

RESUMEN

Este proyecto presenta el desarrollo de un robot de dos ruedas con accionamiento diferencial, capaz de navegación autónoma y ejecución de tareas utilizando un mecanismo de pinza. A través de la integración de odometría, fusión de sensores y algoritmos de navegación como Bug 0 y Filtro de Kalman Extendido, el sistema logra un movimiento preciso y evita obstáculos. El Filtro de Kalman Extendido utiliza SLAM para localizarse mediante el uso de etiquetas Aruco, mejorando la precisión en la localización y el mapeo del entorno. Esta experiencia de aprendizaje integral resalta el potencial de los sistemas autónomos en aplicaciones del mundo real, fortaleciendo habilidades en robótica y preparando para futuros proyectos. Además, la implementación de funciones vitales como mapeo, localización y evitación de colisiones subraya la importancia del proyecto en la robótica móvil.

DESCRIPCION DEL PROBLEMA

El problema de investigación aborda el desarrollo y la funcionalidad de un robot de dos ruedas con navegación autónoma. Se exploran los aspectos matemáticos de la odometría, los algoritmos de navegación y visión, y el uso del Filtro de Kalman Extendido. Además, se analiza cómo el robot puede navegar y realizar tareas específicas, como buscar, recoger y depositar un cubo en una base designada.

METODOLOGIA UTILIZADA

- **EKF SLAM:** Se implementó un algoritmo de EKF SLAM, para localizar el robot dentro de su ambiente. Utilizando etiquetas Aruco, identificadas por sus ID únicos, para mejorar la estimación del estado del robot. El estado incluye la posición y orientación del robot, así como la posición de cada etiqueta. El EKF usa mediciones del encoder y de la cámara para actualizar la predicción y la integración del estado según el Aruco visible.
- **Bug 0:** Se implementó un algoritmo conocido como bug, en este caso el 0, para evadir obstáculos en el camino del robot conforme se está desplazando en el mapa. Para ello utiliza un sensor llamado LiDAR el cual obtiene las distancias de los objetos a su alrededor, cuando existe uno cerca de la tolerancia establecida este decide realizar un giro al lado contrario donde se detecta el objeto y lo evade.
- **Gripper Design:** Se diseñó una pinza en OnShape e impresa en 3D con PLA para recoger y transportar el cubo. La pinza, compuesta por base, enlaces, pinzas y un círculo central, usa un servo conectado a la Hackerboard y controlado por ROS2.
- **State Controller:** Para implementar el comportamiento de mapeo, búsqueda, agarre y entrega, creamos un nodo que puede rastrear todos los estados posibles y cambiar entre ellos según corresponda. Los estados principales son: MAPPING, MOVE TO POINT, GRIP, RETURN TO HOME. En el estado MAPPING, el robot se mueve por el mapa mientras evita los obstáculos frontales. Una vez que se detecta un cubo, se utiliza el estado GRIP y el algoritmo correspondiente.

