazo de prismas con un torque calculado de $2,964 \cdot 10^{-2} \text{ N}\cdot\text{m}$, seguido del posicionador de elementos ópticos con $2,778 \cdot 10^{-2} \text{ N}\cdot\text{m}$, la unidad de calibración móvil $1,597 \cdot 10^{-2} \text{ N}\cdot\text{m}$ y la rueda de filtros $4,314 \cdot 10^{-3} \text{ N}\cdot\text{m}$. Los datos de inercia de carga se calcularon para realizar perfiles de velocidad y tener un movimiento de los motores más caracterizado en un trabajo a futuro.

La gráfica 6.12 hace referencia a lo que se buscó para este tipo de pruebas: el cálculo del torque mediante las variables de control (la frecuencia y el ciclo de trabajo), modificables en el programa cargado a la placa Rabbit. En esta gráfica se muestra que a partir de los 10 kHz las curvas de comportamiento del torque tienden a ser constantes; sin embargo, en general, al ir aumentando el porcentaje de ciclo de trabajo aumenta el valor del torque, alcanzando su máximo con un ciclo de trabajo a 90 % para las pruebas realizadas, con un valor máximo de $0.8 \text{ N}\cdot\text{m}$ y un valor mínimo de $0.6 \text{ N}\cdot\text{m}$ para un ciclo de trabajo de 60 %. En los cálculos teóricos el máximo torque requerido es de $0.03 \text{ N}\cdot\text{m}$, es decir que las salidas de corriente del microcontrolador Rabbit son más que

suficientes para manejar todos los subsistemas. En el programa de control se ajustó la frecuencia a 50.0 Hz y el ciclo de trabajo a 25 % para la rueda de filtros, para los elementos ópticos se ocupó un ciclo de trabajo de 50 % y la misma frecuencia.

El sistema de control que se desarrolló a partir de la tarjeta Rabbit BL4S200, puede tener uso en el campo del control automático y la robótica, en donde las operaciones impliquen procesos complejos, donde se tengan instrumentos de lectura, eventos de tiempo con precisión y control de motores entre otros tipos de actuadores.

Esta aplicación desarrollada para la instrumentación astronómica, puede tener aplicación en otros campos de la instrumentación científica o la industria, donde se requiera un sistema de medición y manejo de datos o adaptabilidad a los cambios, también en el control de dispositivos de energías renovables como antenas de captación de radiación solar o aerogeneradores, en mantener y controlar las instalaciones de tratamiento de aguas residuales o controlar el consumo de energía eléctrica; en la industria médica para el movimiento remoto de los dispositivos médicos y monitorear la instrumentación médica al igual que construirla; en la industria de manufactura en los procesos de control para las líneas de producción, automatización de robots industriales y en el monitoreo de los niveles de fluidos en tanques industriales; en la industria de transporte en el monitoreo de los subsistemas de los vehículos, en identificaciones de fallas en ferrocarriles para proteger las mercancías y en la gestión de tráfico. Además tiene aplicación en el desarrollo y transformación de espacios inteligentes, como la creación de alambrado público inteligente o en el control de alarmas y sistemas de seguridad a distancia. Igualmente sería posible aplicar este tipo de tecnología al "Internet de las cosas" (IoT), rama que está en constante desarrollo y popularidad.

9.2 Problemas encontrados

La última clase de cambios se refieren a la instalación del sistema y tratan de ajustes mecánicos que hacen que el montaje sea más fácil. Por lo que el diseño inicial del controlador tiene poco impacto en este tipo de cambios. La instalación de código de diagnóstico para una prueba especial no requiere un re-trabajo en el controlador del sistema, siendo capaz de ahorrar mucho tiempo y dinero.

Un problema que se encontró fue en la instalación mecánica final del sistema de control en el instrumento PUMA, debido a insuficiencia de espacio para los conectores serial al controlador CS100 y de red Ethernet; sin embargo este problema se resolvió aumentando la distancia de separación entre el sistema de control y la placa estructural, colocando entre ellos un ensamble de dos perfiles atornillados y sobrepuestos de tipo *junquillo* (32.11 mm x 19.05 mm), creando así dos soportes fijos. Este problema se hubiera podido resolver desde la etapa de diseño si se hubieran tenido las medidas correctas para el modelo 3D del instrumento completo.

9.3 Lecciones aprendidas

Se reconoce el valor de haber aprendido a manejar un proyecto que requería varios conocimientos de diversos campos de la ingeniería, de la parte electrónica fue aprender a manejar un microcontrolador de una sola placa y a fabricar una placa de circuito impreso confiable; en la parte de programación se enfocó en el análisis estructurado del programa para no descartar ningún dato para el buen funcionamiento del instrumento; con respecto a la parte mecánica se hizo uso de conocimientos básicos de esta rama para determinar los torques necesarios y como ayuda tecnológica se ocupó un software de diseño de modelos sólidos para calcular masas y distribuciones de los subsistemas; así como las cantidades que se necesitaban.

También el desarrollo de un sistema de pruebas para comprobar que la placa Rabbit era adecuada para manejar al instrumento PUMA y la inmersión a un nuevo conocimiento como es la aplicación de conocimientos electro-ópticos a la astronomía y cómo se lleva a cabo un proyecto de instrumentación astronómica de inicio a fin y el valor de sus resultados.

9.4 Mejora para un trabajo futuro

Como siguiente clase de modificaciones de liberación o de post-fabricación, se dirige a las exigencias inevitables de la mejora de funciones. La razón de las modificaciones del sistema final de control pueden ser diferentes a los objetivos principales del proyecto, pero el trabajo real requerido es el mismo. Estas mejoras de los productos requerirán modificaciones mínimas en el software, el hardware o ambos.

En este caso para el trabajo a futuro se planea que el sistema pueda medir la temperatura del entorno (en donde se encuentra el instrumento) y esta pueda ser monitoreada y registrada. El siguiente código está dirigido específicamente para el termistor con la características de $\beta_T = 3965$ (se explica en el Apéndice I.), unido a una resistencia en serie $R_T = 2,5k\Omega$ y resistencia a $25[^\circ C] = 3k\Omega$. La placa BL4S200 utiliza un convertidor de analógico a digital, cuenta con 11 bits de resolución, 8 canales de entrada y una interfaz en serie, entregando hasta 52 k muestras por segundo.

Los cálculos necesarios para determinar los valores de temperatura de lectura son los siguientes: la temperatura en Celsius se evaluará como $T_c = T_k - 273,15$ y la temperatura en grados Fahrenheit será $T_f = 1,8 \cdot (T_k)$

La temperatura T_k es la temperatura en kelvin y se calcula como:

$$T_k = \frac{\beta_T \cdot T_{KSTD}}{T_{KSTD} \cdot \log \left(\left| \frac{-D_{RAW} \cdot R_S}{R_{TSTD} \cdot (D_{RAW} - (D_{MAX} \cdot Gain))} \right| \right) + \beta_T} \quad (9.1)$$

En donde T_{KSTD} es la temperatura estándar en $^\circ K$. R_S es la resistencia de estímulo, R_{TSTD} es la resistencia del termistor a temperatura estándar, $Gain$ es el canal de ganancia calculado en $\frac{V}{V}$, D_{MAX} es el código máximo ADC y D_{RAW} es el código de ADC medido (datos sin procesar). Las

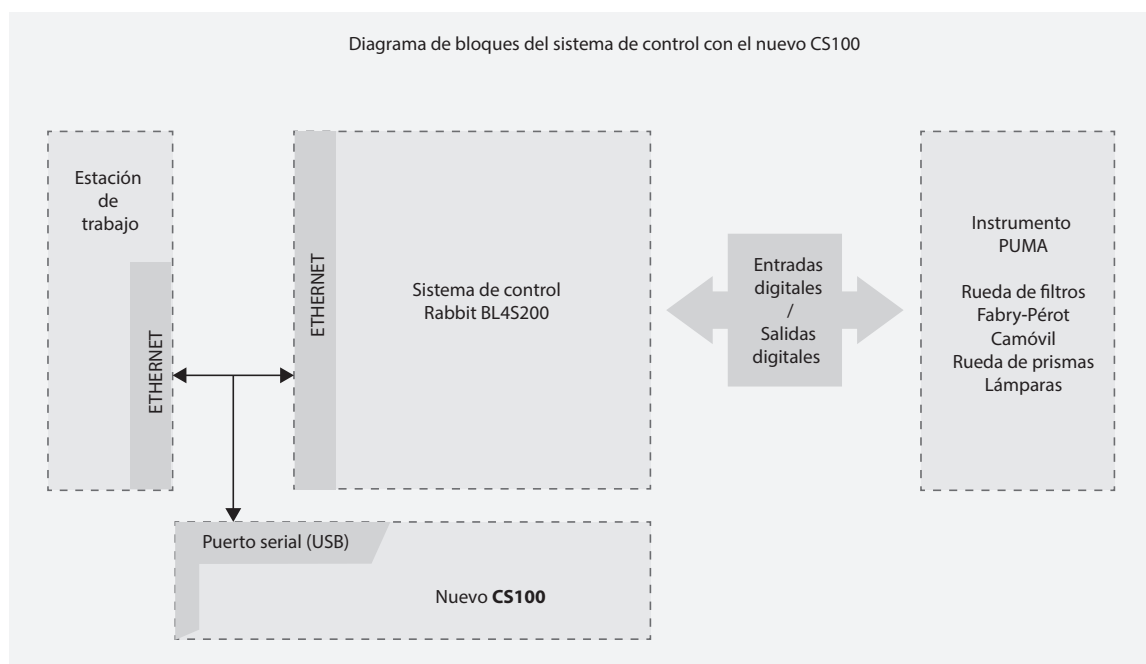
bibliotecas de la placa Rabbit proporcionan la función ANAIn que entrega el valor bruto de un canal analógico.

Donde el modo de operación soporta entradas de simple terminación, diferenciales o miliamperios. El valor del código de ganancia prevista una ganancia de 1 a 20, dependiendo del rango de tensión de entrada.

```
1 D_{raw} = anaIn(channel#, operation_mode, gain_code)
```

Antes de ejecutar el programa en la placa de prototipos BL4S200, el termistor debe ser conectado entre la AIN7 y las entradas AGND del circuito impreso del módulo externo de control, ya que se destinaron esas salidas para esta aplicación.

También se planea sustituir el controlador CS100 por uno más actualizado (ya adquirido) cuyas salidas son seriales, como lo muestra el (Cuadro 9.2). Todos los puertos serie de la placa BL4S200 pueden ser utilizados como dos puertos serie asíncronos de conexión simple, o si se conectara directamente, de este modo sería lo más adecuado para la conducción sin requerir el intercambio o modificación de hardware.



Cuadro 9.2: Comunicaciones del sistema de control con el nuevo controlador CS100.

Para un estudio más preciso sería necesario describir el volumen de datos, los tiempos de respuesta a las diversas entradas, las restricciones políticas sobre las modalidades de implantación, las restricciones ambientales y las restricciones de seguridad y confiabilidad.

Igualmente se planea mejorar el sistema haciéndolo más inteligente, es decir, en caso de que los carros móviles del instrumento PUMA o la rueda de filtros no lleguen a las posiciones en las que se les mandó llegar, ya sea por su mecanismo mecánico o eléctrico, estos errores ya no se corregirían con un simple reinicio del instrumento (como se hace actualmente para corregir este tipo de posibles fallos), y se mejoraría con un programa de control que contemplaría estas previsiones, mostrando información de en qué salida/entrada fue el error y corregir el movimiento o proporcionarle al usuario decidir sobre que acción tomar.

Con respecto a proporcionar mayor seguridad y confianza al astrónomo, y cuidar del instrumento PUMA, el sistema de control puede ser capaz de desactivar todas sus salidas utilizadas, en algún tiempo de transición en que no se esté observando, esto se realizó con la función *Finalizar* dentro del programa de control y la cual se podría incluir en la interfaz de usuario como una opción agregada; de esta manera no habría problema si alguna lámpara se dejara encendida (también se pensó en esto al diseñar el módulo auxiliar para la placa Rabbit, al ocupar determinado tipo de transistores).

Actualmente se planean migrar los programas de análisis y reducción de datos al lenguaje de programación Python, por lo que se tendría que determinar si hay alguna posibilidad de aprovechar este cambio y que se trabaje en conjunto con esta nueva modificación y con el nuevo sistema de control para realizar las observaciones astronómicas mucho más eficientes con el instrumento PUMA.

Parte II

DOCUMENTOS

ADJUNTOS

CAPÍTULO

A

Programa de control en Dynamic C

Apéndice A. Programa de control en Dynamic C

```
1 #class auto
2 #define TCPCONFIG 1
3
4 #define _PRIMARY_STATIC_IP "192.168.0.19"
5 #define _PRIMARY_NETMASK "255.255.255.0"
6 #define MY_NAMESERVER "132.248.10.2"
7 //#define MY_NAMESERVER "10.0.4.2"
8 #define MY_GATEWAY "192.168.0.254"
9
10 #memmap xmem
11 #use "dcrtcp.lib"
12 #use "BLxS2xx.lib"
13
14 #define PORT 12007
15
16 int leeComando(const char *control);
17 void selecciona(int selector);
18 void Delay1ms(int mSdelay);
19 void Eje_X(int posicion_X);
20 void Eje_Y(int posicion_Y);
21 void Eje_Z(int posicion_Z);
22 void IniciaFiltros(void);
23 void RuedaFiltros(int Filtro);
24 void Lamparas(void) ;
25 void EnciendeLamp(int estado) ;
26 void Dispositivo(char disp, int modo);
27 void Clear(void);
28 void Reset_CS100(void);
29 void Status(void);
30 void Inicializando(void);
31 //void _Externo(void);
32 //void _ClsServo(void);
33 void LoopCls();
34 void Butea(void);
35 void Finalizar(void);
36 void Overload(void);
37
38 tcp_Socket socket;
39
40 ///////////////buffers requeridos
41 char bufSaca[2]; //buffer para obtener valores posicion
42 char buffComandFP[20]; //buffer para valor de comando del Fabry-Perot
43
44 static char respuesta[100]=""; //Respuesta enviada al socket
45
46 int comando=0;
47 int termina=1;
48 int eje=0;
49 ////////////////Variables de posicion de todos los componentes//
50 static int Z=0;
51 static int Z_ant=0;
52 static int X=0;
53 static int X_ant=0;
54 static int Y=0;
55 static int Y_ant=0;
56 static int C=0;
57 static int CX=0;
58 static int CY=0;
59 static int CZ=0;
60 static int j;
61 static int Filtro=0;
```

Apéndice A. Programa de control en Dynamic C

```
static int Lamp=1;
63 static int Edo=0;
static int Etalon=1;
65 static int Prismas=1;
static int Cmovil=1;
67 static int Lazo=0;
//static int ejecutando=0;
69
//////////Variables de posicion para interrupciones//
71
static int bp_handle0;
73 static int bp_handle1;
static int bp_handle6;
75 static int bp_handle7;

77 static char button_event0;
static char button_event1;
79 static char button_event6;
static char button_event7;
81
static char_buffer[80]=""; //Buffer para lectura de Socket
83
// Button Press Interrupt Handler (ISR)
85 root void button_press0()
{
87 //set event flag
    button_event0 = 1;
89 //Turn off all interrupt flags serviced by this handler
    RSB_CLEAR_ALL_IRQ(bp_handle0);
91 }

93 root void button_press1()
{
95 //set event flag
    button_event1 = 1;
97 //Turn off all interrupt flags serviced by this handler
    RSB_CLEAR_ALL_IRQ(bp_handle1);
99 }

101 root void button_press6()
{
103 //set event flag
    button_event6 = 1;
105 //Turn off all interrupt flags serviced by this handler
    RSB_CLEAR_ALL_IRQ(bp_handle6);
107 }

109 root void button_press7()
{
111 //set event flag
    button_event7 = 1;
113 //Turn off all interrupt flags serviced by this handler
    RSB_CLEAR_ALL_IRQ(bp_handle7);
115 }
//////////Main//////////////////////////////////////
117
void main()
119 {
    int bytes_read;
121 int cnt1;
    int result;
123 int i= 0;
```

