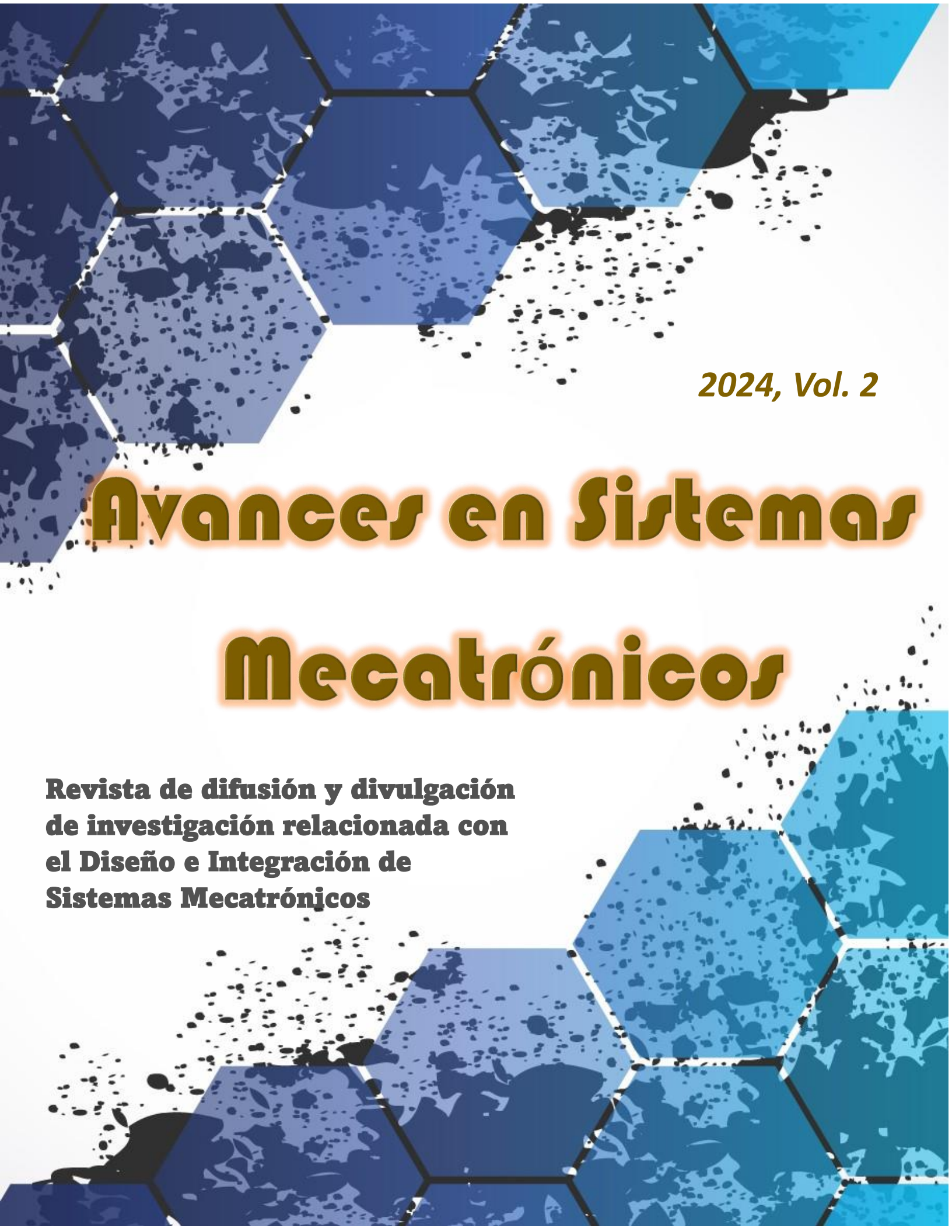




2024, Vol. 2

Avances en Sistemas Mecatrónicos

**Revista de difusión y divulgación
de investigación relacionada con
el Diseño e Integración de
Sistemas Mecatrónicos**

Directorio

Dr. Víctor Hugo Jiménez Arredondo

*Director de la División de Ingenierías
Campus Irapuato-Salamanca*

Dr. Antonio de Jesús Balvantín García

*Director del Departamento de Ingeniería
Mecánica*

Dr. Israel Martínez Ramírez

*Coordinador del Programa Educativo de
Licenciatura en Ingeniería Mecatrónica*

Dr. Felipe de Jesús Torres del Carmen

Editor en jefe

La revista de Avances de Sistemas Mecatrónicos es dedicada a la difusión y divulgación de investigación relacionada con los Sistemas Mecatrónicos en temas relevantes para la sociedad en general. Puede ser consultada en el sitio:

<https://www.ugto.mx/campusirapuatosalamanca/sisimek2024>

Las opiniones expresadas por los autores no reflejan necesariamente las opiniones del consejo editorial de la publicación

Consejo Editorial

Dr. Felipe de Jesús Torres del Carmen

Dr. Diego Alfredo Núñez Altamirano

Dr. Antonio de Jesús Balvantín García

Dr. Juan Pablo Ignacio Ramírez Paredes

Dr. Israel Martínez Ramírez

Dr. Gustavo Capilla González

*División de Ingenierías Campus Irapuato-
Salamanca*

Universidad de Guanajuato

Consejo Arbitral

M.C. Fabio Abel Aguirre Cerrillo

Instituto Tecnológico de Lázaro Cárdenas

Dr. Diego Alfredo Núñez Altamirano

Universidad de Guanajuato

Dr. Antonio de Jesús Balvantín García

Universidad de Guanajuato

Dr. Juan Pablo Ignacio Ramírez Paredes

Universidad de Guanajuato

Dr. Gustavo Capilla González

Universidad de Guanajuato

Dr. Daniel Edahi Urueta Hinojosa

Universidad Politécnica del Estado de Guerrero

Dr. Marco Antonio Gómez Álvarez

Universidad de Guanajuato

Dr. Felipe de Jesús Trujillo Romero

Universidad de Guanajuato

Dr. Eric Noé Hernández Rodríguez

Universidad de Guanajuato

Dr. Ricardo García García

Universidad de Guanajuato

Dr. Israel Martínez Ramírez

Universidad de Guanajuato

Dra. Miroslava Cano Lara

TecNM/Instituto Tecnológico Superior de Irapuato

M.C. Juan Esteban Mota Cruz

Universidad Politécnica del Estado de Guerrero

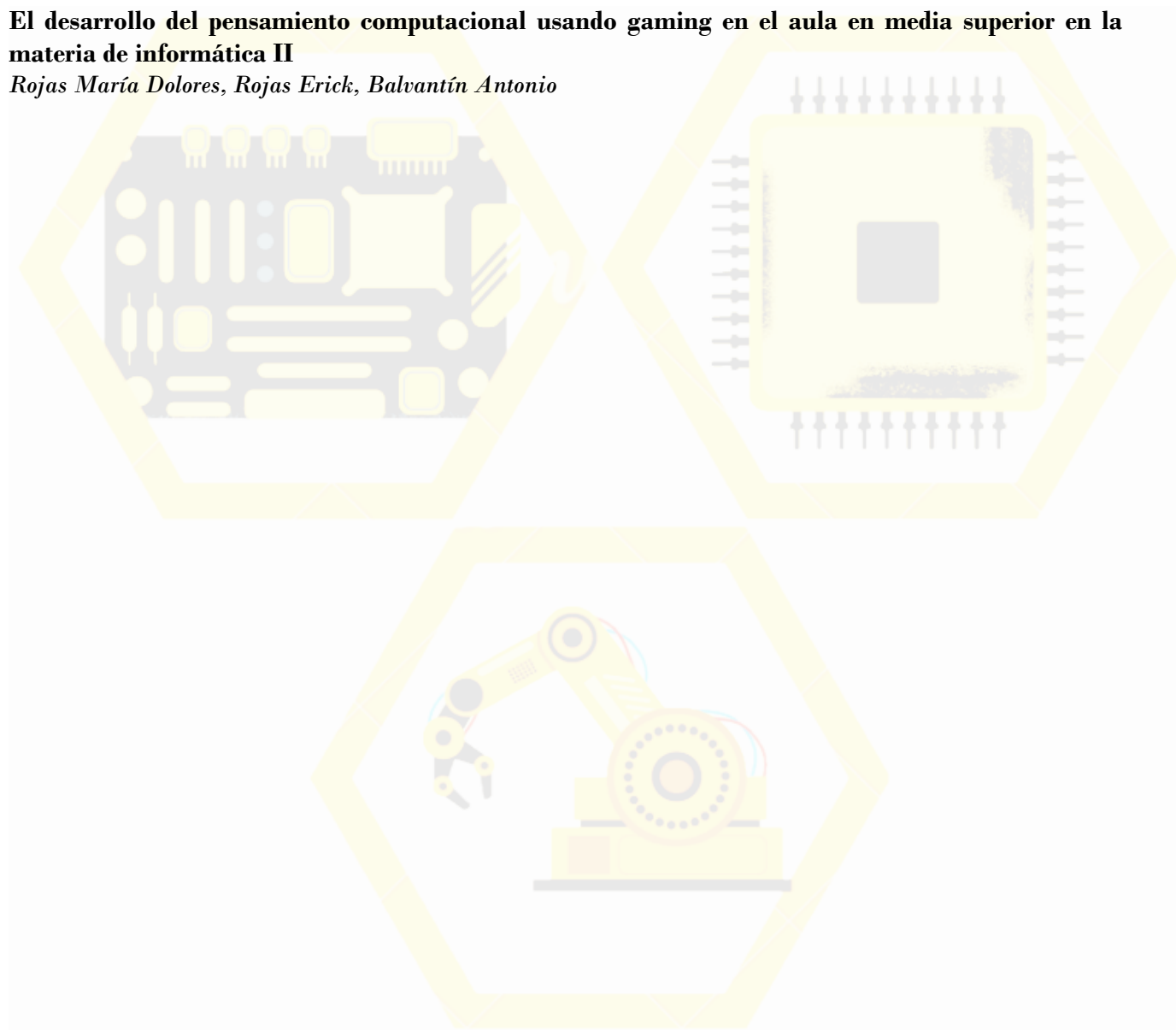
Contenido

Evaluación Objetiva de Procesadores Personales de Alimentos: Un Enfoque Basado en Técnicas de Inteligencia Artificial	7
<i>Urueta Daniel Edahi, Mota Juan Esteban, Lavín Jorge Enrique, Valois Marco Antonio.</i>	
Estación modular mecatrónica – Eje lineal y Pick and place giratorio	14
<i>Aguirre Cerrillo Fabio Abel, Luna Castro Arturo, López Salas Fernando y Chávez Herrera Daniel Alejandro</i>	
Representación del control de velocidad de un vehículo autónomo por un sistema Bola Balancín	21
<i>Rafael Dassaev Flores López</i>	
Interfaz de Monitoreo de la Trayectoria de un Brazo Robótico de Dos Grados de Libertad	29
<i>Mendoza Rai, Muñoz Johan, Ponce Jonathan, Reyes Alejandro</i>	
Adquisición de datos de sobrealimentador eléctrico para motor de combustión interna	36
<i>Ayala García José Francisco, Muñoz Carpio Vicente Daniel, Torres Luna Juan Ignacio, Chávez Razo Francisco Gustavo</i>	
Control PD de un quadrotor	42
<i>Lavín Delgado Jorge Enrique, Mota Cruz Juan Esteban, Urueta Hinojosa Daniel Edahi</i>	
Reconstrucción de la Marcha Utilizando Unidades de Medición Inerciales en Miembros Inferiores	47
<i>Barón Casique Jeymar J., Mireles Vázquez Miguel Ángel, Ramírez Paredes Juan Pablo I., Correa Tomé Fernando E., Torres del Carmen Felipe, Arellano Vázquez Magali</i>	
Trazado de Trayectorias Libres de Colisiones con Optimización por Enjambre de Partículas	53
<i>Duarte Rostro Axel J., Zaragoza Hernández Francisco A., Hernández Belmonte Uriel H., Ramírez Paredes Juan Pablo I.</i>	
Exploración Robótica Autónoma con Vehículos Terrestres no Tripulados Aplicando Minimización de Entropía	59
<i>Rojas-Andrade Miguel A., Ramírez-Paredes Juan Pablo I., Hernández-Belmonte Uriel H.</i>	
Implementación de Control de Carril Basado en Visión Artificial para Vehículos Autónomos a Escala	66
<i>Alcocer Soriano Carlo Leonardo, Mireles Vázquez Miguel Ángel, Ramírez Paredes Juan Pablo I.</i>	
Monitoreo del Comportamiento Térmico en Probetas PU Expuestas a Flamabilidad	71
<i>Sandoval Martínez Karen J., Tafolla Barajas Jessica G., García Guzmán José M., Razón González Juan P., Rostro González Horacio, Cano Lara Miroslava</i>	
Simulación cinemática de robot SCARA	79
<i>García García Ricardo, Rico Martínez José María, Cervantes Sánchez J. Jesús, García Murillo Mario Alberto, Rodríguez Ramírez Cristian, Ayala García Francisco</i>	

Refactorización de Diagramas de Clase UML Utilizados en Estrategias de Enseñanza-Aprendizaje <i>Desiderio-Patiño Arelia, Fragoso-Díaz Olivia Graciela, Rojas-Pérez Juan Carlos</i>	88
--	-----------

Integración de sistema de adquisición de sonidos para análisis de frecuencias de resonancia en jitomates <i>Rojas-Mancera Erick, Flores-Placencia Giovana, Delgado-Franco Estefanía, Pulido-Ruiz Frida, Balvantín-García Antonio</i>	94
--	-----------

El desarrollo del pensamiento computacional usando gaming en el aula en media superior en la materia de informática II <i>Rojas María Dolores, Rojas Erick, Balvantín Antonio</i>	99
---	-----------





Evaluación Objetiva de Procesadores Personales de Alimentos: Un Enfoque Basado en Técnicas de Inteligencia Artificial.

Urueta Daniel Edahi^{*1}, Mota Juan Esteban¹, Lavín Jorge Enrique¹, Valois Marco Antonio¹

¹ *Universidad Politécnica del Estado de Guerrero, Ingeniería en Redes y Telecomunicaciones, Carretera
Federal Iguala-Taxco, Comunidad de Puente Campuzano.*

** Autor de correspondencia: daniel.urueta@upeg.edu.mx*

Resumen. Los procesadores personales de alimentos son dispositivos diseñados para mezclar y licuar alimentos en porciones individuales. Estos aparatos que integran componentes eléctricos y electrónicos están estrechamente vinculados a la automatización de tareas y el control inteligente, conceptos fundamentales en la mecatrónica. Aunque estos dispositivos parecen tener prestaciones similares, la publicidad y el costo sugieren diferencias significativas. Por ello, la Procuraduría Federal del Consumidor (PROFECO) realizó un estudio en 2024 para comparar detalladamente estos productos y ayudar a los consumidores a tomar decisiones. El estudio clasificó a estos dispositivos en una escala de: excelente (E), muy bueno (MB) y bueno (B). Sin embargo, el estudio no proporciona detalles sobre la metodología utilizada, lo que podría introducir un sesgo en los resultados.

Para abordar esta limitación, este trabajo propone una metodología basada en algoritmos no supervisados de inteligencia artificial para analizar y clasificar los procesadores personales de alimentos de manera objetiva. Esta metodología permite identificar diferencias reales en las prestaciones de los dispositivos y comparar los resultados con las clasificaciones existentes. Además, proporciona ideas para mejorar el diseño y fabricación de los mismos, fomentando la innovación en la industria mecatrónica.

Finalmente, los datos y análisis presentados en este estudio pueden ser utilizados como caso de estudio en programas de ingeniería, demostrando la aplicación de técnicas de análisis estadístico y algoritmos de inteligencia artificial en el campo de la mecatrónica.

Palabras clave. *Estudio comparativo, procesadores personales de alimentos, reconocimiento de patrones.*



I. INTRODUCCIÓN

La mecatrónica desempeña un papel crucial en el desarrollo de tecnologías avanzadas que mejoran la vida cotidiana, la importancia de esta disciplina en la innovación tecnológica se ve reflejada en la capacidad de integrar sistemas mecánicos, electrónicos y de control en dispositivos compactos y funcionales, como los procesadores personales de alimentos.

Los procesadores personales de alimentos han emergido como dispositivos esenciales en las cocinas modernas, diseñados para automatizar una variedad de tareas culinarias y simplificar la preparación de alimentos mediante la mezcla, licuado y triturado de ingredientes como frutas, verdura y hielo en porciones individuales sin necesitar demasiado espacio.

No obstante, de que a primera vista los procesadores personales de alimentos pueden parecer similares en cuanto a sus capacidades, la variabilidad en su publicidad y costo sugiere que existen diferencias significativas en sus prestaciones y calidad. Esta disparidad resalta la necesidad de realizar estudios detallados que permitan a los consumidores tomar decisiones informadas. En respuesta a esta necesidad, la Procuraduría Federal del Consumidor (PROFECO) llevó a cabo en 2024 un estudio exhaustivo de estos dispositivos (PROFECO, 2024), con la finalidad de proporcionar una comparación detallada y objetiva que permita a los consumidores tomar la mejor decisión previo a adquirir un dispositivo de esta índole.

El estudio de PROFECO evaluó diez modelos de procesadores personales de alimentos pertenecientes a ocho marcas diferentes. La evaluación incluyó la recolección de datos cuantitativos sobre múltiples aspectos de desempeño, tales como: consumo energético, nivel de ruido, estabilidad, hermeticidad, y precio; estos aspectos sirvieron como punto de partida para puntuar a estos dispositivos en una escala que los clasifica en tres categorías: excelente (E), muy bueno (MB) y bueno (B). Sin embargo, la metodología y el sistema de clasificación utilizados en este estudio no fueron detalladamente explicados, lo que puede introducir un sesgo en los resultados y limitaciones en la interpretación de los mismos.

Para abordar esta limitación y lograr una evaluación objetiva, el presente trabajo propone una metodología innovadora basada en algoritmos no supervisados de inteligencia artificial para analizar y clasificar los procesadores personales de alimentos de manera más rigurosa, destacando las verdaderas diferencias en sus prestaciones.

II. OBJETIVOS

a. General

- Usar técnicas de reconocimiento de patrones para clasificar los procesadores personales de alimentos analizados por PROFECO en 2024.

b. Específicos

- Comparar la clasificación reportada por PROFECO con la obtenida por el modelo propuesto.
- Complementar los estudios realizados por PROFECO.
- Ofrecer a los consumidores información sobre el proceso de clasificación y los resultados, que les permita tomar mejores decisiones cuando deseen adquirir un procesador de alimentos personal.

III. MARCO TEÓRICO

a. PROFECO

Es un instituto creado para proteger y defender los derechos de las y los consumidores, generar una cultura de consumo responsable por medio de información oportuna y objetiva para la toma de decisiones de consumo e implementando métodos de atención pronta y accesible a la diversidad de consumidoras y consumidores mediante el uso de tecnologías de la información (Gobierno de México, 2024).



b. Aprendizaje no supervisado

Dentro de la inteligencia artificial, existen algoritmos los cuales no necesitan ser entrenados para poder clasificar un conjunto de datos, sino que su clasificación la hacen por medio de las características inherentes de cada elemento, este tipo de técnicas son denominadas de clasificación no supervisada o *clustering* y son particularmente útiles cuando no se tiene un etiquetado previo, es decir, que se desconoce cómo puede organizarse un conjunto de datos y es necesario clasificarlos; por otro lado, también se emplean con fines de comparación y retroalimentación entre los resultados que arroja el modelo y aquellos que determina un especialista.

c. Modelo de Mezclas Gaussianas (GMM)

Es un enfoque probabilístico que parte del supuesto de que todos los datos en un conjunto provienen de una mezcla finita de distribuciones gaussianas, cada una con parámetros desconocidos. Se puede considerar que GMM es una generalización del algoritmo *k-medias* que añade información sobre la estructura de covarianza de los datos y los centros de las distribuciones latentes. El objetivo de un GMM es identificar la mezcla de distribuciones gaussianas multidimensionales que mejor se ajuste a un determinado conjunto de datos (Urueta *et al.*, 2021).

d. Coeficiente de la silueta

Un problema del algoritmo GMM es que para funcionar es necesario especificar de antemano el número de grupos (*clusters*) en los que se clasificará; para determinar este valor existen numerosos métodos, en este trabajo se usó el coeficiente de la silueta.

El coeficiente de silueta o puntaje de silueta es una medida de validez de clúster para evaluar el rendimiento de la agrupación. Para calcular el puntaje de silueta para cada observación/punto de datos, se deben encontrar las siguientes distancias:

1. Distancia promedio entre la observación y todos los demás puntos de datos en el mismo cluster.

Esta distancia también puede llamarse distancia intracluster promedio. Esta distancia promedio se denota por $a(i)$.

2. Distancia promedio entre la observación y todos los demás puntos de datos del cluster más cercano siguiente. Esta distancia también puede llamarse distancia media al cluster más cercano. La distancia promedio se denota por $b(i)$.
3. El puntaje de silueta, $s(i)$, para cada muestra se calcula utilizando la siguiente fórmula:

$$s(i) = \frac{b(i) - a(i)}{\max \{a(i) - b(i)\}} \quad (1)$$

Los valores del coeficiente de silueta van desde -1 hasta 1.

En donde los valores cercanos a 1 indican que los clusters están bien separados entre sí y claramente distinguibles. Un valor de 0 indica que los clusters son indiferentes, o podemos decir que la distancia entre los clústeres no es significativa, y los valores negativos indican que la configuración de agrupación puede tener demasiados o demasiados pocos clusters. Dado que los coeficientes de silueta se utilizan para estudiar la distancia de separación entre los clústeres resultantes, es posible utilizarlos para seleccionar el número de clusters en técnicas de agrupación.

e. Análisis de Componentes Principales: PCA

Es una técnica que permite reducir las dimensiones de un conjunto de datos, tratando de preservar con cada uno de sus componentes, la mayor variación de los mismos. Esto lo hace buscando posibles correlaciones por medio de líneas, planos e hiperplanos en el espacio original que aproximan los datos lo mejor posible en el sentido de los mínimos cuadrados (Urueta Hinojosa *et al.*, 2024).

f. Coeficiente de correlación de Pearson

Es una medida que cuantifica la relación lineal entre dos variables. Este coeficiente se denota con la letra r .



El resultado oscilará entre -1 y +1, y se puede interpretar como en la figura 1.

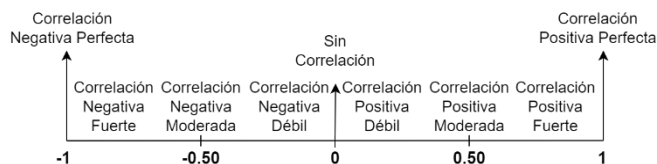


Fig. 1. Explicación de coeficientes de correlación.

IV. MATERIALES Y MÉTODOS

a. Materiales

- Base de datos generada a partir de los estudios reportados por PROFECO en (PROFECO, 2024) tomando en cuenta las siguientes características: Garantía (años), Potencia (W), Desempeño de licuado, Nivel de ruido, Rupturas, Calentamiento y Costo.
- Lenguaje de programación Python para analizar las bases de datos generadas.

b. Métodos

- Transformar variables categóricas a valores numéricos.
- Normalizar la base de datos.
- Usar el coeficiente de la silueta para determinar el número adecuado de clases para la base de datos.
- Usar GMM para clasificar la base de datos con el número de clases obtenido previamente.
- Analizar los resultados de la clasificación.
- Comparar los resultados obtenidos con la clasificación proporcionada por PROFECO.
- Usar PCA sobre la base de datos para hacer más comprensible y visual la comunicación de resultados.

V. RESULTADOS

a. Determinar el número de grupos

Como se mencionó en apartados previos, antes de usar el algoritmo GMM, es necesario determinar el número adecuado de grupos de cada base de datos particular, a continuación, se muestran los resultados de la implementación del coeficiente de la silueta para la base de datos generada.

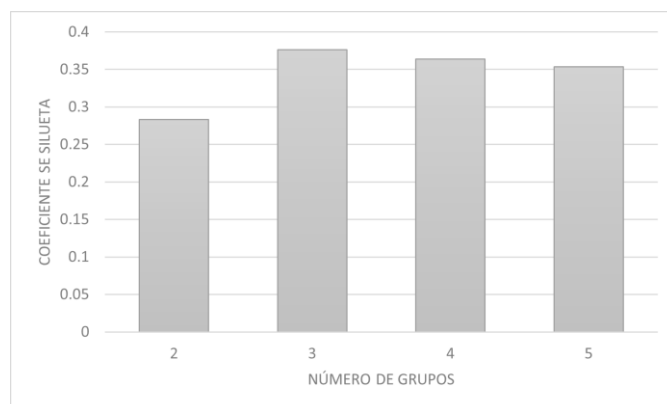


Fig. 2. Mejores resultados del coeficiente de la silueta.

Analizando la Fig. 2 se determina de manera clara que, para esa base de datos, el número recomendable (valor más alto) de grupos para clasificar es de 3.

b. Clasificación y reducción de dimensiones

Posterior a determinar el número de clases se usó el algoritmo de GMM para hacer la clasificación en la cual, si bien fueron consideradas todas sus características de la base de datos original, fue necesario usar PCA para reducir las dimensiones de la misma a 3 para poder visualizar mejor los resultados, manteniendo una representatividad de los datos del 87%. Los resultados se muestran a continuación.

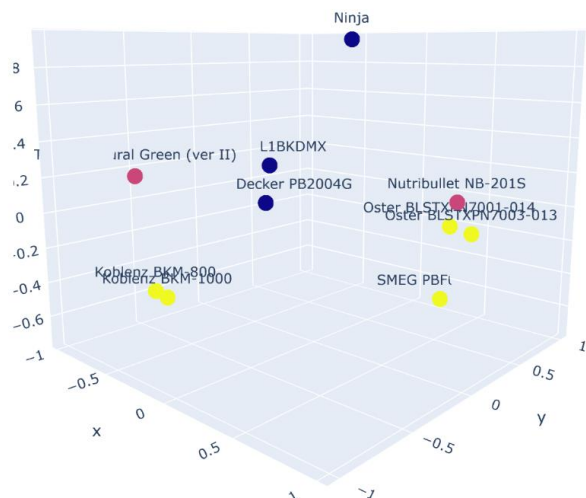


Fig. 3. Resultados de la clasificación y PCA con clasificaciones de PROFECO.

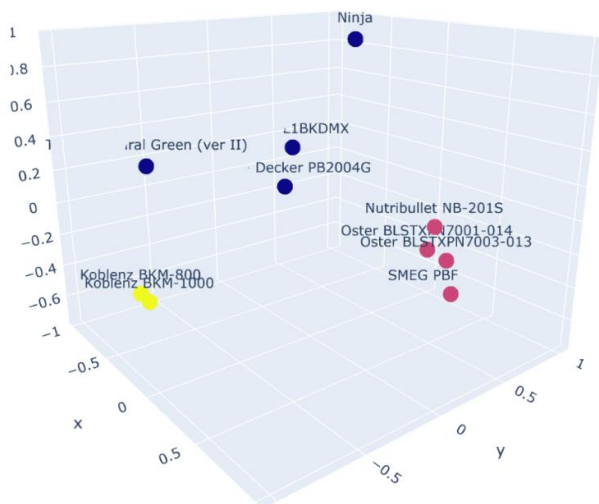


Fig. 4. Resultados de la clasificación y PCA con clasificaciones GMM.

En ambos casos, los valores mostrados en la Fig. 3 y Fig. 4, contrastan en amarillo aquellos productos calificados como: Bien; en rosa los: Muy bien y en morado los: Excelente.

Análisis de resultados con correlaciones

Usando el coeficiente de correlación de Pearson se analizaron las características y las clasificaciones de PROFECO y del modelo propuesto.

Garantía (años)	1.00	0.19	-0.00	0.09	0.58	-0.17	-0.58	-0.05
Potencia (W)	0.19	1.00	0.30	0.83	-0.03	0.54	0.51	-0.49
Desempeño de licuado	-0.00	0.30	1.00	0.41	0.25	-0.25	-0.14	0.43
Nivel de ruido	0.09	0.83	0.41	1.00	0.09	0.36	0.42	-0.40
Rupturas	0.58	-0.03	0.25	0.09	1.00	-0.17	-0.54	-0.05
Calentamiento	-0.17	0.54	-0.25	0.36	-0.17	1.00	0.81	-0.89
Costo	-0.58	0.51	-0.14	0.42	-0.54	0.81	1.00	-0.61
Calificación PROFECO	-0.05	-0.49	0.43	-0.40	-0.05	-0.89	-0.61	1.00

Fig. 5. Matriz de correlaciones de Pearson con la clasificación de PROFECO.

Garantía (años)	1.00	0.19	-0.00	0.09	0.58	-0.17	-0.58	0.46
Potencia (W)	0.19	1.00	0.30	0.83	-0.03	0.54	0.51	-0.38
Desempeño de licuado	-0.00	0.30	1.00	0.41	0.25	-0.25	-0.14	0.42
Nivel de ruido	0.09	0.83	0.41	1.00	0.09	0.36	0.42	-0.24
Rupturas	0.58	-0.03	0.25	0.09	1.00	-0.17	-0.54	0.46
Calentamiento	-0.17	0.54	-0.25	0.36	-0.17	1.00	0.81	-0.91
Costo	-0.58	0.51	-0.14	0.42	-0.54	0.81	1.00	-0.92
Calificación cluster GMM	0.46	-0.38	0.42	-0.24	0.46	-0.91	-0.92	1.00

Fig. 6. Matriz de correlaciones de Pearson con la clasificación GMM.

Comparando las matrices de correlación de Pearson, existen similitudes entre ambas, estas son:



Correlación positiva fuerte

- Nivel de ruido - Potencia
- Calentamiento - Costo

Correlación positiva moderada

- Rupturas - Garantía
- Calentamiento - Potencia
- Potencia - Costo

Correlación negativa moderada

- Costo - Garantía
- Costo - Rupturas

Sin embargo, la diferencia entre ambas radica en las correlaciones de las clasificaciones proporcionadas por PROFECO y el modelo propuesto.

Clasificación de PROFECO:

Correlación negativa fuerte

- Calificación PROFECO - Calentamiento

Correlación negativa moderada

- Calificación PROFECO - Costo
- Calificación PROFECO - Potencia

Correlación positiva moderada

- Calificación PROFECO - Desempeño de licuado

Clasificación del modelo propuesto:

Correlación negativa fuerte

- Calificación cluster GMM – Calentamiento
- Calificación cluster GMM – Costo

Correlación positiva moderada

- Calificación cluster GMM – Garantía
- Calificación cluster GMM – Desempeño de licuado

Por lo anterior, es posible determinar que el modelo propuesto otorga una mejor calificación a menor costo y menor calentamiento, así como también resalta aspectos como: garantía y desempeño de licuado.

c. Comparación de resultados

Compilando los resultados mostrados anteriormente se generó una tabla comparativa que resume los hallazgos (ver Tab. 1).

Tab. 1. Resultados comparativos de las clasificaciones.

Nombre	Clasificación PROFECO	Clasificación propuesta
Ninja	Excelente	Excelente
Black + Decker PB2004G	Excelente	Excelente
T-Fal BL1BKDMX	Excelente	Excelente
Nutribullet NB-201S	Muy bien	Bien
Taurus Natural Green (ver II)	Muy bien	Excelente
Koblenz BKM-800	Bien	Muy bien
Koblenz BKM-1000	Bien	Muy bien
Oster BLSTXPN7003-013	Bien	Bien
Oster BLSTXPN7001-014	Bien	Bien
SMEG PBF01PKUS	Bien	Bien

En la Tab. 1. Se muestran los resultados de las clasificaciones correspondientes de la base de datos generada a partir de los estudios de PROFECO, en estas se muestran las diferencias existentes entre su clasificación y la obtenida con el modelo propuesto. Destacando que existen algunas diferencias sustanciales entre algunos productos como: Nutribullet NB-201S, quien en un principio fue clasificado como: muy bien y



después fue como: bien. Por el contrario, existen dispositivos que obtuvieron una mejor clasificación como: Taurus Natural Green (ver II), que pasó de: muy bien, a: excelente. Y los productos de Koblenz BKM de la serie 800 y 1000, que pasaron de: bien a: muy bien. Estos cambios se pueden explicar principalmente por las características correlacionadas entre la clasificación GMM y la de los productos. Concluyendo así, que existe una clasificación equilibrada e insesgada.

VI. CONCLUSIONES

En este trabajo se muestra una metodología fundamentada en herramientas de inteligencia artificial para el análisis, comparación y clasificación de procesadores personales de alimentos. Los resultados de este análisis benefician a los consumidores y también contribuyen al avance del conocimiento en el campo de la mecatrónica.

La objetividad y precisión de la metodología propuesta en este estudio ofrece la posibilidad de comparar los resultados obtenidos con las clasificaciones previas establecidas por estudios como el de PROFECO; esta comparación no solo valida las nuevas formas de clasificación insesgadas, sino que también proporciona ideas valiosas para la mejora del diseño, fabricación y evaluación de estos dispositivos, fomentando un ciclo continuo de innovación en la industria mecatrónica.

Finalmente, los datos y análisis presentados en este estudio tienen un valor educativo significativo ya que pueden servir como un caso de estudio en programas de ingeniería, ilustrando la aplicación práctica de técnicas de análisis estadístico y algoritmos de inteligencia artificial en la evaluación de dispositivos mecatrónicos, enriqueciendo a la formación de los estudiantes destacando la relevancia de la mecatrónica en el desarrollo de tecnologías domésticas avanzadas.

REFERENCIAS

- Gobierno de México. (2024). *Procuraduría Federal del Consumidor | Gobierno | gob.mx*.
<https://www.gob.mx/profeco/que-hacemos>
- PROFECO. (2024). Estudio comparativo de procesadores personales de alimentos. *Revista Del Consumidor*, 567.
https://www.profeco.gob.mx/revista/RevistaDeConsumidor_567_MAYO_2024.pdf
- Urueta, D. E., Lara, P., Gutiérrez, M. Á., de-los-Cobos, S. G., Rincón, E. A., & Mora, R. A. (2021). A Comparative Ranking Model among Mexican Universities Using Pattern Recognition. *Mathematics*, 9(14), 1615.
<https://doi.org/10.3390/math9141615>
- Urueta Hinojosa, D. E., Lavín Delgado, J. E., & Mota Cruz, J. E. (2024). Análisis de Nutrición y Clasificación de Alimentos Secos para Gatos Utilizando Técnicas de Reconocimiento de Patrones. *XIKUA Boletín Científico de La Escuela Superior de Tlahuelilpan*, 12(Especial), 20–24.
<https://doi.org/10.29057/xikua.v12iEspecial.12711>



Estación modular mecatrónica – Eje lineal y Pick and place giratorio

Aguirre Cerrillo Fabio Abel^{*1}, Luna Castro Arturo¹, López Salas Fernando² y Chávez Herrera Daniel Alejandro³

¹ Tecnológico Nacional de México Campus Lázaro Cárdenas, Departamento de metal-mecánica, Av. Melchor Ocampo No. 2555 Cuarto Sector de Fideicomiso CP 60950 Lázaro Cárdenas Michoacán

² Tecnológico Nacional de México Campus Culiacán, Departamento de metal-mecánica, Juan de Dios Batiz No. 310 poniente col. Guadalupe CP 80220 Culiacán Sinaloa

³ Tecnológico Nacional de México Campus Lázaro Cárdenas, Ingeniería Electromecánica, Av. Melchor Ocampo No. 2555 Cuarto Sector de Fideicomiso CP 60950 Lázaro Cárdenas Michoacán

^{*} Autor de correspondencia: fabio.aguirre@lcardenas.tecnm.mx

Resumen. En este trabajo se presenta el diseño y construcción de un sistema mecatrónico con fines didácticos para ser utilizado para la enseñanza-aprendizaje en áreas de la electromecánica y la mecatrónica. El sistema se compone de dos módulos: Eje lineal y Pick and place giratorio. La tarea del módulo del Eje lineal es transportar una pieza de un extremo a otro. Este módulo tiene un sistema de polea y banda dentada para transmitir el movimiento y su actuador es un motor a pasos. El módulo Pick and place giratorio es un sistema con actuadores neumáticos que toma la pieza y la transporta a otro lugar al realizar un movimiento giratorio. Cada módulo tiene su propio microcontrolador para el control de la secuencia establecida. Este trabajo muestra el diseño mecánico, los componentes eléctricos y/o electrónicos, los diagramas neumáticos y el flujo de las señales para un funcionamiento correcto del sistema.

Palabras clave. *Mecatrónica, estación modular, Eje lineal, Pick and place.*

I. INTRODUCCIÓN

El tiempo en que se vive es una época en donde la ciencia y la tecnología tienen un ritmo acelerado de crecimiento en beneficio del desarrollo de las aplicaciones industriales y la vida del ser humano en general. La cuarta revolución industrial está presente por medio de la llamada “industria 4.0” (Joyanes, 2017). En la actualidad los sistemas electromecánicos y/o mecatrónicos contribuyen para la fabricación de una gran cantidad de dispositivos para la industria: alimenticia, electrónica, automotriz, aeroespacial, entre otras. Uno de los elementos base de la industria 4.0 es la colaboración entre las máquinas y el ser humano. Al fin y al cabo, son las personas quienes utilizan e impulsan las tecnologías. Se requiere entonces de personal

cualificado que afronte los retos que la industria demanda. El conjunto de personal cualificado incluye: obreros, técnicos especialistas, ingenieros, entrenadores y desde luego, los profesores que trabajan en centro educativos de niveles medio superior y superior en el mundo. El personal docente son elementos indispensables en la formación de estudiantes en el proceso enseñanza-aprendizaje con la finalidad de cerrar la brecha existente entre el aprendizaje práctico en el puesto de trabajo y el aprendizaje teórico en el centro de formación, las universidades y los institutos tecnológicos (Festo, 2024). La preparación y la educación son elementos clave para atender el presente y configurar los cambios futuros de e impulsar la innovación. Se requiere que los estudiantes tengan habilidades técnicas y



vivencias en los laboratorios, esto es, hacer algo que pase a ser una experiencia vital, que sean acontecimientos y situaciones que marquen a un estudiante. Para que los estudiantes tengan vivencias en los laboratorios se requiere de equipo didáctico con enfoque pedagógico. Existen empresas en el mundo que venden y proporcionan soluciones en equipamiento de estaciones mecatrónicas (Festo Didactic SE, 2021) y (SMC International Training, 2024). El costo en dinero de dichas estaciones mecatrónicas impide (en la mayoría de las veces) que las instituciones educativas adquieran los equipos necesarios y por tanto, que los laboratorios se encuentren debidamente equipados. Lo anterior es lamentable pero también representa una oportunidad para trabajar en el autoequipamiento.

El presente trabajo es un esfuerzo laboral y académico que tiene el objetivo de construir una estación mecatrónica por módulos que pueda ser utilizada como herramienta para la enseñanza-aprendizaje de la mecatrónica para los estudiantes de la carrera de Ingeniería Electromecánica del Instituto Tecnológico de Lázaro Cárdenas.

II. DESCRIPCIÓN DE LA ESTACIÓN

La estación mecatrónica se muestra en la Fig. 1. Se compone de dos módulos: módulo Eje lineal y módulo Pick and Place giratorio. Además se puede ver un cubo llamado Producto que tiene dimensiones de 50x50 mm en sus lados y una altura de 40 mm. El material del Producto es de PLA (ácido poliláctico) y se ha fabricado utilizando tecnología de impresión 3D.

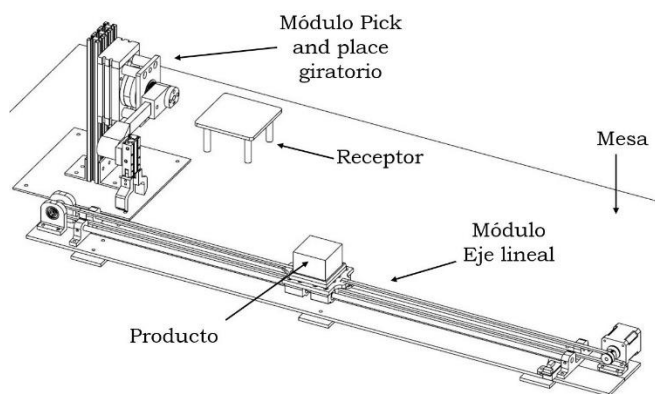


Fig. 1. Elementos de la estación mecatrónica.

En al Fig. 2 se pueden ver los componentes principales del módulo Eje lineal. La función de módulo es llevar el

Producto de un extremo a otro al colocarlo en un carrito con 700 mm de carrera útil. El carrito tiene cuatro rodamientos lineales (Sc8uu) montados sobre dos varillas lisas de 8 mm. El actuador es un motor a pasos NEMA 17 de modelo 17HS8401. Se utiliza un sistema de banda y polea dentada para convertir el movimiento circular en movimiento lineal en el carrito. La banda dentada es del tipo Gt2 de 6 mm de ancho.

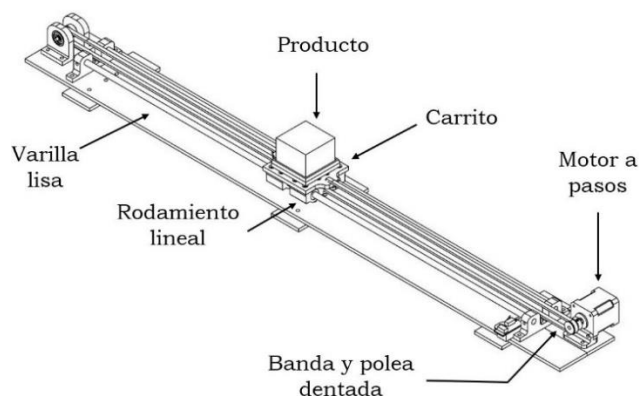


Fig. 2. Módulo Eje lineal.

El eje lineal cuenta con un sensor mecánico de final de correa tal y como lo muestra la Fig. 3. El carrito en su movimiento de izquierda a derecha acciona el sensor mecánico y se origina una señal eléctrica. Con la señal eléctrica se hace trabajar al motor a pasos y colocar al carrito en una posición de inicio.

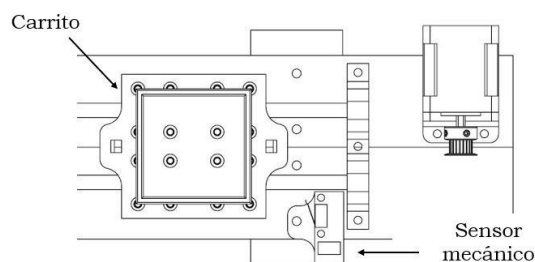


Fig. 3. Montaje del sensor mecánico.

En al Fig. 4 se pueden ver los componentes principales del módulo Pick and place giratorio. Sobre una placa metálica 3/16 pulgadas de espesor se tienen montados tres perfiles de aluminio de 20x20 mm. Sobre el perfil de aluminio se encuentra un actuador neumático giratorio que produce un giro de 180 grados. Dicho actuador transmite su movimiento a un brazo por medio de un eje mecánico.

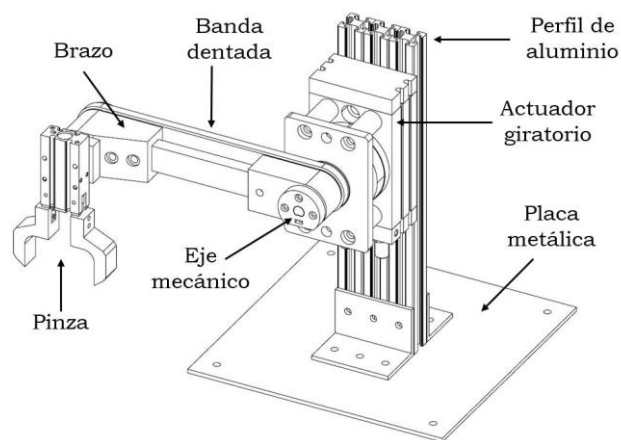


Fig. 4. Módulo Pick and place giratorio.

En el brazo se tiene una pinza neumática del tipo radial. La pinza es el elemento que se encarga de tomar y dejar el producto de la estación. La pinza debe de mantener siempre la misma posición con respecto a la horizontal en el momento que gira el brazo. Para lograr lo anterior observar la Fig. 4 y la Fig. 5. En una misma pieza (impresa en 3D) se tiene una placa con una polea dentada y que es fija. Entre dicha placa y el eje mecánico se tiene un rodamiento que permite el giro. Se tiene además una banda dentada del tipo Gt2 de 6 mm de ancho que transmite el movimiento a la polea dentada móvil. Al girar el eje mecánico se produce un giro en el brazo y al mismo tiempo gira la polea dentada móvil. Lo anterior ocasiona que la pinza realice un giro en la misma proporción que el eje mecánico. La polea de la placa y la polea dentada móvil tienen 50 dientes y son del mismo tipo que la banda dentada para su compatibilidad.

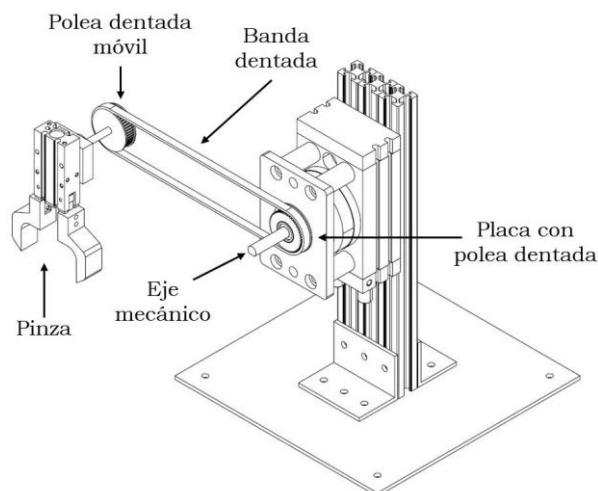


Fig. 5. Transmisión del movimiento a la pinza.

En la Fig. 6 se muestra el circuito neumático del módulo pick and place. Tanto el actuador giratorio como la pinza se controlan por medio de una válvula 5/2 vías de accionamiento manual y eléctrico (1V1 y 2V1). El voltaje de operación de las bobinas es de 24 VCD. Se tiene la posibilidad de regular la velocidad de los actuadores por medio de las válvulas de estrangulación y antirretorno (1V2, 1V3, 2V2 y 2V3). Para los dos actuadores se tienen montados sensores de final de carrera llamados contactos Reed (Ebel, Idler, Prede, y Scholz, 2010). En la Tab. 1 se enlistan los elementos del circuito neumático (SMC España, 2024).

