

PROTORIPO FALLENNT

Reporte de Integradora

Presentan

García Carrazco, Mario Kevin	I301910446
Juárez Martínez, Luis Eduardo	I301730050
Lázaro Torres, Juan Valentín	I301010166
Palacios Ayala, Jesus Alejandro	I301910336
Reyes Chávez, Andrés Aldair	I301920087
Rodríguez Preciado, Francisco Ariel	I301910353
Rosas Cortés, Edwin Aldrin	I301510812
Sánchez Cerda, Edgar Israel	I301811221
Vazquez Rivera, Francisco Manuel	I301910661

INDICE

1. INTRODUCCIÓN	3
2. CAPÍTULO I	4
2.1. Objetivo General	4
2.2. Marco Teórico	4
3. CAPÍTULO II	9
3.1. Descripción del proceso del robot Fallennt	9
3.2. Análisis de requerimientos y alcance	11
3.3. Planteamiento del problema	13
3.4. Actividades y roles	14
4. CAPÍTULO III	16
4.1. Desarrollo del prototipo	16
4.2. Problemas enfrentados	18
4.3. Alternativas de solución propuestas.....	18
5. CAPÍTULO IV	20
5.1. Resultados obtenidos	20
5.2. Propuestas para la continuación de trabajo	21
6. CONCLUSIÓN	22
7. BIBLIOGRAFÍA	23
8. ANEXOS	25

INTRODUCCIÓN

La industria de hoy en día avanza a pasos agigantados en la rama de la tecnología, y por esta misma razón los peligros al estar en contacto directo con la maquinaria son muchos. Aun siguiendo las medidas de prevención de accidentes laborales, se no puede garantizar la seguridad del usuario por completo. Según el informe del Observatorio de Seguridad y Salud en el Trabajo, del Consejo Colombiano de Seguridad (CCS), ha reportado que durante el primer trimestre de año 2022 se presentaron 111.396 accidentes de trabajo, con una tasa de 0,99 accidentes por cada 100 trabajadores; y un promedio de 1252 accidentes al día, esto tan solo en el país colombiano. [1]

Con esto nos preguntamos, ¿habrá alguna solución viable para decrementar el número de accidentes?

El proyecto Fallennt, cuyo objetivo es el traslado de objetos a diferentes áreas, otorga la solución a esta problemática. La funcionalidad de Fallennt, hace que sea de gran utilidad en ambientes laborales donde se requiera realizar traslado de productos peligrosos (o de sumo cuidado), así mismo en departamentos para movilizar componentes a distintas áreas donde, para un humano, puede llegar a ser mortal colisionar con robots de grandes escalas que trabajan sin interrupción. Bien es cierto que, se pueden colocar sistemas para la detención de producción en caso de presencia humana, pero dicho paro, causado por aquella presencia implica grandes pérdidas para una empresa. La visión del proyecto Fallennt es disminuir los accidentes en empresas mediante el uso de este robot trasladador, así mismo poder contar con sistemas anticolidión, sistema de orientación, mapeo, y ser un apoyo para los trabajadores y empresarios.

El prototipo Fallennt, que se presenta en la semana tecnológica del 27 de 31 marzo 2023, tendrá una de las funciones presentadas en los párrafos anteriores sobre el Proyecto Fallennt.

El nombre 'Fallennt' proviene de la palabra inglesa *fallen*, que significa "caído" en español, y de la negación inglesa *not* (*n't*). El propósito de este nombre es dar a entender que es un robot anticaída. El sistema anticaída está en base a la estabilización sobre una pelota, a este tipo de robots se les conoce actualmente como "Ballbots". Debido a que es un prototipo, el sistema de navegación y traslado no serán incluidos, sin embargo, la respuesta del sistema al movimiento del sensor IMU (*Inertial Measurment Unit*) se mantendrá.

A lo largo de este reporte, se estará mostrando a forma de memoria del proyecto integrador, los procedimientos que se llevaron a cabo para la obtención del modelo de respuesta a la orientación del sensor de unidad de medición inercia que, mediante un controlador de tipo proporcional, se brindará el manejo de la velocidad de corrección a los actuadores.

CAPÍTULO I

Objetivo general

A forma de prototipo se visualizará la actuación de los motores para demostrar el funcionamiento de la estabilización sin estar montado. A su vez, mediante el código de Python se realiza la simulación de como funcionaría el modelo, si cumpliera con los requisitos mecánicos, y un control de estabilización más avanzado.

Marco Teórico

El sensor Adafruit 9-DOF Absolute Orientation IMU Fusion Breakout – BNO055 a diferencia de otros sensores que suelen requerir de múltiples tareas para hacer que funcione correctamente, hace una combinación óptima de sus algoritmos para extraer las medidas del acelerómetro, giroscopio y magnetómetro, haciendo que este sensor sea una implementación de bajo costo, pero efectiva en tiempos reales de muestreo.

La primera parte del sensor conjunto consiste, primordialmente en la medida proporcionada por medio del acelerómetro. Este es un dispositivo electromecánico utilizado para medir las fuerzas de aceleración que recibe sobre sus ejes. Su análisis de señales es dado por el cambio de la velocidad/rapidez dividida entre el tiempo. Algunas de las fuerzas que puede percibir este dispositivo suelen ser de tipo estáticas, séase como la gravedad que actúa en el acelerómetro directamente a eje z del sensor, esto es así cuando este está acoplado normal a una superficie horizontal. A pesar de su eficacia, este dispositivo sufre de variaciones o ruidos no deseables al extraerse su señal para algún tipo de sistema de control, por ende, es

recomendable utilizar este sensor en conjunto con un giroscopio por medio de la técnica de fusión de sensores.