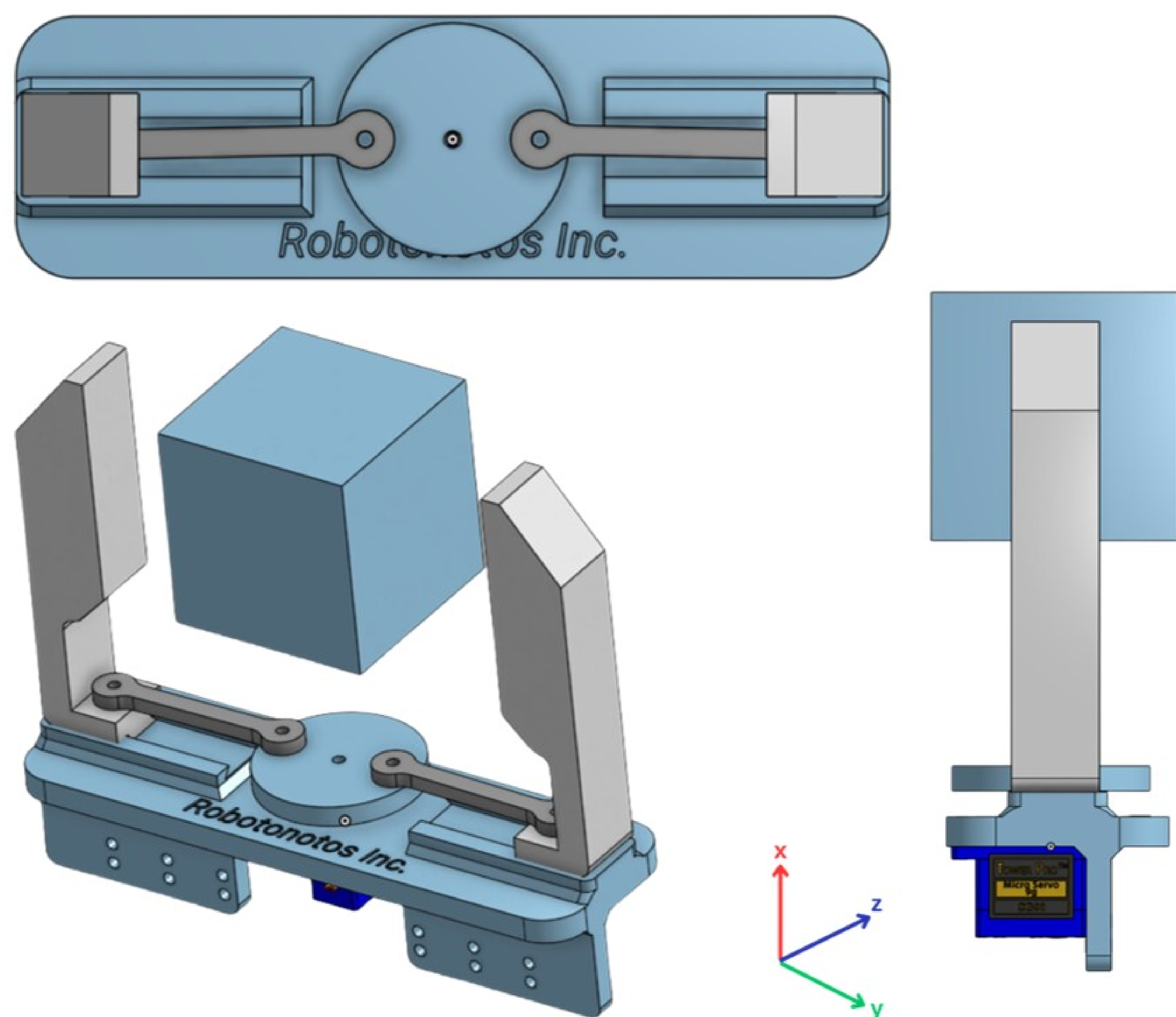


Figura 1: Gripper Mechanism Draw .

RESULTADOS Y APORTACIONES

Para este experimento, se colocó el Puzzlebot en una posición aleatoria dentro de nuestro entorno de prueba. Durante los primeros 40 segundos, no se colocó el cubo, de manera que el robot tuviera tiempo para mapear las etiquetas Aruco en su entorno. Luego, se colocó el cubo en una ubicación aleatoria, dando al Puzzlebot la oportunidad de buscarlo y agarrarlo. Se habían definido una ubicación de entrega y una ubicación de inicio preestablecidas. El robot logró entregar el cubo exitosamente al destino final y se estacionó en las ubicaciones predefinidas.

