



**UNIVERSIDAD POLITÉCNICA SALESIANA
SEDE QUITO
CARRERA DE INGENIERÍA ELECTRÓNICA**

**DESARROLLO DE UN TÚNEL DE LEVITACIÓN NEUMÁTICA UTILIZANDO
UN SISTEMA EMBEBIDO PARA LA OPTIMIZACIÓN DE SISTEMAS DE
MANDO Y CONTROL**

Trabajo de titulación previo a la obtención del
Título de Ingeniero Electrónico

AUTOR: Gustavo Alejandro Reyes Navarrete

TUTOR: Andrés Sebastián Calero Calero

Quito-Ecuador

2022

CERTIFICADO DE RESPONSABILIDAD Y AUTORÍA DEL TRABAJO DE TITULACIÓN

Yo, Gustavo Alejandro Reyes Navarrete con documento de identificación N°. 1725328650 manifiesto que:

Soy el autor y responsable del presente trabajo; y, autorizo a que sin fines de lucro la Universidad Politécnica Salesiana pueda usar, difundir, reproducir o publicar de manera total o parcial el presente trabajo de titulación.

Quito, 14 de septiembre del año 2022

Atentamente,



Gustavo Alejandro Reyes Navarrete

1725328650

**CERTIFICADO DE CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO DE
TITULACIÓN A LA UNIVERSIDAD POLITÉCNICA SALESIANA**

Yo, Gustavo Alejandro Reyes Navarrete con documento de identificación N°. 1725328650, expreso mi voluntad y por medio del presente documento cedo a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que soy autor del Proyecto Técnico: “Desarrollo de un túnel de levitación neumática utilizando un sistema embebido para la optimización de sistemas de mando y control”, el cual ha sido desarrollado para optar por el título de: Ingeniero Electrónico, en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En concordancia con lo manifestado, suscribo este documento en el momento que hago la entrega del trabajo final en formato digital a la Biblioteca de la Universidad Politécnica Salesiana.

Quito, 14 de septiembre del año 2022

Atentamente,



Gustavo Alejandro Reyes Navarrete

1725328650

CERTIFICADO DE DIRECCIÓN DEL TRABAJO DE TITULACIÓN

Yo, Andrés Sebastián Calero Calero con documento de identificación N°. 1719252346, docente de la Universidad Politécnica Salesiana, declaro que bajo mi tutoría fue desarrollado el trabajo de titulación: DESARROLLO DE UN TÚNEL DE LEVITACIÓN NEUMÁTICA UTILIZANDO UN SISTEMA EMBEBIDO PARA LA OPTIMIZACIÓN DE SISTEMAS DE MANDO Y CONTROL, realizado por Gustavo Alejandro Reyes Navarrete con documento de identificación N°. 1725328650, obteniendo como resultado final el trabajo de titulación bajo la opción Proyecto Técnico que cumple con todos los requisitos determinados por la Universidad Politécnica Salesiana.

Quito, 14 de septiembre del año 2022

Atentamente,



Ing. Andrés Sebastián Calero Calero, MSc
1719252346

DEDICATORIA

A Dios, cuya voluntad es el motor esencial de todo lo que se construye.

A todos quienes han aportado, de cualquier manera, en la construcción de lo que hoy se materializa.

A quienes con su testimonio de vida han sabido inspirar la mía.

A quienes supieron dejar su recuerdo y enseñanzas vivas para inspirarnos.

AGRADECIMIENTO

A Dios, la Madre Dolorosa y el Santo Hno. Miguel, por siempre haber puesto luz en el caminar de este proceso académico.

A mi mami, Cecilia, que me ha sabido inculcar el deseo de saber, y que con las palabras precisas y de la forma adecuada me ha enseñado que el desánimo social y las incoherencias del entorno no deben apagar jamás un espíritu ardiente. A mi papi, Jorge, que con su minuciosidad y alto nivel de responsabilidad me ha enseñado que siempre es importante cumplir con lo que es debido. A ambos, que, en horizontalidad, han sabido construir el soporte sobre el cual me he apoyado siempre.

A Ire, Susi y mi Papi Hernán, que me han mostrado que la vocación de servicio, el esfuerzo, compromiso, y trabajo duro son imprescindibles en cualquier aspecto de la vida.

A Andrés, Diego y Daniel, por ser ejemplo vivo de que la perseverancia, constancia y dedicación son la mejor expresión del compromiso que conlleva la excelencia académica y profesional.

A María José, que con su inventiva, habilidad, recursividad, paciencia, apoyo y celeridad ha sido un apoyo esencial y soporte central para este proceso.

A Sebastián Rojas y Paola Flor que, más allá de convertirse en mis mentores en el mundo de la electrónica y la robótica, me supieron brindar su amistad sincera.

Al Ing. Roberto Pérez, cuya forma de trabajo y experticia me ha inspirado en más de una forma de accionar a nivel profesional; además del aprecio por haber sido un buen amigo, excelente docente y la mejor persona con la que pudimos haber asumido el reto técnico que en su momento lo hicimos.

Al Ing. Andrés Calero por la apertura y consideración que expresó hacia mi desde el primer instante que tomó el rol de tutor de este trabajo y por su forma desinteresada de entregar un esfuerzo adicional al momento de desempeñar su labor.

A la Universidad Politécnica Salesiana y los espacios que me brindó para poder servir, con total entrega, a los distintos estamentos de la institución que lo han necesitado.

A quienes, al final de esta etapa, puedo considerar maestros, que supieron inculcar, en mí, conocimientos, entregarme desinteresadamente enseñanzas, y que me han mostrado que en el desempeño profesional la recompensa a veces es confusa, pero que eso no debe influir en la manera de hacer las cosas.

A todos quienes, por su valor como personas, pude llamar compañeros o amigos a lo largo de este camino académico.

A las historias que no se han acabado de narrar, a las páginas que no se han acabado de escribir, a los números que faltan por contar, a los girasoles por observar, a los caminos por peregrinar, a las distancias por recorrer, a las palabras por pronunciar, a las ideas por materializar, a las balas perdidas, los efectos secundarios, los cristales rotos, los amaneceres contemplados y los anocheceres esperados.

ÍNDICE

INTRODUCCIÓN	1
CAPÍTULO 1	2
ANTECEDENTES	2
1.1 Problemática	2
1.2 Justificación	3
1.3 Objetivos del proyecto	4
1.3.1 Objetivo general	4
1.3.2 Objetivos específicos	4
CAPÍTULO 2	5
MARCO CONCEPTUAL	5
2.1 Túnel de viento	5
2.2 Electric ducted fan (EDF)	6
2.3 Motor sin escobillas	7
2.4 Controlador electrónico de velocidad (ESC)	7
2.5 Sistemas de control automáticos	8
2.6 Controlador PID	10
2.6.1 Acción Proporcional	10
2.6.2 Acción Integral	11
2.6.3 Acción Derivativa	11
2.7 Connected LaunchPad TM4C1294 Tiva C Series	12
2.8 Energia IDE	13
CAPÍTULO 3	14
MARCO METODOLÓGICO	14
3.1 Propuesta de desarrollo	14
3.1 Etapas de desarrollo	15
3.2 Desarrollo mecánico	15
3.2.1 Principio de funcionamiento y fuerzas intervinientes	15
3.2.2 Estructura tubular para recorrido de perdigón	16
3.2.3 Perdigón	16
3.2.4 Hélice	20
3.2.5 Electric ducted fan	22
3.2.6 Motor sin escobillas	24

3.2.7	Diseño mecánico integral	24
3.3	Desarrollo electrónico	25
3.3.1	Control electrónico de velocidad	26
3.3.2	Sensor ultrasónico de distancia	28
3.3.3	Sensor infrarrojo de distancia	29
3.3.4	Convertor de nivel lógico.....	32
3.3.5	Fuente de alimentación	33
3.3.6	Integración de componentes electrónicos.....	34
3.4	Función de transferencia	38
3.5	Programación	38
3.5.1	Mando por posiciones definidas	39
3.5.2	Mando por sensor externo	43
3.5.3	Mando por HMI en HTML.....	48
CAPÍTULO 4		53
PRUEBAS Y AJUSTES.....		53
4.1	Mando por posiciones definidas (Comunicación Serial)	53
4.2	Mando por sensor externo	63
4.3	Mando por HMI en HTML	67
4.4	Resumen de algoritmos	70
CONCLUSIONES.....		71
RECOMENDACIONES		73
REFERENCIAS BIBLIOGRÁFICAS		74

ÍNDICE DE TABLAS

Tabla 3.1: Comparativa características de perdigones	20
Tabla 3.2: Parámetros de operación ESC y motor sin escobillas	27
Tabla 3.3: Datos de caracterización sensor infrarrojo	30
Tabla 3.4: Correspondencia pines periféricos con tarjeta de desarrollo.....	35
Tabla 3.5: Período de oscilación promedio	46
Tabla 3.6: Direcciones dispositivos de la red.....	52
Tabla 4.1: Parámetros limitantes de altura	53
Tabla 4.2: Fórmulas de ajuste constantes controlador PID	54
Tabla 4.3: Parámetros y constantes PID ajustados por Ziegler-Nichols	55
Tabla 4.4: Constantes controlador PID definitivos	60
Tabla 4.5: Resumen de algoritmos para optimización de métodos de mando	70

ÍNDICE DE FIGURAS

Figura 2.1: Túnel de viento	5
Figura 2.2: Túnel de viento vertical	6
Figura 2.3: Partes de un ventilador eléctrico de ducto	6
Figura 2.4: Estructura motor sin escobillas	7
Figura 2.5: Diagrama simplificado ESC	8
Figura 2.6: Diagrama de bloque control en lazo abierto	9
Figura 2.7: Diagrama de bloque control en lazo cerrado	9
Figura 2.8: Diagrama de bloques controlador PID.....	10
Figura 2.9: Diagrama de pines Tiva TM4C1294.....	12
Figura 2.10: Energía IDE.....	13
Figura 3.1: Boceto túnel de viento.....	14
Figura 3.2: Diagrama de fuerzas en levitación neumática	16
Figura 3.3: Perdigón en forma de disco.....	17
Figura 3.4: Movimiento perdigón discal	17
Figura 3.5: Perdigón de doble tapa.....	18
Figura 3.6: Perdigón cilíndrico.....	19
Figura 3.7: Hélice de 9 palas	21
Figura 3.8: Perfil de ángulo de curvatura de pala de hélice	21
Figura 3.9: Vista de perfil hélice de ventilador de extremo BN.....	22
Figura 3.10: Perfil de pluma para ducto de EDF.....	22
Figura 3.11: Medidas principales carcasa EDF	23
Figura 3.12: Carcasa EDF	23
Figura 3.13: Motor sin escobillas	24
Figura 3.14: Diseño mecánico y modelo construido del túnel de levitación neumática	25
Figura 3.15: Esquema de la planta del sistema propuesto.....	25
Figura 3.16: Circuito general ESC	26
Figura 3.17: ESC 30[A].....	27
Figura 3.18: Sensor HC-SR04.....	28
Figura 3.19: Sensor SHARP infrarrojo 4 a 30[cm]	30
Figura 3.20: Gráfica distancia vs. lectura ADC.....	31
Figura 3.21: Conversor de nivel lógico	32
Figura 3.22: Convertidor de nivel lógico Sparkfun.....	33

Figura 3.23: Fuente 12VDC 30A	34
Figura 3.24: Diseño esquemático de PCB de conexión	35
Figura 3.25: Diseño PCB.....	36
Figura 3.26: Vista 3D de PCB de conexiones	36
Figura 3.27: Vista 3D de Tiva con shield para conexión de periféricos	37
Figura 3.28: Diagrama funcional de la planta	37
Figura 3.29: Diagrama de bloques de planta	38
Figura 3.30: Diagrama de flujo de mando por posiciones definidas.....	39
Figura 3.31: Código fuente lectura serial	40
Figura 3.32: Lectura distancia HC-SR04	41
Figura 3.33: Código control por histéresis	42
Figura 3.34: Código ESC librería servo	43
Figura 3.35: Lectura ADC vs. tiempo (Sin filtro y aplicado filtro EMA pasa bajo).....	44
Figura 3.36: Gráfica período de oscilación	46
Figura 3.37: Código PID discreto.....	47
Figura 3.38: Código comunicación Ethernet.....	49
Figura 3.39: Obtención de datos desde cliente web en HTML	49
Figura 3.40: Cliente web en HTML para envío de datos a tarjeta Tiva TM4C1294.....	50
Figura 3.41: Topología de red comunicación Ethernet TM4C1294.....	51
Figura 4.1: Gráfica primera prueba de funcionamiento de planta con controlador PID	56
Figura 4.2: Gráfica segunda prueba de funcionamiento de planta con controlador PID.....	57
Figura 4.3: Gráfica tercera prueba de funcionamiento de planta con controlador PID.....	58
Figura 4.4: Gráfica cuarta prueba de funcionamiento de planta con controlador PID	59
Figura 4.5: Gráfica quinta prueba de funcionamiento de planta con controlador PID.....	60
Figura 4.6: Prueba de funcionamiento de planta con primer método de mando (comunicación serial)	61
Figura 4.7: Primera prueba de funcionamiento controlador PID comandado por sensor externo	64
Figura 4.8: Segunda prueba de funcionamiento controlador PID comandado por sensor externo	65
Figura 4.9: Prueba de funcionamiento de planta con segundo método de mando (sensor externo).....	66
Figura 4.10: Prueba de funcionamiento controlador PID comandado por HTML vía Ethernet.....	67

Figura 4.11: Prueba de funcionamiento de planta con tercer método de mando (HMI en HTML) 68

RESUMEN

Un túnel de levitación neumática es un dispositivo que ha evolucionado desde su creación, pasando de ser un equipo enteramente experimental y para comprobación; a diversificar su uso a ámbitos de entrenamiento profesional y entretenimiento en general. Al dejar de tener en su interior únicamente objetos, mecanismos o máquinas sujetas a comprobación de sus características aerodinámicas, y pasar a ser de uso personal; el correcto control del flujo de aire dentro de estos dispositivos se volvió una necesidad imperiosa. Un control enteramente manual de velocidad o flujo de aire caracteriza a este tipo de túneles de levitación neumática, o mejor conocidos como túneles de viento. La presencia de una persona dentro del túnel, realizando una actividad recreativa, profesional o de entrenamiento, exige un control más avanzado para garantizar la seguridad de quien lo utiliza, siendo lo esencial diversificar sus medios de mando.

Por lo indicado, en este documento se presenta el desarrollo de un túnel de levitación neumática que utiliza un sistema embebido para la optimización de medios de mando y control. El sistema es una representación a pequeña escala de un túnel de viento con la inclusión de métodos de mando variados a través de una tarjeta de desarrollo TIVA TM4C129. El sistema tiene un control por retroalimentación, con la adquisición de posición de un perdigón por medio de un sensor de distancia de tecnología ultrasónica, que va al controlador para manipular la salida del sistema en la magnitud que sea necesario para satisfacer las consignas de funcionamiento. El sistema varía el flujo de aire presente dentro del túnel para controlar la posición del perdigón gracias a la acción de un EDF (Electric Ducted Fan) de velocidad controlable.

Los métodos de mando incorporados para este túnel se implementaron en concordancia con la optimización deseada para el sistema, como con una imperante necesidad de incremento de los niveles de seguridad que podrían estar incluidos en los sistemas a gran escala. Se adicionaron métodos de mando de manera natural a través de una comunicación tradicional, mandos gráficos mediante HMI y comunicación Ethernet, y mando mediante un sensor colocado para interacción humana.

Palabras Clave: EDF, Ethernet, HMI, túnel de viento.

ABSTRACT

A pneumatic levitation tunnel is a device that has evolved since its creation, going from an entirely experimental equipment and for verifications; to diversify its usage to professional training and general entertainment spheres. By no longer having inside it only objects, mechanisms, and machines subjects of verification of their aerodynamic characteristics, and going to a usage with people involved. Therefore, the correct control of the airflow inside of these devices became an imperative need. An entirely manual control of the airflow is the main characteristic of these types of pneumatic levitation tunnels, or better known as wind tunnels. The presence of a person inside the tunnel, doing a recreative, professional or training activity, demands a more advanced control to guarantee the safety of who is using it, being the essential to diversify its means of command.

Due to all wrote down above, in this document is shown the development of a pneumatic levitation tunnel using an embedded system for the optimization of the command's means and control. The system is a small-scale representation of a wind tunnel with the inclusion of assorted command's means trough a TIVA TM4C129 development board. The system has a feedback control, with the acquisition of the position of a pellet with the usage of an ultrasonic technology sensor, which goes to the controller to manage the system output in the necessary size to satisfy the operating set points. The system varies the airflow inside the tunnel to control pellet position with the action of an EDF (Electric Ducted Fan) of controllable Speed.

Command's means incorporated for this tunnel were implemented in concordance with the desired optimization for the system, as a prevailing need of increasing the security levels that could be included in the big scale systems. There were added command's means in natural way through traditional communication, graphic command through HMI and Ethernet communication, and command through a sensor placed for human interaction.

Keywords: EDF, Ethernet, HMI, wind tunnel.

INTRODUCCIÓN

Cuando el dispositivo de túnel de viento, creado alrededor del año 1871, evolucionó al uso expandido para entrenamiento profesional y entretenimiento en general; surgió la necesidad de contar con un sistema de control más preciso, para aumentar los estándares de bienestar y seguridad del usuario del dispositivo. La inclusión de sistemas de control o métodos de mando nuevos en sistemas de este tipo, donde la seguridad del usuario es el plano central, se dificulta al ser implementado en los dispositivos que ya se encuentran en operación constante. Por esta razón, se recurre a los modelos a pequeña escala, que permiten etapas de desarrollo, implementación, pruebas y robustecimiento de los métodos de mando, controladores y algoritmos sin involucrar ni arriesgar la integridad de los usuarios.

Dado este panorama se procede a desarrollar un túnel de levitación neumática utilizando un sistema embebido para la optimización de sistemas de mando y control, con la perspectiva de simular las características de dispositivos a gran escala.

El proceso de desarrollo de este proyecto se presenta en cuatro capítulos. En el primero se abarca la problemática del proyecto, su importancia y el objetivo a ser alcanzado para su solución. El segundo enmarca el marco conceptual del proyecto, que justifica el enfoque teórico de las técnicas, elementos, dispositivos y algoritmos utilizados. El tercero muestra el marco procedimental y el proceso para el diseño, simulación e implementación del proyecto. Y, el cuarto incorpora las pruebas llevadas a cabo que incluyen ajuste de parámetros ante las circunstancias observadas. Finalmente, se presentan las conclusiones y recomendaciones y documentación anexa.

CAPÍTULO 1

ANTECEDENTES

1.1 Problemática

Los túneles de viento son dispositivos de apoyo experimental, es decir, fueron originalmente pensados y creados como punto de apoyo para la industria aeronáutica y aeroespacial al momento de hacer estudios aerodinámicos. Por ejemplo, al crear vehículos aéreos, la comprobación final de sus características aerodinámicas se realiza en un túnel de viento. Estos túneles conducen enormes flujos de aire dentro de una cámara, y con diversos aparatos medidores se registran los diferentes fenómenos que provoca el aire en los vehículos.

A partir de la década de 1980, el movimiento de aplicación de tecnologías para el desarrollo de pruebas y el crecimiento de la industria del entretenimiento, generaron un fenómeno en constante crecimiento alrededor de la popularización de los túneles de viento. El creciente acceso a los dispositivos tecnológicos gracias a su desarrollo exponencial y el decremento de sus precios permitió que un túnel de viento pase, de ser un dispositivo experimental exclusivo de las grandes industrias a un objeto presente en entornos más comunes.

Originalmente utilizados únicamente en los estudios de aeronáutica y aerodinámica espacial, los túneles de viento incursionaron de manera fuerte en la industria del entrenamiento y del entretenimiento.

Ante la imperiosa necesidad de entrenar de una manera segura a los aspirantes a paracaidistas militares, y con un mayor acceso a las plataformas tecnológicas integrales, los túneles de viento se convirtieron en un nuevo dispositivo de importancia (muchas veces necesario e incluso imprescindible) en bases militares de alto desempeño. Esta, al igual que gran cantidad de la tecnología militar, dio un salto impresionante al uso del público en general a partir de la mitad de la primera década de los 2000; popularizándose, así, estos dispositivos para el entretenimiento y entrenamiento particular.

Sean para entretenimiento o para niveles profesionales; los túneles de viento poseen grandes prestaciones en el ámbito de la experiencia de cualquier tipo de usuario. Por ejemplo, se incluyen simulaciones de condiciones ambientales desde regulares, pasando por reales y

llegando incluso a hostiles; sin embargo, se presta poca atención en el modo de control del flujo de aire que ingresa al túnel como tal. La consola de mando de la mayoría de los modelos de túneles de viento permite activar y desactivar el paso del viento proporcionalmente, mediante perillas manuales. El operador del túnel de viento es el encargado del monitoreo visual constante y la identificación de perturbaciones que se presenten en el flujo de aire dentro del túnel, para poder realizar correcciones de manera manual. Además, el operador se encarga de la precisión en el movimiento de las perillas según su habilidad manual. Este tipo de control de la consola genera a la posibilidad de que existan gran cantidad de errores en el manejo, decantándose, posteriormente, en riesgos para quien está utilizando el túnel.

Por lo que, teniendo en cuenta el nivel de seguridad que se debe prever en este tipo de túneles; se prioriza la necesidad de buscar la mejora e inclusión de métodos de mando variados en sistemas de control dinámicos para emulación de túneles de levitación neumática, de manera que se pueda llevarlos a los entornos de prueba y a gran escala.

1.2 Justificación

Desde 1871, con el desarrollo del primer túnel aerodinámico, autoría de Francis Herbert Wenham, los túneles de viento se convirtieron en uno de los dispositivos de validación experimental más versátil y utilizado a nivel industrial. A partir del desarrollo de estos dispositivos, el concepto se diversifica al de túnel de viento vertical, cuya utilidad se centra en la simulación de movimientos de caída libre para el entrenamiento de paracaidistas o similares. Todas estas actividades, sean profesionales o amateur, (de caída libre) son las que más peligro y dificultad representan para quienes las practican e instruyen; debido a la cantidad de fuerzas poco predecibles e incontrolables presentes en el aire a la altura en la que se desarrollan. La característica central de este tipo de túneles está generalizada en el mando de control que se basa solo en la actividad del operario por su contacto visual constante con usuario.

Es, por lo tanto, relevante proponer el desarrollo de un túnel de levitación neumática (túnel de viento) a pequeña escala; para establecer un control de métodos de mando variados, que permita la interacción dinámica entre el controlador y el cuerpo presente dentro del túnel. Los métodos de control propuestos brindarán precisión al sistema, corrección rápida de errores, adaptación a diferentes condiciones e inclusión de mecanismos de seguridad. Esto se centra en gran utilidad para los desarrolladores de sistemas de pruebas aerodinámicas y

de levitación neumática de objetos, que utilizan túneles de viento para la verificación de estándares de industrias tradicionales y de entretenimiento extremo.

1.3 Objetivos del proyecto

1.3.1 Objetivo general

- Desarrollar un túnel de levitación neumática controlado por métodos de mando variados mediante la tarjeta TIVA TM4C129 y la herramienta Energia IDE, aplicando algoritmos que optimicen su mando y control.

1.3.2 Objetivos específicos

- Estudiar los métodos de mando de los túneles de viento comerciales utilizados en aplicaciones para entrenamiento, mediante la investigación de documentos técnicos para determinar las características más comunes de sus sistemas de mando y control.
- Elaborar un túnel de levitación neumática con filamento PLA impreso en tecnología FDM, para desplazamiento lineal vertical de un perdigón a lo largo del recorrido permitido en un tubo.
- Implementar un control de posición del perdigón utilizando posiciones predefinidas proporcionales al recorrido total, y un control PID cuyo set point se defina mediante una interacción humana con un sensor externo, que permitan comandar la posición vertical del perdigón dentro del túnel.
- Gestionar un control HMI mediante HTML a través de la plataforma Energia para el control del movimiento vertical del perdigón dentro del túnel.
- Desarrollar pruebas de funcionamiento del emulador del túnel de viento, mediante la aplicación de los diferentes métodos de mando implementados para determinar los mejores parámetros de operación de cada uno.

CAPÍTULO 2

MARCO CONCEPTUAL

2.1 Túnel de viento

Los túneles de viento son estructuras tubulares con aire que entra en su interior, utilizados para recrear la respuesta de las características aerodinámicas de un objeto que se somete a flujos de aire directos o indirectos. Son herramientas de investigación que sirven de soporte en las entidades que analizan los efectos aerodinámicos sobre ciertos dispositivos y buscan acercar las condiciones experimentales lo más posibles a las condiciones reales. Estos dispositivos se utilizan, generalmente, para el análisis de las condiciones aerodinámicas de automóviles, aviones, paracaídas, o similares (Muñoz Mejías, 2012).

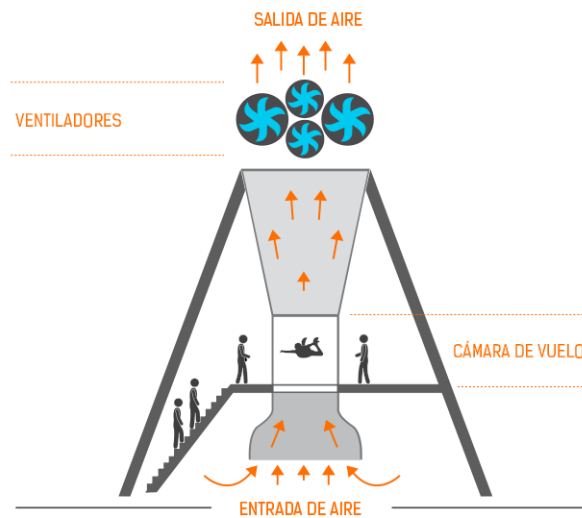
Figura 2.3: Túnel de viento



Túnel de viento subsónico con conexión a PC, TecnoEdu (s.f.)

Dentro de la diversificación de este tipo de dispositivos, y guardando su principio de funcionamiento, se desarrolló el túnel de viento vertical, cuyo enfoque se centra en mayor medida para el vuelo humano, llegando incluso a definirse como simulador de vuelo o “indoor skydiving tunnel”. El uso de estos túneles de viento verticales recrea, de manera comparable, la sensación de un vuelo en paracaídas real, incrementando el tiempo promedio de la actividad de 1 minuto en promedio a más de 10. Además, las condiciones de clima se independizan de la posibilidad de hacer actividades de práctica paracaidista (Viento Aventura, 2016).