Los giroscopios son dispositivos que se usan para detectar la desviación de un objeto de con respecto a una posición deseada. En el pasado se solían utilizar giroscopios mecánicos que eran hechos con materiales metálicos, moldeándolos a forma de rueda para obtener giros rápidos. Actualmente son usados dentro brújulas y sistemas de piloto automático en naves, su uso es el mismo que antes pero con análisis más detallados, ya que ayudan a mantener las posiciones adecuadas a su tarea, y otorgan la medición específica de la rapidez de giro por medio de ángulos. La desventaja de usar un giroscopio sin juntar su señal con la de algún otro sensor es que con el tiempo el sensor se ira desfasando de su valor real. Para evitar este problema, se junta con el acelerómetro, usualmente, para que proporcione su rapidez y exactitud en captura de datos momentáneos, y por parte del acelerómetro le otorga el beneficio de que no se desfase o pierda de su punto de origen.

El magnetómetro es un instrumento que sirve para medir la fuerza y la dirección de los campos magnéticos que le rodean. Uno de los campos que mide primordialmente es el de la tierra. Este sensor otorga orientación de direccionamiento gracias a sus mediciones con respecto al campo magnético terrestre.

El cuaternión es parte de números hipercomplejos pertenecientes a los sistemas no conmutativos pero asociativos de la división del algebra. Estos se representan por los siguientes símbolos: H , \mathbf{H} y Q_8 . Este tipo de números, como los números complejos, requieren de una parte real y una imaginaria.

$$H = a \cdot 1 + bi + cj + dk.$$

$$H = \begin{bmatrix} z & w \\ -\bar{w} & \bar{z} \end{bmatrix} = \begin{bmatrix} a + ib & c + id \\ -c + id & a - ib \end{bmatrix}$$

Ecuación 1. Representación de los cuaterniones en forma de combinación lineal y en forma de matriz de 2x2.

El propósito del uso de los cuaterniones es para poder obtener la orientación exacta por medio de las rotaciones de los ejes, esta forma un grupo de orientación tridimensional. Cabe resaltar el uso de la palabra orientación, mas no posición.

Uno de los conceptos primordiales para la elaboración de un sistema de control complejo y eficiente es la combinación de las señales como uno solo, este tipo de táctica es conocida como Sensor Fusion. Específicamente, este hace capaz el juntar diferentes entradas de datos, ya sean cámaras, radares, o sensores de algún otro tipo, pero con el requisito de que se encuentren en el mismo ambiente cercano. El resultado de juntar estas señales es la obtención de un modelo más exacto ya que toma los puntos fuertes de cada sensor y deshecha las desventajas de los sensores. A medida que se aumentan las señales que debe juntar la táctica de sensor fusion se vuelve más compleja, pero se obtiene la oportunidad de mejorar el rendimiento. A continuación, se muestran las ventajas de usar sensor fusión con el sistema del BNO055.

Características	Acelerómetro	Giroscopio	Magnetómetro	Cuaternión	Sensor Fusion
Previene Desviaciones	+	X	X	O	+
Confiabilidad en medidas sin ruido	X	O	O	+	+
Medición de vibraciones	+	O	O	+	+
Rapidez con filtros de atenuación	X	O	O	+	+
Orientación	X	X	O	+	+
Perturbación a campos magnéticos electromecánicos	+	O	X	X	+
Exactitud en generación de ángulos	O	O	O	+	+
Modelado continuo de representación 3D	O	O	O	+	+

Table 1. Modalidades múltiples de sensado requerido, y su integración por medio de Sensor Fusion. Simbología para identificación de características: + = Fortaleza; O = Capacidad; X = Debilidad.

La Tabla 1 indica las fortalezas, debilidades, y capacidades de cada señal de entrada del BNO055, y en su última columna muestra de forma completa como realiza la toma de las mejores características del sensor múltiple.

Dead Reckoning es utilizado para el proceso de calcular la posición actual de un objeto en movimiento al utilizar una determinada posición de referencia. Este proceso provee una estimación de la rapidez, dirección de movimiento, y tiempo transcurrido en la trayectoria recorrida. Esta estrategia o proceso para la navegación, aunque es eficiente es sujeto a errores de aproximación grandes, para la solución a este error se debe asegurar por medio de técnicas de fusión y rendimiento que la rapidez y dirección sean proporcionadas en todo momento a tiempo real.

Debido a la obtención de cuaterniones, para que estos sean representados de forma virtual o más bien, para que estos sean representados en un espacio tridimensional para efectuar rotaciones, es necesario utilizar la fórmula de rotación Rodrigues ya que esta provee un método eficiente computacional para las matrices de rotación fijadas con respecto a un vector unitario de posición.

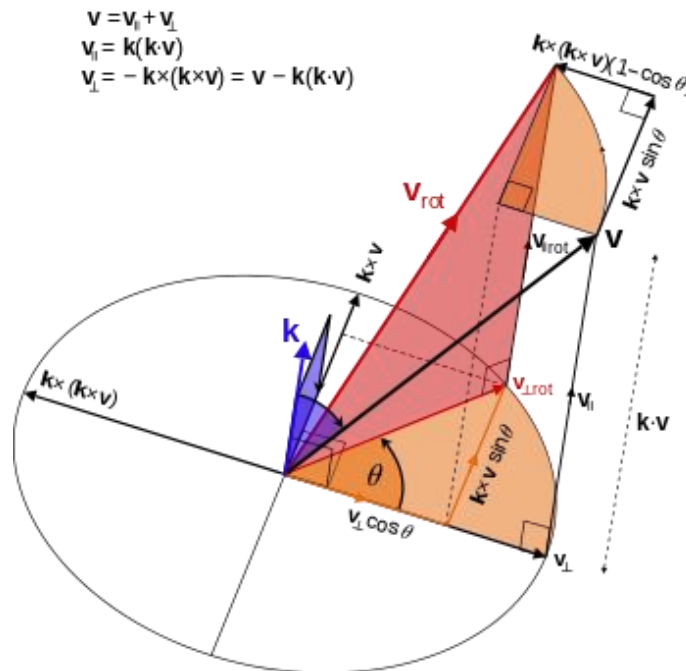


Ilustración 1. Geometría de vectores de la fórmula de rotación de Rodrigues, así también como la descomposición en paralelo y perpendicular de sus componentes. La ecuación se encuentra en la parte superior de la figura.