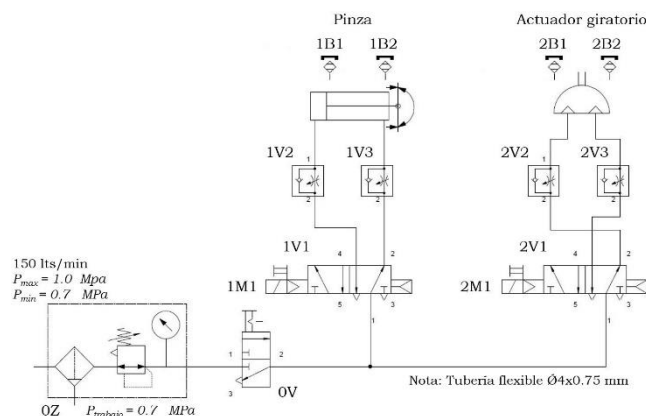


Fig. 6. Circuito neumático módulo Pick and place.

Tab. 1. Elementos neumáticos módulo Pick and Place .

Elemento	Marca	Modelo
Filtro regulador	OZ	AC10A-M5G-A
Válvula manual	0V	VHK3A-04F-04F
Electroválvula	1V1, 2V1	SY3120-5DZ-M5
Válvula reguladora	1V2, 1V3, 2V2, 2V3	SY3120-5DZ-M5
Pinza	-----	MHY2-10D
Actuador giratorio	-----	MSQB-20 ^a
Sensor Reed	1B1, 1B2, 2B1, 2B2	D-A93

III. CONTROL DE LA ESTACIÓN.

Ya se ha establecido que el módulo Eje lineal es accionado por un motor a pasos NEMA 17. Para lograr lo anterior se utiliza el controlador de motor a pasos TB6600 y el diagrama de conexiones eléctricas se muestra en la Fig. 7. Las características generales son:

- Voltaje de operación 9-42 VCD.
- Control de ajuste de corriente.
- Control de ajuste de paso 1, 1/2, 1/4, 1/8, 1/16 y 1/32.
- Control de dirección.

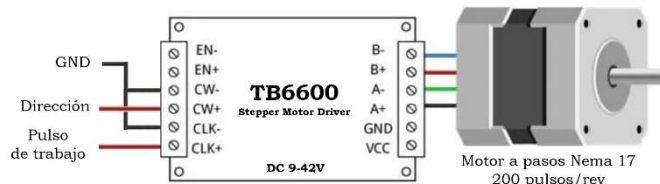
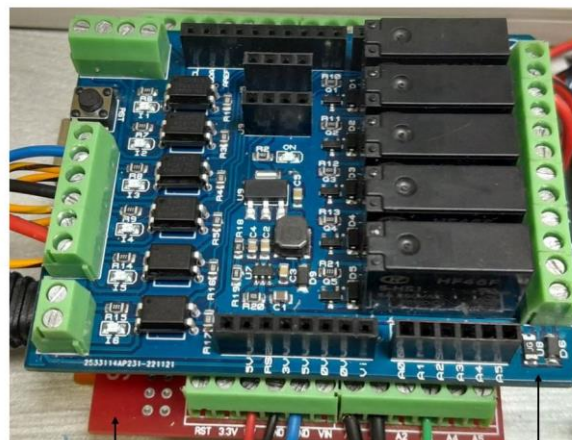


Fig. 7. Controlador del motor a pasos.

El controlador TB6600 es compatible con sistemas TTL 5V y se requiere de dos pines para el control. Si el pin de *Dirección* se encuentra en *estado lógico 0* el motor a pasos puede girar en sentido de las manecillas del reloj. Si su estado es *lógico 1* entonces puede girar en dirección contraria. El otro pin de control es el *pulso de trabajo*. Si se aplican 400 pulsos y el controlador TB6600 se encuentra configurada con un ajuste de paso de $\frac{1}{4}$ para un motor de 200 pulsos/rev, el motor a pasos gira una revolución y el eje lineal se desplaza 40 mm. Por tanto se tiene que $40 \text{ mm}/400 \text{ pulsos}$ da como resultado una resolución de 0.1 mm/pulso .

Para el control del controlador a pasos TB6600 se utiliza la placa de desarrollo Arduino Uno. Es una placa basada en el microcontrolador ATmega328P (Arduino, 2024). Contiene pines que se pueden configurar como entradas y salidas en niveles de voltaje de 5 VCD (Reyes y Cid, 2015). En dicha placa se realiza la programación del módulo Eje lineal tomando en cuenta las entradas y salidas del sistema y la comunicación con la placa que controla el módulo Pick and place. Para realizar la conexión física de las señales TTL en niveles de 5 VCD se utiliza un shield clemas. El shield clemas facilita la conexión eléctrica ya que permite atornillar los cables a una cema para garantizar conexiones seguras a los pines de entrada y salida. Se monta directamente sobre el Arduino Uno. Se tienen también señales a 24 VCD y se utiliza una tarjeta que se llama comercialmente PLC Control Shield Arduino tal y como lo muestra la Fig. 8 (JELECTRONICA, 2024). Esta tarjeta contiene:

- Hasta 6 entradas optoacopladas de 5 a 24 VCD. Conectadas a la pines D7, D8, D9, D10, D11 y D12.
- Hasta 5 salidas a relevador 5A 250 VCA, 5A 30 VCD. Conectadas a los pines D2, D3, D4, D5 y D6.
- Entrada de alimentación de 8 a 35Vdc (recomendado 8 a 24 VCD).
- Adicional, viene disponible el acceso a todas las demás líneas de entrada/salida del Arduino Uno.



Shield clemas

PLC Control Shield
Arduino

Fig. 8. Tarjeta Control Shield Arduino.

En la Fig. 9 se muestra conceptualmente la conexión de la tarjeta de control del módulo Eje lineal y la relación de las entradas y salidas.

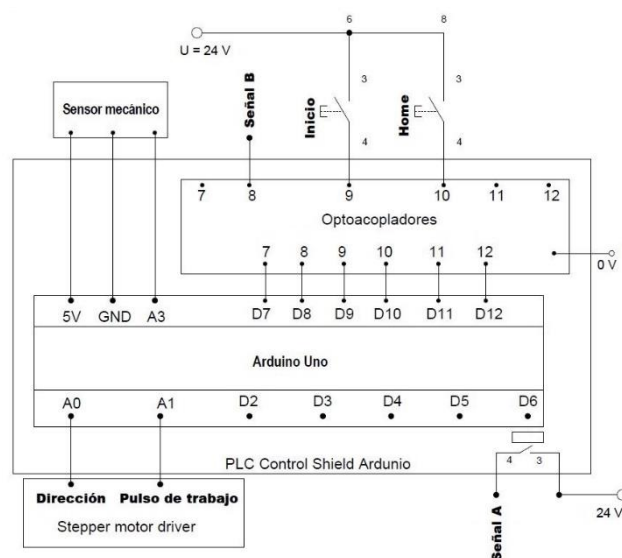


Fig. 9. Conexión de la Tarjeta de control módulo Eje lineal.

- Se tienen tres señales de entrada a niveles de 24 VCD: Entrada B, Inicio y Home conectadas a los pines 8, 9 y 10, respectivamente).
- La señal B es una señal que viene de la tarjeta controladora del módulo Pick and place giratorio y que se explicará su utilidad mas adelante.
- La señal del botón pulsador Inicio es para empezar el proceso de trabajo de la estación mecatrónica.



- La señal del botón pulsador Home es para ejecutar un programa con la finalidad de que el carrito se traslade a una posición de inicio.
- El pin A3 es una entrada que por default es una señal analógica. Se configura por software con el IDE de Arduino como señal de entrada digital (Arduino, 2024). Es una señal que viene del sensor mecánico de acuerdo con lo explicado anteriormente en la Fig. 3.
- Al igual que el pin A3, los pines A0 y A1 son pines de señal analógica que configuran por software como señales de salida digital para controlar el motor a pasos. Sus niveles son de 5 VCD.
- La señal A es una señal de comunicación que se conecta como entrada a tarjeta controladora del módulo Pick and place giratorio.

Para el control del módulo Pick and place se utiliza una tarjeta que se llama comercialmente PLC Esp32 30 Pin 9 Entradas 10 Salidas tal y como lo muestra la Fig. 10 (JELECTRONICA, 2024). Utiliza la tarjeta de desarrollo NodeMCU ESP32 que se monta directamente sobre la tarjeta PLC Esp32 (Punto flotante s.a., 2022) y se puede programar con el IDE de Arduino (Arduino, 2024).

- Hasta 9 entradas optoacopladas de 12 a 24 VCD, conectadas a la pines D17, D18, D19, D21, D22, D23, D27, D32 y D33.
- Hasta 10 salidas a relevador 5A 250 VCA , 5A 30 VCD, conectadas a los pines D1, D2, D3, D4, D5, D12, D13, D14, D15 y D16.

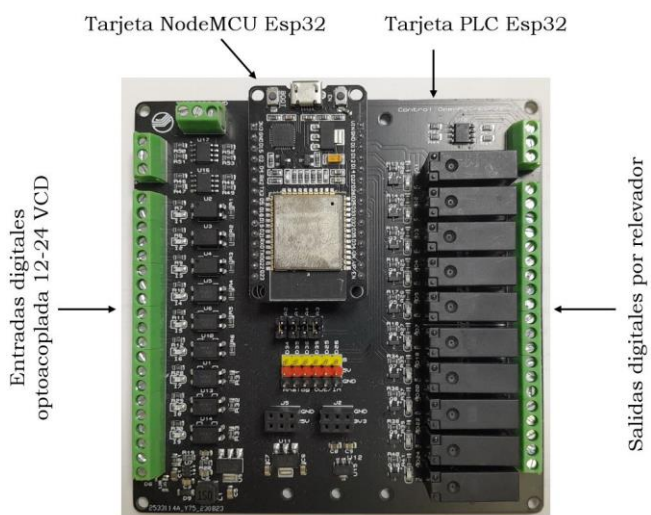


Fig. 10. Tarjeta PLC Esp32.

En la Fig. 11 se muestra conceptualmente la conexión de la tarjeta de control del módulo Pick and place y la relación de las entradas y salidas.

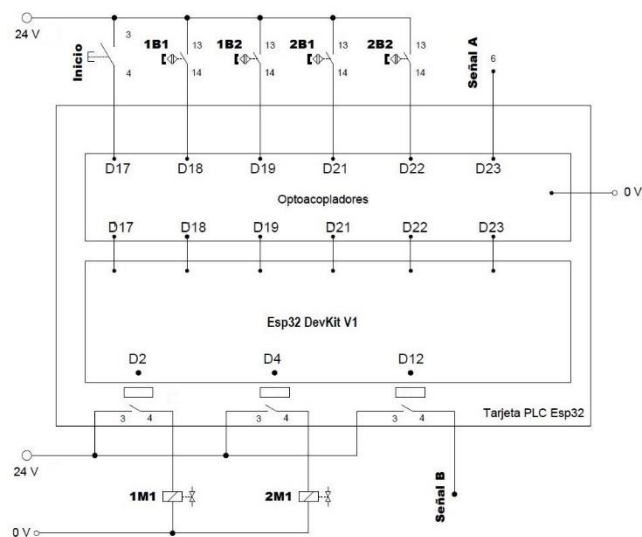


Fig. 11. Conexión de la Tarjeta de control módulo Pick and place.

- Se tienen seis señales de entrada a niveles de 24 VCD: Inicio, 1B1, 1B2, 2B1, 2B2 y señal A conectadas a los pines D17, D18, D19, D21, D22 Y D23 respectivamente).
- Se tienen tres señales de salida por relevador con conexión eléctrica de 24 VCD: 1M1, 2M1 y Señal B conectadas a los pines D2, D4 y D12.
- Las señales 1M1 y 2M2 son señales que activan las bobinas de las electroválvulas del circuito neumático mostrado con anterioridad en la Fig. 6.
- La señal B es una señal de comunicación que se conecta como entrada a tarjeta controladora del módulo Eje lineal.

IV. SECUENCIA DE TRABAJO Y RESULTADOS

En la Fig. 12 se muestra una fotografía de la construcción física de la estación mecatrónica. Sobre la fotografía se han escrito el nombre de algunos de los elementos principales.

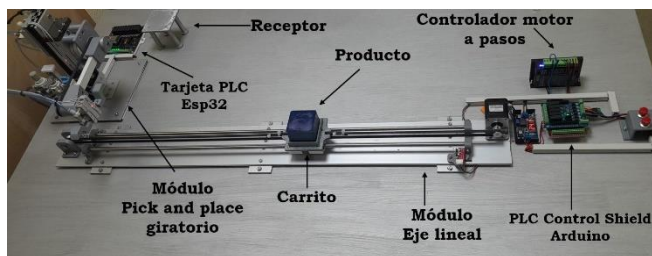


Fig. 12. Fotografía de la estación mecatrónica.



En la Fig. 13 se muestra el flujo de señales para la operación de la estación mecatrónica. La programación de las tarjetas que controla cada módulo se ha hecho utilizando software del IDE de Arduino (Arduino, 2024).

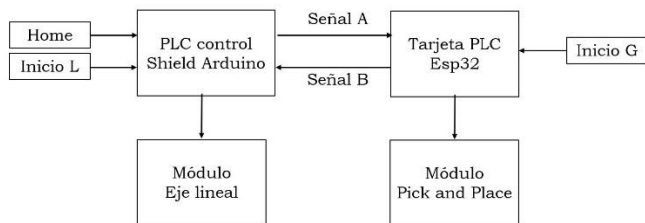


Fig. 13. Flujo de señales.

Al oprimir el botón de Home se ejecuta un proceso para llevar al carrito a la posición de inicio. El proceso consiste en mover el carrito en la dirección del sensor mecánico (ver Fig 3). Una vez activado el sensor mecánico se desplaza al carrito en una dirección opuesta una distancia de 20 mm y que se le llama posición Inicial del carrito. Al oprimir el botón de Inicio L, el carrito que lleva al producto se desplaza 683 mm en dirección del módulo Pick and Place y que se llama posición final del carrito. El módulo PLC Control Shied Arduino activa la señal A que la recibe la tarjeta PLC Esp32. Con la señal A se ejecuta el proceso Tomar Pieza siempre y cuando se tengan activados las señales de los sensores 1B1 y 2B2:

- Se energiza la bobina de la electroválvula 1M1. La pinza se cierra tomando al Producto y se activa el sensor 1B2.
- Con la señal 1B2 se energiza la bobina de la electroválvula 2M1 y el actuador giratorio realiza un movimiento de 180 grados llevándose al Producto y se activa el sensor 2B1.
- Con la señal de 2B1 se desactiva la bobina de la electroválvula 1M1. La pinza se abre dejando al Producto en el Receptor y se activa el sensor 1B1.
- Con la señal 1B1 se produce una señal B que hace llega al PLC Control Shied Arduino para que ejecute las órdenes y hacer que el carrito se desplace a la posición inicial del carrito. También con la señal 1B1 se desactiva la bobina de la electroválvula 2M1 y hace que actuador regrese 180 grados y se activa el sensor 2B2 para estar en condicione de iniciar otro ciclo.

También se puede ver que se tiene un botón de Inicio G. Al oprimirse se ejecuta el proceso Tomar Pieza sin

necesidad de la señal A siempre y cuando se tengan activados las señales de los sensores 1B1 y 2B2. De las figuras Fig. 14 a la Fig. 19 se muestran una secuencia de fotografías del proceso al oprimirse el botón Inicio L. En la Fig. 19 se puede ver que el Producto ya se encuentra en el Receptor y no se visualiza el carrito. Significa que el carrito ha regresado a la posición inicial.

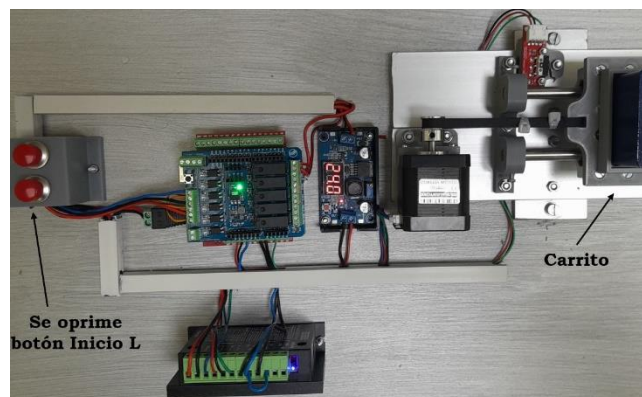


Fig. 14. Posición de inicio.

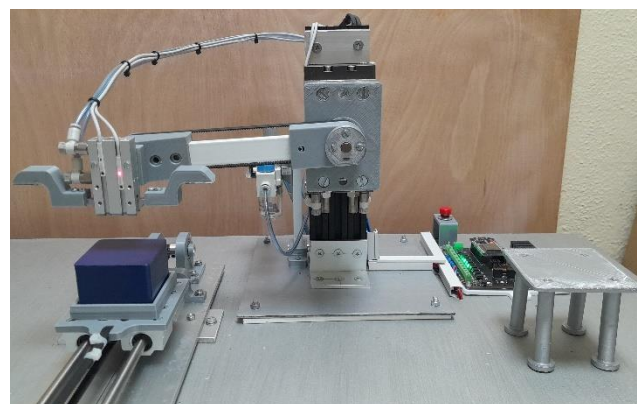


Fig. 15. Posición final del carrito y pinza abierta.

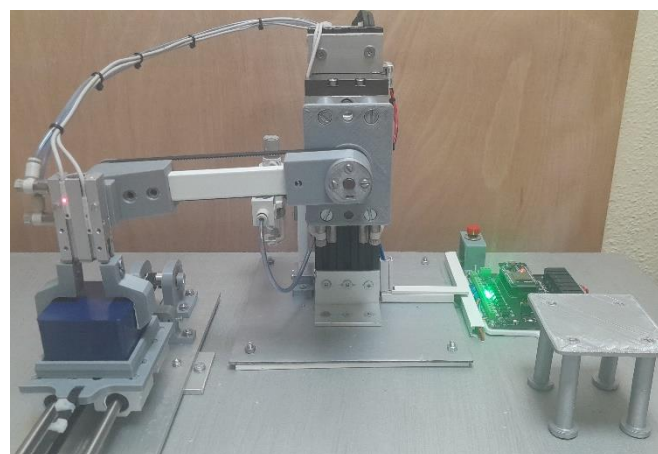


Fig. 16. Posición final del carrito y pinza cerrada.

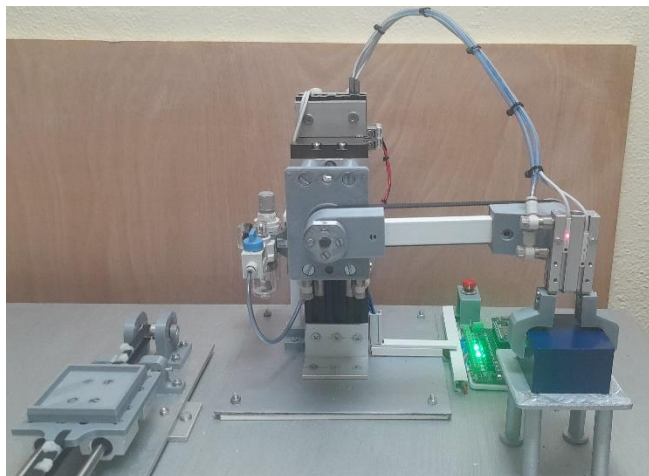


Fig. 17. Producto en Receptor y pinza cerrada.

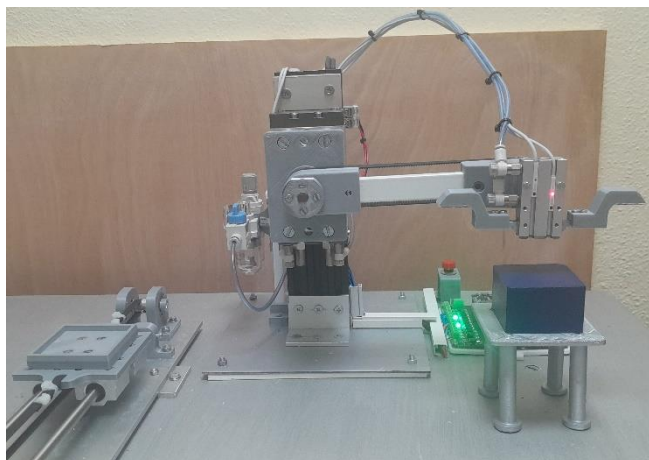


Fig. 18. Producto en Receptor y pinza abierta.

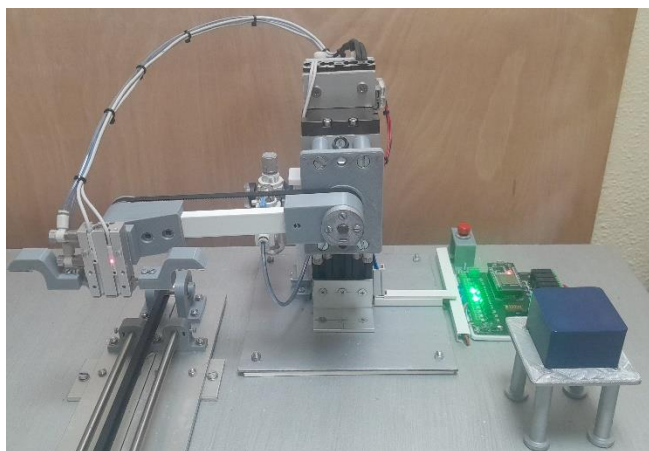


Fig. 19. Producto en Receptor.

V. CONCLUSIONES

Cada vez existen más elementos mecánicos que requieren de la integración de la electrónica y la programación de dispositivos de control. Los avances de

la mecatrónica exigen que se cuente con personal cualificado para proporcionar el servicio a los sistemas mecatrónicos de la industria. Los sistemas educativos de nivel medio superior y superior del país deben de contar con profesores preparados y laboratorios adecuados y equipados para que los estudiantes tengan vivencias de aprendizaje con la finalidad de que su formación profesional afronte los retos que la sociedad demanda. Se ha presentado el diseño y la construcción de una estación mecatrónica que puede servir como apoyo de enseñanza-aprendizaje de la mecatrónica. La estación es modular e integra diferentes tecnologías para realizar un proceso de una aplicación industrial. Como trabajos futuros se recomienda realizar un análisis con la finalidad de agregar otros módulos que puedan trabajar en conjunto con la estación mecatrónica mostrada.

REFERENCIAS

- Arduino, (2024). *Pagina oficial de Arduino*. Obtenida el Febrero de 2024, de <https://www.arduino.cc>.
- Ebel, F., Idler, S., Prede, G. y Scholz, D. (2010). *Neumática Electroneumática Fundamentos*, Festo Didactic GmbH & Co. KG, 73770 Denkendorf, Alemania.
- Festo, (2017). *Festo Didactic Leadership in Technical Education*, Folleto recuperado en Agosto de la web https://www.festo.com/mx/es/e/educacion-id_3776/.
- Festo Didactic SE, (2017). *Automatización industrial – Sistemas de aprendizaje para la formación técnica*, Catalogo, Esslingen Alemania.
- JELETRONICA, (2017). *Recuperado en Enero 2024 de https://www.mercadolibre.com.mx/pagina/jeleletronica8670*.
- Joyanes, Luis Aguilar (2017). *La Industria 4.0 La cuarta revolución industrial*, Primera Edición Alfaomega, México.
- Punto flotante s.a., (2022). *Manual básico de pruebas del Node MCU versión ESP32 devKit V1. Arduino IDE*. Recuperado de <https://www.puntoflotante.net>.
- Reyes, F. y Cid, J. (2015). *Arduino Aplicaciones en Robótica Mecatrónica e Ingenierías*, Alfaomega Grupo Editor, Primera edición, México.
- SMC International Training, (2024). *Catalogo digital SMC International Training*, Recuperado Junio de la web <https://www.smctraining.com/es>.



Representación del control de velocidad de un vehículo autónomo por un sistema Bola Balancín

Rafael Dassaev Flores López

*Universidad de Guanajuato, Ingeniería Mecatrónica, Carretera Salamanca - Valle de Santiago km 3.5 + 1.8
Comunidad de Palo Blanco, 36787 Salamanca, Gto.
rd.floreslopez@ugto.mx*

Resumen. En este trabajo se ha realizado el diseño y fabricación de las piezas necesarias. Utilizamos una fresadora para crear piezas de Nylamid, como la base del servomotor, la base para la balanza, las bisagras y los topes. También fabricamos un riel de aluminio y un eje para transmitir la fuerza del servomotor a la balanza. Luego, nos enfocamos en el código, Desarrollamos un programa en Arduino IDE para recibir la señal del sensor Sharp 2Y0A21. El objetivo era controlar el servomotor para mantener una pelota de ping pong en la zona media de la balanza de aluminio. Finalmente, ensamblamos todo, Unimos todas las piezas de la planta, incluyendo el servomotor Futaba S3003 en su base para ajustar el ángulo de la balanza y colocamos el sensor Sharp en un extremo de la balanza. Conectamos el sensor y el servomotor a un Arduino Uno previamente programado con nuestro código, para luego realizar las pruebas y ajustes hasta lograr que el dispositivo mantuviera la pelota en la zona media de la balanza. Y todo esto para de una forma muy simple lograr visualizar el control de algo más grande como lo es el control de velocidad de un vehículo autónomo.

Palabras clave. *Estabilidad, Autónomo, Representación visual, Retroalimentación.*

I. INTRODUCCIÓN

En el contexto de la conducción autónoma, el control preciso de la velocidad es fundamental para garantizar la seguridad y la eficiencia en las carreteras. Nuestro proyecto se centra en diseñar un sistema de bola balancín que simula este control de velocidad en un entorno controlado.

II. OBJETIVO

Imagina un vehículo autónomo en una autopista. Su objetivo es mantener una velocidad constante y segura mientras se adapta a las condiciones cambiantes del tráfico. De manera similar, nuestro sistema de bola balancín busca mantener una pelota en equilibrio en la zona media de una balanza de aluminio. La posición de la pelota representa la velocidad deseada del vehículo.

Para lograr esto, nuestro objetivo en esta fase es fabricar las piezas que se necesitan para nuestra planta y ensamblarlas cuidadosamente. También realizaremos un

código para el Arduino que nos permita controlar el servomotor que ajustará el ángulo de la balanza según la señal del sensor Sharp. Si la pelota se desplaza hacia un extremo, el servomotor actúa para corregir el desequilibrio, análogo al control de velocidad en un automóvil.

Este proyecto no solo es una demostración ingeniosa, sino también una metáfora visual de cómo los sistemas autónomos mantienen una velocidad constante y segura en la carretera.

III. JUSTIFICACIÓN

En el campo de la conducción autónoma, la precisión en el control de velocidad es esencial para garantizar la seguridad de los pasajeros y otros usuarios de la vía. Los vehículos autónomos deben adaptarse a las condiciones cambiantes del tráfico y mantener velocidades óptimas para evitar colisiones y optimizar la eficiencia del viaje.



Nuestro sistema de bola balancín no solo es una demostración ingeniosa, sino también una metáfora visual de cómo los sistemas autónomos gestionan la velocidad. Al mantener una pelota en equilibrio en la zona media de la balanza de aluminio, representamos simbólicamente el control de velocidad en un vehículo autónomo. Cada pequeño ajuste del servomotor para mantener la pelota en su posición refleja las decisiones que un vehículo autónomo toma para mantener una velocidad constante y segura.

El sistema de bola balancín proporciona una representación tangible y fácil de entender del control de velocidad. Los observadores pueden ver directamente cómo los cambios en la posición de la pelota se correlacionan con las acciones del servomotor.

Este proyecto es ideal para entornos educativos. Los estudiantes pueden aprender sobre conceptos de control, sensores y actuadores mientras interactúan con un sistema físico.

Aunque es una simplificación, el sistema de bola balancín sienta las bases para comprender sistemas más avanzados utilizados en vehículos autónomos, como los algoritmos de control PID y la retroalimentación de sensores.

En resumen, nuestro proyecto no solo es un ejercicio técnico, sino también una forma creativa de comunicar la importancia del control de velocidad en la conducción autónoma. Esperamos que inspire a otros a explorar soluciones innovadoras en este campo en constante evolución.

IV. ANTECEDENTES

Prototipo de Vehículo Eléctrico Experimental con Control de Velocidad (Actualidad): Se diseñó e implementó un sistema de control de velocidad para un vehículo eléctrico experimental. El vehículo contaba con un motor de corriente continua de 48V, una controladora electrónica y un sensor de efecto Hall. El objetivo era mantener una velocidad constante y segura.

Este prototipo proporciona valiosas lecciones que pueden influir en nuestro proyecto de sistema de bola balancín pues el prototipo se centró en mantener una velocidad constante y segura. Esto es relevante para nuestro sistema de bola balancín, ya que también buscamos mantener una posición estable (representada

por la pelota) mediante ajustes precisos. El uso del sensor de efecto Hall en el prototipo nos recuerda la importancia de la retroalimentación. En nuestro sistema, el sensor Sharp cumple un papel similar al proporcionar información sobre la posición de la pelota. La controladora electrónica en el prototipo es fundamental para ajustar la velocidad. En nuestro caso, el Arduino actúa como nuestro “controlador” para mantener la pelota en equilibrio. Al igual que en un vehículo autónomo, nuestro sistema debe adaptarse a cambios en la posición de la pelota. Al observar cómo el prototipo manejó las variaciones de velocidad puede inspirarnos en nuestro enfoque. En resumen, este prototipo nos ofrece ideas valiosas sobre control, sensores y adaptabilidad que podemos aplicar en nuestro proyecto.

En un proyecto desarrollado en la Universidad de Valladolid, se aborda el control teórico y real de un sistema bola-balancín. Este sistema consta de una plataforma con una bola que puede rodar por ella y alcanzar diferentes posiciones entre sus extremos¹².

Este trabajo proporciona una base sólida para comprender los principios de control y modelado aplicados a sistemas similares.

Un estudio realizado en la Universidad de Valladolid desarrolló tanto el control teórico como real de un sistema bola-balancín. Este trabajo proporciona una base sólida para entender cómo se puede controlar un sistema bola-balancín.

Universidad Pontificia Bolivariana aquí, en las aulas y laboratorios, se desarrolló un sistema Ball and Beam para implementar estrategias de control mediante Labview. Este proyecto es relevante porque muestra cómo se puede utilizar un sistema similar para implementar diferentes estrategias de control.

Universidad Loyola Andalucía Aquí, un proyecto se centra en la simulación y control de un Vehículo Autónomo de Superficie (VAS). Aunque este proyecto no utilizó un sistema bola-balancín, es relevante porque se centró en el control de un vehículo autónomo. Este proyecto nos conecta con el mundo real de los vehículos autónomos.



V. DISPOSITIVOS Y METODOLOGÍA

A continuación, describiré las características técnicas de los dispositivos utilizados y la metodología para lograr el funcionamiento deseado en nuestro proyecto del sistema de bola balancín.

Servomotor Futaba S3003:

Características:

- Motor de corriente continua.
- Voltaje de operación: 4.8V a 6V.
- Torque: 3 kg·cm.
- Velocidad de rotación: 0.23 s/60°.

Función en el Proyecto: Ajusta el ángulo de la balanza para mantener la pelota en equilibrio.



Fig. 1. Servomotor Futaba S3003.

Sensor Sharp 2Y0A21:

Características:

- Sensor infrarrojo de distancia.
- Rango de medición: 10 cm a 80 cm.
- Salida analógica proporcional a la distancia.

Función en el Proyecto: Proporciona información sobre la posición vertical de la pelota.



Fig. 2. Sensor Sharp 2Y0A21.

Arduino Uno:

Características:

- Microcontrolador basado en ATmega328P.
- Entradas y salidas digitales y analógicas.
- Programable mediante Arduino IDE.

Función en el Proyecto: Controla el servomotor y procesa la señal del sensor.

Metodología

Diseño y Fabricación de Piezas: Diseñamos las piezas de Nylamid en software CAD. Utilizamos una fresadora para fabricar la base del servomotor, la base para la balanza y otras partes. Construimos el riel de aluminio y el eje para transmitir la fuerza del servomotor. Comenzamos con una de las piezas que nos servirán para transmitir la fuerza del servomotor hacia el riel de aluminio lo que permitirá controlar la inclinación del riel y con esto la posición de la pelota; para ello creamos un disco que va conectado a las aspas con las que viene el servomotor Futaba S3003 y este disco a su vez irá conectado a un eje que junto con unas bisagras se conectarán al riel de aluminio.

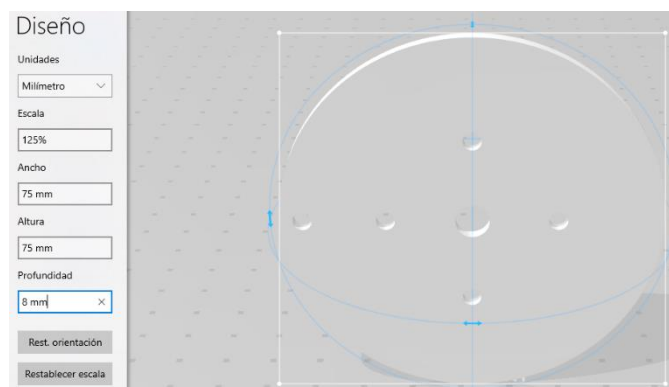


Fig. 3. Diseño disco transmisor de fuerza.



Para el disco, así como para la mayoría de las piezas que se utilizarán se emplea Naylamid, un material plástico de buena dureza y que nos permite que sea trabajado en torno y fresadora. En este caso fue utilizado el torno para el desbaste hasta lograr las medidas deseadas para el disco.



Fig.4. Disco transmisor de fuerza del servo.

Para el balance del riel se conecta a unas bisagras tanto en la base como en el eje que transmite la fuerza del servo, para esto se diseñaron bisagras que cumplieran un tamaño de acuerdo con ambas.

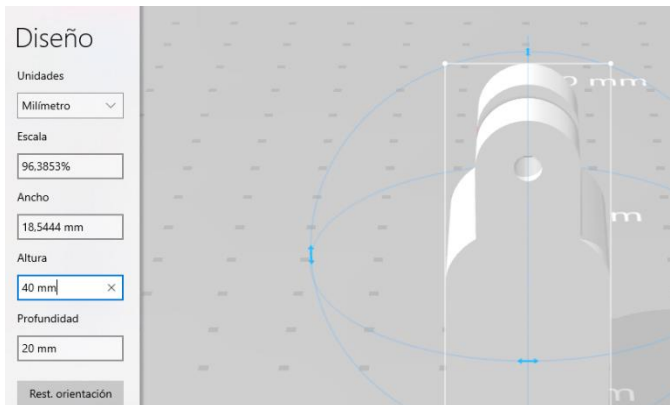


Fig. 5. Diseño bisagra.

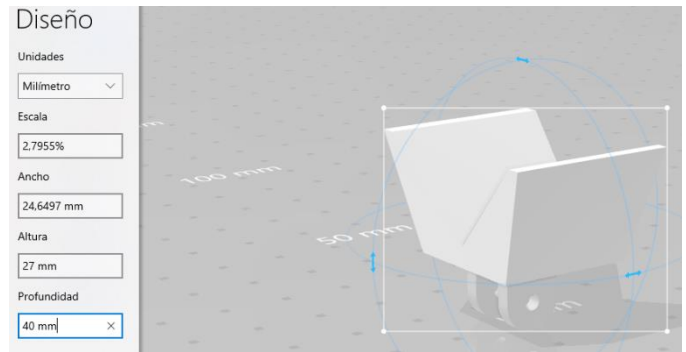


Fig. 6. Diseño bisagra.

Se prosigue trabajando con Naylamid para estas piezas, las cuales son procesadas en fresadora para darles las dimensiones a cada una de las piezas que conforman tanto las bisagras de la base como las del eje del motor.



Fig. 7. Bisagra eje del servo.

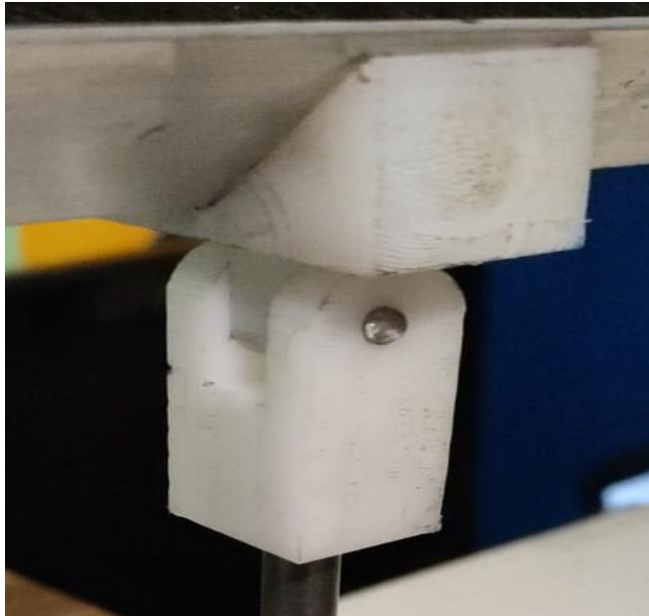


Fig. 8. Bisagra del eje base.

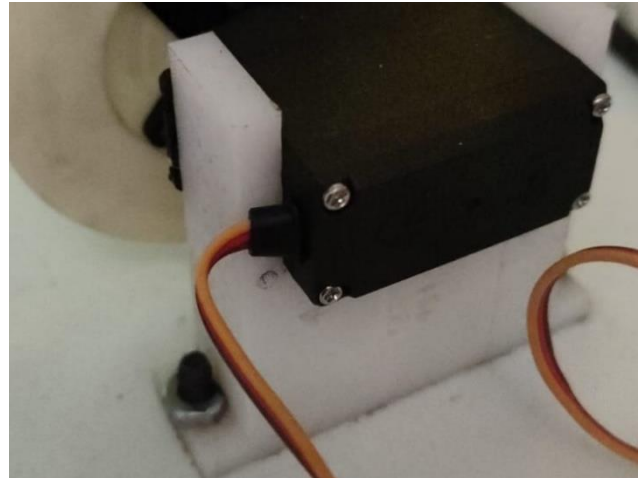


Fig. 10. Soporte del servo.

A continuación, realizamos el diseño de la base para nuestro servomotor, considerando las dimensiones de este, así como también teniendo muy en cuenta que el propósito de esta base es darle al servo una posición fija y que permita que se pueda rotar hasta los grados que sean necesarios sin que ninguno de sus componentes choque con la base de madera.

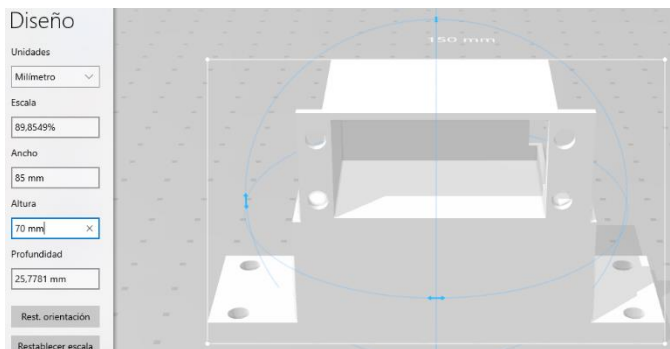


Fig. 9. Diseño soporte para servo.

Se trabaja el material de Naylamid en fresadora, con un grado de tolerancia muy alto pues no es una pieza que requiera de una gran precisión, haciendo en esta misma máquina las perforaciones para ensamblar la pieza en la base de madera, así como para fijar el servo en la posición deseada

Ensamblaje y Pruebas: Unimos todas las piezas para ensamblar la planta. Colocamos el servomotor y el sensor en sus posiciones adecuadas.

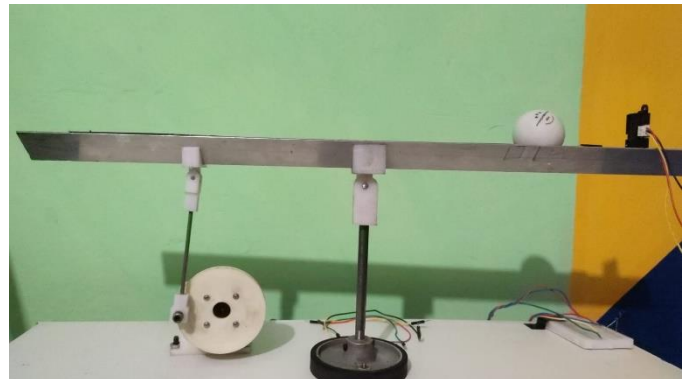


Fig. 11. Planta Bola-Balancín ensamblada.

Programación en Arduino IDE: Desarrollamos un código en Arduino IDE. Configuramos el Arduino Uno para recibir la señal del sensor Sharp y controlar el servomotor. Ajustamos los parámetros para mantener la pelota en la zona media de la balanza.

En primer lugar, en nuestro código hacemos uso de las bibliotecas que nos serán útiles, en este caso se utiliza un servomotor entonces colocamos la biblioteca necesaria para su control, así como la creación del objeto "myservo". También definimos la entrada A0 para la lectura de nuestro sensor analógico y nombramos las demás variables que tendremos en nuestro sistema. Luego definimos las constantes K_i , K_p y



Kd que nos serán útiles, estas fueron definidas a prueba y error de lo cual nos dimos cuenta de que con la constante Kp es suficiente para que nuestro sistema fuera estable; también definimos la distancia media a la que deseamos que se mantenga la pelota.

```

1  #include <Wire.h>
2  #include <Servo.h>
3
4
5
6  //Inputs/outputs//
7  int Analog_in = A0;
8  Servo myservo;
9  //
10
11
12 //Variables//
13 int Read = 0;
14 float distance = 0.0;
15 float elapsedTime, time, timePrev;
16 float distance_previous_error, distance_error;
17 int period = 50;
18 //
19
20 //PID constants//
21
22 float kp=10;
23 float ki=0;
24 float kd=0;
25 float distance_setpoint = 22;
26 float PID_p, PID_i, PID_d, PID_total;
27 //

```

Fig. 12. Código parte 1.

En seguida insertamos la función de configuración que se ejecutara una vez al iniciar el Arduino. Aquí mismo hacemos que se inicie la comunicación serial a 9600 baudios, asignamos el pin 9 al servomotor, movemos el servo a la posición inicial e inicializamos la variable de tiempo.

```

31 void setup() {
32   //analogReference(EXTERNAL);
33   Serial.begin(9600);
34   myservo.attach(9);
35   myservo.write(125);
36   pinMode(Analog_in, INPUT);
37   time = millis();
38 }

```

Fig. 13. Código parte 2.

Ahora configuramos la función principal que se ejecutará repetidamente, en la cual dentro del “if” verifica si ha pasado el periodo establecido para ejecutar el código del PID, actualiza la variable de tiempo, obtiene y calcula la distancia con la función

“get_dist”. Para esto calcula el error de distancia y los componentes del PID.

También limita y mapea el valor total del PID para estar a un rango adecuado para mover el servo. Por último, actualiza el error anterior con el error actual.

```

40 void loop() {
41   if (millis() > time+period)
42   {
43     time = millis();
44     distance = get_dist(100);
45     distance_error = distance_setpoint - distance;
46     PID_p = kp * distance_error;
47     float dist_diference = distance_error - distance_previous_error;
48     PID_d = kd*((distance_error - distance_previous_error)/period);
49
50     if(-3 < distance_error && distance_error < 3)
51     {
52       PID_i = PID_i + (ki * distance_error);
53     }
54     else
55     {
56       PID_i = 0;
57     }
58
59     PID_total = PID_p + PID_i + PID_d;
60     PID_total = map(PID_total, -150, 150, 0, 150);
61
62     if(PID_total < 20){PID_total = 20;}
63     if(PID_total > 160) {PID_total = 160; }
64
65     myservo.write(PID_total+30);
66     distance_previous_error = distance_error;
67   }
68 }

```

Fig. 14. Código parte 3.

Al final colocamos la función que calcula la distancia promediando ‘n’ lecturas del sensor y aplicando una fórmula matemática para convertirlo a centímetros. En este caso se hace uso de una función con la que ya se ha trabajado en otros códigos para el mismo sensor.

```

73 float get_dist(int n)
74 {
75   long sum=0;
76   for(int i=0;i<n;i++)
77   {
78     sum=sum+analogRead(Analog_in);
79   }
80   float adc=sum/n;
81
82   float distance_cm = 17569.7 * pow(adc, -1.2062);
83
84   return(distance_cm);
85 }

```

Fig.15. Código parte 4.

Realizamos pruebas iterativas para verificar el funcionamiento y ajustamos los valores de PID para el sistema según sea necesario. Con esta metodología y los dispositivos mencionados, logramos el funcionamiento deseado del sistema de bola balancín.



VI. RESULTADOS

A continuación, se detallan los resultados obtenidos en nuestro proyecto del sistema de bola balancín:

a. Estabilidad del Sistema.

Logramos mantener la pelota en equilibrio en la zona media de la balanza de aluminio.

El servomotor ajustó continuamente el ángulo de la balanza para contrarrestar cualquier desplazamiento de la pelota.

b. Precisión del Control.

El sistema respondió de manera rápida y precisa a las variaciones en la posición de la pelota.

Ajustamos los parámetros del controlador para minimizar los errores y mantener la pelota en la posición deseada.

c. Representación Visual.

La pelota en equilibrio sobre la balanza proporcionó una representación visual efectiva del control de velocidad en un vehículo autónomo.

Los observadores pudieron comprender intuitivamente cómo los ajustes del servomotor mantenían la pelota en el centro.

d. Aprendizaje y Adaptación:

Durante las pruebas, identificamos desafíos y ajustamos el sistema según fuera necesario.

Este proceso de aprendizaje nos permitió comprender mejor los principios de control y retroalimentación.

El fin de este trabajo era lograr visualizar el control de velocidad de un vehículo autónomo, para ello pensemos en que nuestro auto controlado, por ejemplo, va por una carretera en la que la velocidad permitida es hasta máximo 60 km/h, nuestro vehículo se encuentra entre otros dos autos que respetan y mantienen esta velocidad máxima; entonces nuestro vehículo debe estar a una distancia segura entre estos dos autos y además mantenerla, para ello debe ser controlada su aceleración y desaceleración en caso de rebasar la velocidad.



Fig. 16. Representación del sistema en autopista.

Al lograr llegar a la velocidad deseada, nuestro vehículo deberá mantenerla estable y de esta manera también mantendrá la distancia entre los dos autos; esto claro pensando en que todo es ideal.