Apéndice A. Programa de control en Dynamic C

```
125     char *extraebuff;
127 //Inicializa SBC Rabbit
    brdInit();
129 //ejes
    setDigOut(18,0);
131     setDigOut(19,0);
    setDigOut(20,0);
133     setDigOut(21,0);
    setDigOut(16,0);
135     setDigOut(17,0);
//clear
137     setDigOut(25,1);
    setDigOut(26,1);
139     setDigOut(24,1);
//Reset
141     setDigOut(28,1);
//External control
143     setDigOut(27,0);

145 //Setup interrupt on channel canal and attach button_press ISR handler
//IER mask not yet known, so set it to zero for now
147     bp_handle0 = addISR(0, BL_INPUT_BLOCK, 0, &button_press0);
    bp_handle1 = addISR(1, BL_INPUT_BLOCK, 0, &button_press1);
149     bp_handle6 = addISR(6, BL_INPUT_BLOCK, 0, &button_press6);
    bp_handle7 = addISR(7, BL_INPUT_BLOCK, 0, &button_press7);
151

//Setup external interrupt on falling edge for channel 0
153     setExtInterrupt(0, BL_IRQ_RISE, bp_handle0);
    setExtInterrupt(1, BL_IRQ_RISE, bp_handle1);
155     setExtInterrupt(6, BL_IRQ_FALL, bp_handle6);
    setExtInterrupt(7, BL_IRQ_FALL, bp_handle7);
157

//Initialize event flags, counter and time stamp
159     button_event0 = 0;
    button_event1 = 0;
161     button_event6 = 0;
    button_event7 = 0;
163

//Enable ISR's
165     enableISR(bp_handle0,1);
    enableISR(bp_handle1,1);
167     enableISR(bp_handle6,1);
    enableISR(bp_handle7,1);
169

//configura puenteH como tri-state.
171     digOutTriStateConfig_H(0XFF);

173 //Inicia el Instrumento
    Inicializando();
175

//Start network and wait for interface to come up (or error exit).
177     sock_init();
    memset(buffer,0,sizeof(buffer));
179     while(termina==1){
        i = tcp_listen(&socket,PORT,0,0,NULL,0);
181         printf("\nWaiting for connection...\n");
        while(!sock_established(&socket) && sock_bytesready(&socket) <=0)
183             tcp_tick(NULL);
        printf("Connection received...\n");
185     do{
```

Apéndice A. Programa de control en Dynamic C

```

        bytes_read=sock_fastread(&socket ,buffer ,sizeof (
buffer));
187         if(bytes_read>0){
            buffer [bytes_read]=0;
189             printf ("comando= %s\n",buffer);
            extraebuff=strchr (buffer ,'\r');
191             if (extraebuff !=NULL){
                cntl=leeComando (buffer);
193                 selecciona (cntl);
            }
195         }
            buffer [0]='\0';
197     } while(tcp_tick(&socket));
}
199     printf ("Programa Terminado\n");
} //main
201
int leeComando(const char *control)
203 {
    int i;
205     char Valida [71]="x,y,z,c,e,f,i,k,l,p,v,r,s,F,o,n";
    char Buff1 [4];
207
    Buff1 [0]=control [0];
209     Buff1 [1]='\0';
    memset (buffComandFP ,'\0' ,sizeof (buffComandFP));
211     if (strstr (Valida ,Buff1) !=NULL){
        //printf ("control %s buff1 %s\n",control , Buff1);
213
        if (control [0]=='x'){
215             for (i=1;i<7;i++){
                buffComandFP [i-1]=control [i];
217             }
            comando=1;
219         }
        //FPX
221         if (control [0]=='y'){
            for (i=1;i<7;i++){
223                 buffComandFP [i-1]=control [i];
            }
            comando=2;
225         }
        //FPY
227         if (control [0]=='z'){
            for (i=2;i<7;i++){
229                 buffComandFP [i-2]=control [i];
            }
            //printf ("valor de z%s\n",buffComandFP);
231             comando=3;
        }
        //FPZ
233         if (control [0]=='c'){
            //printf ("c\n");
            bufSaca [0]=control [1];
239             bufSaca [1]='\0';
            comando=4;
241         }
        //CLR
243         if (control [0]=='e'){
            bufSaca [0]=control [1];
245             bufSaca [1]='\0';
            comando=5;

```


Apéndice A. Programa de control en Dynamic C

```
247     }
248     //Eta
249     if(control[0]=='f'){
250         bufSaca[0]=control[1];
251         bufSaca[1]=='\0';
252         comando=6;
253     }
254     //Filtros
255     if(control[0]=='i'){
256         comando=7;
257     }
258     //IniFil
259     if(control[0]=='k'){
260         bufSaca[0]=control[1];
261         bufSaca[1]=='\0';
262         comando=17;
263     }
264     //Lamp
265     if(control[0]=='l'){
266         bufSaca[0]=control[1];
267         bufSaca[1]=='\0';
268         comando=8;
269     }
270     //Lamp
271     if(control[0]=='p'){
272         bufSaca[0]=control[1];
273         bufSaca[1]=='\0';
274         comando=9;
275     }
276     //Prism
277     if(control[0]=='v'){
278         bufSaca[0]=control[1];
279         bufSaca[1]=='\0';
280         comando=10;
281     }
282     //Cmovil
283     if(control[0]=='r'){
284         comando=11;
285     }
286     //Reset CS100
287     if(control[0]=='s'){
288         comando=13;
289     }
290     //Status
291     if(control[0]=='n'){
292         comando=14;
293     }
294     //Inicializando
295     if(control[0]=='t'){
296         comando=15;
297     }
298     //Control externo
299     if(control[0]=='o'){
300         comando=16;
301     }
302     //Cierre lazo de control
303     if(control[0]=='F' && control[1]=='I' && control[2]=='N'){
304         comando=12;
305     }
306 }
307 else{
308     sock_write(&socket, "?:", strlen("?:"));

```

Apéndice A. Programa de control en Dynamic C

```
309     sock_flush(&socket);
        printf("Comando no Reconocido\n");
311 }
    return comando;
313 }//leeComando.

315 void selecciona(int selector)
    {
317     int eje=0;
        switch(selector){
319     case 1:
            eje=atoi(buffComandFP);
321             printf("eje x Fabry-Perot %d\n",eje);
                Eje_X(eje);
323             break;
        case 2:
            eje=atoi(buffComandFP);
325             printf("eje y Fabry-Perot %d\n",eje);
                Eje_Y(eje);
327             break;
        case 3:
            eje=atoi(buffComandFP);
331             printf("eje z Fabry-Perot %d\n",eje);
                Eje_Z(eje);
333             break;
        case 4:
            Clear();
335             printf("Clear al Eje %.1s de FP\n",bufSaca);
337             break;
        case 5:
            Etalon=atoi(bufSaca);
339             Dispositivo('e',Etalon);
341             printf("Posicion Etalon Seleccionada %d\n",Etalon);
            break;
343     case 6:
            Filtro=atoi(bufSaca);
345             Filtro=Filtro-1;
                printf("Filtro seleccionado %d\n",Filtro);
347             RuedaFiltros(Filtro);
            break;
349     case 7:
            IniciaFiltros();
351             printf("Rueda de Filtros Lista %d\n",Filtro);
            break;
353     case 8:
            Lamp=atoi(bufSaca);
355             printf("Lampara seleccionada %d\n",Lamp);
                Lamparas();
357             break;
        case 9:
            Prismas=atoi(bufSaca);
359             printf("Posicion de Prismas Seleccionada %d\n",Prismas);
                Dispositivo('p',Prismas);
361             break;
        case 10:
            Cmovil=atoi(bufSaca);
365             printf("Posicion de Camovil Seleccionada %d\n",Cmovil);
                Dispositivo('v',Cmovil);
367             break;
        case 11:
            printf("Reset al CS100\n");
369             Reset_CS100();
```

```

371     break;
case 12:
373     printf("Terminando Programa\n");
        Finalizar();
375     termina=0;
        break;
377 case 13:
        printf("Status\n");
379     Status();
        break;
381 case 14:
        Inicializando();
383     break;
case 16:
385     printf("Overload\n");
        // ClsServo();
387     break;
case 17:
389     Edo=atoi(bufSaca);
        EnciendeLamp(Edo);
391     break;
    }
393     comando=0;
}
395
void Delay1ms(int mSdelay)
397 {
    auto unsigned long t0;
399     t0= MS_TIMER;
        while (MS_TIMER < (t0 + mSdelay));
401 }
403
void IniciaFiltros(void)
{
405     digOutTriState_H(0,1);
    //configura canal 1 de puentes H
407     setPWM_H(1, 50.0, 25.0,HCPWM_TRI_LOW, 0);
    sock_write(&socket, ":", strlen(":"));
409     sock_flush(&socket);
    do{
411         while(button_event1==0);
            digOutTriState_H(0,2);
413             digOutTriState_H(1,2);
                Delay1ms(500);
                RuedaFiltros(1);
415                 Delay1ms(500);
                    RuedaFiltros(0);
417                     digOutTriState_H(0,2);
                        digOutTriState_H(1,2);
419
                Filtro=0;
                C=0;
                button_event1=0;
423                 sock_write(&socket, "#", strlen("#"));
                    sock_flush(&socket);
425
    }
    return;
427 }
429
void RuedaFiltros(int Filtro)
{
431     button_event0=0;
        sock_write(&socket, ":", strlen(":"));

```

Apéndice A. Programa de control en Dynamic C

```
433 sock_flush(&socket);
    //printf( "\n\t\n El filtro esta en %d\n\t\n",C);
435 while(C!=Filtro){
    if((Filtro-C)>0 && (abs(Filtro-C)>=4)){
437         C=C-1;
            digOutTriState_H(1,1);
439         setPWM_H(0, 50.0, 25.0,HCPWM_TRI_LOW, 0);
            //printf("Filtros moviendose por sentido horario\n\t\n");
441         Delay1ms(500);
            button_event0=0;
443         do{}
            while(button_event0==0);
445     }
    if((Filtro-C)>0 && (abs(Filtro-C)<4)){
447         C=C+1;
            digOutTriState_H(0,1);
449         setPWM_H(1, 50.0, 25.0,HCPWM_TRI_LOW, 0);
            //printf("Filtros moviendose por sentido antihorario\n\t\n");
451         Delay1ms(500);
            button_event0=0;
453         do{}
            while(button_event0==0);
455     }
    if((Filtro-C)<0 && (abs(Filtro-C)<4)){
457         C=C-1;
            digOutTriState_H(1, 1);
459         setPWM_H(0, 50.0, 25.0,HCPWM_TRI_LOW, 0);
            Delay1ms(500);
461         button_event0=0;
            //printf("Filtros moviendose por sentido horario\n\t\n");
463         do{}
            while(button_event0==0);
465     }
    if((Filtro-C)<0 && (abs(Filtro-C)>=4)){
467         C=C+1;
            digOutTriState_H(0,1);
469         setPWM_H(1, 50.0, 25.0,HCPWM_TRI_LOW, 0);
            //printf("Filtros moviendose por sentido antihorario\n\t\n");
471         Delay1ms(500);
            button_event0=0;
473         do{}
            while(button_event0==0);
475     }
    if((C)<0){
477         C=7;
    }
    if((C)>7){
479         C=0;
    }
481     button_event0=0;
483 }
    digOutTriState_H(0, 2);
485 digOutTriState_H(1, 2);
    C=Filtro;
487 sock_write(&socket, "#", strlen("#"));
    sock_flush(&socket);
489 //printf("\n\t\n Llego al filtro %d \n\t\n",C);
    return;
491 }

493 void Dispositivo(char disp, int modo)
    {
```

Apéndice A. Programa de control en Dynamic C

```
495 sock_write(&socket, ":", strlen(":"));
496 sock_flush(&socket);
497 switch(dispatch){
498     case 'e':{//subrutina de etalon
499         if(modos==2){ //Adentro
500             setDigIn(8);
501             setDigIn(9);
502             //printf("\n\t\n Sensor %d \n\t\n",digIn(8));
503             //printf("\n\t\n Sensor %d \n\t\n",digIn(9));
504             digOutTriState_H(7,1);
505             setPWM_H(6, 50.0, 50.0,HCPWM_TRI_LOW, 0);
506             do{}
507                 while(digIn(8)==1);
508             Etalon=2;
509         }
510         if(modos==1){
511             //printf("\n\t\n Sensor %d \n\t\n",digIn(8));
512             //printf("\n\t\n Sensor %d \n\t\n",digIn(9));
513             digOutTriState_H(6,1);
514             setPWM_H(7, 50.0, 50.0,HCPWM_TRI_LOW, 0);
515             do{}
516                 while(digIn(9)==1);
517             Etalon=1;
518         }
519         digOutTriState_H(7,2);
520         digOutTriState_H(6,2);
521         break;
522     }
523     case 'v':{//subrutina de camovil
524         if(modos==2){//if
525             //printf("\n\t\n Sensor %d \n\t\n",digIn(4));
526             //printf("\n\t\n Sensor %d \n\t\n",digIn(5));
527             setDigIn(4);
528             digOutTriState_H(5,1);
529             setPWM_H(4, 50.0, 50.0,HCPWM_TRI_LOW, 0);
530             do{}
531                 while(digIn(4)==1);
532             Cmovil=2;
533         }
534         if(modos==1){
535             //printf("\n\t\n Sensor %d \n\t\n",digIn(4));
536             //printf("\n\t\n Sensor %d \n\t\n",digIn(5));
537             setDigIn(5);
538             digOutTriState_H(4,1);
539             setPWM_H(5, 50.0, 50.0,HCPWM_TRI_LOW, 0);
540             do{}
541                 while(digIn(5)==1);
542             Cmovil=1;
543         }
544         digOutTriState_H(5,2);
545         digOutTriState_H(4,2);
546         break;
547     }
548     case 'p':{//subrutina de prismas
549         if(modos==1){//if
550             //printf("\n\t\n Sensor %d \n\t\n",digIn(2));
551             // printf("\n\t\n Sensor %d \n\t\n",digIn(3));
552             setDigIn(2);
553             digOutTriState_H(3,1);
554             setPWM_H(2, 50.0, 50.0,HCPWM_TRI_LOW, 0);
555             do{}
556                 while(digIn(2)==1);
```

Apéndice A. Programa de control en Dynamic C

```
557         Prismas=1;
558     }
559     if(modo==2){
560         //printf("\n\t\n Sensor %d \n\t\n",digIn(2));
561         //printf("\n\t\n Sensor %d \n\t\n",digIn(3));
562         setDigIn(3);
563         digOutTriState_H(2,1);
564         setPWM_H(3, 50.0, 50.0,HCPWM_TRI_LOW, 0);
565         do{
566             while(digIn(3)==1);
567             Prismas=2;
568         }
569         digOutTriState_H(3,2);
570         digOutTriState_H(2,2);
571         break;
572     }
573 }
574 //sock_write(&socket,"#",strlen("#"));
575 sock_write(&socket,"#",2);
576 sock_flush(&socket);
577 return;
578 }
579
580 void Eje_X(int posicion_X)
581 {
582     if(posicion_X > -2048 && posicion_X < 2047){
583         X=posicion_X-X_ant;
584         //printf("valor de incremeto x%d\n",X);
585         for(CX=1;CX<=abs(X)&&button_event7==0;CX++){
586             if(X>0){
587                 //printf("UP X\n");
588                 Delay1ms(5);
589                 digOut(18,1);
590                 Delay1ms(5);
591                 digOut(18,0);
592             }
593             if(X<0){
594                 //printf("DW X\n");
595                 Delay1ms(5);
596                 digOut(19,1);
597                 Delay1ms(5);
598                 digOut(19,0);
599             }
600         }
601         X_ant=posicion_X;
602     }
603     if(button_event7!=0){
604         printf("EL valor es %d\n",button_event7);
605         LoopCls();
606     }
607     button_event7=0;
608     sock_write(&socket,":",strlen(":"));
609     sock_flush(&socket);
610     return;
611 }
612
613 void Eje_Y(int posicion_Y)
614 {
615     if(posicion_Y > -2048 && posicion_Y < 2047){
616         Y=posicion_Y-Y_ant;
617         //printf("valor de incremento y%d\n",Y);
618         for(CY=1;CY<=abs(Y)&&button_event7==0;CY++){
```

Apéndice A. Programa de control en Dynamic C

```
619     if(Y>0){
620         //printf("UP Y\n");
621         Delay1ms(5);
622         digOut(20,1);
623         Delay1ms(5);
624         digOut(20,0);
625     }
626     if(Y<0){
627         //setDigOut(21,1);
628         Delay1ms(5);
629         digOut(21,1);
630         Delay1ms(5);
631         digOut(21,0);
632     }
633 }
634 Y_ant=posicion_Y;
635 }
636 if(button_event7!=0){
637     printf("EL valor es %d\n",button_event7);
638     LoopCls();
639 }
640 button_event7=0;
641 sock_write(&socket,":",strlen(":"));
642 sock_flush(&socket);
643 return;
644 }
645
646 void Eje_Z(int posicion_Z)
647 {
648     if(posicion_Z > -2048 && posicion_Z < 2047){
649         Z=posicion_Z-Z_ant;
650         //printf("valor de incremento z%d\n",Z);
651         for(CZ=1;CZ<=abs(Z)&&button_event7==0;CZ++){
652             if(Z>0){
653                 //printf("UP Z\n");
654                 Delay1ms(5);
655                 digOut(16,1);
656                 Delay1ms(5);
657                 digOut(16,0);
658             }
659             if(Z<0){
660                 //printf("DW Z\n");
661                 Delay1ms(5);
662                 digOut(17,1);
663                 Delay1ms(5);
664                 digOut(17,0);
665             }
666         }
667         Z_ant=posicion_Z;
668     }
669     if(button_event7!=0){
670         printf("EL valor es %d\n",button_event7);
671         LoopCls();
672     }
673     button_event7=0;
674     sock_write(&socket,":",strlen(":"));
675     sock_flush(&socket);
676     return;
677 }
678
679 void Lamparas(void)
680 {
```