CAPÍTULO II

Descripción del proceso del robot Fallennt

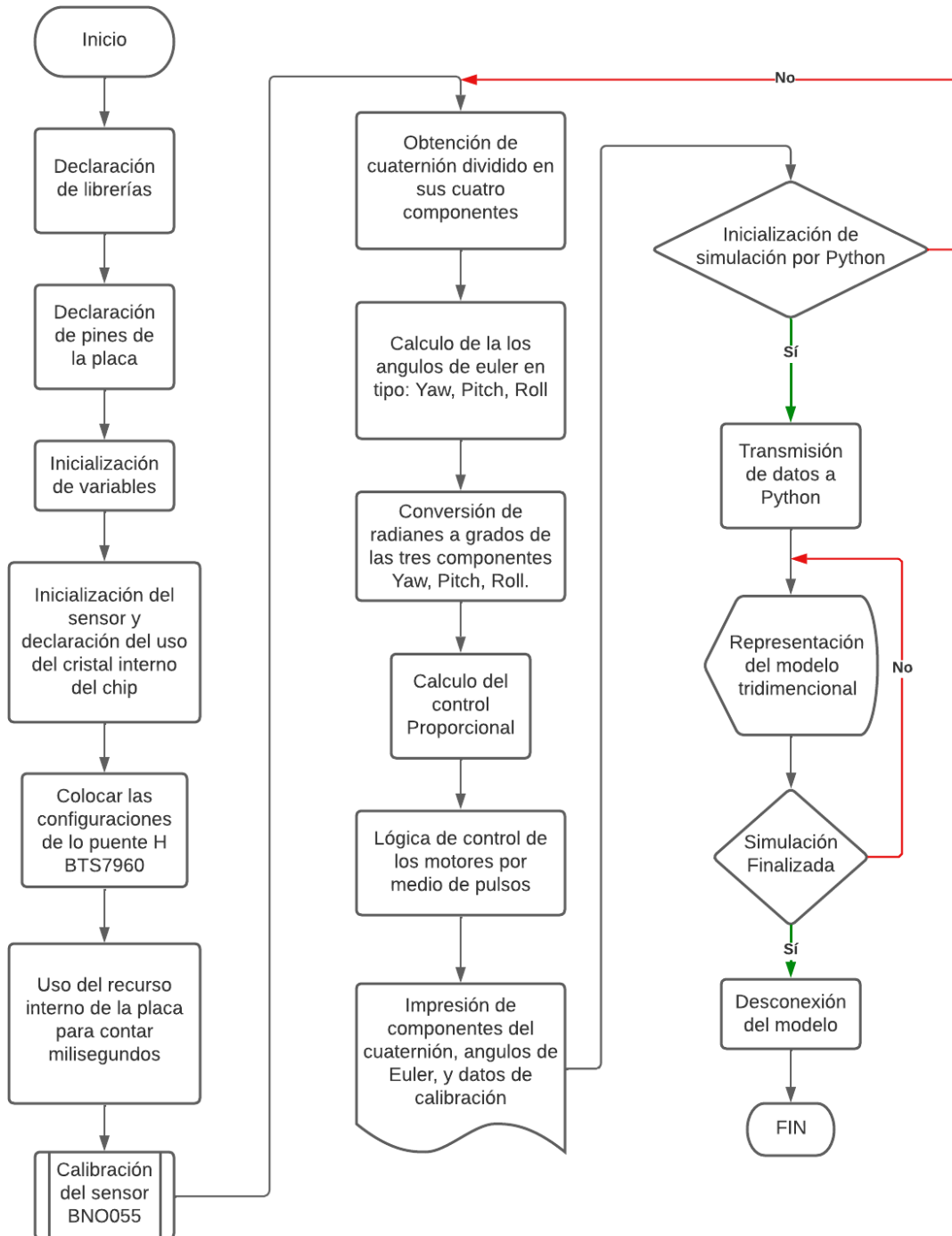


Ilustración 2. Diagrama de Flujo del Proceso del robot Fallennt.

Como se muestra anteriormente, el principio del funcionamiento del robot comienza en la declaración de librerías, variables, dirección de los puertos, iniciación de sensores, colocando las ordenes requeridas del puente H BTS7960 y también el comienzo del conteo del tiempo mediante el chip interno del BNO055, esto con el fin de que el sistema no se vea sobrecargado contando el tiempo y realizando las acciones correctivas.

Posteriormente para la calibración del sistema se requirió de la división de estos componentes internos del sensor, estos son el acelerómetro, el giroscopio, magnetómetro y sistema. Para poder calibrar el primero, se requiere de mover el dispositivo a un ángulo de 45° por cada una de sus componentes, cuando en la hiperterminal se alcance un valor de 3, la calibración estará completa para el acelerómetro. mientras tanto el giroscopio deberá moverse en todas direcciones para poderse calibrar. Para el sistema y el magnetómetro solamente deberán estar en reposo en una superficie horizontal coma así podrán llegar a su punto de calibración.

Al ya haber obtenido las calibraciones, se le indica al sensor que deberá de proporcionar en base a álgebra los cuaterniones y que a su vez estos serán divididos en sus cuatro componentes, o también llamados términos. Cada una de sus componentes otorgarán cierta representación sobre los sensores previamente dichos. Mediante el cálculo de funciones trigonométricas inversas se pueden obtener los ángulos requeridos de Euler llamados Yaw, Pitch y Roll, pero esos estarán dados en términos de radianes. Para poder convertir los radianes a grados se tiene que efectuar una conversión, esta conversión es el número de radianes obtenidos divididos entre la cantidad de 2π que el término es multiplicado completamente por 360° entre uno.

Una vez que se obtienen los ángulos de Euler se procederá analógica del programa que incluye el control proporcional de acuerdo a la medida del ángulo que proveerá la velocidad de corrección del motor y la dirección.

Para iniciar la simulación mediante el programa de Python se deberá iniciar la comunicación entre el puerto serial del microcontrolador y el sistema emulador de Python, esto al ser ejecutado abrirá un servidor local para la representación tridimensional del sistema.