En nuestro sistema esto se ve reflejado de la siguiente manera. Primero tenemos representados los elementos del sistema real con la pelota haciendo de nuestro vehículo autónomo, pues esta se moverá de acuerdo con la inclinación del riel, la cual a su vez dependerá del movimiento del servomotor. Los autos entre los que se encuentra el vehículo controlado son representados por los límites 1 y 2 y el acelerador del vehículo sería entonces en este caso el servomotor; señalando que el giro en sentido horario visto desde la Fig. 17 sería aceleración en el vehículo pues esto lo acercaría a el límite 1 que representa el auto delante de nosotros y el giro en sentido antihorario representa una desaceleración del vehículo haciendo que el auto de atrás se acerque a nosotros; recalcando una vez más que ambos autos mantienen siempre una velocidad constante, mientras nuestro auto es el único que varía esta velocidad.

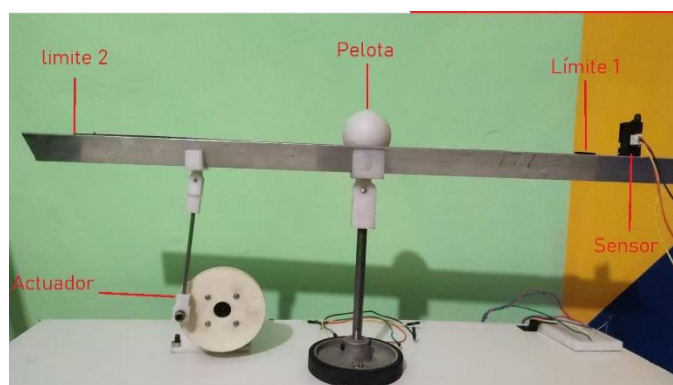


Fig. 17. Sistema Bola-Balancín.



Al iniciar nuestro sistema con la pelota pegada en uno de los límites logramos observar como enseguida el servomotor actúa para que la pelota se encuentre en la distancia media entre los dos límites, alejándola así rápidamente de este extremo, reaccionando en sentido contrario antes de que llegue al otro extremo; haciendo esto repetidamente hasta que la pelota logra ser posicionada en el medio. En este caso fueron adaptados los valores de PID para controlar el servo, sin embargo, no fue necesario este tipo de control, pues bastó con el control P. Logramos llegar a esta conclusión de manera empírica hasta que logramos observar un buen control en el sistema.

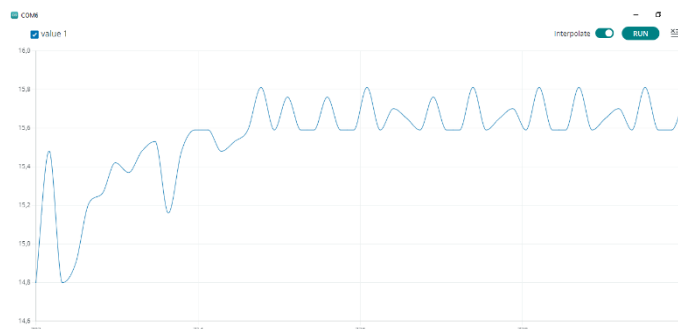


Fig. 18. Sistema Bola-Balancín.

En resumen, nuestro sistema de bola balancín demostró la aplicabilidad de los conceptos de control y la importancia de la retroalimentación en sistemas autónomos.

VII. CONCLUSIONES

En este proyecto, logramos diseñar y construir un sistema de bola balancín que representa el control de velocidad de un vehículo autónomo.

El sistema de bola balancín proporcionó una representación visual clara y tangible del control de velocidad. La pelota en equilibrio sobre la balanza simboliza cómo los vehículos autónomos mantienen una velocidad constante.

Aplicamos principios de control para ajustar el ángulo de la balanza y mantener la pelota en la posición deseada.

Esto se relaciona directamente con la aplicación de sistemas mecatrónicos en la automatización y el control preciso.

La retroalimentación del sensor Sharp fue crucial para tomar decisiones en tiempo real.

Similar a los sistemas mecatrónicos, nuestro sistema se adaptó a cambios y corrigió errores.

Este proyecto es didáctico y educativo, especialmente para estudiantes interesados en mecatrónica y control automático.

Aprendimos sobre la importancia de la interacción entre sensores, actuadores y controladores.

Nuestro proyecto se alinea con las aplicaciones de sistemas mecatrónicos de varias maneras.

El sistema de bola balancín es un ejemplo de automatización y control aplicado a un sistema físico.

La combinación de servomotor, sensor y Arduino refleja la integración típica en sistemas mecatrónicos.

Se requieren conocimientos de mecánica, electrónica y programación, como en muchos sistemas mecatrónicos.

En resumen, nuestro proyecto demuestra cómo los principios mecatrónicos pueden aplicarse creativamente para representar conceptos complejos como el control de velocidad en vehículos autónomos.

REFERENCIAS

Dickmanns, E. (1987). Vehículo Autónomo Guiado por Visión. Universidad de Múnich.

Universidad de Valladolid. (2020). Diseño e Implementación de un Prototipo de Vehículo Eléctrico Experimental con Control de Velocidad.

Instructables. (s.f.). Arduino Ball and Beam (Sistema Barra Bola con Arduino).

ANFAC. (2021). Informe sobre Vehículos Autónomos y Conectados. Asociación Española de Fabricantes de Automóviles y Camiones.



Interfaz de Monitoreo de la Trayectoria de un Brazo Robótico de Dos Grados de Libertad

Mendoza Rai, Muñoz Johan, Ponce Jonathan, Reyes Alejandro.

*Universidad de Guanajuato DICIS, Departamento de Ingeniería Mecánica, Carretera Salamanca - Valle de Santiago km 3.5 + 1.8 Comunidad de, Palo Blanco, 36787 Salamanca, Gto.
jo.munozcontreras@ugto.mx*

Resumen.

Para el desarrollo de la interfaz de monitoreo de la trayectoria del brazo robótico primero se diseñó un brazo robótico de 2GDL, se buscó diseñar un mecanismo minimalista ya que será necesario para trazar las trayectorias en la interfaz, con el desarrollo culminado hemos de mencionar los avances obtenidos respecto a la última revisión.

Una vez se terminó de construir el brazo robótico se comenzó a desarrollar la interfaz de monitoreo, a grandes rasgos hicimos uso de dos herramientas de *software*: Arduino y Python. De Arduino obtuvimos los datos crudos de la posición en espacial de nuestro mecanismo, el cual fue graduado con valores normalizados (de 0 a 1 y de 1 a 2) y enviados a través del puerto serial hacia un programa desarrollado en Python que interpreta esa trama de valores y en una interfaz que permite graficar en 3D estas variables, se puede visualizar la posición de los eslabones del brazo robótico en tiempo real.

Palabras clave. 2GDL, Interfaz, Angular, Robótico.

I. INTRODUCCIÓN

Para el modelado cinemático del brazo, consideramos la metodología usada por Nasr M. Ghaleb et al. (2018) en el que parte de la matriz de Denavit-Hartenberg, en este caso solo optamos por obtener los parámetros de posición angular, ya que el objetivo general del proyecto es mapear esta trayectoria en una interfaz visual mientras que en la bibliografía consultada se utilizaron diferentes herramientas matemáticas para el modelado y la mayoría de estos buscaba realizar un control PID como es el caso de los autores Miuler Blast et al. (2021) y Ayodeji Okubanjo et al. (2017). En el cuerpo de este avance mostraremos los avances obtenidos haciendo énfasis en el proceso creativo y la metodología de análisis para el planteamiento de las ecuaciones necesarias para entender cómo es que se comporta el brazo en condiciones cinemáticas y también dar un adelanto de lo que se logrará a partir de los avances obtenidos hasta el momento.

II. DESARROLLO

1. Materiales

1.1. Brazo robótico

Para el desarrollo del brazo se utilizaron múltiples elementos, para el brazo se imprimió con filamento ABS, esto para garantizar una óptima resistencia y que tiene la capacidad de ser perforado, pulido, pintado, etc.

1.2. Transmisión de energía

Para transmitir la potencia hacia los eslabones se tuvieron varias opciones, motorreductores, motores DC o servomotores, en este caso optamos por servomotores ya que tras varios días de investigación vimos que tienen una alta viabilidad para la aplicación que se está desarrollando.



1.3. Adquisición de variables

En primera instancia, se consideró usar el giroscopio MPU6050, pero al ver que los servomotores nos podían dar la posición angular mapeando la señal *PWM* con un valor angular, llegamos a la conclusión que no sería necesario utilizarlo. Los datos fueron adquiridos por una tarjeta Arduino.

1.4. Herramientas de conexión

Para esto simplemente utilizamos cables tipo Dupont y cable USB tipo B a USB tipo 3.0 para realizar la conexión con la computadora.

2. Desarrollo

2.1. Brazo robótico

Para desarrollar el brazo robótico se inició planteando un diseño a lápiz considerando el uso de 2 servomotores y un sensor MPU6050. Posteriormente se pasó a 3D mediante el uso del software Inventor donde se realizaron los elementos por separado para al final ensamblarlos y una vez no se tuviera algún problema, imprimir los elementos. En un corto lapso e investigado más acerca de formas para medir ángulos en un brazo robótico, se descubrió que con el *encoder* incluido en el servomotor se puede “medir” la posición angular de su eje, solo se necesitan unas cuantas líneas de programación en la interfaz Arduino para lograrlo, esto en cierta forma simplificó el montaje de los elementos en el brazo además de que no será necesario calibrar el sensor cada que se quiera reiniciar el robot como sería con el MPU6050. Una vez impresos los elementos se procedió a la preparación de las piezas puliendo y uniendo los eslabones, se diseñó una base que contuviera el cableado y diera firmeza al brazo para así evitar fallas de forma prematura y en funcionamiento posdesarrollo. Entonces, se comenzó con la conexión de los servomotores; para ello únicamente usamos cables Dupont macho-macho y macho-hembra y dimos más fuerza a las uniones entre estos con cinta adhesiva.

2.2. Control del brazo robótico

En este caso, se optó por realizar un control en lazo abierto de los servomotores mediante *joysticks*,

los cuales se interconectan en la plataforma Arduino y se regulan mediante una señal eléctrica regulada por la variación en los potenciómetros de este. Tras una serie de experimentos y pruebas, se llegó a la conclusión que por lo menos el tipo de *joysticks* que se usaron para este proyecto causaban mucho ruido electrónico y su resolución causaba tirones en la interfaz de monitoreo, la solución que se dio fue utilizar potenciómetros convencionales y se desarrolló una placa PCB para mantenerlos fijos y facilitar la conexión a la par del transporte de todo el proyecto.

2.3. Obtención de parámetros (Arduino)

Para obtener los datos de interés (θ_1 y θ_2) fue necesario realizar una investigación extensa, ya que como se puede apreciar, al inicio del proyecto se tuvo la idea de usar un sensor giroscópico, pero al descubrir que únicamente con el servomotor se podían conocer esos ángulos, optamos por cambiar esa parte del planteamiento del proyecto, por lo tanto, se desarrolló un código de Arduino que nos permitirá controlar los servomotores mediante un *joystick* / potenciómetro como ya se mencionó y a su vez, “mapear” el cambio en posición del potenciómetro con la posición angular del servomotor.

2.4. Obtención de parámetros (Ecuaciones)

Para modelar el sistema físico fue necesario realizar una serie de cálculos para obtener los parámetros de Denavit-Hartenberg que se utilizan para derivar las matrices de transformación homogéneas entre los eslabones en el brazo robótico, a continuación, presentamos estos parámetros en la Tabla 1.

Tabla 1. Parámetros del mecanismo.

Eslabón	L	θ	V	A
1	L_1	θ_1	-	-
2	L_2	θ_2	-	-

Valores de las coordenadas espaciales obtenidas:

$$Px = \cos(\theta_1) * \cos(\theta_2) \quad (6)$$

$$Py = \sin(\theta_1) * \cos(\theta_2) \quad (7)$$

$$Pz = \sin(\theta_2) + 1 \quad (8)$$

III. RESULTADOS

En este apartado se hará un repaso histórico del desarrollo del proyecto con imágenes, diagramas y bocetos como evidencia.

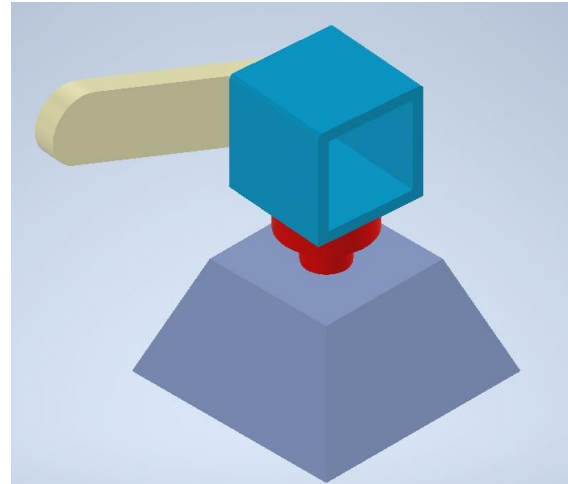


Ilustración 3. Elementos del brazo robótico en 3D.

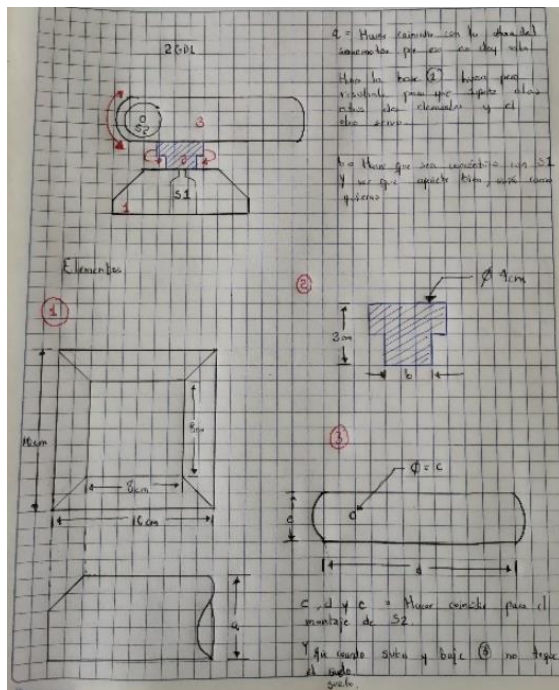
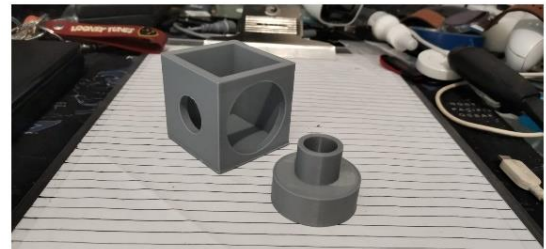


Ilustración 1. Boceto inicial del brazo robótico.



Fotografía 1. Elementos del brazo robótico en físico.

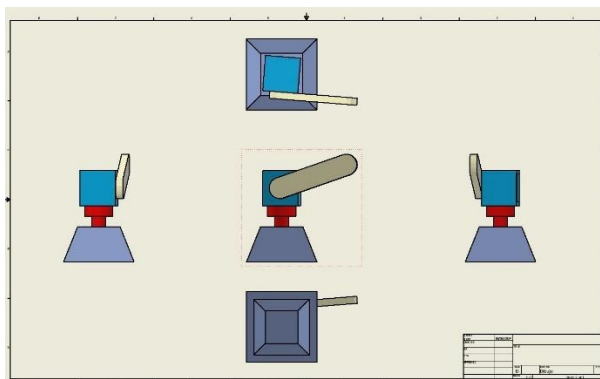


Ilustración 2. Diseño de los elementos del brazo en 2D.



Fotografía 2. Brazo robótico, diseño con joysticks.

```

1 #include <Servo.h>
2 #include <math.h>
3 // Definir los pines para los servomotores
4 #define SERVO_PIN_1 3
5 #define SERVO_PIN_2 11
6
7 // Definir los pines para los joysticks
8 #define JOYSTICK_1_X A0
9 // #define JOYSTICK_1_Y A1
10 #define JOYSTICK_2_X A2
11 // #define JOYSTICK_2_Y A3
12
13 // Definir los objetos Servo
14 Servo servo1;
15 Servo servo2;
16
17 // Variables para almacenar la posición actual de los servos
18 int servo1Pos = 90;
19 int servo2Pos = 90;
20
21 float Angle1;
22 float Angle2;
23
24 void setup() {
25     // Inicializar los objetos Servo
26     servo1.attach(SERVO_PIN_1);
27     servo2.attach(SERVO_PIN_2);
28
29     // Configurar el puerto serie
30     Serial.begin(9600);
31 }
32
33 void loop() {
34     // Leer los valores de los joysticks
35     int joy1X = analogRead(JOYSTICK_1_X);
36     int joy2X = analogRead(JOYSTICK_2_X);
37
38     // Mapear los valores de los joysticks al rango de movimiento del servo
39     int targetServo1Pos = map(joy1X, 0, 1023, 0, 180); // Ángulo del servo 1 controlado por el joystick 1
40     int targetServo2Pos = map(joy2X, 0, 1023, 120, 0); // Ángulo del servo 2 controlado por el joystick 2
41     // Suavizar el movimiento de los servos interpolando linealmente
42     servo1Pos = smoothStep(servo1Pos, targetServo1Pos, 0.05);
43     servo2Pos = smoothStep(servo2Pos, targetServo2Pos, 0.05);

```

Captura 1. Código Arduino (Parte 1).

```

44 // Controlar los servos con las posiciones suavizadas
45 servo1.write(servo1Pos);
46 servo2.write(servo2Pos+45);
47
48 // Convertir los ángulos a radianes
49 float tt1 = (servo1Pos*PI)/180;
50 float tt2 = (servo2Pos*PI)/180;
51
52 // Calcular las coordenadas x, y y z
53 float x = cos(tt1)*cos(tt2);
54 float y = sin(tt1)*cos(tt2);
55 float z = sin(tt2)+1;
56
57 // Imprimir los valores de x, y y z
58 Serial.print(x);
59 Serial.print(",");
60 Serial.print(y);
61 Serial.print(",");
62 Serial.println(z);
63
64 // Esperar un breve periodo de tiempo antes de la próxima lectura
65 delay(100);
66 }
67
68 // Función para suavizar el movimiento interpolando linealmente
69 int smoothStep(int current, int target, float smoothing) {
70     return current + (target - current)*smoothing;
71 }
72
73
74

```

Captura 2. Código Arduino (Parte 2).

```

1 import tkinter as tk
2 from PIL import Image, ImageTk, ImageSequence
3 import matplotlib.pyplot as plt
4 import matplotlib.animation as animation
5 from matplotlib.backends.backend_tkagg import FigureCanvasTkAgg
6 import serial
7 import math
8
9 class AnimatedGIFLabel(tk.Label):
10     def __init__(self, master, filename):
11         super().__init__(master)
12         self._frames = []
13         self._delay = 100 # Adjusted delay to 100 ms
14         self._load_frames(filename)
15         self._frame_index = 0
16         self._is_animating = False
17
18     def _load_frames(self, filename):
19         gif = Image.open(filename)
20         for frame in ImageSequence.Iterator(gif):
21             self._frames.append(ImageTk.PhotoImage(frame))
22
23     def show_next_frame(self):
24         if self._is_animating:
25             self._frame_index = (self._frame_index + 1) % len(self._frames)
26             self.config(image=self._frames[self._frame_index])
27             self.after(self._delay, self.show_next_frame)
28
29     def start_animation(self):
30         if not self._is_animating:
31             self._is_animating = True
32             self.show_next_frame()

```

Captura 3. Código Python (Parte 1).



```

33
34 # Function to calculate angles
35 def Angle(x, y, z):
36     hip = math.hypot(x, y)
37     r1 = y / hip
38     r2 = (z-1) / 1
39     TT1 = math.degrees(math.sin(r1)) * 2
40     TT2 = math.degrees(math.sin(r2)) * 2
41     return TT1, TT2
42
43 # Function to update the plot in real-time
44 def animate(i):
45     try:
46         serial_line = ser.readline().decode().strip()
47         update_plot(serial_line)
48     except:
49         pass
50
51 # Function to update the plot
52 def update_plot(data):
53     global x_data, y_data, z_data
54     x, y, z = map(float, data.split(','))
55     x_data.append(x)
56     y_data.append(y)
57     z_data.append(z)
58
59     # Update the red point position
60     point.set_data(x_data[-1:], y_data[-1:])
61     point.set_3d_properties(z_data[-1:])
62
63     # Draw the line between the blue and red points

```

Captura 4. Código Arduino (Parte 2).

```

64     line.set_data([0, x_data[-1]], [0, y_data[-1]])
65     line.set_3d_properties([1, z_data[-1]])
66
67     # Calculate angles
68     TT1, TT2 = Angle(x_data[-1], y_data[-1], z_data[-1])
69
70     # Update text in the left frame
71     label_text.set(f"Posición en X:\n(x:.2f)\nPosición en Y:\n(y:.2f)\nPosición en Z:\n(z:.2f)")
72     label_text2.set(f"Valor de Theta1:\n(TT1:.2f)\nValor de Theta2:\n(TT2:.2f)")
73
74 # Establish serial connection with Arduino
75 arduino_port = 'COM10' # Replace with the correct port for your Arduino
76 baud_rate = 9600
77 ser = serial.Serial(arduino_port, baud_rate)
78
79 # Initialize data for the plot
80 x_data = []
81 y_data = []
82 z_data = []
83
84 # Create the main Tkinter window
85 root = tk.Tk()
86 root.geometry("1000x736")
87 root.title("Interfaz con GIF y Gráfica")
88 root.configure(bg='white')
89
90 # Create a frame for displaying x, y, z values on the left
91 left_frame = tk.Frame(root, width=200, height=736, bg='white', relief=tk.SOLID)
92 left_frame.pack(side=tk.RIGHT, fill=tk.Y)
93 left_frame.pack_propagate(False)

```

Captura 5. Código Arduino (Parte 3).

```

94 # Create label to display x, y, z values
95 label_text = tk.StringVar()
96 label = tk.Label(left_frame, textvariable=label_text, bg='white', font=('Arial', 15, 'bold'))
97 label.pack(pady=20)
98
99 # Create label to display Theta1 and Theta2 values
100 label_text2 = tk.StringVar()
101 label = tk.Label(left_frame, textvariable=label_text2, bg='white', font=('Arial', 15, 'bold'))
102 label.pack(pady=140)
103
104 # Path to the GIF file
105 gif_path = "RECUP_GUI.gif"
106
107 # Create a custom label widget for the animated GIF
108 animated_label = AnimatedGIFLabel(root, filename=gif_path)
109 animated_label.pack()
110
111 # Set up the 3D plot
112 fig = plt.figure(figsize=(8, 8), facecolor='white')
113 ax = fig.add_subplot(111, projection='3d')
114 ax.set_xlabel('X')
115 ax.set_ylabel('Y')
116 ax.set_zlabel('Z')
117 ax.set_xlim(-1, 1)
118 ax.set_ylim(-1, 1)
119 ax.set_zlim(0, 2)
120
121 # Points to be plotted with Matplotlib
122 constant_point = ax.plot([0], [0], [1], marker='h', markersize=20, color='b', linestyle='None')
123 line = ax.plot([], [], [], color='black', linewidth=5)
124 line2 = ax.plot([0, 0], [0, 0], [0, 1], color='black', linewidth=5)

```

Captura 6. Código Arduino (Parte 4).

```

126 point = ax.plot([], [], [], marker='o', markersize=10, color='r', linestyle='None')
127 constant_point = ax.plot([0], [0], [0], marker='h', markersize=20, color='b', linestyle='None')
128
129 # Add the plot to the Tkinter window
130 canvas = FigureCanvasTkAgg(fig, master=root)
131 canvas.get_tk_widget().pack()
132 canvas.get_tk_widget().configure(bg='white')
133
134 # Run the animation function for real-time plotting
135 ani = animation.FuncAnimation(fig, animate, interval=30)
136
137 # Start the GIF animation
138 animated_label.start_animation()
139
140 # Function to close the serial connection when closing the window
141 def on_closing():
142     ser.close()
143     root.destroy()
144
145 root.protocol("WM_DELETE_WINDOW", on_closing)
146
147 # Run the application
148 root.mainloop()
149

```

Captura 7. Código Arduino (Parte 5).

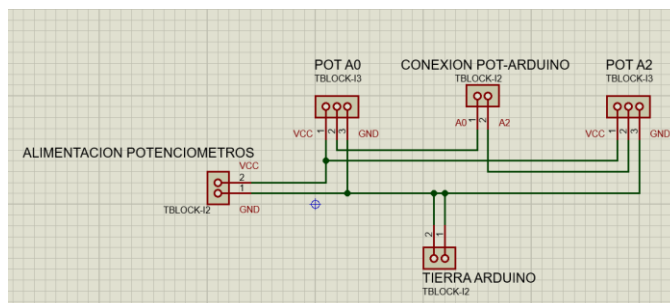


Diagrama 1. Esquemático del control para el brazo robótico.

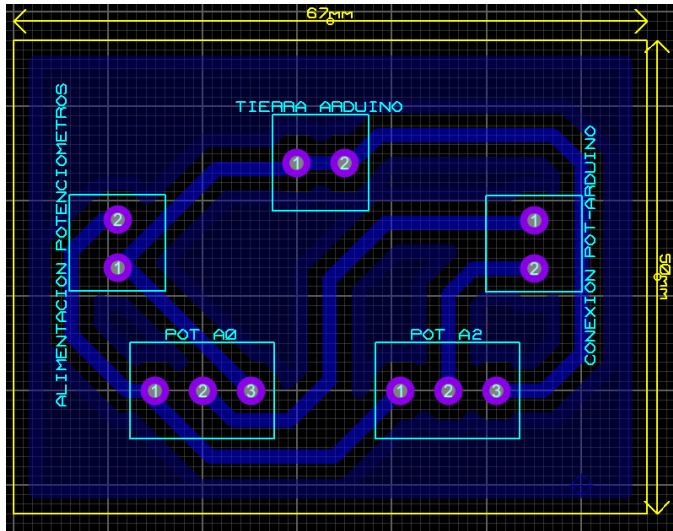


Diagrama 2. Diseño de pistas de impresión del control para el brazo robótico.



Fotografía 3. Control para el brazo robótico terminado.

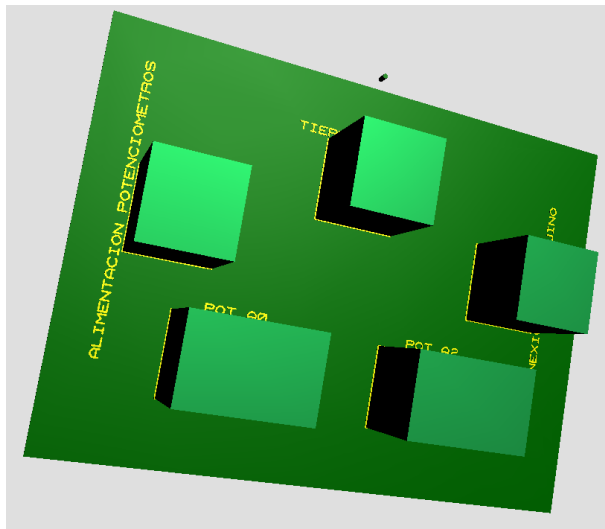
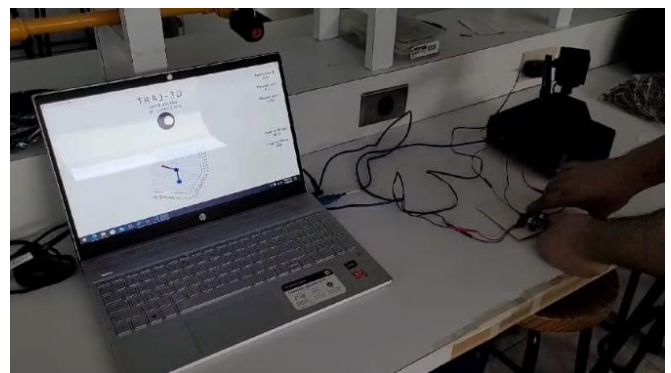


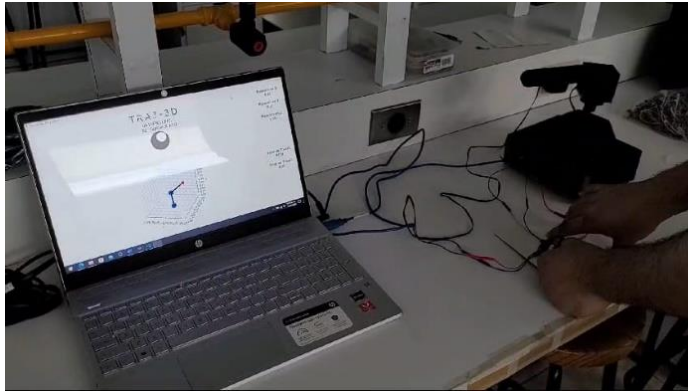
Imagen 1. Vista 3D del control diseñado en Proteus.



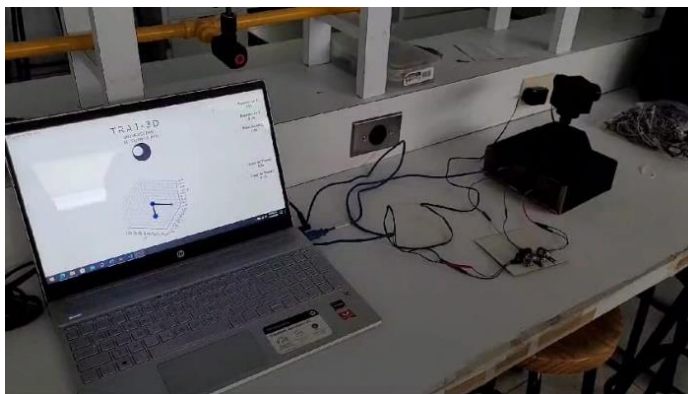
Fotografía 4. Brazo robótico ensamblado con el control y en funcionamiento.



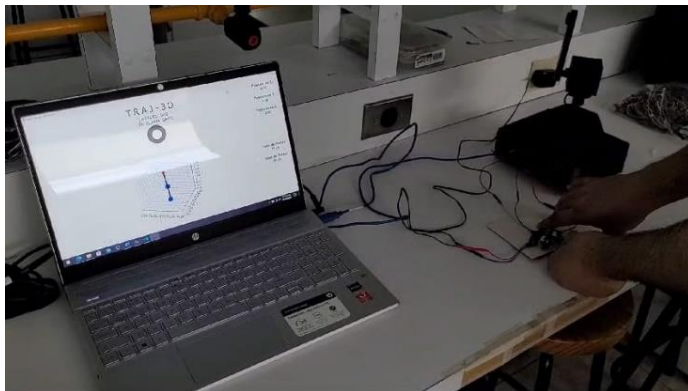
Fotografía 5. Prueba 1 con brazo robótico e interfaz.



Fotografía 6. Prueba 2 con brazo robótico e interfaz.



Fotografía 7. Prueba 3 con brazo robótico e interfaz.



Fotografía 8. Prueba 4 con brazo robótico e interfaz.

IV. REFERENCIAS

Nasr M. Galeb, Ayman A. Aly (2018). Modeling and Control of 2-DOF Robot Arm. International Journal of Eamerging Engineering Research and Technology, 6(11), 24-31. [Modeling-and-Control-of-2-DOF-Robot-Arm.pdf \(researchgate.net\)](#)

Ayodeji Okubango, Oyetola Oluwadamilola, Osifeko Martins, Olawale Olaluwoye. (n.d.). Modeling of 2-DOF robot Arm and Control. ResearchGate. Retrieved December 15, 2017, from https://www.researchgate.net/publication/324531716_Modeling_of_2-DOF_robot_Arm_and_Control

Blas Marquina, M. J., Flores Ramírez Jose, C. J., Mejías Sandoval, A. H., Terrones Escobedo, A. O., & Alva Alcántara, J. H. (2021). IDENTIFICACIÓN NO PARAMÉTRICA DE UN BRAZO ROBÓTICO DE 2 GDL UTILIZANDO REDES NEURONALES RBF. Revista de investigaciones, 10(4), 370–378. <https://doi.org/10.26788/riepg.v10i4.3501>



Adquisición de datos de sobrealimentador eléctrico para motor de combustión interna.

**Ayala García José Francisco^{*1}, Muñoz Carpio Vicente Daniel², Torres Luna Juan Ignacio²,
Chávez Razo Francisco Gustavo²**

^{1, 2} Tecnológico Nacional de México/ITS de Abasolo, Ingeniería en Sistemas Automotrices, Blvd. Cuitzeo de los Naranjos #401, Col. Cuitzeo de los Naranjos, CP. 36976, Abasolo, Guanajuato.

^{*} Autor de correspondencia: frank._.22@outlook.com, francisco.ag@abasolo.tecnm.mx

Resumen. Este proyecto se centra en la instrumentación de un sobrealimentador eléctrico acoplado a un motor de combustión interna, con el objetivo de adquirir datos que permitan optimizar la eficiencia energética del motor y reducir las emisiones de gases contaminantes. El propósito principal es realizar un análisis detallado de los parámetros de funcionamiento del motor con sobrealimentación eléctrica. Para ello, se han instalado una serie de sensores en el sobrealimentador, que permiten monitorear variables tales como presión, temperatura y caudal.

Para medir y registrar estos valores, se ha utilizado un microprocesador que facilita el almacenamiento de los datos capturados por los sensores y su representación gráfica en tiempo real, permitiendo un monitoreo continuo del estado operativo del sistema. Además, se ha empleado un microcontrolador para establecer la comunicación entre los sensores y la unidad de procesamiento, desempeñando un papel crucial en la integración y funcionalidad del sistema.

La instrumentación del sistema para la adquisición de datos ha demostrado ser efectiva en la mejora de la eficiencia, la reducción del retardo y la disminución de las emisiones contaminantes. Gracias a la correcta selección de sensores, la comunicación eficiente entre dispositivos, la calibración, las pruebas y ajustes precisos, así como el desarrollo de una interfaz gráfica para la visualización de datos en tiempo real, se lograron mediciones precisas que han contribuido significativamente al cumplimiento de los objetivos del proyecto.

Palabras clave. *Sobrealimentador, microcontrolador, microprocesador, instrumentación.*

I. INTRODUCCIÓN.

En la actualidad se han desarrollado distintos tipos de motores como los son: el ciclo Otto, diesel y eléctricos, con el propósito de mejorar sus características. Hoy en día el motor de gasolina de ciclo Otto tiene una eficiencia de entre el 20% y el 30%, en el mejor de los casos; es ahí

donde surge la necesidad de incrementar su eficiencia, lo cual puede ser logrado con la implementación de un sobrealimentador. Existen actualmente dos tipos de turbocompresores: el turbo convencional y el turbo eléctrico. El turbo convencional utiliza un sistema sencillo pero eficaz, que aprovecha los gases de escape



del motor para absorber aire del ambiente, enriqueciendo así la mezcla de combustión. Por otro lado, el turbo eléctrico, a diferencia del convencional, cuenta con un pequeño motor eléctrico que facilita la absorción de gases, lo que lo hace más eficiente y proporciona un impulso inmediato, entre otras ventajas.

Existe otra alternativa que es un sobrealimentador, el cual ofrece varias ventajas sobre el turbo convencional, como una mayor eficiencia y un impulso inmediato, entre otras características. Sin embargo, una de sus principales desventajas es su costo, que es significativamente más elevado.

En este proyecto, se busca analizar el comportamiento de un motor de combustión equipado con un sobrealimentador acoplado a un motor de Corriente Directa (CD). Dado que el sobrealimentador presenta un cierto retardo (lag), surgió la necesidad de instrumentar el motor para adquirir datos de entrada del sobrealimentador y observar su funcionamiento.

II. MARCO TEORÍCO.

El **turbocompresor** es un dispositivo mecánico que aumenta la presión y la densidad del aire que se suministra a un motor de combustión interna. El turbo funciona mediante el aprovechamiento de la energía de los gases de escape del motor, los cuales hacen girar una turbina. (Suárez-Llanos Outeiriño, L, 2015).

El **turbo lag** se refiere al tiempo que tarda el turbocompresor en alcanzar la velocidad suficiente para generar la presión de sobrealimentación necesaria que proporcione una salida de potencia significativa. Durante este periodo, en el que el turbocompresor aún no ha comenzado a producir impulso, el motor puede sentirse lento o poco receptivo, lo que puede resultar frustrante para el conductor. Este efecto es especialmente perceptible cuando el conductor pisa el pedal del acelerador por primera vez, ya que el motor puede tardar un momento en responder antes de que el turbocompresor empiece a proporcionar la potencia adicional.

Un **sobrealimentador**, también conocido como "supercargador", es un dispositivo mecánico que aumenta la presión o densidad del aire que se introduce en el motor de combustión interna. Esto permite que se

queme más combustible dentro del cilindro, incrementando la potencia del motor sin necesidad de aumentar su tamaño.

El sobrealimentador es impulsado mecánicamente por el motor, generalmente mediante una correa conectada al cigüeñal, lo que significa que consume parte de la potencia del motor para funcionar. Sin embargo, la ganancia en potencia total del motor es significativamente mayor que la pérdida de potencia causada por el funcionamiento del sobrealimentador.

Un transductor de presión es un dispositivo diseñado para medir y convertir la presión aplicada sobre él en una señal eléctrica o electrónica que pueda ser interpretada y utilizada en sistemas de control, monitoreo o adquisición de datos. Estos dispositivos son ampliamente utilizados en una variedad de aplicaciones en la industria, la ingeniería, la investigación y muchos otros campos. (Morcelle del Valle, P. 2018).

Sensor termopar tipo k: Un termopar tipo K es un sensor de temperatura que se utiliza para medir temperaturas en un rango de -200 a 1350 grados Celsius. Consiste en dos metales diferentes, generalmente aleaciones de níquel-cromo y níquel-aluminio, que están unidos en un extremo para formar una junta de medición. (Rodríguez , Ocampo, & Ortega , 2007)

III. DESARROLLO.

Para la implementación del sistema de recopilación de datos se hizo uso de un equipo previamente armado, el cual consta de una motobomba POWER CAT, es un motor de 4 tiempos el cual cuenta con una potencia de 2.5 [Hp], un sobrealimentador y un motor de CD; se utilizó un conjunto de componentes electrónicos para la obtención y visualización de los datos.

La unidad principal y de tratamiento de datos será el microprocesador Raspberry Pi 3+. Dado que dicho microprocesador tiene limitaciones en el tratamiento de señales analógicas, se empleó un Arduino Uno para leer las señales analógicas de los sensores de flujo y el transductor de presión, además, se usó un convertidor de voltaje de 110 V_{DC} a 12 V_{DC} para alimentar el módulo controlador del motor con potenciómetro. Para la selección de los sensores se tomarán en cuenta



diferentes parámetros como precio, características, precisión las cuales se presentan en la Tabla 1.

Tab. 1. Sensores utilizados.

Sensor	Características	Precisión	Precio
Sensor de temperatura tipo k	- Rango de temperatura: 0-800 C. - Aislamiento interno: Fibra de vidrio. - Cortos tiempos de respuesta.	Con un grado de error de 2.2 C	150 MXN
Transductor Presión Full Gauge Sb69	- Gran estabilidad y precisión. - Tiene un rango de 0-200 psi. - Protección: IP69K		2500 MXN
Sensor de flujo YF-S201B	- Corriente de operación: 15 mA (5 V) - Paso de caudal: 1 ~ 30 L/min		150MXN

Junto con el termopar tipo K se empleó un módulo de convertidor de señal analógica a digital MAX6675 conectado directamente a la Raspberry.

Pruebas con el sensor termopar.

Se utilizó una Raspberry Pi para visualizar los valores a través de una interfaz creada en Python, lo que permitió monitorear el comportamiento del sensor. Dado que la Raspberry Pi solo puede leer señales digitales, fue necesario utilizar el módulo MAX6675 para convertir las señales analógicas del termopar en un formato digital que la Raspberry Pi pudiera interpretar. Esto facilitó la lectura de las temperaturas medidas por el sensor a través de la interfaz. Las conexiones entre el sensor y la Raspberry Pi se detallan en la Fig. 1, donde se muestra el módulo MAX6675, sus conexiones con el sensor y la Raspberry Pi.



Fig. 1. Conexiones del sensor termopar.

Una vez realizadas las conexiones, se utilizó un código simple que permitió obtener las mediciones del sensor, con el objetivo de evaluar su precisión. Las primeras mediciones se realizaron a temperatura ambiente, lo que

permitió verificar las lecturas. Como se muestra en la Fig. 2, las mediciones obtenidas oscilaron entre 29°C y 30°C, dado que la temperatura exterior en el momento de la medición era de 30°C, se puede concluir que las mediciones del sensor son correctas.

```

1 Termopar = MAX6675.MAX6675(CSK,C5,S0)
2
3 try:
4     while True:
5         temp_celsius = Termopar.readTempC()
6         temp_fahrenheit = (temp_celsius * 9/5) + 32
7         print(f"Temperatura (°C): {temp_celsius:.2f}")
8
9
10
11
12

```

Shell

```

Temperatura (°C): 29.75, Temperatura (°F): 85.55
Temperatura (°C): 29.75, Temperatura (°F): 85.55
Temperatura (°C): 29.75, Temperatura (°F): 85.55
Temperatura (°C): 29.75, Temperatura (°F): 85.55
Temperatura (°C): 30.00, Temperatura (°F): 86.00
Temperatura (°C): 29.25, Temperatura (°F): 84.65
Temperatura (°C): 29.25, Temperatura (°F): 84.65
Temperatura (°C): 29.75, Temperatura (°F): 85.55
Temperatura (°C): 29.25, Temperatura (°F): 84.65
Temperatura (°C): 29.25, Temperatura (°F): 84.65

```

Fig. 2. Mediciones a temperatura ambiente.

Después de realizar las primeras mediciones se realizaron mediciones con objetos más fríos con el objetivo de someter al sensor a otro tipo de condiciones y observar su comportamiento, en la Fig. 3 se muestra como la temperatura bajó desde 30°C hasta los 22°C.

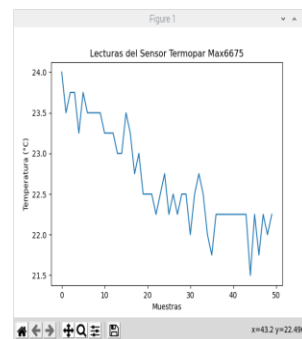


Fig. 3. Mediciones con temperaturas más bajas.

Pruebas con el transductor de presión.

Para el sensor de presión se realizaron de igual manera algunas mediciones con un código sencillo el cual permite solo leer los datos del sensor, de este modo se puede comprobar su efectividad con la diferencia de que en este sensor se utilizó la placa Arduino para poder convertir las señales analógicas a digitales, en la Fig. 4 se muestran las conexiones del transductor de presión hacia el Arduino. Posteriormente se realizó la medición de la presión del entorno. Los resultados obtenidos fueron aceptables, aunque se detectó un poco de ruido en las mediciones. Sin embargo, no fue significativo, ya que las variaciones entre los resultados obtenidos y los

valores reales fueron mínimas, como se puede observar en la Fig. 5.

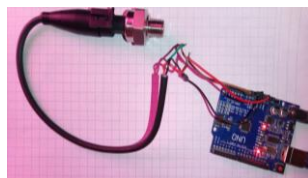


Fig. 4. Conexiones del transductor de presión.

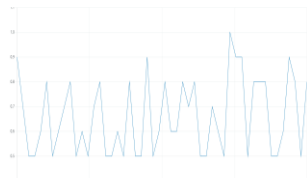


Fig. 5. Mediciones del transductor de presión.

Pruebas con el sensor de caudal.

Para el sensor de caudal, se colocaron dos tubos, uno en cada extremo del sensor como se muestra en la Fig. 6, esto con el fin de permitir el suministro del fluido. El funcionamiento del sensor se basa en el paso del fluido a través de uno de los extremos, lo que hace girar una pequeña turbina en su interior. La cantidad de giros de esta turbina determina las mediciones obtenidas.



Fig. 6. Conexión del sensor de flujo.

Una vez realizadas las conexiones, se desarrolló un pequeño código para probar el funcionamiento del sensor. Los resultados obtenidos, mostrados en la Fig. 7, reflejan el caudal medido al introducir el fluido por uno de los extremos del sensor. Estos resultados permiten evaluar la eficacia del sensor en la medición del caudal bajo las condiciones establecidas.

3.52 L/m
3.60 L/m
3.60 L/m
3.67 L/m
3.67 L/m
3.75 L/m
3.83 L/m
3.83 L/m
3.91 L/m
3.91 L/m
3.99 L/m
3.99 L/m
4.07 L/m
4.07 L/m
4.15 L/m
4.15 L/m
4.23 L

Fig. 7. Mediciones del sensor de flujo.

Para la instalación de los sensores se adaptó la tubería de salida del sobrealimentador, el sensor termopar se colocó en la parte inferior de la tubería como se muestra en la Fig. 9.



Fig. 9. Instalación del sensor de presión.

En la Fig. 10 se muestran las conexiones realizadas para la comunicación del sensor de caudal con la placa Arduino. Este sensor tiene la función de medir el caudal de la tubería de salida del sobrealimentador y registrar los datos en la interfaz correspondiente.

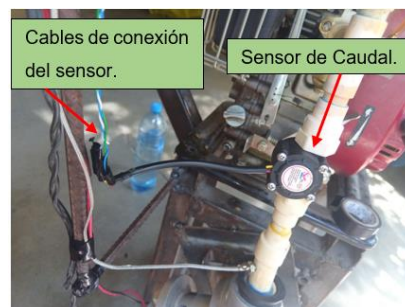


Fig. 10. Instalación del sensor de caudal.

El transductor de presión se colocó en la parte superior de la tubería con ayuda de algunos acoples, se logró instalar de modo que se evitaran pérdidas de los gases quedando sellado herméticamente, como se muestra en la Fig. 11.



Fig. 11. Instalación del transductor de presión.



En la Fig. 12, se observa el circuito controlador de voltaje, la función principal es regular el voltaje que se suministra al motor eléctrico ajustado mediante un potenciómetro, permitiendo así regular la velocidad del motor y controlando la entrada de aire del sobrealimentador. Gracias a esta configuración, es posible controlar de manera efectiva el funcionamiento del sobrealimentador, optimizando su rendimiento y eficiencia.

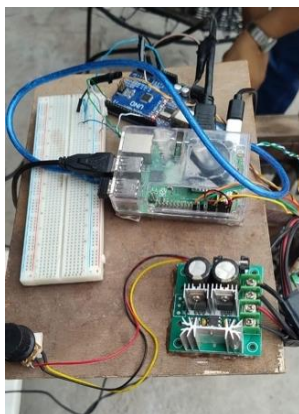


Fig. 12. Conexiones del circuito.