Figura 2.3: Túnel de viento vertical

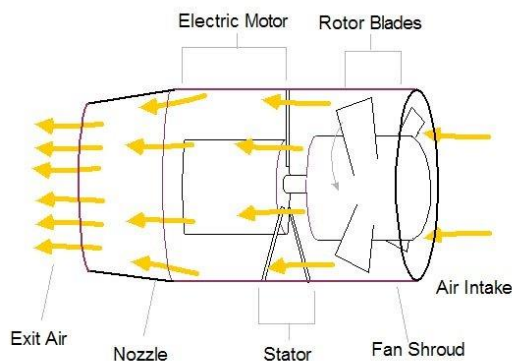


Túnel de viento vertical, Vuela (2018)

2.2 Electric ducted fan (EDF)

Una “electric ducted fan”, también definida como una turbina eléctrica; es un sistema de propulsión, formado por un ventilador de varias palas, impulsado por un motor eléctrico de alta velocidad con una cobertura tipo carcasa. Se utilizan en dispositivos o sistemas en los que es necesario incrementar el empuje regular del motor utilizado. Es común en aplicaciones aeronáuticas, porque aumenta la eficiencia del motor, reduce el ruido y reordenada el flujo de aire circulante (Centre for Aerodynamics and Environmental Flow, s.f.).

Figura 2.3: Partes de un ventilador eléctrico de ducto

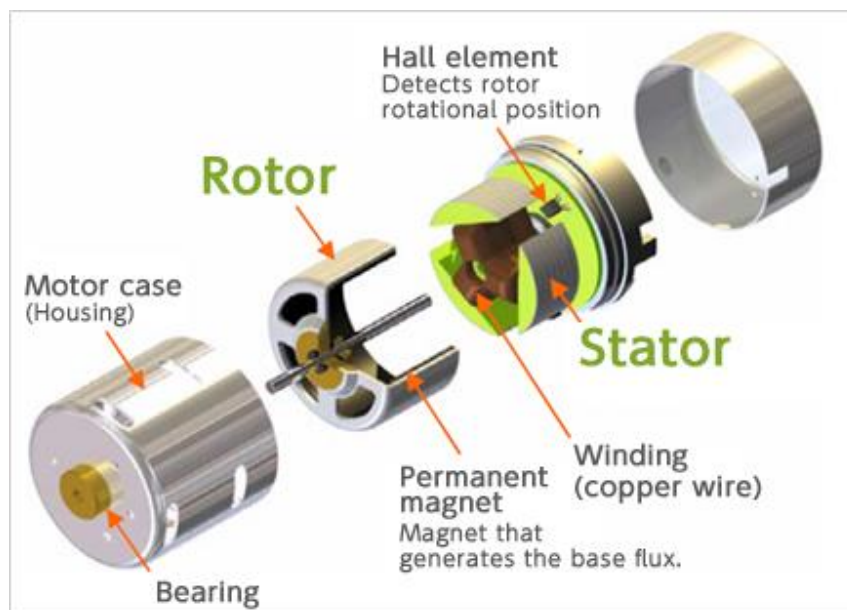


Esquema de un ventilador eléctrico de ducto, Sgueglia, Schmolgruber, Bartoli, & Atinault (2018)

2.3 Motor sin escobillas

Un motor sin escobillas de corriente continua o BLDC (por sus siglas en inglés), es un tipo de motor popularizado por su mejora de características de velocidad vs. torque, alta respuesta dinámica, alta eficiencia, bajo ruido, alta velocidad, supresión de chispas y reducción del ruido eléctrico. Estos motores no utilizan escobillas para su conmutación, porque dicho proceso se realiza de manera electrónica. Este motor se clasifica como un tipo de motor síncrono (Yedamale, 2003).

Figura 2.3: Estructura motor sin escobillas



Estructura interna de un motor si escobillas de giro exterior, (Nidec Corporation, s.f.)

El control de este tipo de motores se realiza energizando un par de polos alternados para provocar el movimiento de un tercero, por atracción de un imán. Este proceso es complicado de realizar de formas tradicionales, por lo que se emplea un dispositivo específico llamado “Control electrónico de velocidad”

2.4 Controlador electrónico de velocidad (ESC)

Un controlador electrónico de velocidad se define como un circuito electrónico utilizado para cambiar la velocidad de un motor eléctrico y actuar como freno dinámico. Su uso se ha expandido con la creciente popularidad de los motores sin escobillas, ya que entregan una potencia eléctrica partida en 3 etapas para su funcionamiento.

Su construcción se realiza con un regulador de voltaje, un procesador y una etapa de conmutación de potencia conformada por transistores tipo FET. Este arreglo de transistores provoca que la alimentación de las 3 entradas de un motor sin escobillas ocurra de manera precisa y en una secuencia definida (Gong, MacNeill, & Verstraete, 2018).

Figura 2.3: Diagrama simplificado ESC

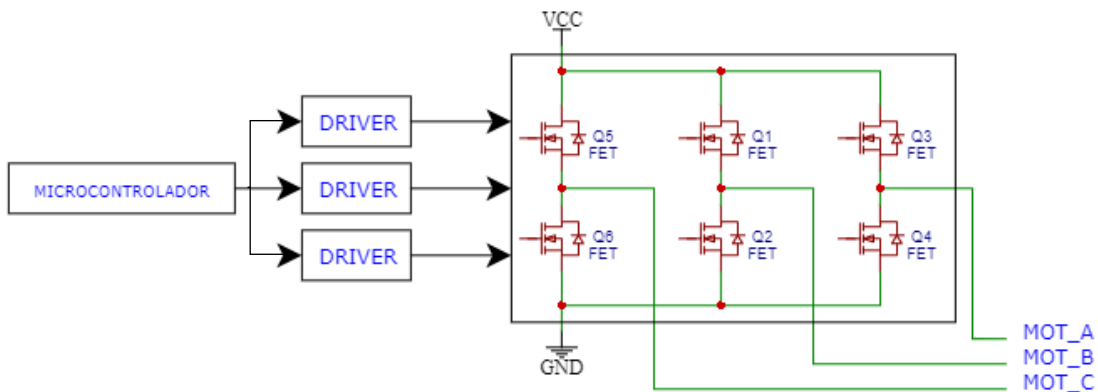


Diagrama simplificado referencial de la estructura interna de un controlador electrónico de velocidad, Gustavo Reyes

2.5 Sistemas de control automáticos

Un sistema de control se define como el conjunto de los elementos necesarios para lograr una manipulación de variables para lograr a un objetivo definido. Estos elementos rigen el comportamiento de un sistema mecánico, informático o mixto para realizar una acción definida. Un sistema de control se define como automático, cuando no necesita una intervención externa para manipular la variable de salida, sino que lo hace con la única intervención de sus propios elementos. Según su funcionalidad, los sistemas de control se clasifican como sistemas en lazo cerrado o en lazo abierto (Martínez, 2018).

En un sistema de control de lazo abierto se tiene una señal de entrada y una de salida, siendo la primera totalmente independiente de la segunda. La señal de entrada y su cambio responde a condiciones fijas de operación, sin realimentación comparativa, y siendo dependiente de calibración.

Figura 2.3: Diagrama de bloque control en lazo abierto

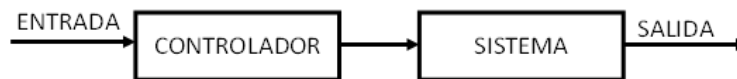


Diagrama de bloque simplificado sistema de control en lazo abierto, Gustavo Reyes

Un sistema de control en lazo cerrado incorpora una señal de realimentación, que se procesa para incluir una señal de error al controlador, para que este la interprete y lleve la salida del sistema al valor más adecuado o cercano al esperado. Este tipo de sistemas de control busca ser más estable, robusto y eficiente que uno de lazo abierto.

Figura 2.3: Diagrama de bloque control en lazo cerrado

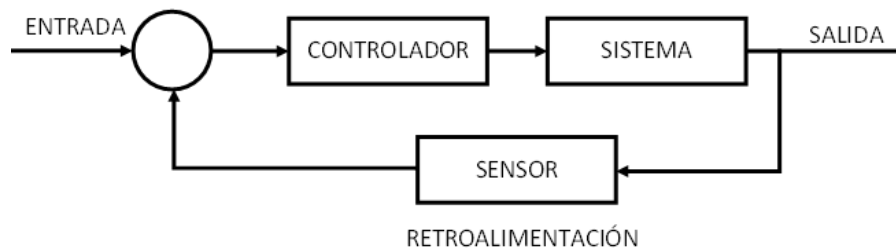


Diagrama de bloque simplificado sistema de control en lazo cerrado, Gustavo Reyes

Los sistemas de control tienen una gran variedad de elementos que dependen de la propia naturaleza de cada sistema. Sin embargo, existen algunos elementos que se pueden definir como los mínimos componentes del sistema, entre los que están la señal de entrada, la señal de salida, la variable manipulada, la variable controlada, la retroalimentación, el control, el proceso y la planta (Braslavsky, 2002).

La planta es el sistema en físico que va a ser controlado y está regido bajo la dinámica propia de los elementos del sistema. El proceso y el control son las partes informáticas del proceso que contienen el algoritmo y su traducción a señales eléctricas del sistema de control. La señal de entrada y la de salida son las señales eléctricas que ingresa y sale del sistema, respectivamente; y que, en conjunto con la señal de retroalimentación representan los estímulos y respuestas del sistema. La variable manipulada es aquella que debe ser modificada y la variable controlada es la que experimenta cambio por la modificación realizada; se podrían definir como la representación de la señal de entrada y salida, respectivamente (Muñoz V. & Espinosa N., 2010).

2.6 Controlador PID

Un controlador PID (Proporcional, Integral, Derivativo) es un sistema de control en lazo cerrado para llegar a una salida deseada. Este controlador es ampliamente utilizado en la industria por su robustez y sencillez al momento de ser aplicado. La denominación de controlador proporcional, integral y derivativo se refiere a las acciones que se realizan sobre la señal comparada con el error para su retorno a la planta o sistema. (Mazzone, 2002)

Se tiene una señal de entrada denominada como referencia que es el estado deseado de salida del sistema; además, esta señal de salida se realimenta al sistema para poder tener su respectiva corrección. La diferencia entre la referencia y la señal de retroalimentación se conoce como error, y mientras menor sea, el sistema se encuentra más cerca de llegar a su estado deseado. (Ogata, 1998)

Figura 2.3: Diagrama de bloques controlador PID

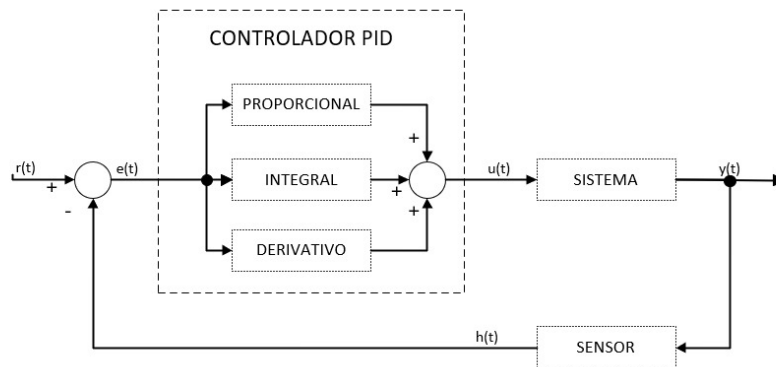


Diagrama de bloques controlador PID en lazo cerrado, Gustavo Reyes

Al tener tres acciones posibles de realizar, el controlador PID como tal no es único, sino que se define como una familia de controladores, que pueden darse por las combinaciones de sus parámetros. Estos controladores son: P, I, PI, PD, PID. Cada uno de los cuales otorgan diferentes parámetros de comportamiento a la planta o sistema, y su uso depende de la particularidad de cada caso.

2.6.1 Acción Proporcional

La acción de control es proporcional a la señal de error $e(t)$, tiene una ganancia proporcional ajustable K_p y una señal de salida denominada $u(t)$. Su salida se define por la ecuación 2.1.

$$u(t) = K_p e(t) \quad \text{Ec. (2.1)}$$

Esta acción se utiliza para intentar reducir el error del sistema, y el aumento de la constante proporcional ocasiona incremento en la velocidad de respuesta del sistema, al mismo tiempo que disminuye el error en régimen permanente del sistema pero aumentando su inestabilidad.

2.6.2 Acción Integral

La acción de control integral es proporcional al error acumulado, ocasionando una señal de salida $u(t)$ diferente de cero cuando el error $e(t)$ es cero. Su salida se define por la ecuación 2.2.

$$u(t) = K_i \int_0^t e(\tau) d\tau \quad \text{Ec. (2.2)}$$

Esta acción se utiliza para calcular la integral (sumatoria) del error y el aumento de su constante K_i provoca una disminución del error del sistema en régimen permanente, aumenta su velocidad, pero también incrementa su inestabilidad.

2.6.3 Acción Derivativa

La acción derivativa logra hacer más rápida la acción de control. Su salida $u(t)$ se define por la ecuación 2.3.

$$u(t) = K_d \frac{de(t)}{dt} \quad \text{Ec. (2.3)}$$

Esta acción no se puede utilizar de manera aislada, por lo que lo indicado en la ecuación siempre deberá ir en combinación con acciones complementarias. Al incrementar la constante K_d se aumenta la estabilidad del sistema, sin mostrar cambios en el error de régimen permanente, pero disminuye la velocidad del sistema aunque en poca magnitud.

La ecuación que describe de manera íntegra el comportamiento del controlador PID puede ser expresada de dos maneras, en término de la constante proporcional y constantes de tiempo, o en términos de las constantes de cada acción. Lo expresado se puede apreciar en las ecuaciones 2.4 y 2.5.

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{de(t)}{dt} \quad \text{Ec. (2.4)}$$

$$u(t) = K_p e(t) + \frac{K_p}{T_i} \int_0^t e(\tau) d\tau + K_p T_d \frac{de(t)}{dt} \quad \text{Ec. (2.5)}$$

Los parámetros del controlador PID, es decir K_p , K_i y K_d , tienen limitaciones en los sistemas y plantas reales. Por ejemplo, llegará un momento en el que aumentar K_p no supondrá cambio en la salida, es decir se saturará. Es por esto, que este controlador debe ser sintonizado, para lo que existe varios métodos de ajuste, siendo uno de los más populares y adecuados el de Ziegler-Nichols.

2.7 Connected LaunchPad TM4C1294 Tiva C Series

La tarjeta Tiva C Series TM4C1294 es una plataforma de desarrollo desarrollada por Texas Instruments de bajo costo basado en microcontroladores ARM Cortex-M4F. La tarjeta tiene incluido un conector Ethernet 10/100, un puerto USB2.0, modo de hibernación, modulación de ancho de pulso, conectividad serial, cuatro LEDs y dos pulsadores de usuario.

Figura 2.3: Diagrama de pines Tiva TM4C1294

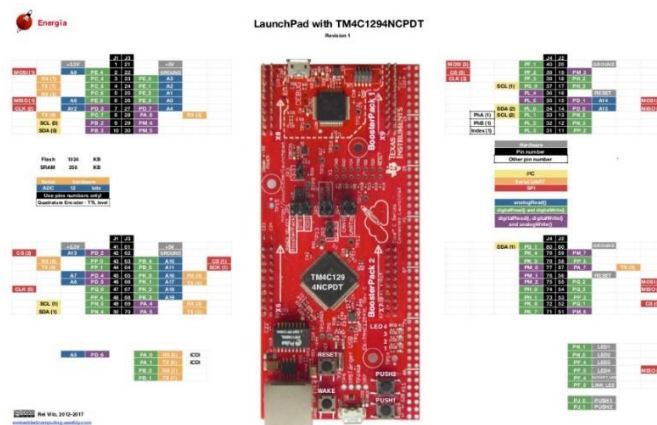


Diagrama de pines Connected LaunchPad TM4C1294 Tiva C Series, (Energia, s.f.)

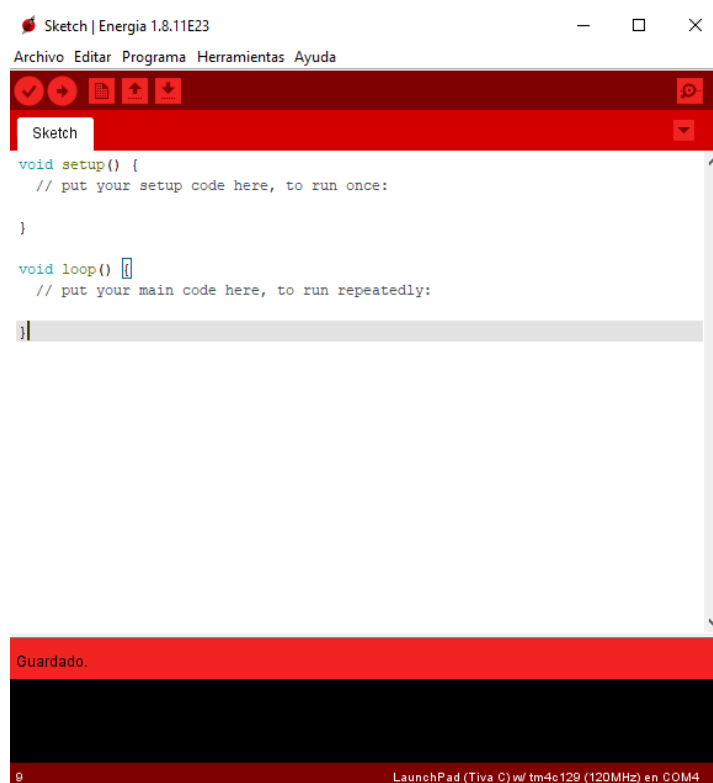
El microcontrolador en el que se basa esta tarjeta de desarrollo es un ARM Cortex-M4F de 32 bits con una memoria flash de 1024 kB, memoria SRAM de 256 kB y EEPROM de 6kB. El microcontrolador opera a una frecuencia de hasta 120MHz gracias a un multiplicador de reloj que manipula la frecuencia del cristal de 25MHz y la tarjeta está diseñada para su

integración con tarjetas de expansión de manera sencilla. (Texas Instruments Incorporated, 2016)

2.8 Energia IDE

En el entorno de desarrollo Energia es una plataforma de código abierto desarrollada para llevar la estructura de Wiring y Arduino a las tarjetas de desarrollo de Texas Instruments. Habilitado para múltiples sistemas operativos, este entorno utiliza el compilador mspgcc, que es portable y amigable con el usuario. (Energia, s.f.)

Figura 2.3: Energia IDE



Ventana de Energia IDE para escritura de sketch, Gustavo Reyes

CAPÍTULO 3

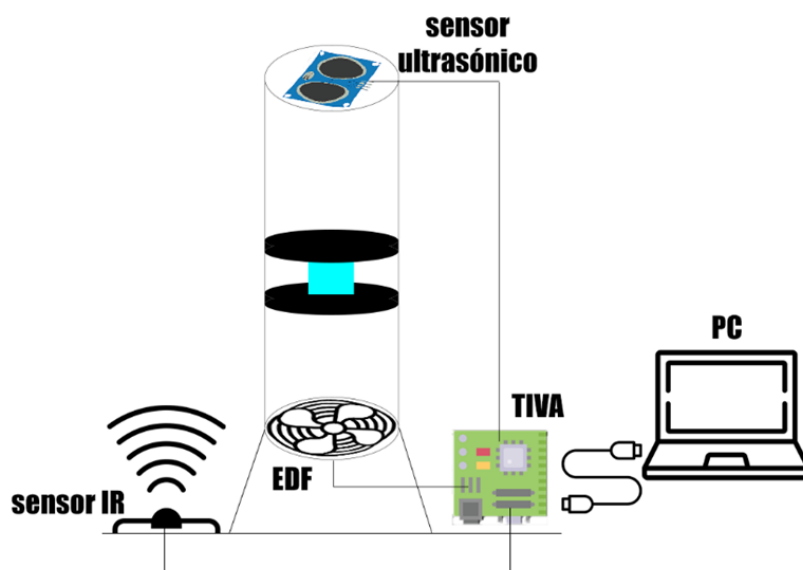
MARCO METODOLÓGICO

3.1 Propuesta de desarrollo

Para el desarrollo del proyecto se tomaron en cuenta algunas características imprescindibles en su correcto funcionamiento así como los principios y parámetros de operación descritos en los capítulos previos.

El proyecto se concentra en el desarrollo de una plataforma a pequeña escala para simulación de un túnel de viento vertical. La plataforma está conformada por una EDF con un motor sin escobillas que ingresa aire a un tubo de acrílico transparente para mover un perdigón que se encuentra en su interior. La posición del perdigón dentro del tubo se obtiene gracias a un sensor ultrasónico colocado en la parte final del recorrido. El sistema de control se aloja en una tarjeta TM4C1294 Tiva C Series y consta de tres métodos de adquisición del punto de referencia: por el puerto de comunicación serial con Energia IDE, mediante la interacción del operario con un sensor de distancia adicional, y mediante comunicación Ethernet gracias al puerto incluido en la tarjeta de desarrollo.

Figura 3.4: Boceto túnel de viento



Boceto de la plataforma de simulación de túnel de viento a pequeña escala, Gustavo Reyes

3.1 Etapas de desarrollo

El proyecto se ejecutó en 4 fases que comprenden el desarrollo mecánico, el desarrollo electrónico, la programación y la etapa de pruebas y ajustes al final. En cada una de estas etapas se realizaron ajustes para llegar al mejor funcionamiento de cada uno de los elementos y su correcta integración en la etapa de implementación.

3.2 Desarrollo mecánico

Para el proceso de desarrollo mecánico, se establecieron las condiciones de operación, los componentes a utilizar durante el desarrollo y la fabricación de los componentes personalizados o no comerciales necesarios. La fabricación de los componentes se desarrolló con manufactura aditiva y sustractiva dependiendo el caso.

3.2.1 Principio de funcionamiento y fuerzas intervinientes

El principio de funcionamiento general radica en que, como para cualquier cuerpo, se tiene una fuerza ejercida por la gravedad con dirección al centro de la Tierra (es decir, hacia abajo) que debe ser vencida. La fuerza ejercida por el aire impulsado dentro del túnel de viento debe superar esta fuerza para elevar el objeto y debe igualarla para mantenerlo a cierta altura.

La primera fuerza que se involucra en el funcionamiento del túnel de viento es la que la gravedad ejerce sobre el perdigón, que es constante y está definida por la Segunda Ley de Newton, y por consecuencia es dependiente de su masa. Esto se muestra en la Ecuación 3.1, indicando que la fuerza de ejercida por la gravedad (F_g) corresponde a la multiplicación de la masa (m) por la aceleración de la gravedad (g).

$$F_g = mg \quad \text{Ec. (3.1)}$$

Otra de las fuerzas presentes en esta interacción es la fuerza ejercida por el viento (o aire) que ingresa al túnel de viento por la parte inferior. Ésta se define mediante la fórmula que se indica en la ecuación 3.2, que establece que la fuerza que el viento ejerce sobre un área plana es directamente proporcional al área de aplicación y su velocidad al cuadrado multiplicado por el factor 1.22 obtenido de la experimentación.

$$F_{viento} = 1.22Av^2 \quad \text{Ec. (3.2)}$$

La última fuerza que se involucra dentro de la levitación del túnel de viento es la de rozamiento. Si bien el perdigón no está en constante contacto con el cilindro que compone el túnel de viento, existen sectores de este que pueden presentar variaciones en su área propias de su construcción. Se buscó reducir en la mayor magnitud posible el efecto de esta fuerza hasta volverla despreciable.

Figura 3.4: Diagrama de fuerzas en levitación neumática

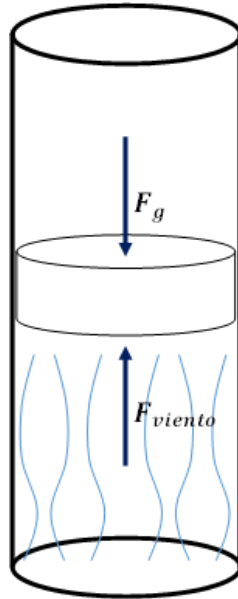


Diagrama de fuerzas a las que está sometida un perdigón en levitación neumática despreciando la fuerza de rozamiento, Gustavo Reyes

3.2.2 Estructura tubular para recorrido de perdigón

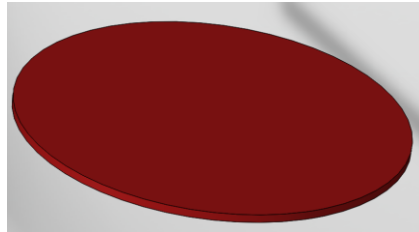
El tubo utilizado para el recorrido del túnel de viento es un tubo de polimetilmetacrilato transparente de 100[mm] de diámetro. Se utilizó este material ya que, al ser uno de los denominados “plásticos de ingeniería” posee propiedades mecánicas mejoradas respecto a los de uso común como el PVC, entre otros.

3.2.3 Perdigón

El perdigón es el cuerpo que va a desplazarse a lo largo del recorrido dentro del tubo, simulando al usuario final del producto, y es imprescindible que cumpla con ciertas características. La masa es uno de los parámetros más importante de cuidar, ya que, para el modelo de simulación, ésta se debe encontrar en un equilibrio; debe estar en un rango para que el cuerpo sea lo suficientemente ligero para ser levantado pero no tanto como para que la fuerza ejercida por el viento lo lleve al final de su recorrido en un tiempo muy reducido.

El primer diseño para ser probado es un disco plano de 3[mm] de espesor y 98[mm] diámetro fabricado con manufactura aditiva de modelado por deposición fundida (FDM), lo que garantizó su uniformidad en la distribución de la masa y densidad constante en toda su estructura.

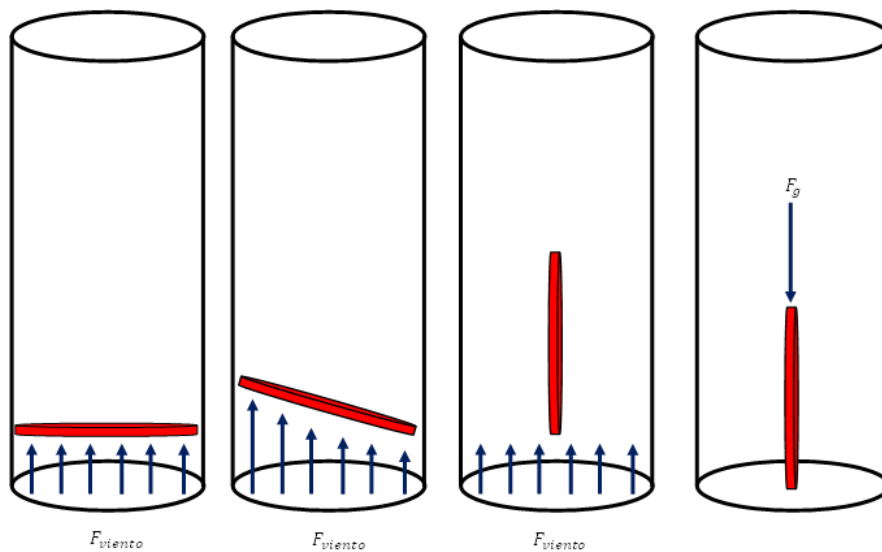
Figura 3.4: Perdigón en forma de disco



Diseño de perdigón en forma de disco, Gustavo Reyes

La forma de disco del perdigón garantiza una pequeña masa y no impone resistencia considerable a la fuerza ejercida por el viento, sin embargo presenta el problema de la inestabilidad por su espesor reducido. Esta inestabilidad es apreciable al momento de ingresar el aire, provocando una rotación del disco en uno de sus ejes, colocándose de forma vertical y cayendo de manera abrupta, como se observa en la figura 3.4.