Análisis de requerimientos y alcance

Un robot de tipo ballbot requiere de distintos sistemas para su funcionamiento correcto. Uno de dichos sistemas es el de los actuadores, que en este caso son motores reductores con codificadores ópticos integrados o acoplados para la verificación de la corrección de posición. Requiere a su vez de un sistema de potencia lo suficientemente fuerte como para proveerle a los motores la energía de empuje necesaria para mover el modelo y también para mantenerlo en reposo cuando sea requerido. Es vital el uso de un sensor por medio de tipo IMU con al menos seis ejes, o de manera ideal nueve ejes. Estos ejes se dividen equitativamente entre sus dos/tres (depende del caso de la cantidad de ejes) señales provenientes del chip. La primera de estas señales es la del acelerómetro, la segunda el giroscopio, y la tercera (en el caso de los nueve ejes) el magnetómetro. Dependiendo de las necesidades se puede variar la selección de un sistema de seis ejes con acelerómetro y magnetómetro o la otra configuración de seis ejes por igual, sería con acelerómetro y giroscopio. Continuando con los sistemas, este ballbot requiere de tarjetas de control con protocolos de comunicación entre las mismas, un chip de GPS (por sus siglas en inglés *Global Positioning System*).

De manera similar el robot Fallennt cuenta con los sistemas descritos previamente, con la excepción del sistema de posicionamiento global y los codificadores ópticos, esto es ya que, principalmente el robot no se moverá de posición

ya que por ahora es un prototipo, segundo, en el caso actual no requiere de codificadores ópticos debido a que en esta integradora solo se tratara sobre el control de velocidad y reacciones, la conexión y transmisión de señales por I2C para comunicación entre el sensor IMU y la placa seleccionada.

Debido a las necesidades de la orientación exacta del modelo, se dará uso de una extensión de los números reales que es similar a la de los números complejos, este tipo de numero el cuaternión. El uso de este tipo de número calculado en base al acelerómetro, giroscopio y magnetómetro proporciona la orientación sin ruido.

Planteamiento del problema

Al desarrollar algún tipo de mecanismo, o nuevo tipo de maquina como tal, es sumamente necesario realizar un análisis teórico para ser capaces de que el modelo pueda ser controlado correcta y eficientemente, sin embargo, realizar un control apropiado se debe elaborar un proceso matemático extenso como realizar una comprensión del sistema, un diagrama de cuerpo libre, su modelo matemático desarrollado, aplicar la transformada de Laplace, obtener la función de transferencia, determinar en base a la función de transferencia que controlador será el requerido, y si requiere de ser linealizado, se podrá linealizar por medio de la Transformada Z.

Debido a las complicaciones que conlleva realizar todo el análisis teórico y el tiempo que se toma para un solo sistema, se decidió realizar el análisis simplificado de la estabilización y respuesta por medio de los actuadores y el IMU y que este a su vez mostrara en un entorno tridimensional.

Actividades y roles

Durante el proyecto los distintos integrantes del equipo se dividieron en distintos roles a desempeñar con el fin de aligerar la carga de trabajo de la creación del prototipo Fallent. En la siguiente tabla se muestra cómo se dividieron los roles.

Nombre	Departamento	Actividades
García Carrazco, Mario Kevin	Compras/Programación	<ul style="list-style-type: none">• Análisis de precios, obtención de suministros/componentes• Investigaciones respectivas para el desarrollo del control• Simulación en Python• Programación en IDE
Juárez Martínez, Luis Ernesto	Instrumentación	<ul style="list-style-type: none">• Elaboración de diagramas
Lázaro Torres, Juan Valentín	Administración/Sistema mecánico	<ul style="list-style-type: none">• Elaboración de documentos• Elaboración de estructura y funcionamiento
Palacios Ayala, Jesús Alejandro	Diseño	<ul style="list-style-type: none">• Desarrollo de protocolos, crear, dimensionar y estructurar
Reyes Chávez, Andrés Aldair	Manufactura	<ul style="list-style-type: none">• Corte de piezas, armado de estructura y funcionamiento

Nombre	Departamento	Actividades
Rodríguez Preciado, Francisco Ariel	Administración/Manufactura	<ul style="list-style-type: none"> • Elaboración de documentos • Elaboración de estructura y funcionamiento
Rosas Cortes, Edwin Aldrin	Sistema mecánico/Soldador	<ul style="list-style-type: none"> • Elaboración de estructura y funcionamiento • Soldadura de estructura del equipo
Sánchez Cerda, Edgar Israel	Manufactura	<ul style="list-style-type: none"> • Corte de piezas, armado de estructura y funcionamiento
Vázquez Rivera, Francisco Manuel	Administración/Sistema mecánico	<ul style="list-style-type: none"> • Elaboración de documentos • Elaboración de estructura y funcionamiento

Table 2. Actividades que se desempeñaron por parte de los integrantes del equipo Fallentt.

Como se muestra en esta tabla, cada integrante cumplió con determinado rol que brindó soporte a la creación del prototipo.

CAPÍTULO III

Desarrollo del prototipo

Para la creación del prototipo Fallennt se inició fabricando una guarda utilizando material de policarbonato, que posteriormente mediante el uso de una cortadora con disco de corte se efectuaron las medidas siguientes.

Guarda	Altura	Ancho	Largo	Grosor
Centímetros	65cm	25cm	25cm	.5cm
Metros	.65m	.25m	.25m	.005m

Tabla 3. Medidas de la guarda.

Siguiendo con la fabricación de la estructura del prototipo, se empezó a usar PTR soldando con varilla 6013 de 3/32. Con esto se empezó a darle una forma cuadrada a la base y en la altura se le dio forma rectangular, las medidas siguientes son las que representan el alto, ancho y base del cuadrado y rectángulo.

Estructura	Altura	Ancho	Largo	Grosor
Centímetros	65cm	25cm	25cm	2cm
Metros	.65m	.25m	.25m	.02m

Tabla 4. Medidas del rectángulo y cuadrado.

Ya habiendo finalizado la estructura y las guardas, se comienzan a tomar medidas para determinar los puntos de perforación, se utiliza un taladro con broca de 5/16 de tipo estándar o también llamada de 8 mm. Tomando las medidas en base a la referencia de punto cero de abajo hacia arriba de la estructura y la guarda, teniendo en cuenta que se hará en base a punto

incremental. A continuación, se mostrarán las medidas para la colocación de las pijas.