En la Fig. 13 se muestra el diagrama del circuito para la adquisición de datos, se utilizó un conjunto de circuitos y componentes electrónicos para la obtención, tratamiento y visualización de estos. En este diagrama se muestra la estructura general del circuito utilizado para la recopilación de los datos, así como su funcionamiento lógico.

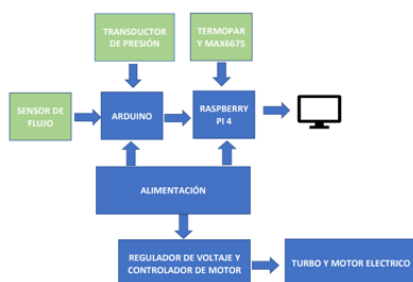


Fig. 13. Diagrama de comunicación del sistema.

IV. RESULTADOS.

Para verificar el correcto funcionamiento del sistema, se propuso una serie de casos de estudio bajo diferentes condiciones de operación. Inicialmente, se realizaron mediciones con el motor de combustión apagado y, de

manera gradual, se incrementó la velocidad de rotación del motor de CD mediante la variación del voltaje con el potenciómetro.

La prueba realizada con el motor apagado arrojó una presión promedio de 7.22 psi. Este valor puede interpretarse como la presión atmosférica existente en el entorno de la medición. Esta interpretación se basa en el hecho de que, con el motor apagado, no hay generación adicional de presión interna por parte del sistema del motor o del sobrealimentador. Por lo tanto, la lectura refleja predominantemente la presión atmosférica, el transductor de presión inicia la medición de datos cuando el sobrealimentador alcanza el 50% de su operación, lo cual equivale a 6V. En los primeros instantes de funcionamiento se observan cambios moderados en la presión del sistema, como se puede observar en la Fig. 14, se aprecia un cambio significativo en la medición conforme se incrementa la velocidad.



Figura 14. Señal de motor puesto a marcha.

En la Fig 15, se observa un aumento significativo de la presión, alcanzando más de 20 psi antes de disminuir a aproximadamente 18 psi; este comportamiento se registró durante una prueba en la que el sobrealimentador operaba a plena carga y a la máxima capacidad de su motor eléctrico. Este aumento inicial de la presión puede atribuirse al funcionamiento óptimo del sobrealimentador bajo condiciones de máxima exigencia.



Figura 15. Señal de transductor de presión, motor encendido.

Con el motor apagado, el sensor termopar, acoplado a la tubería del sobrealimentador, registró la temperatura ambiente, la cual era de 31°C aproximadamente como se muestra en la Fig. 16. Esta medición es crucial, ya que proporciona una referencia base para evaluar el



rendimiento térmico del sistema cuando el motor esté en funcionamiento.



Fig. 16. Lectura de temperatura, con el motor apagado.

Ahora en la Fig. 17 se presenta el inicio del motor con 6V, lo cual corresponde al 50% de su potencia nominal, considerando que el motor acoplado al sobrealimentador opera a un voltaje nominal de 12V, se observa una disminución significativa de la temperatura, registrando valores por debajo de la temperatura ambiental. Inicialmente, la temperatura ambiente era de 33°C, y durante la prueba, la temperatura medida descendió a un promedio de 4°C.

Esta notable reducción puede ser atribuida a varios factores, el efecto de enfriamiento del sistema, la influencia de corrientes de aire frío, o la absorción térmica por parte de los componentes del sobrealimentador.

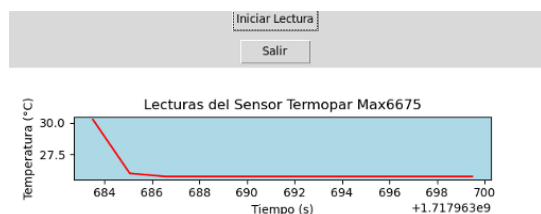


Fig. 17. Disminución de la temperatura.

Durante la prueba, el caudal inicial medido fue de 0 L/min, lo cual era el resultado esperado debido a la ausencia de actividad en el sobrealimentador.

En la Fig. 18 se muestra un incremento significativo del caudal debido al funcionamiento del sobrealimentador a plena carga. Este aumento en el caudal tiene un efecto visible en el motor, ya que se observa un incremento en las revoluciones del motor sin necesidad de manipular el acelerador.

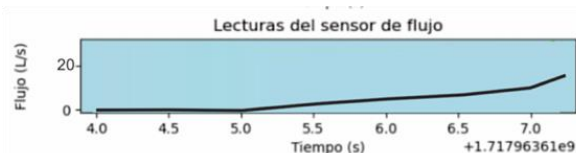


Figura 18. Incremento del flujo de aire en el motor.

V. CONCLUSIONES.

El uso de un sobrealimentador en un motor de combustión interna proporciona un notable incremento en la potencia y eficiencia del motor al aumentar la cantidad de aire disponible para la combustión. Esto compensa la diferencia de presiones en el entorno, mejorando así el desempeño del motor. Esta mejora en la combustión puede resultar en un aumento del rendimiento y una mayor capacidad de respuesta del vehículo. Durante las pruebas, se observaron tanto gráficamente como físicamente las mejoras en el desempeño del motor al incrementar la cantidad de aire suministrado. Se evidenciaron cambios en diferentes variables, y se notó un aumento considerable en las RPM del motor, aunque las mediciones no fueron completamente precisas debido a las limitaciones en la calidad de los sensores. Sin embargo, los cambios en el sistema fueron claramente perceptibles. La implementación de un sistema de adquisición de datos es una buena opción para mejorar el rendimiento y la eficiencia de los motores de combustión interna, esto permitiría un monitoreo continuo de variables críticas del sobrealimentador y con ayuda de un regulador ajustar activamente el flujo de aire suministrado al motor.

REFERENCIAS.

- Suárez-Llanos Outeiriño, L. (2015). *TurboCompresor en banco de ensayo: Características operativas* [Trabajo de Fin de Grado, Universidad Carlos III de Madrid, Escuela Politécnica Superior].
- Rodríguez , E. J., Ocampo, J. W., & Ortega , C. A. (Mayo De 2007). *Revistaciencia*. Obtenido De Medición De Temperatura: Sensores Termoeléctricos: <https://revistas.utp.edu.co/index.php/revistaciencia/article/view/5509/3181>
- Rodríguez , E. J., Ocampo, J. W., & Ortega , C. A. (Mayo De 2007). *Revistaciencia*. Obtenido De Medición De Temperatura: Sensores.

AGRADECIMIENTOS.

Agradecemos la colaboración en el proyecto a los estudiantes: Aldaco Murillo Brauil y Ramírez Mosqueda Carlos Manuel.



Control PD de un quadrotor

Lavín Delgado Jorge Enrique^{*1}, Mota Cruz Juan Esteban¹, Urueta Hinojosa Daniel Edahi¹

¹ Universidad Politécnica del Estado de Guerrero, Dirección de Ingeniería en Redes y Telecomunicaciones, Carretera Federal Iguala-Taxco, Puente Campuzano, Taxco de Alarcón, Guerrero, México.

* Autor de correspondencia: jlavin@upeg.edu.mx

Resumen. En este trabajo se diseña y desarrolla un controlador PD de un quadrotor para tareas de seguimiento de trayectoria. El modelo dinámico del sistema es obtenido mediante el formalismo de Newton-Euler. El controlador es sintonizado con el algoritmo de búsqueda de Cucko, con el error cuadrático medio (RMSE) como función a minimizar. Simulaciones numéricas demostraron un desempeño deseable bajo diferentes condiciones de operación, como cambios de trayectoria de referencia y perturbaciones externas.

Palabras clave. Controlador PD, Quadrotor, Modelo dinámico.

I. INTRODUCCIÓN

El estudio de quadrotores no tripulados ha cautivado la atención de la comunidad científica en años recientes. Estos se caracterizan por su capacidad de vuelo estacionario, capacidad de despegue y aterrizaje verticales y alta maniobrabilidad, así como su alta movilidad en espacios restringidos. De este modo, son ampliamente utilizados en tareas de vigilancia [1], rescate [2], entrega de paquetes [3] y transporte [4], por mencionar algunas. Por otra parte, el control de estos sistemas es una tarea desafiante debido a su dinámica multi-entrada multi-salida (MIMO) y no lineal, además de su sensibilidad a perturbaciones externas como ráfagas de viento e incertidumbres paramétricas [5].

II. Modelo dinámico

Un quadrotor se compone de cuatro hélices en configuración cruzada, dos de ellas giran en el sentido de las manecillas del reloj y dos en sentido contrario. De este modo, al variar la velocidad angular las hélices, puede realizar diferentes movimientos de rotación y traslación [6]. La Fig. 1 muestra el diagrama de un quadrotor, con referencial $\mathbf{O}_b \mathbf{x}_b \mathbf{y}_b \mathbf{z}_b$ unido a su centro de masas, y el referencial mundo $\mathbf{O}_w \mathbf{x}_w \mathbf{y}_w \mathbf{z}_w$. Ahora bien,

sean $\mathbf{q}(t) = [x, y, z, \phi, \theta, \psi]^T$ las coordenadas generalizadas del sistema, donde $\xi(t) = [x, y, z]^T$ representa la posición del centro de masa con respecto al referencial del mundo, mientras que $\eta(t) = [\phi, \theta, \psi]^T$ denota los ángulos de alabeo, cabeceo y guiñada respectivamente, que describen la orientación del quadrotor. Las fuerzas que actúan sobre el sistema son la gravedad y las fuerzas de empuje, F_i , generadas por cada uno de los rotores [7]. De acuerdo con la formulación de Newton-Euler la ecuación de movimiento lineal se expresa como

$$m\ddot{\xi}(t) = \begin{bmatrix} 0 \\ 0 \\ -mg \end{bmatrix} + \mathbf{R}_b^w(t) \begin{bmatrix} 0 \\ 0 \\ \sum_{i=1}^4 F_i(t) \end{bmatrix}, \quad (1)$$

con $F_i = b\omega_i^2(t)$, donde m , b y $\omega_i(t)$ denotan la masa del sistema, el factor de empuje y la velocidad del i -ésimo rotor respectivamente. Además, $\mathbf{R}_b^w(t)$ es la matriz de rotación entre ambos referenciales dada por



$$\mathbf{R}_b^w(t) \begin{bmatrix} C_\theta C_\psi - S_\phi S_\theta S_\psi & -C_\theta S_\psi & S_\theta C_\psi + S_\phi C_\theta S_\psi \\ C_\theta S_\psi + S_\phi S_\theta C_\psi & C_\theta C_\psi & C_\theta S_\psi - S_\phi C_\theta C_\psi \\ -C_\phi S_\theta & S_\phi & C_\phi C_\theta \end{bmatrix}, \quad (2)$$

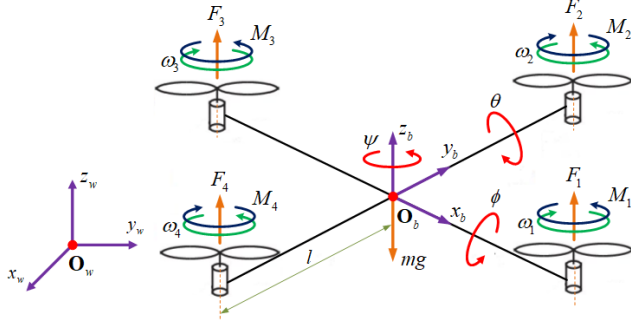


Fig. 1: Quadrotor.

donde S_x y C_x representan $\sin(x)$ y $\cos(x)$ respectivamente. Por otra parte, cada rotor también produce un momento, $M_i = d\omega_i^2(t)$, donde d representa el factor de arrastre. La ecuación de movimiento angular viene dada por

$$\mathbf{I}\dot{\boldsymbol{\omega}}(t) = -\boldsymbol{\omega}(t) \times \mathbf{I}\boldsymbol{\omega}(t) + \boldsymbol{\tau}(t), \quad (3)$$

donde $\mathbf{I} = \text{diag}\{I_{xx}, I_{yy}, I_{zz}\}$ denota la matriz de inercia referente al centro de masa a lo largo de los ejes $\mathbf{x}_b$, $\mathbf{y}_b$ y $\mathbf{z}_b$, $\boldsymbol{\omega}(t) = [\omega_x, \omega_y, \omega_z]^T$ la velocidad angular del quadrotor relativa al referencial $\mathbf{O}_b\mathbf{x}_b\mathbf{y}_b\mathbf{z}_b$, y $\boldsymbol{\tau}(t)$ los momentos aplicados al sistema por los rotores, calculados como

$$\boldsymbol{\tau}(t) = \begin{bmatrix} \tau_\phi(t) \\ \tau_\theta(t) \\ \tau_\psi(t) \end{bmatrix} = \begin{bmatrix} l(F_2 - F_4) \\ l(F_3 - F_1) \\ M_2 + M_4 - M_1 - M_3 \end{bmatrix}, \quad (4)$$

con l la distancia medida del eje de rotación de los rotores hasta el centro del quadrotor. Ahora bien, las componentes de la velocidad angular del quadrotor están relacionados con la derivada de los ángulos de balanceo, cabeceo y guiñada según mediante

$$\begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} = \mathbf{R}_\omega^\xi(t) \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix}, \quad (5)$$

donde

$$\mathbf{R}_\omega^\xi(t) = \begin{bmatrix} C_\theta & 0 & -C_\phi S_\theta \\ 0 & 1 & S_\phi \\ S_\theta & 0 & C_\phi C_\theta \end{bmatrix}, \quad (6)$$

Finalmente, la Eq. (1) puede expresarse como

$$m\ddot{\boldsymbol{\xi}}(t) = \begin{bmatrix} 0 \\ 0 \\ -mg \end{bmatrix} + \mathbf{R}_b^w(t) \begin{bmatrix} 0 \\ 0 \\ u(t) \end{bmatrix}, \quad (7)$$

con $u = \sum_{i=1}^4 F_i(t)$.

III. Control PD

El control proporcional derivativo (PD) es una estrategia de control ampliamente utilizada por su simplicidad matemática, facilidad para implementarse en sistemas en tiempo real y por ser computacionalmente eficiente.

Sea $\mathbf{r}_d = [x_d, y_d, z_d, \psi_d]^T$ la trayectoria de referencia, el error de seguimiento se define como

$$\mathbf{e}(t) = \mathbf{r}_d(t) - \mathbf{r}(t), \quad (8)$$

donde $\mathbf{r} = [x, y, z, \psi]^T$ es la trayectoria actual del quadrotor. En este contexto, se debe garantizar que el error converja asintóticamente a cero. Por lo tanto, se diseña una ley de control de PD como [8]

$$\ddot{\mathbf{r}}_d(t) - \ddot{\mathbf{r}}_c(t) + \mathbf{K}_d \dot{\mathbf{e}}(t) + \mathbf{K}_p \mathbf{e}(t) = 0, \quad (9)$$

donde $\mathbf{K}_d$ y $\mathbf{K}_p$ son matrices diagonales definidas positivas, mientras que $\ddot{\mathbf{r}}_c(t)$ es la aceleración calculada por el controlador. De este modo, se diseña un controlador para vuelo estacionario linealizando la dinámica. De la Eq. (9) resulta:

$$\begin{aligned} \ddot{x}_c(t) &= \ddot{x}_d(t) + k_{d_x} [\dot{x}_d(t) - \dot{x}(t)] + k_{p_x} [x_d(t) - x(t)], \\ \ddot{y}_c(t) &= \ddot{y}_d(t) + k_{d_y} [\dot{y}_d(t) - \dot{y}(t)] + k_{p_y} [y_d(t) - y(t)], \\ \ddot{z}_c(t) &= \ddot{z}_d(t) + k_{d_z} [\dot{z}_d(t) - \dot{z}(t)] + k_{p_z} [z_d(t) - z(t)], \end{aligned} \quad (10)$$



Se diseña un control para el vuelo estacionario linealizándola dinámica del sistema, obteniendo:

$$\begin{aligned}\phi_c(t) &= \frac{1}{g} [\ddot{x}_c(t) S_{\psi_d} - \ddot{y}_c(t) C_{\psi_d}], \\ \theta_c(t) &= \frac{1}{g} [\ddot{x}_c(t) C_{\psi_d} + \ddot{y}_c(t) S_{\psi_d}], \\ \psi_c(t) &= \psi_d(t),\end{aligned}\quad (11)$$

Por lo tanto, las leyes de control se pueden escribir como:

$$u(t) = m[g + z_c(t)], \quad (12)$$

$$\begin{bmatrix} \tau_\phi(t) \\ \tau_\theta(t) \\ \tau_\psi(t) \end{bmatrix} = \mathbf{I} \begin{bmatrix} k_{p_\phi}(\phi_c - \phi) + k_{d_\phi}(\omega_{x_c} - \omega_x) \\ k_{p_\theta}(\theta_c - \theta) + k_{d_\theta}(\omega_{y_c} - \omega_y) \\ k_{p_\psi}(\psi_c - \psi) + k_{d_\psi}(\omega_{z_c} - \omega_z) \end{bmatrix}, \quad (13)$$

Note que $[\omega_{x_c}, \omega_{y_c}, \omega_{z_c}]^T$ se calcular a partir de las Eqs.

(5) y (11). Para atenuar el efecto de las perturbaciones externas, el control de conmutación se define como

$$\begin{aligned}u(t) &= \sigma_s[u(t)], \\ \tau_\phi(t) &= \sigma_s[\tau_\phi(t)], \\ \tau_\theta(t) &= \sigma_s[\tau_\theta(t)], \\ \tau_\psi(t) &= \sigma_s[\tau_\psi(t)],\end{aligned}\quad (14)$$

donde a es una constante positiva y $\sigma_s[\cdot]$ es una función de saturación de la forma:

$$\sigma_s[x] = \begin{cases} -a & \text{si } x < -a \\ x & \text{si } -a \leq x \leq a, \\ a & \text{si } x > a \end{cases}, \quad (15)$$

IV. Resultados de simulación

En esta sección se muestra el desempeño del controlador PD para un quadrotor mediante simulaciones realizadas en Matlab/Simulink R2021a (versión de 64 bits) para Windows con 8 GB de RAM. La Tabla 1 muestra los parámetros del sistema utilizados en este trabajo.

Parámetro	I_{xx}	I_{yy}	I_{zz}	m
Valor	0.00025	0.000232	0.0003738	0.18

Tabla 1: Parámetros del quadrotor.

Simulation 1. La primera tarea es llevar el cuadricóptero a la posición $[x_d, y_d, z_d]^T = [0.7, 0, 4]^T$ m con una orientación de $[\phi_d, \theta_d, \psi_d]^T = [0, 0, 0]^T$ rad, y después realizar el seguimiento de la trayectoria dada por

$$\begin{aligned}x_d(t) &= a_1 [1 + b_1 \cos(\alpha_1 t)] \cos(\omega_1 t), \\ y_d(t) &= a_2 [1 + b_2 \cos(\alpha_2 t)] \cos(\omega_2 t), \\ z_d(t) &= 4,\end{aligned}\quad (16)$$

Los valores de los parámetros de esta trayectoria se muestran en la Tabla 2, mientras que las condiciones iniciales consideradas son $\mathbf{q}(t) = [0, 0, 0, 0, 0, 0]^T$ y $\dot{\mathbf{q}}(t) = [0, 0, 0, 0, 0, 0]^T$. La Tabla 3 describe las ganancias del controlador, obtenidas mediante el algoritmo Cuckoo Search [9] minimizando el error cuadrático medio (RMSE) dado por

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{k=1}^n (q_{d_k} - q_k)^2}, \quad (17)$$

donde n es el número de mediciones y q_d la trayectoria deseada. El valor umbral utilizado en la función de saturación para las leyes de control τ_ϕ , τ_θ y τ_ψ es $a = 0.2$; mientras que para u , los umbrales inferior y superior se consideran 0 y $2mg$ respectivamente.

Parámetro	a_1	a_2	b_1	b_2	α_1	α_2	ω_1	ω_2
Valor	0.65	0.65	0.10	0.10	1.05	1.05	0.15	0.15

Tabla 2: Parámetros de la trayectoria de referencia.

Ganancia	k_{p_x}	k_{p_y}	k_{p_z}	k_{d_x}	k_{d_y}	k_{d_z}
Valor	4	4	25	4	4	10
Ganancia	k_{p_ϕ}	k_{p_θ}	k_{p_ψ}	k_{d_ϕ}	k_{d_θ}	k_{d_ψ}
Valor	625	625	625	50	50	50

Tabla 3: Ganancias del controlador.



La Fig. 2 muestra la trayectoria 3D seguida por el quadrotor, mientras que las Figs. 3 y 4 describen el seguimiento de trayectoria y la señal de error para todas las coordenadas generalizadas del sistema respectivamente. En ellas se observa que el controlador PD tiene la capacidad de estabilizar el sistema atenuando los sobre impulsos a la vez que presenta un rendimiento deseable para el seguimiento de la trayectoria. La Tabla 4 muestra los resultados del error cuadrático medio (RMSE) para todas las variables de estado del sistema.

Variable	Sin perturbaciones	Con perturbaciones
x	0.0036	0.0036
y	0.0016	0.0016
z	0.0000	0.0001
ϕ	0.0002	0.0299
θ	0.0018	0.0372
ψ	0.0019	0.0044

Tabla 4: Valores de RMSE.

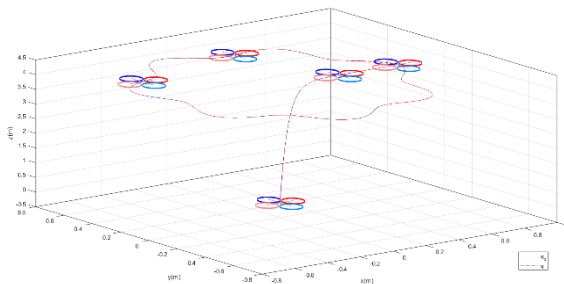


Fig. 2: Trayectoria 3D.

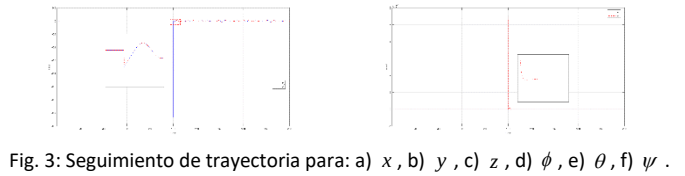
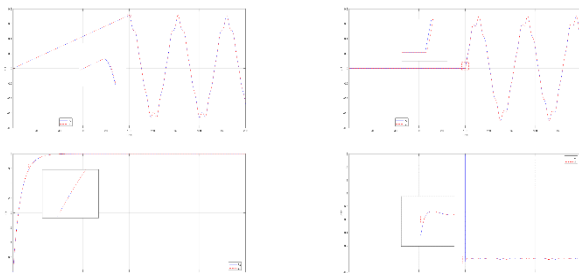


Fig. 3: Seguimiento de trayectoria para: a) x , b) y , c) z , d) ϕ , e) θ , f) ψ .

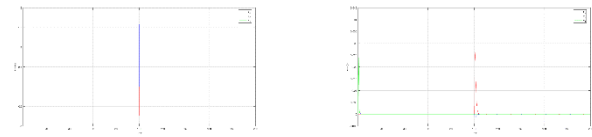


Fig. 4: Error de seguimiento para: a) x , b) y , c) z , d) ϕ , e) θ , f) ψ .

Simulation 2. Los quadrotores son generalmente son afecatos por perturbaciones externas durante el vuelo, como ráfagas de viento y vibraciones de carga, entre otras, que provocan inestabilidad en el sistema [10]. En este contexto, para probar la robustez del sistema de control PD, se considera una fuerza externa al modelo dinámico (7), obteniendo

$$m\ddot{\xi}(t) = \begin{bmatrix} 0 \\ 0 \\ -mg \end{bmatrix} + \mathbf{R}_b^w(t) \begin{bmatrix} 0 \\ 0 \\ u(t) \end{bmatrix} + \begin{bmatrix} f_x \\ f_y \\ f_z \end{bmatrix}, \quad (18)$$

donde se establece $[f_x, f_y, f_z]^T = [0.12, 0.15, 0.1]^T$ N. Las Figuras 12-14 ofrecen una comparación completa entre ambos sistemas de control, donde se observa un mejor seguimiento de la trayectoria con nuestro enfoque en relación con su versión tradicional. La Tabla 4 describe los valores RMSE obtenidos.

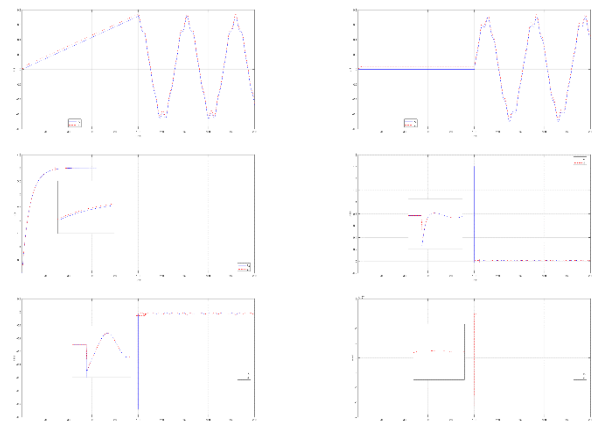


Fig. 5: Seguimiento de trayectoria con perturbaciones para: a) x , b) y , c) z , d) ϕ , e) θ , f) ψ .

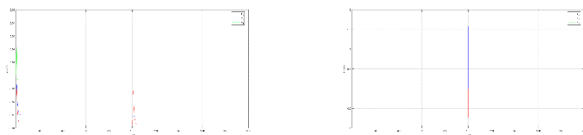


Fig. 6: Error de seguimiento con perturbaciones para: a) x , b) y , c) z , d) ϕ , e) θ , f) ψ .

V. CONCLUSIONES

Este trabajo desarrolla y prueba experimentalmente un sistema de control PD para aplicaciones de seguimiento de trayectoria de un quadrotor. El modelo dinámico del sistema es obtenido a través del formalismo de Newton-Euler. El algoritmo de búsqueda Cuckoo se utiliza para ajustar las ganancias del controlador con el RMSE como una función de costo a minimizar. El desempeño del control PD es puesto a prueba mediante el seguimiento de trayectoria con y sin perturbaciones externas. Simulaciones numéricas confirman que el sistema de control tiene un buen desempeño en términos del RMSE.

REFERENCIAS

[1] Chongju Mai and Anwar Haque. Developing a smart home surveillance system using autonomous drones. In 2023 IEEE 9th World Forum on Internet of Things (WF-IoT), pages 1–6. IEEE, 2023.

[2] Longhao Qian and Hugh HT Liu. Robust stabilization of a quadrotor as a 5g access point using acceleration feedback. In 2023 IEEE 9th World Forum on Internet of Things (WF-IoT), pages 1–6. IEEE, 2023.

[3] Nihal Dalwadi, Dipankar Deb, and Stepan Ozana. Performance evaluation of rw-quadrotor and bi-quadrotor for payload delivery. IET Intelligent Transport Systems, 17(11):2221–2236, 2023.

[4] Ali Roohi Ahangar, Abdolreza Ohadi, and Mohammad A Khosravi. Adaptive sliding mode control of a quadrotor with tilting rotors transporting a cable-suspended payload. In 2023 11th RSI International Conference on Robotics and Mechatronics (ICRoM), pages 431–436. IEEE, 2023.

[5] Paulo Jefferson Dias de Oliveira Evald, Vivian Misaki Aoki, César Bastos da Silva, Dayana Santos Cardoso, Pedro Miranda Pinheiro, Silvia Silva da Costa Botelho, and Paulo Lilles Jorge Drews Junior. A review on quadrotor attitude control strategies. *International Journal of Intelligent Robotics and Applications*, pages 1–21, 2024.

[6] Wei Li, Kai Zhang, ChunPeng Zhang, Qiang Wang, and Yi Zhang. A fuzzy enhanced adaptive PID control algorithm for quadrotor aircraft. *International Journal of Modelling, Identification and Control*, 43(1):26–35, 2023.

[7] Moussa Labbadi, Yassine Boukal, Mohamed Cherkaoui, Moussa Labbadi, Yassine Boukal, and Mohamed Cherkaoui. Quav modeling. *Advanced Robust Nonlinear Control Approaches for Quadrotor Unmanned Aerial Vehicle: Roadmap to Improve Tracking-Trajectory Performance in the Presence of External Disturbances*, pages 19–47, 2022.

[8] Reham H Mohammed. Quadrotor control using advanced control techniques. *International Journal of Image, Graphics & Signal Processing*, 11(2), 2019.

[9] Shaunak Shiralkar, Atharv Bahulekar, and Samidha Jawade. The cuckoo search algorithm: A review. *International Research Journal of Engineering and Technology*, pages 1238–1246, 2022.

[10] Zhaoji Wang and Tong Zhao. Adaptive-based linear active disturbance rejection attitude control for quadrotor with external disturbances. *Transactions of the Institute of Measurement and Control*, 44(2):286–298, 2022.



Reconstrucción de la Marcha Utilizando Unidades de Medición Inerciales en Miembros Inferiores

Barón Casique Jeymar J.¹, Mireles Vázquez Miguel Angel¹, Ramírez Paredes Juan Pablo I.^{*1}, Correa Tomé Fernando E.¹, Torres del Carmen Felipe², Arellano Vázquez Magali³

¹ Universidad de Guanajuato, Departamento de Ingeniería Electrónica, Carretera Salamanca - Valle de Santiago km 3.5 + 1.8 Comunidad de Palo Blanco, 36787, Salamanca, Gto.

² Universidad de Guanajuato, Departamento de Ingeniería Mecánica, Carretera Salamanca - Valle de Santiago km 3.5 + 1.8 Comunidad de Palo Blanco, 36787, Salamanca, Gto.

³ INFOTEC Aguascalientes, Laboratorio de Sistemas Embebidos, Cto. Tecnopolo Sur 112, 20326, Pocitos, Ags.

* Autor de correspondencia: jpi.ramirez@ugto.mx

Resumen. Se presenta un modelo para la predicción de las posiciones de las articulaciones de los miembros inferiores durante la marcha, utilizando información capturada por sensores inerciales, como acelerómetros y giroscopios, colocados en las piernas de una persona. A diferencia de otros enfoques, esta predicción no requiere secuencias de datos en el tiempo, sino que puede operar en instantes aislados. Las aplicaciones potenciales de este modelo incluyen el análisis de la marcha en el sector salud y el desarrollo de dispositivos de asistencia. Los resultados obtenidos demuestran que el modelo desarrollado alcanza un bajo nivel de error, no excediendo los 15 mm, y muestra una alta correlación ($R^2 > 0.9$) entre las predicciones y los datos reales, todo ello con una complejidad computacional moderada.

Palabras clave. *Análisis de la marcha, Estimación y predicción, Redes neuronales.*

I. INTRODUCCIÓN

El análisis de la marcha es fundamental en diversos campos, como la rehabilitación post-lesiones, la prevención de lesiones deportivas, el diseño de prótesis y ortesis, y el desarrollo de dispositivos de asistencia para la movilidad. Un análisis preciso de la marcha humana permite evaluar y tratar a personas con condiciones que afectan su capacidad para caminar. Por ejemplo, es útil para comprender las desviaciones biomecánicas asociadas con la osteoartritis de rodilla. Tradicionalmente, este análisis se basaba en la observación visual, complementada más tarde con grabaciones de video para una evaluación detallada (Ahmadi et al., 2016).

Aunque los sistemas tradicionales de análisis de la marcha, como aquellos que emplean captura de movimiento óptica (e.g., Vicon y OptiTrack) o cintas de correr instrumentadas con plataformas de fuerza (e.g., Noraxon), ofrecen alta precisión, a menudo son costosos, complejos y limitados a entornos controlados de laboratorio (Feigl et al., 2020). Estas restricciones los hacen menos adecuados para evaluar la marcha en entornos naturales y no controlados. Además, los sistemas ópticos como Vicon suelen requerir un procesamiento posterior complejo y personal especializado para la recolección y análisis de datos. Aunque los sistemas basados en marcadores son precisos, pueden presentar errores debido a la oclusión de los marcadores y a los desafíos en la medición de rotaciones en el plano transversal (Ahmadi et al., 2016).



La necesidad de llevar el análisis de la marcha más allá del laboratorio ha impulsado la investigación en alternativas portátiles y accesibles. Las Unidades de Medición Inercial (IMUs, por sus siglas en inglés), basadas en la tecnología de Sistemas Micro-Electromecánicos, han surgido como una solución prometedora para capturar datos de marcha en entornos reales. Estos sensores pequeños y ligeros, que típicamente contienen acelerómetros y giroscopios de tres ejes, pueden ser colocados en el cuerpo para medir el movimiento en condiciones reales. Las IMUs han ganado popularidad en el análisis de la marcha, gracias a su asequibilidad y capacidad para monitorear la marcha en entornos no controlados.

Este trabajo presenta una prueba de concepto de un sistema para la recopilación de datos de marcha utilizando IMUs colocadas en las extremidades inferiores, con el objetivo de ofrecer una alternativa portátil y práctica a los sistemas tradicionales de análisis de la marcha. Se reconoce, no obstante, las limitaciones inherentes de las IMUs, particularmente su susceptibilidad a la deriva en la precisión con el tiempo, un desafío común en la detección inercial. Para mitigar este problema, se propone el uso de aprendizaje automático y un modelo basado en redes neuronales para procesar los datos recopilados. Este enfoque se alinea con los avances recientes en el análisis de la marcha, donde el aprendizaje automático, especialmente a través de arquitecturas como los Autoencoders Siameses de Reducción de Ruido, ha mostrado ser eficaz en la reconstrucción precisa de movimientos de marcha a partir de datos ruidosos de IMU, considerando factores como la variabilidad en la vestimenta y el tiempo (Sheng & Li, 2020). Al incorporar algoritmos que tengan en cuenta la naturaleza cíclica de la marcha y las restricciones biomecánicas humanas, como se ha demostrado con la integración de redes neuronales y Primitivas de Movimiento Kernelizadas en investigaciones recientes (Zou et al., 2021), este sistema busca proporcionar un análisis fiable de la marcha en una variedad de entornos reales.

II. MATERIALES Y MÉTODOS

Los modelos cinemáticos que se utilizan para analizar la marcha humana suelen modelar los segmentos de los miembros inferiores como eslabones rígidos, mientras

que las articulaciones son representadas como uniones revolutas. En el presente estudio, hemos adoptado un enfoque bidimensional para el modelado cinemático, el cual se encuentra ilustrado en la Fig. 1. En esta figura, se define el eje X como una dirección horizontal, que se extiende de izquierda a derecha, mientras que el eje Y se dispone de manera vertical, ascendiendo desde la parte inferior hacia la parte superior del plano. Los puntos de interés dentro de este modelo, que han sido designados como P1 a P5, corresponden a las articulaciones clave cuya posición debe ser determinada con precisión durante la marcha. Partiendo del supuesto de que las longitudes de los eslabones entre estas articulaciones se mantienen constantes, es posible reconstruir la posición de todos estos puntos de interés siempre que se conozca la posición inicial de uno de ellos y los ángulos de cada eslabón en relación con la vertical.

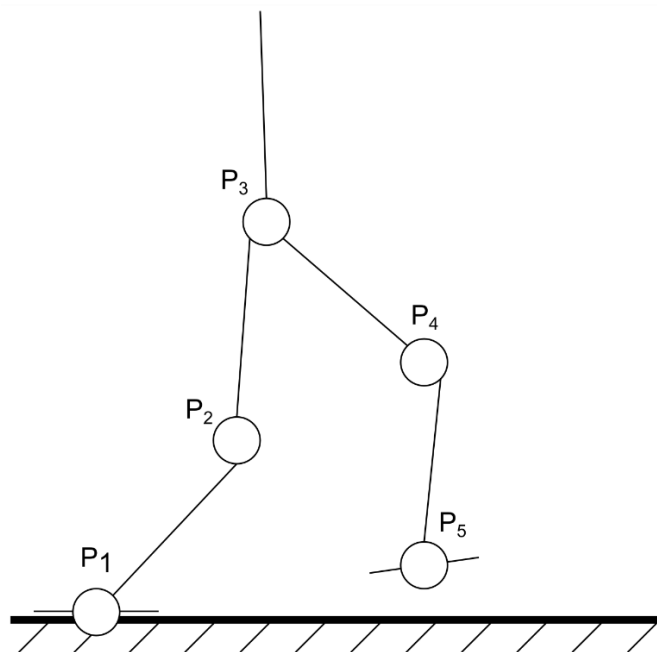


Fig. 1. Nomenclatura de los marcadores y secciones de las piernas.

En este trabajo, proponemos el desarrollo de un dispositivo portátil, ligero y fácil de usar, que en la literatura especializada se conoce como "wearable". La función principal de este dispositivo es detectar los ángulos formados por los eslabones en el modelo cinemático descrito anteriormente. Para lograr esta tarea, el dispositivo está equipado con sensores inerciales, tales como acelerómetros y giroscopios de tres ejes, que son capaces de capturar los datos



necesarios para el análisis de la marcha en tiempo real. Los acelerómetros se encargan de medir la aceleración lineal en las tres direcciones espaciales, mientras que los giroscopios miden las velocidades angulares, permitiendo así una estimación precisa de los ángulos relativos entre los segmentos corporales.

Como parte de una demostración inicial del concepto propuesto, se llevó a cabo un estudio que incluyó simulaciones computacionales y un análisis detallado de datos obtenidos antes de la implementación física del dispositivo portátil. El objetivo de esta fase preliminar fue confirmar la viabilidad técnica de la propuesta y establecer una base de referencia para futuras pruebas experimentales. Para la generación de los datos de referencia sobre la marcha humana, recurrimos al estudio realizado por (Schreiber y Moissenet, 2019), en el cual un grupo de voluntarios permitió la captura de datos detallados de su movimiento al caminar mediante un avanzado sistema de seguimiento óptico. En dicho estudio, se colocaron múltiples marcadores en los miembros inferiores de los sujetos mientras caminaban en una banda sinfín, permitiendo la obtención de un conjunto de datos exhaustivo que documenta con precisión el movimiento de las articulaciones durante la marcha.

En nuestro trabajo, seleccionamos los marcadores más relevantes de los datos capturados por (Schreiber y Moissenet, 2019) y los designamos como los puntos P1 a P5 dentro de nuestro modelo cinemático. A partir de las secuencias de marcadores (P1, P2), (P2, P3), y así sucesivamente, se calcularon los ángulos de los eslabones correspondientes, lo que permitió una reconstrucción precisa de las posiciones articulares en cada instante de la marcha.

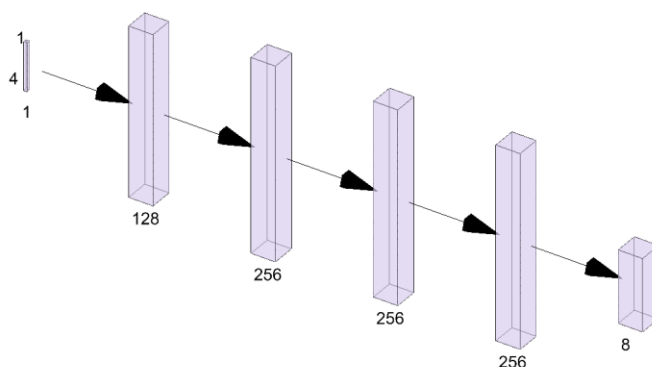


Fig. 2. Arquitectura de red neuronal utilizada, vista esquemática.

La reconstrucción de la marcha, es decir, el proceso mediante el cual se calculan las posiciones de los marcadores a partir de los ángulos de los eslabones, se realizó utilizando un modelo de red neuronal multicapa. Este modelo está diseñado para recibir los ángulos de los segmentos corporales como entradas y producir las posiciones articulares como salidas. La arquitectura de la red neuronal se presenta en la Fig. 2, donde se detallan las capas densamente conectadas que conforman el modelo, junto con las configuraciones específicas de cada capa y las funciones de activación empleadas, las cuales se resumen en la Tab. 1.

En todos los casos, se implementó la técnica de "drop-out" con una probabilidad de 0.25 para reducir el riesgo de sobreentrenamiento del modelo, lo que permite una mejor generalización cuando se enfrenta a nuevos datos. Durante el desarrollo del modelo, se realizaron pruebas preliminares utilizando arquitecturas más complejas, que incluían un mayor número de capas, pero no se observó una mejora significativa en el desempeño del modelo. Como resultado, el número de capas se limitó a cuatro, priorizando una arquitectura más sencilla y eficiente. El proceso de entrenamiento del modelo se llevó a cabo utilizando el optimizador Adam, el cual es ampliamente reconocido por su capacidad para manejar problemas de optimización no convexos. La tasa de aprendizaje se estableció en 0.001, y se empleó la función de pérdida basada en el error cuadrático medio, que es adecuada para minimizar las desviaciones entre las predicciones del modelo y los valores reales.



Tab. 1. Capas de la red neuronal artificial.

Núm. de capa	Núm. de neuronas	Tipo de activación
1	128	ReLU
2	256	ReLU
3	256	ReLU
4	256	ReLU

III. RESULTADOS

Para evaluar el desempeño del modelo desarrollado, se seleccionó una muestra específica de la marcha de un individuo del conjunto de datos proporcionado por (Schreiber y Moissenet, 2019). La evaluación consistió en comparar las posiciones de los marcadores, tal como se encuentran en la base de datos de referencia, con las posiciones predichas por el modelo propuesto. Las entradas para el modelo consistieron en los ángulos de los segmentos correspondientes a las piernas, que fueron utilizados para calcular las posiciones articulares a lo largo del tiempo. Como se detalla en la Tab. 2, se realizaron cálculos de diversas métricas para cuantificar la precisión del modelo en función de esta comparación.

En la Tab. 2 se muestra que las posiciones de los marcadores se reportan en milímetros, y el error absoluto promedio observado no excede los 13 mm por marcador. Este resultado indica una alta precisión en las predicciones realizadas por el modelo, especialmente considerando que el desplazamiento de la mayoría de los marcadores en la porción de la marcha analizada supera los 400 mm. Esta diferencia en la escala del desplazamiento y el error observado refleja la capacidad del modelo para manejar variaciones significativas en los datos de entrada y aun así proporcionar resultados precisos.

Tab. 2: Métricas de desempeño de la prueba.

Métrica	Valor
Error absoluto promedio	11.0439 (mm)
Error cuadrático promedio	316.4463 (mm ²)
Error RMS	17.7889 (mm)
Coefficiente R2	0.9125

Para proporcionar una visualización clara de cómo las predicciones del modelo se asemejan o se desvían de los

datos reales, se incluyen varias gráficas comparativas. Las Fig. 3 y Fig. 4 presentan el movimiento del marcador P3, que está asociado con la cadera del individuo, en una vista lateral. Estas gráficas permiten observar cómo el error se mantiene bajo a lo largo de casi toda la trayectoria de la marcha, aunque se nota un incremento en el error hacia el final del segmento analizado, particularmente en las últimas 20 muestras. En la Fig. 4, el máximo error registrado para el marcador P3 es de apenas 15 mm, lo que refleja un buen desempeño del modelo en la estimación de la posición de este marcador en particular.

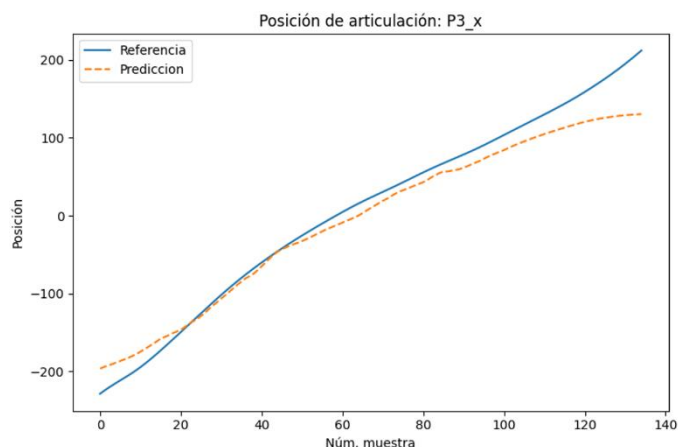


Fig. 3: Cambio en la posición del marcador P3 en el eje X, con respecto al tiempo.

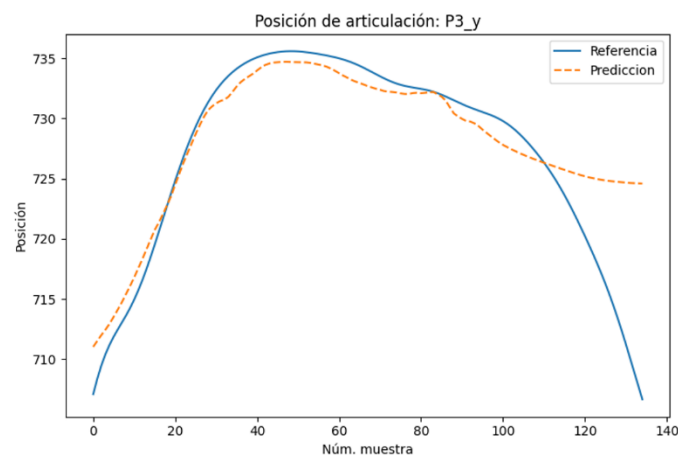


Fig. 4: Cambio en la posición del marcador P3 en el eje Y, con respecto al tiempo.

El marcador P3, mencionado previamente, juega un papel importante en la descripción del desplazamiento total del cuerpo, dado que se mueve en sincronía con el torso del individuo. Para los demás marcadores que representan las posiciones de las rodillas y tobillos, las



Fig. 5 y 6 ilustran la evolución temporal de la posición del marcador P5, que corresponde a un tobillo. Estas gráficas muestran que, en el caso del marcador P5, el nivel de error es significativamente menor en comparación con el marcador P3, lo que sugiere que el modelo ofrece una mayor precisión en la predicción de las posiciones de los tobillos.

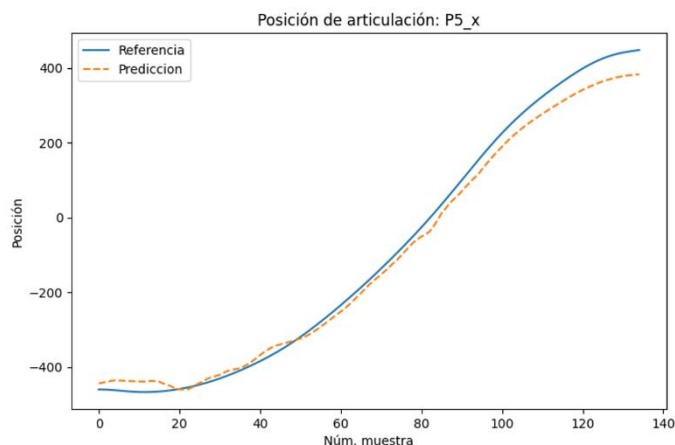


Fig. 5: Cambio en la posición del marcador P5 en el eje X, con respecto al tiempo.