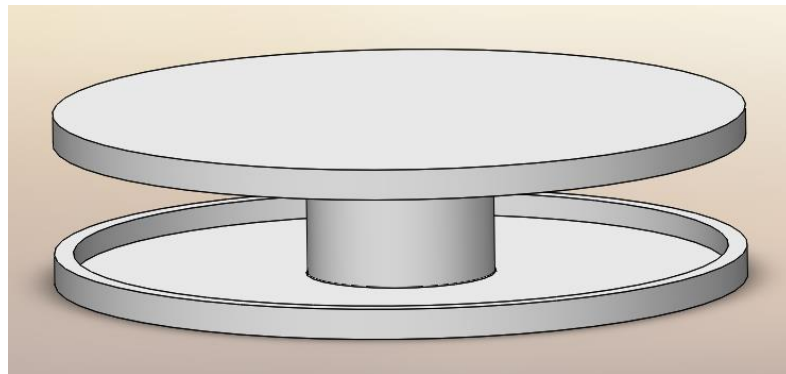
Figura 3.4: Movimiento perdigón discal



Progresión de movimiento de perdigón discal al ser sometido a un flujo de aire con sus fuerzas actuantes, Gustavo Reyes

Al observar ese comportamiento, el diseño de perdigón fue cambiado a un modelo compuesto de un cilindro central con dos tapas, cada una de las cuales tiene un espesor externo de 4mm y un espesor interno de 1[mm]; el cilindro central es de 20[mm] de altura y fabricado en su totalidad con ácido poliláctico (PLA) utilizando manufactura por FDM. En este caso se redujo el diámetro exterior a 91[mm] para subsanar los inconvenientes de esta característica observada con el diseño anterior. Este diseño se observa en la figura 3.5.

Figura 3.4: Perdigón de doble tapa



Diseño de perdigón de doble tapa y cilindro central, Gustavo Reyes

Este diseño de perdigón presentó una serie de limitantes como su giro errático y dificultad para ser elevado. Esto se dio, en gran medida por la forma de su parte superior e inferior que permite el ingreso de corrientes de aire en el medio de su estructura con comportamiento aleatorio que provocan su inestabilidad. Además, gracias a este modelo se identificó el problema del material de fabricación del perdigón.

El perdigón, al estar fabricado con PLA con manufactura FDM, necesita un porcentaje mínimo de relleno para garantizar su solidez. Esta solidez le incorpora a este elemento, estabilidad, protección contra caídas abruptas en fase de pruebas entre otros. Sin embargo, es imposible llegar al grado de solidez necesario con rellenos bajos, siendo imprescindible el uso de más material en su manufactura, es decir aumentando su masa. Por esta razón, se procedió a reemplazar la manufactura del perdigón a una manera artesanal.

El criterio de diseño se mantuvo respecto al modelo anterior para conserva la estabilidad que representa tener un cilindro entre dos discos para un cuerpo que va a ser sometido a levitación neumática. Se eligieron para materiales de fabricación: poliestireno expandido, cartón y papel encerado para su recubrimiento. El primero para la fabricación de los discos de la parte

superior e inferior del perdigón, que además ayuda a lograr solidez pero con una masa reducida. El segundo para el cilindro central, que esta ocasión se optó por hacerlo hueco, para reducir en mayor magnitud su masa. El tercero se utilizó a manera de cubierta exterior para conservar la integridad del perdigón, aumentar la superficie que se encuentra paralela al tubo, y por su característica de ser encerado, para que se reduzca la fuerza de rozamiento en caso de existir contacto.

Los discos de poliestireno se colocaron en forma de tapas, tanto en la parte superior como inferior, con diámetro de 93[mm] y espesor de 15[mm] cada una. La altura del cilindro de cartón hueco es de 50[mm], para hacer una altura total del perdigón de 80[mm], parámetro que, más adelante, se convertirá en esencial al momento de establecer el recorrido de este elemento.

Figura 3.4: Perdigón cilíndrico



Perdigón cilíndrico definitivo de poliestireno, cartón y papel encerado, Gustavo Reyes

Para evaluar las características primordiales de cada tipo de perdigón, fue necesario establecer una comparación entre los 3 tipos desarrollados, sintetizando los resultados en la Tabla 3.1.

Tabla 3.4: Comparativa características de perdigones

Característica	Perdigón de disco	Perdigón con tapas y cilindro central	Perdigón cilíndrico recubierto
Diámetro	98 [mm]	91 [mm]	93 [mm]
Altura	3 [mm]	22 [mm]	80 [mm]
Masa	11 [g]	62 [g]	9 [g]
Área de contacto con el flujo de aire entrante	75.43 [mm ²]	65.04 [mm ²]	67.93 [mm ²]

Tabla de parámetros de los diferentes perdigones experimentados, Gustavo Reyes

Por lo expuesto en la tabla previa, se define que el balance correcto de las características es el del perdigón cilíndrico recubierto, por lo que este se tomó como el perdigón definitivo. Esto significa que presenta un área de contacto más amplia que el segundo modelo pero con una masa significativamente menor. Además, fue primordial la estabilidad presentada al momento de las pruebas, así como su facilidad para ser elevado pero sin ser demasiado ligero para no poder ser controlado.

3.2.4 Hélice

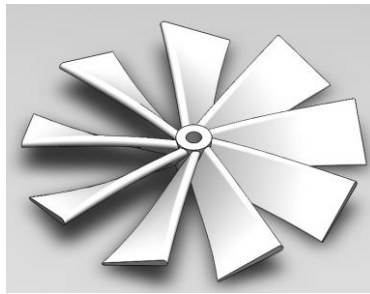
Para que la levitación neumática como tal pueda ejecutarse se necesita ingresar un flujo de aire a una velocidad considerable a la estructura tubular del túnel de viento para levantar el objeto dentro de ella. Esto se logra gracias al efecto que realizan las hélices moviendo masas de aire de su parte frontal a su parte posterior. Este proceso se debe a los momentos que ocurren en ambas partes de la hélice. Sin embargo, definir un momento en función de la masa del aire trasladado se vuelve una tarea para equipamiento sumamente especializado en la parte práctica, por lo que se recurre al cálculo de velocidad máxima o velocidad de punta de pala de una hélice según su diámetro y velocidad de giro, descrito por la ecuación 3.3.

$$v_{max} = \frac{\pi D \omega}{60} \quad \text{Ec. (3.3)}$$

La ecuación indica que la velocidad máxima (v_{max}) de empuje del aire de una hélice está definida por su diámetro (D) multiplicada por su velocidad angular (ω) máxima nominal multiplicado por el número pi (π) y dividido entre 60; esto para una hélice sencilla de dos palas.

Se entiende que mientras menos palas tenga una hélice, más cerca de su velocidad nominal alcanzará la velocidad real del motor que la soporte; sin embargo, la carga que soporta cada una de las palas se incrementa. Por esta razón, y aunque se sacrifique la velocidad de rotación de la hélice, para evitar las vibraciones que esto provoca, se eligió una hélice de 9 palas de 50[mm]. Esta hélice provoca una reducción del ruido notable y un menor esfuerzo del motor, lo que se convierte en esencial al momento de utilizar un motor sin escobillas, con la característica principal del sacrificio del torque para incrementar su velocidad.

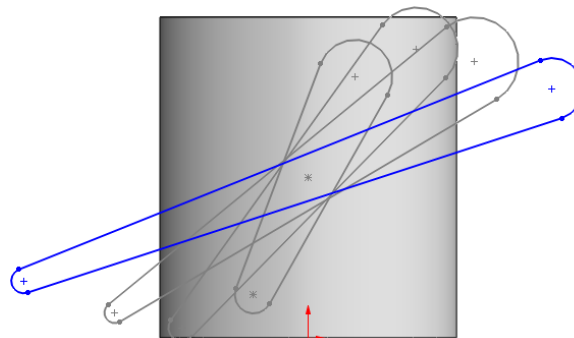
Figura 3.4: Hélice de 9 palas



Diseño de modelo de hélice de ventilador de alta presión estática de paso corto de 9 palas, Gustavo Reyes

El ángulo de pala inicial de esta hélice es de 65° , siendo de paso corto, y su ángulo final es de 19° . La curvatura generada por este cambio de ángulo le entrega la característica de hélice de ventilador de alta presión estática, que provoca una gran fuerza de aire a su salida en forma de huracán. Esta salida de aire que emula un huracán ordena las líneas de proyección y dirección del flujo de aire, reduciendo las perturbaciones que se puedan generar dentro del túnel de viento.

Figura 3.4: Perfil de ángulo de curvatura de pala de hélice



Modelo de perfil de ángulo de curvatura de una pala de hélice de ventilador de alta presión estática, Gustavo Reyes

Para aprovechar en la mayor magnitud posible el área superficial de la hélice y generar más empuje y arrastre, se utilizó una hélice de extremo tipo “Bull Nose” (BN). Esta forma de extremo de hélice representa un mayor consumo de corriente del motor a la que se encuentra unida, además de una ligera baja en la eficiencia, sin embargo incrementa su potencia.

Figura 3.4: Vista de perfil hélice de ventilador de extremo BN



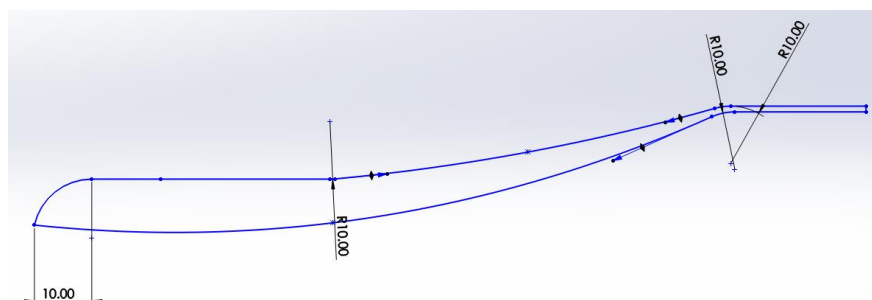
Vista de perfil de hélice de 9 palas de ventilador de presión estática de extremo BN, Gustavo Reyes

3.2.5 Electric ducted fan

Para un ordenamiento más estructurado del flujo de aire y un incremento de su empuje se utilizó la inclusión del sistema de propulsión dentro de una EDF (Electric Ducted Fan). Este mecanismo es un conjunto de un motor eléctrico en cuyo eje se monta una hélice, siendo todo albergado en una carcasa cilíndrica llamado ducto. De manera general, en una estructura de tipo ducto, se utilizan hélices con más palas de longitud relativamente corta que si el sistema de propulsión estaría libre. Esta última característica es consistente con lo expuesto respecto al desarrollo de la hélice de 9 palas de 50[mm].

La estructura de la carcasa o ducto de este pequeño subsistema tiene como parámetros un diámetro mayor, que es el del agujero por donde ingresa el aire, y un diámetro inferior que es por el lugar donde se traslada ya al túnel de viento. Además, presenta un grosor variable, que se puede denominar perfil de pluma, que al ser el director de un sólido de revolución provoca que en la entrada de la estructura la presión en el extremo de la pala de hélice se reduzca en gran magnitud.

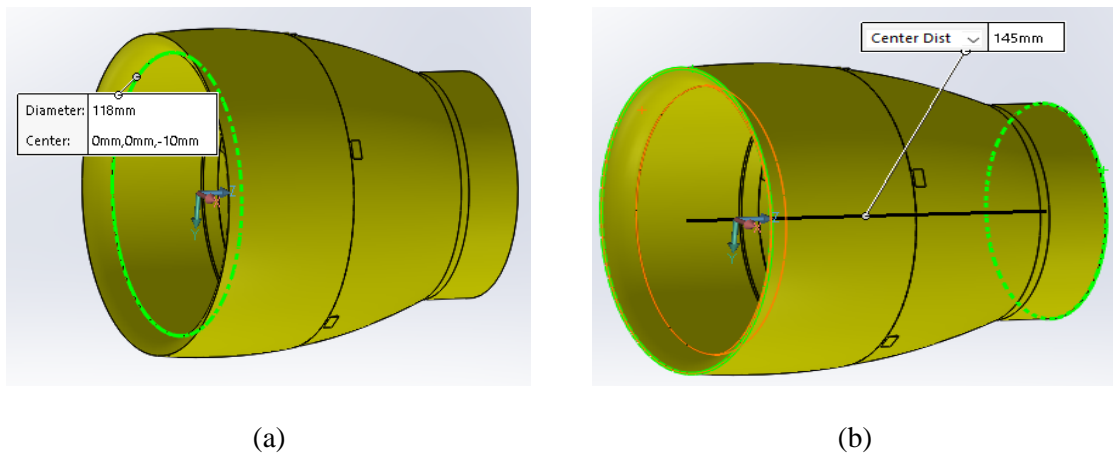
Figura 3.4: Perfil de pluma para ducto de EDF



Perfil de pluma para sólido de revolución constituyente de ducto o carcasa de EDF, Gustavo Reyes

Para el diseño de esta estructura se consideró un diámetro interior mayor de 118[mm] y diámetro exterior menor de 10[mm], como muestra la figura 3.11. La curvatura de la entrada del ducto presenta un radio de 10.25[mm], correspondiente a un incremento de 2.5% respecto a los radios de curvatura del radio medio y radio final exterior e interior. El radio exterior menor corresponde a la medida del tubo de polimetilmetacrilato. El grosor menor es de 1[mm], pasando a un grosor mayor de 10[mm], para así lograr un incremento del empuje del flujo de aire en una razón de 1 a 10. La longitud es dependiente de la altura de motor y hélice que debe ser albergado en su interior, dando una resultante de 145[mm].

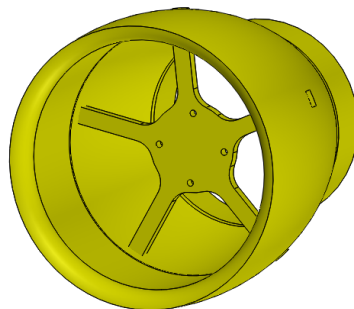
Figura 3.4: Medidas principales carcasa EDF



(a) Diámetro mayor interior carcasa EDF. (b) Longitud central carcasa EDF, Gustavo Reyes

Dentro de la estructura, tomando como referencia la parte superior, es decir el diámetro menor, a una distancia de 86[mm] se colocó un soporte en forma pentagonal para incluir el motor sin escobillas. La figura 3.12 muestra lo descrito previamente.

Figura 3.4: Carcasa EDF



Carcasa de cubierta o ducto componente de una EDF con soporte interno para motor, Gustavo Reyes

3.2.6 Motor sin escobillas

Como elemento de tracción principal se utilizó un motor sin escobillas A2212/6T (ver figura 3.13) con corriente máxima de carga de 21.5 Amperios [A]. Según esta corriente, para el control de este motor, es necesario dimensionar el controlador electrónico de velocidad (ESC) a, mínimo, 22 [A]. Al considerar una banda de tolerancia para funcionamiento de 40% por, el consumo de corriente de arranque y la disponibilidad comercial, se utilizó un ESC de 30[A].

Figura 3.4: Motor sin escobillas

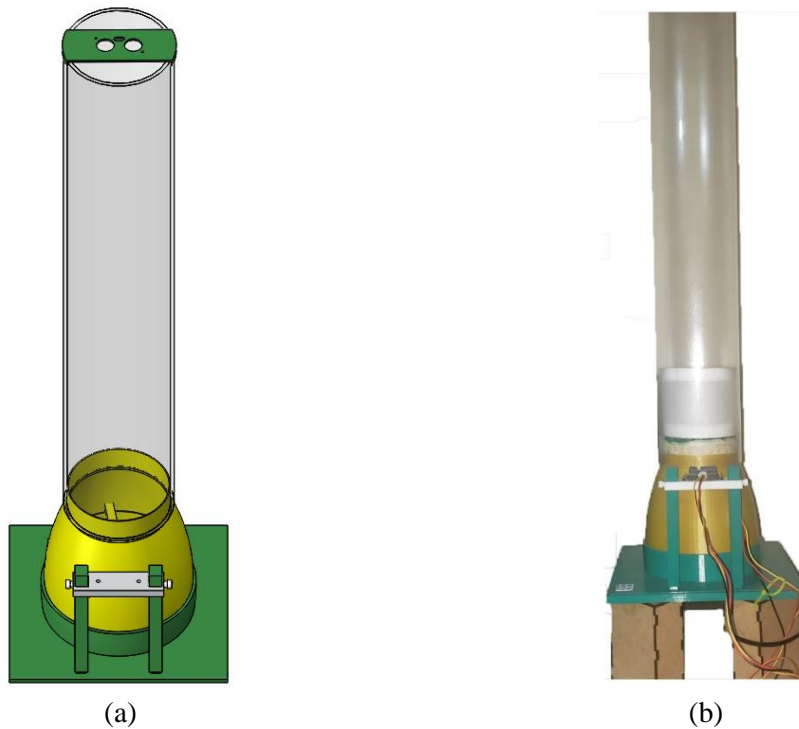


Motor sin escobillas A2212/6T 2200KV de 3 fases, ABRA (2022)

3.2.7 Diseño mecánico integral

Con la implementación de los elementos indicados y los respectivos soportes desarrollados en manufactura aditiva y sustractiva para darle altura al sistema y albergar los elementos adicionales, se puede apreciar el modelo que se construyó de túnel de viento vertical a pequeña escala en la figura 3.14.

Figura 3.4: Diseño mecánico y modelo construido del túnel de levitación neumática

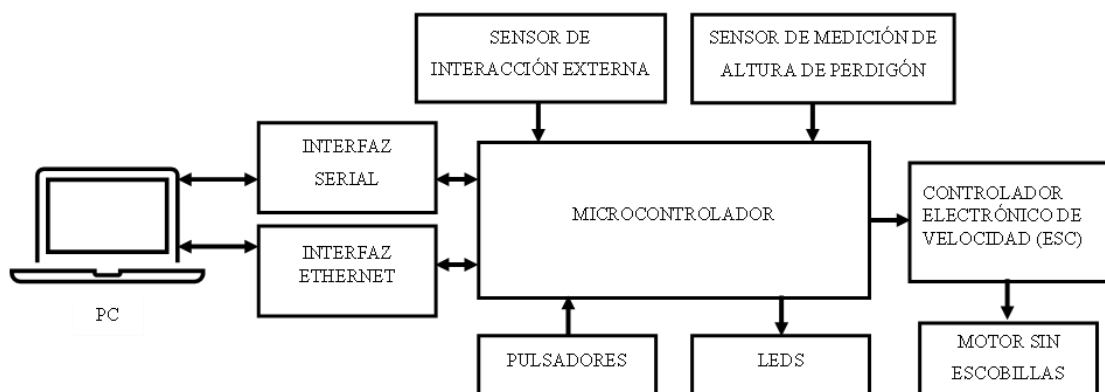


(a) Diseño mecánico de modelo de túnel de levitación neumática con EDF y soporte para sensores.
 (b) Modelo construido de túnel de levitación neumática con EDF y soporte para sensores, Gustavo Reyes

3.3 Desarrollo electrónico

Para la implementación de la parte electrónica se tomaron en cuenta los elementos necesarios para el funcionamiento de la planta, comprendidos por el microprocesador y los periféricos de entrada y salida. Este esquema se puede observar en la figura 3.15.

Figura 3.4: Esquema de la planta del sistema propuesto



Esquema de elementos constituyentes del modelo de túnel de levitación neumática, Gustavo Reyes

Como microcontrolador se utilizó un TM4C1294NCPDT basado en una unidad de procesamiento Arm Corte-M4F de 120[MHz], albergado en una tarjeta de desarrollo Connected LaunchPad TM4C1294 Tiva C Series. La interfaz serial y ethernet utilizadas para la comunicación con su respectivo software en la PC, son aquellas incluidas en la misma tarjeta de desarrollo. Los pulsadores y los LEDs utilizados también están presentes dentro de la misma tarjeta, para de esta manera utilizar la mayor cantidad de prestaciones posibles.

3.3.1 Control electrónico de velocidad

Para el uso del motor sin escobillas detallado en el apartado anterior, se necesitó un controlador electrónico de velocidad (ESC), cuya estructura interna incorpora un microcontrolador, una etapa de potencia compuesta por transistores tipo FET y un regulador de voltaje. El consumo de corriente máximo del motor es de 21.5 [A], que al considerar una tolerancia de 40% adicional para los picos de corriente, requiere un ESC de 30[A]. El circuito de este dispositivo está estructurado, de manera general, como se puede observar en la figura 3.16.

Figura 3.4: Circuito general ESC

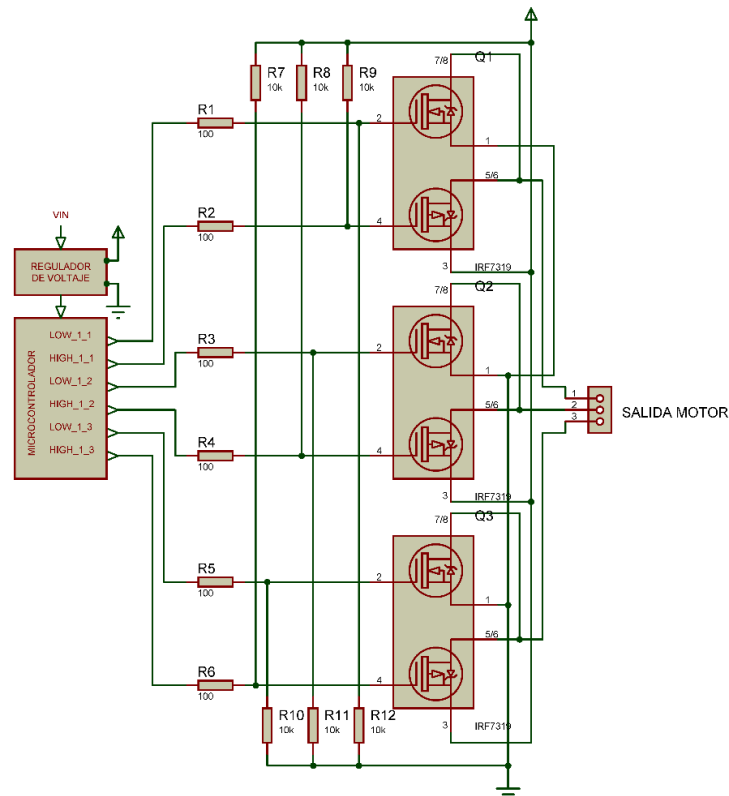


Diagrama esquemático general de un controlador electrónico de velocidad para motor sin escobillas, Gustavo Reyes

Es común que los motores sin escobillas con sus respectivo ESCs sean manejados mediante controles de radiofrecuencia que emiten una señal de modulación de ancho de pulso (PWM) con período de 20 milisegundos. Dentro de esta franja de período, se necesitan pulsos de entre 1 y 2 milisegundos. Para conocer el comportamiento del conjunto de motor y controlador de velocidad elegidos específicamente, se desarrolló un proceso de caracterización rápida, teniendo como resultado los parámetros de operación indicados en la tabla 3.2.

Tabla 3.2: Parámetros de operación ESC y motor sin escobillas

Parámetro	Valor
Período de la señal PWM	20000 [μ s]
Ancho de pulso de inicialización	1000 [μ s]
Ancho de pulso crítico de inicio	1650 [μ s]
Ancho de pulso para velocidad máxima	2000 [μ s]

Tabla de valores de operación según caracterización de ESC y motor sin escobillas, Gustavo Reyes

Para que sea posible el uso de un motor sin escobillas con su respectivo ESC, es necesario seguir un procedimiento específico para su preparación previa. Este proceso debe iniciar por el control con un pulso de 1000[μ s] seguido de una espera de 5 segundos, para permitir ya el uso continuo según la velocidad que sea requerida.

El parámetro del ancho de pulso es uno de los más críticos, ya que esta es la variable manipulada del sistema para el correcto funcionamiento de la planta. Al ser la variable manipulada, su valor será el que se modifique directamente por acción del controlador (ver figura 3.17).

Figura 3.4: ESC 30[A]



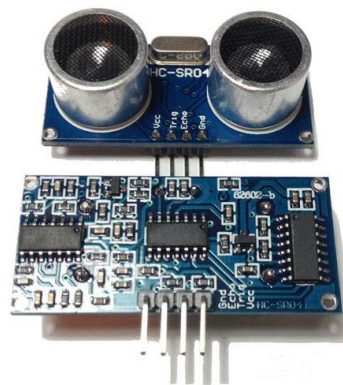
Controlador electrónico de velocidad de 30[A], Talos Electronics (2022)

3.3.2 Sensor ultrasónico de distancia

Al necesitar una variación de la altura del perdigón que se desplaza a lo largo del túnel de levitación neumática, es necesario conocer su posición. Esta posición se conoce gracias a la acción de un sensor de distancia, en este caso de tecnología ultrasónica, en específico el modelo HC-SR04 mostrado en la figura 3.18.

El sensor HC-SR04 actúa al enviar un pulso ultrasónico y esperar su retorno por rebote en un objeto. El parámetro que se interpreta es el tiempo transcurrido entre el envío del pulso y su retorno. Para este proceso, el sensor cuenta con un emisor y un receptor de ultrasonido albergados en una placa de circuito impreso (PCB, por sus siglas en inglés) junto con un circuito de control.

Figura 3.4: Sensor HC-SR04



Sensor ultrasónico HC-SR04 en PCB con circuito de control, Diosdado (2014)

El circuito de control cumple la función de interpretar una señal recibida desde el microcontrolador, que esté en alto por al menos $10[\mu\text{s}]$ (trigger), y provocar el envío de un tren de 8 pulsos a una frecuencia de $40[\text{kHz}]$, además de colocar su salida (echo) en alto. Una vez que la onda rebota y su eco es recibido por el sensor, el circuito de control envía la señal al microcontrolador para detener el conteo del tiempo transcurrido. El rango de medida teórico de este sensor es de $2[\text{cm}]$ a $400[\text{cm}]$, pudiendo variar según las condiciones bajo las que se esté utilizando. Para conocer la distancia a la que se encuentra el objeto en función del tiempo entre envío y retorno del pulso ultrasónico, es necesario hacer una conversión en función de la velocidad del sonido. En condiciones normales, en promedio, la velocidad del sonido en el aire tiene un valor de $343.2 [\text{m/s}]$. Para lograr implementar esto en un microcontrolador, y según las condiciones de la planta, es necesario cambiar la velocidad a

[cm/ μ s] y tener en cuenta que la distancia que recorre la onda de sonido es de ida y vuelta hacia el objeto, por lo que se debe tomar únicamente la mitad del recorrido. En la ecuación 3.4 se puede observar que, la distancia medida desde el sensor al objeto es igual a la velocidad del sonido en el aire convertida a [cm/ μ s] multiplicada por el tiempo de vuelo medido dividido entre 2.

$$d = 343.2 \left[\frac{m}{s} \right] \times \frac{100[cm]}{1[m]} \times \frac{1[s]}{1000000[\mu s]} \times \frac{t_{vuelo}}{2} = 0.01716 t_{vuelo} \quad \text{Ec. (3.4)}$$

La altura (distancia vertical) medida con este sensor se establece como la variable controlada del sistema de control, por lo que es uno de los parámetros que se mantendrá en el desarrollo de los tres métodos de mando propuestos. Al ser esta la variable controlada, también es aquella magnitud a la que se referencia en la consigna, set point u objetivo del controlador.

