TP4 Robótica Móvil

Sacha Varela
Gerónimo González Marino

16 de noviembre de 2025

Introducción

El objetivo de este trabajo práctico es implementar un solucionador de Pose Graph SLAM utilizando la librería GTSAM. En particular, se le pedirá que implemente tanto una solución batch como una solución incremental para los problemas SLAM 2D y 3D utilizando conjuntos de datos estándar.

Los códigos, imagenes y el informe se encuentran en el repositorio dentro de la carpeta TP4/Scripts [1]

Graph-SLAM 2D

A. Leer archivo g2o

Escriba una función para leer el conjunto de datos Intel 2D3 desde el formato G2O y generar poses y aristas. Estas posiciones y aristas se utilizan en problemas posteriores. Puede ser cualquier forma que desee, siempre que pueda usarla para generar el resultado correcto.

De acuerdo a la documentación de G2O se leen los archivos extrayendo poses y vértices de la siguiente manera

```
Vertex
TAG ID CURRENT_ESTIMATE
Edges
TAG ID_SET MEASUREMENT INFORMATION_MATRIX
```

y para los casos 2D y 3D respectivamente

```
2D
VERTEX_SE2 i x y theta
EDGE_SE2 i j x y theta info(x, y, theta)

3D
VERTEX_SE3:QUAT i x y z qx qy qz qw
EDGE_SE3:QUAT i j dx dy dz qx qy qz qw info(I11 I12 I13 I14 I15 I16 I22
↪ I23 I24 I25 I26 I33 I34 I35 I36 I44 I45 I46 I55 I56 I66)
```

Donde las matrices de información solo almacenan la triangular superior, dado que siempre es simétrica. En todos los casos la inversa de esta matriz representa la matriz de covarianza.

El código para cada caso construye las poses y las aristas de la siguiente manera

```
2D
Pose: {id, x, y, theta}
Edge: {from, to, dx, dy, dtheta, Omega(3x3)}
3D
Cada pose = {id, x, y, z, qx, qy, qz, qw}
Cada edge = {from, to, x, y, z, qx, qy, qz, qw, info (6x6)}
```

B. Batch Solution

Una solución por lotes significa que primero construimos el grafo completo y luego lo resolvemos por completo. Cargar data/input_INTEL_g2o.g2o y construir un grafo de factores no lineales 2D usando GTSAM. Utilice el solver de Gauss-Newton. Visualice y compare la trayectoria optimizada con la trayectoria inicial. Incluye la gráfico en el informe. Describir el proceso de construcción del grafo y sus parámetros.