Apéndice A. Programa de control en Dynamic C

```
681     sock_write(&socket, ":", strlen(":"));
682     sock_flush(&socket);
683     return;
684 }
685
686 void EnciendeLamp(int Edo)
687 {
688     if(Lamp==1 && Edo==0){
689         setDigOut(10,1);
690         digOut(10,1);
691         //printf("Lampara %d %d\n\t\n",Lamp,Edo);
692     }
693     if(Lamp==1 && Edo==1){
694         setDigOut(10,1);
695         digOut(10,0);
696         //printf("Lampara %d %d\n\t\n",Lamp,Edo);
697     }
698     if(Lamp==3 && Edo==0){
699         setDigOut(11,1);
700         digOut(11,1);
701         //printf("Lampara %d apagada\n\t\n",Lamp);
702     }
703     if(Lamp==3 && Edo==1){
704         setDigOut(11,1);
705         digOut(11,0);
706         //printf("Lampara %d encendida\n\t\n",Lamp);
707     }
708     if(Lamp==2 && Edo==0){
709         //printf("Lampara %d apagada\n\t\n",Lamp);
710         setDigOut(12,1);
711         digOut(12,1);
712     }
713     if(Lamp==2 && Edo==1){
714         setDigOut(12,1);
715         digOut(12,0);
716         //printf("Lampara %d %d\n\t\n",Lamp,Edo);
717     }
718     sock_write(&socket, ":", strlen(":"));
719     sock_flush(&socket);
720     return;
721 }
722
723 void Clear()
724 {
725     if(bufSaca[0]=='x'){
726         X_ant=0;
727         digOut(25,0);
728         Delay1ms(1000);
729         digOut(25,1);
730         //printf("Valor de X_ant = %d\n\t\n",X_ant);
731     }
732     if(bufSaca[0]=='y'){
733         Y_ant=0;
734         digOut(26,0);
735         Delay1ms(1000);
736         digOut(26,1);
737         //printf("Valor de Y_ant = %d\n\t\n",Y_ant);
738     }
739     if(bufSaca[0]=='z'){
740         Z_ant=0;
741         digOut(24,0);
742         Delay1ms(1000);
```


Apéndice A. Programa de control en Dynamic C

```
743     digOut(24,1);
744     //printf("Valor de Z_ant = %d\n\t\n",Z_ant);
745 }
746 sock_write(&socket,":",strlen(":"));
747 sock_flush(&socket);
748 return;
749 }

751 void Reset_CS100(void)
752 {
753     //External control activado
754     digOut(28,0);
755     Delay1ms(1000);
756     digOut(28,1);

757     printf("clear xyz");
758     digOut(25,0);
759     Delay1ms(1000);
760     digOut(25,1);

761     digOut(26,0);
762     Delay1ms(1000);
763     digOut(26,1);

764     digOut(24,0);
765     Delay1ms(1000);
766     digOut(24,1);
767     //Mandar reset

768     // posicion_X=0;
769     X_ant=0;
770     Y_ant=0;
771     Z_ant=0;
772     //posicion_Z=0;
773     //printf("Valor de Z_ant = %d\n\t\n",Z_ant);

774     //External control desactivado
775     //__digOut(27,0);
776     //Delay1ms(1000);
777     //}
778     sock_write(&socket,":",strlen(":"));
779     sock_flush(&socket);
780     return;
781 }

782

783 void Status(void)
784 {
785     int n;
786     n= sprintf(respuesta,"F= %d L= %d X,Y,Z= %d , %d , %d ETA= %d CMVIL= %d
787     PRIS= %d LAMP %d LAZO %d:",Filtro+1,Lamp,X_ant,Y_ant,Z_ant,Etalon,Cmovil,
788     Prismas,Edo,Lazo);
789     sock_write(&socket,respuesta,strlen(respuesta));
790     sock_flush(&socket);
791     printf("%s",respuesta);
792     respuesta==" ";
793     return;
794 }

795

796 void Inicializando(void)
```

Apéndice A. Programa de control en Dynamic C

```
803 {
    int n;
805     printf("Iniciando\n");
        IniciaFiltros();
807     Dispositivo('e',1);
        Dispositivo('v',1);
809     Dispositivo('p',1);
        Lamparas();
811     EnciendeLamp(0);

813     digOut(25,0);
        Delay1ms(1000);
815     digOut(25,1);

817     digOut(26,0);
        Delay1ms(1000);
819     digOut(26,1);

821     digOut(24,0);
        Delay1ms(1000);
823     digOut(24,1);

825     //___Externo();
        //___ClsServo();
827     //n=sprintf(respuesta,"INICIALIZADO:");
        //sock_write(&socket,respuesta,strlen(respuesta));
829     //sock_flush(&socket);
        //printf("%s",respuesta);
831     //respuesta=="";
        return;
833 }

835 void LoopCls()
837 {
    //printf("Se abrio el lazo del control en\n");
839     Lazo=0;
        //printf("Lazo= %d\n",Lazo);
841     return;
    }

843 void Finalizar()
845 {
    char Bank_2;
847     char Bank_3;
        int canal;

849     //Banco de motores
851     for(canal=0; canal<= 7; ++canal){
            digOutTriState_H(canal, 2);
853     }
        //Banco de Ejes 22&23 N.C.
855     digOutBank(2,Bank_2);
        //Banco de funciones del CS100
857     digOutBank(3,Bank_3);
        return;
859 }

861 void Overload()
    {
863     setDigIn(13);
```

Apéndice A. Programa de control en Dynamic C

```
865 digIn(13)==1;  
867 setDigIn(14);  
    digIn(14)==1;  
869  
    setDigIn(15);  
871 digIn(15)==1;  
    }
```

CAPÍTULO

B

Código del cliente en lenguaje Visual Studio
C++

Apéndice B. Código del cliente en lenguaje Visual Studio C++

```
// winclient3.cpp: define el punto de entrada de la aplicacion de consola.
2
#include "stdafx.h"
4 #pragma comment (lib, "ws2_32.lib")//Winsock Library

6 void timer (int);

8 int _tmain(int argc, _TCHAR* argv[])
{
10 WSADATA wsa;
   SOCKET sck;
12 int cnt;
   struct sockaddr_in ip_info;
14 char server_reply[2000];
   int recv_size;
16 char sendbuff[20];
   char *message;

18     printf ("Installing");
20 for (cnt = 1; cnt <= 3; cnt++){
   timer (1);
22 printf (".");
   }

24 if (WSAStartup (MAKEDWORD (2,2), &wsa) == INVALID_SOCKET)
26 printf ("WSA error\n");
   else

28 //Create a socket
30 if ((sck = socket (AF_INET, SOCK_STREAM, 0)) == INVALID_SOCKET)
   printf ("Socket error\n");
32 else{
   printf("Socket created.\n");
34 ip_info.sin_addr.s_addr = inet_addr("10.10.6.100");
   ip_info.sin_family = AF_INET;
36 ip_info.sin_port = htons (23);
   }

38 //Connect to remote server
40 if (connect (sck, (struct sockaddr *)&ip_info, sizeof (ip_info)) < 0)
   printf ("\nNot connected\n");
42 else
   printf ("\nSession is up\n");

44 //Send some data
46 //message = "x -50;";
   message = "y +10;";
48 //message = "z -100;";
   //message = "cy;";
50 //message = "e0;";
   //message = "f4;";
52 //message = "i;";
   //message = "l1;";
54 //message = "p0;";
   //message = "v1;";
56 //message = "r;";
   //message = "FIN;";

58 if( send(sck , message , strlen(message) , 0) < 0)
60 {
   puts("Send failed");
}
```

Apéndice B. Código del cliente en lenguaje Visual Studio C++

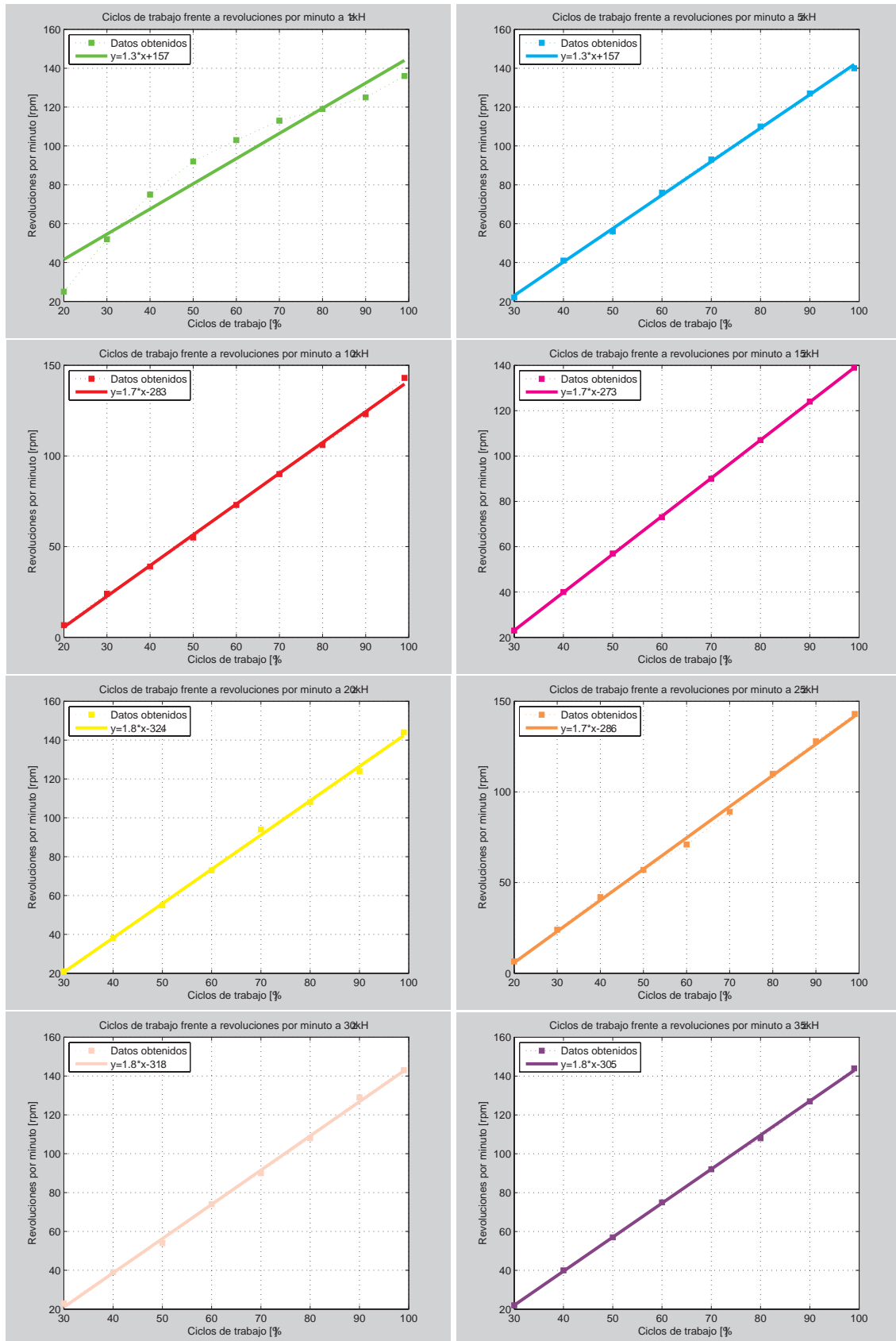
```
62     return 1;
63 }
64 puts("Data Send\n");
65
66 //Receive a reply from the server
67 if((recv_size = recv(sck, server_reply , 2000 , 0)) == SOCKET_ERROR)
68 {
69     puts("recv failed");
70 }
71
72 puts("Reply received\n");
73
74
75 //Add a NULL terminating character to make it a proper string before
76 printing
77 server_reply[recv_size] = '\0';
78 puts(server_reply);
79 closesocket(sck);
80 return 0;
81 }
82
83 // Timer function:
84 void timer (int sec)
85 {
86     clock_t end;
87
88     end = clock () + sec * CLOCKS_PER_SEC;
89     while (clock () <= end);
90 }
```


CAPÍTULO

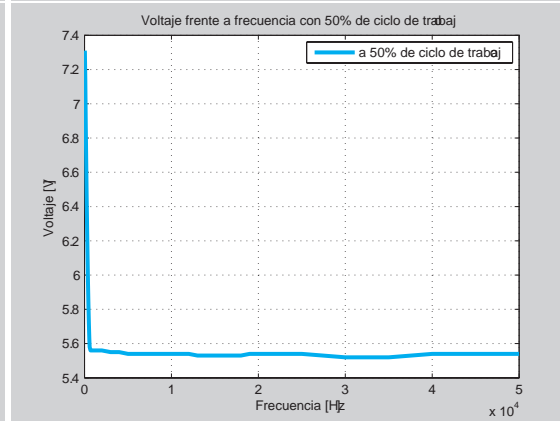
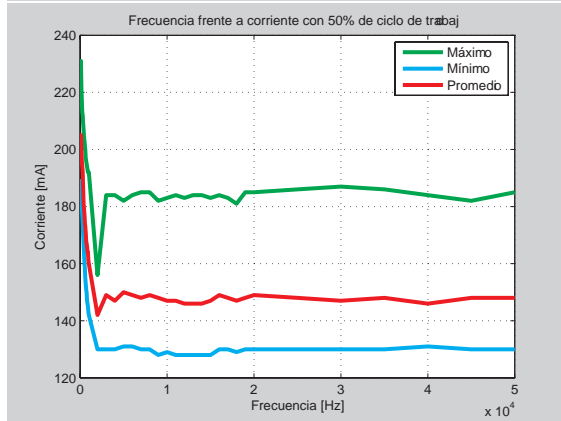
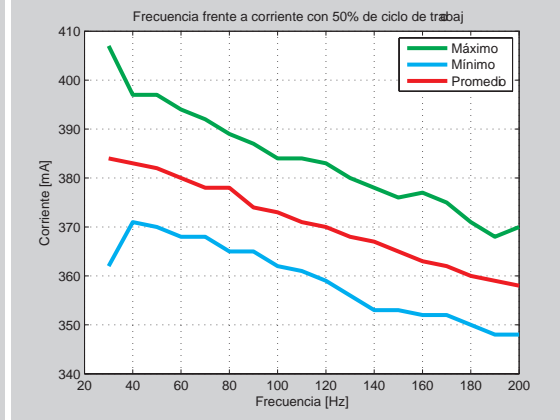
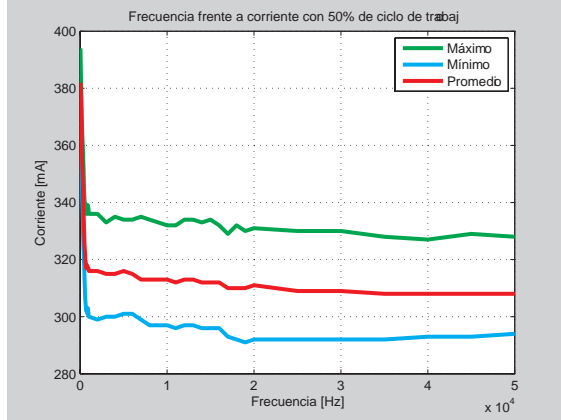
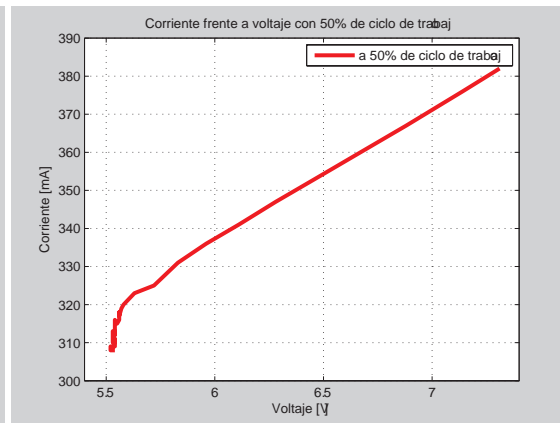
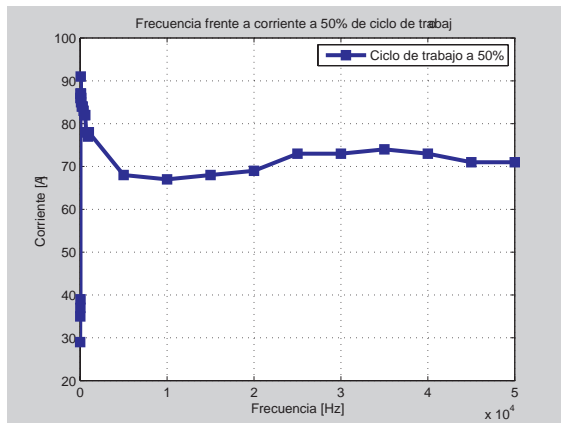
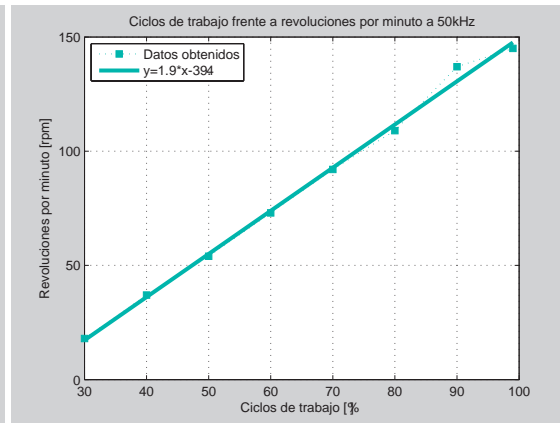
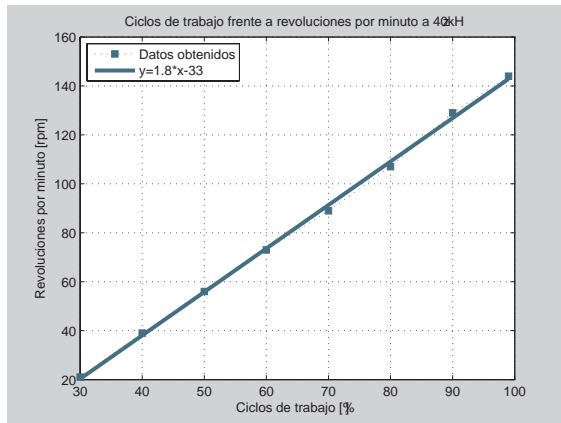
C