3.3.3 Sensor infrarrojo de distancia

Para el método de mando propuesto por interacción humana con sensor externo, es necesario establecer el proceso a seguir. Se tiene un sensor infrarrojo colocado junto al túnel de viento (recorrido del perdigón) que mida la distancia a la cual se coloque la mano, o algún objeto que lo simule. Esta medida tomada se convierte en la referencia (set point) del controlador y su señal debe ser tratada de tal forma que su adquisición sea inmediata y prioritaria.

Para esta función se seleccionó un sensor infrarrojo GP2Y0A41SK0F de la marca Sharp (ver figura 3.19), con un rango de medición de 4[cm] a 30[cm]. Este sensor tiene una salida analógica que permite la medición de distancias por su combinación del detector sensitivo de posición (PSD), un diodo emisor de luz infrarroja (IR LED) y un circuito de procesamiento de señal. Este sensor utiliza el método de triangulación para la medición de distancias, por lo que la temperatura ambiental y el tiempo de operación no se presentan como factores influyentes en su proceso de medición.

El sensor tiene un voltaje de operación de 5[VDC], con lo que se tiene una diferencia de voltaje de salida promedio de 2.3[V]. Esta diferencia de voltaje llega hasta un máximo de alrededor de 3[V], por lo que es posible su medición a través del módulo de lectura analógica de la tarjeta Tiva TM4C1294, que soporta un nivel de 3.3[V].

Figura 3.4: Sensor SHARP infrarrojo 4 a 30[cm]



Sensor de medición de distancia infrarrojo SHARP GP2Y0A41SK0F, Farnell Components SL.
(2013)

Para lograr obtener la relación entre la lectura del conversor análogo digital (ADC) de la tarjeta de desarrollo (resolución de 12 bits) y la distancia medida, se realizaron pruebas de caracterización del sensor. Se tomaron alrededor de 200 muestras para cada distancia entre 4[cm] y 30[cm] con pasos de 1[cm], y con el promedio de los valores de cada distancia se obtuvo la relación entre distancia medida y valor del ADC. En la tabla 3.3 se pueden observar los datos obtenidos de este proceso de caracterización.

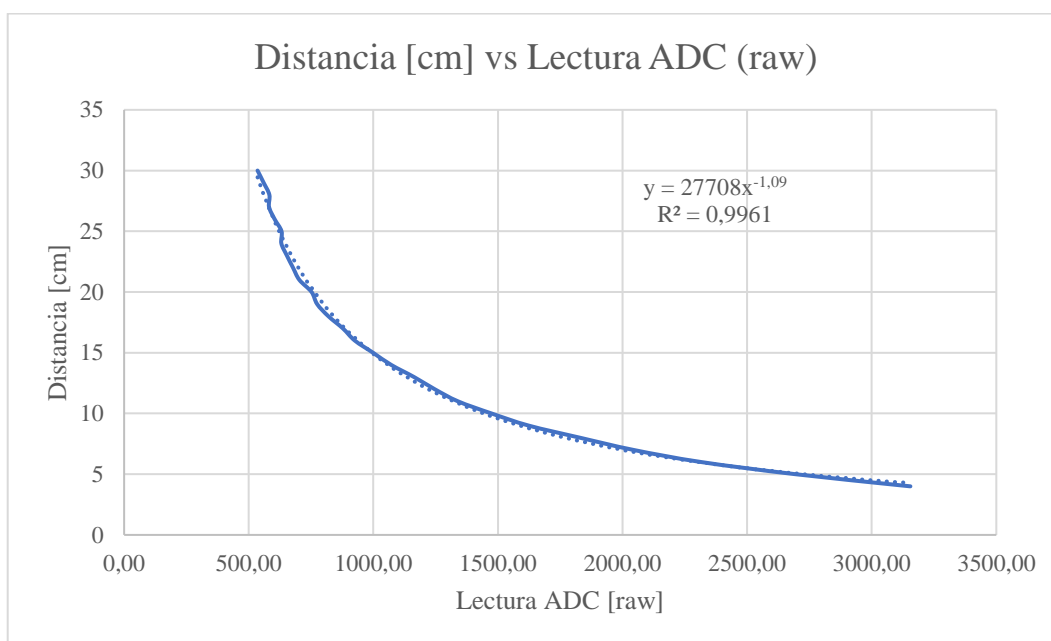
Tabla 3.3: Datos de caracterización sensor infrarrojo

Lectura ADC (raw)	Distancia [cm]	Lectura ADC (raw)	Distancia [cm]
3155.24	4	819.98	18
2693.92	5	774.65	19
2315.49	6	752.31	20
2043.40	7	703.18	21
1829.83	8	677.81	22
1623.69	9	653.17	23
1474.22	10	630.95	24
1339.63	11	631.69	25
1246.08	12	603.79	26
1163.49	13	581.56	27
1071.59	14	582.84	28
999.02	15	559.82	29
925.79	16	535.41	30
878.06	17		

Tabla de valores de caracterización de sensor SHARP infrarrojo GP2Y0A41SK0F para ADC de 12 bits y referencia de 3.3[V] sin acondicionamiento intermedio, Gustavo Reyes

Con los datos obtenidos y mediante el uso del software de manejo de hojas de cálculo Microsoft Excel, se obtuvo la curva característica de los datos de la tabla 3.3. Además, se incluyó el gráfico de distancia vs. dato ADC con una línea de tendencia exponencial y el valor de R^2 (coeficiente de determinación). En la figura 3.20 se puede observar el gráfico mencionado con su respectiva ecuación.

Figura 3.4: Gráfica distancia vs. lectura ADC



Gráfica “Distancia vs. Lectura ADC” de los datos de caracterización del sensor SHARP GP2Y0A41SK0F, Gustavo Reyes

A partir de la gráfica 3.20 se puede definir que el coeficiente de determinación R^2 llega a un valor de 0.9961, que garantiza que la ecuación mostrada tiene un ajuste del modelo muy confiable. Además, se puede establecer que el dato expresado como “y” es la distancia medida, y “x” es la lectura obtenida desde el ADC en datos raw (crudos). La expresión de la ecuación de ajuste de la línea de tendencia de la gráfica se puede observar en la ecuación 3.5.

$$distancia = 27708 (lectura_ADC)^{-1,09} \quad \text{Ec. (3.5)}$$

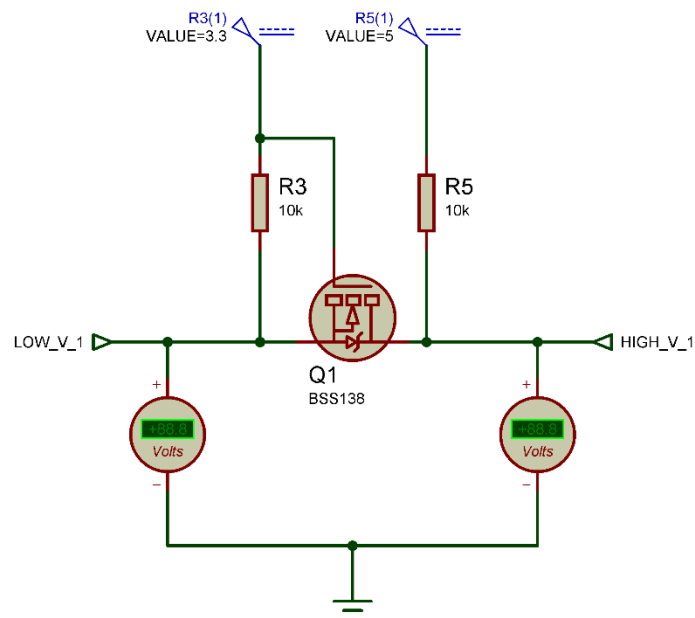
Estos datos son los que se utilizarán en el desarrollo del programa para la obtención y tratamiento de los datos necesarios.

3.3.4 Conversor de nivel lógico

Como parte central de los componentes electrónicos se tiene la tarjeta Tiva TM4C1294. Esta es una tarjeta de desarrollo con tres conjuntos de pines de conexión además de los periféricos y dispositivos de comunicación ya incluidos. A pesar de tener reguladores de voltaje tanto para 5[V] como para 3.3[V], el microcontrolador TM4C1294 solo tiene dos pines que toleran una entrada en nivel lógico de 5[V] y son los que tienen funcionalidades especiales al ser huésped USB (Krakow, Johnson, & Guy, 2013). Por esto, es necesario que las señales de los periféricos sean cambiadas al nivel lógico de 3.3[V].

Las señales de trigger y echo del sensor ultrasónico y la señal de control del controlador electrónico de velocidad del motor sin escobillas están en el nivel lógico TTL (lógica transistor transistor) de 5[V]. Estas señales deben ser trasladadas al nivel lógico de 3.3[V], para lo que se utiliza una configuración de transistor y resistencias, tomando en cuenta que como son de entrada y salida, el conversor debe ser bidireccional. En la figura 3.21 se puede observar el esquema del circuito necesario para este cambio de nivel lógico.

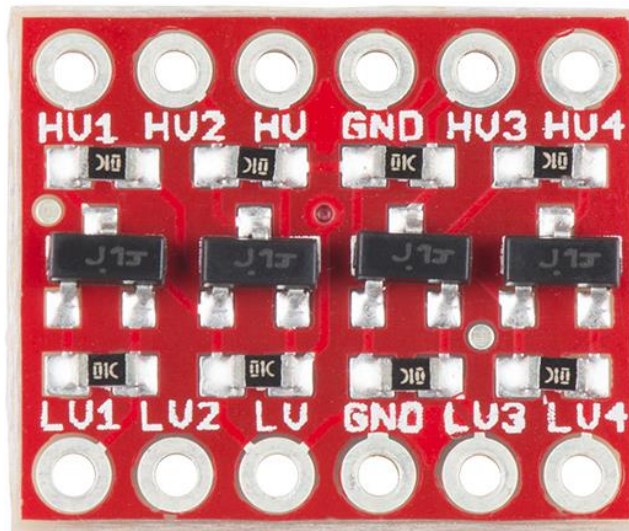
Figura 3.4: Conversor de nivel lógico



Esquemático circuito de conversión de nivel lógico bidireccional, Gustavo Reyes

Gracias a este circuito auxiliar se puede integrar los periféricos de un nivel lógico al nivel lógico de la tarjeta de desarrollo. Al ser requerida la conversión de tres señales, se necesita replicar el circuito de la figura 3.21 tres veces. Por esta razón, se consideró utilizar una placa de circuito impreso (PCB), que en su opción comercial ocupa un espacio reducido sin sacrificar ningún aspecto de su funcionalidad. Basado en un análisis de disponibilidad, se eligió el convertidor de nivel lógico bidireccional de la marca Sparkfun, que se puede observar en la figura 3.22.

Figura 3.4: Convertidor de nivel lógico Sparkfun



Conversor de nivel lógico bidireccional Sparkfun de 4 canales, SparkFun (2014)

La conversión de nivel lógico no fue necesaria realizar a la señal analógica del sensor infrarrojo SHARP ya que su rango de voltaje de salida es tolerado por la entrada del módulo ADC del microcontrolador de la tarjeta Tiva TM4C1294.

3.3.5 Fuente de alimentación

Para alimentar la etapa de potencia de la planta es necesario asegurar una fuente de corriente continua que satisfaga el consumo de corriente del motor sin escobillas (ver figura 3.23). Al tener un controlador electrónico de velocidad (ESC) de 30[A], se estableció la necesidad de una fuente que entregue la misma cantidad de corriente. El voltaje de operación del motor sin escobillas se indicó en 11.1[V], sin embargo ese es un valor normalizado para las baterías de polímeros de litio por celdas, que al utilizar otro tipo de fuente se puede aproximar a 12[V].

La fuente, además de las características mencionadas, es necesario que no exceda su tamaño y pueda ser conectada a una alimentación de 110[V] de corriente alterna para su disponibilidad de operación bajo condiciones regulares, o sencillo de lograr. Por esto, se eligió una fuente conmutada, por su eficiencia promedio de alrededor del 80%, bajo consumo de potencia y tamaño y peso reducido por ausencia de un transformador, lo que provoca también un costo inferior a una fuente lineal de las mismas características.

Figura 3.4: Fuente 12VDC 30A



Fuente de alimentación conmutada 110VAC - 12VDC 30A, Unit Electronics (2016)

Esta fuente se conecta al ESC, ya que la alimentación de la tarjeta Tiva TM4C1294 se realiza a través de la conexión USB, que a su vez se utiliza para la comunicación serial. Además, la parte de control de la ESC debe ser conectada únicamente los pines de señal de control y tierra (GND).

3.3.6 Integración de componentes electrónicos

Para la integración de las conexiones de los elementos electrónicos de la planta, se desarrolló una placa de circuito impreso (PCB). Esta PCB incluye el módulo de conversión de nivel lógico, los conectores de los sensores y los conectores para la tarjeta Tiva TM4C1294. Esto se desarrolló tomando en cuenta los pines a utilizarse en la tarjeta de desarrollo, cuyo detalle y correspondencia se puede ver en la tabla 3.4.

Tabla 3.4: Correspondencia pines periféricos con tarjeta de desarrollo

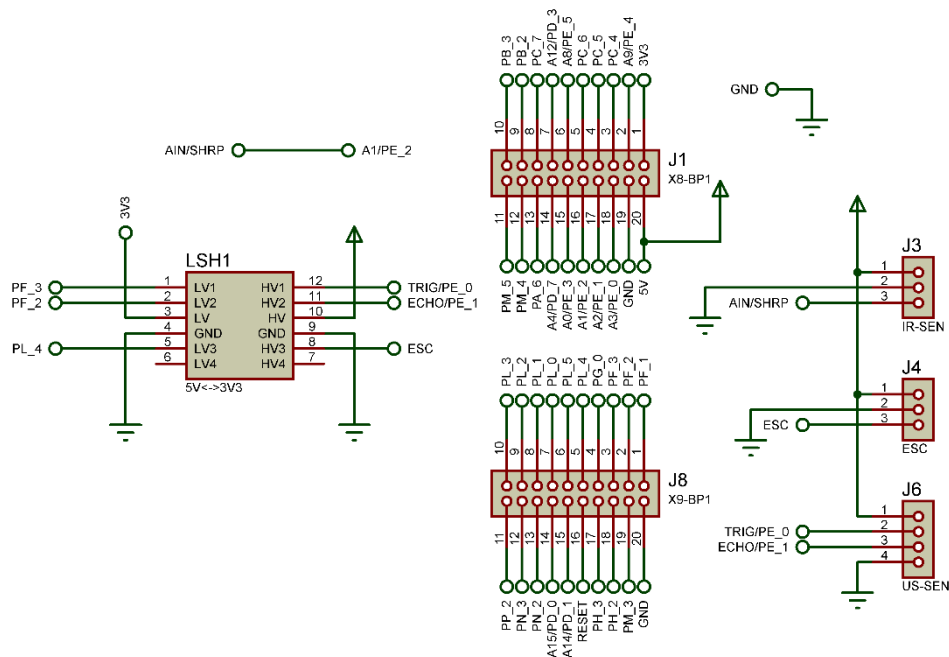
Correspondencia de pines			
Elementos Periféricos		Tarjeta de Desarrollo	
Elemento	Pin	Elemento	Pin
ESC	Signal	TM4C1294	PL_4
HC-SR04	Trigger		PF_3 (PIN 38)
HC-SR04	Echo		PF_2 (PIN 39)
GP2Y0A41SK0F	Analog Out		PE_2

Correspondencia de pines de elementos electrónicos periféricos en la tarjeta de desarrollo Tiva TM4C1294, Gustavo Reyes

Para efectos de practicidad, en la tabla 3.4, de correspondencia de pines de conexión, no se tomó en cuenta la acción intermedia del convertor de nivel lógico.

Para el desarrollo del circuito esquemático se tomó en cuenta los conectores de la tarjeta Tiva TM4C1294, y se colocaron los conectores de los sensores y el ESC. Este circuito se puede observar en la figura 3.24.

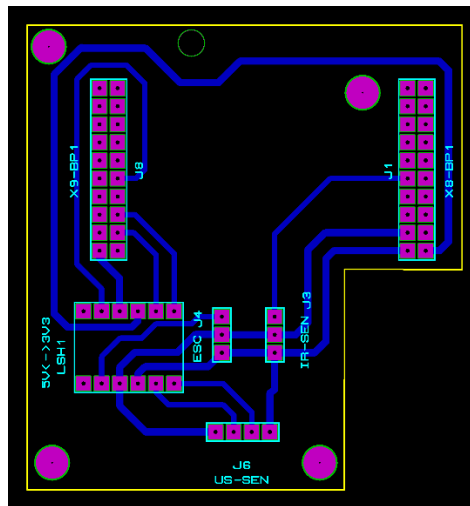
Figura 3.4: Diseño esquemático de PCB de conexión



Diseño de circuito esquemático para conexión de periféricos a tarjeta Tiva TM4C1294 con convertor de nivel lógico, Gustavo Reyes

Para el desarrollo de la PCB se tomó en cuenta la forma y la ubicación de los conectores de la tarjeta Tiva TM4C1294, además de la ubicación de los conectores de los sensores, para que sean accesibles y que se puede usar a manera de “shield” o placa acoplable directamente a la tarjeta de desarrollo. Este diseño se puede observar en la figura 3.25.

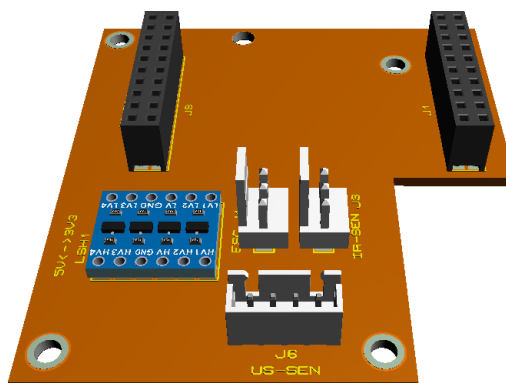
Figura 3.4: Diseño PCB



Diseño de PCB para conexión de periféricos a tarjeta Tiva TM4C1294 con convertor de nivel lógico, Gustavo Reyes

En la figura 3.26 se puede observar el diseño en tres dimensiones de la PCB de conexiones. Además, esta PCB se desarrolló tomando en cuenta los elementos Through Hole o tradicionales y su fabricación se realizó mediante la técnica de termo transferencia y revelado.

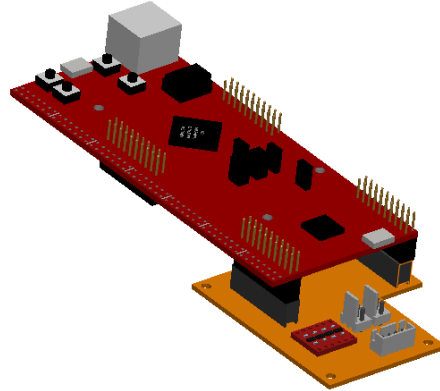
Figura 3.4: Vista 3D de PCB de conexiones



Vista 3D de PCB para conexión de periféricos a tarjeta Tiva TM4C1294 con convertor de nivel lógico, Gustavo Reyes

Una vez colocadas en conjunto, la vista de la tarjeta de desarrollo y su PCB tipo shield se puede observar en la figura 3.27.

Figura 3.4: Vista 3D de Tiva con shield para conexión de periféricos



Vista 3D de Tiva TM4C1294 con shield para conexión de periféricos, Gustavo Reyes

Para referencia, en la figura 3.28 se puede ver el diagrama general de todas las conexiones necesarias para el funcionamiento de la planta. En esta se logra observar los elementos eléctricos, electrónicos y mecánicos activos; se omite la presencia de las estructuras mecánicas como soportes, tubo, y demás.

Figura 3.4: Diagrama funcional de la planta

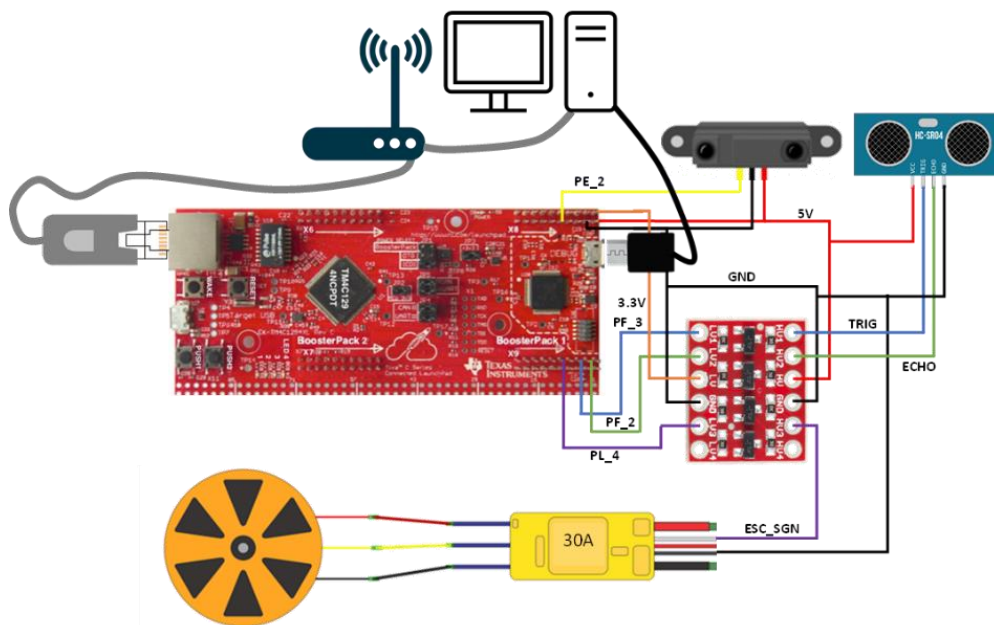


Diagrama de conexión electrónico funcional de la planta, Gustavo Reyes

3.4 Función de transferencia

Mediante la obtención de datos experimentales de la variable controlada (altura del perdigón), y la variable manipulada (ancho de pulso del ESC), se procedió a la identificación de la función de transferencia. Este proceso se realizó con la ayuda del software Matlab y su herramienta System Identification.

Se tomaron un total de 492 datos, en los que se varió el ancho de pulso de la ESC, y se tomaron mediciones de la altura del perdigón mediante el sensor ultrasónico. Esto se incorporó en el software Matlab a través de la lectura de un archivo de hoja de cálculo, y se lo incluyó en el complemento de identificación de sistema. Con esto, se estimó la función de transferencia de la planta, para obtener la que se puede observar en la ecuación 3.6.

$$\frac{-1.93s - 0.004464}{s^2 + 0.4959s + 3.311 \cdot 10^{-10}} \quad \text{Ec. (3.6)}$$

La función de transferencia identificada consta de 2 polos y 1 cero, con 4 términos independientes. Con esto, se establece que el diagrama de bloques es el que se muestra en la figura 3.29.

Figura 3.4: Diagrama de bloques de planta

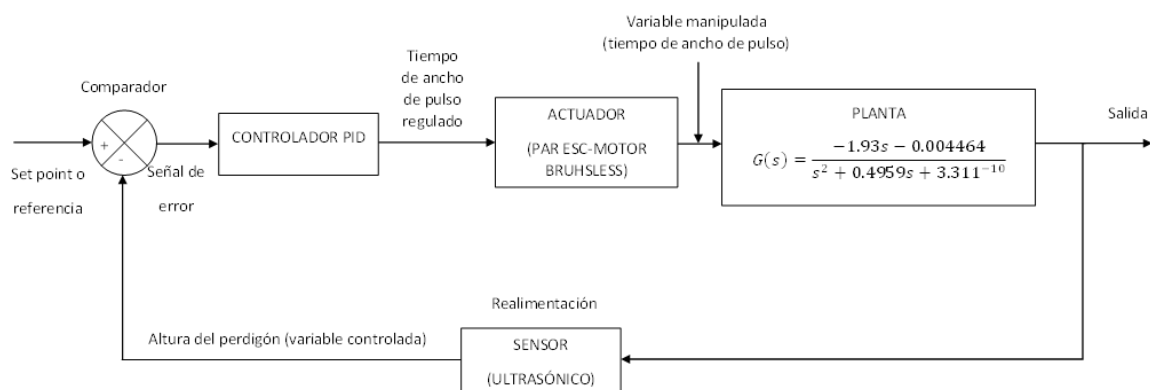


Diagrama de bloques del proceso de controlador de la planta, Gustavo Reyes

3.5 Programación

La programación se desarrolló en el entorno de programación Energia IDE, dirigido para las tarjetas de desarrollo tipo Launchpad de National Instruments, entre ellas la Tiva

TM4C1294. En esta parte se desarrolló el algoritmo de control de la planta desarrollada, estando dividido en tres partes, correspondiente cada una a un método de mando propuesto.

3.5.1 Mando por posiciones definidas

El mando de la planta por posiciones definidas proporcionales al recorrido total del tubo se centra en la ubicación del perdigón en el lugar indicado mediante comandos enviados por el monitor serial de Energia IDE.

Para lo indicado, el proceso consta de 4 pasos esenciales. Se inicia por el establecimiento de la comunicación serial, luego se procede a la lectura del dato enviado por el puerto serial. Posteriormente está la ejecución del algoritmo del proceso de control, y finalmente se pasa el resultado del control a la ejecución del actuador. El proceso, de manera simplificada, se puede observar en el diagrama de flujo de la figura 3.30.

Figura 3.4: Diagrama de flujo de mando por posiciones definidas

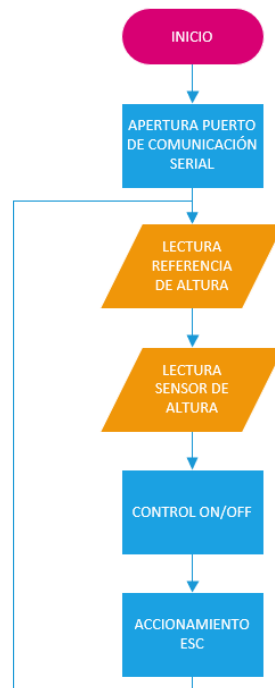


Diagrama de flujo control de perdigón por posiciones definidas mediante comunicación serial,
Gustavo Reyes

El proceso empieza con el establecimiento de la comunicación serial entre la PC y la tarjeta Tiva mediante el cable de conexión USB a la tasa de baudios de comunicación compatible entre la tarjeta y el IDE.