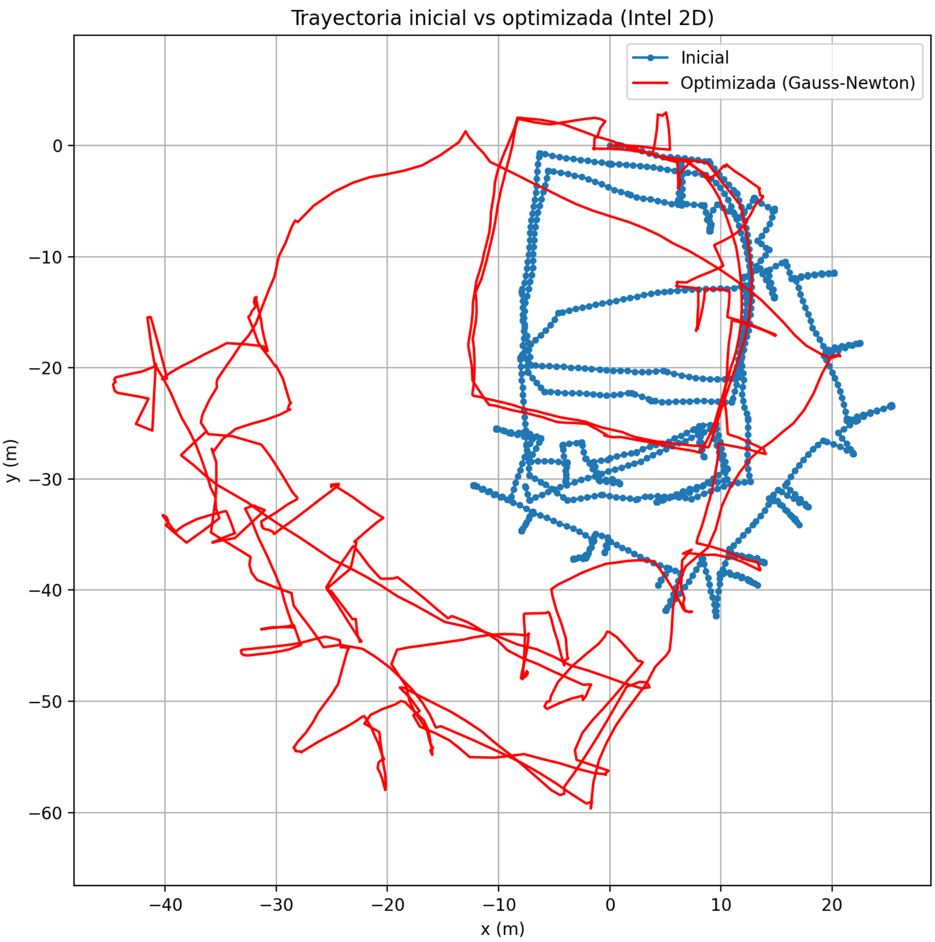

Figura 1: Trayectoria Intel optimizada Gauss-Newton

Para la construcción del grafo y se genera un noisemodel de cada arista utilizando su matriz de covariancia, la que obtenemos luego de invertir la matriz de información asociada. Ademas se añade la primera pose como un prior, definiendo su matriz de covariancia muy cercana a cero, lo que equivale a una alta seguridad de su posición, de manera de evitar el problema de `Gauge Freedom`.

Podemos observar que la solución optimizada no corresponde a un grafo representativo del mapa de Itel. Esto es normal dado que el solver de Gauss-Newton no garantiza converger al mínimo absoluto si no que dependiendo la semilla inicial puede converger a mínimos locales. Algunas estrategias para mitigar este problema son agregar perturbaciones a los datos de la semilla inicial de manera de "tirar.ªl solver lejos del mínimo local, o utilizar solver más robustos como el de Levenberg-Marquardt. Debajo se muestra una figura con el grafo del dataset, el solver original, el solver G-N con una perturbación de 0.05 y el solver L-M

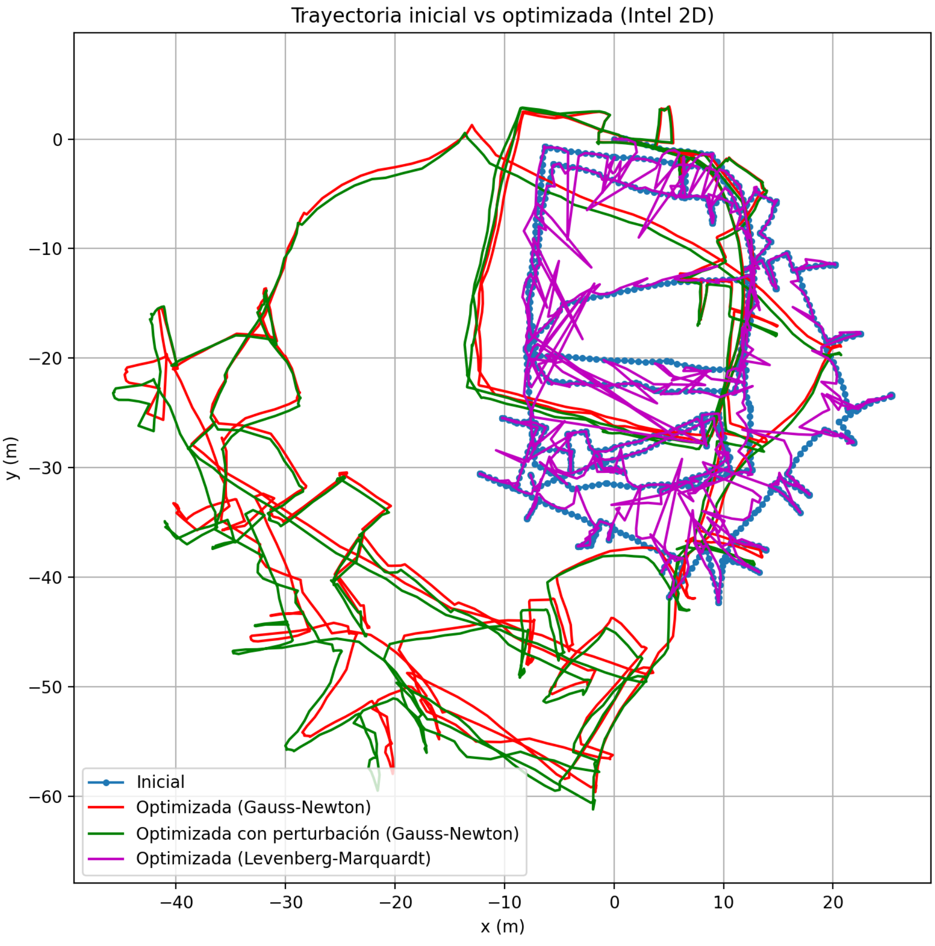

Figura 2: Trayectoria Intel optimizada G-N, perturbación, L-M

Aun con una perturbación (simple) el solver de Gauss-Newton converge a mínimos locales, por otro lado el solver ed L-M parecería acercarse más a la solución pero aún se pueden observar algunas trayectorias erráticas (En este caso L-M utilizo la semilla inicial original sin perturbaciones).