Estructura	Centímetros	Metros
Pija 1	6cm	.6m
Pija 2	29cm	.29m
Pija 3	54cm	.54m
Guarda	Centímetros	Metros
Pija 1	10cm	.10m
Pija 2	33.5cm	.335m
Pija 3	58cm	.58m

Tabla 5. Medidas para perforación.

Cerca de finalizar con la estructura coma se perforarán 4 puntos de esquinados de la base para colocación de los amortiguadores coma los cuales serán la base de los motores creada a base de soleras. Se soldan 4 puntos con 1° de inclinación de 110 grados utilizando una varilla de 6013. Consecuentemente se perforarán en los ángulos con un traslado con broca 5/16, haciendo de este modo la bancada para la moto reductores así instalando las 3 flechas con las 3 llantas omniwheels para la base del balón. Por último, se implementa una base para el soporte del balón como basta con móvil, así no dejando el peso a los motores y los vástagos de estos.

A continuación, se muestra el diagrama de Gantt que se llevó a cabo para la ejecución del proyecto.

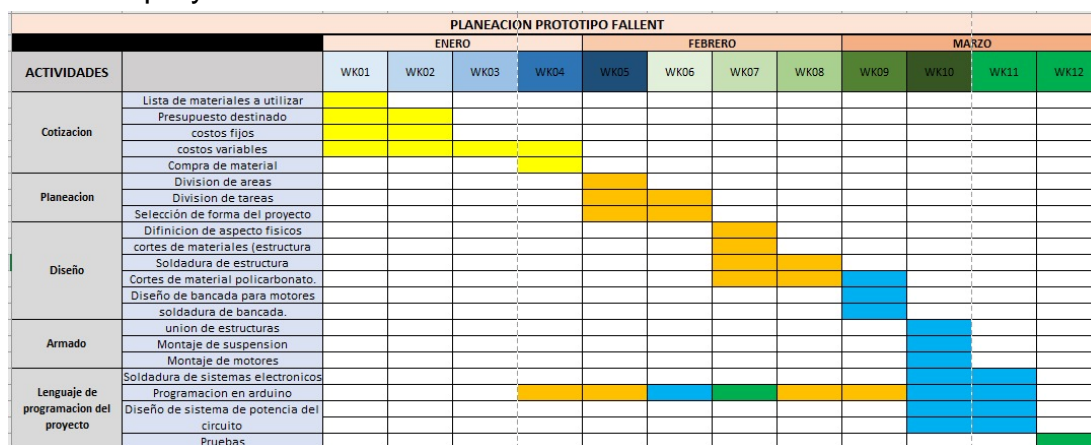


Ilustración 3. Diagrama de Gantt dividido en semanas.

Problemas Enfrentados

Problemática 1. La selección del material óptimo para de la estructura, debía ser un material resistente ante golpes, y que al mismo tiempo no exigiera mucho peso a la estructura, para mantener el modelo simple.

Problemática 2. La inestabilidad de sacar los ángulos Euler para poder hacer uso de las medidas proporcionadas por el sensor IMU (BNO055) realizando funciones tangenciales para obtenerlos, también por medio de incorporar las señales por *sensor fusión* (del acelerómetro, giroscopio y el magnetómetro) y un filtro pasa baja programado.

Problemática 3. Los cálculos de las tangentes para los ángulos de Euler se inestabilizan al llegar a los noventa grados porque se indefine la respuesta.

Problemática 4. La incorporación de los drivers VEX causaban que la velocidad no pudiera controlarse.

Problemática 5. El uso del puente H denominado L298N se sobrecalentaba por la sobredemanda de amperaje, ya que estos están limitados por dos amperes.

Problemática 6. El peso del modelo del robot Fallennt fue demasiado pesado y causó que los vástagos se doblaran, causando inestabilidad.

Alternativas de solución propuestas

Alternativa de solución 1. En la perfilería de Satélite se consiguió un PTR de calibre 20 de $\frac{3}{4}$ por $\frac{3}{4}$ no tan brumoso, de alta resistencia, y que a la vez no resultó tan pesado.

Alternativa de solución 2. Por medio de sensor fusión y el uso del sensor, se extrajeron los cuaterniones y a partir de ellos calcular los ángulos de Euler.

$$\begin{aligned} rollActual &= -\arctan(2 * (q0 * q1 + q2 * q3) 0.1 - 2 * (q1 * q1 + q2 * q2)) \\ pitchActual &= \arcsin(2 * (q0 * q2 - q3 * q1)) \\ yawActual &= -\arctan(2 * (q0 * q3 + q1 * q2) 0.1 - 2 * (q2 * q2 + q3 * q3)) - \pi/2 \end{aligned}$$

Ecuación 2. Ecuaciones para el calculo de los ángulos de Euler en radianes.

Alternativa de solución 3. Se limitaron las respuestas para la respuesta del control por PWM y se configuró a forma de ciclo para que ignorara los valores predecesores.

Alternativa de solución 4. Se cambiaron los drivers VEX por los L298N con el fin de controlar la velocidad del motor por medio de la señal PWM (*Pulse Width Modulation*).

Alternativa de solución 5. Se cambiaron los puentes H (L298N) por los BTS7960, estos últimos son capaces de manejar hasta picos de corriente de hasta cuarenta y tres amperes, por ende, debido a su gran capacidad, se decidieron implementar.

Alternativa de solución 6. Se optó por dividir el robot solo para que mostrara la señal, debido a que ya no se contaba con tiempo y los recursos suficientes para sustituir el material y corregir la estructura.

CAPÍTULO IV

Resultados Obtenidos

El prototipo Fallennt se culminó el día 27 de marzo del año 2023. El resultado de este prototipo culminó en la movilización de la pelota con respecto a los ángulos entrantes, o bien a la orientación y a los movimientos de control para lograr una estabilización simulada en el cual la principal variable controlada fue la velocidad de los motores. A su vez se logró que el sistema se pudiera mostrar haciendo las acciones de corrección al mismo tiempo se mostrará la representación visual en el espacio tridimensional que nos otorga la emulación por Python.