Gráficas

Apéndice C. Gráficas



Apéndice C. Gráficas



CAPÍTULO

D

Datos obtenidos de las pruebas del puente H

Apéndice D. Datos obtenidos de las pruebas del puente H

Cuadro D.1: Mediciones de las simulaciones en laboratorio para los motores

| Duty cycle Frecuencia [Hz] | 40% | | | 50% | | | 60% | | |
|-------------------------------|-------------|----------------|-----------------|-------------|----------------|-----------------|-------------|----------------|-----------------|
| | Voltaje [V] | Corriente [mA] | Velocidad [rpm] | Voltaje [V] | Corriente [mA] | Velocidad [rpm] | Voltaje [V] | Corriente [mA] | Velocidad [rpm] |
| 50 | 6.16 | 398 | 39.4 | 7.32 | 426 | 50.5 | 8.51 | 483 | 63.1 |
| 100 | 5.81 | 385 | 37.2 | 7.08 | 414 | 48.6 | 8.31 | 472 | 60.5 |
| 150 | 5.55 | 374 | 35.1 | 6.81 | 380 | 46 | 8.04 | 457 | 57.4 |
| 200 | 5.32 | 362 | 32.6 | 6.58 | 395 | 44.4 | 7.81 | 448 | 55.8 |
| 250 | 5.13 | 351 | 31 | 6.37 | 332 | 42.2 | 7.59 | 440 | 53.5 |
| 300 | 4.95 | 344 | 29 | 6.18 | 382 | 40.2 | 7.4 | 432 | 52.1 |
| 350 | 4.79 | 337 | 27.8 | 6.01 | 373 | 38.4 | 7.22 | 424 | 50.2 |
| 400 | 4.66 | 331 | 26.5 | 5.87 | 372 | 37.7 | 7.07 | 417 | 48.5 |
| 450 | 4.56 | 326 | 25.3 | 5.74 | 357 | 36.4 | 6.94 | 412 | 46.8 |
| 500 | 4.45 | 324 | 24.2 | 5.62 | 356 | 34.9 | 6.84 | 408 | 46.1 |
| 550 | 4.32 | 322 | 23.2 | 5.57 | 350 | 33.9 | 6.82 | 408 | 45.7 |
| 600 | 4.30 | 320 | 22.4 | 5.56 | 350 | 33.6 | 6.82 | 408 | 45.6 |
| 650 | 4.29 | 319 | 22.2 | 5.56 | 351 | 33.3 | 6.82 | 406 | 45.2 |
| 700 | 4.28 | 317 | 21.9 | 5.55 | 347 | 33.3 | 6.82 | 405 | 45.3 |
| 750 | 4.28 | 318 | 21.9 | 5.55 | 354 | 32.9 | 6.82 | 406 | 44.9 |
| 800 | 4.28 | 316 | 21.7 | 5.55 | 346 | 32.9 | 6.82 | 405 | 44.8 |
| 850 | 4.28 | 317 | 21.7 | 5.55 | 345 | 32.8 | 6.82 | 406 | 44.9 |
| 900 | 4.27 | 317 | 21.5 | 5.55 | 347 | 32.9 | 6.82 | 407 | 44.8 |
| 950 | 4.27 | 317 | 21.2 | 5.55 | 347 | 33.1 | 6.82 | 407 | 44.9 |
| 1000 | 4.27 | 317 | 21.3 | 5.55 | 347 | 33.1 | 6.81 | 405 | 44.6 |
| 2000 | 4.27 | 316 | 20.5 | 5.54 | 352 | 32.3 | 6.81 | 403 | 44.4 |
| 3000 | 4.26 | 316 | 20.2 | 5.54 | 351 | 32.3 | 6.8 | 402 | 44.1 |
| 4000 | 4.26 | 316 | 20.1 | 5.54 | 351 | 32.4 | 6.8 | 402 | 43.9 |
| 5000 | 4.26 | 316 | 20.1 | 5.53 | 359 | 31.8 | 6.8 | 402 | 44.2 |
| 6000 | 4.26 | 317 | 19.9 | 5.53 | 351 | 32 | 6.8 | 402 | 43.8 |
| 7000 | 4.26 | 317 | 19.8 | 5.53 | 351 | 32.2 | 6.8 | 402 | 43.7 |
| 8000 | 4.26 | 315 | 19.8 | 5.53 | 351 | 31.8 | 6.79 | 402 | 43.6 |
| 9000 | 4.26 | 318 | 19.8 | 5.53 | 351 | 31.5 | 6.8 | 401 | 43.6 |
| 10000 | 4.26 | 318 | 19.4 | 5.53 | 353 | 31.3 | 6.79 | 401 | 43.1 |
| 11000 | 4.25 | 315 | 19.4 | 5.53 | 351 | 31.6 | 6.79 | 400 | 43.3 |
| 12000 | 4.25 | 317 | 19.1 | 5.52 | 348 | 31.3 | 6.79 | 399 | 43.3 |
| 13000 | 4.25 | 315 | 19.8 | 5.52 | 349 | 31.1 | 6.79 | 400 | 43.2 |
| 14000 | 4.26 | 314 | 19.5 | 5.52 | 350 | 31.4 | 6.79 | 399 | 43.4 |
| 15000 | 4.27 | 317 | 19.6 | 5.52 | 350 | 30.6 | 6.78 | 399 | 43.1 |
| 16000 | 4.26 | 314 | 19.4 | 5.53 | 351 | 30.6 | 6.78 | 402 | 43.4 |
| 17000 | 4.25 | 316 | 19.3 | 5.52 | 354 | 31.0 | 6.79 | 397 | 43.4 |
| 18000 | 4.26 | 315 | 19.5 | 5.52 | 353 | 30.7 | 6.77 | 400 | 43.2 |
| 19000 | 4.26 | 314 | 19.7 | 5.52 | 351 | 30.4 | 6.77 | 399 | 42.8 |
| 20000 | 4.26 | 315 | 19.5 | 5.52 | 352 | 30.6 | 6.78 | 400 | 43.9 |
| 25000 | 4.27 | 315 | 19.4 | 5.52 | 354 | 30.5 | 6.77 | 399 | 43.3 |
| 30000 | 4.25 | 316 | 19.3 | 5.51 | 351 | 30.4 | 6.77 | 400 | 43.2 |
| 35000 | 4.26 | 315 | 19.4 | 5.5 | 348 | 30.2 | 6.76 | 399 | 42.6 |
| 40000 | 4.27 | 315 | 19.3 | 5.52 | 350 | 30.5 | 6.77 | 399 | 42.6 |
| 45000 | 4.23 | 313 | 18.8 | 5.52 | 350 | 30.5 | 6.78 | 398 | 42.7 |
| 50000 | 4.25 | 314 | 18.9 | 5.52 | 350 | 30.7 | 6.77 | 399 | 42.8 |

Apéndice D. Datos obtenidos de las pruebas del puente H

Cuadro D.2: Mediciones de las simulaciones en laboratorio para los motores

| Duty cycle Frecuencia [Hz] | 70% | | | 80% | | | 90% | | |
|-------------------------------|-------------|----------------|-----------------|-------------|----------------|-----------------|-------------|----------------|-----------------|
| | Voltaje [V] | Corriente [mA] | Velocidad [rpm] | Voltaje [V] | Corriente [mA] | Velocidad [rpm] | Voltaje [V] | Corriente [mA] | Velocidad [rpm] |
| 50 | 9.44 | 517 | 67.2 | 10.28 | 562 | 74.9 | 11.01 | 596 | 78.1 |
| 100 | 9.24 | 510 | 65.5 | 10.06 | 556 | 72.8 | 10.81 | 589 | 76.7 |
| 150 | 9.00 | 505 | 62.7 | 9.86 | 546 | 71.0 | 10.64 | 584 | 74.9 |
| 200 | 8.78 | 495 | 60.6 | 9.65 | 539 | 69.0 | 10.63 | 583 | 74.5 |
| 250 | 8.58 | 484 | 58.6 | 9.47 | 531 | 67.9 | 10.61 | 580 | 74.3 |
| 300 | 8.40 | 477 | 57.0 | 9.35 | 524 | 66.4 | 10.61 | 577 | 74.3 |
| 350 | 8.23 | 470 | 55.3 | 9.35 | 522 | 65.6 | 10.61 | 576 | 74.2 |
| 400 | 8.11 | 464 | 53.8 | 9.34 | 523 | 65.4 | 10.61 | 573 | 73.6 |
| 450 | 8.09 | 463 | 53.8 | 9.34 | 520 | 65.7 | 10.61 | 574 | 73.3 |
| 500 | 8.08 | 464 | 53.3 | 9.34 | 520 | 65.5 | 10.61 | 575 | 73.4 |
| 550 | 8.08 | 461 | 52.6 | 9.34 | 518 | 66.1 | 10.61 | 575 | 73.1 |
| 600 | 8.08 | 462 | 53.4 | 9.34 | 523 | 65.1 | 10.60 | 576 | 73.2 |
| 650 | 8.08 | 462 | 53.3 | 9.34 | 520 | 65.3 | 10.61 | 575 | 73.5 |
| 700 | 8.08 | 461 | 52.8 | 9.34 | 519 | 64.6 | 10.61 | 576 | 73.1 |
| 750 | 8.08 | 461 | 52.7 | 9.34 | 525 | 64.4 | 10.61 | 574 | 73.2 |
| 800 | 8.08 | 462 | 52.9 | 9.34 | 533 | 65.0 | 10.60 | 575 | 73.5 |
| 850 | 8.08 | 462 | 52.5 | 9.34 | 526 | 64.9 | 10.60 | 575 | 73.4 |
| 900 | 8.08 | 463 | 52.0 | 9.34 | 522 | 64.6 | 10.60 | 577 | 73.2 |
| 950 | 8.07 | 462 | 52.0 | 9.34 | 520 | 64.5 | 10.60 | 575 | 73.3 |
| 1000 | 8.07 | 459 | 52.6 | 9.34 | 519 | 64.4 | 10.60 | 577 | 72.9 |
| 2000 | 8.07 | 460 | 51.8 | 9.33 | 520 | 64.0 | 10.59 | 576 | 72.9 |
| 3000 | 8.06 | 461 | 51.8 | 9.32 | 519 | 63.7 | 10.59 | 576 | 73.2 |
| 4000 | 8.06 | 459 | 49.7 | 9.32 | 520 | 63.5 | 10.59 | 574 | 73 |
| 5000 | 8.06 | 460 | 47.3 | 9.31 | 519 | 63.6 | 10.58 | 574 | 73.3 |
| 6000 | 8.05 | 460 | 46.6 | 9.31 | 518 | 63.8 | 10.57 | 572 | 72.8 |
| 7000 | 8.05 | 459 | 46.3 | 9.31 | 517 | 63.5 | 10.57 | 573 | 73.1 |
| 8000 | 8.05 | 459 | 46.3 | 9.30 | 518 | 63.6 | 10.56 | 572 | 72.7 |
| 9000 | 8.05 | 459 | 49.9 | 9.30 | 517 | 63.5 | 10.56 | 572 | 72.7 |
| 10000 | 8.04 | 459 | 50.5 | 9.30 | 517 | 63.5 | 10.56 | 571 | 72.7 |
| 12000 | 8.03 | 459 | 48.7 | 9.29 | 516 | 63.5 | 10.56 | 572 | 72.6 |
| 14000 | 8.04 | 459 | 48.8 | 9.28 | 517 | 63.7 | 10.55 | 573 | 72.6 |
| 16000 | 8.04 | 460 | 48.3 | 9.29 | 518 | 63.8 | 10.54 | 570 | 72.3 |
| 18000 | 8.03 | 458 | 46.9 | 9.28 | 517 | 64.0 | 10.54 | 570 | 72.2 |
| 20000 | 8.03 | 459 | 46.0 | 9.28 | 515 | 63.8 | 10.55 | 570 | 72.1 |
| 17000 | 8.03 | 458 | 46.3 | 9.28 | 513 | 63.3 | 10.53 | 571 | 72.5 |
| 18000 | 8.02 | 459 | 46.4 | 9.27 | 514 | 63.6 | 10.53 | 572 | 72.6 |
| 19000 | 8.01 | 459 | 46.3 | 9.27 | 517 | 63.1 | 10.52 | 572 | 72.6 |
| 20000 | 8.02 | 457 | 46.7 | 9.27 | 514 | 63.5 | 10.52 | 570 | 72.2 |
| 25000 | 8.01 | 457 | 47.0 | 9.25 | 515 | 63.3 | 10.52 | 571 | 72.1 |
| 30000 | 8.00 | 458 | 47.0 | 9.23 | 513 | 63.6 | 10.51 | 569 | 71.9 |
| 35000 | 8.00 | 459 | 47.4 | 9.23 | 513 | 63.4 | 10.54 | 571 | 71.7 |
| 40000 | 8.00 | 460 | 51.4 | 9.25 | 515 | 63.6 | 10.60 | 571 | 72.3 |
| 45000 | 7.99 | 458 | 47.0 | 9.24 | 514 | 62.9 | 10.94 | 587 | 74.6 |
| 50000 | 8.00 | 459 | 46.8 | 9.25 | 515 | 63.3 | 11.14 | 597 | 76.1 |
| | | | | | | 63.2 | 11.34 | 605 | 78.1 |