EL proceso de la solución Batch consiste en optimizar el grafo completo, una vez dada la semilla de cada pose se desea minimizar el error global cuadrático de la siguiente manera

$$
\begin{aligned}
\mathbf{x}^* &= \underset{\mathbf{x}}{\operatorname{argmin}}\, F(\mathbf{x}) \longleftarrow \boxed{\text{global error (scalar)}} \\
&= \underset{\mathbf{x}}{\operatorname{argmin}} \sum_i e_i(\mathbf{x}) \longleftarrow \boxed{\text{squared error terms (scalar)}} \\
&= \underset{\mathbf{x}}{\operatorname{argmin}} \sum_i \mathbf{e}_i^T(\mathbf{x}) \boldsymbol{\Omega}_i \underbrace{\mathbf{e}_i(\mathbf{x})}_{\boxed{\text{error terms (vector)}}}
\end{aligned}
$$

Figura 3: Teoría error global

Las funciones de error son suaves en la vencidad del mínimo (con surte global)por lo que se pueden realizar linearizaciones locales iterativas.

Para esto se calculan las matrices c_i, que no dependen del termino Δx, las matrices b_i^T, que dependen del Jacobiano de cada termino de error local (linearización), y las matrices H_i Hesseianas, asociadas a la derivada segunda local en el error. Luego con todas estas matrices se calcula el error global cuadrático como

$$F(\mathbf{x} + \Delta\mathbf{x}) = c + 2\mathbf{b}^\top \Delta\mathbf{x} + \Delta\mathbf{x}^\top \mathbf{H} \Delta\mathbf{x}$$

Figura 4: Error cuadrático global

y se minimiza a través del solver de Gauss-Newton. Esto se condice con un enfoque probabilístico porque la minimización del error global es equivalente maximizar el log likelihood de Gaussianas independientes de los terminos del error.

C. Incremental Solution

Utilizar el solver ISAM2 para optimizar la trayectoria de forma incremental (a medida que construye el grafo gradualmente). Visualice y compare la trayectoria optimizada con la trayectoria inicial. Incluir el gráfico en el pdf. Describir el proceso de construcción del grafo y sus parámetros.