Posterior al establecimiento de la comunicación, es necesario realizar la lectura del valor ingresado desde el monitor serial de Energia. Para esto, se comprueba la disponibilidad del puerto, seguido de la lectura del dato, y, por último su conversión a dato entero para poder ser tratado. Este segmento del código se puede observar en la figura 3.31.

Figura 3.4: Código fuente lectura serial

```
void setup() {  
  Serial.begin(9600);  
}  
  
void loop() {  
  if(Serial.available() > 0){  
    String dato = Serial.readStringUntil ('\n');  
    int altura = dato.toInt();  
  }  
}
```

Código fuente para lectura de datos a través del monitor serial de Energia IDE, Gustavo Reyes

Luego de esto, es necesario obtener el dato de la altura a la que se encuentra el perdigón dentro del tubo, lo que se realiza mediante la lectura desde el sensor ultrasónico HC-SR04. Para lo cual, es necesario la definición de los pines del sensor, Trigger y Echo, que se colocaron en los pines de la tarjeta Tiva 38 y 39, respectivamente (ver tabla 3.4). Además, se debe definir al pin del trigger como salida y al de echo como entrada. La definición de los pines se realiza previo a las funciones del código, y la definición de los pines dentro de la función *setup*. Dentro de la función *loop* se coloca el código de lectura del sensor, que cumple los siguientes pasos:

- Pin trigger a nivel bajo durante 2 microsegundos
- Pin trigger a nivel alto durante 10 microsegundos
- Pin trigger a nivel bajo
- Obtención del tiempo hasta estado alto del pin echo mediante la función *pulseIn*
- Conversión del tiempo medido a distancia, mediante lo expresado en la ecuación 3.4.

Si bien la lectura del sensor estaría completada, para reforzar el algoritmo y tener una lectura más limpia del dato de la altura, se incluyó un filtro de media móvil. Este es un filtro digital de fácil implementación y que no sacrifica de manera sustancial la eficiencia del algoritmo.

Este filtro se vale de un buffer circular que recorre sus elementos del final al inicio; en este caso el tamaño del buffer se estableció en 20. Para esto, se implementó el código que se puede observar en la figura 3.32.

Figura 3.4: Lectura distancia HC-SR04

```

const int echoPin = 39;
const int trigPin = 38;

long distance;
long duration;

const int nbaca=20;
int baca[nbaca];
int indice=0;
int total=0;
int media=0;

void setup() {
  pinMode(trigPin,OUTPUT);
  pinMode(echoPin,INPUT);
}

void loop() {
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  duration = pulseIn(echoPin, HIGH);
  distance = 0.01716*duration;

  total = total - baca[indice];
  baca[indice] = distance;
  total = total + baca[indice];
  indice = indice+1;
  if(indice >= nbaca){
    indice = 0;
  }
  media = total/nbaca;
}

```

Definición pines tarjeta Tiva

Definición de variables de medida sensor HC-SR04

Definición de variables y constantes de filtro promedio

Lectura dato distancia de sensor HC-SR04

Aplicación filtro de media móvil

Código fuente para lectura de distancia sensor HC-SR04 con filtro de media móvil, Gustavo Reyes

Por último, para lograr medir la altura del perdigón dentro del tubo, es necesario convertir la altura obtenida en valores del 0% al 100% de recorrido del tubo. Por mediciones ejecutadas, se conoce que la distancia medida cuando el perdigón está en su punto más bajo, la medida de distancia es de 58[cm], mientras que en el extremo superior del tubo es de 4[cm]. Con estos datos, y utilizando el comando *map*, se logró realizar el cambio de referencia y guardarlo en una variable que funciona como la entrada del sistema de control. Como consideración general, todas las operaciones respecto a valores de distancia, altura, o similares se realiza con una precisión de 1[cm].

Para la ejecución del algoritmo de control (o controlador) que rige la planta es necesario tener un valor de salida respecto a la entrada obtenida. Para este primer tipo de mando, se utilizó un control por histéresis. Se entiende que un control por histéresis es de tipo ON/OFF

o Todo/Nada, es decir, el actuador de la planta se encenderá para alcanzar a ingresar a la ventana de histéresis definida y se apagará si sale de esa ventana por su umbral superior.

Para este tipo de control, se definió un umbral de 2[cm] de tolerancia y dos condicionales para definir la salida del sistema. En el primer condicional se evalúa si la entrada del sistema (altura del perdigón) es mayor a la altura deseada (dato obtenido desde el monitor serial) más el umbral de histéresis; en caso de ser verdad, la salida será la velocidad mínima para ser asignada al motor sin escobillas. La segunda estructura condicional evalúa si la entrada del sistema es menor a la altura deseada menos el umbral; si es verdadero, la salida será la velocidad máxima del motor. Este tipo de control se puede observar en la figura 3.33.

Figura 3.4: Código control por histéresis

```
int threshold=2;

void setup() {
}

void loop() {
  if(input>height+threshold){
    output=vel_min;
  }
  if(input<height-threshold){
    output=vel_max;
  }
}
```

Código fuente control por histéresis Energia IDE, Gustavo Reyes

Para finalizar el control por este método de mando, es necesario asignar la velocidad al ESC para que comande el motor sin escobillas, esto se realiza mediante la escritura de un pulso, según lo indicado en el apartado 3.3.1. Esto se realiza con la ayuda de la librería *Servo*, que permite generar de manera sencilla el pulso necesario para la activación del ESC. Para utilizar esta librería se crea su respectivo objeto y se asigna el pin en el que está conectado. Previo al funcionamiento continuo, es necesario inicializar el motor sin escobillas, lo que se logra al enviar un pulso de 1[ms] con una espera de al menos 5[s]; luego de lo que es posible asignar la velocidad obtenida de la salida del controlador. El código necesario para ejecutar lo mencionado se puede observar en la figura 3.34.

Figura 3.4: Código ESC librería servo

```
#include <Servo.h>

Servo ESC;

void setup() {
  ESC.attach(59);
  ESC.writeMicroseconds(1000);
  delay(5000);
}

void loop() {
  ESC.writeMicroseconds(ouput);
}
```

Código fuente manejo motor sin escobillas mediante ESC con librería servo, Gustavo Reyes

3.5.2 Mando por sensor externo

En este mando, la altura deseada del perdigón dentro del túnel de viento viene dada por la distancia medida por un sensor infrarrojo que mide la altura a la que se coloca la mano en su rango de medición. Además, se integra un controlador PID para la acción de control y así mantener un comportamiento fluido y estable del perdigón dentro del túnel.

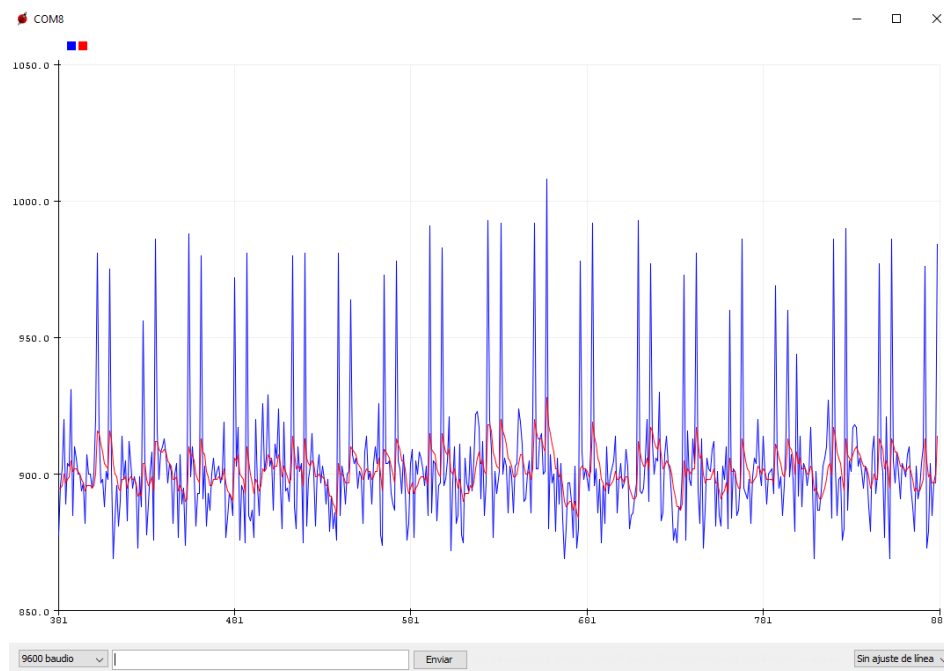
Para la lectura de la altura deseada, que va a convertirse en la referencia del controlador PID, es necesario obtener los datos del sensor SHARP GP2Y0A41SK0F a través del módulo convertidor análogo digital (ADC) de la tarjeta Tiva. Durante el proceso de prueba de lectura de datos, a pesar de mantener estático el objeto en el rango de medición, se presentan datos inestables, para lo que es necesario implementar un filtro a la señal. A manera de filtro por hardware, se coloca un capacitor de desacoplamiento de la señal, complementado con un filtro digital por software de tipo pasa bajo EMA (exponencial de media móvil).

Este filtro EMA, utiliza la base de media de pesos ponderados y un factor α comprendido entre 0 y 1 que se encarga de suavizar la señal o conservar su fidelidad. En este caso, luego de varias pruebas, se eligió un factor α de 0.1 dentro de la expresión mostrada en la ecuación 3.7.

$$EMA_n = \alpha * lectura_ADC + (1 - \alpha) * EMA_{n-1} \quad \text{Ec. (3.7)}$$

Se observa que el filtro se aplica sobre los datos directos de la lectura del ADC, teniendo como resultado lo mostrado en la figura 3.35. En la figura se observan, en azul los datos provenientes de la lectura del ADC y en rojo el resultado de la aplicación del filtro EMA pasa bajo.

Figura 3.4: Lectura ADC vs. tiempo (Sin filtro y aplicado filtro EMA pasa bajo)



Gráfica datos obtenidos desde ADC y filtrados adquiridos por Serial Plotter de Energia IDE en función del tiempo, Gustavo Reyes

Para la conversión de los datos obtenidos, de crudo (raw) a datos de distancia en [cm] se utilizó la ecuación 3.5, obteniendo así la referencia para poder ser ingresado al controlador PID previamente mencionado. Además, la medición de la altura del perdigón se realizó siguiendo el mismo proceso indicado en el apartado 3.4.1.

Para el control PID, se utilizó una versión discretizada (digital) de este, que es apta para ser implementada en microcontroladores. Esta versión de controlador PID reemplaza la integral y la derivada de la versión tradicional (véase apartado 2.6) con una aproximación adecuada para el modo de procesamiento de un microcontrolador. Esta versión aproximada, o discretizada, se expresa con la ecuación 3.8, en donde *out* es la salida del controlador (valor del pulso de asignación de velocidad del motor sin escobillas), *error* es el error actual,

$sumError$ es el error acumulado perteneciente a la parte integral, $rateError$ es el error de la parte derivativa, y K_p , K_i y K_d corresponden a las constantes de ajuste del controlador.

$$out = K_p * error + K_i * sumError + K_d * rateError \quad \text{Ec. (3.8)}$$

Para conocer el valor de $error$ es necesario definirlo mediante la resta entre la referencia (set point) y el valor medido, en este caso de la altura actual del perdigón. El error acumulado $sumError$ se define como la suma del error acumulado con el error actual y multiplicado por la mitad del tiempo de muestreo T_s . Por último, el error de la parte derivativa $rateError$ se define como la resta entre error actual y el error anterior dividido para T_s . Cada uno de estos errores se caracterizan por las ecuaciones 3.9, 3.10 y 3.11, respectivamente.

$$error = setpoint - height \quad \text{Ec. (3.9)}$$

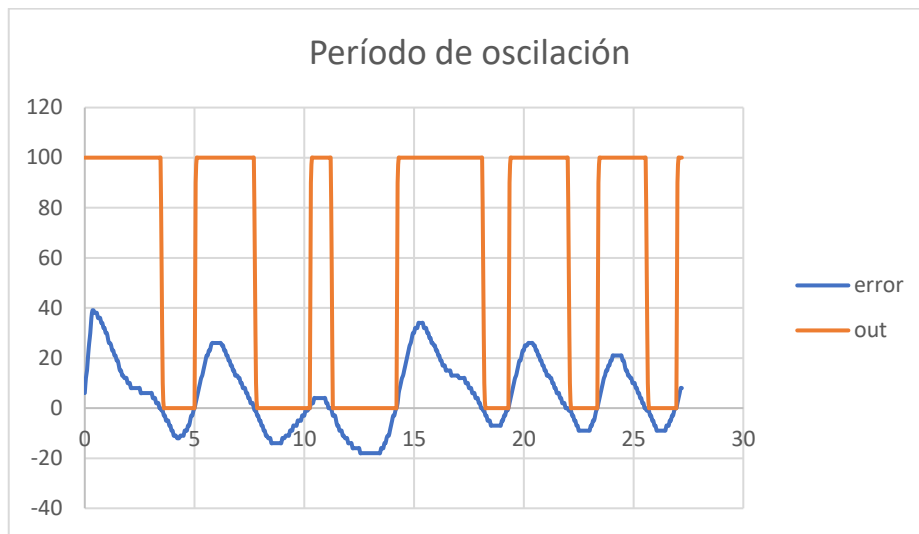
$$sumError = (sumError + error) * T_s / 2 \quad \text{Ec. (3.10)}$$

$$rateError = \frac{(error - lastError)}{T_s} \quad \text{Ec. (3.11)}$$

El último error, que en la ecuación 3.11 se denomina $lastError$, es el error obtenido en el proceso de control previo, y es uno de los parámetros que le da la característica de acumulativo al controlador.

Además, el tiempo de muestreo es un parámetro que, si bien está definido por una relación matemática, también depende del tiempo de oscilación, obtenido de manera experimental. Este tiempo de oscilación se obtiene al anular las constantes integral y derivativa del control e incrementar la constante proporcional K_p hasta que el sistema oscila de manera constante y sostenida (en este caso $K_p = 45$). Con una muestra de más de 500 datos obtenidos mediante el monitor serial de Energía IDE, se obtuvo la gráfica del período de oscilación del sistema, en el que se superponen tanto el error como la salida del sistema, considerando que la salida está en una escala del 0% al 100%. Esta gráfica se puede observar en la figura 3.36.

Figura 3.4: Gráfica período de oscilación



Gráfica de datos de error y salida obtenidos para determinar el período de oscilación del sistema,
Gustavo Reyes

Por la naturaleza del sistema, una oscilación de período constante es únicamente posible en sistema ideales. Por esta razón se tomó la medición del tiempo de cada uno de los ciclos medidos y se obtuvo un promedio resultante de 4.5[s]. Estos datos se pueden observar en la tabla 3.5.

Tabla 3.5: Período de oscilación promedio

Ciclo	Período [s]
1	5.1
2	5.3
3	4.0
4	5.2
5	4
6	3.6
Promedio	4.5

Medida de períodos de oscilación constantes, Gustavo Reyes

Para definir, a partir del período de oscilación, el valor del período de muestreo, se utiliza la relación mostrada en la ecuación 3.12. Esta relación indica que el período de muestreo debe ser 10 veces menos que el tiempo de oscilación del sistema.

$$T_s < \frac{4.5}{10} \quad \text{Ec. (3.12)}$$

De dicha operación, se obtiene que el período de muestreo será menor a 0.45 segundos, siendo ese el punto de partida para realizar ajustes pertinentes en caso de ser necesario. Con esto, se tienen los parámetros necesarios para desarrollar el proceso de la escritura del código para el controlador PID, incluyendo sus variables, e incluirlo en la totalidad del código fuente de la planta. Esto se puede observar en la gráfica 3.37.

Figura 3.4: Código PID discreto

```
double kp
double ki
double kd

unsigned long currentTime, previousTime;
double Ts, error, lstError, sumError, rateError;
double height, out, setpoint;

void pid() {
    error = setpoint-height;
    sumError = sumError+(error*(Ts/2));
    rateError = (error-lstError)/Ts;
    out = (kp*error)+(ki*sumError)+(kd*rateError);
    lstError=error;
}
```

Código fuente controlador PID discreto de la planta en Energia IDE, Gustavo Reyes

El valor de salida *out* es el que representa la velocidad para ser asignada al motor sin escobillas a través del ESC. Sin embargo, al tener una entrada mapeada para valores entre 0% y 100%, es recomendable que el valor de salida sea delimitado a esos mismos valores. Finalmente, este valor debe volver a ser mapeado entre 1500 y 2000, que son los valores en microsegundos del pulso de inicio y máximo, respectivamente. El código fuente para manejo del motor sin escobillas es el mismo que el indicado al finalizar el apartado 3.4.1.

Para sintonizar las constantes de funcionamiento del controlador PID, se consideran a nivel general los ajustes heurísticos o los basados en reglas. Para determinar el mejor método de ajuste se establecen las características de cada uno de ellos, y así se define el del mejor desempeño. El ajuste heurístico se descarta en primera instancia por tomar un tiempo

demasiado extenso en ser ajustado, además de no garantizar la robustez del funcionamiento del sistema. Dentro de los métodos basados en reglas, se consideraron los métodos de Ziegler Nichols, Lambda y Kappa Tau.

El método Kappa Tau se considera una evolución del método de Ziegler Nichols, mejorando la respuesta oscilatoria del sistema y su acción frente a perturbaciones; sin embargo, su incapacidad de determinar parámetros de desempeño, como la necesidad de un sobre pico, lo descarta para el uso en esta planta. El método de sintonización Lambda para control PID, como característica principal presenta su uso eficiente únicamente en controladores PI, además de ser indicado para sistemas de primer orden, por lo que se descarta. El método de sintonización de Ziegler Nichols es el de mejor adaptación para sistemas de segundo orden, así como para aplicaciones de altas características experimentales, por lo que se considera el apropiado para esta planta.

3.5.3 Mando por HMI en HTML

En el método de mando por HMI (Interfaz Humano Máquina) en HTML y comunicación Ethernet desde la tarjeta de desarrollo Tiva TM4C1294, se conserva el proceso de lectura de la altura del perdigón y del controlador PID con su respectiva asignación de velocidad al motor sin escobillas a través del ESC, variando únicamente su manera de adquirir el valor de referencia para el controlador. En este caso, la tarjeta Tiva TM4C1294 opera a modo de servidor, siendo la encargada de albergar el código fuente (en HTML-HyperText Markup Language) de la HMI, que integra un formulario y ejecuta la acción de envío de datos. Además de albergar la interfaz, la tarjeta Tiva, se encarga de leer el dato enviado desde la acción del formulario, para extraerlo y transformarlo a un número que pueda ser tratado como tal. El proceso de conexión a la HMI se realiza gracias a la librería “Ethernet” incluida en el software Energia IDE para la tarjeta Tiva TM4C1294, que incorpora funciones de parametrización de la comunicación, establecimiento de la conexión y lectura de datos. Para el establecimiento de esta comunicación, es necesario también establecer una comunicación serial a un *baud rate* de 115200 para ejecutar comprobaciones de datos mientras se utiliza la HMI. En la figura 3.38 se puede observar la inclusión de la librería Ethernet, los parámetros de los dos tipos de comunicación a establecerse y sus respectivos inicios.

Figura 3.4: Código comunicación Ethernet

```

#include "Ethernet.h"
byte mac[] = { 0xAA, 0xBB, 0xCC, 0xDD, 0xEE, 0xFF };
EthernetServer server(80);

void setup() {
  Serial.begin(115200);
  Serial.println("Connecting to Ethernet...");
  IPAddress ip = IPAddress(192,168,0,102);
  IPAddress gw = IPAddress(192,168,0,1);
  IPAddress mask = IPAddress(255,255,255,0);
  Ethernet.begin(mac, ip);
  server.begin();
}

EthernetClient client;

```

Código establecimiento comunicación Ethernet y serial, y parámetros de comunicación en Energia IDE, Gustavo Reyes

Para la lectura del dato enviado desde la interfaz se necesita la integración de 3 procesos secuenciales, que son: verificación de conexión de cliente, lectura e interpretación de dato obtenido de formulario de la interfaz, y ejecución del código HTML que muestra la HMI. El código desarrollado para estos procesos se puede observar en la figura 3.39.

Figura 3.4: Obtención de datos desde cliente web en HTML

```

void loop() {
  client = server.available();
  if (client) {
    Serial.print("new cliente on port");
    Serial.println(client.port());
    String currentLine = "";
    while (client.connected()) {
      if (client.available()) {
        char c = client.read();
        Serial.print(c);
        if (currentLine.length() < 100) {
          currentLine += c;
        }
        if (c == '\n') {
          if (currentLine.indexOf("temp") > 0) {
            int pos = currentLine.indexOf("=");
            int n = ((currentLine[pos+1] - 48) * 10) + currentLine[pos+2] - 48;
          }
          client.println("HTTP/1.1 200 OK");
          client.println("Content-Type: text/html");
          client.println();
          client.println("");
          break;
        }
      }
    }
    delay(1);
    currentLine="";
    client.stop();
  }
}

```

Código lectura de datos desde cliente web por comunicación Ethernet en Energia IDE, Gustavo Reyes

El cliente web, como se ha denotado, está desarrollado mediante código HTML para que responda a la conexión Ethernet de la Tarjeta Tiva. Para esto se incluyeron títulos en diferentes niveles con datos concernientes a la temática de la interfaz y el proyecto que comanda, así como una caja de texto y un botón, que en conjunto envían el dato que sirve como referencia del sistema de control. El código desarrollado para cumplir con lo expresado se puede observar en la figura 3.40 (a), y su visualización en la figura 3.40 (b).

Figura 3.4: Cliente web en HTML para envío de datos a tarjeta Tiva TM4C1294

```

1 <html>
2 <head>
3 <title>HMI TM4C1294 Tiva C Series</title>
4 </head>
5
6 <body><center>
7 <h1 style = font-size:60px>
8 <FONT COLOR=navy face=righteous>UNIVERSIDAD POLITÉCNICA SALESIANA</FONT>
9 </h1>
10
11 <h2 style = font-size:50px>
12 <FONT COLOR=black face=righteous>CARRERA DE INGENIERÍA ELECTRÓNICA</FONT>
13 </h2>
14
15 <h2 style = font-size:35px>
16 <FONT COLOR=teal face=righteous>CONTROL DE POSICIÓN DE UN PERDIGÓN</FONT>
17 </h2>
18
19 <h2 style = font-size:35px>
20 <FONT COLOR=teal face=righteous>DENTRO DE UN TÚNEL DE VIENTO VERTICAL</FONT>
21 </h2>
22
23 <h2 style = font-size:30px><FONT COLOR=red face=righteous>TIVA TM4C1294 C SERIES</FONT></h2>
24 <h2 style = font-size:29px><FONT COLOR=black face=righteous>Altura deseada</FONT></h2>
25 <h2 style = font-size:29px><FONT COLOR=navy face=righteous>(Referencia o Set Point en [cm])</FONT></h2>
26
27 <form action='/' method='get'>
28 <input type='number' name='height' style = font-size:28px>
29 <input type='submit' style = font-size:28px>
30 </form><br><br><br>
31
32 <h3 style = font-size:15px><FONT COLOR=black face=righteous>Gustavo Reyes Navarrete</FONT></h3>
33 </center></body>
34 </html>

```

(a)

UNIVERSIDAD POLITECNICA SALESIANA

CARRERA DE INGENIERIA ELECTRONICA

CONTROL DE POSICION DE UN PERDIGON

DENTRO DE UN TUNEL DE VIENTO VERTICAL

TIVA TM4C1294 C SERIES

Altura deseada

(Referencia o Set Point en [cm])

SetPoint: 25

Gustavo Reyes Navarrete

Si se va a Ingresar un solo digito, recuerde Ingresar con un cero Inicial

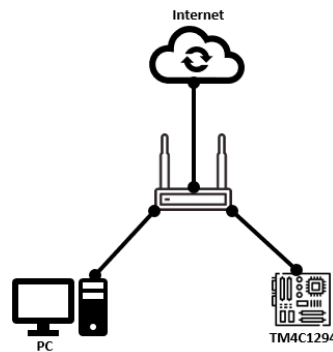
(b)

(a) Código en HTML de cliente web para lectura y envío de dato de referencia de controlador de altura de perdigón dentro de un túnel de viento vertical. (b) Visualización de HMI en HTML para lectura de referencia de controlador de altura de perdigón dentro de un túnel de viento vertical,

Gustavo Reyes

Para el montaje físico de la red se necesitaron tres elementos: un router, una PC y la tarjeta Tiva TM4C1294. El router es el dispositivo encargado de direccionar los paquetes que intervengan en la comunicación. La tarjeta Tiva funciona como servidor, albergando el código de la HMI con sus funcionalidades, además de controlar la planta. Por último el PC toma la característica de cliente, siendo desde aquí, a través de un navegador web, que se accede a la HMI alojada en el servidor, para así poder enviar el dato de referencia para funcionamiento de la planta. Esta red se conecta a Internet, que si bien no es indispensable para su funcionamiento como tal, es necesario para mantener la conexión a Internet de la PC. La topología de conexión de lo indicado se ve en la figura 3.41.

Figura 3.4: Topología de red comunicación Ethernet TM4C1294



Topología de red para comunicación Ethernet de servidor web en tarjeta Tiva TM4C1294 hacia cliente HTML, Gustavo Reyes

Se realizó una configuración básica en el router y PC, y mediante código en la tarjeta Tiva entorno a las características de red de cada uno de estos dispositivos. La información compilada respecto a lo mencionado se puede observar en la tabla 3.6.

Tabla 3.6: Direcciones dispositivos de la red

Característica	Dirección
IP puerta de enlace predeterminada	192.168.0.1
Dirección IPv4 de red	192.168.0.100
Máscara de subred	255.255.255.0
Dirección servidor DNS	8.8.8.8
Dirección IPv4 PC	192.168.0.101
Dirección IPv4 TM4C1294	192.168.0.102
Dirección MAC TM4C1294	70:FF:76:1D:6B:DE

Direcciones de dispositivos de red para comunicación Ethernet Tiva TM4C1294, Gustavo Reyes

Con lo indicado en este capítulo, se puso en marcha la planta, para ser sujeto de pruebas y ajustes, de ser necesario, para garantizar el correcto funcionamiento del sistema con cada uno de sus métodos de mando y características particulares. El resultado de este proceso se explica en el Capítulo 4.

CAPÍTULO 4

PRUEBAS Y AJUSTES

En el proceso de comprobación del funcionamiento de la planta se realizaron pruebas del sistema, ajustando los parámetros de ejecución, conservando sus características esenciales. A lo largo de la ejecución de este procedimiento, se realizaron diversos ajustes en los algoritmos de funcionamiento, más allá del hardware, que únicamente fue sometido a mediciones para comprobación. A continuación, se detalla el proceso de pruebas y ajuste para cada método de mando propuesto inicialmente.