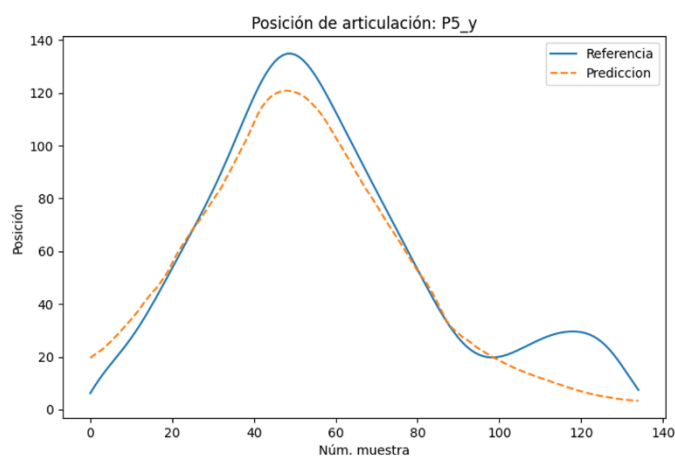


Fig. 6: Cambio en la posición del marcador P5 en el eje Y, con respecto al tiempo.

Aún cuando los datos utilizados provienen de un entorno controlado, los resultados muestran que la predicción de los marcadores puede dar buenos resultados sin utilizar secuencias de datos en el tiempo, sino únicamente los datos que corresponden a cada instante para cada muestra, lo que permite anticipar que la técnica propuesta podrá tener aplicación en entornos de exteriores o en condiciones distintas a las de laboratorio.

IV. CONCLUSIONES

El modelo desarrollado en este estudio ha demostrado ser capaz de predecir con precisión las posiciones de varios marcadores de interés en las piernas, utilizando únicamente como entrada los ángulos de los muslos y la pierna inferior. Los resultados indican que el error promedio en una marcha típica puede ser inferior a 2 cm, lo que sugiere que la aplicación de una red neuronal de tipo perceptrón multicapa es una solución efectiva para el análisis de la marcha y el seguimiento del desplazamiento de personas. Dentro de las limitaciones del esquema propuesto podemos mencionar que los sensores inerciales de bajo costo requieren filtrado de su ruido inherente, por lo que el algoritmo de aprendizaje de máquina se deberá complementar con técnicas de estimación. Como siguiente etapa de este proyecto, se planea ampliar el conjunto de datos, recolectando información de voluntarios que utilicen nuestro dispositivo equipado con unidades de medición inerciales para la captura de datos.

V. AGRADECIMIENTOS

Agradecemos el apoyo obtenido para este proyecto a través de la Convocatoria Institucional de Investigación Científica de la Universidad de Guanajuato, como parte del convenio 121/2024. J.J. Barón Casique agradece al Conahcyt el apoyo recibido con la Beca Nacional CVU 1235534.

REFERENCIAS

- Ahmadi, A., Destelle, F., Unzueta, L., Monaghan, D. S., Teresa Linaza, M., Moran, K., & O'Connor, N. E. (2016). 3D Human Gait Reconstruction and Monitoring Using Body-Worn Inertial Sensors and Kinematic Modeling. *IEEE Sensors Journal*, 16(24), 8823-8831. IEEE Sensors Journal. <https://doi.org/10.1109/JSEN.2016.2593011>
- Feigl, T., Gruner, L., Mutschler, C., & Roth, D. (2020). Real-Time Gait Reconstruction For Virtual Reality Using a Single Sensor. *2020 IEEE International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct)*, 84-89. <https://doi.org/10.1109/ISMAR-Adjunct51615.2020.00037>



- Schreiber, C., & Moissenet, F. (2019). A multimodal dataset of human gait at different walking speeds established on injury-free adult participants. *Scientific Data*, 6(1), 111. <https://doi.org/10.1038/s41597-019-0124-4>
- Sheng, W., & Li, X. (2020). Siamese denoising autoencoders for joints trajectories reconstruction and robust gait recognition. *Neurocomputing*, 395, 86-94. <https://doi.org/10.1016/j.neucom.2020.01.098>
- Zou, C., Huang, R., Cheng, H., & Qiu, J. (2021). Learning Gait Models With Varying Walking Speeds. *IEEE Robotics and Automation Letters*, 6(1), 183-190. IEEE Robotics and Automation Letters. <https://doi.org/10.1109/LRA.2020.3006818>



Trazado de Trayectorias Libres de Colisiones con Optimización por Enjambre de Partículas

**Duarte Rostro Axel J.¹, Zaragoza Hernández Francisco A.¹, Hernández Belmonte Uriel H.²,
Ramírez Paredes Juan Pablo I. ^{*1}**

¹ Universidad de Guanajuato, Departamento de Ingeniería Electrónica, Carretera Salamanca - Valle de Santiago km 3.5 + 1.8 Comunidad de, Palo Blanco, 36787 Salamanca, Gto.

² Universidad de Guanajuato, Departamento de Arte y Empresa, Carretera Salamanca - Valle de Santiago km 3.5 + 1.8 Comunidad de, Palo Blanco, 36787 Salamanca, Gto.

* Autor de correspondencia: jpi.ramirez@ugto.mx

Resumen. En robótica, es frecuente la necesidad de trazar rutas de costo mínimo en espacios tridimensionales, incorporando simultáneamente la evasión de obstáculos. Presentamos una aplicación del algoritmo de optimización enjambre de partículas (PSO) para suavizar una ruta tridimensional generada por el algoritmo A*, la cual se genera entre dos puntos de un entorno conocido, incluyendo obstáculos. Esta aplicación funciona como un filtro para un mapa donde los obstáculos, posición inicial y destino son conocidos. Nuestros resultados muestran que, en el caso general, las rutas generadas por A* pueden ser mejoradas significativamente al aplicar el paso de optimización con PSO, sin requerir un aumento en la resolución de la discretización del espacio necesaria para la ejecución de A*.

Palabras clave. *Optimización, enjambre de partículas, ruta 3D, optimización de rutas, robótica móvil.*

I. INTRODUCCIÓN

La planeación eficiente de rutas 3D se ha convertido en un área crítica para la investigación actualmente en el campo de la robótica. Esta importancia reside en el manejo eficiente de diversos robots como lo son los vehículos autónomos, drones y brazos robóticos utilizados en manufactura y cirugía.

Una navegación eficiente depende de una ruta confiable. Para los vehículos aéreos no tripulados, por ejemplo, esto es indispensable dentro de entornos urbanos, para poder realizar evasión de obstáculos y llevar a cabo tareas como la vigilancia y entrega de paquetes.

Los avances en la planeación de rutas 3D guían las capacidades de los sistemas robóticos modernos, permitiéndoles operar de forma autónoma y eficiente en entornos complejos.

El continuo desarrollo de algoritmos es esencial para extender los límites de lo que puede hacer la robótica, esta innovación aborda las aplicaciones posibles dentro de diversos dominios. Algunos de estos algoritmos y estrategias han obtenido una amplia relevancia en la generación de rutas en el espacio tridimensional como lo son la Exploración Rápida por Árboles Aleatorios (RRT por sus siglas en inglés) en el cuál, Kuffner y LaValle proponen que, partiendo de un nodo se generan puntos aleatorios dentro de un espacio de búsqueda realizando conexiones entre los nodos más cercanos. (Kuffner & LaValle, 2000), Seo y Kim, a través de la geometría



diferencial utilizan la información de un vector de línea, otro de visión y uno de velocidad relativa para generar un comando de guía en tiempo real (Seo et al., 2013). También podemos mencionar la propuesta de Kamel y su equipo sobre un modelo no lineal para control predictivo en la navegación (Kamel et al., 2015).

La optimización es clave en esta área de la investigación. Una ruta optimizada asegura un desplazamiento más corto en distancia y/o tiempo, reduciendo así la energía y diversos recursos indispensables.

Algunos métodos de optimización se han inspirado en procesos biológicos. Ejemplos de estos métodos son la Robótica de Enjambre (Alias et al., 2017), el cual se inspira en el comportamiento colectivo de algunos insectos como lo son las hormigas o abejas y la búsqueda “Cuckoo”, inspirada en el parasitismo de cría de algunas especies de ave cuco (Yang & Deb, 2009).

La forma en que resolvemos este problema en nuestro trabajo permite aplicarse en sistemas subacuáticos. Un ejemplo de estos sistemas lo presentan Wang y colaboradores, en una aplicación inspirada en el algoritmo de Búsqueda de Alimentos Bacteriana para planificar rutas de Vehículos Subacuáticos no Tripulados, en 3D (Wang et al., 2022)

La combinación de algoritmos bio inspirados y algoritmos de optimización se ha convertido en una práctica común para obtener buenos resultados. Ejemplo de ello es la combinación de optimización por colonia de hormigas con algoritmo genético en el trabajo de Hu (Hu et al., 2023) o el algoritmo híbrido de murciélago con evolución diferencial que nos presenta Iztok Fister (Fister Jr et al., 2013), por este motivo es que presentamos una combinación de un algoritmo común de generación de rutas A* para crear la ruta base y optimizarla a través de PSO.

II. METODOLOGÍA

El espacio 3D está limitado por valores específicos en los ejes X, Y, y Z. Así, vemos el escenario como un conjunto de cubos con aristas que interpretará el algoritmo A*

como un grafo. Los obstáculos se definen por las posiciones de sus centros y su radio. Estos obstáculos se representan en el espacio 3D como esferas.

Dado un camino definido como la secuencia de puntos $P = \{p_1, p_2, \dots, p_n\}$ en el espacio tridimensional, donde cada punto $p_i = [x_i \ y_i \ z_i]^T$, la función de costo

$$f(P) = \sum_{i=1}^{n-1} \|p_{i+1} - p_i\| + \sum_{i=1}^n C(p_i) \quad (1)$$

que toma en cuenta la distancia y una penalización por colisiones, se somete a un proceso de optimización. En dicha función, $\|p_{i+1} - p_i\|$ representa la distancia euclidiana entre los puntos p_i y p_{i+1} :

$$\|p_{i+1} - p_i\| = \sqrt{(p_{i+1} - p_i)^T (p_{i+1} - p_i)} \quad (2)$$

Sea $\Omega \subset \mathbb{R}^3$ el conjunto de los puntos en el espacio que pertenecen a los obstáculos. Entonces $C(p_i)$ es una función de costo de colisión definida como:

$$C(p_i) = \begin{cases} 0 & \text{si } p_i \in \Omega \\ +\infty & \text{si } p_i \notin \Omega \end{cases} \quad (3)$$

Propusimos el algoritmo A* como una solución inicial al problema de generación de rutas debido a que este algoritmo combina aspectos de la búsqueda de costo uniforme y la búsqueda heurística para encontrar de manera eficiente el camino de costo mínimo desde un nodo de inicio hasta un nodo objetivo.

En el algoritmo A*, establecemos un punto de inicio predefinido y el algoritmo realiza la búsqueda para alcanzar el punto objetivo. Mantiene dos conjuntos de nodos: los que deben ser evaluados y los ya evaluados. Este algoritmo utiliza la distancia euclidiana para estimar la distancia entre los nodos analizados y el objetivo. Para cada nodo, calcula el costo para moverse a sus vecinos y actualiza las puntuaciones. En A* se agrega a los vecinos de un nodo al conjunto abierto si no están en colisión con obstáculos y tienen un costo menor.



La optimización de enjambre de partículas (PSO por sus siglas en inglés) es un algoritmo metaheurístico de optimización que simula el comportamiento colectivo de enjambres o grupos de organismos, como aves o peces, al buscar una solución óptima a un problema. Fue propuesto por Kennedy y Eberhart en 1995 y se basa en la idea de que las partículas de un grupo comparten información sobre las mejores soluciones que han encontrado, lo que permite una búsqueda cooperativa en el espacio de soluciones (Kennedy & Eberhart, 1995). El algoritmo comienza generando un conjunto de partículas distribuidas aleatoriamente en un espacio de búsqueda. Cada partícula representa una posible solución al problema y tiene como propiedades su posición (solución que se está evaluando) y su velocidad (dirección y magnitud de su movimiento en el espacio de búsqueda). En cada iteración, se evalúa la calidad de la solución asociada a la posición actual de cada partícula a través de una función de aptitud. Esta función determina qué tan buena es la solución propuesta por la partícula.

Cada partícula recuerda su mejor posición “personal”, es decir, la posición en la que encontró la mejor solución. Además, todo el grupo de partículas comparte información sobre la mejor solución global encontrada por cualquiera de las partículas en la población. La velocidad y posición de cada partícula se ajusta teniendo en cuenta el componente personal, el cual impulsa a la partícula a moverse hacia su mejor posición personal y el componente social, es el que impulsa a la partícula a moverse hacia la mejor posición global encontrada por el enjambre; la combinación de estos dos factores permite la exploración de nuevas áreas del espacio de búsqueda mientras convergen hacia las mejores soluciones ya identificadas. El proceso de evaluación y actualización se repite durante un número determinado de iteraciones o hasta que se cumple un criterio de parada, en nuestro caso, es la minimización del costo de la ruta.

Nuestra implementación de PSO inicializa un enjambre de partículas, cada una representando una solución potencial (camino), este camino se toma de copias del camino inicial obtenido con A*. Las partículas se mueven a través del espacio de soluciones basándose en sus

velocidades, posiciones de mejor desempeño personal, y la mejor posición global.

La aptitud de cada partícula se evalúa utilizando y calculando la distancia del camino y, posteriormente, verifica posibles colisiones. Las velocidades y posiciones de las partículas se actualizan iterativamente para encontrar un camino óptimo que evite los obstáculos. El algoritmo tiene como objetivo encontrar un camino que minimice la distancia total mientras evita colisiones.

Optimización Global:

El problema que se resuelve es encontrar un camino P^* que optimice $f(P)$, es decir:

$$P^* = \arg \min_P f(P)$$

(4)

III. EXPERIMENTOS Y RESULTADOS

A continuación, se presentan los resultados obtenidos en diferentes pruebas realizadas a través de Python para mostrar el correcto funcionamiento del algoritmo y la comparación entre las rutas. El entorno, incluidos los límites y los obstáculos esféricos, se visualiza en un gráfico 3D.

El espacio de búsqueda para PSO en nuestros experimentos es $\mathbb{R}^{3 \times n}$, donde n es el número de puntos que componen la ruta tras la ejecución de A*. El factor de inercia utilizado es 0.5 y se utilizaron 20 partículas en la optimización por PSO.

En la Fig. 1 podemos observar el caso especial en donde la ruta ideal sería una línea recta entre los puntos de inicio y la meta, debido a la ausencia de obstáculos intermedios, por lo que en este escenario no sería útil la optimización por PSO.

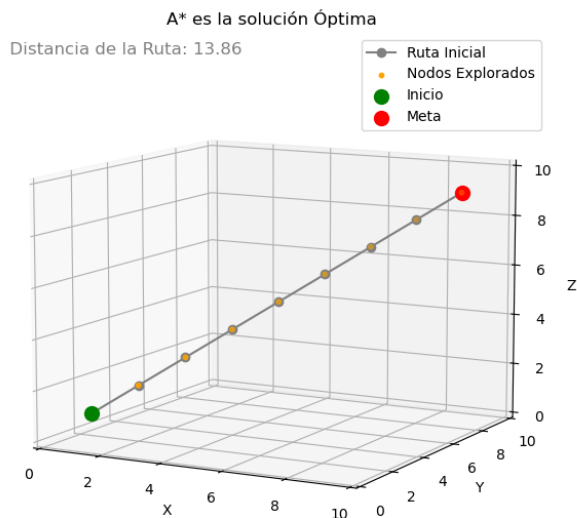


Fig. 1. Representación de ruta cuando el A generado es la ruta óptima.*

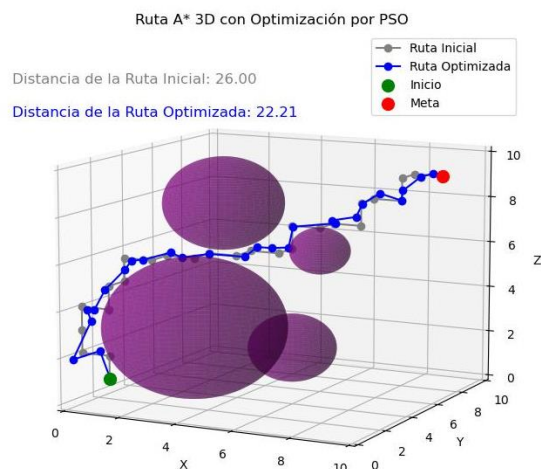


Fig. 2. Optimización con punto inicial (1,1,1), punto final (9,9,9). Podemos observar la ruta generada por el A comparando con su optimización y la correcta evasión de los obstáculos.*

La Fig. 2 representa un escenario normal en el que se presentan obstáculos obstruyendo la ruta, demostrando una mejora con la optimización y la presencia de obstáculos no representa problema alguno para ninguno de los 2 casos (la generación de la ruta inicial y su optimización).

Tanto el camino inicial (encontrado por A*) como el camino optimizado (refinado por PSO) se representan en

la gráfica. El punto de inicio se marca en verde, el punto objetivo en rojo, el camino inicial en gris, el camino optimizado en azul y en naranja podemos visualizar los nodos que fueron evaluados por el A*.

Se calcula y muestra la distancia total tanto del camino inicial como del camino optimizado. Esto proporciona una comparación clara de la efectividad del algoritmo PSO en la optimización del camino.

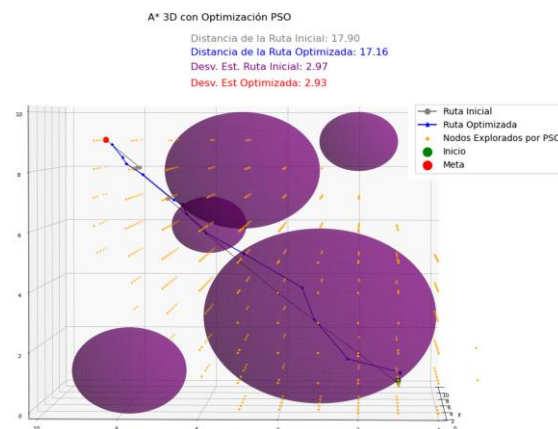


Fig. 3. Optimización con punto inicial {1,1,1}, punto final {9,9,9} mostrando la desviación estándar y datos de apoyo.

Los nodos visitados son herramientas visuales que permiten una mejor observación del entorno y nos ayuda a comprender tanto la mejora de la ruta como la forma en que nuestro algoritmo evalúa los posibles espacios a visitar de forma correcta, por lo que, junto a la desviación estándar, son mostradas en la Fig. 4 en un escenario de prueba.

Se generaron pruebas en diferentes escenarios para evaluar el desempeño promedio del algoritmo, ya que, al ser un método estocástico, debe someterse varias veces en entornos iguales. La Tab. 1 describe la representación de algunos de los escenarios en casos especiales puestos a prueba.



Tab. 1. Descripción de 6 escenarios para realizar pruebas al algoritmo.

Núm. Escenario	Inicio x, y, z	Meta x, y, z	Obstáculos (x, y, z y radio)
1	1, 2, 4	8, 8, 8	(3, 1, 2, r = 1) (4, 5, 5, r = 2) (7, 2, 6, r = 1)
2	0, 0, 0	10, 10, 10	(2, 1, 2, r = 1.5) (4, 5, 8, r = 2) (7, 2, 9, r = 1)
3	0, 0, 0	10, 10, 10	(4, 8, 1, r = 1.5) (3, 5, 8, r = 2) (7, 2, 9, r = 3) (6, 6, 6, r = 1)
4	10, 0, 0	0, 10, 10	(4, 8, 1, r = 1.5) (3, 5, 8, r = 2) (7, 2, 9, r = 3) (6, 6, 6, r = 1)
5	0, 10, 0	10, 0, 10	(4, 8, 1, r = 1.5) (3, 5, 8, r = 2) (7, 2, 9, r = 3) (6, 6, 6, r = 1)
6	10, 10, 0	0, 0, 10	(4, 8, 1, r = 1.5) (3, 5, 8, r = 2) (7, 2, 9, r = 3) (6, 6, 6, r = 1)

Los resultados obtenidos en los escenarios propuestos muestran ser satisfactorios al disminuir el costo de las rutas luego de ser optimizadas, a continuación, en la Tab. 2, se presentan las pruebas realizadas 15 veces para cada uno de los casos.

Tab. 2. Resultados obtenidos en 15 pruebas por cada uno de los 6 escenarios. Núm. Escenario	Costo Inicial con A*	Costo después de PSO Promedio	Reducción de costo (%)
1	17	14.4107	15.231 %
2	30	26.2640	12.453 %
3	30	29.2086	2.638 %
4	30	27.2263	9.249 %
5	30	27.7430	7.524 %
6	30	26.7080	10.973 %

Como se puede observar, el algoritmo A* es útil para brindar una solución rápida a estos problemas. Si analizamos los gráficos podemos ver un camino suave una vez aplicamos el PSO; minimizando así el costo y haciendo que sea más fácil la realización de la ruta, permitiendo también, la creación de una trayectoria más sencilla.

La desviación estándar es utilizada como una herramienta para evaluar las rutas, ya que nos permite analizar la suavidad que tienen, lo cual es un problema bastante importante para resolver en el campo de la navegación con drones.

IV. CONCLUSIONES

El algoritmo PSO en esta aplicación refina el camino inicial ajustándolo de manera iterativa, aprovecha una población de caminos candidatos (partículas) y los actualiza en función de su desempeño y el de otros. El proceso continúa hasta que el enjambre encuentra un camino optimizado que minimiza la distancia mientras evita obstáculos, demostrando ser una técnica de optimización efectiva para problemas de búsqueda de caminos.

Los resultados muestran que, en general, la optimización con PSO posterior a la generación de rutas con A* resulta en reducciones importantes de costo. Una manera de mejorar el resultado de A* sin PSO es haciendo una discretización más fina del espacio donde se trazará la ruta, pero esto eleva considerablemente el costo computacional, por lo que la optimización con PSO es una buena solución en el caso general.

AGRADECIMIENTOS

A. J. Duarte Rostro y F. A. Zaragoza Hernández agradecen al Conahcyt por el apoyo a través de las Becas Nacionales con CVU 1289219 y 1322364, respectivamente.

REFERENCIAS



- Alias, N. A., Huq, M. S., Ibrahim, B. S. K. K., & Omar, R. (2017). The Efficacy of State of the Art Overground Gait Rehabilitation Robotics: A Bird's Eye View. *Procedia Computer Science*, 105, 365-370.
<https://doi.org/10.1016/j.procs.2017.01.235>
- Fister Jr, I., Fister, D., & Yang, X.-S. (2013). A hybrid bat algorithm. *CoRR*, abs/1303.6310.
<http://arxiv.org/abs/1303.6310>
- Hu, J., Er, M. J., Liu, T., & Wang, S. (2023). A Hybrid Algorithm Based on Ant Colony and Genetic Algorithm for AUV Path Planning. *2023 2nd Conference on Fully Actuated System Theory and Applications (CFASTA)*, 888-893.
<https://doi.org/10.1109/CFASTA57821.2023.10243252>
- Kamel, M., Alexis, K., Achtelik, M., & Siegwart, R. (2015). Fast nonlinear model predictive control for multicopter attitude tracking on SO(3). *2015 IEEE Conference on Control Applications (CCA)*, 1160-1166.
<https://doi.org/10.1109/CCA.2015.7320769>
- Kennedy, J., & Eberhart, R. (1995). Particle swarm optimization. *Proceedings of ICNN'95 - International Conference on Neural Networks*, 4, 1942-1948 vol.4.
<https://doi.org/10.1109/ICNN.1995.488968>
- Kuffner, J. J., & LaValle, S. M. (2000). RRT-connect: An efficient approach to single-query path planning. *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065)*, 2, 995-1001 vol.2.
<https://doi.org/10.1109/ROBOT.2000.844730>
- Seo, J., Kim, Y., & Tsourdos, A. (2013). Differential Geometry based Collision Avoidance Guidance for Multiple UAVs. *IFAC Proceedings Volumes*, 46(19), 113-118.
<https://doi.org/10.3182/20130902-5-DE-2040.00061>
- Wang, Z., Chen, Y., Wang, X., & Zhao, J. (2022). A 3D AUV Path Planner Based on Modified Bacterial Foraging Algorithm. En M. Wu, Y. Niu, M. Gu, & J. Cheng (Eds.), *Proceedings of 2021 International Conference on Autonomous Unmanned Systems (ICAUS 2021)* (pp. 3211-3221). Springer Singapore.
- Yang, X.-S., & Deb, S. (2009). Cuckoo Search via Lévy flights. *2009 World Congress on Nature & Biologically Inspired Computing (NaBIC)*, 210-214.
<https://doi.org/10.1109/NABIC.2009.5393690>



Exploración Robótica Autónoma con Vehículos Terrestres no Tripulados Aplicando Minimización de Entropía

Rojas-Andrade Miguel A.¹, Ramírez-Paredes Juan Pablo I.^{1*}, Hernández-Belmonte Uriel H.²

¹ Universidad de Guanajuato, Departamento Ingeniería Electrónica, Carretera Salamanca - Valle de Santiago km 3.5 + 1.8 Comunidad de Palo Blanco, Salamanca, Guanajuato

² Universidad de Guanajuato, Departamento de Arte y Empresa, Carretera Salamanca - Valle de Santiago km 3.5 + 1.8 Comunidad de Palo Blanco, Salamanca, Guanajuato

* Autor de correspondencia: jpi.ramirez@ugto.mx

Resumen. Este estudio explora y compara dos estrategias de mapeo de entorno de interiores utilizando un robot móvil terrestre Pioneer 3-AT con un algoritmo de mapeo y localización simultáneas. El objetivo principal es evaluar la eficiencia y efectividad del mapeo en términos de la cobertura del mapa y la rapidez en completar el proceso de mapeo. La primera estrategia es un enfoque dirigido por la entropía de Shannon, donde el robot se mueve hacia el punto en la frontera con la mayor entropía, para maximizar la reducción de incertidumbre en el mapa. La segunda estrategia es un método aleatorio, donde se selecciona y se dirige al robot hacia puntos de la frontera sin ninguna consideración específica, basándose únicamente en la aleatoriedad. Nuestros resultados indican que guiar la exploración mediante el criterio de reducción de entropía reduce significativamente el tiempo requerido para cubrir completamente el mapa, obteniendo una reducción promedio del tiempo de un 35 % en comparación con el método de frontera aleatoria.

Palabras clave. *Entropía, frontera, mapeo, navegación autónoma.*

I. INTRODUCCIÓN

El mapeo de entornos interiores utilizando robots móviles es un desafío fundamental en la robótica móvil, esencial para la implementación efectiva de la autonomía robótica. Esta tarea implica la integración de múltiples disciplinas, incluyendo la planificación de rutas, el uso de sensores robustos para la adquisición de datos ambientales y la implementación de algoritmos de localización y mapeo simultáneo (SLAM). Uno de los métodos pioneros en este campo fue desarrollado por Yamauchi (Yamauchi, 1997), quien introdujo el enfoque de "frontera cercana", que guía al robot hacia la frontera más cercana no explorada para maximizar la eficiencia del mapeo. Además, el trabajo de (Elfes, 1989) ha sido crucial para entender cómo representar un entorno

mediante celdas de ocupación. En su método, cada celda del mapa se clasifica como ocupada, libre o desconocida, asignando a cada una un valor numérico que refleja esta categorización. Este enfoque permite a los robots no solo navegar por el entorno sino también interactuar de manera más intuitiva con su entorno percibido. Con esta abstracción del entorno, es posible implementar una estrategia de exploración que maximice la información obtenida del mapa (Carrillo et al., 2015). Nuestro grupo ha explorado también el uso de asignaciones de fronteras para múltiples robots, con base en distancias y divisiones del espacio (Lopez-Perez et al., 2018).

En este estudio, extendemos estos fundamentos explorando cómo la selección de puntos fronterizos



afecta la eficiencia y efectividad del mapeo en entornos interiores. Comparamos un método dirigido por entropía, que optimiza la información ganada al dirigirse al punto de mayor incertidumbre en la frontera, contra un enfoque aleatorio que selecciona un punto fronterizo sin una estrategia definida. Nuestro objetivo es evaluar cuál de estos métodos proporciona la cobertura más completa del mapa y cómo afectan la autonomía del robot en cuanto a decisiones de navegación y tiempo de mapeo. Este análisis proporciona una visión más profunda de las técnicas de planificación estratégica en robótica móvil y su impacto en la eficiencia del mapeo.

II. METODOLOGÍA

La implementación de los algoritmos de mapeo autónomo en nuestro sistema se divide en dos metodologías distintas para la selección de fronteras: una basada en la maximización de entropía y otra en la selección aleatoria. Ambos enfoques proceden de manera idéntica hasta el punto de decisión sobre la frontera de exploración del robot. Estos métodos utilizan el filtro de Sobel (Sobel & Feldman, 1973) para procesar el mapa de ocupación. Este paso permite identificar las fronteras entre las áreas mapeadas (conocidas) y las no mapeadas (desconocidas), facilitando la detección de fronteras para la navegación y exploración. La aplicación del filtro de Sobel resalta gradientes de intensidad que son interpretados como fronteras, las cuales son cruciales para determinar hacia qué zona del mapa se debe dirigir el robot continuando con el mapeo. En la Fig. 1 se presenta la frontera detectada mediante el algoritmo de Sobel. La primera etiqueta en color morado indica las zonas que no corresponden a fronteras, mientras que la segunda etiqueta que cualquier otro color que no sea morado son las fronteras identificadas por el algoritmo.

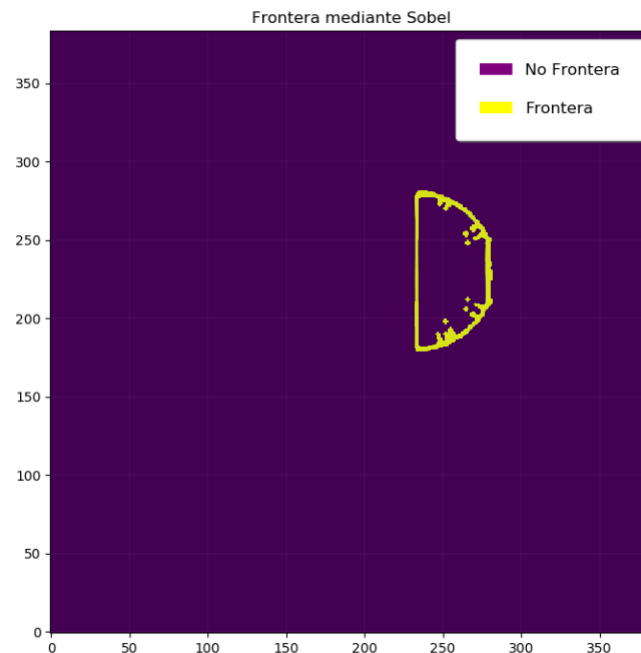


Fig. 1. Fronteras encontradas mediante el algoritmo de Sobel.

Mapeo basado en entropía

La entropía de Shannon, formulada por Claude E. Shannon (Shannon, 1948), es una medida en la teoría de la información que cuantifica la incertidumbre en una variable aleatoria. Se calcula como la suma ponderada de las probabilidades logarítmicas de todos los posibles estados de la variable, donde una mayor entropía indica una mayor incertidumbre en la información. En el contexto del mapeo robótico, la entropía de Shannon se utiliza para evaluar y seleccionar puntos de frontera para la exploración. Al medir la incertidumbre en los puntos fronterizos del mapa, el sistema puede priorizar puntos (x,y) que proporcionen la mayor cantidad de información nueva y útil sobre el entorno (Cover & Thomas, 2005). Para calcular la entropía en un punto específico de la frontera, nuestro método involucra la selección de un punto fronterizo seguido por la definición de una circunferencia centrada en este punto. Se consideran todas las celdas del mapa de ocupación dentro de esta circunferencia. La entropía en nuestro método se calcula sumando las contribuciones de cada celda con un peso reasignado de la malla de ocupación proporcionada por gmapping. Los valores de las probabilidades p de los estados de la sub-malla de ocupación que se utilizan para calcular la entropía se muestran en la Tab. 1.



Tab. 1. Ponderación de estados de la sub-malla de ocupación.

Valor p	Estado	Descripción
0.0	Libre	Celda transitable.
1.0	Obstáculo	Celda intransitable.
0.5	Desconocido	Celda inexplorada.

El objetivo de esta ponderación de celdas en la malla de ocupación es priorizar la exploración de zonas con mayor incertidumbre y potencial informativo, representadas por valores $p = 0.5$. Se asigna un valor más alto a las celdas desconocidas para incentivar al sistema de mapeo a dirigirse hacia estas áreas y obtener información detallada sobre el entorno. Se evaluó cada punto de la frontera y seleccionar hacia dónde dirigir la exploración, nuestro método suma los valores de todas las celdas que se encuentran dentro de un radio de 1.8 metros alrededor de cada punto considerado. Se selecciona el punto cuyo entorno circundante tiene la suma más alta de valores positivos, indicando un mayor potencial informativo y de exploración.

$$S = - \sum_{i \in C} p_i \log p_i \quad (1)$$

Donde en Ec. (1):

- C : conjunto de celdas dentro de la circunferencia.
- p_i : valor de la celda i .

El esquema seguido de mapeo basado en entropía se muestra en la Fig. 2.

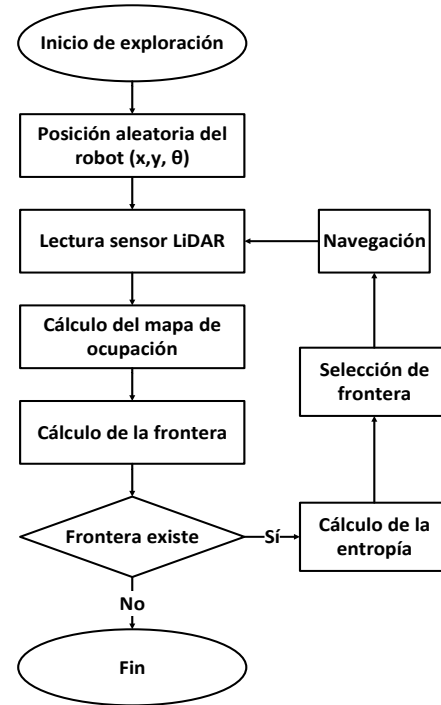


Fig. 2. Diagrama de búsqueda de frontera con mayor entropía.

Mapeo basado en frontera aleatoria

El método de mapeo basado en la selección de una frontera aleatoria utiliza una distribución uniforme discreta para la selección de esta. Este método encuentra todas las fronteras y selecciona una aleatoriamente a la que el robot debe navegar. Se mantiene la frontera hasta que esta cambia de estado y ya no se considera frontera. Se siguió el enfoque que se muestra en la Fig. 3.

Comparativa Visual de Métodos de Selección de Frontera

En la Fig. 4 se muestran las comparaciones visuales entre los métodos de selección de frontera por entropía y por selección aleatoria. Se aprecia que la selección de frontera por entropía selecciona el punto en la frontera donde se obtendrá la máxima reducción de entropía, optimizando así la eficiencia de la exploración al dirigirse hacia áreas que aporten una mayor información del mapa. La selección de frontera aleatoria únicamente busca un punto en la frontera cualquiera resultando en una exploración menos dirigida.

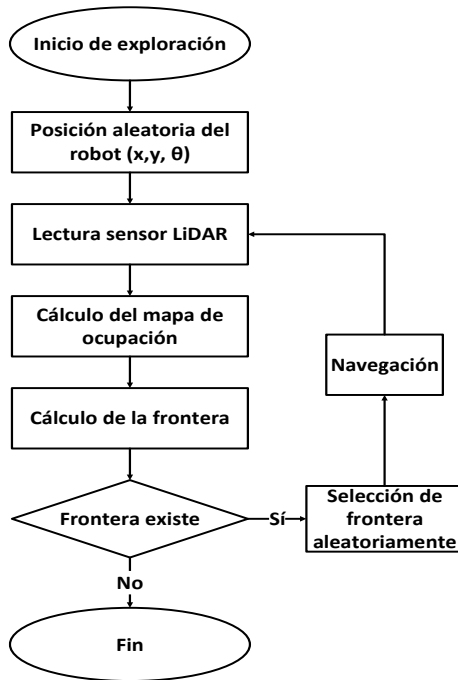


Fig. 3. Diagrama de búsqueda de frontera aleatoriamente.

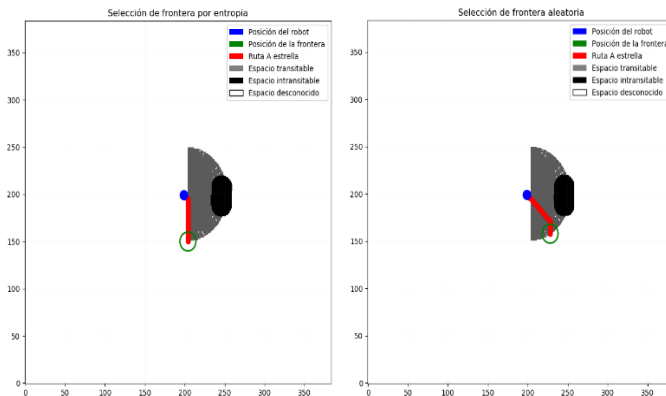


Fig. 4. Comparativa de frontera mediante selección por entropía vs selección aleatoria.

Entorno de simulación:

Un entorno de simulación es un programa desarrollado para construir modelos virtuales que emulan situaciones del mundo real. Estos modelos se fundamentan en la replicación precisa de las leyes física, permitiendo la evaluación del comportamiento de diversos agentes cuando interactúan dentro de distintos escenarios.

- **Plataforma de Simulación:** Se utilizó Gazebo como entorno de simulación.
- **Interfaz de Control:** El control del robot simulado se realizó a través de ROS (Robot Operating System).

Características del Robot

- Simulación del robot:** Se simuló el robot móvil terrestre Pioneer 3-AT con dimensiones de 0.497 por 0.508 [m]. El robot simulado se muestra en Fig. 5.

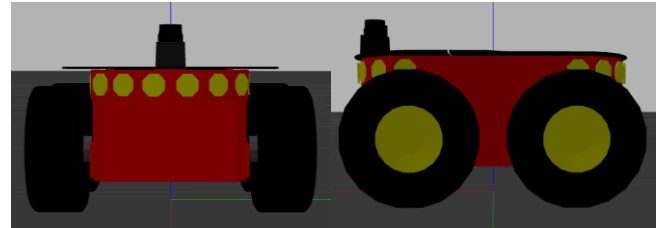


Fig. 5. Vista frontal y lateral del Pioneer 3-AT.

- Sensores:** El robot utiliza un sensor LiDAR con un rango de visión de 180 grados con un alcance máximo de 2.5 metros.
- Control del robot:** se implementó utilizando una estrategia de control en lazo cerrado. Este enfoque se divide en dos fases:
 - Orientación hacia el Destino:** El robot se orienta hacia su destino. Este ajuste de la orientación se realiza mediante un controlador proporcional que minimiza el ángulo entre la dirección actual del robot y el punto de destino.
 - Navegación hacia el Destino:** Una vez el robot se ha orientado, se desplaza hacia su destino. Se aplica un control proporcional para regular la velocidad de avance, ajustando dinámicamente la magnitud de la velocidad en función de la distancia restante hasta el destino.

Mapeo y Localización:

El mapeo y la localización, en el contexto de la robótica, se refiere al proceso mediante el cual un robot o vehículo autónomo genera un mapa o representación de su entorno mediante la recopilación de datos sensoriales (sensores LiDAR, ultrasonidos, cámaras, etc.) para crear modelos del mundo que rodea al robot. La localización es en el que el robot determina su posición y orientación dentro del mapa.

- SLAM:** Para la tarea de mapeo y localización, se implementó el paquete gmapping



proporcionado por el sistema de ROS. La configuración de los parámetros del algoritmo gmapping se basó en los ajustes propuestos en el estudio («An Extensive Analysis and Fine-Tuning of Gmapping's Initialization Parameters», 2023). En dicho estudio, se realizó un ajuste fino de los parámetros para optimizar la precisión del mapeo con un LiDAR que posee un rango máximo de detección de 3.5 metros.

- b. **Navegación:** Seguimiento de trayectorias se utilizó el algoritmo A estrella, recortando la ruta a 10 píxeles, aproximadamente 0.5 metros.

Diseño de escenarios:

Se crearon 3 escenarios cerrados con un tamaño de 8 X 8 metros con diferentes disposiciones de obstáculos, cajas y cilindros, mostradas en la Fig. 6a, 6b y 6c.

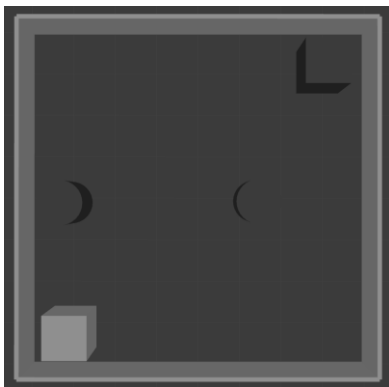


Fig. 6a. Vista aérea del escenario 1.

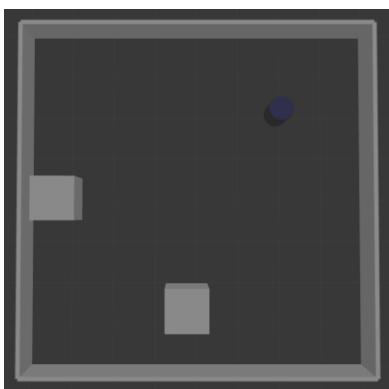


Fig. 6b. Vista aérea del escenario 2.

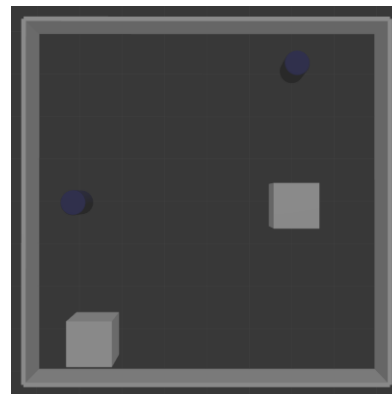


Fig. 6c. Vista aérea del escenario 3.

III. RESULTADOS

Se realizaron 10 simulaciones por escenario con cada enfoque de mapeo, donde se midió el tiempo de mapeo del entorno.

Tab. 2. Estadística de mapeo escenario 1.

Método	Promedio [seg]	Desviación estándar [seg]
Basado en entropía	349.7268	91.0059
Basado en frontera aleatoria	529.5033	292.4112

Tab. 3. Estadística de mapeo escenario 2.

Método	Promedio [seg]	Desviación estándar [seg]
Basado en entropía	448.2969	77.2677
Basado en frontera aleatoria	608.7114	218.0978

Tab. 4. Estadística de mapeo escenario 3.

Método	Promedio [seg]	Desviación estándar [seg]
Basado en entropía	562.6693	280.3214
Basado en frontera aleatoria	955.5629	317.7117

Las Tab. 2 a Tab. 4, muestran los resultados estadísticos de los métodos: basado en entropía y basado en frontera



aleatoria. Los resultados son evaluados en términos de tiempo promedio (segundos) necesario para completar el mapeo del escenario y la desviación estándar asociada a estos tiempos. La Tab. 5 compara los métodos de mapeo basados en entropía y en frontera aleatoria a través de los tres escenarios, destacando que el método basado en entropía reduce el tiempo de mapeo en un 35.0125% y demuestra mayor fiabilidad, según se refleja en su menor desviación estándar. Estos hallazgos sugieren que, para mapeo interior, el método basado en entropía es más efectivo y consistente.

Tab. 5. Comparación estadística de mapeo de los métodos en conjunto de los tres escenarios.

Método	Promedio [seg]	Desviación estándar [seg]
Basado en frontera aleatoria	2093.7776	276.0735
Basado en entropía	1360.6930	149.5316
Porcentaje de reducción	35.0125	45.8363

IV. CONCLUSIONES

El método de mapeo basado en entropía demostró una reducción significativa en el tiempo de mapeo en comparación con el método basado en frontera aleatoria obteniendo un porcentaje de reducción en los tres escenarios en conjunto de 35.0125%. La implementación exhibe una notable precisión en el mapeo del escenario, como se evidencia en la navegación del robot, que evita colisiones y se dirige eficazmente hacia áreas que requieren exploración. Este comportamiento se logra gracias al apoyo de la implementación de SLAM llamada Gmapping, que facilita un mapeo detallado y confiable del entorno. Este resultado sugiere que el enfoque basado en entropía no solo es más eficiente, sino que también podría servir como una base para futuras mejoras. Dado su desempeño fue mejor, este método se podría aplicar mejoras para reducir el tiempo de mapeo. Estas modificaciones podrían ser refinar los parámetros

utilizados en el algoritmo de entropía, como la ponderación de las celdas, submuestrear las zonas de frontera para acelerar la ejecución del algoritmo, optimizar los parámetros del esquema SLAM Gmapping, para observar si estos cambios ofrecen mejoras adicionales en la velocidad y precisión del mapeo; y probar el algoritmo en una variedad de entornos para asegurar que la mejora en el tiempo de mapeo sea consistente en diferentes condiciones y desafíos, como en entornos interiores densamente poblados o exteriores con variaciones significativas en el terreno.

AGRADECIMIENTOS

M. A. Rojas Andrade agradece al Conahcyt por el apoyo recibido a través de la Beca Nacional con CVU 1320004.

REFERENCIAS

- An Extensive Analysis and Fine-Tuning of Gmapping's Initialization Parameters. (2023). *International Journal of Intelligent Engineering and Systems*, 16(3), 126-138. <https://doi.org/10.22266/ijies2023.0630.10>
- Carrillo, H., Dames, P., Kumar, V., & Castellanos, J. A. (2015). Autonomous robotic exploration using occupancy grid maps and graph SLAM based on Shannon and Rényi Entropy. *2015 IEEE International Conference on Robotics and Automation (ICRA)*, 487-494. <https://doi.org/10.1109/ICRA.2015.7139224>
- Cover, T. M., & Thomas, J. A. (2005). Entropy, Relative Entropy, and Mutual Information. En *Elements of Information Theory* (pp. 13-55). John Wiley & Sons, Inc. <http://onlinelibrary.wiley.com/doi/10.1002/047174882X.ch2/summary>
- Elfes, A. (1989). Using occupancy grids for mobile robot perception and navigation. *Computer*, 22(6), 46-57. <https://doi.org/10.1109/2.30720>
- Lopez-Perez, J. J., Hernandez-Belmonte, U. H., Ramirez-Paredes, J.-P., Contreras-Cruz, M. A., & Ayala-Ramirez, V. (2018). Distributed Multirobot Exploration Based on Scene Partitioning and Frontier Selection. *Mathematical Problems in Engineering*, 2018, 2373642. <https://doi.org/10.1155/2018/2373642>



Shannon, C. E. (1948). A mathematical theory of communication. *The Bell System Technical Journal*, 27(3), 379-423. The Bell System Technical Journal. <https://doi.org/10.1002/j.1538-7305.1948.tb01338.x>

Sobel, I., & Feldman, G. (1973). A 3×3 isotropic gradient operator for image processing. *Pattern Classification and Scene Analysis*, 271-272.