Apéndice D. Datos obtenidos de las pruebas del puente H

Cuadro D.3: Mediciones de las simulaciones en laboratorio para los motores

| Frecuencia [Hz] | Duty cycle | | | | | | | | |
|-----------------|------------|------|------|------|------|------|------|--|--|
| | 40 % | 50 % | 60 % | 70 % | 80 % | 90 % | 99 % | | |
| 50 | 39.4 | 50.5 | 63.1 | 67.2 | 74.9 | 78.1 | 82.9 | | |
| 100 | 37.2 | 48.6 | 60.5 | 65.5 | 72.8 | 76.7 | 82.0 | | |
| 150 | 35.1 | 46.0 | 57.4 | 62.7 | 71.0 | 74.9 | 81.3 | | |
| 200 | 32.6 | 44.4 | 55.8 | 60.6 | 69.0 | 74.5 | 81.3 | | |
| 250 | 31.0 | 42.2 | 53.5 | 58.6 | 67.9 | 74.3 | 80.4 | | |
| 300 | 29.0 | 40.2 | 52.1 | 57.0 | 66.4 | 74.3 | 80.0 | | |
| 350 | 27.8 | 38.4 | 50.2 | 55.3 | 65.6 | 74.2 | 79.6 | | |
| 400 | 26.8 | 37.7 | 48.5 | 53.8 | 65.4 | 73.6 | 79.4 | | |
| 450 | 25.3 | 36.4 | 46.8 | 53.6 | 65.7 | 73.3 | 78.0 | | |
| 500 | 24.2 | 34.9 | 46.1 | 53.3 | 65.5 | 73.4 | 76.5 | | |
| 550 | 23.2 | 33.9 | 45.7 | 52.6 | 66.1 | 73.1 | 79.0 | | |
| 600 | 22.4 | 33.6 | 45.6 | 53.4 | 65.1 | 73.2 | 79.4 | | |
| 650 | 22.2 | 33.3 | 45.2 | 53.3 | 65.3 | 73.5 | 79.1 | | |
| 700 | 21.9 | 33.3 | 45.3 | 52.8 | 64.6 | 73.1 | 79.9 | | |
| 750 | 21.9 | 32.9 | 44.9 | 52.7 | 64.4 | 73.2 | 79.9 | | |
| 800 | 21.7 | 32.9 | 44.8 | 52.9 | 65.0 | 73.5 | 79.4 | | |
| 850 | 21.7 | 32.8 | 44.9 | 52.5 | 64.9 | 73.4 | 78.8 | | |
| 900 | 21.5 | 32.9 | 44.8 | 52.0 | 64.6 | 73.2 | 79.2 | | |
| 950 | 21.2 | 33.1 | 44.9 | 52.0 | 64.5 | 73.3 | 79.5 | | |
| 1000 | 21.3 | 33.1 | 44.6 | 52.6 | 64.4 | 72.9 | 79.9 | | |
| 2000 | 20.8 | 32.3 | 44.4 | 52.6 | 64.0 | 72.9 | 79.2 | | |
| 3000 | 20.2 | 32.3 | 44.1 | 51.8 | 63.7 | 73.2 | 79.6 | | |
| 4000 | 20.1 | 32.4 | 43.9 | 49.7 | 63.5 | 73.0 | 77.3 | | |
| 5000 | 20.1 | 31.8 | 44.2 | 47.3 | 63.6 | 73.3 | 78.9 | | |
| 6000 | 19.9 | 32.0 | 43.8 | 46.6 | 63.8 | 72.8 | 78.4 | | |
| 7000 | 19.8 | 32.2 | 43.7 | 46.3 | 63.5 | 73.1 | 77.8 | | |
| 8000 | 19.8 | 31.8 | 43.6 | 46.3 | 63.6 | 72.7 | 76.2 | | |
| 9000 | 19.8 | 31.5 | 43.6 | 49.9 | 63.5 | 72.7 | 77.1 | | |
| 10000 | 19.4 | 31.3 | 43.1 | 50.5 | 63.5 | 72.7 | 77.1 | | |
| 11000 | 19.4 | 31.6 | 43.3 | 48.7 | 63.5 | 72.6 | 77.9 | | |
| 12000 | 19.1 | 31.3 | 43.3 | 48.8 | 63.7 | 72.6 | 77.8 | | |
| 13000 | 19.8 | 31.1 | 43.2 | 48.3 | 63.8 | 72.3 | 77.4 | | |
| 14000 | 19.5 | 31.4 | 43.4 | 46.9 | 64.0 | 72.2 | 75.9 | | |
| 15000 | 19.6 | 30.6 | 43.1 | 46.0 | 63.8 | 72.1 | 76.0 | | |
| 16000 | 19.4 | 30.6 | 43.4 | 46.3 | 63.6 | 72.5 | 78.1 | | |
| 17000 | 19.3 | 31.0 | 43.4 | 46.9 | 63.6 | 72.6 | 79.2 | | |
| 18000 | 19.5 | 30.7 | 43.2 | 46.4 | 63.1 | 72.6 | 77.9 | | |
| 19000 | 19.7 | 30.4 | 42.8 | 46.3 | 63.5 | 72.2 | 77.5 | | |
| 20000 | 19.5 | 30.6 | 43.0 | 46.7 | 63.3 | 72.1 | 77.6 | | |
| 25000 | 19.4 | 30.5 | 43.3 | 47.0 | 63.6 | 71.9 | 77.8 | | |
| 30000 | 19.3 | 30.4 | 43.2 | 47.0 | 63.4 | 71.7 | 78.4 | | |
| 35000 | 19.4 | 30.2 | 42.6 | 47.4 | 63.6 | 72.3 | 77.8 | | |
| 40000 | 19.3 | 30.5 | 42.6 | 51.4 | 62.9 | 74.6 | 77.2 | | |
| 45000 | 18.8 | 30.5 | 42.7 | 47.0 | 63.3 | 76.1 | 76.7 | | |
| 50000 | 18.9 | 30.7 | 42.8 | 46.8 | 63.2 | 78.1 | 77.0 | | |

CAPÍTULO

E

Hojas de datos de los dispositivos electrónicos

Single-Channel: 6N137, HCPL2601, HCPL2611 Dual-Channel: HCPL2630, HCPL2631 High Speed 10MBit/s Logic Gate Optocouplers

Features

- Very high speed – 10 MBit/s
- Superior CMR – 10 kV/μs
- Double working voltage-480V
- Fan-out of 8 over -40°C to +85°C
- Logic gate output
- Strobable output
- Wired OR-open collector
- U.L. recognized (File # E90700)

Applications

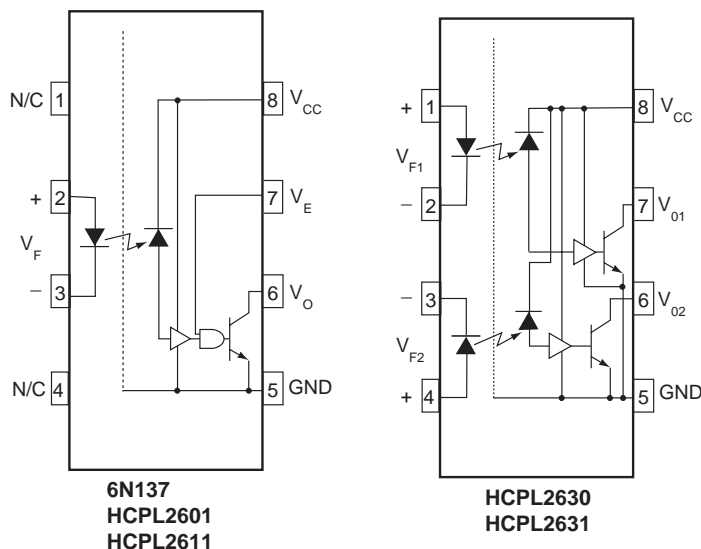
- Ground loop elimination
- LSTTL to TTL, LSTTL or 5-volt CMOS
- Line receiver, data transmission
- Data multiplexing
- Switching power supplies
- Pulse transformer replacement
- Computer-peripheral interface

Description

The 6N137, HCPL2601, HCPL2611 single-channel and HCPL2630, HCPL2631 dual-channel optocouplers consist of a 850 nm AlGaAs LED, optically coupled to a very high speed integrated photo-detector logic gate with a strobable output. This output features an open collector, thereby permitting wired OR outputs. The coupled parameters are guaranteed over the temperature range of -40°C to +85°C. A maximum input signal of 5mA will provide a minimum output sink current of 13mA (fan out of 8).

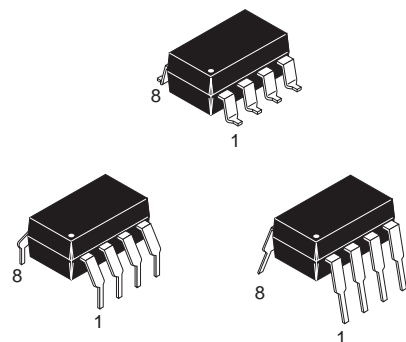
An internal noise shield provides superior common mode rejection of typically 10kV/μs. The HCPL2601 and HCPL2631 has a minimum CMR of 5kV/μs. The HCPL2611 has a minimum CMR of 10kV/μs.

Schematics



A 0.1μF bypass capacitor must be connected between pins 8 and 5⁽¹⁾.

Package Outlines



Truth Table (Positive Logic)

| Input | Enable | Output |
|-------|--------|--------|
| H | H | L |
| L | H | H |
| H | L | H |
| L | L | H |
| H | NC | L |
| L | NC | H |

Absolute Maximum Ratings ($T_A = 25^\circ\text{C}$ unless otherwise specified)

Stresses exceeding the absolute maximum ratings may damage the device. The device may not function or be operable above the recommended operating conditions and stressing the parts to these levels is not recommended. In addition, extended exposure to stresses above the recommended operating conditions may affect device reliability. The absolute maximum ratings are stress ratings only.

| Symbol | Parameter | | Value | Units |
|-----------------------------------|---|-----------------------------|----------------|------------------|
| T_{STG} | Storage Temperature | | -55 to +125 | $^\circ\text{C}$ |
| T_{OPR} | Operating Temperature | | -40 to +85 | $^\circ\text{C}$ |
| T_{SOL} | Lead Solder Temperature (for wave soldering only)* | | 260 for 10 sec | $^\circ\text{C}$ |
| EMITTER | | | | |
| I_{F} | DC/Average Forward | Single Channel | 50 | mA |
| | Input Current | Dual Channel (Each Channel) | 30 | |
| V_{E} | Enable Input Voltage Not to Exceed V_{CC} by more than 500mV | Single Channel | 5.5 | V |
| V_{R} | Reverse Input Voltage | Each Channel | 5.0 | V |
| P_{I} | Power Dissipation | Single Channel | 100 | mW |
| | | Dual Channel (Each Channel) | 45 | |
| DETECTOR | | | | |
| V_{CC} (1 minute max) | Supply Voltage | | 7.0 | V |
| I_{O} | Output Current | Single Channel | 50 | mA |
| | | Dual Channel (Each Channel) | 50 | |
| V_{O} | Output Voltage | Each Channel | 7.0 | V |
| P_{O} | Collector Output | Single Channel | 85 | mW |
| | Power Dissipation | Dual Channel (Each Channel) | 60 | |

*For peak soldering reflow, please refer to the Reflow Profile on page 11.

Recommended Operating Conditions

The Recommended Operating Conditions table defines the conditions for actual device operation. Recommended operating conditions are specified to ensure optimal performance to the datasheet specifications. Fairchild does not recommend exceeding them or designing to absolute maximum ratings.

| Symbol | Parameter | Min. | Max. | Units |
|-----------------|----------------------------|------|-----------------|------------------|
| I_{FL} | Input Current, Low Level | 0 | 250 | μA |
| I_{FH} | Input Current, High Level | *6.3 | 15 | mA |
| V_{CC} | Supply Voltage, Output | 4.5 | 5.5 | V |
| V_{EL} | Enable Voltage, Low Level | 0 | 0.8 | V |
| V_{EH} | Enable Voltage, High Level | 2.0 | V_{CC} | V |
| T_{A} | Low Level Supply Current | -40 | +85 | $^\circ\text{C}$ |
| N | Fan Out (TTL load) | | 8 | |

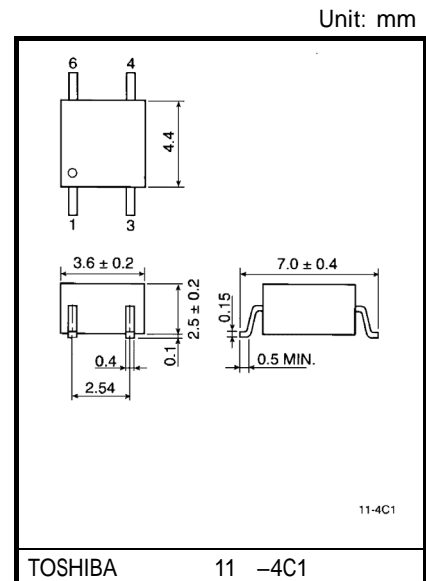
*6.3mA is a guard banded value which allows for at least 20% CTR degradation. Initial input current threshold value is 5.0mA or less.

TLP127

Programmable Controllers
 DC -Output Module
 Telecommunication

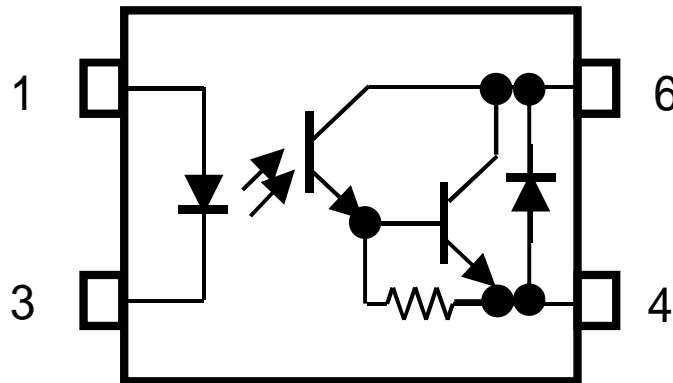
The TOSHIBA mini-flat coupler TLP127 is a small outline coupler, suitable for surface mount assembly. TLP127 consists of a gallium arsenide infrared emitting diode, optically coupled to a Darlington photo transistor with an integral base resistor, and provides 300V V_{CEO}.

- Collector -emitter voltage: 300 V (min)
- Current transfer ratio: 1000 % (min)
- Isolation voltage: 2500 Vrms (min)
- UL recognized: UL1577, file no. E67349
- BSI approved: BS EN60065:2002, certificate no.8927
 BS EN60950-1:2002, certificate no.8928



Weight: 0.09 g (typ.)

Pin Configurations (top view)



- 1: ANODE
- 3: CATHODE
- 4: EMITTER
- 6: COLLECTOR

Start of commercial production
 1988 /04

Absolute Maximum Ratings (Ta = 25°C)

| Characteristic | | Symbol | Rating | Unit |
|--|--|------------------------|------------------------------|---------|
| LED | Forward current | I_F | 50 | mA |
| | Forward current derating | $I_F / ^\circ\text{C}$ | -0.7 (Ta = 53°C) | mA / °C |
| | Pulse forward current | I_{FP} | 1 (100 μs pulse, 100pps) | A |
| | Reverse voltage | V_R | 5 | V |
| | Junction temperature | T_j | 125 | °C |
| Detector | Collector –emitter voltage | V_{CEO} | 300 | V |
| | Emitter –collector voltage | V_{ECO} | 0.3 | V |
| | Collector current | I_C | 150 | mA |
| | Collector power dissipation | P_C | 150 | mW |
| | Collector power dissipation derating (Ta = 25°C) | $P_C / ^\circ\text{C}$ | -1.5 | mW / °C |
| | Junction temperature | T_j | 125 | °C |
| Storage temperature range | | T_{stg} | -55~125 | °C |
| Operating temperature range | | T_{opr} | -55~100 | °C |
| Lead soldering temperature | | T_{sol} | 260 (10s) | °C |
| Total package power dissipation | | P_T | 200 | mW |
| Total package power dissipation derating (Ta = 25°C) | | $P_T / ^\circ\text{C}$ | -2.0 | mW / °C |
| Isolation voltage (Note 1) | | BV_S | 2500 (AC, 1minute, R.H. 60%) | Vrms |

Note: Using continuously under heavy loads (e.g. the application of high temperature/current/voltage and the significant change in temperature, etc.) may cause this product to decrease in the reliability significantly even if the operating conditions (i.e. operating temperature/current/voltage, etc.) are within the absolute maximum ratings.

Please design the appropriate reliability upon reviewing the Toshiba Semiconductor Reliability Handbook ("Handling Precautions"/"Derating Concept and Methods") and individual reliability data (i.e. reliability test report and estimated failure rate, etc.).

(Note 1) Device considered a two terminal device: Pins 1, 3 shorted together and pins 4, 6 shorted together.