4.1 Mando por posiciones definidas (Comunicación Serial)

En el proceso de mando por posiciones definidas, el método de comunicación entre el usuario y la planta es la comunicación serial nativa de la tarjeta Tiva TM4C1294. Por este medio, y gracias al Monitor Serial incluido en el software Energia IDE, se establece la altura deseada para el posicionamiento del perdigón.

Como primer ajuste, se tomó en cuenta la altura del perdigón para poder ser considerada dentro de las medidas del sensor de distancia que determina su posición, así como para limitar los valores de referencia a los cuales se puede acceder. Recordando que, el sensor ultrasónico de distancia HC-SR04 entrega medidas confiables entre los 4[cm] y 80[cm], y considerando su elevación fuera del recorrido posible del tubo, se llegó a los parámetros que se observan en la tabla 4.1.

Tabla 4.5: Parámetros limitantes de altura

Parámetro	Valor
Altura del perdigón	8 [cm]
Longitud completa del tubo	55 [cm]
Rango de medición sensor	4 [cm] – 51[cm]
Referencia mínima (altura)	8 [cm]
Referencia máxima (altura)	47 [cm]

Parámetros limitantes para la medición de posición del perdigón y establecimiento de referencias,

Gustavo Reyes

Con esto definido, y para garantizar la seguridad y el funcionamiento en caso de que el usuario ingrese un valor que exceda los parámetros, también se establecieron estos límites dentro del código fuente de la planta. Esto permite que los valores de referencia, o altura de operación deseada, se mantengan dentro del límite operativo.

Para los límites de operación del motor que impulsa la EDF, con base en la Tabla 3.2 y pruebas hechas al motor de manera independiente, se establecieron como valor de apagado (OFF) a un pulso de 1665[ms] y como valor de máxima velocidad (ON) a un pulso de 1695[ms]. Con estos valores se realizó la prueba operativa del sistema para una referencia de 25[cm], con un umbral de 2[cm].

Como resultado se observó una oscilación constante, similar a lo observado en la figura 3.35, por lo que se descartó el uso del control por histéresis y se procedió a la aplicación del controlador PID previsto los dos mandos de control restantes. El criterio de descarte del control por histéresis es la imposibilidad de un usuario de permanecer en oscilación constante dentro de un túnel de viento, lo que, por sí mismo, ocasionaría afectaciones a su integridad física.

Según lo indicado en el apartado 3.4.2, se realizó el ajuste de los parámetros del controlador PID, tomando como punto de partida la constante proporcional de oscilación K_c con un valor de 45. Para un controlador PID completo, uno de los métodos de ajuste más fiable es el de Ziegler Nichols, que establece una proporción definida en función del período de oscilación T_c , y la ya mencionada K_c (Ogata, 1998). Esta proporción se define con expresiones matemáticas sencillas, mismas que se recopilan a continuación, en la tabla 4.2.

Tabla 4.2: Fórmulas de ajuste constantes controlador PID

Constante del Controlador PID	Valor en función de K_c y T_c
K_p	$0.59 * K_c$
K_i	$1.18 * K_c * T_c$
K_d	$0.075 * K_c * T_c$

Fórmulas de ajuste de constantes de controlador PID según el método de Ziegler-Nichols, Gustavo

Reyes

Para el sistema desarrollado, en específico, se obtuvieron los resultados de estos parámetros, tal como se muestran, a continuación, en la tabla 4.3.

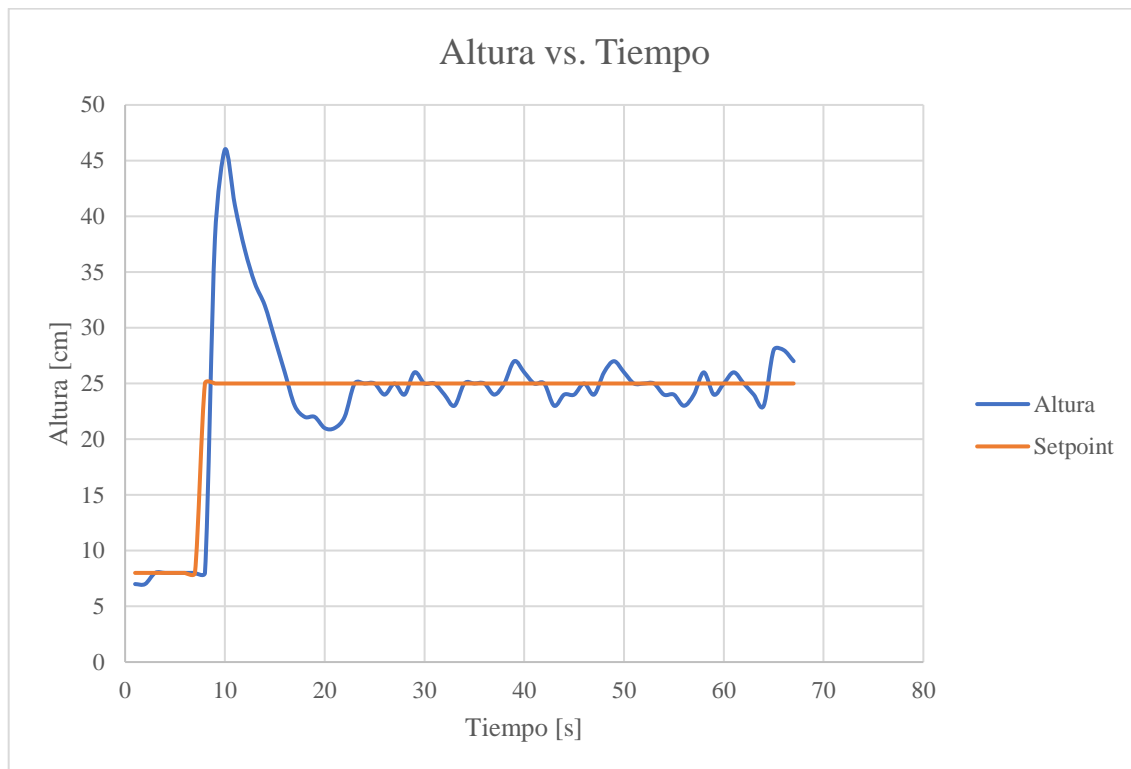
Tabla 4.3: Parámetros y constantes PID ajustados por Ziegler-Nichols

Parámetro	Resultado (redondeado a entero)	Tipo de resultado
K_c	45	Experimental
T_c	5	Experimental
K_p	27	Calculado
K_i	11	Calculado
K_d	17	Calculado

Parámetros de controlador PID regidores y resultantes en función del ajuste por el método de Ziegler-Nichols, Gustavo Reyes

Adicional, se realizó un ajuste de proporción a las tres constantes, multiplicando cada una por 1/10, para así lograr una equiparación de la base de medición de tiempo del período crítico con la base del tiempo de ejecución del algoritmo. Por último, fue necesario un ajuste en la salida del controlador PID, limitándola entre los valores de 0 a 100 para pasar, luego, a un mapeo entre 1650[ms] y 1700[ms] de pulso de control, con lo que se aprecia una ejecución suave en el cambio de velocidad del motor. Esto, para finalmente ejecutar la primera prueba de funcionamiento, con una recopilación de datos de altura y referencia (o SetPoint) de la planta, obteniendo una tendencia de comportamiento representada en la gráfica de Altura vs. Tiempo que se observa en la figura 4.1.

Figura 4.5: Gráfica primera prueba de funcionamiento de planta con controlador PID



Comportamiento de planta con controlador PID ($K_p=2.7$, $K_i=1.1$, $K_d=1.7$) con referencia fija, Gustavo Reyes

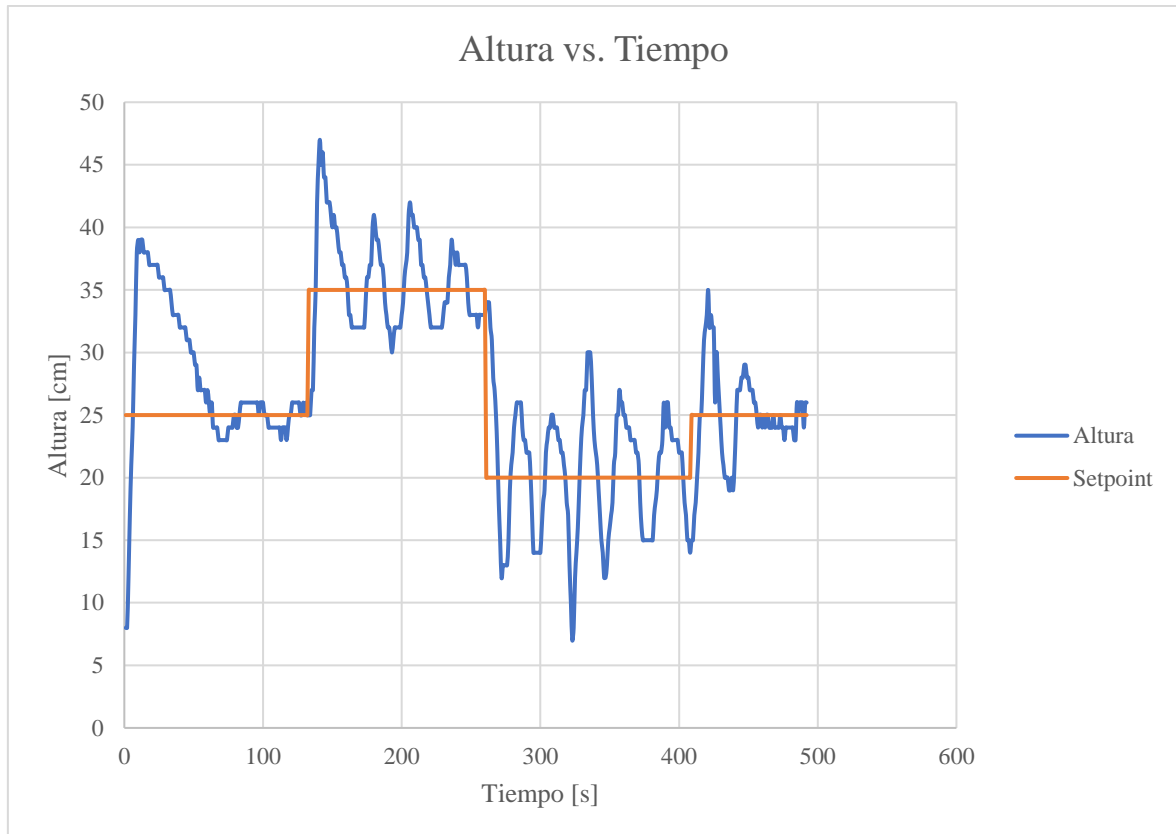
De la primera prueba, observando la figura 4.1, para una referencia o Setpoint de 25[cm], se tiene un sistema con un sobre pico alto, es decir sub amortiguado. Este fenómeno es deseable, ya que es necesario romper una fuerte inercia inicial para la elevación de una masa en reposo (coloquialmente llamada “peso muerto”), que es lo que supone la elevación de un sujeto dentro de un túnel de viento.

Respecto al error, por la resolución de la altura (1[cm]), tanto en medida como en establecimiento de referencia, es mucho más aceptable entregar el rango de error en unidades y no en porcentajes, como se acostumbra regularmente. En este caso, en estado estable, el error se presenta en una banda de +/-3[cm]. Por la naturaleza de funcionamiento de la planta, así como las características del perdigón, este se considera como un error aceptable.

A pesar de esto, y con el fin adicional de realizar más pruebas y establecer una comparativa del funcionamiento del sistema, se realizaron 4 pruebas adicionales con pequeños ajustes cada una de ellas.

Para la segunda prueba, el enfoque fue el cambio de referencia a lo largo del tiempo de funcionamiento de la planta, manteniendo los parámetros del controlador PID. El comportamiento que responde a esto se puede observar en la gráfica de Altura vs. Tiempo que se observa en la figura 4.2.

Figura 4.5: Gráfica segunda prueba de funcionamiento de planta con controlador PID



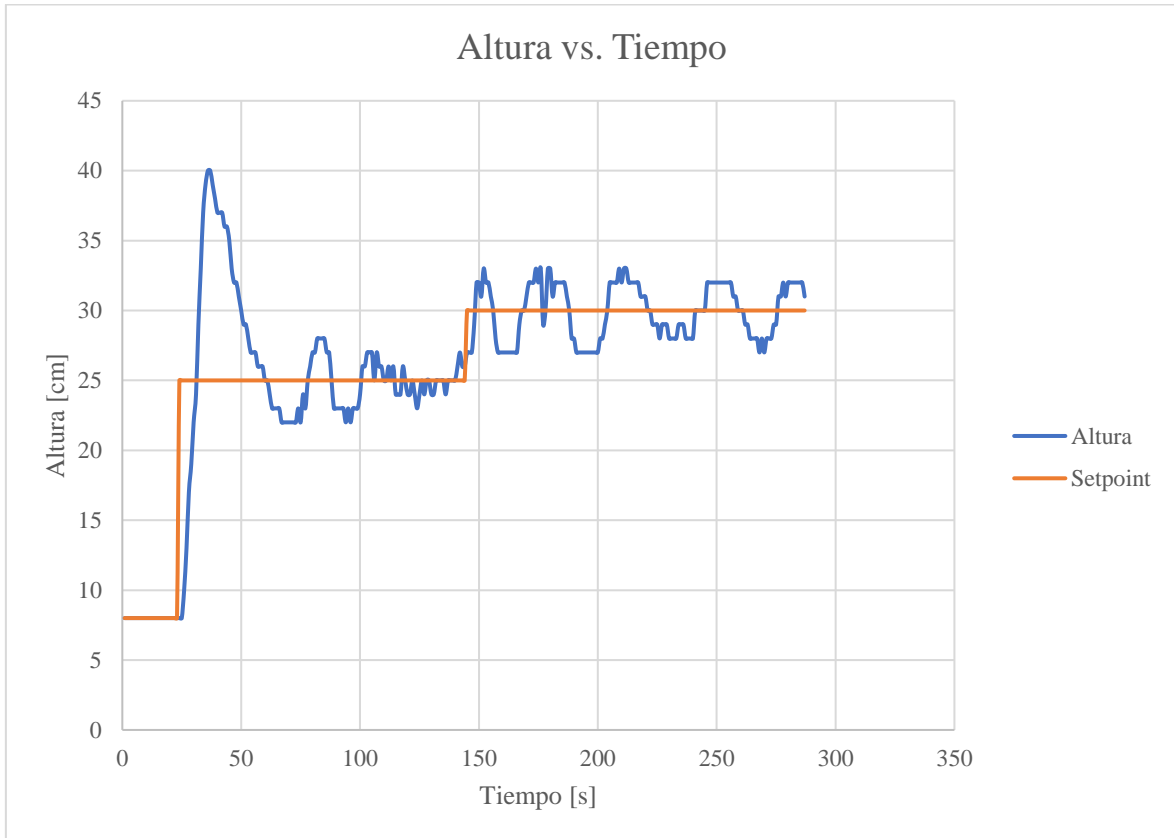
Comportamiento de planta con controlador PID ($K_p=2.7$, $K_i=1.1$, $K_d=1.7$) con referencia variable,
Gustavo Reyes

En la segunda prueba, con lo representado en la figura 4.2, se puede observar que, además de un claro sobre pico en cada cambio de referencia, existe un comportamiento oscilatorio incongruente con la necesidad de funcionamiento de la planta, llegando a una banda de error de hasta ± 10 [cm]. Por esto, se descartan estos parámetros de funcionamiento, forzando a una reducción de la constante proporcional (K_p) del sistema para la tercera prueba.

En la tercera prueba, se mantuvo la directriz de cambiar la referencia del controlador PID durante su funcionamiento, además de un cambio de K_p de 2.7 a 2.2, para así ocasionar una

reducción de la oscilación, brindándole estabilidad al sistema, aunque sacrificando el error en estado estable. El comportamiento de esta prueba se puede observar a continuación, en la gráfica de Altura vs. Tiempo que se observa en la figura 4.3.

Figura 4.5: Gráfica tercera prueba de funcionamiento de planta con controlador PID

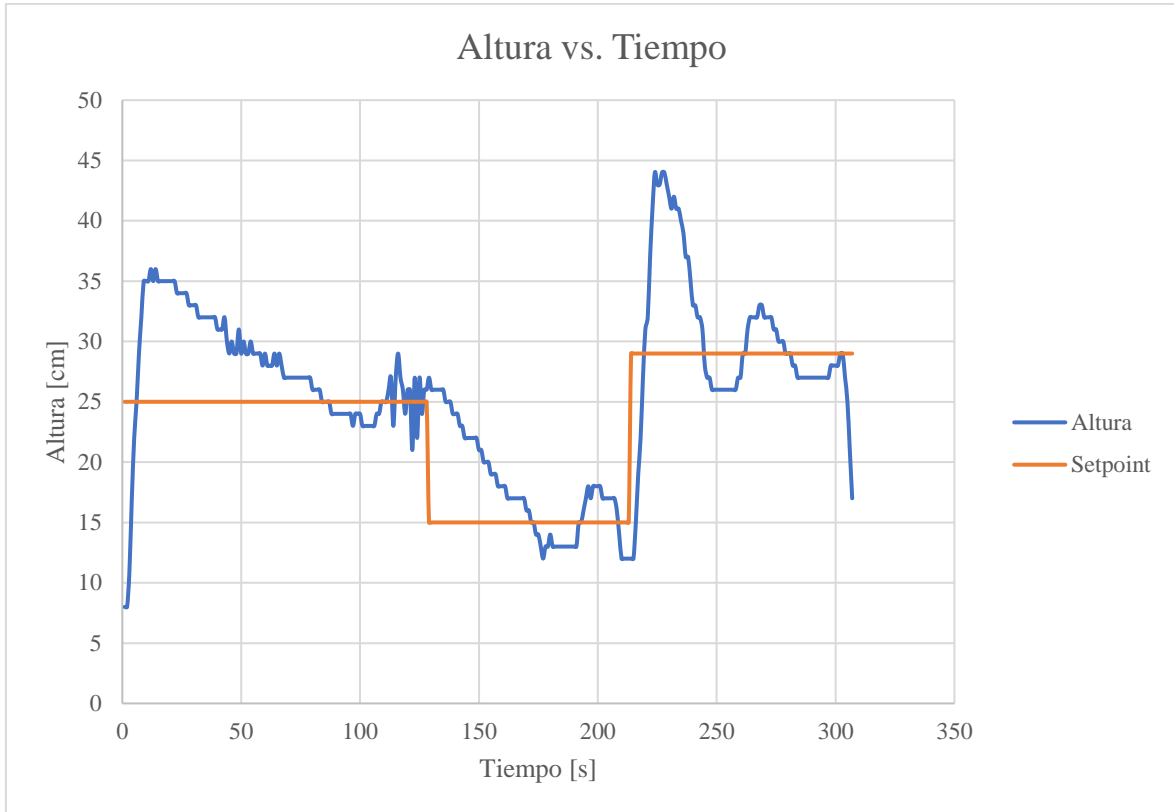


Comportamiento de planta con controlador PID ($K_p=2.2$, $K_i=1.1$, $K_d=1.7$) con referencia variable, Gustavo Reyes

En esta prueba, se mantiene el sobre pico inicial, en el primer establecimiento de la referencia, pero este desaparece al ser cambiado este valor nuevamente, lo que implica una mejora en el comportamiento del sistema. Para este comportamiento, se tiene una banda de error en estado estable de ± 3 [cm], lo que es normal por la reducción en el valor de K_p .

Para mitigar el efecto de la reducción del valor de K_p , en la cuarta prueba, se incrementó el valor de K_i de 1.1 a 1.5, para disminuir el error del sistema en estado estable. Conservando el cambio de referencia a lo largo del funcionamiento del sistema, se obtuvo la gráfica de Altura vs. Tiempo que caracteriza su comportamiento, misma que se puede observar en la figura 4.4.

Figura 4.5: Gráfica cuarta prueba de funcionamiento de planta con controlador PID

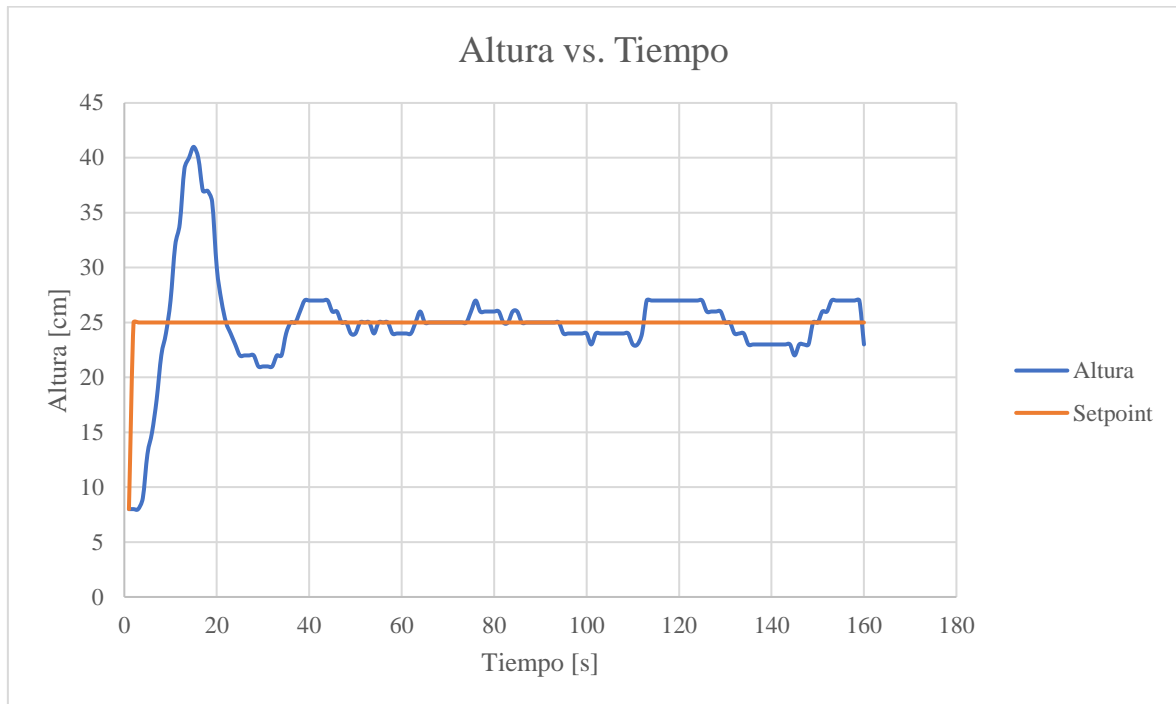


Comportamiento de planta con controlador PID ($K_p=2.2$, $K_i=1.5$, $K_d=1.7$) con referencia variable,
Gustavo Reyes

De esta prueba, se observa un comportamiento altamente oscilatorio, al que le es imposible llegar al estado estable o régimen permanente, llegando a bandas de error de hasta ± 17 [cm]. Además, se vuelve a apreciar un claro sobre pico al cambiar de valor de referencia del controlador PID. Por esto, se descartan estos parámetros, y se establece como el mejor comportamiento al observado en la tercera prueba. Para corroborar, se realiza una última prueba volviendo a las constantes previas, pero manteniendo un set point fijo.

De la cuarta y última prueba para el establecimiento de la referencia a través de comunicación serial, se pudo observar su comportamiento en la gráfica de Altura vs. Tiempo mostrada en la figura 4.5.

Figura 4.5: Gráfica quinta prueba de funcionamiento de planta con controlador PID



Comportamiento de planta con controlador PID ($K_p=2.2$, $K_i=1.1$, $K_d=1.7$) con referencia fija, Gustavo Reyes

En este último caso, el error en estado estable vuelve a la banda de ± 2 [cm], además de la conservación del sobre impulso inicial, ya definido como necesario. Con esto, además, se corroboran los mejores parámetros de funcionamiento del sistema, mismos que se establecen como los definitivos, incluso en los métodos de mando faltantes de comprobar. Estos parámetros se resumen, a continuación, en la tabla 4.4.

Tabla 4.4: Constantes controlador PID definitivos

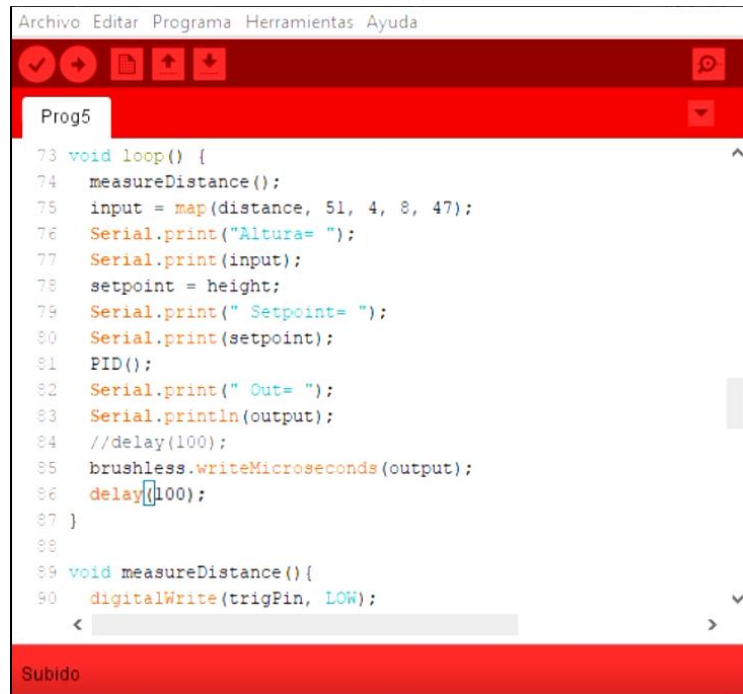
Parámetro	Abreviatura	Valor
Constante proporcional	K_p	2.2
Constante integral	K_i	1.1
Constante derivativa	K_d	1.7

Constantes definitivas posteriores a ajuste de controlador PID para mejor funcionamiento del sistema, Gustavo Reyes

Con este conjunto de parámetros se realizaron las pruebas de funcionamiento, en la que el set point es ingresado al sistema por comunicación serial a través del teclado de la PC que alberga el IDE Energia. En este caso se tecldea un número que sirva como referencia (en

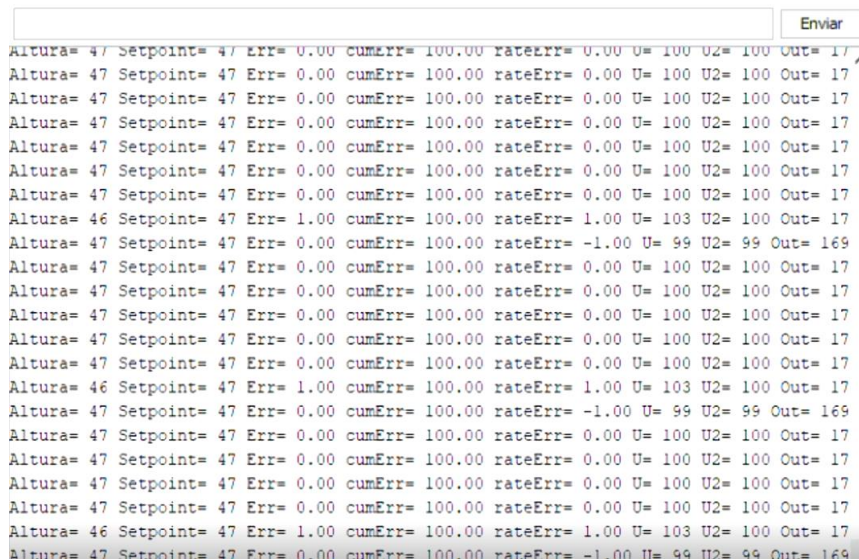
centímetros), el cual está auto limitado a los límites de funcionamiento de la planta, generando los resultados que se pueden ver en la figura 4.6.