Yamauchi, B. (1997). A frontier-based approach for autonomous exploration. *Proceedings 1997 IEEE International Symposium on Computational Intelligence in Robotics and Automation CIRA'97. «Towards New Computational Principles for Robotics and Automation»*, 146-151. <https://doi.org/10.1109/CIRA.1997.613851>



Implementación de Control de Carril Basado en Visión Artificial para Vehículos Autónomos a Escala

Alcocer Soriano Carlo Leonardo¹, Mireles Vázquez Miguel Ángel¹, Ramírez Paredes Juan Pablo I.^{2,*}

¹ Universidad de Guanajuato, Departamento de Ingeniería Mecánica, Carretera Salamanca - Valle de Santiago km 3.5 + 1.8 Comunidad de Palo Blanco, Salamanca, Guanajuato.

² Universidad de Guanajuato, Departamento de Ingeniería Electrónica, Carretera Salamanca - Valle de Santiago km 3.5 + 1.8 Comunidad de Palo Blanco, Salamanca, Guanajuato.

* Autor de correspondencia: jpi.ramirez@ugto.mx

Resumen. En el presente artículo se presenta la implementación de un sistema de control de carril para la navegación autónoma de un vehículo Autominy, estableciendo las bases para el desarrollo de un prototipo de mayor complejidad. El enfoque principal es diseñar y validar un sistema capaz de mantener el vehículo en su carril utilizando sensores y algoritmos de control que permiten ajustar la dirección con exactitud y precisión. La implementación propuesta no solo demuestra la viabilidad de utilizar el Autominy como una plataforma desarrollo para sistemas autónomos de bajo costo, sino que también sirve como un punto de partida para futuras mejoras en el control de la navegación. Los resultados obtenidos muestran que el sistema es capaz de responder adecuadamente a diversas condiciones del entorno, garantizando un desplazamiento seguro y eficiente dentro de los límites establecidos. Esta investigación contribuye al campo de la robótica móvil al ofrecer una solución práctica y accesible para la navegación autónoma en escenarios controlados.

Palabras clave. *Visión por computadora, Sistemas de control, Navegación autónoma, Python.*

I. INTRODUCCIÓN

La navegación autónoma ha revolucionado el campo de la robótica móvil, permitiendo que vehículos y robots operen de manera independiente en diversos entornos (Ramírez, 2003). Uno de los desafíos clave en la navegación autónoma es la detección y seguimiento de líneas, ya que estas proporcionan referencias visuales esenciales para mantener la trayectoria correcta en escenarios como pistas de prueba y carreteras (Moreano, 2019). Para abordar este desafío, se emplean técnicas avanzadas de visión por computadora que permiten al vehículo interpretar su entorno visualmente

y ajustar su movimiento en tiempo real (Rodríguez, 2015).

La detección de líneas es fundamental para guiar a los vehículos autónomos, ya que facilita la identificación de los límites de los carriles y otros elementos de interés en la vía. OpenCV, una de las bibliotecas más utilizadas en el procesamiento de imágenes y visión por computadora, ofrece una amplia gama de herramientas y algoritmos que simplifican la implementación de estos sistemas. Desde la aplicación de filtros y segmentación de colores hasta la detección de bordes y la Transformada de Hough, OpenCV proporciona las bases para desarrollar



soluciones robustas que permiten a los vehículos identificar y seguir líneas con precisión (Kaur, 2015).

En este contexto, se desarrolló un sistema basado en OpenCV para detectar y procesar las líneas de carril, utilizando métodos como la segmentación de color y la detección de bordes para extraer la información necesaria del entorno. Este enfoque permite no solo una detección precisa, sino también una integración efectiva con los sistemas de control del vehículo, mejorando la capacidad de navegación autónoma y sentando las bases para futuras mejoras en escenarios más complejos.

II. METODOLOGÍA

La implementación de nuestra propuesta se llevó a cabo sobre el Vehículo Autónomo Miny 4.0. Esta plataforma es un vehículo a escala, equipado con sensores de visión y distancia, además de su propia computadora embebida ejecutando un sistema operativo Linux y el software Robot Operating System (ROS, por sus siglas en inglés). Este vehículo es una iniciativa del grupo del Dr. Raúl Rojas de la Universidad Libre de Berlín¹.

Cámara

Se utilizó una cámara Intel RealSense Depth Camera D435 como dispositivo de entrada de imágenes (Fig 1). La cámara está compuesta por dos lentes infrarrojas, un lente RGB y un proyector láser infrarrojo, para el propósito de este proyecto solo será necesario el utilizar el lente RGB.



Fig. 1 Intel RealSense Depth Camera D435

Para este proyecto, se utilizó únicamente la lente RGB de la Intel RealSense Depth Camera D435, que captura

imágenes en color de alta resolución. Aunque la cámara también ofrece sensores de profundidad, la lente RGB es suficiente para los objetivos, ya que se enfocará en la detección visual de elementos clave, como las líneas de carril, simplificando el procesamiento de datos sin integrar la información de profundidad.

Detección de líneas

a. Transformación de imagen

La imagen capturada por la cámara puede presentar problemas debido al ángulo y al campo de visión del lente. Para abordar estas dificultades, es necesario generar una imagen con una vista en perspectiva aérea, o "vista de pájaro" (Ma, 2004). Esta transformación facilita un procesamiento más preciso al proporcionar una representación más adecuada del entorno, ajustando la imagen para superar las limitaciones del ángulo y la amplitud del lente.

En la Fig. 2 se observa el cambio de perspectiva para mostrar una vista aérea en la que, a diferencia de la imagen original, en la transformada las líneas del carril se asemejan más a dos paralelas verticales.

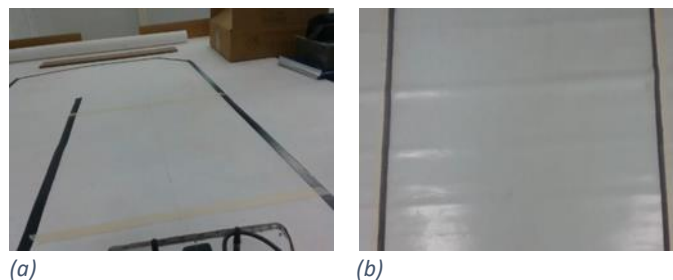


Fig. 2. (a) Imagen capturada sin perspectiva aérea, (b) Imagen capturada con perspectiva aérea

b. Filtrado de la imagen

Existen diversos métodos para detectar líneas en una imagen, cada uno con sus propias ventajas y limitaciones. En este trabajo, se seleccionó el método de segmentador de color debido a su simplicidad, rapidez y eficacia en la

¹ El sitio oficial del proyecto "Autominy" se puede consultar en <https://autominy.github.io/AutoMiny/>



detección de líneas específicas del carril. Este método funciona identificando y separando los píxeles de la imagen que corresponden al color de las líneas deseadas, lo cual permite generar una imagen binaria en la que solo las líneas coincidentes se destacan (ver Fig. 3).

El segmentador de color es particularmente útil cuando las líneas de carril tienen un color distintivo que contrasta con el entorno, lo que facilita su identificación incluso en condiciones variables de iluminación (Molina, 2001). Además, este enfoque no requiere un filtrado extenso previo de la imagen, lo que reduce significativamente el tiempo de procesamiento y mejora el rendimiento del sistema en tiempo real. Gracias a estas características, el segmentador de color se convierte en una herramienta eficiente y confiable para el control de carril en aplicaciones de navegación autónoma.

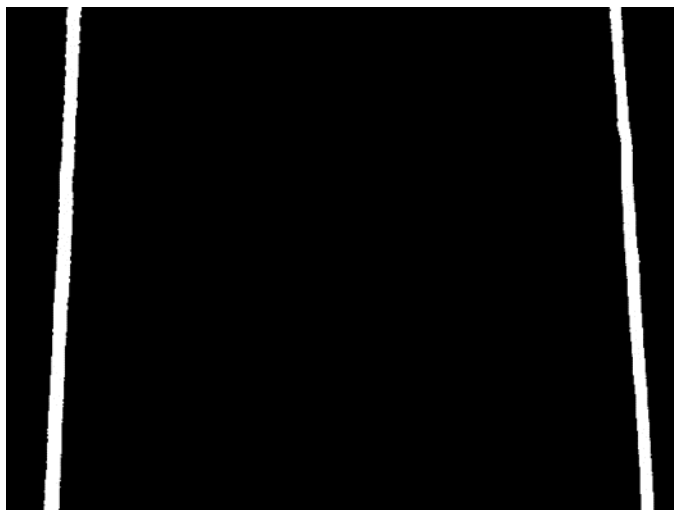


Fig. 3. Imagen con segmentador de color aplicado.

c. Transformada de Hough

La Transformada de Hough es una técnica utilizada en procesamiento de imágenes para la detección de formas geométricas, especialmente líneas y curvas. Este método convierte el problema de encontrar alineaciones de puntos en una imagen en la identificación de picos en un espacio de parámetros, facilitando la detección de patrones lineales incluso en imágenes ruidosas o incompletas. (Gonzalez, 2009).

En OpenCV, la transformada de Hough devuelve una lista de puntos iniciales y finales que definen las líneas encontradas en una imagen de interés (ver Fig. 4)

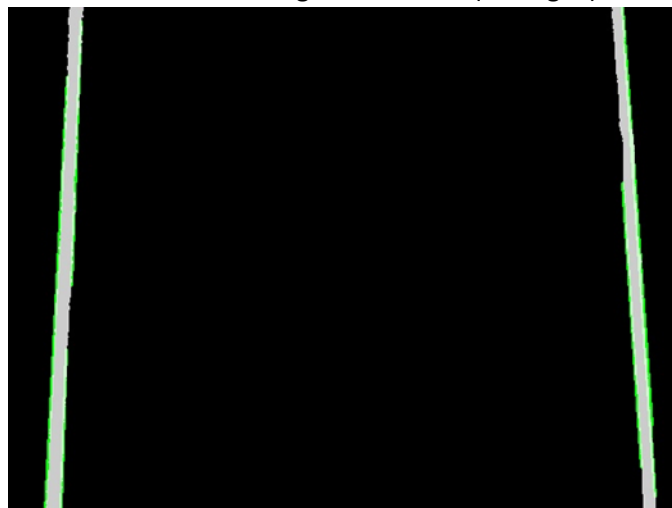


Fig. 4. Líneas encontradas en imagen aplicando transformada de Hough.

Se desarrolló un programa en Python para procesar las imágenes capturadas por la cámara del vehículo. El proceso comienza con el filtrado de la imagen para resaltar las líneas del carril, utilizando técnicas como el ajuste de color, la conversión a escala de grises y la aplicación de filtros de detección de bordes. Una vez identificadas, las líneas se clasifican en dos grupos: líneas de la izquierda y líneas de la derecha. Esta clasificación se realiza analizando la posición de cada línea en la imagen y su ángulo de inclinación respecto al eje vertical.

Tras la clasificación, se calcula el promedio de las líneas detectadas en cada grupo para formar una línea maestra a la izquierda y otra a la derecha, simplificando la representación del carril. Este enfoque reduce el impacto de líneas espurias provocadas por objetos u otros elementos que no pertenecen al carril. Las líneas maestras permiten al sistema de control mantener la trayectoria correcta, utilizando una línea central imaginaria paralela al eje vertical de la imagen como referencia para el cálculo del error de posición y orientación del vehículo.

d. Sistema de control



Para el control del movimiento, se implementaron dos controladores PID: uno dedicado al ajuste de la velocidad del vehículo y otro encargado de corregir el ángulo de giro de las llantas. El controlador PID para la velocidad regula la aceleración y desaceleración del vehículo en función de la distancia relativa a las líneas maestras, asegurando que el vehículo mantenga una velocidad segura y constante. Por otro lado, el controlador PID para el ángulo de giro utiliza el ángulo de las líneas maestras y la línea central como referencia para ajustar la dirección de las llantas, manteniendo el vehículo alineado dentro del carril.

La sintonización de los controladores PID se realizó experimentalmente. Se ajustaron las ganancias de ambos controladores a través de un proceso empírico, evaluando el rendimiento del vehículo en diferentes condiciones de navegación. El objetivo fue lograr una respuesta rápida y exacta del sistema, minimizando el error de trayectoria y asegurando una conducción estable. Los ajustes finales permitieron una navegación autónoma correcta y fluida, optimizando el tiempo de vuelta del vehículo y mejorando su capacidad de respuesta ante cambios en el entorno.

III. RESULTADOS

La Fig. 5 muestra cómo, tras el procesamiento de la imagen, se obtienen las líneas maestras, las cuales se despliegan en una nueva imagen para que el usuario pueda apreciarlas visualmente.

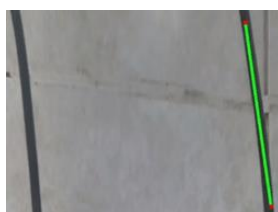


Fig. 5. Secuencia de imágenes vistas por la cámara del Autominy mientras sigue el carril

Para el controlador de velocidad de las llantas, se determinó que era suficiente utilizar un controlador proporcional con una ganancia de 0.4. Esta configuración permitió ajustar eficazmente la velocidad del vehículo sin necesidad de añadir componentes integrales o derivativos al sistema de control.

Se evaluaron diferentes valores de las ganancias del controlador angular, destacándose los valores presentados en la Tabla 1, junto con los tiempos de cada vuelta. Los resultados mostraron que no fue necesario implementar un controlador PID completo, ya que las ganancias ajustadas proporcionaron un desempeño óptimo en la navegación autónoma. El vehículo completó un circuito de 10.22 metros por vuelta, realizando al menos 10 vueltas efectivas.

Tab. 1. Ganancias proporcionales (k_p) y derivativas (k_d) aplicadas al controlador angular de las llantas.

k_p	k_d	Tiempo (s)
0.005	0.0003	13
0.0045	0.0008	14

IV. CONCLUSIONES

Se implementó un sistema de control capaz de mantener la navegación autónoma del vehículo en un entorno controlado, estableciendo una base sólida para futuras mejoras y ampliaciones del sistema. Este enfoque integral no solo garantiza un control confiable, sino que también proporciona una plataforma versátil que permite la adaptación del vehículo a diferentes escenarios y condiciones de prueba.

Sin embargo, se identificó como limitante el tiempo de transferencia y procesamiento de datos, ya que



actualmente todo el procesamiento se realiza en una computadora externa, lo que requiere una conexión inalámbrica estable para evitar la pérdida de información crítica. Para superar esta limitación, se propone integrar el código directamente en la computadora a bordo del vehículo, lo que eliminaría la dependencia de una conexión externa y mejoraría tanto la eficiencia como la fiabilidad del sistema de navegación autónoma.

AGRADECIMIENTOS

Agradecemos al Centro de Investigación en Matemáticas (CIMAT) y a la Alianza en Inteligencia Artificial por la asignación del Vehículo Autónomo Miny Versión 4.0 utilizado en este trabajo.

REFERENCIAS

- Ramírez, G. (2003). Método de aprendizaje simple para navegación de minirobots móviles rodantes. *Dyna*, 70(138), 59-66.
- Kaur, G., & Kumar, D. (2015). Lane detection techniques: A review. *International Journal of Computer Applications*, 112(10).
- Gonzalez, R. C. (2009). *Digital image processing*. Pearson education india.
- Ma, Y., Soatto, S., Košecák, J., & Sastry, S. (2004). *An invitation to 3-d vision: from images to geometric models* (Vol. 26). New York: springer.
- Rodriguez-Garavito¹², C. H., Guindel, C., & Armingol, J. M. (2015). Sistema de asistencia a la conducción para detección y clasificación de carriles.
- Molina, A. (2001). *Algoritmos jerarquicos para segmentacion de imágenes en color* (Doctoral dissertation, Universidad de Málaga).
- Moreano, G., Cajamarca, J., & Tenicota, A. (2019). Agricultura de Precisión: Preprocesamiento y Segmentación de Imágenes para Obtención de una Ruta de Navegación Autónoma Terrestre. *Revista Politécnica*, 44(2), 43-50.



Monitoreo del comportamiento térmico en probetas PU expuestas a flamabilidad

Sandoval Martínez Karen J.¹, Tafolla Barajas Jessica G.¹, García Guzman José M.², Razón González Juan P.², Rostro González Horacio³, Cano Lara Miroslava¹

¹ Tecnológico Nacional de México/ITS de Irapuato, Depto. de Mecatrónica, Irapuato, Gto. México.

² Tecnológico Nacional de México/ITS de Irapuato, Depto. de Electromecánica, Irapuato, Gto. México.

³ DICIS/ Universidad de Guanajuato, Depto. De Electrónica, Salamanca, Gto. México.

Autor de correspondencia: LIS20110601@irapuato.tecnm.mx, miroslava.cl@irapuato.tecnm.mx

Resumen. Se crearon probetas de resina de espuma de poliuretano (PU) con refuerzo orgánico de Pulverizado orgánico de cascaras y fibras naturales, con la finalidad de favorecer con su capacidad calorífica y crear una propiedad capaz de retardar el efecto térmico en pruebas de flamabilidad, la flama consta de una temperatura de 270°C, con ello se tiene el material PU como concentrado en muestras de probetas, por si solo se tuvo una temperatura externa de 109°C mientras que el sensado interno conto de 97°C consecuentemente, el sistema de monitoreo térmico se desarrolló mediante sensores térmicos actuando a la vez en la probeta, los parámetros establecidos de acuerdo con el arreglo físico de la muestra es interno y externo como se mencionó con anterioridad, del lado interno se tienen dos multímetros registrando la temperatura y a la par dos sensores LM35 corroborando su estado y registro, en conjunto se correlacionó el ingreso de datos térmicos en el programa mediante Arduino y Python, almacenando los archivos en formato CSV, mientras que del lado externo tenemos la captura de datos térmicos mediante una cámara termográfica obteniendo como resultado fotografías térmicas de las pruebas de flama exteriores, por otro lado, tenemos las curvas de entalpía presentando termogramas con los datos almacenados mostrando el sistema de sensado mediante una serie de gráficas que determinan el periodo de tiempo expuesto desde temperatura ambiente, pico máximo y estado de reposo.

Palabras clave. Monitoreo, estimación térmica, sensores de temperatura, pulverizados orgánicos

I. INTRODUCCIÓN

Los materiales compuestos se caracterizan por una fase continua llamada matriz; una fase discontinua llamada refuerzo y la interfase, que corresponde a la superficie donde matriz y refuerzo entran en contacto [1]. La matriz es utilizada para asegurar la unión y distribución uniforme de partículas o fibras de refuerzo que generan un composite [2]. Adquiriendo las propiedades físicas y químicas del material añadido, que no se pueden obtener en materiales originales.

Entre los diversos materiales compuestos, los compuestos de polímeros reforzados con fibras naturales son más utilizados debido a su fácil procesamiento y sus enormes ventajas ecológicas. Se han estudiado con diferentes tipos de fibras naturales y sus propiedades como posible reemplazo de fibras sintéticas [3].

La espuma de poliuretano que se ha utilizado en diversos campos como matriz polimérica o el uso puro de la misma. El estudio de las propiedades mecánicas de los



compuestos reforzados con fibras naturales depende en gran medida del grado de adhesión entre la fibra natural y la matriz [4]. Se tienen fibras para mejorar las características particulares, ejemplo de ello es la fibra de coco, y fibras de agave que se preparan en forma de matriz, se mezclan con resinas apropiadas para lograr una unión adecuada del material, donde los resultados de la experimentación se comparan con otros tipos de compuestos poliméricos reforzados [5].

Con la oportunidad de mejora sobre retardantes térmicos de llama el hidróxido de aluminio, hidróxido de magnesio en un material compuesto de fibra de vidrio con matriz polimérica utilizado en la fabricación del interior de carrocerías metálicas y su influencia en el índice de flamabilidad [6]. Dentro de los rangos de temperatura que se tiene de la espuma de poliuretano (PU) se encuentra la temperatura de servicio, es la temperatura más alta a la que se puede usar el material donde alcanza los 90°C, mientras que la temperatura mínima es de -60°C, además el poliuretano no supera el valor de conductividad térmica inicial de 0.022 W/m*K este valor de conductividad va aumentando ligeramente con el tiempo hasta llegar finalmente a estabilizarse [7].

La eficiencia térmica del poliuretano utilizado como aislante térmico se experimenta con el acabado PU, donde el instrumento de medida son los calorímetros, además con el fin de corroborar temperaturas se implementó un sensor LM35 y un sensor de temperatura PT-100 donde se puso a prueba los materiales más frecuentes del mercado frigorífico los cuáles son poliestireno expandido, lana de vidrio y poliuretano, recolectando los datos y tabulando los datos reflejados por el comprobador digital de los calorímetros. Para determinar la eficiencia se realizó los cálculos pertinentes los cuales reflejaron que el poliuretano es el mejor aislante térmico resultando un 65,76% de eficiencia, siendo este el mejor porcentaje comparado con los otros aislantes [8].

El comportamiento interno de la reacción térmica de la resina de espuma de poliuretano es de un comportamiento endotérmico ya que se almacena el calor interno dentro de ella, mientras que en distintas literaturas se tienen curvas calorimétricas diferencial de

barrido (DSC) de los materiales reforzados mezclados en donde se realizan una serie de termogramas en relación con la temperatura y el flujo de calor, en las cuales comparten picos exotérmicos.

En el presente trabajo se implementa el pulverizado de cáscara de arroz y pulverizado de fibra de coco como refuerzo orgánico por la capacidad calorífica que presenta para la creación de probetas de material compuesto como inhibidor calorífico, simultáneamente se realiza el sensado de temperatura con sensores LM35 distribuidos a lo largo de la probeta, con la finalidad de registrar la variación de temperatura a lo largo de la muestra.

II. MARCO TEÓRICO

La resina de espuma de poliuretano (PU) es muy utilizada ante una amplia variedad de propiedades físicas, derivada de las reacciones químicas entre poliol e isocianato en la formación de enlaces uretano generando burbujas y expandirse hasta endurecerse, desarrollando una matriz uniforme de celdas [9], encapsulando y unir a otros materiales compuestos [10]. Es relevante que la mezcla esté en las proporciones adecuadas, manteniendo una relación 1:1 con respecto a la superficie y las sustancias [11].

En el comportamiento térmico de los termogramas, contando con 4 fases:

- Temperatura ambiente: establece un punto de referencia de temperatura inicial.
- Fase de calentamiento o expansión: introducción de la llama a la probeta, el cual, absorbe y distribuye la energía térmica por un lapso de tiempo hasta retirar la llama.
- Punto endotérmico: Una vez retirada la flama, alcanza una máxima de absorción de calor, pasando la siguiente fase de descender.
- Fase de enfriamiento: Descenso gradual de temperatura después del punto endotérmico, donde el material libera calor lentamente hasta llegar a temperatura ambiente.

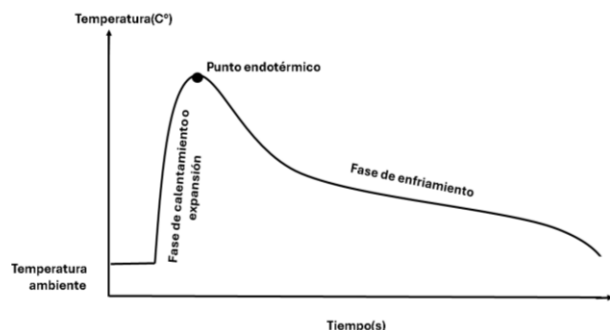


Fig. 1. Termograma del comportamiento térmico en probetas PU y CA.CC

III. METODOLOGÍA

Se ha considerado la incorporación los compuestos orgánicos en una la matriz que podría adquirir propiedades de alta resistencia térmica, adecuadas para diversas aplicaciones. Fabricando probetas que serían sometidas a pruebas de flamabilidad. Los materiales empleados son PU, además, de una variante reforzada por la combinación de cáscara de arroz (CA) y pulverizado de fibra de coco (CC), las siguientes tablas identifican su composición química y sus propiedades Tab. 1 y Tab. 2. En la Fig. 2, se muestra la delineación de las probetas.

Tab. 1. Composición química PU y AC

Composición química		
PU	Poliol	50-60%
	Isocianato	40-50%
	Agentes de expansión (agua, químicos, etc)	1-5%
	Catalizadores y aditivos	-1%
Cáscara de arroz	Celulosa	35%
	Hemicelulosa	25%
	Lignina	20%
	Proteína cruda	3%
	Ceniza	17%
Fibra de coco	Celulosa	35-43%
	Lignina	40-45%

Hemicelulosa	0,15-24%
Solubles en agua	5,25%
Pectinas	3,3-4,0%
Materia mineral (cenizas)	2,2%
Grasas y ceras	-
Cristalinidad	25-33%
Ángulo de orientación de Herman (grados)	41-45 grados

Tab. 2. Propiedades de PU y AC.

Propiedades	
Densidad	PU: 0.05 - 1.7 g/cm ³ . CA: Densidad Aparente. 100 a 160 kg/m ³ . Densidad Real. 670 a 740 kg/m ³ . CC: Densidad aparente: 50 y 200 kg/m ³ . Densidad real: aprox. 1.15g/cm ³ .
Conductividad térmica	PU: 0,024 W/m°C. CA: 0.0359 W/m°C. CC: 0.043 W/m°C.
Dureza	PU: Espuma flexible: 45-98 Shore A Espuma rígida: 51 – 85 Shore D. CA: 6 Mohs. CC: -
Capacidad calorífica	PU: Aprox. 1.4 KJ/kg*K. CA: 3300 a 3600 cal/kg. CC: Aprox. 1.84 kJ/kg*K.
Temperatura de incineración	PU: Aprox. 225°C y degradación térmica comienza 90°C. CA: 800 a 1000°C. CC: Se puede descomponer a temperaturas superiores a los 200°C, pero por completo es alrededor de los 800°C.

Para el estudio del sensado de probetas de material compuesto expuestas a flamabilidad se emplearon sensores de temperatura (LM35). También, tarjeta Arduino y el software en Python. El código de Python actúa como receptor de estos, anexando una condicional

en cuanto se cierra el puerto almacena todos los valores extraídos en un archivo tipo CSV Fig. 3.

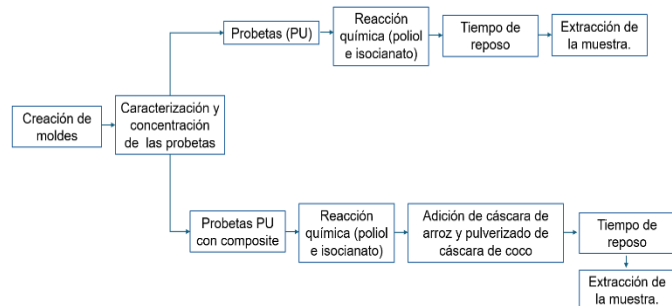


Fig. 2. Delineación de probetas PU y CA.CC.

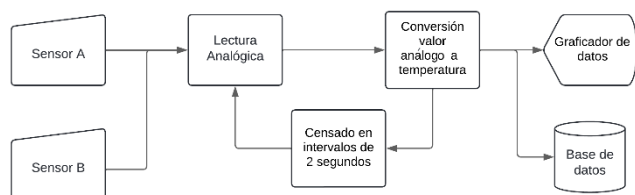


Fig. 3. Esquema de registro de datos en el programa.

Las probetas de composite se caracterizan y se controla su tamaño, espesor y materiales base, excepto de las muestras de probeta PU Fig. 4. Las condiciones pueden alterar el estado de las probetas, se definen como variables. Variables fijas: condiciones ambientales. Variables no controlables: temperatura ambiental durante el secado, se disminuye la temperatura, se retrasa el tiempo de extracción de la probeta.

Tab. 3. Etiquetado y materiales

Etiquetado de probeta	Descripción de materiales
PU	Resina de espuma de poliuretano.
CA	Pulverizado de cáscara de arroz.
CC	Pulverizado de fibra de coco.
CA.CC	Resina de espuma de poliuretano con pulverizado de cáscara de arroz y pulverizado de fibra de coco.

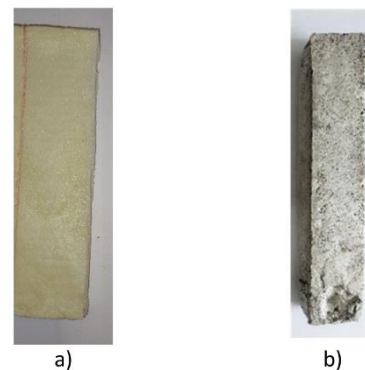


Fig. 4. Probeta a) PU y b) CA.CC

En el conjunto existe el monitoreo interno conformado por dos sensores LM35 denominados “A” y “B” en ellos se encuentra la particularidad de estar a diferente distancia entre sí, con la finalidad de captar ambas variaciones de temperatura en relación a la flama, donde una se encuentra en mayor contacto en cuestión de flamabilidad, mientras que la otra está en una zona más lejana del área, a continuación, en la Fig. 5.a, se muestra la distancia correspondiente de los sensores, simultáneamente se encuentran encapsuladas las puntas de temperatura “termopar tipo K” de dos multímetros profesionales “MUL03 SURTEK NCV” y “STEREN DIGITAL MULTIMETER MUL-500” respectivamente denominados “A” y “B” en relación a la distancia de los sensores al costado de la probeta. Mientras el sistema de monitoreo térmico superficial está conformado por una estructura metálica capaz de resistir altas temperaturas donde la probeta es colocada de manera vertical con relación a la flamabilidad del mechero Fig. 5.b.

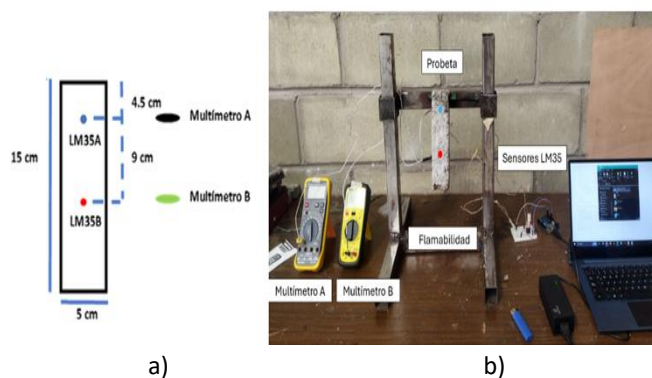


Fig. 5. a) Distribución de sensores y b) Sistema de monitoreo térmico.



Ante los parámetros mencionados se encapsularon los sensores LM35 dentro de las probetas PU y CA. CC mostrados a continuación en la Fig. 6, donde se realizaron una serie de muestras de probetas para capturar los datos de temperatura.

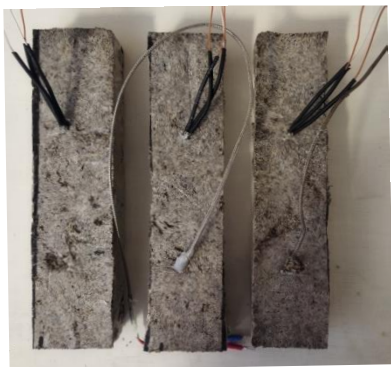


Fig. 6. Encapsulados de los sensores CA.CC.

En la extracción de los valores térmicos a través de los sensores LM35 el código de Arduino los almacena por medio de los pines analógicos A3 y A4 correspondientes a cada sensor mostrado en la Fig. 7, transformando el valor analógico de rango 0 a 1023 del valor de temperatura con un retardo de 2 segundos.

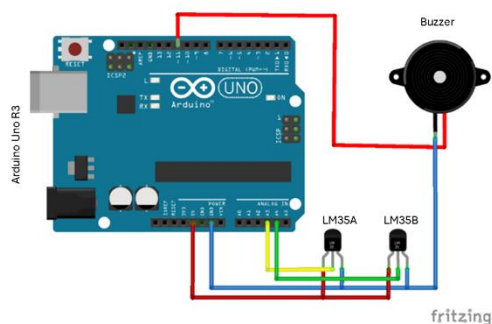


Fig. 7. Esquema de conexión de los sensores LM35 "A" y "B".

Al mismo tiempo se registra la temperatura con la cámara termográfica "FLUKE IR Fusion Thermal Imager".

IV. RESULTADOS

En el análisis del sensado térmico de la probeta empezó desde temperatura ambiente 25°C c un tiempo de reposo de 16 segundos, posteriormente se introdujo la flama del mechero a una temperatura de 270°C, el comportamiento que produce es endotérmico al absorber el calor aumentará gradualmente en un tiempo de 58 segundos, en este lapso incrementa el calor

interno conforme la flama esté en la probeta; no obstante, en un lapso determinado de tiempo al retirar el incentivo térmico seguirá almacenando calor, pero no rebasará el pico más alto, al terminar este tiempo empieza la etapa de decremento en la temperatura dentro de la probeta actuando en un lapso de 480 segundos.

La probeta PU fue estudiada por 4 señales de sensores; dos sensores de LM35 "A" y "B" y dos termopares tipo K "A" y "B", colocados como se muestran en la Fig. 5.a, evaluando su comportamiento. Destacándose que el sensor LM35 "B" alcanzó el pico endotérmico más alto entre de 97 °C en un lapso de 82 segundos y sensor del multímetro "B" alcanzó 40 °C en 132 segundos, por otro lado, el sensor LM35 "A" su punto endotérmico fue de 73°C en 83 segundos y multímetro "A" de 40°C en 84 segundos.

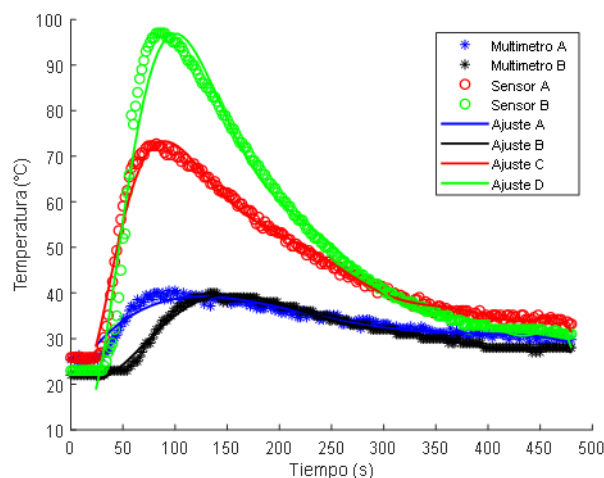


Fig. 8. P2_PU_2. Termograma de reacción de probeta de PU.

Al momento de su elevación ocurre una propagación de calor, en tanto, externa Fig. 12 como interna Fig. 8 Liberando gases inflamables y sustancias químicas dañinas como monóxido de carbono y dióxido de carbono, adquiriendo la apariencia de oxidación, descomponiéndose en la parte donde está en contacto más la flama provocando una deformación de proporciones y volviendo a productos más simples Fig. 10.a. Observándose que protege de esta oxidación su parte interna Fig. 11. a, evitando que se consuma y afecte internamente, pero la propagación de calor es alta.



Finalmente, cuando se retira la llama se extingue automáticamente el fuego que la cubre, gracias a sus propiedades Tab. 1 y Tab. 2, y comienza el descenso de forma lenta de temperatura hasta llegar a temperatura ambiente. En tanto, la probeta CA.CC se observó su punto endotérmico más alto LM35 "B" con una temperatura de 46°C en 110 segundos y multímetro "B" 37°C en 96 segundos y se mantuvo 57 segundos constante, mientras LM35 "A" 38°C en 55 segundos y multímetro "A" 30°C en 95 segundos. Reduciendo evidentemente su propagación calorífica interna de la probeta.

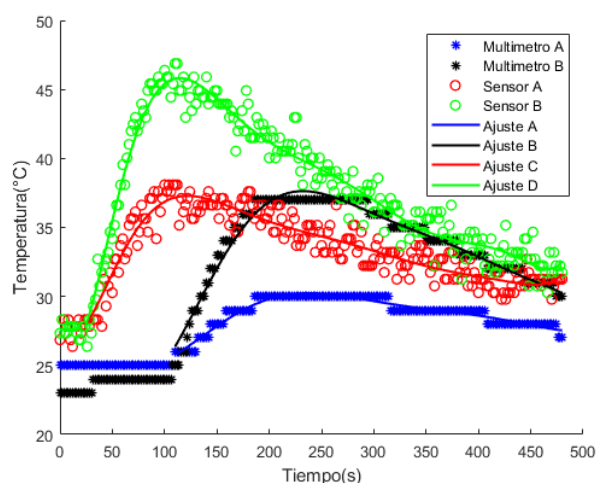


Fig. 9. P2_AC_2. Termograma de reacción de probeta de CA.CC.

Al comenzar la exposición de la flama, identificó una diferencia de temperatura interna Fig. 9 y externa Fig. 13, el cual, se redujo debido a las propiedades adquiridas de la composición química de los elementos orgánicos de la probeta, provocando una carbonización en la parte inferior de la probeta, dando como resultado un almacenamiento de calor y evitar su difusión acelerada internamente, además, sufriendo una menor deformación Fig. 10.b, y mayor protección del núcleo Fig. 11. b. Generando un menor valor térmico en sus puntos endotérmicos y que al retirar la llama volvió lentamente a su temperatura ambiente.



a) b)
Fig. 10. Probeta a) PU b) CA.CC



a) b)
Fig. 11. Probeta a) PU b) CA.CC.

Aclarando en la Fig. 11. b, se realizó una prueba de larga duración doblando el tiempo original.

Dentro del rango superficial de temperatura se encuentra el registro termográfico con la captura de fotografías térmicas donde se analiza el comportamiento externo de las probetas expuestas a la flama en comparación al sensado interno, a continuación, en la Fig. 12. Y se muestran los parámetros de color térmicos de la probeta PU, donde se observa un registro externo de 109.1°C, conforme a la Fig. 8, el punto más alto registrado internamente es de alrededor 97°C, obteniendo una diferencia de temperatura de 11°C. La diferencia de temperatura se da con el sensado interno y externo.

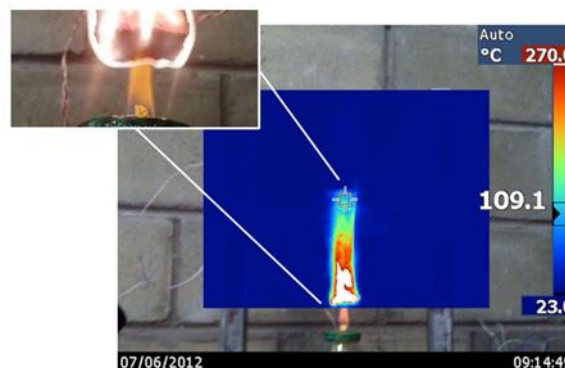


Fig. 12. Registro de temperatura externa probeta de PU.



Mientras que el registro de temperatura externo de la probeta AC con composite Fig. 13, es de 88.9°C simultáneamente el pico más alto interno registrado es de alrededor 46°C de acuerdo con la Fig. 9, con una diferencia de temperatura de 42.9°C.

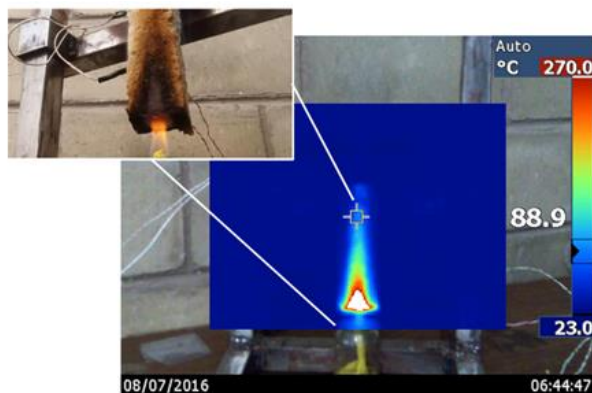


Fig. 13. Registro de temperatura externa de probeta de AC

V. CONCLUSIÓN

Al trabajar con espuma de poliuretano puede ser una gran utilidad debido a su ligereza y que puede ser aplicado a cualquier superficie como envolvente o relleno. Sin embargo, cuenta con una muy importante desventaja que no es idealizada para el retardo térmico al momento que entra en contacto con la flama, ya que las propiedades de sí misma no favorecen la muestra, además presenta un alto índice combustible, pero al retirarla se autoextingue. En comparación a la probeta AC que al someterla a una llama directa no es propensa a la propagación de la flama, al contrario, genera un solo punto térmico que se propaga a lo largo de la muestra, mientras que el aumento de temperatura interna es más lento; de acuerdo con los resultados el punto más alto de la probeta de PU fue de 97°C mientras que el de la probeta de composite fue de 46°C obteniendo una diferencia de 51°C.

VI. AGRADECIMIENTOS

Se agradece al proyecto “Inteligencia Artificial y análisis predictivo de propiedades mecánicas de compuestos poliméricos reforzados con materiales naturales” 19781.24-PD con registro en el TecNM.

VII. REFERENCIAS

- [1] C. E. Njoku, K. K. Alaneme, J. A. Omotoyinbo, and M. O. Daramola, (2019). *Adv. Mat. Lett.* 10, 682
- [2] Reblar, B. (2022, 11 abril). Los materiales compuestos | Ciencia de los materiales. Ciencia De Los Materiales.
- [3] T. Gurunathan, Smita Mohanty, Sanjay K. Nayak, (2015). A review of the recent developments in biocomposites based on natural fibres and their application perspectives, *Composites Part A: Applied Science and Manufacturing*, Volume 77, 2015, Pages 1-25, ISSN 1359-835X, <https://doi.org/10.1016/j.compositesa.2015.06.007>.
- [4] Chunhua Zhang, Junqing Li, Zhen Hu, Fenglei Zhu, Yudong Huang. (2012). Correlation between the acoustic and porous cell morphology of polyurethane foam: Effect of interconnected porosity, *Materials & Design*, Volume 41, 2012, Pages 319-325, ISSN 0261-3069, <https://doi.org/10.1016/j.matdes.2012.04.031>. (<https://www.sciencedirect.com/science/article/pii/S0261306912002749>).
- [5] University Malaysia Pahan, (2016). Study on fatigue life and fracture behaviour of fibreglass reinforced composites. UMP PUBLISHER, pp. 2300-2310.
- [6] Byron, C. M. C. (2019). Análisis de la aplicación de retardantes de llama (hidróxido de aluminio, hidróxido de magnesio) en un material compuesto de fibra de vidrio con matriz polimérica utilizado en la fabricación del interior de carrocerías metálicas y su influencia en el índice de inflamabilidad.
- [7] CTE. (2016). Libro blanco de Poliuretano Proyectado e Inyectado Guía de ventajas y soluciones de espuma rígida de poliuretano proyectado e inyectado para aislamiento térmico, acústico e impermeabilización, conforme al CTE. L (4.0). <https://globalpur.pt/wp-content/uploads/2024/01/Catalogo-Espuma-de-Poliuretano-Projetada.pdf>.



- [8] Proaño Quezada, Héctor Mauricio. (2014). Determinación de la eficiencia térmica del poliuretano utilizado como aislante térmico, en la empresa ESPROM-PUR Ambato. Escuela Superior Politécnica de Chimborazo. Riobamba.
- [9] Connor, N. (2023, septiembre 18). Espuma de poliuretano | Fórmula, propiedades y aplicación. Material Properties.
- [10]Schmolke, R., & Schmolke, R. (2023, 26 mayo). Cómo funcionan las espumas de poliuretano: Química, personalización y aplicaciones de uso final. Polywater.
- [11]Charles Kuranchie, Abu Yaya, & Yaw Delali Bensah. (2021). African Institute of Mathematical Sciences, The effect of natural fibre reinforcement on polyurethane composite foams. Elsevier.
<https://doi.org/10.1016/j.sciaf.2021.e00722>



Simulación cinemática de robot SCARA

García García Ricardo^{*1}, Rico Martínez José María¹, Cervantes Sánchez J. Jesús¹, García Murillo Mario Alberto¹, Rodríguez Ramírez Cristian¹, Ayala García Francisco²

¹ Universidad de Guanajuato, Departamento de Ingeniería Mecánica, Carretera Salamanca-Valle de Santiago km 3.5+1.8. Comunidad de Palo Blanco, 36885 Salamanca, Guanajuato

² Tecnológico Nacional de México/ITS de Abasolo, Ingeniería en Sistemas Automotrices, Blvd. Cuitzeo de los Naranjos #401, Col. Cuitzeo de los Naranjos, CP. 36976, Abasolo, Guanajuato

* Autor de correspondencia: garciar@ugto.mx

Resumen. En este trabajo se muestra un ejemplo práctico del uso del software GeoGebra para la simulación de la cinemática directa e inversa del manipulador SCARA RRRP mediante el uso de los parámetros Denavit-Hartenberg (DH) en su convención proximal. El objetivo es demostrar la aplicación del software en el análisis de manipuladores robóticos. El procedimiento expuesto repasa la solución al problema de la cinemática directa de posición y la solución al problema de la cinemática inversa de posición para el robot estudiado. Después, los modelos matemáticos son implementados en GeoGebra, donde las construcciones geométricas son elaboradas a partir de estos modelos dentro del mismo software. Los resultados son simulaciones interactivas y dinámicas del robot, las cuales estimulan el auto-aprendizaje a través de la exploración de las construcciones. De esta forma se demuestra que los programas de geometría dinámica como GeoGebra son capaces de simular la cinemática de sistemas robóticos para fines educativos y de investigación.

Palabras clave. GeoGebra, Robot, Modelado, Cinemática.