En la siguiente imagen se muestra claramente la pelota cambiando de posición y a la vez el modelo de representación tridimensional está orientado en 2 rotaciones distintas comparando las 2 imágenes una frente a la otra.

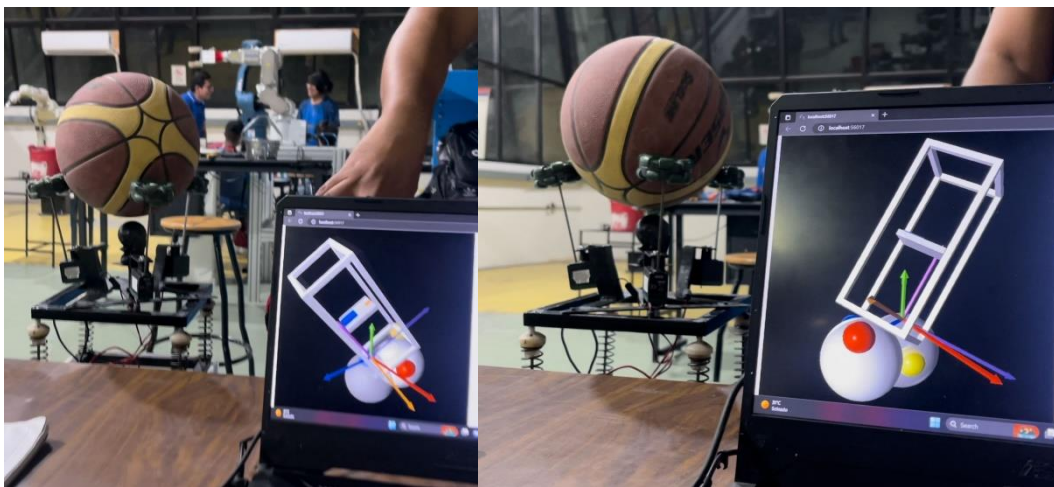


Ilustración 4. Prototipo funcionando y mostrando la simulación a tiempo real.

Propuestas para la continuación del programa de trabajo

Para la continuación de este prototipo, se tendría que incorporar un sistema de control más completo realizando su análisis matemático y colocando un controlador específico para este sistema. Debido a un análisis que se realizó para este prototipo, los integrantes se pudieron dar cuenta que es un sistema no lineal, por ende, pudiera usarse un controlador de tipo LQR (*Linear-quadratic regulator*) para su control exacto.

Debido a que no se pudo realizar el modelo estabilizándose sobre su propio peso, también pudiera quedar a propuesta que para un futuro este proyecto pudiera ser estabilizado sobre su propio sistema, pero para poder lograr esto se deberá contar con un sistema de potencia adecuado, también con actuadores que cuenten con codificadores ópticos y sean capaces de movilizar la estructura, y que a la vez estos tengan una rapidez de entre 70 revoluciones por minuto a 130 revoluciones por minuto. Cabe destacar que este prototipo en otras universidades se desarrolló con un margen de tiempo más amplio debido a que contaban con tres semestres para su elaboración completa y dentro del posgrado (Maestría).

Ya teniendo los componentes necesarios para que el *ballbot* funcione correctamente, y sobre su propia estructura, nada más faltaría añadirle un sistema de posicionamiento global (GPS) para que éste pudiera ser direccionado mediante un control de navegación a distintas partes, y así finalmente cumpliendo con la función primordial del proyecto Fallennt.

CONCLUSION

El equipo del proyecto Fallennt pudo concluir que para la elaboración de un proyecto con controladores ya sean simples o avanzados, se deberá de contar con el análisis matemático del sistema a controlar. Dicho análisis o también llamado modelo matemático deberá de tomar la mayoría de los factores que intervengan dentro del sistema y estos que junto con uno o más controladores puedan hacer las acciones de control necesarias para su funcionamiento específico.

Durante la realización de este proyecto, otro de los aspectos que al equipo le gustaría mencionar es que, es de vital importancia seguir las indicaciones del diseño mecánico y utilizar los materiales apropiados, ya que al no seguir los diagramas el modelo se verá afectado al tratar de realizar acciones de control, ya que probablemente ese sistema no estaba contemplado dentro del análisis.

BIBLIOGRAFÍA

Goodrich, R. (2013, octubre 1). *Accelerometers: What They Are & How They Work*.

LiveScience. <https://www.livescience.com/40102-accelerometers.html>

Britannica, The Editors of Encyclopaedia. "gyroscope". Encyclopedia Britannica, 19

Dec. 2022, <https://www.britannica.com/technology/gyroscope>. Accessed 28 March 2023.

Insider, M. (2020, marzo 3). *What Is Sensor Fusion?* APTIV.

<https://www.aptiv.com/en/insights/article/what-is-sensor-fusion>

Britannica, T. Editors of Encyclopaedia (2016, March 21). magnetometer.

Encyclopedia Britannica.

<https://www.britannica.com/technology/magnetometer>

Weisstein, Eric W. "Quaternion." From MathWorld--A Wolfram Web Resource.

<https://mathworld.wolfram.com/Quaternion.html>

NA. (2023, marzo 27). *Quaternions and spatial rotation*. Wikipedia.

https://en.wikipedia.org/wiki/Quaternions_and_spatial_rotation

NA. (2022, noviembre 27). *Rodrigues' rotation formula*. Wikipedia.

https://en.wikipedia.org/wiki/Rodrigues%27_rotation_formula

Belongie, Serge. "Rodrigues' Rotation Formula." From MathWorld--A Wolfram Web Resource, created by Eric W. Weisstein.