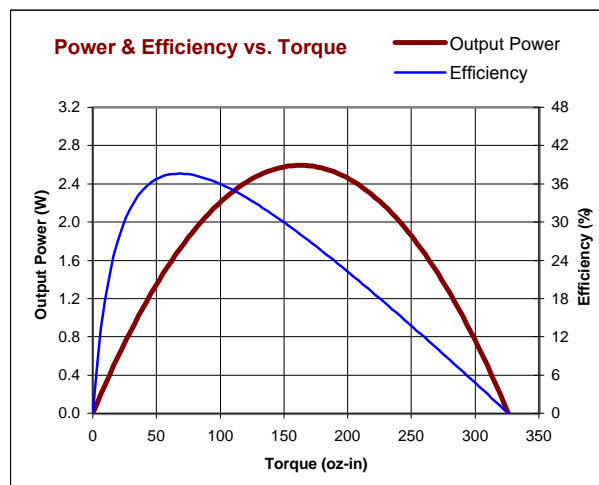
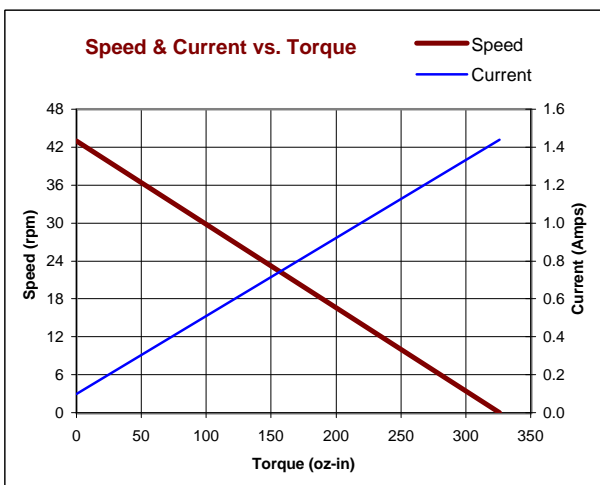
GM9413-3

Lo-Cog® DC Gearmotor

| Assembly Data | Symbol | Units | Value | |
|---------------------------------------|----------------|---|---------|------------|
| Reference Voltage | E | V | 12 | |
| No-Load Speed | S_{NL} | rpm (rad/s) | 43 | (4.5) |
| Continuous Torque (Max.) ¹ | T_C | oz-in (N-m) | 136 | (9.6E-01) |
| Peak Torque (Stall) ² | T_{PK} | oz-in (N-m) | 326 | (2.3E+00) |
| Weight | W_M | oz (g) | 15.6 | (442) |
| Motor Data | | | | |
| Torque Constant | K_T | oz-in/A (N-m/A) | 5.60 | (3.95E-02) |
| Back-EMF Constant | K_E | V/krpm (V/rad/s) | 4.14 | (3.95E-02) |
| Resistance | R_T | Ω | 8.33 | |
| Inductance | L | mH | 6.17 | |
| No-Load Current | I_{NL} | A | 0.10 | |
| Peak Current (Stall) ² | I_P | A | 1.44 | |
| Motor Constant | K_M | oz-in/ \sqrt{W} (N-m/ \sqrt{W}) | 1.94 | (1.37E-02) |
| Friction Torque | T_F | oz-in (N-m) | 0.50 | (3.5E-03) |
| Rotor Inertia | J_M | oz-in-s ² (kg-m ²) | 3.9E-04 | (2.8E-06) |
| Electrical Time Constant | τ_E | ms | 0.74 | |
| Mechanical Time Constant | τ_M | ms | 14.7 | |
| Viscous Damping | D | oz-in/krpm (N-m-s) | 0.011 | (7.6E-07) |
| Damping Constant | K_D | oz-in/krpm (N-m-s) | 2.8 | (1.9E-04) |
| Maximum Winding Temperature | θ_{MAX} | °F (°C) | 311 | (155) |
| Thermal Impedance | R_{TH} | °F/watt (°C/watt) | 66.4 | (19.1) |
| Thermal Time Constant | τ_{TH} | min | 11.1 | |
| Gearbox Data | | | | |
| Reduction Ratio | | | 65.5 | |
| Efficiency | | | 0.66 | |
| Maximum Allowable Torque | | oz-in (N-m) | 175 | (1.24) |
| Encoder Data | | | | |
| | | | | |
| | | | | |
| | | | | |

- Included Features**
- 2-Pole Stator
 - Ceramic Magnets
 - Heavy-Gauge Steel Housing
 - 7-Slot Armature
 - Silicon Steel Laminations
 - Stainless Steel Shaft
 - Copper-Graphite Brushes
 - Diamond Turned Commutator
 - Motor Sleeve Bearings
 - Output Sleeve Bearing
 - Standard Gears
- Customization Options**
- Alternate Winding
 - Sleeve or Ball Bearings
 - Modified Output Shaft
 - Custom Cable Assembly
 - Special Brushes
 - EMI/RFI Suppression
 - Alternate Gear Material
 - Special Lubricant
 - Optional Encoder
 - Fail-Safe Brake

1 - Specified at max. winding temperature at 25°C ambient without heat sink. 2 - Theoretical values supplied for reference only.



All values are nominal. Specifications subject to change without notice. Graphs are shown for reference only.

© 2001 Pittman.

| Housing Style - Rectangular | Part Number | ID Number | Features | Embeddable | Sensing Range (mm) | Output |
|---|--------------------|-----------|-------------------|------------|--------------------|---------------|
| <p>8.0 mm - Embeddable, Potted-In Cable</p> | Bi 5-Q08-AD4X/S34 | S4414550 | Weld-field Immune | • | 5 | 2-Wire DC |
| | Bi 5U-Q08-AN6X2 | S1608911 | Uprox | • | 5 | 3-Wire DC NPN |
| | Bi 5-Q08-AN6X2/S34 | | Weld-field Immune | • | 5 | |
| | Bi 7-Q08-AN6X2 | S1601620 | Ext. Range | • | 7 | |
| | Bi 8U-Q08-AN6X2 | S1662007 | Uprox | • | 8 | |
| | Bi 5U-Q08-AP6X2 | S1608901 | Uprox | • | 5 | 3-Wire DC PNP |
| | Bi 5-Q08-AP6X2/S34 | S1600800 | Weld-field Immune | • | 5 | |
| | Bi 7-Q08-AP6X2 | S1601600 | Ext. Range | • | 7 | |
| | Bi 8U-Q08-AP6X2 | S1662006 | Uprox | • | 8 | |
| | Bi 5-Q08-VN6X2 | S1600200 | Comp. Outputs | • | 5 | 4-Wire DC NPN |
| | Bi 5-Q08-VN6X2/S34 | | Weld-field Immune | • | 5 | |
| | Bi 7-Q08-VN6X2 | S1600920 | Ext. Range | • | 7 | |
| | Bi 5-Q08-VP6X2 | S1600100 | Comp. Outputs | • | 5 | 4-Wire DC PNP |
| | Bi 5-Q08-VP6X2/S34 | S1600101 | Weld-field Immune | • | 5 | |
| | Bi 7-Q08-VP6X2 | S1600900 | Ext. Range | • | 7 | |
| Bi 5-Q08-Y1X | S4054000 | | | • | 5 | 2-Wire NAMUR |

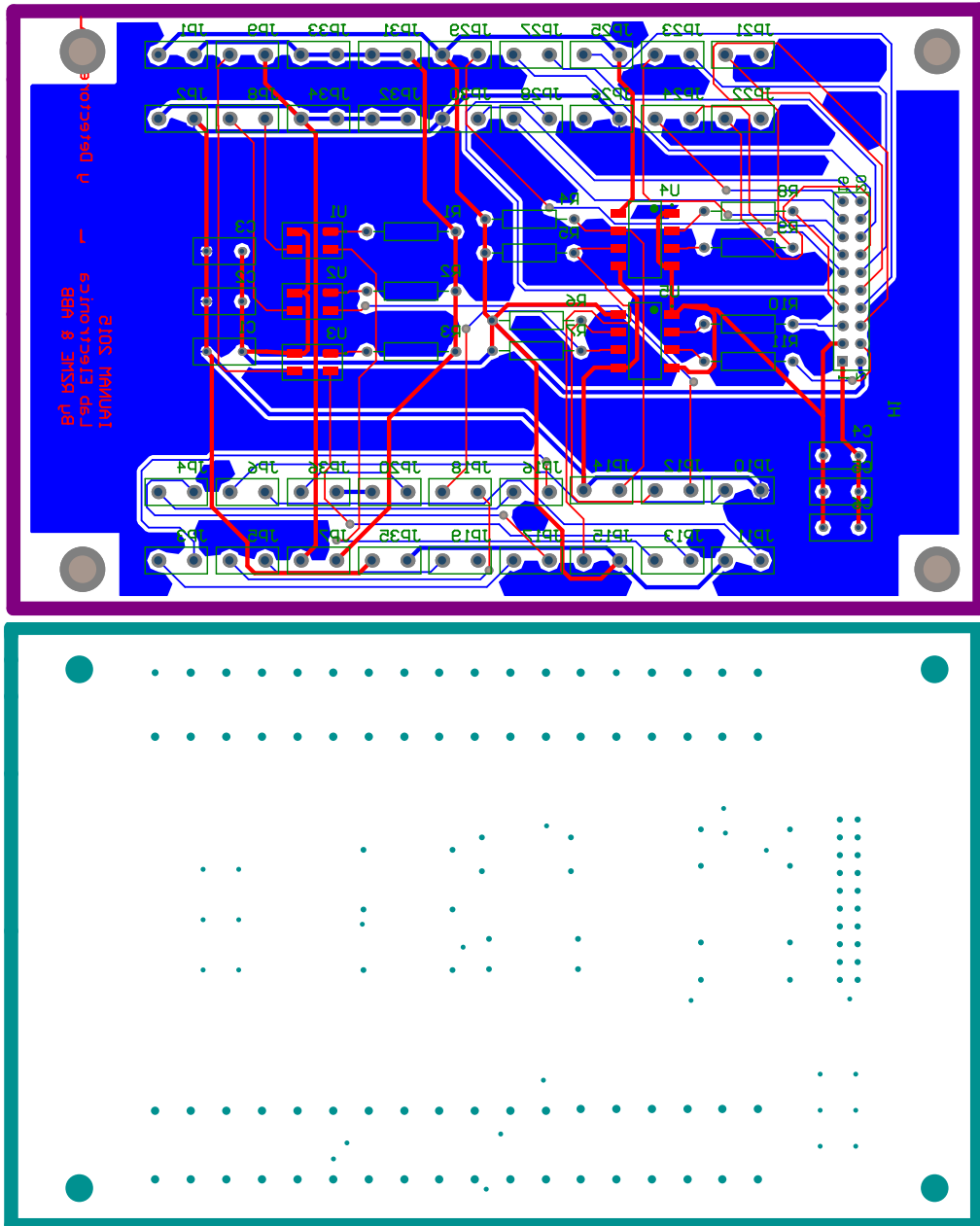
| Voltage | Switching Freq. (Hz) | Operating Current (mA) | Operating Temp. (°C) | Protection | Housing | Front Cap/Face | Power LED | Output LED | Mating Cord, Cable Length/Jacket | Wiring Diagram # | Wiring Diagrams |
|-----------|----------------------|------------------------|----------------------|------------|---------|----------------|-----------|------------|----------------------------------|------------------|-----------------|
| 10-65 VDC | 50 | ≤100 | -25 to +70 | IP 67 | Zinc | PA 12 | N/A | YE | 2M/PVC | 1 | Diagram 1 |
| | | | | | | | | | | | |
| 10-30 VDC | 1000 | ≤200 | -30 to +85 | IP 67 | Zinc | PA 12 | GN | YE | 2M/PVC | 2 | Diagram 2 |
| | 300 | ≤100 | -25 to +70 | IP 67 | Zinc | PA 12 | GN | YE | 2M/PVC | 2 | |
| | 500 | ≤200 | -25 to +70 | IP 67 | Zinc | PA 12 | GN | YE | 2M/PVC | 2 | |
| | 1000 | ≤200 | -30 to +85 | IP 67 | Zinc | PA 12 | GN | YE | 2M/PVC | 2 | |
| 10-30 VDC | 1000 | ≤200 | -30 to +85 | IP 67 | Zinc | PA 12 | GN | YE | 2M/PVC | 3 | Diagram 3 |
| | 300 | ≤100 | -25 to +70 | IP 67 | Zinc | PA 12 | GN | YE | 2M/PVC | 3 | |
| | 500 | ≤200 | -25 to +70 | IP 67 | Zinc | PA 12 | GN | YE | 2M/PVC | 3 | |
| | 1000 | ≤200 | -30 to +85 | IP 67 | Zinc | PA 12 | GN | YE | 2M/PVC | 3 | |
| 10-30 VDC | 1000 | ≤200 | -25 to +70 | IP 67 | Zinc | PA 12 | GN | YE | 2M/PVC | 4 | Diagram 4 |
| | 300 | ≤100 | -25 to +70 | IP 67 | Zinc | PA 12 | GN | YE | 2M/PVC | 4 | |
| | 500 | ≤200 | -25 to +70 | IP 67 | Zinc | PA 12 | GN | YE | 2M/PVC | 4 | |
| 10-30 VDC | 1000 | ≤200 | -25 to +70 | IP 67 | Zinc | PA 12 | GN | YE | 2M/PVC | 5 | Diagram 5 |
| | 300 | ≤100 | -25 to +70 | IP 67 | Zinc | PA 12 | GN | YE | 2M/PVC | 5 | |
| | 500 | ≤200 | -25 to +70 | IP 67 | Zinc | PA 12 | GN | YE | 2M/PVC | 5 | |
| 5-30 VDC | 1000 | Remote | -25 to +70 | IP 67 | Zinc | PA 12 | N/A | YE | 2M/PVC | 6 | Diagram 6 |
| | | | | | | | | | | | |

CAPÍTULO

F

PCB del módulo electrónico

Apéndice F. PCB del módulo electrónico



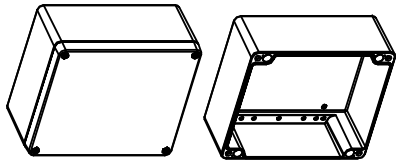
CAPÍTULO

G

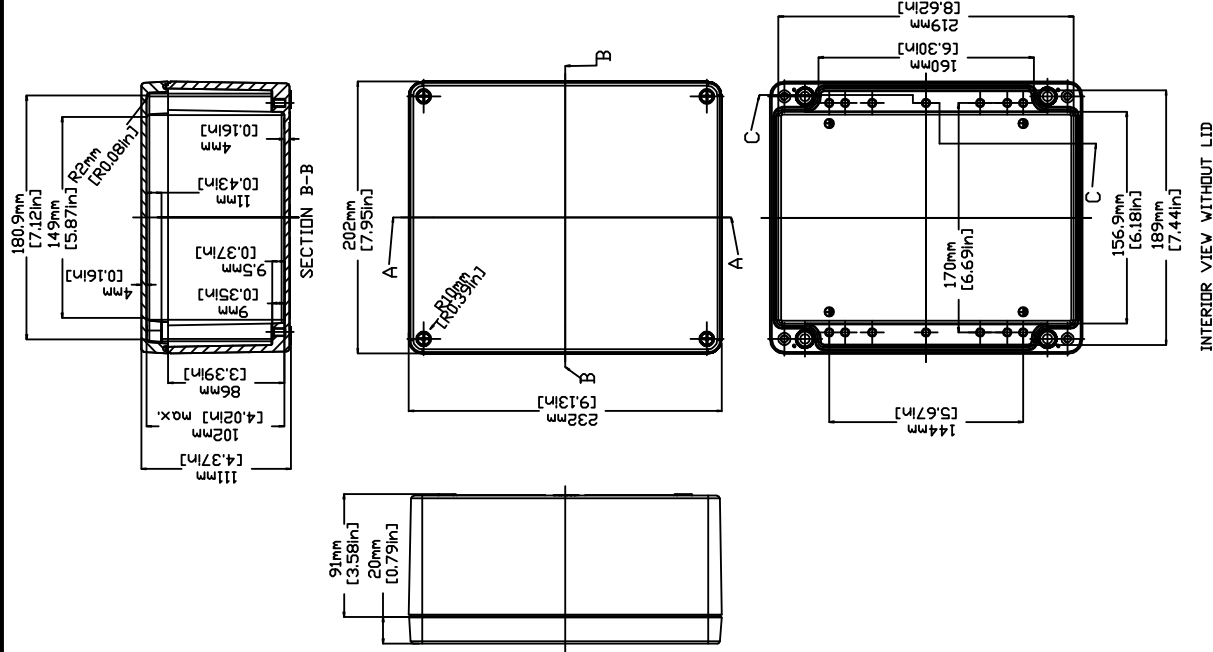
Planos mecánicos

REVISIONS FOR DWG # 3EN-3IS-AL-AL2023-11

| REV | CH.NO. | DESCRIPTION | DATE | REVISED BY |
|-----|--------|-----------------|----------|------------|
| - | - | ORIGINAL DESIGN | 08/01/11 | ROLAND G |



1:2



| | | | | |
|---------------------------|------------|-----------------------|------------|--------------|
| TITLE: ALUMINUM ENCLOSURE | | DRAWING NO.: | | REV.: |
| AL 2023-11 (150-014) | | 3EN-3IS-AL-AL 2023_11 | | - |
| DRAWN BY: | 08/01/2011 | CHECKED BY: | 08/01/2011 | RELEASED BY: |
| ROLAND GYORGYAL | JEFF ROYE | ROLAND G. | | |
| SCALE: - | | SHEET: 1 | | OF: 1 |

This Drawing is the property of **Altech Corp.**
 AND IS A CONFIDENTIAL DISCLOSURE LENT WITHOUT CONSIDERATION OTHER THAN THE BORROWER'S AGREEMENT THAT IT SHALL NOT BE REPRODUCED, COPIED, LENT, OR DISPOSED OF DIRECTLY OR INDIRECTLY NOR USED FOR ANY PURPOSE OTHER THAN THAT FOR WHICH IT IS SPECIFICALLY FURNISHED.
 35 Royal Road
 Farnington NJ 08822-6000
 (908) 808-9400 • Fax (908) 808-9490

| | |
|-------------------------------------|---------|
| UNLESS OTHERWISE SPECIFIED | |
| • ALL DIMENSIONS ARE IN MILLIMETERS | |
| • THIRD ANGLE PROJECTION | |
| • TOLERANCE | |
| XX | ≡ XX mm |
| XX.X | ≡ XX mm |
| XX.XX | ≡ XX mm |
| ALL | ≡ XX mm |