I. INTRODUCCIÓN

El software de geometría dinámica (SGD) ha adquirido una importancia significativa en el campo de la enseñanza en las últimas dos décadas, ya que estimula el aprendizaje autónomo de los estudiantes mediante la exploración y construcción interactiva de modelos matemáticos altamente dinámicos. Programas como The Geometric Supposer, Cabri, Cinderella, The Geometer Sketchpad y Euklid fueron pioneros en la representación dinámica computarizada de la geometría euclidiana [Driscoll, 1985; Kortenkamp, 1998; Kortenkamp, 1999; Lehrer, 1988; King, 1997]. Sin embargo, en tiempos recientes se ha

popularizado notoriamente el uso del software GeoGebra [Roanes-Lozano, 2023], el cual es un software de matemáticas dinámicas (SMD) que incluye numerosas herramientas educativas. Por ejemplo, incluye un módulo para el cómputo de álgebra simbólica (CAS), hojas de cálculo, herramientas geométricas – incluye un SGD –, junto con una variedad de funciones enfocadas al cálculo, álgebra, probabilidad, estadística, ecuaciones diferenciales y graficado de funciones. Es un software para todos los niveles educativos que apoya al movimiento CTIM (Ciencia, Tecnología, Ingeniería y Matemáticas, STEM por sus siglas en inglés), es gratuito,



multilenguaje, multiplataforma, y posee una creciente comunidad de estudiantes, profesores e investigadores alrededor del mundo a través de su plataforma en internet, en donde es posible intercambiar recursos educativos fácilmente.

El software GeoGebra es bien conocido en el campo educativo para la enseñanza de geometría básica, en donde numerosos estudios [Chivai, 2022, Tomié, 2019; Helsa, 2023; Türkoğlu 2024] demuestran su efectividad en el mejoramiento de características cognitivas del estudiante como la percepción espacial, el pensamiento computacional y en la mejora del razonamiento para resolver problemas. Sin embargo, sus capacidades no han sido aprovechadas adecuadamente en el campo de la robótica.

La literatura actual muestra el desarrollo de paquetes de software específicos aplicados a la enseñanza de mecanismos y robótica. Por ejemplo, software específico para la enseñanza de mecanismos planos se encuentra en [Macho, 2021; Müller, 2020; Beckerman, 2019], mientras que para mecanismos espaciales se encuentra el software desarrollado en [Sagar, 2020]. Sin embargo, pocos trabajos se encuentran para la enseñanza de la robótica teórica mediante el software GeoGebra, algunos de éstos trabajos se encuentran en [Suwarno, 2024; Husty, 2019]. Debido a ello, el objetivo del presente estudio es mostrar ejemplos de aplicación del software GeoGebra en el campo de la robótica.

II. Análisis del robot SCARA

El robot SCARA (Selective Compliance Assembly Robot) es un robot ampliamente usado en la industria debido a sus prestaciones notables de rapidez, bajo costo, y alta precisión. Fue desarrollado por Hiroshi Makino en la Universidad de Yamanashi Japón en 1978 principalmente para atender procesos de ensamblaje de circuitos integrados. Los

resultados fueron presentados en la primera conferencia internacional sobre la automatización del ensamblaje [Makino y Furuya, 1980], y el modelo fue patentado en 1982 [Makino, 1982]. El diseño consiste de eslabones y revolutas dispuestos de tal forma que permiten tres traslaciones independientes y una rotación alrededor de un eje de orientación fija, como el que se muestra en la Figura 1.

El diseño original de Makino consiste de dos revolutas, R1 y R2, junto con un par cinemático prismático P, en cuyo extremo se encuentra el efector final del robot. Sin embargo, en la actualidad es común la variante que incorpora una tercera revoluta, R3, para girar el par prismático y posicionar el objeto tomado por la pinza en la orientación adecuada, como muestra la Figura 2.

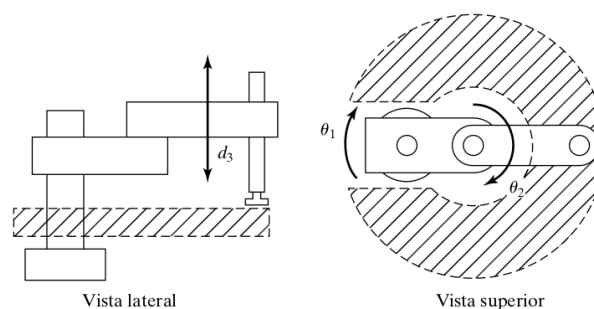


Fig. 1: Diseño conceptual del primer robot SCARA.
[Craig, 2005]

a. Parámetros Denavit-Hartenberg.

Un procedimiento para la solución de los problemas cinemáticos directo e inverso de robots requiere de la aplicación del procedimiento desarrollado por Denavit y Hartenberg (DH), lo cual implica el cálculo de cuatro parámetros $(a, \alpha; d, \theta)$, éstos determinan la posición relativa de un par de eslabones interconectados.

El procedimiento comienza determinando las direcciones de los pares cinemáticos, que en general se seleccionan de manera arbitraria. Note que es posible seleccionar dos sentidos



para el desplazamiento de un par cinemático. Para el ángulo de rotación de pares de revoluta se emplea la regla de la mano derecha respecto al sentido positivo de la dirección del par. En este procedimiento, se usará la variante proximal de la convención DH, la cual establece un sistema coordenado rígidamente unido a determinado eslabón en su primer extremo, i.e., el más cercano al sistema coordenado fijo. La Figura 2 muestra el modelo cinemático del robot SCARA con sus elementos rotulados y con los ejes z indicados. La determinación de los parámetros (a, α) , conocidos como los parámetros del eslabón, son la traslación y la rotación a lo largo y alrededor del eje z del primer sistema coordenado para describir el modelo del eslabón entre éste sistema coordenado y su consecutivo. La determinación de los parámetros (d, θ) , conocidos como los parámetros de la junta cinemática, son la traslación y rotación a lo largo del eje z del segundo sistema coordenado necesarios para determinar el modelo de la segunda revoluta respecto de la primera.

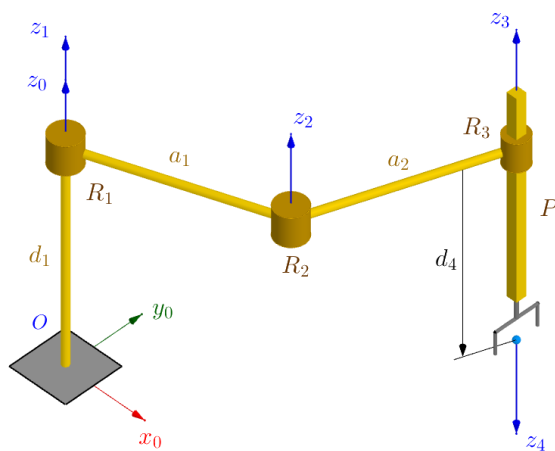


Fig. 2: Modelo del robot SCARA RRRP.

La composición de los cuatro movimientos establece el segundo sistema coordenado respecto del primero. De esta forma, en la Tabla 1 se especifican todos los parámetros DH para pasar de un sistema coordenado al siguiente.

En la Tabla 1 las coordenadas generalizadas del sistema son aquellas variables que presentan un asterisco, i.e., $\theta_1, \theta_2, \theta_3, d_4$, y son las variables de entrada del modelo cinemático. Los demás parámetros son parámetros de diseño y son constantes en la simulación del movimiento del modelo.

Tab. 1. Parámetros DH para el robot SCARA en su variante proximal.

i	a	α	d	θ
0	$a_0=0$	$\alpha_0=0^\circ$	-	-
1	$a_1=a_1$	$\alpha_1=0^\circ$	$d_1=d_1$	$\theta_1=\theta_1^*$
2	$a_2=a_2$	$\alpha_2=0^\circ$	$d_2=0$	$\theta_2=\theta_2^*$
3	$a_3=0$	$\alpha_3=180^\circ$	$d_3=0$	$\theta_3=\theta_3^*$
4	-	-	$d_4=d_4^*$	$\theta_4=0$

Los parámetros DH fueron introducidos en el software GeoGebra en la barra de entrada Input asignándoles un valor específico:

Input: $a_1=3$ ↵
 Input: $a_2=4$ ↵
 Input: $d_1=2$ ↵
 Input: $\theta_1=10^\circ$ ↵
 Input: $\theta_2=20^\circ$ ↵
 Input: $\theta_3=30^\circ$ ↵

GeoGebra registra este tipo de parámetros en la Vista Algebraica como números libres, no dependientes de otros objetos, los cuales disponen de barras deslizantes para cambiar su valor dinámicamente. Dichos controles se visualizan en la Vista Gráfica 2D al presionar el pequeño botón circular situado a la izquierda del nombre del objeto en la Vista Algebraica, como muestra la Figura 3.

b. Cinemática directa.

El problema de obtener la posición del actuador final del robot para determinados valores de los parámetros de entrada se conoce como el problema cinemático directo. Éste problema incluye la determinación de la postura del efector final, es decir, la posición y



orientación del último sistema coordenado, comúnmente ubicado en el efector final del robot, resolviéndose mediante el método gráfico o mediante métodos analíticos.

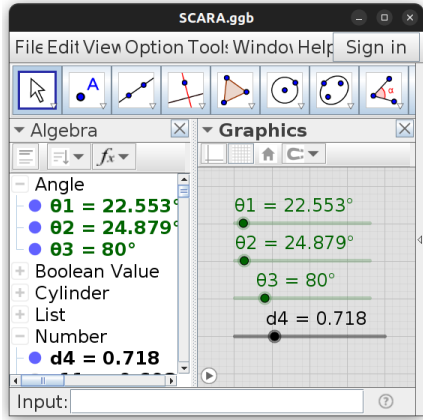


Fig. 3: Vista principal del software GeoGebra

En este trabajo se empleará el método analítico a través de los parámetros DH. El procedimiento de solución es directo para robots seriales, en donde se determina la postura de cada sistema coordenado mediante los parámetros DH aplicados a matrices de dimensión 4x4, conocidas como matrices de transformación homogéneas. En este sentido, la matriz de transformación homogénea usada para determinar la postura de un sistema coordenado a partir de su predecesor se calcula con una matriz de transformación especial que incluye el efecto de tornillo de dos transformaciones básicas, una aplicada en el eje x y la otra aplicada en el eje z, tal matriz es:

$$T_i^{i+1} = \begin{bmatrix} C\theta_{i+1} & -S\theta_{i+1} & 0 & a_i \\ C\alpha_i S\theta_{i+1} & C\alpha_i C\theta_{i+1} & -S\alpha_i & -d_{i+1}S \\ S\alpha_i S\theta_{i+1} & S\alpha_i C\theta_{i+1} & C\alpha_i & d_{i+1}C\alpha_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1)$$

Donde las letras C y S denotan las funciones coseno y seno, respectivamente. Aplicando la matriz de la Ec. (1) a cada par de sistemas coordenados consecutivos comenzando con el sistema coordenado fijo (x_0, y_0, z_0) , se obtiene el siguiente conjunto de matrices:

$$T_0^1 = \begin{bmatrix} C\theta_1 & -S\theta_1 & 0 & 0 \\ S\theta_1 & C\theta_1 & 0 & 0 \\ 0 & 0 & 1 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2)$$

$$T_1^2 = \begin{bmatrix} C\theta_2 & -S\theta_2 & 0 & a_1 \\ S\theta_2 & C\theta_2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2)$$

$$T_2^3 = \begin{bmatrix} C\theta_3 & -S\theta_3 & 0 & a_2 \\ S\theta_3 & C\theta_3 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_3^4 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & -d_4 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Cada una de las matrices de la Ec. (2) ubica la postura del sistema coordenado siguiente, denotado con el índice superior, respecto del sistema coordenado anterior, denotado con el índice inferior, referido a la letra T que denota transformación. Las primeras tres coordenadas del último vector columna de cada matriz de transformación representan el vector r de posición del origen coordenado $i+1$, superior, con respecto del origen anterior i , inferior. Las submatrices 3x3 de cada matriz de transformación que ocupa los elementos comprendidos en la región abarcada del elemento (1,1) al elemento (3,3), representa la matriz de orientación Q del sistema coordenado $i+1$ con respecto al sistema i .

Las transformaciones de la Ec. (2) quedan definidas con respecto al sistema coordenado fijo haciendo el producto de matrices siguiente:



$$\begin{aligned} T_0^1 &= T_0^1 \\ T_0^2 &= T_0^1 T_1^2 \\ T_0^3 &= T_0^2 T_2^3 \\ T_0^4 &= T_0^3 T_3^4 \end{aligned} \quad (3)$$

De esta forma, la postura de cada sistema coordinado adjunto rígidamente a su respectivo eslabón del robot se conoce. Estas posturas se calcularon en forma directa dentro del software GeoGebra. En este caso, las matrices de la Ec. (2), que dependen de los parámetros variables de la Tabla 1, se introdujeron usando matrices, las cuales se almacenan dentro del software como listas que contienen otras listas. Los objetos lista son un tipo de dato que almacenan una colección de otros objetos, en este caso otras listas de números. Por ejemplo, la matriz T_0^1 fue introducida como:

```
Input: T01={{cos(θ1), -sin(θ1), 0, 0},
            {sin(θ1), cos(θ1), 0, 0},
            {0, 0, 1, d1},
            {0, 0, 0, 1}}
```

Después, los cálculos de las matrices de la Ec. (3) se hicieron con simples productos de matrices, como se indica a continuación:

```
Input: T02=T01*T12 ↵
Input: T03=T02*T23 ↵
Input: T04=T03*T34 ↵
```

Los cálculos de cada matriz presentada en el conjunto de Ecs. (3) se muestran a continuación, donde para las funciones trigonométricas el subíndice 1 representa el ángulo θ_1 , el subíndice 12 representa la suma de ángulos $(\theta_1 + \theta_2)$, y el subíndice 123 es la suma de ángulos $(\theta_1 + \theta_2 + \theta_3)$.

$$T_0^1 = \begin{bmatrix} C_1 & -S_1 & 0 & 0 \\ S_1 & C_1 & 0 & 0 \\ 0 & 0 & 1 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\begin{aligned} T_0^2 &= \begin{bmatrix} C_{12} & -S_{12} & 0 & a_1 C_1 \\ S_{12} & C_{12} & 0 & a_1 S_1 \\ 0 & 0 & 1 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\ T_0^3 &= \begin{bmatrix} C_{123} & -S_{123} & 0 & a_1 C_1 + a_2 C_{12} \\ S_{123} & C_{123} & 0 & a_1 S_1 + a_2 S_{12} \\ 0 & 0 & 1 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\ T_0^4 &= \begin{bmatrix} C_{123} & S_{123} & 0 & a_1 C_1 + a_2 C_{12} \\ S_{123} & -C_{123} & 0 & a_1 S_1 + a_2 S_{12} \\ 0 & 0 & -1 & d_1 - d_4 \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{aligned} \quad (4)$$

La Vista Gráfica 3D de GeoGebra muestra el modelo geométrico del robot, cuya construcción se hizo al recuperar los vectores de posición r y los vectores columna que conforman la matriz Q de orientación de cada sistema coordinado adjunto a cada eslabón. Por ejemplo, la postura asociada a la matriz del efector final fue recuperada haciendo:

```
Input: r04=(T04(1,4), T04(2,4), T04(3,4)) ↵
Input: vx4=(T04(1,1), T04(2,1), T04(3,1)) ↵
Input: vy4=(T04(1,2), T04(2,2), T04(3,2)) ↵
Input: vz4=(T04(1,3), T04(2,3), T04(3,3)) ↵
```

Con esta información, el origen O_4 se grafica de forma directa junto con los vectores que conforman el sistema coordinado del efector final, los cuales fueron trasladados a este punto mediante el comando Translate, para construir adecuadamente el último sistema de coordenadas. Los demás sistemas coordinados se construyeron de manera similar. Al final, se obtuvo el modelo cinemático del robot, también conocido como modelo de alambre, construido al generar segmentos de recta entre los sistemas coordinados, empleando el comando Segment. Además, las representaciones de los pares cinemáticos, revolutas y prismático, fueron hechas mediante el comando Cylinder y Prism, como se observa en la Figura 3. El comando Angle sirvió para medir y representar el ángulo entre dos entidades.

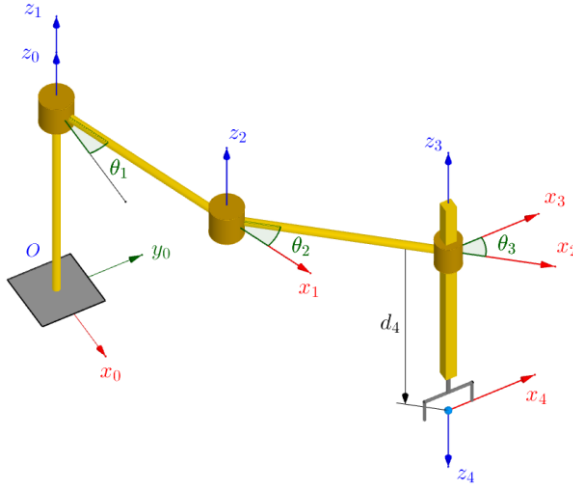


Fig. 4: Modelo SCARA en GeoGebra con sistemas coordenados y parámetros variables

El resultado es una construcción computarizada dentro del software GeoGebra altamente interactiva y dinámica, en donde los parámetros de diseño y las coordenadas generalizadas, i.e., ángulos de revolutas y el desplazamiento del prismático, varían en tiempo real.

c. Cinemática inversa.

La obtención de los desplazamientos de los pares cinemáticos a partir de una postura previamente conocida del efector final se llama problema cinemático inverso de posición. El problema no es trivial, y requiere de la solución de un conjunto no lineal de ecuaciones acopladas. En este caso, la matriz T_0^4 es conocida y los ángulos θ_1 , θ_2 , θ_3 , y el desplazamiento d_4 deben calcularse, es decir, la última matriz del conjunto de matrices de la Ec. (4) debe resolverse para éstas variables, de tal forma que:

$$H = T_0^4 \quad (5)$$

Donde

$$H = \begin{bmatrix} h_{11} & h_{12} & 0 & h_{14} \\ h_{21} & h_{22} & 0 & h_{24} \\ 0 & 0 & -1 & h_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Es la matriz de la postura de entrada, la cual no es general ya que sólo están permitidas tres traslaciones y una rotación en z_4 , esto reduce los cálculos en el modelo. La matriz T_0^4 es la última matriz de la Ec. (4) que involucra las variables a encontrar.

La simulación de la cinemática inversa de posición fue construida en un nuevo proyecto GeoGebra, en donde el primer paso consistió en definir una postura variable para el efector final del robot SCARA por medio del método gráfico y las herramientas del software. En este caso, el origen O_4 se trazó como un punto libre, y los vectores x_4 e y_4 fueron dibujados de manera que permiten rotar el sistema coordenado alrededor de z_4 . De esta forma el sistema coordenado $\{O_4\}$ puede simular las posturas permitidas del robot. Esto se logró mediante la restricción del punto final del vector x_4 en el círculo paralelo al plano xOy Plane con centro O_4 , y la restricción del punto final de y_4 en el rayo perpendicular a x_4 y z_4 . Posteriormente, el cálculo de la matriz Q se realizó al proyectar cada vector del sistema coordenado $\{O_4\}$ en cada vector del sistema coordenado fijo, por ejemplo, para el vector x_4 las proyecciones son $h_{11} = x_4 \cdot x_0$, $h_{21} = x_4 \cdot y_0$, $h_{31} = x_4 \cdot z_0 = 0$, que conforman la primera columna de la matriz de rotación. El vector de posición r_4 es el vector asociado al origen O_4 . De esta forma, la matriz H es conocida. Posteriormente, los parámetros DH se introdujeron de forma semejante a la sección anterior.

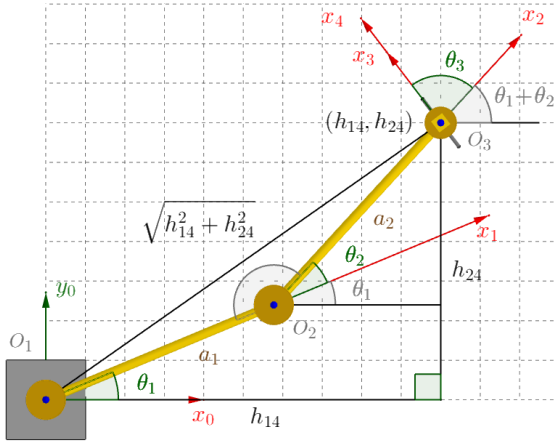


Fig. 5: Vista superior del robot

De la Figura 5, es posible encontrar las siguientes relaciones:

$$\begin{aligned} a_1 C_1 + a_2 C_{12} &= h_{14} \\ a_1 S_1 + a_2 S_{12} &= h_{24} \end{aligned} \quad (6)$$

Al restar la primera relación de la Ec. (6) multiplicada por $a_2 S_2$ de la segunda relación de la Ec. (6) multiplicada por $a_1 + a_2 C_2$ se obtiene la siguiente ecuación:

$$S_1 = \frac{(a_1 + a_2 C_2)h_{24} - a_2 S_2 h_{14}}{h_{14}^2 + h_{24}^2} \quad (7)$$

Siguiendo un procedimiento similar, se obtiene la siguiente relación:

$$C_1 = \frac{(a_1 + a_2 C_2)h_{14} + a_2 S_2 h_{24}}{h_{14}^2 + h_{24}^2} \quad (8)$$

Por tanto, el ángulo θ_1 queda definido mediante la Ec. (7) y la Ec. (8) al usar la función arco tangente de dos parámetros:

$$\theta_1 = \text{atan}(S_1, C_1) \quad (9)$$

Para el ángulo θ_2 se aplica la ley de cosenos al lado $\sqrt{h_{14}^2 + h_{24}^2}$ del triángulo $\{O_1, O_2, O_3\}$, resultando en la relación:

$$\cos(\theta_2) = \frac{h_{14}^2 + h_{24}^2 - a_1^2 - a_2^2}{2a_1 a_2} \quad (10)$$

Mediante la identidad $\sin(\theta_2)^2 + \cos(\theta_2)^2 = 1$ la función $\sin(\theta_2)$ se calcula como:

$$\sin(\theta_2) = \pm \sqrt{1 - \cos^2(\theta_2)} \quad (11)$$

Por lo tanto, el ángulo θ_2 queda definido al usar nuevamente la función tangente inversa de dos parámetros:

$$\theta_2 = \text{atan}(S_2, C_2) \quad (12)$$

El ángulo $\theta_{123} = \theta_1 + \theta_2 + \theta_3$ se determina mediante los elementos h_{11} y h_{21} de la matriz H de la Ec. (5) y los elementos correspondientes de la matriz T_0^4 de la Ec. (4). De esta forma, el ángulo θ_{123} es:

$$\theta_{123} = \text{atan}(h_{21}, h_{11}) \quad (13)$$

Por consiguiente, el ángulo θ_3 queda definido como:

$$\theta_3 = \theta_{123} - \theta_1 - \theta_2 \quad (14)$$

Finalmente, el desplazamiento d_4 es determinado al igualar h_{34} de la Ec. (5) con el elemento correspondiente de la matriz T_0^4 de la Ec. (4). Entonces, el desplazamiento del par prismático es:

$$d_4 = d_1 - h_{34} \quad (15)$$

Las Ecs. (7) a (15) fueron introducidas en GeoGebra mediante la barra de entrada para posteriormente calcular las matrices de transformación mostradas en la Ec. (4). Así, la postura de cada sistema coordenado fue calculada, y los correspondientes eslabones fueron dibujados con los mismos procedimientos y comandos descritos en la sección anterior. De esta forma, el usuario puede trasladar el origen O_4 con el puntero del ratón y rotar el sistema coordenado correspondiente al mover el punto final del



vector x_4 . Más aún, es posible definir una trayectoria y restringir el punto O_4 en esta, de tal manera que al activar la animación el robot la siga, como muestra la Figura 6.

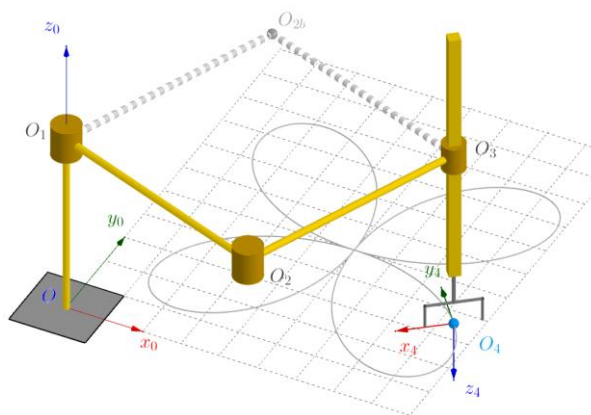


Fig. 6: Solución de la cinemática inversa de posición

III. RESULTADOS Y DISCUSIÓN

Las Figuras 1, 3, 5 y 6 son el resultado de la implementación de los procedimientos de análisis descritos en este trabajo, que demuestran cómo aplicar el software GeoGebra para la simulación de la cinemática de posición del manipulador SCARA RRRP, tanto para el análisis de posición directo como para el análisis de posición inverso. Los procedimientos de análisis usados para cada caso se basan en los parámetros DH en su variante proximal, los cuales fueron repasados e implementados de manera directa dentro del software. El modelo visual del robot fue hecho a través de las herramientas que proporciona el software. De esta forma, resultaron dos simulaciones altamente interactivas y dinámicas en donde el usuario experimenta el comportamiento de la simulación al modificar los parámetros de entrada o al generar la animación del robot. Cabe resaltar que, comparado con otros paquetes de software CAD/CAE comerciales, no es necesario el modelado 3D de las piezas que conforman al robot, en su lugar, elementos geométricos primitivos fueron

usados, e.g., puntos, líneas, rayos, círculos, arcos, y vectores. Además, tampoco fue requerido conocer algún lenguaje de programación.

IV. CONCLUSIONES

En este trabajo fue demostrado cómo analizar y construir simulaciones interactivas y dinámicas a través del software GeoGebra para el robot SCARA RRRP. Las simulaciones resultantes comprueban la teoría básica de la robótica al experimentar los efectos de los cambios de los valores de los parámetros de entrada. De esta forma, es posible utilizar la herramienta GeoGebra para la construcción de simulaciones interactivas de robots con fines educativos o de investigación.

REFERENCIAS

- Driscoll, M., & Confrey, J. (1985). Teaching Mathematics: Strategies that Work. Teachers Writing to Teachers Series. Chelmsford, MA: Education Resources Information Center (ERIC).
- Kortenkamp, U. H., & Richter-Gebert, J. (1998). Geometry and education in the internet age. *ED-MEDIA/EL TELECOM 98 World Conference on Educational Multimedia and Telecommunications* (págs. 790-799). Freiburg, Germany: ERIC.
- Kortenkamp, U. H. (1999). *Foundations of dynamic geometry*. Ph.D. thesis, Swiss Federal Institute of Technology, Zürich.
- Lehrer, R., & Chazan, D. (1988). *Designing learning environments for developing understanding of geometry and space*. NY: Routledge.
- King, J., & Schattschneider, D. (1997). *Geometry Turned on*:



- dynamicsoftware in learning, teaching, and research.
- Roanes-Lozano, E., & Solano-Macías, C. (2023). Some reflections about the success and bibliographic impact of the dynamic geometry system geogebra. *Mathematics in Computer Science*, 17(2), 11.
- Chivai, C. H., Soares, A. A., & Catarino, P. (2022). Application of geogebra in the teaching of descriptive geometry: Sections of solids,. *Mathematics* , 10(17), 3034.
- Tomić, M. K., Aberšek, B., & I. Pesek. (2019). Geogebra as a spatial skills training tool among science, technology engineering and mathematics students. *Computer applications in engineering education* , 27(6), 1506-1517.
- Helsa, Y., Suparman, S., Juandi, D., Turmundi, T., & Ghazali, M. B. (2023). A meta-analysis of the utilization of computer technology in enhancing omputational thinking skills: Direction for mathematics learning. *International Journal of Instruction*, 16(2), 735-758.
- Türkoğlu, H., & Yalçınalp, S. (2024). Investigating problem-solving behaviours of university students through an eye-tracking system using geogebra in geometry: A case study. *Education and Information Technologies*.
- Macho Mier, E., Urízar Arana, M., Petuya Arcocha, V., & Hernández Frías, A. (2021). Improving Skills in Mechanism and Machine Science Using GIM Software.
- Müller, M., Hüsing, M., Beckermann, A., & Corves, B. (2020). Linkage and cam design with MechDev based on non-uniform rational B-splines. *Machines*, 8(1), 5.
- Beckermann, A., Müller, M., Hüsing, M., & Corves, B. (2019). Analytical Kinematic Analysis of Cam Mechanisms in MechDev. In *Advances in Mechanism and Machine Science: Proceedings of the 15th IFToMM World Congress on Mechanism and Machine Science 15* (pp. 429-438). Springer International Publishing.
- Sagar, K., Ramadoss, V., Zlatanov, D., & Zoppi, M. (2020, October). STORM: Screw theory toolbox for robot manipulator and mechanisms. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (pp. 7233-7240). IEEE.
- Suwarno, D. U., & Putri, R. C. E. (2024). 2D and 3D robot arms simulation using GeoGebra. In *Journal of Physics: Conference Series* (Vol. 2684, No. 1, p. 012002). IOP Publishing.
- Husty, M., & Zsombor-Murray, P. (2019). On the Use of Gröbner Bases in a Robotics Course. In *New Trends in Educational Activity in the Field of Mechanism and Machine Theory: 2014-2017* (pp. 20-28). Springer International Publishing.
- Makino, H., y Furuya, N. (1980). *Selective compliance assembly robot arm*, Proceedings of the First International Conference on Assembly Automation, Brighton, March 25-27.
- Makino, H., (1982), *Assembly Robot*, US4341502.
- Craig, J. J. (2005), *Introduction to Robotics*. Prentice Hall, NJ.



Refactorización de Diagramas de Clase UML Utilizados en Estrategias de Enseñanza-Aprendizaje

Desiderio-Patiño Arelia ^{*1}, Fragoso-Díaz Olivia Graciela ¹, Rojas-Pérez Juan Carlos ¹

¹ Departamento de Ciencias de la Computación, TecNM/CENIDET, Cuernavaca, Morelos 62490 México.

* Autor de correspondencia: m23ce040@cenidet.tecnm.mx

Resumen. Los métodos de enseñanza-aprendizaje utilizados actualmente en los modelos de educación basados en competencia incluyen entre otros, el aprendizaje basado en proyectos, aprendizaje colaborativo, aprendizaje basado en problemas, entre otros. Estas estrategias buscan situar al estudiante en un ambiente semejante al laboral. Cuando estas estrategias se aplican en programas relacionados al diseño y desarrollo de software, es muy probable que los estudiantes requieran utilizar diagramas de clase UML para modelar la estructura del software que diseñarán o estudiarán. Con el auge del internet, los estudiantes acuden a éste para obtener ideas o ejemplos, pero sin verificar la calidad de estos recursos para fines de aprendizaje. En el área de desarrollo de software, para garantizar la calidad desde fases iniciales de diseño y creación de software robusto y bien estructurado, se propone una herramienta web que contribuye a la evaluación de diagramas de clases en el lenguaje de modelado unificado (UML, por sus siglas en inglés) tomando como referencia un modelo de calidad de un trabajo antecedente. Además, en caso de que un recurso de aprendizaje obtenga una valoración negativa, la herramienta permite reestructurar el diagrama de clases. De esta manera, se previenen defectos que pudieran afectar la calidad de los diagramas de clases usados como recursos de aprendizaje asegurando que los materiales empleados en cursos relacionados con el diseño y desarrollo de software cumplan con los estándares de calidad en la ingeniería de software.

Palabras clave. *Diagramas de clases, recurso de aprendizaje, powerpoint, xml.*

I. INTRODUCCIÓN

La evaluación de la calidad de los diagramas de clases UML utilizados como recursos de aprendizaje es fundamental para garantizar su efectividad en la enseñanza de conceptos de programación orientada a objetos. Estos diagramas de clases UML son una herramienta clave para representar la estructura y las relaciones de los componentes de un sistema, sirviendo como guía visual en las fases iniciales de diseño (Fowler, 2004). Sin embargo, cuando estos diagramas se utilizan como recursos de enseñanza sin una revisión adecuada

de su calidad, pueden inducir a errores de comprensión y diseño, afectando negativamente la adquisición de conocimientos en el desarrollo de software.

En este sentido, la evaluación de la calidad de estos diagramas es crucial para garantizar que cumplan con los estándares de diseño y principios de ingeniería de software. Un proceso riguroso de evaluación ayuda a identificar posibles deficiencias, como la falta de coherencia, la ambigüedad en la representación de relaciones, o el uso incorrecto de elementos del lenguaje UML. De este modo, la verificación de la calidad no solo



asegura la correcta transmisión de conocimientos, sino que también contribuye a la formación de estudiantes y profesionales que siguen buenas prácticas en el desarrollo de software.

Dentro de las estrategias de enseñanza-aprendizaje que pueden requerir el uso de diagramas de clases UML, se tiene el aprendizaje basado en proyectos donde los estudiantes deben crear diagramas de clase UML para modelar la estructura del software que desean diseñar (Barron, 2008).

En el método de aprendizaje colaborativo, los estudiantes trabajan en equipos e implementan los diagramas de clase UML para discutir y refinar el diseño de los componentes de software (Johnson, 2009).

Otra estrategia es el aprendizaje basado en problemas, donde los estudiantes utilizan los diagramas de clases UML para modelar la solución a un determinado problema (Savery, 2015). Además, se podrían mencionar el uso de estudios de caso (Yin, 2017) y las simulaciones de entornos empresariales (Kolodner, 2003).

De lo anterior, se puede observar que los diagramas de clase UML como recursos de aprendizaje se pueden implementar en varias estrategias de enseñanza-aprendizaje, por lo que toma relevancia la necesidad de contar con una herramienta que permita verificar la calidad de estos recursos.

II. TRABAJOS RELACIONADOS

Para llevar a cabo este trabajo, se tuvieron en cuenta estudios relacionados con la calidad, la evaluación y la reestructuración de diagramas de clases UML, los cuales se mencionan a continuación.

a. Atributos de calidad en recursos de aprendizaje.

Varias regiones de países desarrollados de Europa y Norteamérica se están favoreciendo del crecimiento en el acceso a internet, siendo la educación un tema favorecido por ello, concretamente en (Alenezi, 2020) se destaca la importancia de la educación en línea en estos

lugares, y concluye que existe una fuerte correlación entre el uso de recursos de aprendizaje de calidad y el desempeño tanto de estudiantes como de docentes. Evaluar un recurso de aprendizaje requiere considerar su naturaleza específica; en este caso, los diagramas de clases UML pueden analizarse tanto como representaciones gráficas como elementos que deben cumplir con las normas y estándares de ingeniería de software.

Aun cuando la evaluación se puede realizar desde diferentes enfoques, en (Abdulrahman, 2020) se propone un marco para evaluar la calidad de recursos de aprendizaje multimedia que abarca no solo imágenes y texto, como en los diagramas de clases UML, sino también otros medios como audio, video y animaciones 3D. El autor subraya que la evaluación no debe limitarse al contenido del recurso, sino también considerar las características de la audiencia.

Por su parte, los autores de (Pasichnyk, 2023) incorporan aspectos como la capacidad de modificación, la arquitectura y la funcionalidad del recurso, además de asegurar que cumpla con los estándares educativos. Para la evaluación, utiliza un enfoque basado en lógica difusa.

b. Evaluación de la calidad de diagramas de clases.

La evaluación de la calidad de los diagramas de clases UML abarca varios aspectos, incluida la complejidad estructural, el diseño y la corrección. Los investigadores han propuesto diferentes enfoques para evaluar estas cualidades. En (Xu, 2004) introdujeron una medida de complejidad estructural basada en gráficos de dependencia de clases ponderados y distancia de entropía.

En párrafos anteriores se había establecido que la audiencia debe considerarse para la evaluación, diferentes usuarios podrán tener diferentes ponderaciones al momento de evaluar. En (Karasneh, 2015) compararon las evaluaciones de expertos y estudiantes de seis atributos de calidad y encontraron diferencias significativas entre los dos grupos.



Un ejemplo de las bondades de utilizar herramientas informáticas se encuentra en (Bergström, 2022) donde desarrollaron un método basado en aprendizaje automático para evaluar la calidad del diseño utilizando características extraídas de imágenes de diagramas. Este enfoque automatizado tiene aplicaciones potenciales en la industria, la educación y el diseño de algoritmos.

Otro ejemplo de un sistema sistematizado es el de (Ali, 2007) quienes diseñaron un sistema de evaluación llamado UCDA que evalúa los diagramas de clases en función de la estructura, la corrección y el uso del lenguaje.

Estos estudios demuestran colectivamente la importancia de la evaluación de la calidad en los diagramas de clases UML y ofrecen diversas metodologías de evaluación, que van desde la evaluación manual de expertos hasta técnicas automatizadas de aprendizaje automático.

c. Reestructuración de diagramas de clases.

Se encontró un sistema con un enfoque dual realizado por (Hemati, 2024) donde se presenta una herramienta de reestructuración que considera tanto el código como el modelo UML, reconociendo que ciertas tareas son más sencillas de realizar en uno u otro ámbito. Esta herramienta interactiva asiste al desarrollador en la detección de errores de diseño a través del análisis del código y proporciona ayudas visuales para facilitar la selección de soluciones viables. Tras la reestructuración, es el propio desarrollador quien debe validar las modificaciones sugeridas, asegurando su adecuación al diseño.

Técnicas que están en boga, como las Redes Neuronales y Ciencia de Datos, han sido implementadas en (Sidhu, 2022) quienes proponen un sistema alternativo que evalúa diagramas UML centrándose en las descomposiciones funcionales en lugar de errores específicos. Utilizar técnicas novedosas permite mejorar la representación de los modelos y reducir los ciclos de refactorización.

Otro ejemplo del uso de estas técnicas se tiene en (Alazba, 2024) que también exploran el uso de

Inteligencia Artificial para la detección de errores de diseño en diagramas UML, aunque destacan la necesidad de intervención manual en situaciones con grandes volúmenes de información. Para abordar este desafío, proponen un enfoque no supervisado que optimiza el proceso de evaluación.

III. ATRIBUTOS DE CALIDAD

Los atributos de calidad contemplados en este trabajo se toman de un trabajo antecedente, el cual define un Modelo de Calidad (Sandoval, 2023). El modelo propuesto que se muestra en la Fig. 1, evalúa diagramas de clases UML usados como recursos de aprendizaje.

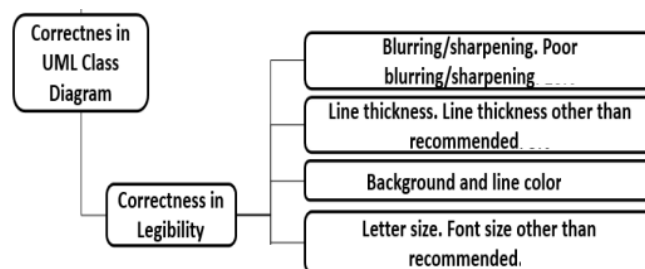


Fig. 1 Atributos del Factor de Legibilidad.

Este trabajo se centra en la correctitud en legibilidad, es decir, en evaluar cuán comprensible es el diagrama de clases UML desde una perspectiva visual. La importancia de que los recursos de aprendizaje tengan una buena calidad visual radica en motivar a los estudiantes a emplearlos, ya que se ha reportado en la literatura que utilizar diagramas de clases UML es favorable en el entorno profesional (Dzidek, 2008).

En el modelo antes mencionado, el atributo principal para medir el nivel de calidad se denomina Correctitud. Este atributo se divide en dos factores principales: Correctitud en dominio UML y Correctitud en Legibilidad.

Nos centraremos en el factor de correctitud en legibilidad, el cual se compone de otros parámetros: nitidez del diagrama, debido a que un diagrama ilegible es inútil. El segundo parámetro se refiere al grosor de los



bordes de las clases, destacando que los bordes demasiado gruesos pueden oscurecer el contenido y dificultar la claridad visual.

El tercer parámetro se enfoca en la elección de colores, recomendando principalmente el uso de un fondo blanco con trazos en negro para maximizar el contraste y facilitar la lectura. El cuarto parámetro evalúa el tamaño de la fuente, que debe ser equilibrado, evitando tamaños demasiado pequeños o grandes, y priorizando el uso de texto plano para mejorar la legibilidad.

Tab. 1. Atributos de calidad evaluados.

Parámetro		% original	% actual
Difuminación/ Nitidez	Difuminación/ Nitidez	10	34
Grosor de línea	Clases	2	7
	Relaciones	3	10
Color de fondo y líneas	Color de fondo	4	14
	Color de líneas	4	14
Tamaño de fuente	Tamaño de fuente	6	21

Los parámetros mencionados anteriormente se listan en la Tab. 1, donde cada uno de ellos tienen una ponderación específica. El % original corresponde a la ponderación propuesta por (Sandoval, 2023) y el % actual corresponde a la ponderación utilizada en este trabajo. El porcentaje general obtenido indica el nivel de correctitud del diagrama de clases UML, donde 100% representa la puntuación máxima.

IV. DESARROLLO

El desarrollo de este trabajo se basa en el modelo de calidad descrito en la Sección III, el cual se utiliza para evaluar la calidad de los recursos de aprendizaje. Actualmente, la herramienta se encuentra en una fase inicial de desarrollo, enfocándose únicamente a aspectos relacionados con legibilidad. La integración del resto de los elementos contemplados en el modelo de calidad aún está pendiente.

La herramienta utiliza tecnologías web y su función principal es corregir diagramas de clases UML almacenados en archivos de PowerPoint en formato pptx. La metodología empleada se basa en la extracción de metadatos del archivo digital, los cuales son procesados para realizar la evaluación de los diagramas.

En la página de inicio, una vez autenticado el usuario, puede cargar en el servidor un archivo digital en formato pptx que contiene el recurso de aprendizaje que desea evaluar. Tras cargar el archivo, el sistema queda preparado para realizar la evaluación.

En la Fig. 2 se muestran algunas de las opciones disponibles, donde después del proceso de evaluación, el usuario puede seleccionar los atributos específicos que desea modificar. Si no se selecciona ningún atributo, la herramienta no generará un nuevo archivo. Sin embargo, si el usuario elige al menos un atributo a corregir, el sistema generará un nuevo archivo pptx con el recurso de aprendizaje modificado.

El archivo PowerPoint actualizado estará disponible para que el usuario lo descargue. El usuario podrá optar por iniciar un nuevo proceso para otro recurso de aprendizaje o cerrar sesión.

La aplicación web se ejecuta sobre un servidor Apache Tomcat 8, utilizando MySQL Server 5.5 como gestor de bases de datos. Para su desarrollo, se implementó el patrón Modelo-Vista-Controlador (MVC), donde la capa del modelo fue programada en Java 1.8, las vistas fueron construidas con JavaServer Pages (JSP), y el controlador se desarrolló utilizando servlets en combinación con Dynamic Web Module 3.1.

Los servlets desarrollados aportan dinamismo al sistema web, gestionando funciones clave como el manejo de sesiones de usuarios, la carga y descarga de recursos de aprendizaje, y la interacción con la clase Java encargada de evaluar y modificar los recursos. La herramienta almacena un solo recurso de aprendizaje en el servidor de forma temporal, el cual se elimina automáticamente al cerrar la sesión del usuario, procurando no saturar el almacenamiento del servidor.



Las principales funcionalidades de la herramienta son: cargar archivo, evaluar archivo, procesar archivo y descargar archivo. Cada una de estas acciones cuenta con su propia vista específica.



Fig. 2: Vista "procesar archivo" de la herramienta.

La herramienta permite al usuario seleccionar los parámetros de calidad a reestructurar, proporcionando flexibilidad para ajustar el recurso de aprendizaje según sus necesidades.

V. RESULTADOS

Como ejemplo del funcionamiento de la herramienta, en la Fig. 3 se presenta un diagrama de clases que no cumple con los criterios de legibilidad establecidos. El diagrama mostrado incumple características consideradas de acuerdo con los criterios de correctitud, como son el grosor de las relaciones entre clases. Estas elecciones de diseño pueden causar confusión, lo que compromete su efectividad como recurso de aprendizaje, al dar más énfasis a unas líneas entre relaciones que otras, lo que se puede prestar a malas interpretaciones.

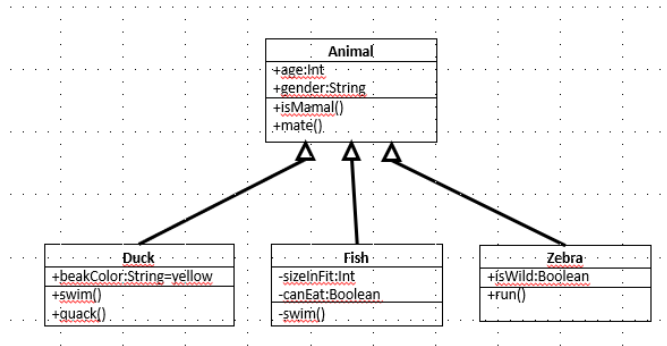


Fig. 3: Diagrama de clase sin atributos de calidad

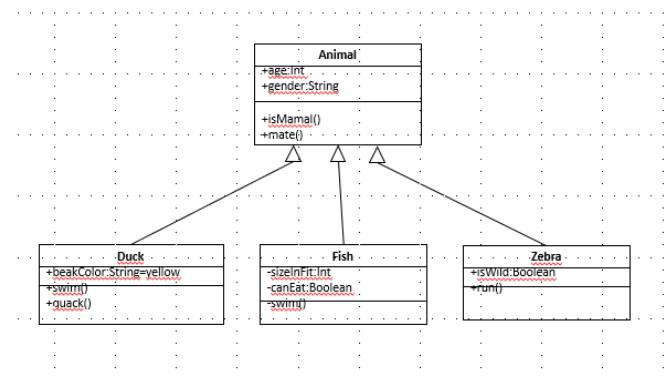


Fig. 4: Diagrama de clase después de ser procesado

En la Fig. 4 se muestra el resultado de realizar la reestructuración al recurso de aprendizaje. Se observa que uno de los principales cambios es la modificación del grosor de las relaciones entre clases, que originalmente tenían un mayor grosor, el cual se ha disminuido, mejorando significativamente la claridad visual del diagrama, lo que también contribuye a mejorar la legibilidad del recurso de aprendizaje.

VI. CONCLUSIONES

La herramienta actualmente permite reestructurar aspectos visuales de diagramas de clases UML, características enfocadas al criterio de legibilidad.

El sistema permite evaluar las características visuales de los elementos que componen un diagrama de clases UML y a solicitud del usuario, ajustarlas conforme a un modelo de calidad enfocado para recursos de aprendizaje. Al momento, la herramienta se ha centrado en mejorar aspectos de legibilidad en los diagramas de clases UML, facilitando la creación de recursos de aprendizaje más



claros y accesibles para los usuarios. Esto contribuye a que quienes consumen estos recursos puedan concentrarse en el contenido esencial, evitando distracciones por elementos gráficos innecesarios. Como parte de los planes a futuro, se incluirá la capacidad de evaluar la correcta implementación de clases, herencias y relaciones dentro del contexto de la programación orientada a objetos, mejorando aún más la utilidad de los diagramas de clases UML como recursos de aprendizaje.