<https://mathworld.wolfram.com/RodriguesRotationFormula.html>

McWhorter, P. (2019, agosto 15). *9 Axis Inertial Measurement Units With Arduino Tutorial*. Youtube.

https://www.youtube.com/watch?v=2AO_Gmh5K3Q&list=PLGsoVKk2DiYwEo-komjIkWXlkrJWAU4L9

ANEXOS

Código

```
//Programación del robot
#include <Wire.h>
#include <Adafruit_Sensor.h>
#include <Adafruit_BNO055.h>
#include <utility/imuMaths.h>
#include <math.h>

//Motor 1 (PWM divided into two factors) YAW
#define R_IS_yaw 8
#define R_EN_yaw 9
#define R_PWM_yaw 2
#define L_IS_yaw 10
#define L_EN_yaw 11
#define L_PWM_yaw 3

//Motor 2 (PWM divided into two factors)PITCH
#define R_IS_pitch 22
#define R_EN_pitch 24
#define R_PWM_pitch 4
#define L_IS_pitch 26
#define L_EN_pitch 28
#define L_PWM_pitch 5

//Motor 3 (PWM divided into two factors)ROLL
#define R_IS_roll 30
#define R_EN_roll 32
#define R_PWM_roll 6
#define L_IS_roll 34
#define L_EN_roll 36
#define L_PWM_roll 7

float q0;
float q1;
float q2;
float q3;

float k1=.7; //proportional
//float k2=70; //derivative
//float k3=.001; //integral

int milliOld;
int milliNew;
int dt;

float rollTarget=0; //Desired Value
float rollActual; //Value of the real position
float rollServoVal=0; //Value sent to the servo
to perform

float rollError=0; //difference between Target
value and Actual value
float rollErrorOld; //the last time through the
loop
//float rollErrorChange;
//float rollErrorSlope; //change of the error over
the time
//float rollErrorArea=0;

float pitchTarget=0; //Desired Value
float pitchActual; //Value of the real position
float pitchServoVal=0; //Value sent to the servo
to perform
float pitchError=0; //difference between Target
value and Actual value
float pitchErrorOld; //the last time through the
loop
//float pitchErrorChange;
//float pitchErrorSlope; //change of the error
over the time
//float pitchErrorArea=0;

float yawTarget=0; //Desired Value
float yawActual; //Value of the real position
float yawServoVal=0; //Value sent to the servo
to perform
float yawError=0; //difference between Target
value and Actual value
float yawErrorOld; //the last time through the
loop
//float yawErrorChange;
//float yawErrorSlope; //change of the error
over the time
//float yawErrorArea=0;

/*If the ACTUAL value that I am measuring is
greater than the TARGET value, the code must
subtract one degree from it until its the same*/

#define BNO055_SAMPLERATE_DELAY_MS
(100) //This indicates the system, how fast it
will work per sample rate

Adafruit_BNO055 myIMU =
Adafruit_BNO055(); //sensor object

void setup() {
```

```
Serial.begin(115200); //the program needs to
be faster because it will show several types of
data
```

```
myIMU.begin(); //Turns on the thing on
delay(1000);
int8_t temp = myIMU.getTemp(); //store a
number from 128 to 127
//Serial.println(temp);
myIMU.setExtCrystalUse(true); //says not to
use the crystal of the chip but on the board
```

```
//Declaring all the pins as input or output
pinMode(R_IS_yaw,OUTPUT);
pinMode(R_EN_yaw,OUTPUT);
pinMode(R_PWM_yaw,OUTPUT);
pinMode(L_IS_yaw,OUTPUT);
pinMode(L_EN_yaw,OUTPUT);
pinMode(L_PWM_yaw,OUTPUT);
```

```
pinMode(R_IS_pitch,OUTPUT);
pinMode(R_EN_pitch,OUTPUT);
pinMode(R_PWM_pitch,OUTPUT);
pinMode(L_IS_pitch,OUTPUT);
pinMode(L_EN_pitch,OUTPUT);
pinMode(L_PWM_pitch,OUTPUT);
```

```
pinMode(R_IS_roll,OUTPUT);
pinMode(R_EN_roll,OUTPUT);
pinMode(R_PWM_roll,OUTPUT);
pinMode(L_IS_roll,OUTPUT);
pinMode(L_EN_roll,OUTPUT);
pinMode(L_PWM_roll,OUTPUT);
```

```
//Original Way of begin working
digitalWrite(R_IS_yaw,LOW);
digitalWrite(R_EN_yaw,HIGH);
digitalWrite(L_IS_yaw,LOW);
digitalWrite(L_EN_yaw,HIGH);
```

```
digitalWrite(R_IS_pitch,LOW);
digitalWrite(R_EN_pitch,HIGH);
digitalWrite(L_IS_pitch,LOW);
digitalWrite(L_EN_pitch,HIGH);
```

```
digitalWrite(R_IS_roll,LOW);
digitalWrite(R_EN_roll,HIGH);
digitalWrite(L_IS_roll,LOW);
digitalWrite(L_EN_roll,HIGH);
```

```
milliNew=millis();
}
```

```
void loop() {
  uint8_t system, gyro, accel, mg = 0; //efficient
  way to store data
```

```
myIMU.getCalibration(&system, &gyro,
&accel, &mg);
imu::Quaternion quat=myIMU.getQuat(); //Ask
the BNO055 to get the quaternion vector for u
```

```
q0=quat.w();
q1=quat.x();
q2=quat.y();
q3=quat.z();
```

```
rollActual=-atan2(2*(q0*q1+q2*q3),1-
2*(q1*q1+q2*q2)); //Calculation of the actual
position of the roll on radians
pitchActual=asin(2*(q0*q2-q3*q1));
yawActual=-atan2(2*(q0*q3+q1*q2),1-
2*(q2*q2+q3*q3))-3.141592654/2;
```

```
rollActual=rollActual/(2*3.141592654)*360;
//Calculation of the actual position of the roll on
degrees
pitchActual=pitchActual/(2*3.141592654)*360;
yawActual=(yawActual/(2*3.141592654))*360;
//
milliOld=milliNew;
milliNew=millis();
dt=milliNew-milliOld;
```

```
//*****PID roll
rollErrorOld=rollError;
rollError=rollTarget-rollActual; //p
//rollErrorChange=rollError-rollErrorOld;
//rollErrorSlope=rollErrorChange/dt; //d
//rollErrorArea=rollErrorArea+rollError*dt; //i
```

```
rollServoVal=k1*rollError;
//+k2*rollErrorSlope+k3*rollErrorArea;
```

```
//Determine whether it will go right or left
```

```
if(rollActual>10 && rollActual<=87){ //it will start
correcting the value depending on the
direction
// digitalWrite(in1yaw,HIGH); //these are not
longer needed when using BRS7960
// digitalWrite(in2yaw,LOW);
```

```
analogWrite(R_PWM_yaw,0); //speed and
direction
```

```
analogWrite(L_PWM_yaw,abs(rollServoVal)+1
5); //one one zero to indicate the direction in
which is going
```

```
analogWrite(R_PWM_pitch,abs(rollServoVal)+
15);
analogWrite(L_PWM_pitch,0);
```

```

    analogWrite(R_PWM_roll,0);
    analogWrite(L_PWM_roll,0);
}
if(rollActual<-10 && rollActual>=87){
// digitalWrite(in1yaw,LOW);
// digitalWrite(in2yaw,HIGH);

analogWrite(R_PWM_yaw,abs(rollServoVal)+15);
analogWrite(L_PWM_yaw,0);
analogWrite(R_PWM_pitch,0);

analogWrite(L_PWM_pitch,abs(rollServoVal)+15);
analogWrite(R_PWM_roll,0);
analogWrite(L_PWM_roll,0);
}
else{
    analogWrite(R_PWM_yaw,rollServoVal);
    analogWrite(L_PWM_yaw,0);
    analogWrite(R_PWM_pitch,rollServoVal);
    analogWrite(L_PWM_pitch,0);
    analogWrite(R_PWM_roll,rollServoVal);
    analogWrite(L_PWM_roll,0);
}