Figura 4.5: Prueba de funcionamiento de planta con primer método de mando (comunicación serial)



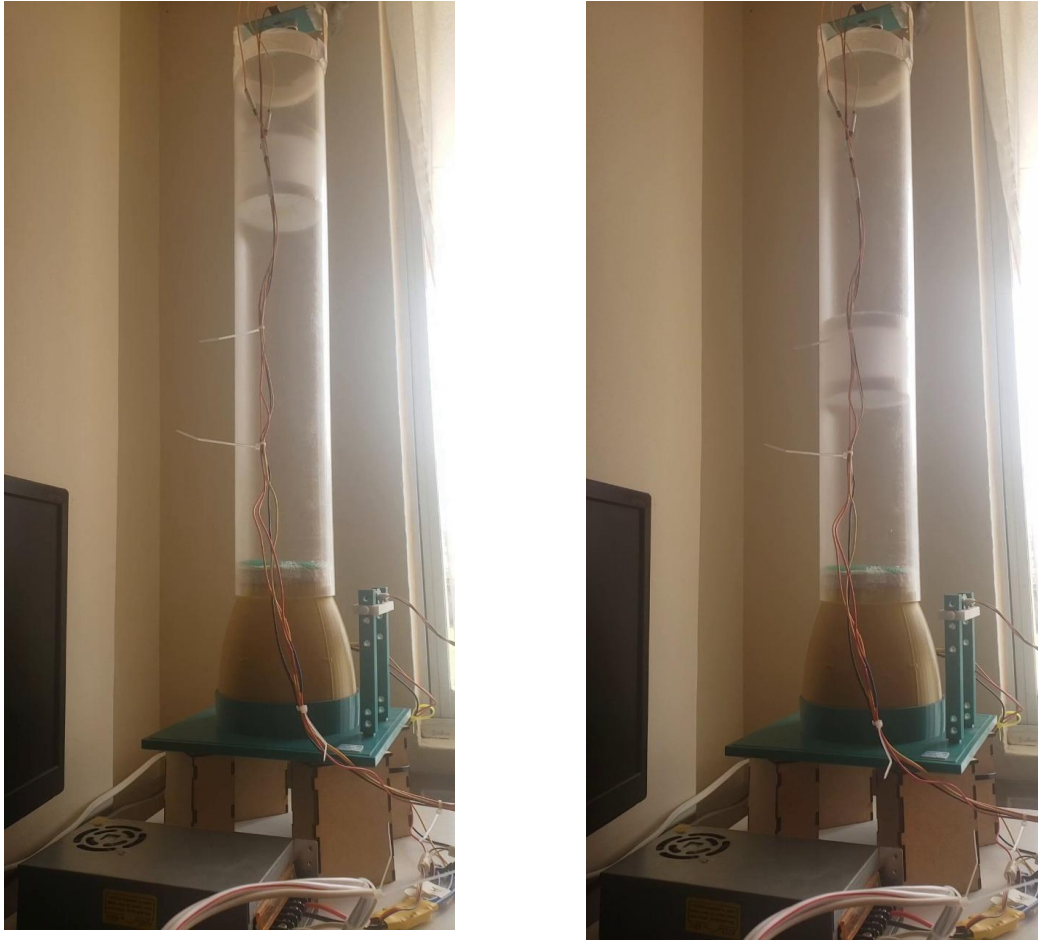
```
Archivo Editar Programa Herramientas Ayuda
Prog5
73 void loop() {
74   measureDistance();
75   input = map(distance, 51, 4, 8, 47);
76   Serial.print("Altura= ");
77   Serial.print(input);
78   setpoint = height;
79   Serial.print(" Setpoint= ");
80   Serial.print(setpoint);
81   PID();
82   Serial.print(" Out= ");
83   Serial.println(output);
84   //delay(100);
85   brushless.writeMicroseconds(output);
86   delay(100);
87 }
88
89 void measureDistance(){
90   digitalWrite(trigPin, LOW);
Subido
```

(a)



```
Enviar
Altura= 47 Setpoint= 47 Err= 0.00 cumErr= 100.00 rateErr= 0.00 U= 100 U2= 100 Out= 17
Altura= 47 Setpoint= 47 Err= 0.00 cumErr= 100.00 rateErr= 0.00 U= 100 U2= 100 Out= 17
Altura= 47 Setpoint= 47 Err= 0.00 cumErr= 100.00 rateErr= 0.00 U= 100 U2= 100 Out= 17
Altura= 47 Setpoint= 47 Err= 0.00 cumErr= 100.00 rateErr= 0.00 U= 100 U2= 100 Out= 17
Altura= 47 Setpoint= 47 Err= 0.00 cumErr= 100.00 rateErr= 0.00 U= 100 U2= 100 Out= 17
Altura= 46 Setpoint= 47 Err= 1.00 cumErr= 100.00 rateErr= 1.00 U= 103 U2= 100 Out= 17
Altura= 47 Setpoint= 47 Err= 0.00 cumErr= 100.00 rateErr= -1.00 U= 99 U2= 99 Out= 169
Altura= 47 Setpoint= 47 Err= 0.00 cumErr= 100.00 rateErr= 0.00 U= 100 U2= 100 Out= 17
Altura= 47 Setpoint= 47 Err= 0.00 cumErr= 100.00 rateErr= 0.00 U= 100 U2= 100 Out= 17
Altura= 47 Setpoint= 47 Err= 0.00 cumErr= 100.00 rateErr= 0.00 U= 100 U2= 100 Out= 17
Altura= 47 Setpoint= 47 Err= 0.00 cumErr= 100.00 rateErr= 0.00 U= 100 U2= 100 Out= 17
Altura= 47 Setpoint= 47 Err= 0.00 cumErr= 100.00 rateErr= 0.00 U= 100 U2= 100 Out= 17
Altura= 46 Setpoint= 47 Err= 1.00 cumErr= 100.00 rateErr= 1.00 U= 103 U2= 100 Out= 17
Altura= 47 Setpoint= 47 Err= 0.00 cumErr= 100.00 rateErr= -1.00 U= 99 U2= 99 Out= 169
Altura= 47 Setpoint= 47 Err= 0.00 cumErr= 100.00 rateErr= 0.00 U= 100 U2= 100 Out= 17
Altura= 47 Setpoint= 47 Err= 0.00 cumErr= 100.00 rateErr= 0.00 U= 100 U2= 100 Out= 17
Altura= 47 Setpoint= 47 Err= 0.00 cumErr= 100.00 rateErr= 0.00 U= 100 U2= 100 Out= 17
Altura= 46 Setpoint= 47 Err= 1.00 cumErr= 100.00 rateErr= 1.00 U= 103 U2= 100 Out= 17
Altura= 47 Setpoint= 47 Err= 0.00 cumErr= 100.00 rateErr= -1.00 U= 99 U2= 99 Out= 169
```

(b)



(c)

Comportamiento de planta con controlador PID con referencia por comunicación serial (a) visualización desde Energia IDE, (b) visualización datos de control y referencia de algoritmo, (c) vista de planta en funcionamiento, Gustavo Reyes

En la figura 4.6 (a) se observa el código ejecutado para el funcionamiento del control PID de la planta a través de la referencia entregada por comunicación serial, teniendo en cuenta que de manera previa, se realizó todo el proceso de sintonización de este algoritmo. En la figura 4.6 (b), se observan los datos del algoritmo de control en funcionamiento en tiempo real, entre los que se observan la altura medida, la referencia, los errores y la salidas no tratada y tratada del sistema.

Es preciso indicar que, para considerar el mejor funcionamiento del sistema, no se tomó en cuenta el parámetro del tiempo o velocidad de estabilización del sistema, por no ser un parámetro imprescindible en el escenario de aplicación previsto de esta planta. Al emular un sistema para que una persona ocupe el lugar del perdigón, la velocidad de estabilización pasa

a segundo plano, e incluso le da un valor agregado el hecho de que el sistema no haga correcciones bruscas que puedan afectar al usuario.

También, se añade el factor de la definición de la referencia por teclado con un comportamiento de pseudo interrupción, dotando así al sistema de una oportuna lectura del parámetro y su inclusión en el algoritmo. Este comportamiento se logra al establecer la lectura por el puerto serial como un evento, que si bien no es una interrupción como tal, sí tiene una característica de prioridad en cada ciclo de ejecución del programa, otorgando así una rápida respuesta del algoritmo para establecer cada vez su nueva referencia.

Por último, el controlador PID implementado cuenta con dos características imprescindibles de precisar. La primera es la ausencia del parámetro del tiempo de muestreo, ya que al estar procesándose en un microcontrolador con ciclos de máquina y ciclos de reloj definidos, cada una de las repeticiones del bucle se ejecuta en, aproximadamente, el mismo tiempo. La segunda característica es la limitación del error acumulado del controlador, o “Integral Windup” para permitir el cambio de las referencias sin convertir el PID en un controlador inestable.

4.2 Mando por sensor externo

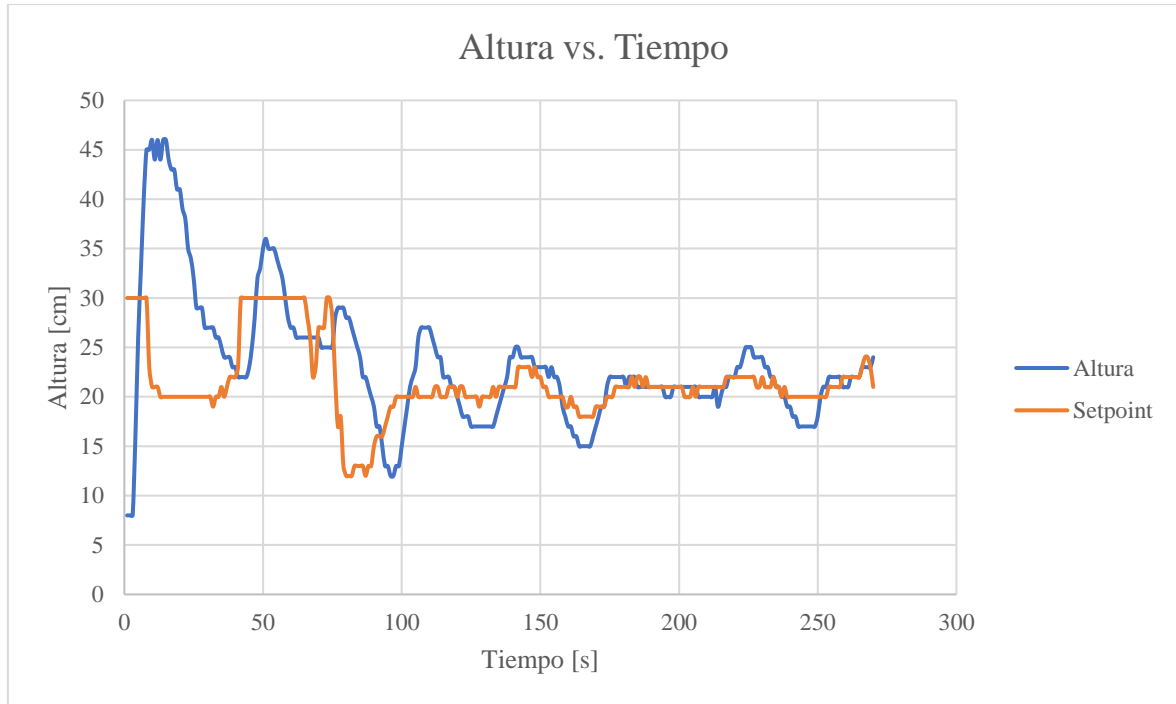
El hecho de haber comprobado los parámetros de funcionamiento del controlador PID provoca que no exista la necesidad de volver a sintonizar o ajustar el controlador. Por esto, el proceso de comprobación se reduce considerablemente.

En cuanto a los parámetros de funcionamiento del controlador PID, las constantes se mantuvieron las mostradas en la tabla 4.4. De igual manera, se mantuvo la limitación de la medición de la posición del perdigón, del error acumulado y de la salida del controlador. El parámetro que fue manipulado fue la limitación de la lectura del sensor externo que establece la referencia del sistema, que responde al rango de lectura de este. Al tener un sensor infrarrojo SHARP GP2Y0A41SK0F que entrega datos fiables entre los 4[cm] y 30[cm], es necesario establecer sus límites, que en este caso van de 8[cm] a 30[cm], porque se toma como mínimo la altura del perdigón.

Con esto, se realizaron dos pruebas para comprobar el correcto desempeño de este método de mando del túnel de viento, variando, únicamente, la cantidad de muestras entre la una y

la otra. El comportamiento de la primera prueba se puede observar en la gráfica de Altura vs. Tiempo mostrada, a continuación, en la figura 4.7.

Figura 4.5: Primera prueba de funcionamiento controlador PID comandado por sensor externo

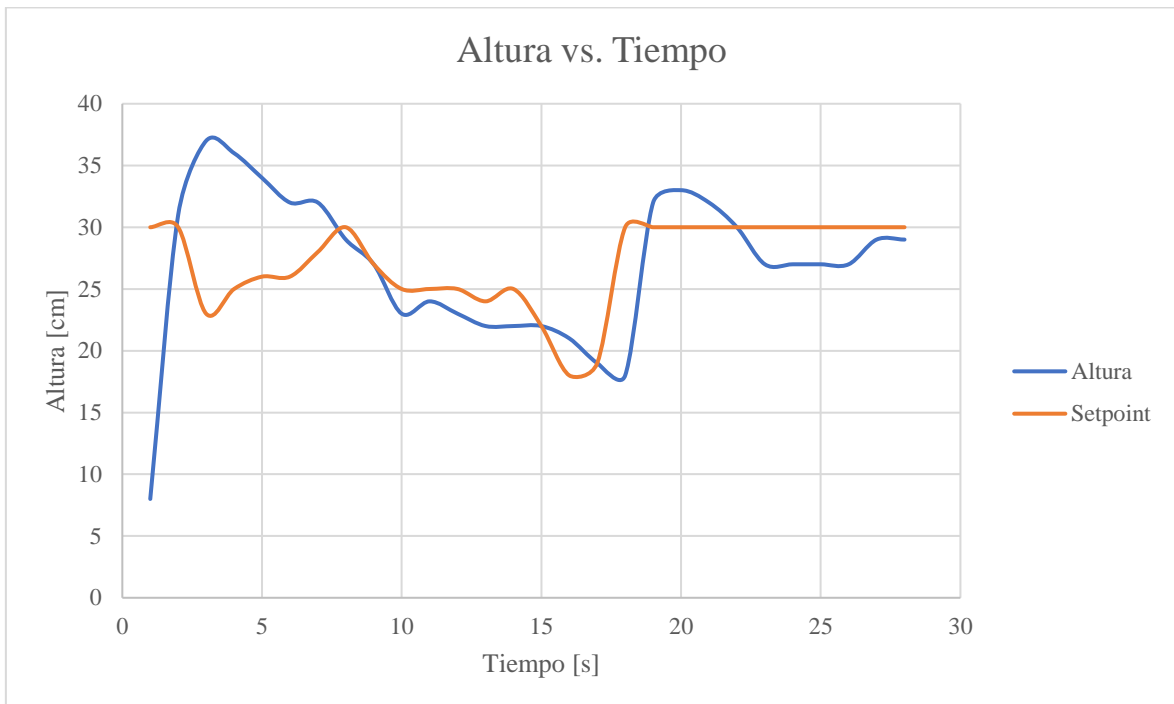


Comportamiento de planta con controlador PID ($K_p=2.2$, $K_i=1.1$, $K_d=1.7$) con referencia comandada por sensor externo, Gustavo Reyes

De la figura 4.6, se puede observar que la referencia del controlador PID presenta un comportamiento variante, e incluso cuando está estable, presenta ligeras perturbaciones que afectan al controlador. Al momento de llegar a la región de estabilidad, el controlador presenta una banda de error en régimen permanente de ± 3 [cm], conservando el pico de sobre impulso al momento de los cambios bruscos de referencia.

Se realizó una segunda prueba con una reducción significativa en el número de muestras tomadas, únicamente para comprobar la tendencia de funcionamiento de la planta. Este comportamiento se puede observar en la gráfica de Altura vs. Tiempo mostrada, a continuación, en la figura 4.8.

Figura 4.5: Segunda prueba de funcionamiento controlador PID comandado por sensor externo



Comportamiento de planta con controlador PID ($K_p=2.2$, $K_i=1.1$, $K_d=1.7$) con referencia comandada por sensor externo con muestras reducidas, Gustavo Reyes

Lo observado en la figura 4.8, únicamente corrobora el comportamiento correctivo del controlador PID y la variación constante de los valores de la referencia del sistema. Al igual que en pruebas anteriores, la banda de error en estado estable es de +/-3[cm].

A nivel operativo, el sistema provocó la igualación de altura entre mano y perdigón, siempre que la primera se encuentre dentro de los límites permitidos por el sistema. Esta operación se puede observar en la figura 4.9.

Figura 4.5: Prueba de funcionamiento de planta con segundo método de mando (sensor externo)

```
ADCProg | Energia 1.6.10E18
Archivo Editar Programa Herramientas Ayuda
ADCProg
88 Serial.print("Altura= ");
89 Serial.print(input);
90 measureSetpoint();
91 setpoint = distance_IR;
92 Serial.print(" Setpoint= ");
93 Serial.print(setpoint);
94 PID();
95 Serial.print(" Out= ");
96 Serial.println(output);
97 //delay(100);
98 brushless.writeMicroseconds(output);
99 delay(1000);
100 }
101
102 void measureSetpoint() {
103   long suma = 0;
104   for(int u=0; u<20; u++){
105     suma = suma + (analogRead(A1));
106     delay(5);
107   }
108   promedio = floor((suma/n));
109
110   distance_IR=27708*pow(promedio,-1.09);
111   if(distance_IR >= 30){
112     distance_IR = 30;
113   }else if(distance_IR <= 8){
114     distance_IR = 8;
115   }
116 }
117
118 void measureDistance() {
119   digitalWrite(trigPin, LOW);
120   delayMicroseconds(2);
121   digitalWrite(trigPin, HIGH);
122   delayMicroseconds(10);
123   digitalWrite(trigPin, LOW);
```

(a)



(b)

Comportamiento de planta con controlador PID con referencia por mando con sensor externo (a) visualización desde Energia IDE, (b) vista de planta en funcionamiento, Gustavo Reyes

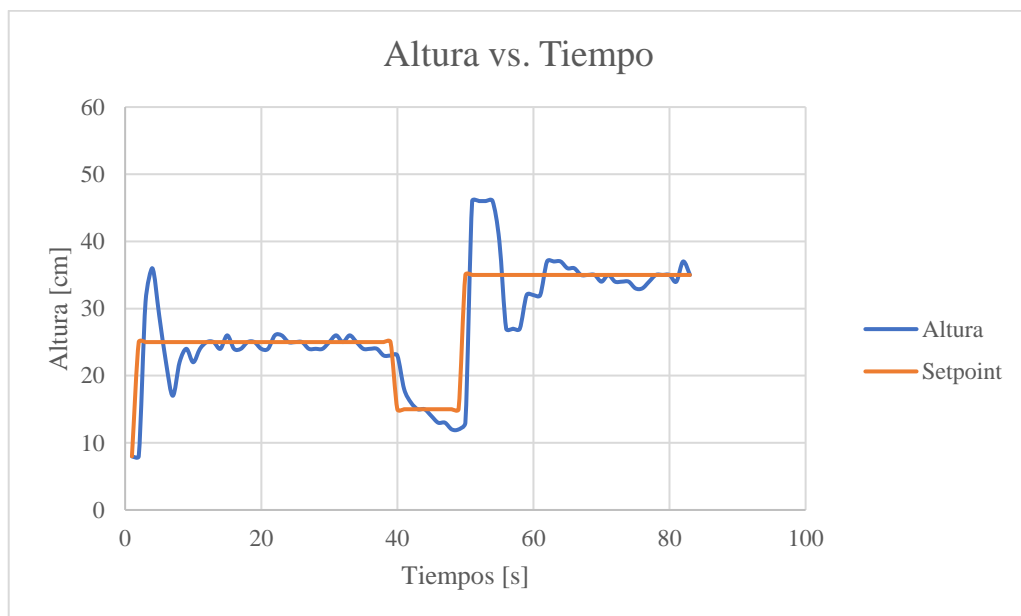
En la figura 4.9 (a) se observa el registro del código implementado para el control de altura del perdigón mediante un sensor externo que permite equiparar la altura a la que se ubique el perdigón con la altura a la que se coloque la mano. En la figura 4.9 (b) se observa esto de manera real, con la consideración, que si la ubicación de la mano excede los límites permitidos por el sensor y el sistema, este lo limitará de manera automática, ya sea al mínimo o máximo valor soportado.

4.3 Mando por HMI en HTML

En el último método de mando, se realizó una única prueba, que compruebe el funcionamiento del control PID. El cambio de referencia del sistema, en este caso, se establece a través de la comunicación Ethernet, lo que otorga una mayor estabilidad al sistema. Esto ocurre porque el tiempo de refresco necesario para el vaciado del buffer que permita la lectura del dato desde el formulario proporciona una mayor estabilidad del sistema.

En cuanto a las constantes del controlador PID, estas se mantienen en los valores indicados en la tabla 4.4. El comportamiento del sistema en esta prueba se puede observar en la gráfica de Altura vs. Tiempo mostrada en la figura 4.10.

Figura 4.5: Prueba de funcionamiento controlador PID comandado por HTML vía Ethernet



Comportamiento de planta con controlador PID ($K_p=2.2$, $K_i=1.1$, $K_d=1.7$) con referencia enviada por comunicación Ethernet, Gustavo Reyes

Del comportamiento de la planta mostrado en la figura 4.10, se puede apreciar que el sobre impulso se mantiene, y al llegar al estado estable, el error se concentra en una banda de +/- 3[cm], sin embargo estos son valores con poca persistencia, siendo, en mayor relevancia, el error de +/-1[cm]. Cabe recalcar que para este método de mando no se realizó ninguna otra modificación adicional a las expuestas en los apartados 3.4.3, 4.1 y 4.2.

A nivel real se realizó el acceso al hospedador de la interfaz de control a través de la dirección IP configurada en un navegador web de PC. En esta interfaz, se solicita al usuario ingresar mediante teclado la referencia del sistema en centímetros y con dos dígitos. Además, el sistema está establecido para que muestre esta referencia a susceptibilidad de cambio, además de auto limitar el valor en caso de exceder los límites. Esta prueba se puede observar a continuación, en la figura 4.11.

Figura 4.5: Prueba de funcionamiento de planta con tercer método de mando (HMI en HTML)

```

Eth2 | Energia 1.6.10E18
Archivo Editar Programa Herramientas Ayuda
Eth2
103 void loop() {
104     client = server.available();
105     if(client){
106         //Serial.print("New client on port ");
107         //Serial.println(client.port());
108         String currentLine = "";
109         while(client.connected()){
110             if(client.available()){
111                 char c = client.read();
112                 if(currentLine.length() < 100){
113                     currentLine = currentLine + c;
114                 }
115                 if(c == '\n'){
116                     if(currentLine.indexOf("height") > 0 ){
117                         int pos = currentLine.indexOf("=");
118                         m = ((currentLine[pos+1] - 48) *10) + currentLine[pos+2]-48;
119                         Serial.println(m);
120                         if(m >= 47){
121                             m = 47;
122                         }else if(m <= 8){
123                             m = 8;
124                         }
125                         ma = m;
126                     }
127                     html_code();
128                     break;
129                 }
130             }
131         }
132         delay(1);
133         currentLine = "";
134         client.stop();
135     }
136 }
137 measureDistance();

```

(a)

UNIVERSIDAD POLITECNICA SALESIANA

CARRERA DE INGENIERIA ELECTRONICA

CONTROL DE POSICION DE UN PERDIGON DENTRO DE UN TUNEL DE VIENTO VERTICAL

TIVA TM4C1294 C SERIES

Altura deseada

(Referencia o Set Point en [cm])

25

SetPoint: 25

Gustavo Reyes Navarrete

Si se va a Ingresar un solo digito, recuerde Ingresar con un cero Inicial

(b)



(c)

Comportamiento de planta con controlador PID con referencia desde HMI en HTML vía Ethernet

(a) visualización desde Energia IDE, (b) visualización de HMI, (c) vista de planta en funcionamiento, Gustavo Reyes

4.4 Resumen de algoritmos

Para resumir, el algoritmo central de control utilizado es un controlador PID (proporcional, integral, derivativo); sin embargo, se incorporaron algoritmos de soporte dentro del funcionamiento de la planta para mejorar su operatividad. A continuación, en la tabla 4.5, se muestra un resumen.

Tabla 4.5: Resumen de algoritmos para optimización de métodos de mando

Tipo de mando		
1. Comunicación Serial		
Algoritmo implementado	Aplicación	Detalle
Algoritmo de control de levitación	PID	Algoritmo para control de posición vertical
Ajuste al algoritmo de control	Integral Windup	Limitación del error excedente por el cambio de set point
Algoritmo de lectura de referencia	Cuasi interrupción por evento	Lectura de dato por evento interrumpido en el programa
2. Sensor externo		
Algoritmo implementado	Aplicación	Detalle
Algoritmo de control de levitación	PID	Algoritmo para control de posición vertical
Ajuste al algoritmo de control	Integral Windup	Limitación del error excesivo por el cambio de set point
Algoritmo de filtrado	Filtro EMA (media móvil exponencial)	Filtrado de señales con el uso de media móvil
3. HMI por Ethernet		
Algoritmo implementado	Aplicación	Detalle
Algoritmo de control de levitación	PID	Algoritmo para control de posición vertical
Ajuste al algoritmo de control	Integral Windup	Limitación del error excesivo por el cambio de set point
Algoritmo de lectura de referencia	Cuasi interrupción por evento	Lectura de dato por evento interrumpido en el programa

Algoritmos implementados para optimización de métodos de mando, Gustavo Reyes

CONCLUSIONES

Con lo expuesto en los capítulos anteriores, se pudo observar el desarrollo integral de un túnel de levitación neumática. Este dispositivo utilizó como controlador, únicamente, la tarjeta de desarrollo TIVA TM4C1294, residiendo todo su procesamiento en esta terminal de desarrollo, cuyo algoritmo fue generado, compilado y subido a través del entorno de desarrollo Energia. Los algoritmos aplicados fueron desarrollados para la implementación de mandos de métodos de mando y control que optimicen su funcionamiento, considerando su posible alcance a gran escala y distribución comercial.