CAPÍTULO

III

Datos y esquema de conexiones

| DB-25 Tablero de Conexiones PUMA | | | 3M RABBIT | CS100 (ELCO) | Interfaz CS100 |
|----------------------------------|-----------------|----------------|-----------|--------------|----------------|
| pin | color | señal | pin | Pin | Pin |
| 22 | negro | clear Y | 18 | U | 7J1 |
| 17 | amarillo | Close servo | 8 | F | 9J1 |
| 9 | café | Ext Cntl | 17 | T | HJ1 |
| 8 | azul | Up Z | 15 | e | 13J1 |
| 7 | naranja | clear Z | 13 | H | 15J1 |
| 19 | violeta | Up X | 12 | a | 19J1 |
| 4 | azul-café | overload Y | 7 | C | 4JP2 |
| 15 | violeta-café | loop closed | 4 | B | 8JP2 |
| 1 | transparente | Tierra | 1 | Tierra CS100 | |
| 23 | Café-blanco | clear X | 20 | N | 5J1 |
| 20 | rojo-blanco | Dwn Y | 14 | J | 8J1 |
| 18 | violeta-blanco | Up Y | 10 | S | 10J1 |
| 5 | azul-blanco | integra | 9 | V | 12J1 |
| 21 | Verde-blanco | reset | 16 | Z | 14J1 |
| 10 | naranja-blanco | Dwn Z | 19 | c | 16J1 |
| 6 | amarillo-blanco | Dwn X | 11 | b | 21J1 |
| 3 | amarillo-café | overload X | 5 | E | 1JP2 |
| 16 | naranja-café | overload Z | 6 | L | 5JP2 |
| 2 | gris-café | VDC +5 cs100 | 3 | D | 10JP2 |
| 14 | verde-café | Vcc +12 Rabbit | 2 | A | 2J1 |

Tabla. *Conexiones Cable Puma-CS100.*

| Pin CPC 3 (Canon) | Señal | Color de cable |
|-------------------|--------------------|----------------|
| 3 | Lampara3 Helio | Amarillo |
| 2 | Lampara2 Hidrógeno | Amarillo |
| 1 | Lampara1 Neón | Amarillo |
| 6 | GND Lámparas | Rojo |
| 4,7 | VCC +12 V | - |
| 5,8 | GND de VCC | - |

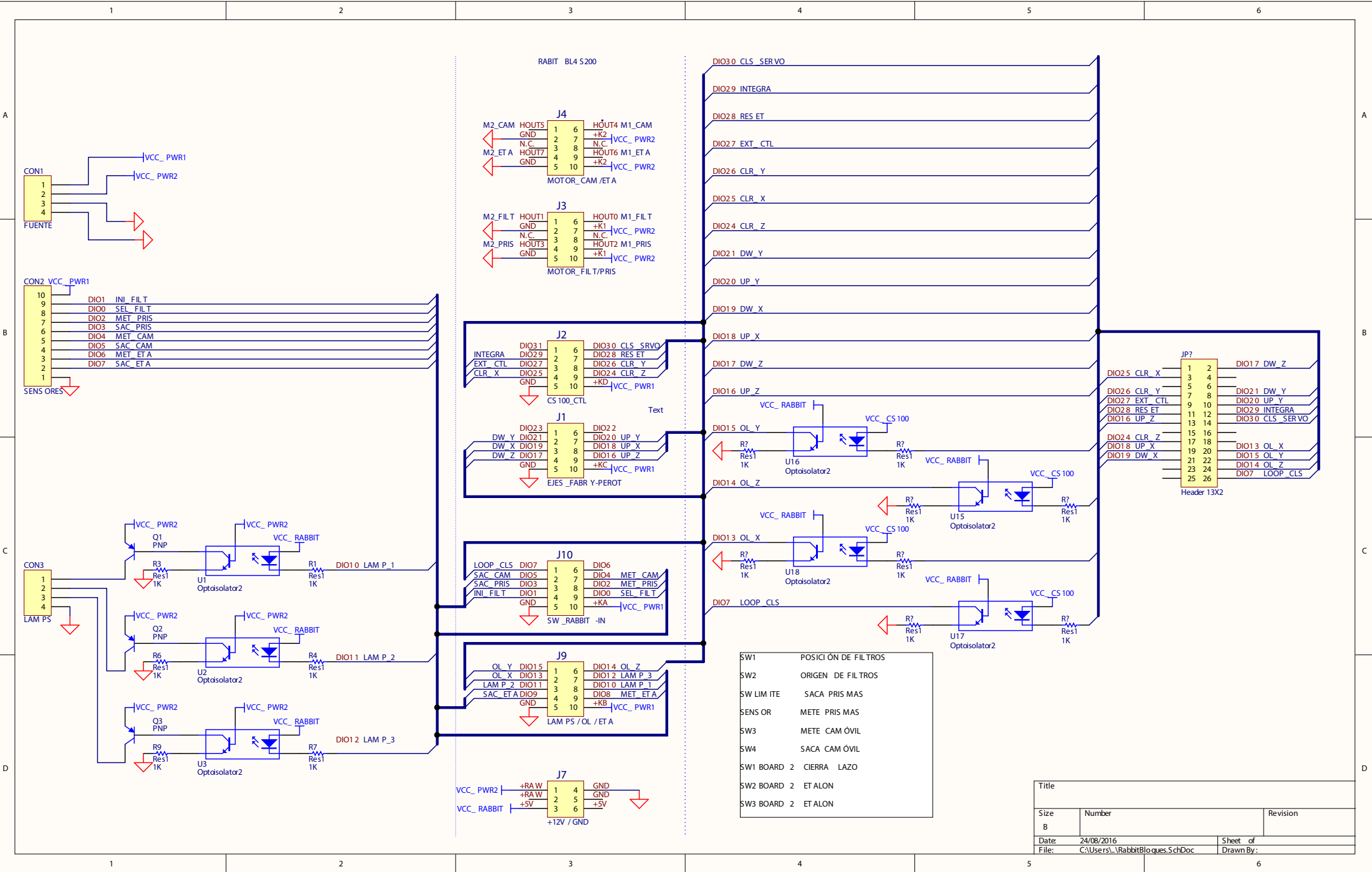
Tabla Lámparas y VCC y GND. Conector 3 Tipo CPC 9 Pines.

| Pin | Señal | Número de Cable |
|-----|-----------------------|-----------------|
| 6 | Origen Filtros | Cable 0 |
| 9 | Posicion Filtros | Cable 1 |
| 1 | Etalon Adentro | Cable 3 |
| 2 | Etalon Afuera | Cable 4 |
| 8 | Brazo Prismas Adentro | Cable 5 |
| 7 | Brazo Prismas Afuera | Cable 6 |
| 4 | Camovil Adentro | Cable 7 |
| 5 | Camovil Afuera | Cable 8 |
| 3 | GND de señal | Cable 8 |

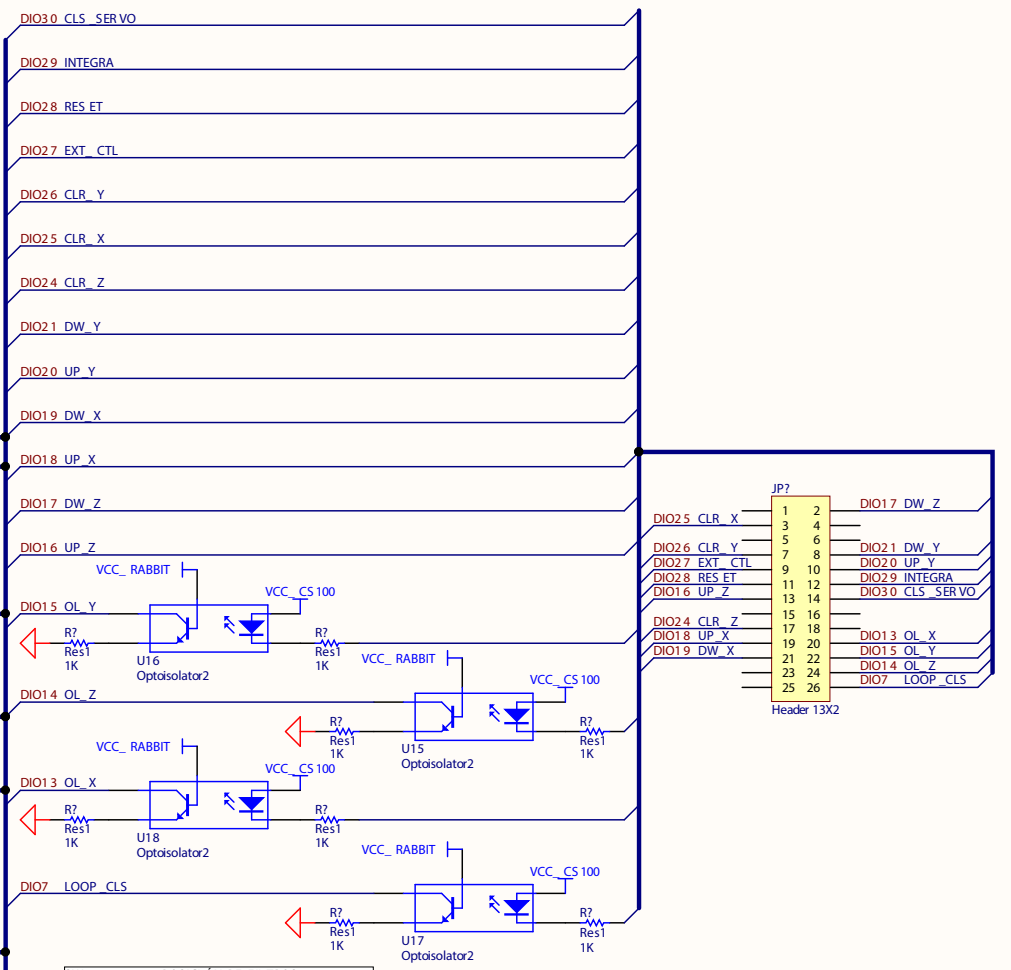
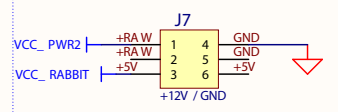
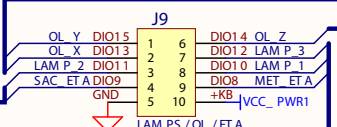
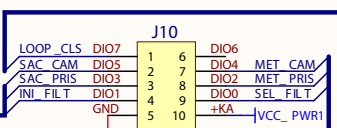
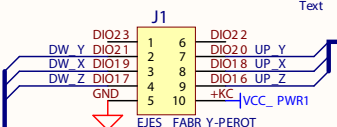
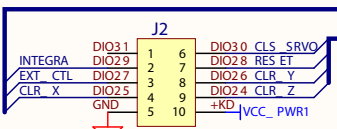
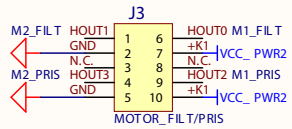
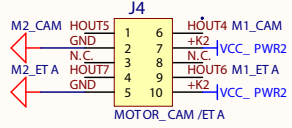
Tabla Sensores de proximidad e interruptores límite. Conector 2 Tipo CPC 9 Pines.

| Pin | Señal | Color de cable |
|-----|--|------------------------|
| 7,4 | Motor Brazo Prismas | Azul oscuro y amarillo |
| 8,9 | Motor Rueda Filtros | Negro y verde |
| 1,2 | Motor Posicionador Elementos Ópticos | Negro y rojo |
| 5,6 | Motor Unidad Movil Sistema Calibración | Azul claro y amarillo |

Tabla Motores. Conector 1 Tipo CPC 9 Pines.



RABIT BL4 S200



- SW1 POSICI ÓN DE FILTROS
- SW2 ORIGEN DE FILTROS
- SW LIM ITE SACA PRIS MAS
- SENS OR METE PRIS MAS
- SW3 METE CAM ÓVIL
- SW4 SACA CAM ÓVIL
- SW1 BOARD 2 CIERRA LAZO
- SW2 BOARD 2 ET ALON
- SW3 BOARD 2 ET ALON

| | | |
|--------|------------------------------------|-----------|
| Title | | |
| Size B | Number | Revision |
| Date | 24/08/2016 | Sheet of |
| File: | C:\Users\... \RabbitBloques.SchDoc | Drawn By: |

CAPÍTULO

I

Termistor NTC

The NTC Thermistors

NTC Thermistors is a negative temperature coefficient resistor that significantly reduces its resistance value as the heat/ ambient temperature rises. Thermistors is sintered in high-temperature (1200 °C to 1500 °C), and manufactured in various shapes. It's comprised of 2 to 4 kinds of metal oxides: iron, nickel, cobalt, manganese and copper.

Features

- Temperature Coefficient of Resistance is negative, and it's extremely large (-2.8 to -5.1 [%/°C]).
- Various shapes, especially compact size components are available.
- Selection of resistance vale is comparatively free, it's available from several tens Ω to several hundred k Ω.

Recommended Applications

- For temperature measurement or temperature detection : Thermometer, temperature controller
- For temperature compensation : Transistor, transistor circuit, quarts oscillation circuit, and measuring instruments

Physical Characteristics of NTC Thermistors

Thermistor is a resistor sensitive to temperature that is utilizing the characteristic of metal oxide semiconductor having large temperature coefficient. And its temperature dependency of resistance value is indicated by the following equation :

$$R = R_0 \exp \left[B \left(\frac{1}{T} - \frac{1}{T_0} \right) \right] \dots\dots\dots (1)$$

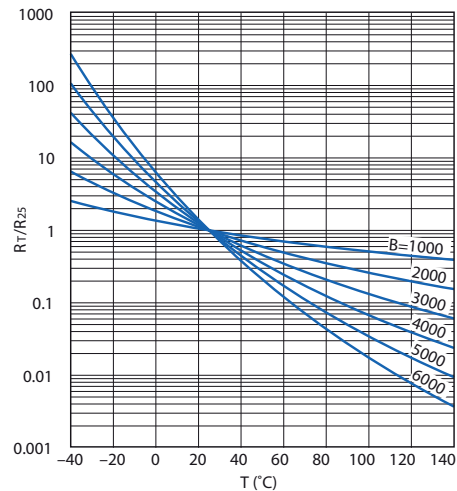
- T₀ : Standard Temperature 298.15 K(25 °C)
- R₀ : Resistance at T₀ [K]
- B : Thermistor Constant [K]

Temperature coeffi cient (a) in general meaning is indicated as follows :

$$a = - \frac{B}{T^2} \dots\dots\dots (2)$$

Since the change by temperature is considerably large, a is not appropriate as a constant. Therefore, B value (constant) is generally used as a coeffi cient of thermistors.

Fig. 1



Major Characteristics of NTC Thermistors

The relation between resistance and temperature of a thermistor is linear as shown in Fig. 2. The resistance value is shown in vertical direction in a logarithmic scale and reciprocal of absolute temperature (adding 273.15 to centigrade) is shown in horizontal direction.

The B value (constant) determines the gradient of these straight lines. The B value (constant) is calculated by using following equation.

$$B = \frac{knR_1 - knR_2}{\frac{1}{T_1} - \frac{1}{T_2}} \dots\dots\dots (3)$$

- R₁: Resistance at T₁ K
- R₂: Resistance at T₂ K

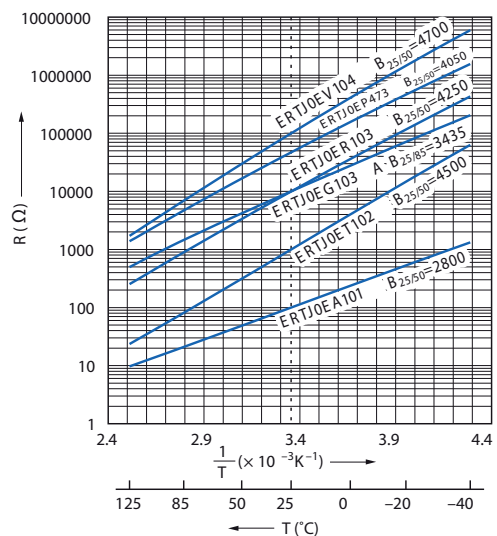
When you calculate this equation, you'll find that B value is not exactly constant. The resistance is expressed by the following equation :

$$R = AT^{-C} \exp D/T \dots\dots\dots (4)$$

In (4), C is a small positive or negative constant and quite negligible except for use in precision temperature-measuring device, therefore, the B value can be considered as constant number.

In Fig. 1, the relation between the resistance ratio R_T/R₂₅ (R₂₅ : Resistance at 25 °C, R_T : Resistance at T °C) and B Value is shown with T °C, in the horizontal direction.