```

```

//*****PID pitch
pitchErrorOld=pitchError;
pitchError=pitchTarget-pitchActual;
//pitchErrorChange=pitchError-pitchErrorOld;
//pitchErrorSlope=pitchErrorChange/dt;
//pitchErrorArea=pitchErrorArea+pitchError*dt;

```

```

pitchServoVal=k1*pitchError;
//+k2*pitchErrorSlope+k3*pitchErrorArea;
//Determine whether it will go right or left

```

```

if(pitchActual>10 && pitchActual<=87){ //it will
start correccting the value depending on the
direction
// digitalWrite(in1yaw,HIGH); //these are not
longer needed when using BRS7960
// digitalWrite(in2yaw,LOW);

```

```

analogWrite(R_PWM_yaw,abs(pitchServoVal)+15); //speed and direction
    analogWrite(L_PWM_yaw,0); //one one cero
to indicate the direction in which is going
    analogWrite(R_PWM_pitch,0);
    analogWrite(L_PWM_pitch,0);
    analogWrite(R_PWM_roll,0);

```

```

    analogWrite(L_PWM_roll,abs(pitchServoVal)+15);
}
if(pitchActual<-10 && pitchActual>=87){
// digitalWrite(in1yaw,LOW);
// digitalWrite(in2yaw,HIGH);

```

```

    analogWrite(R_PWM_yaw,0);

```

```

    analogWrite(L_PWM_yaw,abs(pitchServoVal)+15);
    analogWrite(R_PWM_pitch,0);
    analogWrite(L_PWM_pitch,0);

```

```

    analogWrite(R_PWM_roll,abs(pitchServoVal)+15);
    analogWrite(L_PWM_roll,0);
}

```

```

else{
    analogWrite(R_PWM_yaw,pitchServoVal);
    analogWrite(L_PWM_yaw,0);
    analogWrite(R_PWM_pitch,pitchServoVal);
    analogWrite(L_PWM_pitch,0);
    analogWrite(R_PWM_roll,pitchServoVal);
    analogWrite(L_PWM_roll,0);
}

```

```

//*****PID yaw
yawErrorOld=yawError;
yawError=yawTarget-yawActual;
//yawErrorChange=yawError-yawErrorOld;
//yawErrorSlope=yawErrorChange/dt;
//yawErrorArea=yawErrorArea+yawError*dt;

```

```

yawServoVal=k1*yawError;//+k2*yawErrorSlope+k3*yawErrorArea;

```

```

if(yawActual>10 && yawActual<=87){ //it will
start correccting the value depending on the
direction
// digitalWrite(in1yaw,HIGH); //these are not
longer needed when using BRS7960
// digitalWrite(in2yaw,LOW);

```

```

    analogWrite(R_PWM_yaw,abs(yawServoVal)+15); //speed and direction
    analogWrite(L_PWM_yaw,0); //one one cero
to indicate the direction in which is going

```

```

    analogWrite(R_PWM_pitch,abs(yawServoVal)+15);
    analogWrite(L_PWM_pitch,0);

```

```

analogWrite(R_PWM_roll,abs(yawServoVal)+1
5);
  analogWrite(L_PWM_roll,0);
}
if(yawActual<-10 && yawActual>=-87){

  analogWrite(R_PWM_yaw,0);

analogWrite(L_PWM_yaw,abs(yawServoVal)+
15);
  analogWrite(R_PWM_pitch,0);

analogWrite(L_PWM_pitch,abs(yawServoVal)+
15);
  analogWrite(R_PWM_roll,0);

analogWrite(L_PWM_roll,abs(yawServoVal)+1
5);
}
else{
  analogWrite(R_PWM_yaw,yawServoVal);
  analogWrite(L_PWM_yaw,0);
  analogWrite(R_PWM_pitch,yawServoVal);
  analogWrite(L_PWM_pitch,0);
  analogWrite(R_PWM_roll,yawServoVal);
  analogWrite(L_PWM_roll,0);
}

//printing stuff
Serial.print(quat.w());
Serial.print(",");
Serial.print(quat.x());
Serial.print(",");
Serial.print(quat.y());
Serial.print(",");
Serial.print(quat.z());
Serial.print(", ");
Serial.print(pitchActual); //y
Serial.print(",");
Serial.print(yawActual); //z
Serial.print(",");
Serial.print(rollActual); //x
Serial.print(", ");
Serial.print(accel);
Serial.print(",");
Serial.print(gyro);
Serial.print(",");
Serial.print(mg);
Serial.print(",");
Serial.println(system);

  delay(BNO055_SAMPLERATE_DELAY_MS);
}

```