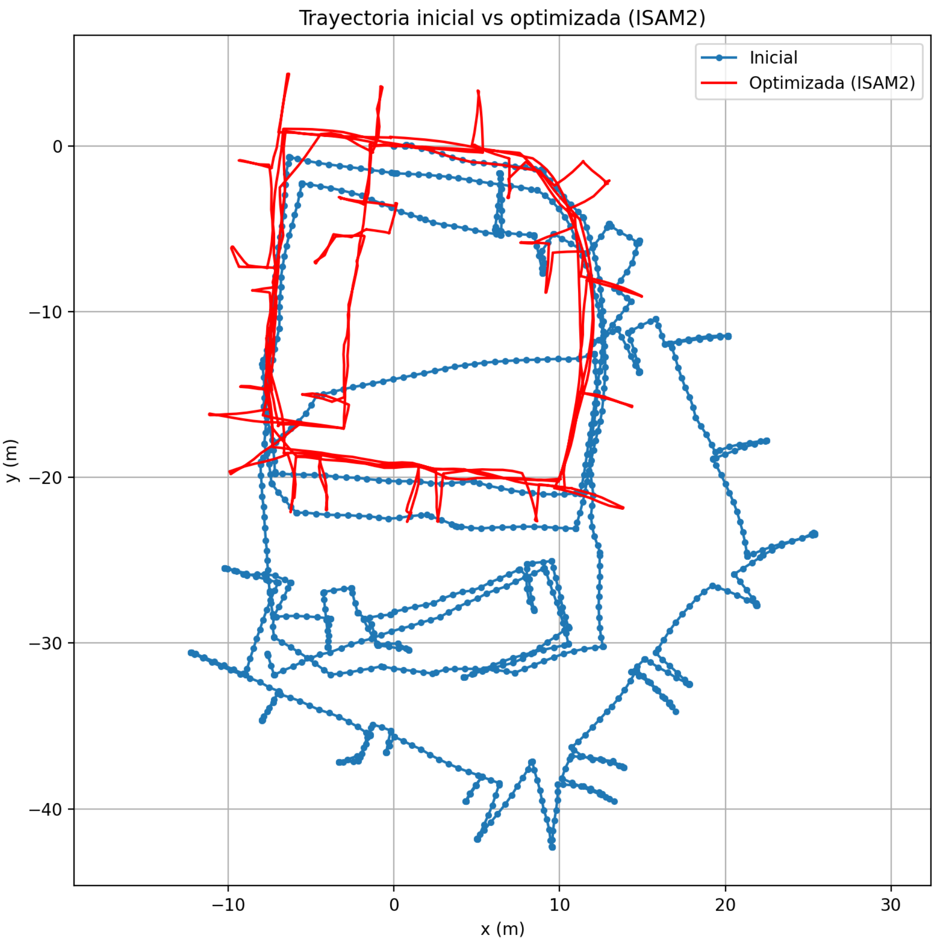

Figura 5: Trayectoria Intel optimizada incrementalmente con ISAM2

En el caso del Pose-Graph incremental queremos computar la trayectoria optima cada vez que recibamos un nueva medición o por lo menos a intervalos regulares. De esta manera construimos un grafo con cada nuevo conjunto de $vertice_i$ y sus $aristas_{ij_{1,\ldots,n}}$ asociadas.

En la solución por batch se re-lineariza toda la matriz H con cada nueva actualización de la iteración, lo que busca la solución incremental es evitar reconstruir toda la matriz global cada vez que llega una nueva medición y actualizar solo las partes relevantes. Esto es posible en problemas lineales, pero en la mayoría de las ocasiones, los modelos de movimiento son no-lineales, y la re-linearización es muy costosa, para esto los algoritmos ISAM se aprovechan de las ventajas de los grafos tipo tree-structured y trabajan en dos pasos: eliminan variables del grafo de factores para obtener un `Bayes tree` y luego explotan la propiedad especial de estas estructuras, que pasan a ser estructuras jerárquicas y permiten una actualización local (información extraida de: Luca Carlone et al. SLAM Handbook. Cambridge University Press, 2025.)

Graph-SLAM 2D

A. Leer archivo g2o

Escriba una función para leer el archivo 3D Garage G2O5 desde el formato G2O y genere poses y aristas.

Como se comento en la sección 2.A el archivo 3D en formato g2o se lee de manera similar al archivo 2D, con la diferencia de que los vértices ahora involucran una matriz de rotación en forma de quaternion, y que se debe tener en cuenta el orden en el que los mismos se pasan a la función Rot3(). La matriz de información es de 6x6 y se construye a partir de los valores de la matriz triangular superior provista por el archivo. Luego se calcula la inversa para obtener la matriz de covariancia.

B. Batch Solution

Cargue data/parking-garage.g2o y construya un factor-graph 3D no lineal usando GTSAM. Utilice el solver de Gauss-Newton. Visualice y compare la trayectoria optimizada con la trayectoria inicial. Incluya un gráfico 3D o dos gráficos 2D en su pdf. Describir el proceso de construcción del grafo y sus parámetros.