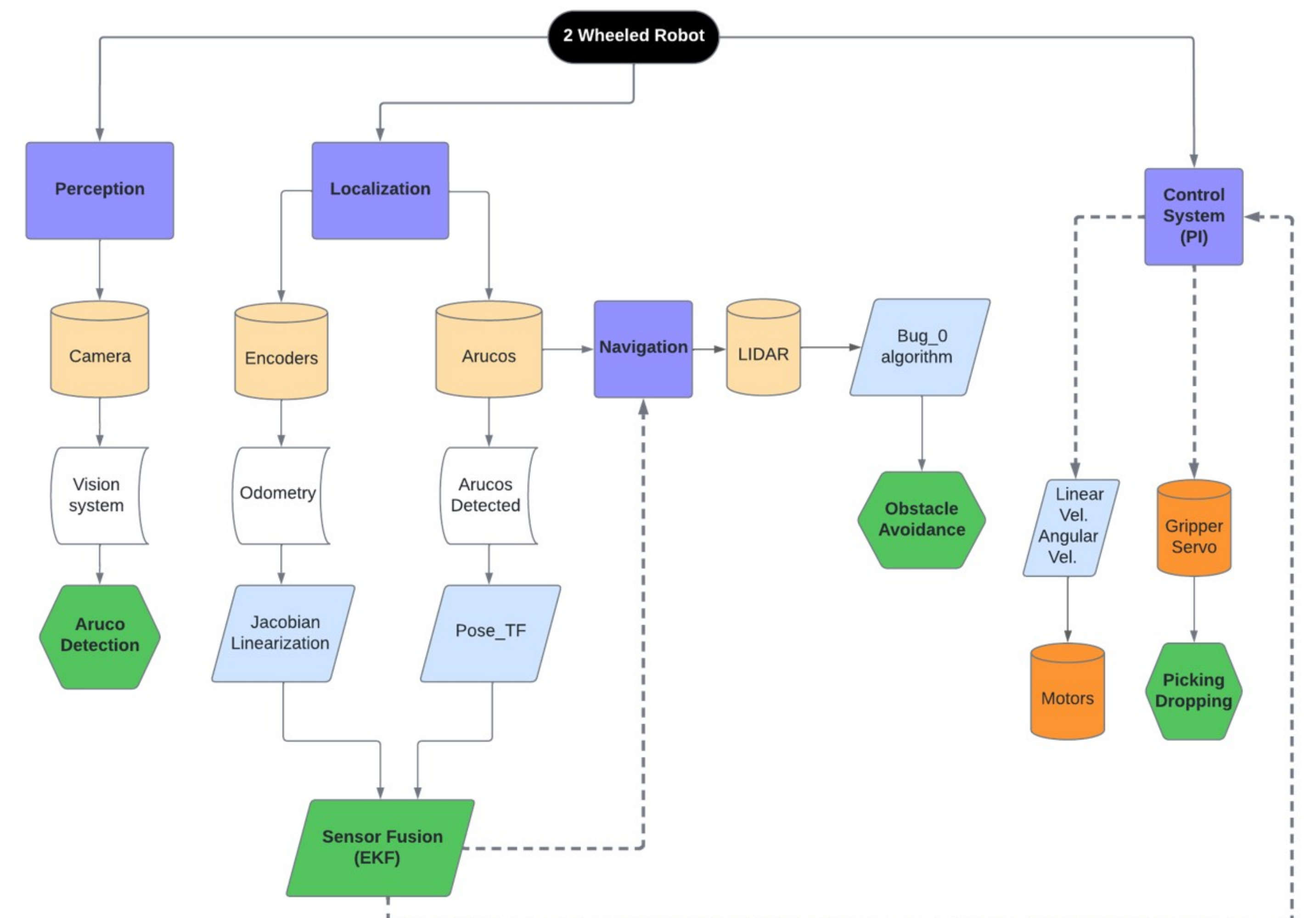


Figura 2: DEMO Diagram .

CONCLUSIONES

El desarrollo de un robot diferencial con sistema de navegación autónoma, requiere de un modelo matemático robusto que filtre la calidad de los datos recibidos provenientes de los sensores que hacen posible el proceso de percepción y localización del vehículo, para que alimente al algoritmo de la navegación propia y, a su vez, envíe datos de calidad al controlador garantizando que los drivers de los motores y el servomotor del gripper sean exactos y precisos. Asimismo, la calidad de los sensores y actuadores deben revisarse detenidamente para asegurar un buen performance del robot y por último las competencias hardskills del equipo de ingenieros es fundamental para el éxito del proyecto.

REFERENCIAS

- [1] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. MIT Press, 2006.
- [2] C. Fairchild and D. T. L. Harman, *ROS Robotics By Example - Second Edition: Learning How to Build and Program Your Own Robots with the Most Popular Open Source Robotics Programming Framework*, 2nd ed. Packt Publishing, 2017.
- [3] G. Bradski, "The opencv library," *Dr. Dobbs's Journal of Software Tools*, 2000.
- [4] P. T. Corporation, "Onshape," <https://www.onshape.com/>, 2012.

Figura 3: ref..

CONTACTO: Abiel Fernandez: contact@abel.dev