Fig. 2



Bibliografía

- Bearings, Aisia (2016). Coefficient friction. <http://www.asiabearings.com/Linear-Bearings/>. Accessed:2016-08-01.
- Bernal, Abel and Martínez, Luis A. (2010). *Sistema de Control Electrónico del Espectrógrafo Integral de Campo PUMA Versión 2*. Publicaciones Técnicas del Instituto de Astronomía, Universidad Nacional Autónoma de México.
- Bershady, Matthew A. (2010). *3D spectroscopic instrumentation*. 3D Spectroscopy in Astronomy, Cambridge University, Department of Astronomy, [astro-ph.IM]. University of Wisconsin. Ph.D. Dissertation.
- Bland, J. and Tully, R. B. (1989). The Hawaii imaging Fabry-Perot interferometer (HIFI). *Astronomical Journal*, 98:723–735.
- Born, Max and Wolf, Emil (2005). *Principles of Optics: Electromagnetic Theory of Propagation, Interference and Diffraction of Light*, volume 1. Cambridge University Press, Cambridge United Kingdom, UK, seventh edition. SBN 0 521 642221.
- Courtès, G., Comte, G., and Marcelin, M. (1995). 3d optical spectroscopic methods in astronomy. *ASP Conference*, Series 17.
- Departamento de Astronomia Observacional, IA-UNAM (2016). Noches de observación. <http://catt.astro.unam.mx/acervo>. Accessed:2016-10-09.
- Digi-Semiconductor (2013a). *BLAS200 C-Programmable Single-Board Computer with Networking User's Manual*. Digi International Inc.
- Digi-Semiconductor (2013b). *Dynamic C Function Reference Manual Rabbit 4000, 5000, and 600 Microprocessors*. Digi International Inc.
- Digi-Semiconductor (2013c). *Dynamic C IO User's Manual Rabbit 4000, 5000 and 6000 Microprocessors*. Digi International Inc.
- Digi-Semiconductor (2013d). *Rabbit RIO User's Manual*. Digi International Inc.
- Digi-Semiconductor (2013e). *RabbitCore RCM4300 C-Programmable Analog Core Module with microSD Card Storage and Ethernet User's Manual*. Digi International Inc.
- Digi-Semiconductor (2013f). *User's Manual An Introduction to TCP IP For Embedded System Designers*. Digi International Inc.
- Digi-Semiconductor (2013g). *User's Manual Dynamic C TCP IP Volume 1*. Digi International Inc.

- Duxbury, Geoffrey (2012). 3d spectroscopy in astronomy, edited by e. mediavilla, s. arribas, m. roth, j. cepa-nogué and f. cepa-nogué: Scope: guide. level: researcher. *Contemporary Physics*, 53(3):267–268.
- Fitzgerald, Arthur Eugene, Higginbotham, David E., and Grabel, Arvin (1981). *Basic electrical engineering*, volume 1. McGraw-Hill Series in Electrical Engineering, Auckland: McGraw-Hill, fifth edition. ISBN 10: 0070662835 ISBN 13: 9780070662834.
- Fuentes Carrera, Isaura Luisa (2001). *Dinámica de galaxias interactuantes*, volume 1. Facultad de ciencias, UNAM, Universidad Nacional Autónoma de México. Tesis de Licenciatura.
- Hecht, Eugene and Zajac, Alfred (2002). *Optics: pearson new international edition*, volume 1. Addison Wesley, 1301 Sansome St., San Francisco, fourth edition. ISBN 0 321 188878 0.
- Hyder, Kamal and Perrin, Bob (2005). *Embedded systems design using the Rabbit 3000 microprocessor: interfacing, networking, and application development*, volume 1. Elsevier, Linacre House, Jordan Hill, Oxford OX2 8DP, UK, first edition. ISBN: 0-7506-7872-0.
- IA-UNAM (2015). Oan-spm. <http://haro.astrossp.unam.mx/indexspm.html>. Accessed:2015-11-02.
- Kasunic, Keith J. (2015). *Optomechanical Systems Engineering*, volume 1. John Wiley & Sons, Inc., Hoboken, New Jersey, first edition. ISBN 978-1-118-80932-7.
- Langarica, Rosalía and Tinoco, Silvio (2005). *El Espectrógrafo Integral de Campo EIC-1 PUMA*, volume 1. Publicaciones Técnicas del Instituto de Astronomía, Universidad Nacional Autónoma de México.
- Le Coarer, Etienne (1992). *Application de l'interféromètre de Pérot-Fabry à l'étude à grand champ de la Galaxie et du Petit Nuage de Magellan. Développement d'un nouvel instrument: Pythéas*, volume 1. Université Paris-Diderot - Paris VII, 1992. Français. <tel-00725457>, Instrumentation et méthodes pour l'astrophysique [astro-ph.IM]. Ph.D. Dissertation.
- Le Coarer, E., Rosado, M., Georgelin, Y., Viale, A., and Goldes, G. (1993). H-alpha survey of the Small Magellanic Cloud. *Astronomy and Astrophysics*, 280:365–372.
- Macleod, H. Angus (2001). *Thin-film optical filters*, volume 1. Institute of Physics Publishing Bristol and Philadelphia, Thin Film Center Inc. Tucson, Arizona, third edition. ISBN 0 7503 0688 2.
- Martínez, Luis A., Langarica, Rosalía, Bernal, Abel, Rosado, Margarita, and Arias, Lorena (2007). *Espectrógrafo Integral de Campo PUMA Modo Interferometro Fabry-Pérot de Barrido Manual de Usuario*. Publicaciones Técnicas del Instituto de Astronomía, Universidad Nacional Autónoma de México.
- Moiseev, AV, Afanasiev, VL, and Dodonov, SN (2002). Fabry-perot observations at the 6m telescope. In *Galaxies: the Third Dimension*, volume 282, page 443. 2 pages, 1 EPS figure, to appear in Astronomical Society of the Pacific Conf. Ser. "Galaxies: The Third Dimension", Eds. M. Rosado, L. Binette, and L. Arias.

- Mulligan, Joseph F (1998). Who were fabry and pérot? *American Journal of Physics*, 66(9):797–802.
- Petersen, Daniel P and Middleton, David (1962). Sampling and reconstruction of wave-number-limited functions in n-dimensional euclidean spaces. *Information and control*, 5(4):279–323.
- Queensgate-Instruments (1981). *CS100 Stabilization System Instruction Manual*, volume 1. Electronic Version, Franklin Road, London.
- Rosado, M., Langarica, R., Bernal, A., Cobos, F., Garfias, F., Gutiérrez, L., Tejada, C., and Tinoco, S. (1995). The unam scanning fabry-perot interferometer (puma) for the study of interstellar medium. *The fifth Mexico-Texas conference on astrophysics: gaseous nebulae and star formation*, volume 3:pages 263–266.
- Sánchez Cruces, Mónica (2013). *Cinematica de superburbujas y supershells de la galaxia NGC 1569*, volume 1. Programa de Posgrado de Astrofísica, Instituto de Astronomía, Universidad Nacional Autónoma de México. Tesis de Maestría.
- Schmidt, Jason Daniel (2010). *Numerical simulation of optical wave propagation with examples in MATLAB*, volume 1. Society of Photo-Optical Instrumentation Engineers (SPIE), Bellingham, Washington 98227-0010 USA, first edition. ISBN 978-0-8194-8326-3.
- Sears, Francis Weston, Ford, A. Lewis, and Freedman, Roger A. (2008). *University Physics with Modern Physics*, volume 2. Addison-Wesley, Bellingham, Washington 98227-0010 USA, twelfth edition. ISBN 9780321501219.
- Smith, Nicholas (2013). The engineer s guide to high-quality pcb design. *Electronic Design*, pages 1–5.
- University of Wisconsin-Madison, Department of Astronomy (2016). Integral field spectroscopy. <http://www.astro.wisc.edu/our-science/research-areas/instrumentation/integral-field-spectroscopy/>. Accessed:2016-17-05.
- Vaughan, M. (1989). *The Fabry-Pérot interferometer: history, theory, practice and applications*, volume Series in Optics and Optoelectronics. CRC press, Vaughan, DRA, Malvern, UK., first edition. ISBN-10: 0852741383 ISBN-13: 978-0852741382.
- Vázquez Pérez, Celia (2015). *Cinematica de galaxias aisladas*, volume 1. Programa de Posgrado de Astrofísica, Instituto de Astronomía, Universidad Nacional Autónoma de México. Tesis de Maestría.
- Yourdon, Edward (1993). *Análisis estructurado moderno*, volume 1. Prentice-Hall Hispanoamericana, S.A., Enrique Jacob 20, Col. Conde Edo. de México, first edition. ISBN: 968-880-303-0.

Índice de cuadros

| | | |
|------|--|----|
| 3.1 | Longitudes de H alpha y [OIII] | 31 |
| 3.2 | Características de longitud de onda de los filtros | 33 |
| 3.3 | Características de las lámparas de calibración | 34 |
| 3.4 | Resoluciones del controlador | 37 |
| 3.5 | Paralelismo y espaciamentos | 38 |
| 3.6 | Computadora Octagon y la estación de trabajo | 41 |
| 4.1 | Diagrama de Bloques del hardware | 54 |
| 5.1 | Diagrama OSI | 60 |
| 5.2 | Diagrama de contexto | 64 |
| 5.3 | Lista de eventos | 65 |
| 5.4 | Diagrama de flujo de datos | 68 |
| 5.5 | Diagrama de transición de estados | 71 |
| 5.6 | Diccionario de datos | 72 |
| 5.7 | Especificación de procesos del programa principal | 76 |
| 5.8 | Especificación del proceso selección de filtro de origen | 79 |
| 5.9 | Análisis del movimiento de la rueda de filtros | 80 |
| 5.10 | Especificación del proceso de la rueda de filtros | 81 |
| 5.11 | Especificación del proceso de los carros móviles | 85 |

| | | |
|------|---|-----|
| 5.12 | Especificación del proceso de los ejes del Fabry-Pérot | 86 |
| 5.13 | Especificación del proceso de selección de lámpara | 87 |
| 5.14 | Especificación del proceso de activación de lámpara | 88 |
| 5.15 | Especificación del proceso del reinicio del controlador CS100 | 89 |
| 5.16 | Especificación del proceso de la presentación del estado del sistema | 90 |
| 6.1 | Motor que representa a los carros móviles: | 103 |
| 6.2 | Motor del brazo de prismas | 103 |
| 6.3 | Motor de los carros móviles, para el <i>étalon</i> , la unidad móvil de calibración y la rueda de filtros. | 104 |
| 6.4 | Motor que representa la carga que tienen los subsistemas del instrumento PUMA: | 104 |
| 6.5 | Resultados de los cálculos teóricos para el análisis mecánico del movi- miento de los carros móviles. | 114 |
| 7.1 | Flujo ideal del diseño de un PCB | 116 |
| 7.2 | Diagrama de bloques del módulo | 118 |
| 7.3 | Descripción de las conexiones del PCB | 125 |
| 8.1 | Observaciones realizadas con el instrumento PUMA en el telescopio de 2.1 m . | 138 |
| 8.2 | Costos de componentes del nuevo sistema de ontról | 139 |
| 9.1 | Metodología | 142 |
| 9.2 | Comunicaciones del sistema de control con el nuevo controlador CS100 | 147 |
| D.1 | Mediciones de las simulaciones en laboratorio para los motores | 177 |
| D.2 | Mediciones de las simulaciones en laboratorio para los motores | 178 |
| D.3 | Mediciones de las simulaciones en laboratorio para los motores | 179 |

Índice de figuras

| | | |
|------|--|----|
| 1.1 | Observatorio Astronómico Nacional | 4 |
| 1.2 | Telescopio de 2.1 metros | 5 |
| 1.3 | Experimento de Airy en 1831 | 7 |
| 1.4 | Modelo del interferómetro de Fabry y Pérot | 8 |
| 1.5 | Anillos de interferencia | 9 |
| 1.6 | Ondas reflejadas y transmitidas | 11 |
| 1.7 | Intensidad de transmisión | 15 |
| 1.8 | Intensidad de reflectancia | 15 |
| 1.9 | Fineza v reflectividad | 17 |
| 1.10 | Contraste v reflectividad | 17 |
| 1.11 | Interferómetro Fabry-Pérot con una fuente extendida | 20 |
| 1.12 | Superficies de corrección de fase | 21 |
| 1.13 | Análisis de los interferogramas | 22 |
| 2.1 | Instituto de Astronomía sede en Ensenada | 24 |
| 2.2 | Microcontrolador Octagon utilizado en el primer sistema de control del instrumento PUMA. | 26 |
| 3.1 | Sistema optomecánico del instrumento PUMA | 29 |
| 3.2 | Interferómetro Fabry-Pérot serie ET | 30 |
| 3.3 | Transductores piezoeléctricos | 31 |

| | | |
|------|--|-----|
| 3.4 | Rueda de ocho filtros | 32 |
| 3.5 | Sistema de calibración | 33 |
| 3.6 | Lámparas | 34 |
| 3.7 | Brazo de prismas | 35 |
| 3.8 | Cámara | 36 |
| 3.9 | Puente de capacitancia | 37 |
| 3.10 | Diagrama de bloques de la computadora Octagon | 39 |
| 3.11 | Ventanas al iniciar la GUI del PUMA | 42 |
| 4.1 | Microcontrolador Rabbit BL4S200 | 44 |
| 4.2 | Diagrama de bloques del microcontrolador Rabbit BL4S200 | 46 |
| 4.3 | Conexiones del sistema de control | 49 |
| 4.4 | Entradas y salidas digitales | 50 |
| 4.5 | Modulación de ancho de pulso | 52 |
| 4.6 | Puente H | 53 |
| 4.7 | Distribución de conexiones del sistema | 55 |
| 5.1 | Movimiento de giro la rueda de filtros | 79 |
| 5.2 | Indexaciones de la rueda de filtros | 80 |
| 5.3 | Movimiento del posicionador de elementos ópticos | 83 |
| 5.4 | Movimiento de la unidad de calibración | 84 |
| 5.5 | Movimiento del brazo de prismas | 84 |
| 6.1 | Interfaz de usuario | 96 |
| 6.2 | Prueba de conexión de red 1 | 99 |
| 6.3 | Prueba de conexión de red 2 | 100 |
| 6.4 | Prueba de conexión de red 3 | 101 |
| 6.5 | Prueba de voltaje | 102 |
| 6.6 | Prueba de corriente | 102 |
| 6.7 | Gráfica: ciclo de trabajo frente a revoluciones por minuto | 105 |
| 6.8 | Gráfica: frecuencia frente a variaciones de corriente | 105 |
| 6.9 | Gráfica: potencia frente a corriente | 106 |
| 6.10 | Gráfica: potencia frente a frecuencia | 106 |
| 6.11 | Gráfica: velocidad frente a velocidad angular | 107 |
| 6.12 | Gráfica: torque frente a frecuencia | 107 |

| | | |
|------|---|-----|
| 6.13 | Gráfica: torque frente a revoluciones por minuto | 108 |
| 6.14 | Gráfica: voltaje frente a corriente | 108 |
| 6.15 | Gráfica: voltaje frente a frecuencia | 109 |
| 6.16 | Análisis dimensional del interferómetro | 109 |
| 6.17 | Análisis dimensional del brazo de prismas | 111 |
| 6.18 | Análisis dimensional de la rueda de filtros | 112 |
| 6.19 | Análisis dimensional de la unidad de calibración móvil | 113 |
| | | |
| 7.1 | Diagrama electrónico para el optoacoplador HCPL2630SV | 119 |
| 7.2 | Diagrama electrónico para el optotransistor TLP127 | 120 |
| 7.3 | Diagrama electrónico para la etapa de filtrado | 121 |
| 7.4 | Diseño esquemático de las conexiones de la tarjeta de control electrónico | 122 |
| 7.5 | Especificación de las conexiones del PCB | 124 |
| 7.6 | Modelo 3D del módulo externo de control electrónico. | 126 |
| 7.7 | PCB Fabricado | 126 |
| | | |
| 8.1 | Modelo 3D para la integración del nuevo sistema de control | 129 |
| 8.2 | Desarme del nuevo sistema de control | 131 |
| 8.3 | Montaje en el instrumento PUMA | 133 |
| 8.4 | Par de galaxias espirales en interacción NGC 2207 y IC 2163 | 134 |
| 8.5 | Archivos cie | 135 |
| 8.6 | Archivos cio | 136 |
| 8.7 | Sistema de control | 137 |
| 8.8 | Sistema de control instalado en el instrumento PUMA | 137 |
| 8.9 | Sistema de control instalado en el telescopio de 2.1 m | 140 |

EPÍLOGO

La culminación de este proyecto que en todos los aspectos lo consideraré como un esfuerzo intelectual en muchas áreas del conocimiento de la física e ingeniería; nunca me dejará. El conocimiento adquirido durante todo el camino de desarrollo ha sido invaluable para mi crecimiento profesional, en el ámbito de la investigación.

Del proyecto realizado descrito en esta tesis, se espera que los astrónomos puedan llevar a cabo sus investigaciones y que del instrumento con el nuevo sistema de control ya instalado se deriven más artículos y más publicaciones, en beneficio de la curiosidad científica y búsqueda de información. Que el esfuerzo y dedicación de este trabajo se exprese en la utilidad y eficiencia en las observaciones que realizan los astrónomos de la Universidad Nacional Autónoma de México y de investigadores nacionales e internacionales que hagan uso del instrumento PUMA.

Cómo dice (Vaughan, 1989) *"la justificación final de cualquier instrumento debe estar en la información que proporciona y en el nuevo conocimiento que conduce, siendo éste el último criterio para cualquier instrumento, para su desempeño en la práctica de las tareas científicas..."*, esperando que el sistema de control sirva a la aplicación, durante muchos años más.