REFERENCIAS

- Abdulrahman, M. D., Faruk, N., Oloyede, A. A., Surajudeen-Bakinde, N. T., Olawoyin, L. A., Mejabi, O. V., & Azeez, A. L. (2020). Multimedia tools in the teaching and learning processes: A systematic review. *Heliyon*, 6(11).
- Alazba, A., Aljamaan, H., & Alshayeb, M. (2024). Automated detection of class diagram smells using self-supervised learning. *Automated Software Engineering*, 31(29).
- Alenezi, A. (2020). The role of e-learning materials in enhancing teaching and learning behaviors. *International Journal of Information and Education Technology*, 10(1), 48-56.
- Ali, N. H., Shukur, Z., & Idris, S. (2007). A design of an assessment system for UML class diagram. En *2007 International Conference on Computational Science and its Applications (ICCSA 2007)* (pp. 539-546), Kuala Lumpur, Malaysia.
- Barron, B. & Darling-Hammond, L. (2008). Teaching for meaningful learning: A review of research on inquiry-based and cooperative learning. *Edutopia*.
- Bergström, G., Hujainah, F., Ho-Quang, T., Jolak, R., Rukmono, S. A., Nurwidiantoro, A., & Chaudron, M. R. V. (2022). Evaluating the layout quality of UML class diagrams using machine learning. *Journal of Systems and Software*, 192.
- Dzidek, W. J., Arisholm, E., & Briand, L. C. (2008). A realistic empirical evaluation of the costs and benefits of UML in software maintenance. *IEEE Transactions on Software Engineering*, 34(3), 407-432.
- Fowler, M. (2004). *UML Distilled: A Brief Guide to the Standard Object Modeling Language* (3rd ed.). Addison-Wesley.
- Hemati Moghadam, I., Ó Cinnéide, M., Sardarian, A., & Zarepour, F. (2024). Model-based source code refactoring with interaction and visual cues. *Journal of Software: Evolution and Process*, 36(5).
- Johnson, D. W., & Johnson, R. T. (2009). An educational psychology success story: Social interdependence theory and cooperative learning. *Educational researcher*, 38(5), 365-379.
- Karasneh, B., Stikkolorum, D. R., & Larios, E. (2015). Quality assessment of UML class diagrams: A study comparing experts and students. En *EduSymp@MoDELS*.
- Kolodner, J. L., Camp, P. J., Crismond, D., et al. (2003). Problem-based learning meets case-based reasoning in the middle-school science classroom: Putting learning by design into practice. *The Journal of the Learning Sciences*, 12(4), 495-547.
- Pasichnyk, V., Kunanets, N., Yunchyk, V., Khomyak, M., Yatsyuk, S., Muliar, V., & Fedonuyk, A. (2023). Model of the recommender system for the selection of electronic learning resources. En *MoMLet+ DS* (pp. 344-355).
- Sandoval-Acosta, J. A. (2023). *Uso de reglas de dominio para evaluar recursos de servicios web de aprendizaje* (Tesis de doctorado). TecNM/CENIDET, Repositorio Institucional – TecNM/CENIDET, Morelos, México.
- Savery, J. R. (2015). Overview of Problem-based Learning: Definitions and Distinctions. *Interdisciplinary Journal of Problem-Based Learning*, 1(1).
- Sidhu, B. K., Singh, K., & Sharma, N. (2022). A machine learning approach to software model refactoring. *International Journal of Computers and Applications*, 44(2), 166-177.
- Xu, B., Kang, D., & Lu, J. (2004). A structural complexity measure for UML class diagrams. En M. Bubak, G. D. van Albada, P. M. A. Sloot, & J. Dongarra (Eds.), *Computational Science - ICCS 2004* (Vol. 3036). Berlin, Heidelberg: Springer.
- Yin, R. K. (2017). *Case study research and applications: Design and methods*. Sage publications.



Integración de sistema de adquisición de sonidos para análisis de frecuencias de resonancia en jitomates

Rojas-Mancera Erick^{*1}, Flores-Placencia Giovana², Delgado-Franco Estefanía³, Pulido-Ruiz Frida⁴, Balvantín-García Antonio⁵

^{1,2,3,4} Tecnológico Nacional de México/ITS de Purísima del Rincón, Departamento de Ingeniería Industrial, Blvd. Del Valle 2301, Col. Guardarrayas, Purísima del Rincón, Guanajuato, 36425, México.

⁵ Universidad de Guanajuato, Departamento de Ingeniería Mecánica Departamento, Carretera Salamanca - Valle de Santiago km 3.5 + 1.8 Comunidad de Palo Blanco, Salamanca 36787, Guanajuato, México

* Autor de correspondencia: erick.rm@purisima.tecnm.mx

Resumen. La adquisición de datos de un fenómeno físico a través de sistemas de medición y su correcto procesamiento es esencial para comprender y mejorar nuestro entorno. Particularmente, el procesamiento de la información para tener una mejor interpretación. El procesamiento de la información contenido en el sonido facilita la identificación de patrones que ayudan en la toma de decisiones. Una técnica común para analizar estos datos es la identificación de frecuencias mediante la transformada de Fourier. El sistema para adquirir información construido con Arduino UNO y un sensor (micrófono KY-037), está diseñado para examinar las frecuencias presentes en un sonido específico. Los datos adquiridos fueron procesados mediante la Transformada Rápida de Fourier (FFT, por sus siglas en inglés) y visualizados en un espectrograma de frecuencias. Se comprobó que el Arduino UNO junto con el sensor KY-037 puede medir con éxito la información a partir de sonidos emitidos por una bocina con frecuencias específicas.

Palabras clave. *Patrones, Sonido, Frecuencia, Transformada de Fourier.*

I. INTRODUCCIÓN

El sonido está compuesto por vibraciones que se propagan en forma de ondas a través de diferentes medios. Para analizar estas vibraciones, se utiliza la técnica de análisis de Fourier. Bobadilla, J., Gómez, P., y Bernal, J. (1999) emplearon la Transformada de Fourier para convertir señales del dominio temporal al dominio de la frecuencia, permitiendo el estudio de las voces de acuerdo con los patrones que presentan los sonogramas, y proporcionaron una demostración matemática útil para quienes se interesan en el formalismo de la transformada de Fourier (Bernal, Gómez, & Bobadilla, 1999). Pinares, R., Machaca, V., Lozano, F., Quispe, A., Ccopa, R., y Calsin, B. (2023) usaron espectroscopia infrarroja por transformada de Fourier (FTIR) para comparar la fibra de vicuña de Puno y Apurímac, descubriendo que, aunque las fibras químicamente

similares, presentaban diferencias en intensidad de color, intensidad de fibra y modulación (Pinares, y otros, 2023). Orsini, S. M. D., y Aparicio, G. P. P. (2021) caracterizaron los grupos funcionales en cuatro tipos de harina (fécula de maíz, almidón de yuca, harina de trigo con y sin endulcolorante) mediante FTIR, encontrando similitudes en la composición química entre la harina de trigo, la fécula de maíz y el almidón de yuca (Orsini & Aparicio, 2021). Pardo, J. D., Castro, J. A., Iseda, G. P., Torres, C., y Mattos, L. (2006) investigaron el reconocimiento automático de señales de habla utilizando la transformada de Fourier y redes neuronales (Pardo, Castro, Iseda, Torres, & Mattos, 2006). Tronelli, M. T. (2014) abordó el tema del muestreo de señales, es decir, la conversión de señales continuas en tiempo a señales discretas, destacando la importancia del teorema de muestreo en la digitalización de señales para



telecomunicaciones y codificación digital de sonido (Tronelli, 2014). Velásquez Vásquez, W. E., Díaz Aragon, B. A., y Gutiérrez Rincón, S. B. (2018) utilizaron espectrometría infrarroja por transformada de Fourier para caracterizar los materiales en mortero, proporcionando una herramienta para mejorar las auditorías y el control de calidad en la construcción (Velásquez Vásquez, Díaz Aragon, & Gutiérrez Rincón, 2018). Tainta, S., Erro, M. J., Garde, M. J., y Muriel, M. A. (2009) propusieron un sistema eléctricamente sintonizable para ajustar la velocidad de repetición en un tren de pulsos ópticos, utilizando un modulador electro-óptico y la transformada de Fourier para validar el sistema (Tainta, Erro, Garde, & Muriel, 2009). Tintaya, C. O. J. (2005) diseñó e implementó un sistema de adquisición de datos con una tarjeta de sonido de PC, que convierte señales analógicas en digitales y permite su análisis en el dominio del tiempo y de la frecuencia mediante FFT (Tintaya, 2005). Este sistema es útil para el procesamiento digital de audio, experimentos de laboratorio, y más. Sámano-Robles, R., Mendoza-Gómez, I., y Neri-Vela, R. (2002) presentaron una descripción de la DFT, su formulación discreta para métodos numéricos, y su optimización por computadora, demostraron que el uso de programas alternativos a MATLAB reduce el tiempo de procesamiento en un 90% o más (Sámano-Robles, Mendoza-Gómez, & Neri-Vela, 2022). Jaramillo Chamba, D., y Chuquimarca Jiménez, L. (2022) desarrollaron una técnica computacional para el análisis de una senoide, la identificación de un sistema Filtro de respuesta al Impulso (FIR, por sus siglas en ingles) y el análisis espectral de señales de audio, validando los algoritmos desarrollados para el análisis de señales digitales (Jaramillo Chamba & Chuquimarca Jiménez, 2022). En este trabajo, se desarrolló la adquisición de datos utilizando Arduino UNO y un micrófono KY-037, con procesamiento de datos mediante FFT, presentando los resultados en un espectro de frecuencias.

II. Teoría

La transformada de Fourier es una operación matemática que sirve para identificar las frecuencias contenidas en una señal. Convierte señales en el dominio del tiempo o espacio, al dominio de la frecuencia.

Si $x(t)$ es la función original, su transformada $X(f)$ será como se muestra en la Ec. (1):

$$x(f) = \int_{-\infty}^{\infty} x[t]e^{-j2\pi ft} dt \quad (1)$$

donde f es la frecuencia.

III. Desarrollo

Para el desarrollo de este trabajo se utilizó un Arduino UNO, las características se muestran en la Fig. 1. Además, un sensor para medir el sonido generado por una bocina a diferentes frecuencias como se muestra en la Fig. 2.

CARACTERÍSTICAS	IMAGEN
Modelo: Arduino Uno	
Microcontrolador: ATmega328	
Voltaje operativo: 5v	
Voltaje de entrada (recomendado): 7-12 v	
Pines de entradas/salidas digital: 14 (de las cuales 6 son salidas PWM)	
Pines de entradas analógicas: 6	
Memoria flash: 32 KB (ATmega328) de los cuales 0,5 KB es usado por Bootloader	
SRAM: 2 KB (ATmega328)	
EEPROM: 1 KB (ATmega328)	
Velocidad del reloj: 16 MHz	
Tensión de funcionamiento: 5 VDC	
Corriente DC por pin E/S: 40 mA	
Corriente DC para pin de 3.3V: 50 mA	
Dimensiones: 68,6 X 53,4 mm	
	DESCRIPCIÓN
	Arduino Uno es una placa electrónica basada en el microprocesador Atmega328

Fig. 1. Características del Arduino UNO.

CARACTERÍSTICAS	IMAGEN
Modelo: KY-037	
Tipo de sensor: Sonido	
Principio de funcionamiento: Micrófono electret	
Voltaje de alimentación: 3.3 a 5VDC	
Corriente típica en operación: 35 mA	
Tipo de interfaz: Analógica o digital	
Interfaz: 4 conexiones VSS-Voltaje de alimentación GND-Tierra o GND DO-Salida digital AO-Salida analógica	
Peso: 5 gramos	
Comparador: LM393	
Dimensiones: 35 X 15 X 14 mm	
	DESCRIPCIÓN
	El KY-037 Módulo sensor de sonido sirve para detectar cualquier tipo de sonido siendo su principal característica una alta sensibilidad.

Fig. 2. Características del sensor KY-037.

Los jitomates tienen una frecuencia de resonancia entre [300,400] kHz (Baltazar, Espina-Lucero, Ramos-Torres, & González-Aguilar, 2007). De acuerdo con las características del Arduino UNO (Fig. 1), tienen la capacidad de medir la frecuencia de resonancia del jitomate. La metodología que se siguió para la adquisición de datos es la siguiente:

1. Conectar el sensor KY-037 en la entrada analógica del Arduino UNO.
2. Conectar el Arduino UNO a la computadora y cargar el código para la adquisición de datos.



3. Generar sonidos a frecuencias específicas usando una bocina para adquirir los datos con el sensor KY-037.

La Fig. 3, muestra el sistema experimental para generar y medir sonido.

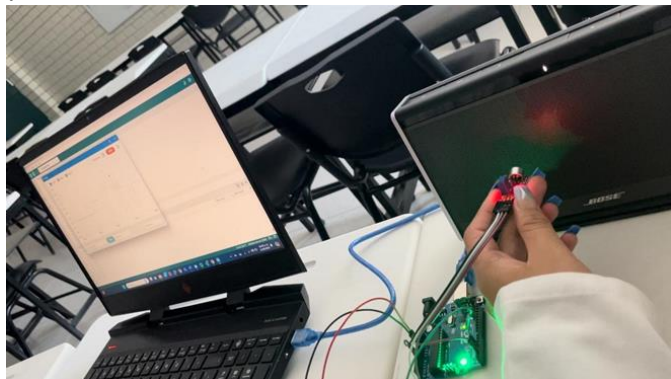


Fig. 3. Sistema experimental y toma de datos.

Además, se imprimió un sistema para golpear el jitomate y generar los valores de frecuencia de resonancia, la Fig. 4 muestra el esquema del funcionamiento del sistema. Es necesario que la esfera impresa en 3D, de masa constante (150 g.), mantenga un ángulo (90°) fijo con respecto a la vertical, tomando como referencia el jitomate, con el fin de determinar su dureza basándose en la frecuencia de resonancia generada por el impacto.

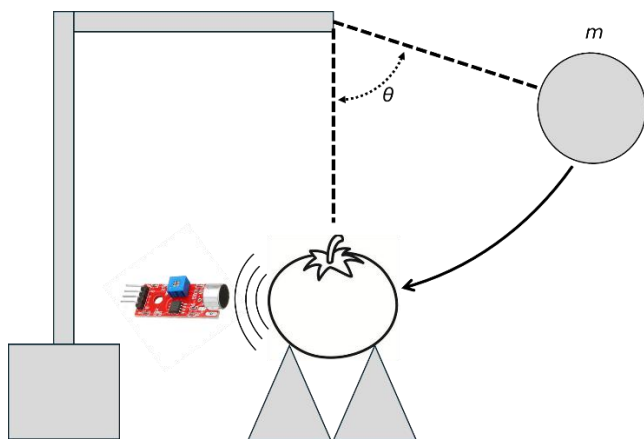


Fig. 4. Sistema para generar frecuencia de resonancia.

La Fig. 5 muestra el sistema experimental que golpeará al jitomate para generar una frecuencia de resonancia. LA esfera y el soporte fueron impresos en una impresora 3D, a 30% de densidad de impresión.

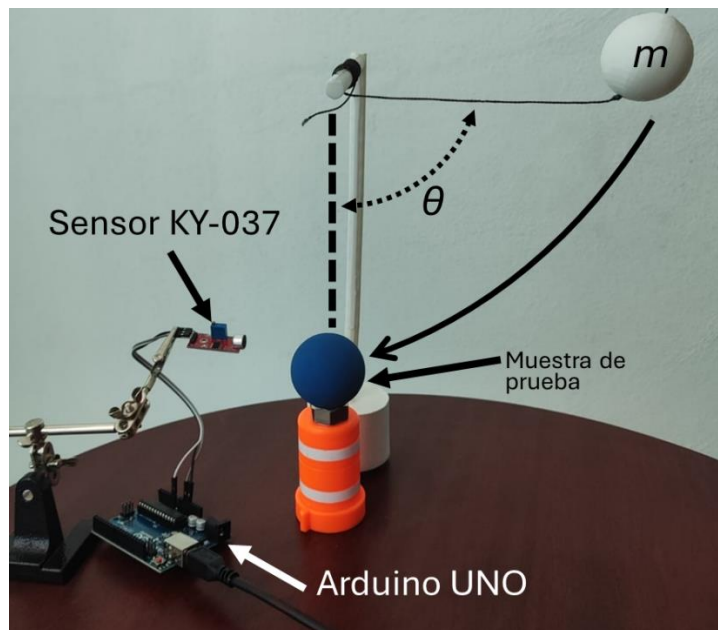


Fig. 5. Sistema experimental.

IV. Resultados

El resultado de adquirir datos con el Arduino UNO y el sensor KY-037 se muestran las Fig. 6,7 y 8. Son señales generadas con sonidos a 100 Hz, 200 Hz y 500 Hz, respectivamente. Se puede observar que conforme aumenta la frecuencia disminuye la amplitud de la señal.

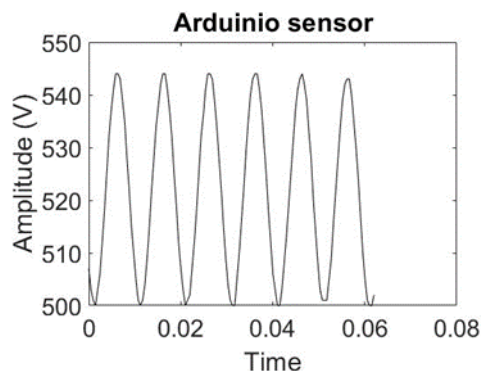


Fig. 6. Sonido generado a 100Hz.

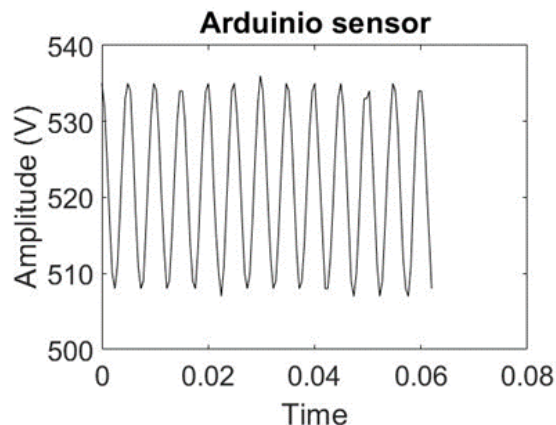


Fig. 7. Sonido generado a 200 Hz.

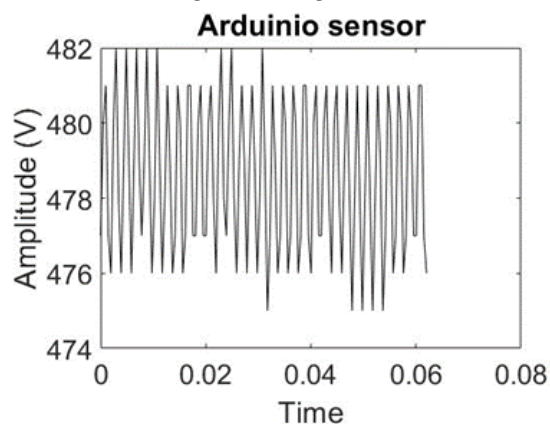


Fig. 8. Sonido generado a 500 Hz.

Las Fig. 9,10 y 11, son los espectros de frecuencias de las señales de las Fig. 6, 7 y 8, respectivamente.

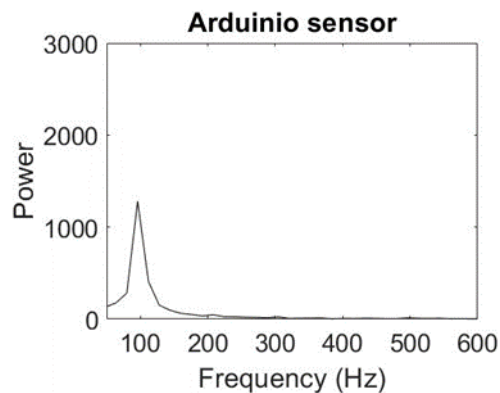


Fig. 9. Espectro de frecuencias de sonido a 100 Hz.

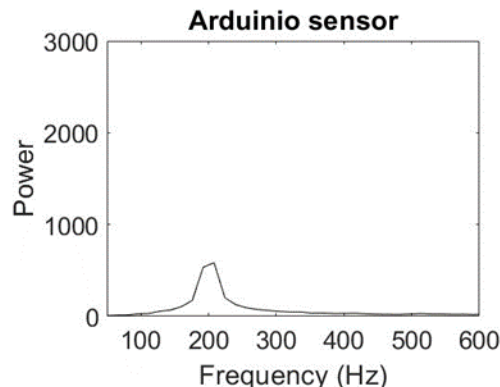


Fig. 10. Espectro de frecuencias de sonido a 200 Hz.

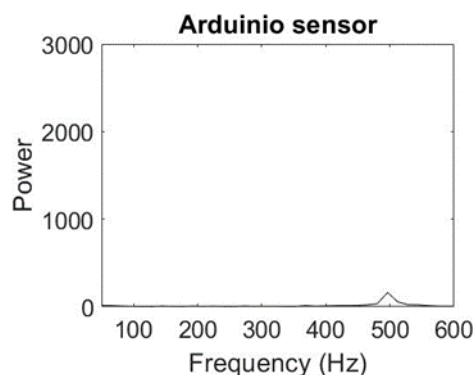


Fig. 11. Espectro de frecuencias de sonido a 500 Hz.

V. Conclusiones

Los resultados muestran que es posible obtener datos del sonido usando un Arduino UNO y un sensor KY-037. Además, se pueden procesar para obtener la trasformada de Fourier y representarla en un espectro de frecuencias.

VI. Trabajo futuro

El trabajo futuro es medir el sonido de los jitomates al ser golpeados con una masa (150 g.) a un ángulo específico (90°). Los sonidos generados por los jitomates serán medidos en un lapso de una semana para analizar el cambio de la frecuencia de resonancia.

REFERENCIAS

- Baltazar, A., Espina-Lucero, J., Ramos-Torres, I., & González-Aguilar, G. (2007). Effect of methyl jasmonate on properties of intact tomato fruit monitored with destructive and nondestructive tests. *Journal of food engineering*, 80(4), 1086-1095.



- Bernal, J., Gómez, P., & Bobadilla, J. (1999). Una visión práctica en el uso de la Transformada de Fourier como herramienta para el análisis espectral de la voz. *Estudios de fonética experimental*, 10, 75-105.
- Jaramillo Chamba, D., & Chuquimarca Jiménez, L. (2022). Estudio comprensivo de la Transformada de Fourier Discreta para el análisis de señales digitales. *Revista Científica y Tecnológica UPSE (RCTU)*. *Revista Científica y Tecnológica UPSE (RCTU)*, 9(1), 75-84.
- Orsini, S., & Aparicio, G. (2021). ESPECTROSCOPIA DE INFRARROJO CON TRANSFORMADA DE FOURIER (FT-IR) PARA ANÁLISIS DE MUESTRAS DE HARINA DE TRIGO, FÉCULA DE MAÍZ Y ALMIDÓN DE YUCA. *Ciencia y Tecnología Alimentaria*, 19(1), 5-16.
- Pardo, J., Castro, J., Iseda, G., Torres, C., & Mattos, L. (2006). RECONOCIMIENTO AUTOMÁTICO DEL HABLA UTILIZANDO LA TRANSFORMADA DE FOURIER Y REDES NEURONALES. *Revista Colombiana de Física*, 38(4).
- Pinares, R., Machaca, V., Lozano, F., Quispe, A., Ccopa, R., & Calsin, B. (2023). Comparaciones de la espectroscopía infrarroja por transformada de Fourier (FTIR), parámetros colorimétricos y porcentaje de medulación en fibra de vicuña. *Revista de Investigaciones Veterinarias del Perú*, 34(4).
- Sámano-Robles, R., Mendoza-Gómez, I., & Neri-Vela, R. (2022). Programación óptima para obtener patrones tridimensionales de antenas de apertura con la Transformada Discreta de Fourier. *Ingeniería Investigación y Tecnología*, 3(3), 93-100.
- Tainta, S., Erro, M., Garde, M., & Muriel, M. (2009). Multiplicador activo de la frecuencia de repetición en trenes de pulsos ópticos ultracortos basado en la transformada óptica de Fourier. *Óptica pura y aplicada*, 42(4), 251-255.
- Tintaya, C. (2005). Software en Matlab para un sistema de adquisición de datos utilizando la tarjeta de sonido de una PC. *Revista de investigación de física*. *Revista de investigación de física*, 8(02), 55-59.
- Tronelli, M. (2014). *Utilización de la Transformada de Fourier en el muestro de señales*. Universidad Nacional del Sur.
- Velásquez Vásquez, W., Díaz Aragon, B., & Gutiérrez Rincón, S. (2018). *Aplicación de espectrometría infrarroja por transformada de fourier (ftir) para la caracterización de la firma espectral de morteros utilizados en la construcción*. Colombia: Universidad Cooperativa de Colombia, Facultad de Ingenierías, Ingeniería Civil, Villavicencio. .



El desarrollo del pensamiento computacional usando gaming en el aula en media superior en la materia de informática II.

Rojas María Dolores^{*1}, Rojas Erick², Balvantín Antonio³.

¹ Escuela de enfermería y Obstetricia del Centro Médico Quirúrgico, Departamento de informática, Blvd Adolfo López Mateos 1000, Col. Centro, 38000 Celaya, Gto

² Tecnológico Nacional de México/ITS de Purísima del Rincón, Departamento de Ingeniería Industrial, Blvd. Del Valle 2301, Col. Guardarrayas, Purísima del Rincón, Guanajuato, 36425, México.

³ Universidad de Guanajuato, Departamento de Ingeniería Mecánica Departamento, Carretera Salamanca - Valle de Santiago km 3.5 + 1.8 Comunidad de Palo Blanco, Salamanca 36787, Guanajuato, México

Resumen.

Esta investigación se sitúa en el ámbito de la didáctica del pensamiento computacional, en concreto en el uso de gaming intrínsecamente ligados a la lógica. A los alumnos de media superior les resulta complicado entender a profundidad y la utilidad de estrategias en contexto de programación, como la aplicación de la lógica computacional para la resolución de problemas en el mundo real, se plantea el uso de juegos para practicarla de un modo lúdico y facilitar la comprensión e integración del conocimiento en los estudiantes. El presente trabajo se planteó como objetivo, el analizar la correlación entre la estrategia de enseñanza y la integración del pensamiento computacional, mediante la aplicación de la estrategia de gaming en el aula, finalmente se discuten las posibles limitaciones a la estrategia aplicada y sus posibles adaptaciones a diferentes niveles educativos. El método de estudio es cuantitativo, cuasi- experimental y de alcance correlacional, utilizando como instrumento el “test de pensamiento computacional” para evaluar a los alumnos. Los resultados obtenidos fueron la correlación en el grupo experimental donde se aplicó el gaming y su afectación positiva en los resultados del test de pensamiento computacional comparados con el grupo de control donde no se aplicó la estrategia de gaming. Se observó que el grupo experimental tuvo un avance significativo en sus resultados en la dimensión de dirección, siendo esta la primera etapa de la programación. El avance significativo después del uso del tangram genera aprendizajes significativos y los alumnos mostraron motivación para participar y utilizar estas estrategias.

Palabras clave. *Pensamiento computacional, Gaming, Programación, Estrategia de enseñanza*

I. INTRODUCCIÓN

Según (Bocconi, 2016), en varias partes del mundo se está incorporando el pensamiento computacional en los

currículos de la enseñanza como un medio para desarrollar las habilidades de resolución de problemas. Mientras que (García-Peñalvo, 2018) explica que las



dos principales motivaciones son: 1) El desarrollo de habilidades de Pensamiento Computacional en niños y jóvenes para que puedan pensar de manera diferente, expresarse a través de una variedad de medios, resolver problemas del mundo real y analizar temas cotidianos desde una perspectiva diferente. Y 2) El desarrollo del Pensamiento Computacional impulsará el crecimiento económico, además de cubrir puestos de trabajo en el área de TIC's y prepararse para futuros empleos.

Los estudiantes deben desarrollar una gran variedad de habilidades que van más allá de la simple codificación de un programa, pues implica aprender a entender un problema (abstraer, modelar, analizar), plantear soluciones efectivas (reflexionar sobre una abstracción, definir estrategias, seguir un proceso, aplicar una metodología, descomponer en problemas más simples), manejar lenguajes para expresar una solución (codificar, entender y respetar una sintaxis), utilizar herramientas que entiendan esos lenguajes (programar, compilar, ejecutar, depurar), probar que la solución sea válida (entender el concepto de corrección y de prueba), justificar las decisiones tomadas (medir, argumentar), entre otras. (Zúñiga, 2014)

La programación es considerada como una competencia que suma el conocimiento, la habilidad y las aptitudes como punto clave para considerar a un alumno competente en el tema del pensamiento computacional, (Bocconi, 2016)

Un estudio realizado en "Dificultades de aprender a programar" señala que los alumnos de segundo semestre de ingeniería en sistemas no realizaban el diagrama de flujo porque no reconocen la importancia de este para la abstracción del problema debido a que desconocen el proceso de traducir sus ideas a código. Por lo que propone establecer un plan de acción para desarrollar los diseños previos a una codificación. (Fuentes-Rosado, 2017)

Esta investigación se centra en aplicar estrategias de enseñanza mediante el uso de gaming en la materia de informática II en alumnos que cursan el segundo semestre de educación media superior en la ciudad de Celaya, Guanajuato que han aprobado el primer semestre de informática con la intención que los alumnos desarrollen habilidades para la resolución de problemas, colaboración y comunicación con una experiencia atractiva de aprendizaje.

Se vuelve una necesidad la enseñanza de la programación en la educación en México debido a que

los planes y programas de la SEP (Secretaría de Educación Pública) que se aplica desde el nivel básico, primaria y secundaria, con la intención de preparar ciudadanos digitales que se integren a una escuela de medio superior de su elección, por lo que los alumnos en sus primeras etapas de aprendizaje de programación deben desarrollar un pensamiento computacional debido a esta estructura educativa los temas que se desarrollan en los primeros semestres de educación media superior se centran en su primer acercamiento a la programación formal. (Secretaría de educación pública, 2023)

El pensamiento computacional es un concepto que fue planteado por primera vez en el año 2006, y a partir de ese momento, ha venido ganando popularidad debido a su prometedor impacto para todas las personas, ya que su premisa principal es que conceptos asociados al ámbito de la computación pueden ser utilizados para la resolución de problemas en otros ámbitos (Angulo, 2022) y según "Muchos consideran que la resolución de problemas es la nueva base del aprendizaje del siglo XXI". Por esa razón, es relevante reflexionar sobre el pensamiento computacional (Trilling, 2009)

Por su denominación en inglés Gamification, es la aplicación de principios y elementos propios de un juego en un ambiente escolar con el objetivo de influir en el comportamiento, incrementar la participación y motivación de los estudiantes. Para (McGonigal, 2011) el mundo de los juegos, aún sin ser lo suficientemente bueno, el jugador puede tener una experiencia muy divertida.

Los beneficios de incorporar un elemento de juego son enriquecedores ya que permiten al alumno a avanzar a un ritmo y una perspectiva personal de la resolución de problemas de manera oportuna y con retroalimentación inmediata.

El juego de tangram, el cual es un rompecabezas formado por piezas poligonales que se obtienen al fraccionar una figura plana y que pueden acoplarse de diferentes maneras para construir distintas figuras geométricas, es de origen chino y su gran popularidad en Europa y en los Estados Unidos surgió a principios del siglo XIX; ésta fue creciendo con el tiempo debido a su carácter lúdico y educativo, de forma que en la actualidad existen numerosos juegos y juguetes basados en él.

Este juego permite crear figuras con las 7 piezas y todas las figuras siempre tendrán la misma área.



El alumno se motiva a racionalizar la organización de piezas de tal manera que pueda representar la imagen que se le solicita mediante el uso de lógica, la cual es indispensable en el pensamiento computacional.

II. MÉTODO

La hipótesis que se tuvo en esta investigación fue: observar si existe una relación en el uso de estrategias lúdicas en el aula que faciliten el desarrollo del pensamiento computacional.

En la Tabla 1 se muestra el tamaño de grupo de control y el tamaño de grupo experimental, en este segundo grupo se aplicó la estrategia de enseñanza lúdica mediante el Tangram.

Tabla 1. Tamaño de grupo

Participantes	Tamaño de grupo
Grupo control	50
Grupo experimental	65
Total	115

Este proyecto se llevó a cabo mediante un diseño cualitativo, longitudinal, cuasi experimental, de alcance correlacional con un método inductivo, aplicando un test a través de un instrumento llamado "test de pensamiento computacional", se aplicó al 100% de los alumnos del grupo de control en el mes de mayo marcando el fin de semestre enero - junio del año 2023 después de finalizar la materia con una enseñanza tradicional. Este mismo instrumento se aplicó al total de alumnos del grupo experimental en el mes de mayo del año 2023 para determinar el nivel del grupo después de aplicar la estrategia de gaming en el aula.

La metodología para aplicar el juego del Tangram en el grupo experimental se llevó a cabo de la siguiente manera.

Dentro de la planeación de la materia de Informática II se ajustaron cuatro días para la aplicación del juego en el horario de la materia, siendo un módulo antes del examen, el tiempo de juego se dio en el módulo completo de la materia (50 minutos) con el fin de crear un espacio adecuado para que los alumnos interactuaran y se creara un ambiente de aprendizaje productivo. El trabajo colaborativo se implementó con la creación de equipos de dos o tres integrantes para la realización de cada actividad. Se utilizó la estrategia de premiación a los equipos que realizaran la actividad correctamente en menor tiempo.

La preparación para los alumnos antes de cada día de juego consistía como se muestra en la Tabla 2. También muestra las diferentes formas de evaluación del juego para crear un ambiente diferente en cada día de juego.

Tabla 2. Aplicación del Tangram

Tema	Actividad de preparación	Actividad para evaluación
La tecnología como área de conocimiento y la técnica como práctica social.	Discusión plenaria de los juegos de siglos pasados y los modernos. El inicio del tangram.	Formar dos figuras con el tangram y escribir una opinión del juego en sus inicios.
Relación de la tecnología con las ciencias naturales y sociales y la resignificación y uso de los conocimientos.	Organización de una mesa redonda para reflexionar sobre la relación de la tecnología con otras ciencias. El tangram en las matemáticas	Formar dos figuras diferentes y obtener el área y perímetro de la figura formada
Desarrollo tecnológico, integración de la tecnología y la ciencia.	Realizar un trabajo de campo que les permita vislumbrar las creaciones tecnológicas con conocimientos previos. Juegos tradicionales para dispositivos electrónicos.	Descargar la aplicación "Tangram Puzzle" en sus dispositivos móviles e investigar otras aplicaciones similares.
Cambio técnico y cambio social	Debatir la influencia de la tecnología para desarrollar la creatividad, la lógica, estrategia y promover el autoaprendizaje. Organización de pasos para la resolución de tareas.	Crear dos formas y crear un diagrama de flujo para la resolución de esas figuras.

Población de estudio: El estudio se llevará a cabo en dos grupos de segundo semestre de nivel medio superior siendo un total de 65 alumnos.

Tipo de muestreo y tamaño de la muestra: El procedimiento de muestreo es intencional y no



probabilístico, dado que los grupos continúan con el programa académico de la Dirección General de Bachilleratos.

Con la siguiente propuesta para el tamaño de muestra.

Tamaño de la población: 65 alumnos

Nivel de confianza: 90%

Margen de error: 5%

Muestra de: 47 alumnos.

Diseño del instrumento

Se propone el Test de Pensamiento Computacional (González, 2015) diseñado bajo los principios y directrices para evaluar el constructo del desarrollo del pensamiento computacional en alumnos de educación básica y media superior y sus dimensiones de; secuencias básicas, bucles, iteraciones, condicionales y funciones. Objetivos: el TPC pretende medir el nivel de desarrollo del pensamiento computacional en el sujeto.

Definición operativa del constructo medido: el pensamiento computacional es la capacidad de formular y solucionar problemas apoyándose en los conceptos fundamentales de la computación, y usando la lógica-sintaxis de los lenguajes informáticos de programación: secuencias básicas, bucles, iteraciones, condicionales, funciones y variables.

Población objetivo: el TPC está diseñado y dirigido a población escolar de entre 14 y 16 años.

Tipo de instrumento: prueba objetiva de elección múltiple con 4 opciones de respuesta (sólo 1 correcta)

Longitud: 28 ítems

Tiempo estimado de realización: 45 minutos

Cada uno de los ítems está diseñado y caracterizado en las siguientes cinco dimensiones:

Concepto computacional abordado: cada ítem aborda uno o más de los siguientes 8 conceptos computacionales, ordenados en una dificultad creciente: Direcciones básicas (4 ítems); Bucles – ‘repetir veces’ (8 ítems); Bucles – ‘repetir hasta’ (4 ítems); Condicional simple – ‘if’ (3 ítems); Condicional compuesto – ‘if/else’ (4 ítems); Mientras que – ‘while’ (4 ítems); Funciones simples (4 ítems); Funciones con parámetros (0 ítems). Estos conceptos están alineados con los estándares de aprendizaje que fija la CSTA para la educación en

ciencias de la computación en estas edades (CSTA, 2011)

Entorno-Interfaz del ítem: los ítems del TPC se presentan en alguno de los siguientes dos entornos gráficos o interfaces: ‘El Laberinto’ (23 ítems); ‘El Lienzo’ (5 ítems)

Estilo de las alternativas de respuesta: en cada ítem, las alternativas de respuesta se pueden presentar en alguno de estos tres estilos: ‘Visual por flechas’ (7 ítems); ‘Visual por bloques’ (17 ítems); ‘Textual’ (4 ítems)

Existencia o inexistencia de anidamiento: dependiendo de si la solución del ítem involucra una secuencia de comandos-órdenes con (29 ítems) o sin (11 ítems) conceptos computacionales anidados (un concepto embebido en otro concepto en un orden de jerarquía superior)

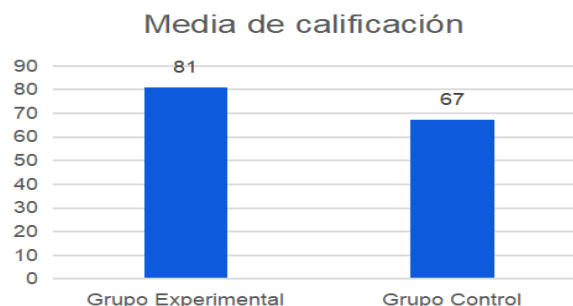
Tarea requerida: dependiendo de cuál de las siguientes tareas cognitivas es necesaria para la resolución del ítem: ‘Secuenciación’, el estudiante debe enunciar de manera ordenada una serie de comandos-órdenes (18 ítems); ‘Completamiento’, el estudiante debe completar un conjunto incompleto de comandos previamente dado (14 ítems); ‘Depuración’, el estudiante debe depurar (‘debug’) un conjunto incorrecto de comandos previamente dado (8 ítems).

III. RESULTADOS

El análisis de resultados se llevó a cabo en el software spss statistics para analizar la hipótesis propuesta en este proyecto.

En un estudio comparativo de la media obtenida de las calificaciones del grupo control y el grupo experimental se tiene que el grupo experimental tuvo un nivel mayor en los aciertos correctos obtenidos como se muestra en la Fig 1.

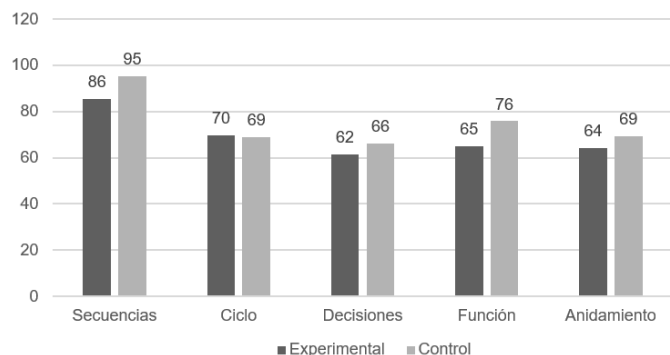
Fig. 1 Comparación de medias





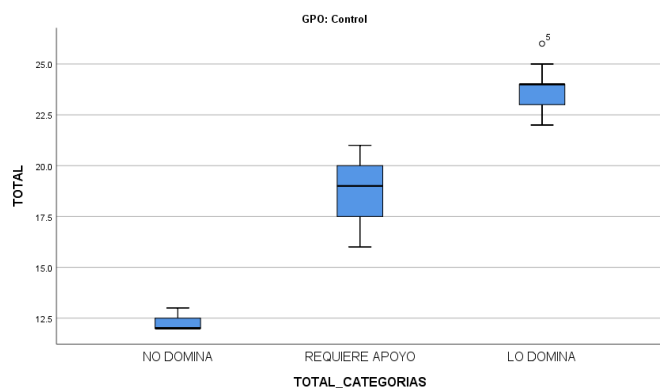
Los resultados obtenidos en cada dimensión del instrumento muestran que, aunque los dos grupos tuvieron un avance durante el curso, el grupo experimental tuvo una mayor comprensión del tema después de aplicar la estrategia de gaming. El avance se muestra en la Fig.2

Fig 2. Promedios comparados en cada dimensión



Los resultados obtenidos en el grupo de control respecto al test de lógica computacional se muestran en la Fig. 3 donde se observan que los alumnos que no dominan las dimensiones del test de pensamiento lógico computacional su media se encuentra en un resultado bajo, los alumnos que requieren apoyo sus resultados muestran que se encuentran en una media positiva y los alumnos que dominan el pensamiento computacional su media se encuentra alta.

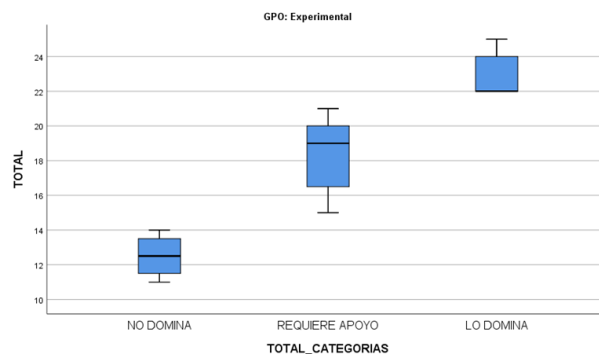
Fig. 3. Resultados correctos en las dimensiones del grupo de control.



En la Fig. 4 se muestran los resultados del grupo experimental donde las respuestas de los alumnos que no dominan las dimensiones del test aplicado se ha elevado su media de respuestas correctas a más de 12 aciertos, mientras que los alumnos que requieren apoyo su media de respuestas correctas se encuentran sobre la media con una tendencia a subir y los alumnos que ya

dominan el pensamiento computacional se han mantenido en sus aciertos en el test.

Fig. 4. Resultados correctos en las dimensiones del grupo experimental.



IV. DISCUSIÓN Y CONCLUSIÓN

En México, las instituciones educativas no han incorporado en un plan académico un aprendizaje de las ciencias de la computación como las habilidades del pensamiento computacional en el nivel medio superior (Secretaría de educación pública, 2011), implicando que al no realizar estas actividades que promuevan el pensamiento computacional, los estudiantes tendrán mayores problemas para razonar, así como se observa en los resultados del presente estudio. Por lo tanto, se puede utilizar este estudio como predictor de resultados que se esperan con las diferentes enseñanzas y la implementación del gaming en el aula y por ende la mejora en su pensamiento computacional con la posibilidad de mejorar la calidad del aprendizaje de los alumnos y su motivación dentro del curso de informática impartido en el nivel medio superior.

Los resultados que se obtuvieron reflejan que existe una correlación en el grupo experimental usando el gaming de tangram en la primera etapa de intervención en el aula. Dentro del test de pensamiento computacional aplicado al grupo de control se observó que los alumnos del grupo de control tienen un nivel más bajo respecto al grupo experimental donde la enseñanza en el aula se utilizó la estrategia didáctica del gaming dando resultados positivos en las respuestas del test.

En este proceso de enseñanza en el aula, usando el gaming para motivar a los alumnos y que a la vez ellos pudieran desarrollar de manera positiva el pensamiento lógico para llevar a cabo la tarea de resolución de



problemas planteado con el juego ha mostrado que se recomienda usar el gaming antes de enseñar los fundamentos de programación a los alumnos que se van adentrando al mundo de la tecnología.

REFERENCIAS

- Angulo, J. A. (17 de Noviembre de 2022). *El pensamiento computacional en la vida cotidiana*. redalyc: <https://www.redalyc.org/journal/5636/563659492016/html/>
- Bocconi, S. C. (2016). Developing computational thinking in compulsory education. *JRC Science for Policy Report*.
<https://doi.org/doi:https://doi.org/10.2791/792158>
- Bocconi, S., Chiocciariello, A., Dettori, G., Ferrari, A., y Engelhardt, K. (2016). Developing computational thinking in compulsory education implications for policy and practice. Sevilla: Joint Research Centre. doi: <http://doi.org/10.2791/792158>
- CSTA. (27 de Junio de 2022). *Computer Science Standards. K-12*
http://csta.acm.org/Curriculum/sub/CurrFiles/CSTA_K-12_CSS.pdf
- Fuentes-Rosado, J. I.-M. (2017). Dificultades de aprender a programar. *Revista Educación En Ingeniería*, 12(24), 76 - 82.
<https://doi.org/doi:https://doi.org/10.26507/rei.v12n24.728>
- García-Peñalvo. (2018). Exploring the computational thinking effects in pre-university. *ScienceDirect*, 80, 407-411.
<https://doi.org/https://doi.org/10.1016/j.chb.2017.12.005>
- González, M. R. (2015). Test de Pensamiento Computacional: principios de diseño, validación de contenido y análisis de ítems. ISBN 978-84-362-6981-9, págs. 291-314
- McGonigal, J. (2011). Reality is broken: Why games make us better and How they can change the world. New York: The Penguin Group.
- Subsecretaría de Educación Superior. (s/f). Gob.mx. Recuperado el 29 de julio de 2023, de <https://www.educacionsuperior.sep.gob.mx/conaces.html>
- Trilling, B., & Fadel, C. (2009). *21st century skills: learning for life in our times*. Jossey-Bass A Wiley Imprint.
- Zúñiga, M. E., Rosas, M. V., Fernández, J., & Guerrero, R. A. (2014). El desarrollo del pensamiento computacional para la resolución de problemas en la enseñanza inicial de la programación. En *Actas del XVI Workshop de Investigadores en Ciencias de la Computación* (pp. 340–343).