Se estableció, mediante investigación, que, en la implementación de los túneles de viento comerciales, se utilizan variaciones empíricas de los flujos de viento entrantes en el túnel, careciendo así de métodos de mando y control que garanticen ciertos parámetros de funcionamiento. Por esto se definieron métodos de mando adecuados para el operador, además de un sistema de control que permita estabilidad en el funcionamiento del túnel de viento. Estos mandos son: control por comunicación serial para permitir un funcionamiento que no presente cambios bruscos de la referencia de altura de vuelo, control por sensor externo para que el mando sea interactivo, y control por comunicación vía Ethernet que abre la posibilidad a mandos distribuidos, múltiples o a distancia.

El prototipo del túnel de viento se desarrolló mediante la integración de técnicas de manufactura aditiva (modelado por deposición fundida de ácido poliláctico) y sustractiva, incluyendo ciertos elementos artesanales, que, en conjunto, permitieron el desplazamiento lineal vertical de un perdigón. Este modo de desarrollo permitió la generación de un modelo semi modular, que facilita el intercambio de partes; además, el movimiento vertical del perdigón es fluido gracias a una elevación del modelo respecto a su plano de asentamiento, creando una cámara de aire semi abierta bajo la EDF.

Para el control de posición del perdigón dentro del túnel de viento se implementó un control proporcional integral derivativo (PID) que permita variar su referencia a través de definición específica por comunicación serial y la lectura de un sensor externo para interacción humana. En este controlador PID se integró la técnica de “Integral Windup” para permitir una variación de referencia sin afectar en gran medida su estabilidad.

El desarrollo de un control mediante una HMI facilitó la acción del operador del sistema, además de agregarle estabilidad a su control por sus características intrínsecas. Esto se logró gracias a la integración de un recurso desarrollado en HTML, albergado dentro de la tarjeta TIVA TM4C1294 operando como servidor, con comunicación vía Ethernet, utilizando el protocolo HTTP para su enlace. Este tipo de control se establece para facilitar a una mejor interacción entre la planta y el operario, pudiendo ésta ser a distancia, de manera distribuida desde varias estaciones, supervisada por un panel maestro, o hasta integrada con tecnologías inmersivas como paneles virtuales (digital, realidad virtual o realidad aumentada).

En las pruebas desarrolladas se evidenciaron las ventajas y desventajas de cada uno de los métodos de mando implementados, así como sus parámetros para el mejor funcionamiento posible. Entre las ventajas se evidenció una mayor fluidez para el control por sensor externo, una mayor estabilidad para el control vía Ethernet y un menor error en estado estable para el control por comunicación serial. Como desventajas se evidenciaron, oscilaciones pronunciadas al cambio de referencia en la comunicación serial y picos de lectura en el control por el sensor externo que representan inestabilidad en la lectura de la referencia. Para los constantes del controlador PID, el mejor desempeño del sistema se evidenció al tener constantes establecidas para los valores de $K_p=2.2$, $K_i=1.1$ y $K_d=1.7$, presentando una banda de error en estado estable de ± 2 [cm].

RECOMENDACIONES

El control de velocidad de un motor sin escobillas se realiza a través de un controlador electrónico de velocidad basados en control senoidal o trapezoidal, lo que ocasiona que existe un pulso mínimo necesario para el arranque. Este arranque del motor se establece en una velocidad considerada, ya, como elevada. Por esta razón, se recomienda como mejora o aplicación futura, el reemplazo del ESC por un controlador FOC (Field Oriented Control), que basa su funcionamiento en el control por campos vectoriales. Este tipo de controlador permitirá el arranque del motor desde bajas velocidades, elevará su eficiencia y permitirá un control más preciso.

Una de las dificultades más grandes dentro de los métodos de mando implementados fue la estabilización de la lectura del sensor externo en el método comandado por este. Esto se da por la naturaleza del sensor, que presenta gran cantidad de ruido, con alto grado de dificultad para ser mitigado sea por hardware o software. Por esta razón, se recomienda el cambio de este elemento a un sensor tipo laser ToF (Time of Flight), que además de entregar una medición más estable, presenta mejores características en cuanto a reflectancia en diferentes superficies y colores de estas.

El sensor HC-SR04 es ultrasónico, por lo que está en su naturaleza verse afectado por la temperatura del aire y la presión atmosférica a la que esté sometido, aunque sea en mínimas cantidades. Considerando esto, se recomienda que, para grandes aplicaciones de este sistema, la medición de la altura del usuario se realice con otro tipo de tecnología, pudiendo ser, incluso, visión artificial.

REFERENCIAS BIBLIOGRÁFICAS

- ABRA. (2022). *Motors*. Obtenido de ABRA Electronics Corp.: <https://abra-electronics.com/quadcopters/motors/mot-qc-02.html>
- Braslavsky, J. (2002). Control Automático. *Congreso de Ingeniería en Automatización y Control Industrial*. Quilmes: Universidad Nacional de Quilmes.
- Centre for Aerodynamics and Environmental Flow. (s.f.). *Electric Ducted Fan Propulsion*. Obtenido de University of Surrey: <https://www.surrey.ac.uk/centre-aerodynamics-environmental-flow/research/electric-ducted-fan-propulsion>
- Diosdado, R. (26 de Noviembre de 2014). *Sensor de ultrasonidos HC-SR04*. Obtenido de Zona Maker: <https://www.zonamaker.com/arduino/modulos-sensores-y-shields/ultrasonido-hc-sr04>
- Energia. (s.f.). *Guide to the TM4C129 Connected LaunchPad*. Obtenido de Energia: <https://energia.nu/pinmaps/ek-tm4c1294xl/>
- Energia. (s.f.). *Home: Energia*. Obtenido de Energia: <https://energia.nu>
- Farnell Components SL. (2013). *Sensores Fotoeléctricos de Distancia*. Obtenido de Farnell : <https://es.farnell.com/sharp/gp2y0a41sk0f/sensor-distancia-analogue-output/dp/1618431>
- Gong, A., MacNeill, R., & Verstraete, D. (2018). Performance Testing and Modeling of a Brushless DC Motor, Electronic Speed Controller and Propeller for a Small UAV Application. *2018 Joint Propulsion Conference*, (pág. 4584). Cincinnati.
- Krakow, K., Johnson, S., & Guy, J. (Octubre de 2013). System Design Guidelines for the TM4C129x Family of Tiva C Series Microcontrollers. *Application Report*. Dallas, Texas, Estados Unidos de América: Texas Instruments.
- Martínez, K. (Octubre de 2018). Conceptos Básicos: Sistemas de Control. *Revista InTech México Automatización*.
- Mazzone, V. (2002). Controladores PID. *Control Automático I Automatización y Control Industrial*. Universidad Nacional de Quilmes, Quilmes, Argentina.

- Muñoz Mejías, A. (Octubre de 2012). Diseño y Análisis Computacional para Túnel de Viento de Baja Velocidad. *Proyecto Fin de Carrera*. Universidad Carlos III de Madrid Escuela Politécnica Superior, Madrid, España.
- Muñoz V., J., & Espinosa N., E. (2010). Introducción a los Sistemas de Control. *Departamento de ingeniería Eléctrica*. Concepción, Chile: Universidad de Concepción.
- Nidec Corporation. (s.f.). *Brushless Motors*. Obtenido de Nidec Corporation: <https://www.nidec.com/en/technology/capability/brushless/>
- Ogata, K. (1998). *Ingeniería de Control Moderna*. Minnesota: Pearson.
- Sgueglia, A., Schmollgruber, P., Bartoli, N., & Atinault, O. (2018). Exploration and Sizing of a Large Passenger Aircraft with Distributed Ducted Electric Fans. *2018 AIAA Aerospace Sciences Meeting*. Toulouse.
- SparkFun. (2014). *Level Shifters*. Obtenido de SparkFun: <https://www.sparkfun.com/products/12009>
- Talos Electronics. (2022). *ESC 30A Multi Axis para Brushless*. Obtenido de Talos Electronics: <https://www.taloselectronics.com/products/controlador-de-velocidad-electronico-esc-30a-multi-axis-para-brushless>
- TecnoEdu. (s.f.). Obtenido de TecnoEdu: <https://tecnoedu.com/Armfield/C1510A.php>
- Texas Instruments Incorporated. (Octubre de 2016). EK-TM4C1294XL User's Guide. *Tiva™ C Series TM4C1294 Connected LaunchPad Evaluation Kit*.
- Unit Electronics. (2016). *Fuentes de Poder*. Obtenido de Unit Electronics: <https://uelectronics.com/producto/fuente-conmutada-12v-30a/>
- Viento Aventura. (04 de Marzo de 2016). *Tunel de Viento*. Obtenido de Viento Aventura: <http://www.vientoaventura.com/tunel-de-viento/que-es-un-tunel-de-viento/>
- Vuela. (2018). *El Túnel*. Obtenido de Vuela Túnel de Viento: <https://www.vuela.com.ar/nosotros.php>
- Yedamale, P. (2003). Brushless DC (BLDC) Motor Fundamentals. AN885. Microchip Technology Inc.

ZIKODRIVE. (2018). *Teoría de motores sin escobillas: cómo funcionan los motores sin escobillas (BLDC) y qué factores influyen en su rendimiento*. Obtenido de Zikodrive: <https://www.zikodrive.com/es/apoyo/zikouniversity-motor-control-teoria-aplicacion/teoria-del-motor-sin-escobillas-como-funcionan-los-motores-blde-y-que-factores-afectan-su-rendimiento/>

ANEXOS

ANEXO 1: Diseño estructura túnel de levitación neumática



ANEXO 2: Diseño integral aproximado de túnel de levitación neumática



**ANEXO 3: Construcción túnel de levitación neumática controlador por Tarjeta TIVA
TM4C1294**



ANEXO 4: Túnel de levitación neumática con comunicación serial en funcionamiento



ANEXO 5: Prueba de funcionamiento de túnel de levitación neumática con comunicación serial



ANEXO 6: Prueba de funcionamiento de túnel de levitación neumática con referencia establecida mediante sensor externo de interacción humana



ANEXO 7: HMI elaborado en HTML como método de mando de túnel de levitación neumática

UNIVERSIDAD POLITECNICA SALESIANA

CARRERA DE INGENIERIA ELECTRONICA

CONTROL DE POSICION DE UN PERDIGON

DENTRO DE UN TUNEL DE VIENTO VERTICAL

TIVA TM4C1294 C SERIES

Altura deseada

(Referencia o Set Point en [cm])

SetPoint: 25

Gustavo Reyes Navarrete

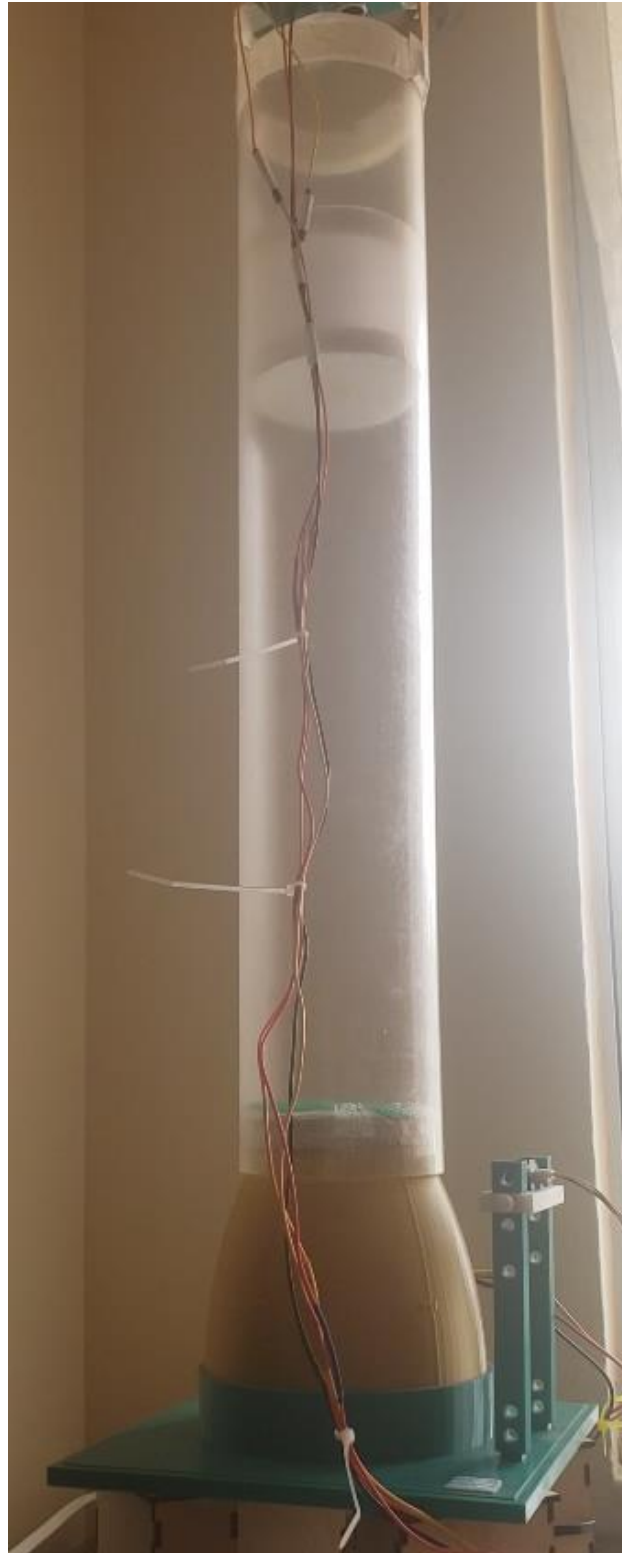
Si se va a ingresar un solo dígito, recuerde ingresar con un cero inicial

ANEXO 8: Prueba de conectividad entre PC de acceso (cliente) y tarjeta TIVA TM4C1294 (servidor) para conexión de HMI vía Ethernet

```
Haciendo ping a 192.168.0.102 con 32 bytes de datos:
Respuesta desde 192.168.0.102: bytes=32 tiempo<1m TTL=255
Respuesta desde 192.168.0.102: bytes=32 tiempo<1m TTL=255
Respuesta desde 192.168.0.102: bytes=32 tiempo<1m TTL=255
Respuesta desde 192.168.0.102: bytes=32 tiempo<1m TTL=255

Estadísticas de ping para 192.168.0.102:
    Paquetes: enviados = 4, recibidos = 4, perdidos = 0
    (0% perdidos),
    Tiempos aproximados de ida y vuelta en milisegundos:
    Mínimo = 0ms, Máximo = 0ms, Media = 0ms
```

ANEXO 9: Prueba de funcionamiento de túnel de levitación neumática comandado desde HMI vía Ethernet



ANEXO 10: Código fuente de algoritmo de control de altura de perdigón en túnel de viento con determinación de referencia a través de comunicación serial

```
//Inclusión de librerías y objetos de librerías
#include <Servo.h>
#define pin_esc 19
Servo brushless;

//Definición de variables de pines y
operación
String str_height = "8";
int height = 8;
int input;
int output = 1000;
int U;
double lastError;
const int echoPin = 39;
const int trigPin = 38;
long distance = 0;
long duration = 0;
int setpoint;
double kp = 22/10;
double ki = 11/10;
double kd = 17/10;
double error = 0;
double prevError = 0;
double cumError = 0;
double rateError = 0;

void setup() {
  brushless.attach(pin_esc);
  //Inicialización motor sin escobillas
  brushless.writeMicroseconds(1000);
  delay(5000);
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  Serial.begin(9600);
  Serial.setTimeout(10);
  Serial.println("Inicializando...");
  brushless.writeMicroseconds(1000);
  delay(1000);
  brushless.writeMicroseconds(1500);
  delay(1000);
  Serial.println("Funcionamiento");
}

void loop() {
  //Lectura y mapeo de altura de perdigón
  measureDistance();
  input = map(distance, 51, 4, 8, 47);
  //Establecimiento límites de altura
  if(height >= 47){
    height = 47;
  }else if(height <= 8){
    height = 8;
  }
  //Desarrollo algoritmo de control
  setpoint = height;
  PID();
  brushless.writeMicroseconds(output);
  delay(100);
}

//Método de medición de altura
void measureDistance(){
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  duration = pulseIn(echoPin, HIGH);
  distance = 0.01716*duration;
}

//Desarrollo controlador PID
void PID(){
  error = setpoint - input;
  cumError = cumError + error;
  //Integral Windup
  if (cumError >= 100) {cumError = 100;}
  else if (cumError <= 0) {cumError = 0;}
  rateError = (error - lastError);
  U = (kp*error) + (ki*cumError) +
(kd*rateError);
  lastError = error;
  //Límite de salida
  if(U >= 100){ U = 100; }
  if(U <= 0 ){ U = 0; }
  output = map(U, 0, 100, 1650, 1700);
}

//Evento de lectura serial de referencia
void serialEvent() {
  while (Serial.available()) {
    str_height = Serial.readStringUntil('\n');
    height = str_height.toInt();
  }
}
```

ANEXO 11: Código fuente de algoritmo de control de altura de perdigón en túnel de viento con determinación de referencia a través de sensor externo

```

//Inclusión de librerías y objetos de librerías
#include <Servo.h>
#define pin_esc 19
Servo brushless;
//Definición de variables
int input;
int output = 1000;
int U;
double lastError;
const int echoPin = 39;
const int trigPin = 38;
long distance = 0;
long duration = 0;
int distance_IR = 8;
int dist_ant;
int promedio;
const int n = 20;
int setpoint;
double kp = 22/10;
double ki = 11/10;
double kd = 17/10;
double error = 0;
double prevError = 0;
double cumError = 0;
double rateError = 0;

void setup() {
  brushless.attach(pin_esc);
  //Inicialización motor sin escobillas
  brushless.writeMicroseconds(1000);
  delay(5000);
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  pinMode(PE_2, INPUT);
  Serial.begin(9600);
  Serial.setTimeout(10);
  Serial.println("Inicializando...");
  brushless.writeMicroseconds(1000);
  delay(1000);
  brushless.writeMicroseconds(1500);
  delay(1000);
  Serial.println("Funcionamiento");
}

void loop() {
  //Lectura y mapeo de altura de perdigón
  measureDistance();
  input = map(distance, 51, 4, 8, 47);
  //Lectura de referencia de controlador
  measureSetpoint();
  //Desarrollo algoritmo de control
  setpoint = distance_IR;
  PID();
  brushless.writeMicroseconds(output);
  delay(100);
}

//Método de medición de altura
void measureDistance(){
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  duration = pulseIn(echoPin, HIGH);
  distance = 0.01716*duration;
}

//Método de medición de referencia
void measureSetpoint(){
  long suma = 0;
  //Filtro promedio
  for(int u=0; u<20; u++){
    suma = suma + (analogRead(A1));
    delay(5);
  }
  promedio = floor((suma/n));
  distance_IR=27708*pow(promedio,-1.09);
  //Establecimiento límites de altura
  if(distance_IR >= 30){
    distance_IR = 30;
  }else if(distance_IR <= 8){
    distance_IR = 8;
  }
}

//Desarrollo controlador PID
void PID(){
  error = setpoint - input;
  cumError = cumError + error;
  //Integral Windup
  if (cumError >= 100) {cumError = 100;}
  else if (cumError <= 0) {cumError = 0;}
  rateError = (error - lastError);
  U = (kp*error) + (ki*cumError) +
(kd*rateError);
  lastError = error;
  //Límite de salida
  if(U >= 100){ U = 100; }
  if(U <= 0){ U = 0; }
  output = map(U, 0, 100, 1650, 1700);
}

```

ANEXO 12: Código fuente de algoritmo de control de altura de perdigón en túnel de viento con determinación de referencia a través HMI con conexión vía Ethernet

```

//Inclusión de librerías, objetos y variables
#include <Ethernet.h>
#include <Servo.h>
#define pin_esc 19
Servo brushless;
byte mac[] = {0x70, 0xFF, 0x76, 0x1D,
0x6B, 0xDE};
EthernetServer server (80);
int m = 8;
int input;
int output = 1000;
int U;
double lastError;
const int echoPin = 39;
const int trigPin = 38;
long distance = 0;
long duration = 0;
int setpoint;
double kp = 22/10;
double ki = 11/10;
double kd = 17/10;
double error = 0;
double prevError = 0;
double cumError = 0;
double rateError = 0;
EthernetClient client;

void setup() {
//Inicialización motor sin escobillas
brushless.attach(pin_esc);
brushless.writeMicroseconds(1000);
delay(5000);
pinMode(trigPin, OUTPUT);
pinMode(echoPin, INPUT);
Serial.begin(9600);
Serial.setTimeout(10);
Serial.println("Iniciando...");
brushless.writeMicroseconds(1000);
delay(1000);
brushless.writeMicroseconds(1500);
delay(1000);
//Inicialización comunicación Ethernet
IPAddress ip = IPAddress(192,168,0,102);
IPAddress gw = IPAddress(192,168,0,1);
IPAddress mask =
IPAddress(255,255,255,0);
Ethernet.begin(mac, ip);
server.begin();
delay(2500);
Serial.println("Funcionamiento");
}

}

void loop() {
//Adquisición de datos vía Ethernet
client = server.available();
if(client){
String currentLine = "";
while(client.connected()){
if(client.available()){
char c = client.read();
if(currentLine.length() < 100){
currentLine = currentLine + c;
}
if(c == '\n'){
if(currentLine.indexOf("height") > 0 ){
int pos = currentLine.indexOf("=");
m = ((currentLine[pos+1] - 48) *10) +
currentLine[pos+2]-48;
Serial.println(m);
if(m >= 47){m = 47;}else if(m <=
8){m = 8;}
}
html_code();
break;
}
}
}
delay(1);
currentLine = "";
client.stop();
}
measureDistance();
input = map(distance, 51, 4, 8, 47);
setpoint = m;
PID();
brushless.writeMicroseconds(output);
delay(1000);
}

//Método de medición de altura
void measureDistance(){
digitalWrite(trigPin, LOW);
delayMicroseconds(2);
digitalWrite(trigPin, HIGH);
delayMicroseconds(10);
digitalWrite(trigPin, LOW);
duration = pulseIn(echoPin, HIGH);
distance = 0.01716*duration;
}
}

```

```

//Desarrollo controlador PID
void PID(){
    error = setpoint - input;
    cumError = cumError + error;
    //Integral Windup
    if (cumError >= 100) {cumError = 100;}
    else if (cumError <= 0) {cumError = 0;}
    rateError = (error - lastError);
    U = (kp*error) + (ki*cumError) +
(kd*rateError);
    lastError = error;
    //Límite de salida
    if(U >= 100){ U = 100; }
    if(U <= 0 ){ U = 0; }
    output = map(U, 0, 100, 1650, 1700);
}

//Servidor en código HTML
void html_code(){
    client.println("HTTP/1.1 200 OK");
    client.println("Content-Type: text/html");
    client.println();
    client.println("<!DOCTYPE HTML>");
    client.println("<html><head><title>HMI
TM4C1294 Tiva C Series</title></head>");
    client.println("<body><center>");
    client.println("<h1 style = font-
size:60px><FONT COLOR=navy
face=righteous>UNIVERSIDAD
POLITECNICA
SALESIANA</FONT></h1>");
    client.println("<h2 style = font-
size:50px><FONT COLOR=black
face=righteous>CARRERA DE
INGENIERIA
ELECTRONICA</FONT></h2>");
    client.println("<h2 style = font-
size:35px><FONT COLOR=teal

```

```

face=righteous>CONTROL DE POSICION
DE UN PERDIGON</FONT></h2>");
    client.println("<h2 style = font-
size:35px><FONT COLOR=teal
face=righteous>DENTRO DE UN TUNEL
DE VIENTO VERTICAL</FONT></h2>");
    client.println("<h2 style = font-
size:30px><FONT COLOR=red
face=righteous>TIVA TM4C1294 C
SERIES</FONT></h2>");
    client.println("<h2 style = font-
size:29px><FONT COLOR=black
face=righteous>Altura
deseada</FONT></h2>");
    client.println("<h2 style = font-
size:29px><FONT COLOR=navy
face=righteous>(Referencia o Set Point en
[cm])</FONT></h2>");
    client.println("<form action='/'
method='get'>");
    client.println("<input type='number'
name='height' style = font-size:28px>");
    client.println("<input type='submit'
value='Establecer' style = font-size:28px>");
    client.println("</form><br><br>");
    client.println("<h3 style = font-
size:15px><FONT COLOR=black
face=righteous>Gustavo Reyes
Navarrete</FONT></h3>");
    client.println("</form><br><br>");
    client.println("<h3 style = font-
size:15px><FONT COLOR=black
face=righteous>Si se va a ingresar un solo
digito, recuerde ingresar con un cero inicial
</FONT></h3>");
    client.println("</center></body></html>");
}

```

ANEXO 13: Código fuente HMI desarrollado en HTML

```
<!DOCTYPE HTML>
<html>
  <head><title>HMI TM4C1294 Tiva C Series</title></head>
  <body><center>
    <h1 style = font-size:60px><FONT COLOR=navy face=righteous>UNIVERSIDAD
POLITECNICA SALESIANA</FONT></h1>
    <h2 style = font-size:50px><FONT COLOR=black face=righteous>CARRERA DE
INGENIERIA ELECTRONICA</FONT></h2>
    <h2 style = font-size:35px><FONT COLOR=teal face=righteous>CONTROL DE
POSICION DE UN PERDIGON</FONT></h2>
    <h2 style = font-size:35px><FONT COLOR=teal face=righteous>DENTRO DE UN
TUNEL DE VIENTO VERTICAL</FONT></h2>
    <h2 style = font-size:30px><FONT COLOR=red face=righteous>TIVA TM4C1294 C
SERIES</FONT></h2>
    <h2 style = font-size:29px><FONT COLOR=black face=righteous>Altura
deseada</FONT></h2>
    <h2 style = font-size:29px><FONT COLOR=navy face=righteous>(Referencia o Set
Point en [cm])</FONT></h2>

    <form action="/" method='get' id="numform" oninput="x.value=parseInt(height.value)">
    <input type='number' name='height' id="height" style = font-size:28px>
    <input type='submit' value='Establecer' style = font-size:28px>
    <br><br>
    <FONT COLOR=red face=righteous style = font-size:28px>SetPoint: </FONT>
    <FONT COLOR=black face=righteous><output form="numform" id="x" name="x" style
= font-size:28px></output></form>
    <br><br>
    <h3 style = font-size:15px><FONT COLOR=black face=righteous>Gustavo Reyes
Navarrete</FONT></h3>
  </form>

  <h3 style = font-size:15px><FONT COLOR=black face=righteous>Si se va a ingresar un
solo digito, recuerde ingresar con un cero inicial </FONT></h3>
  </center></body>
</html>
```