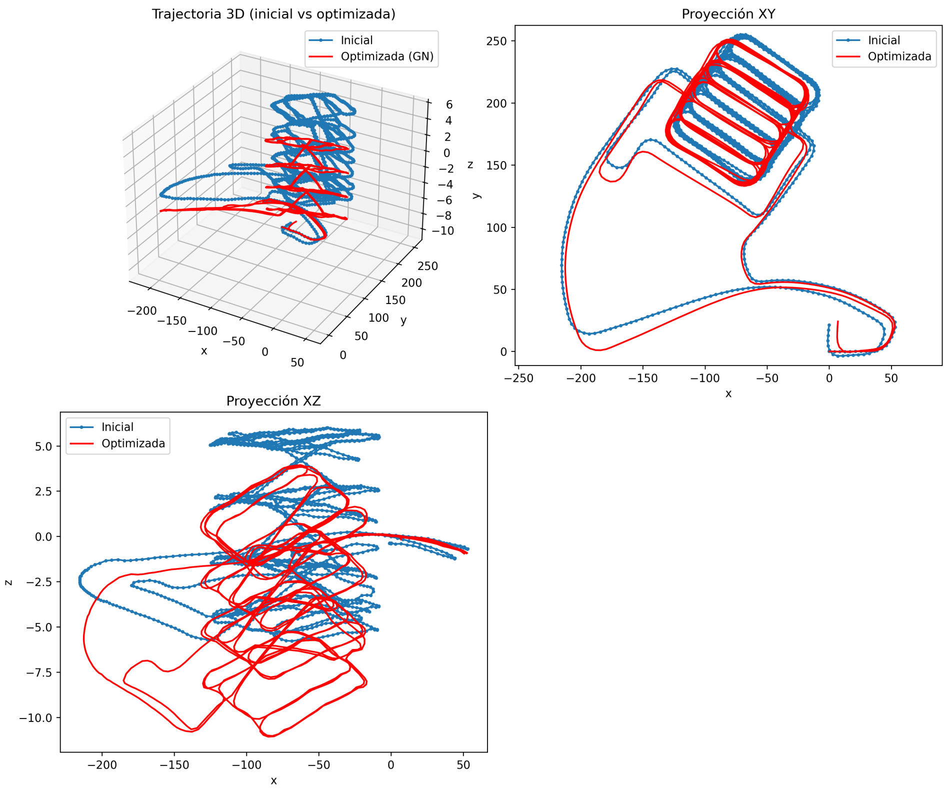

Figura 6: Optimización de trayectoria 3D utilizando Gauss-Newton

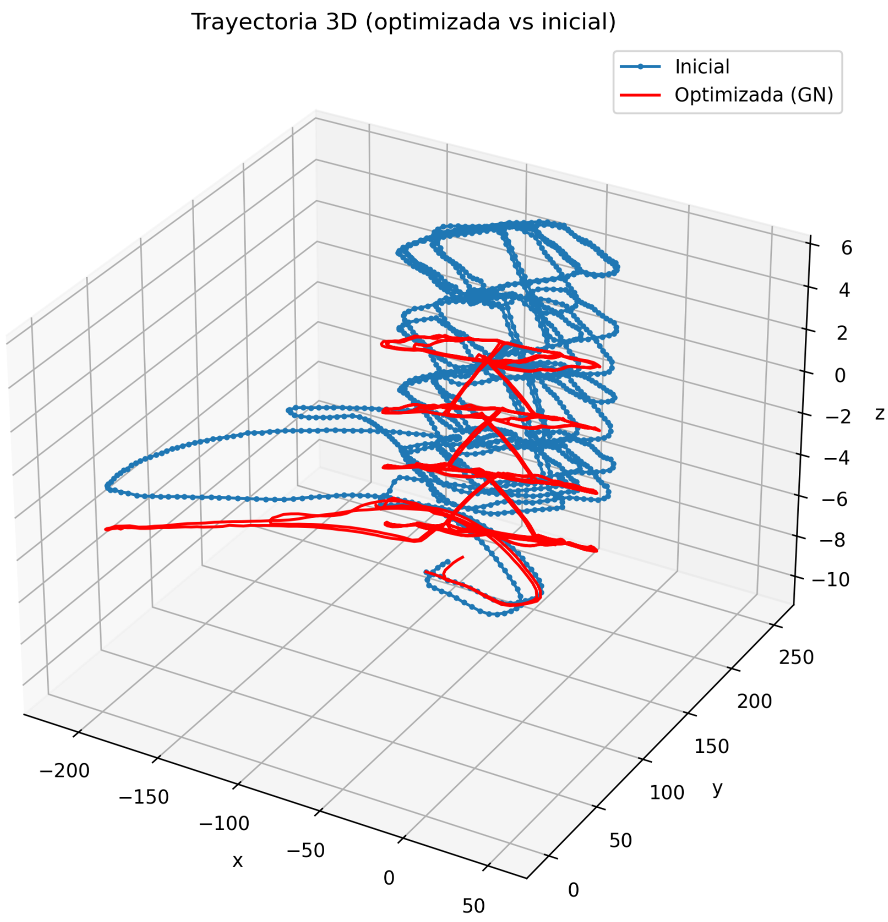

Figura 7: Detalle optimización de trayectoria 3D utilizando Gauss-Newton

Para el caso 3D, los pasos de resolución de la minimización del error son los mismos que para el caso 2D pero las matrices involucradas son mucho más grandes ya que el robot tiene 6 grados de libertad a definir en cada pose y así también las aristas asociadas a cada nodo. El costo computacional crece exponencialmente y se debe prestar especial atención al calculo de las inversas de las matrices de información, definiendo métodos para el calculo de la pseudoinversa y estrategias para obtener matrices definidas positivas.

C. Incremental Solution

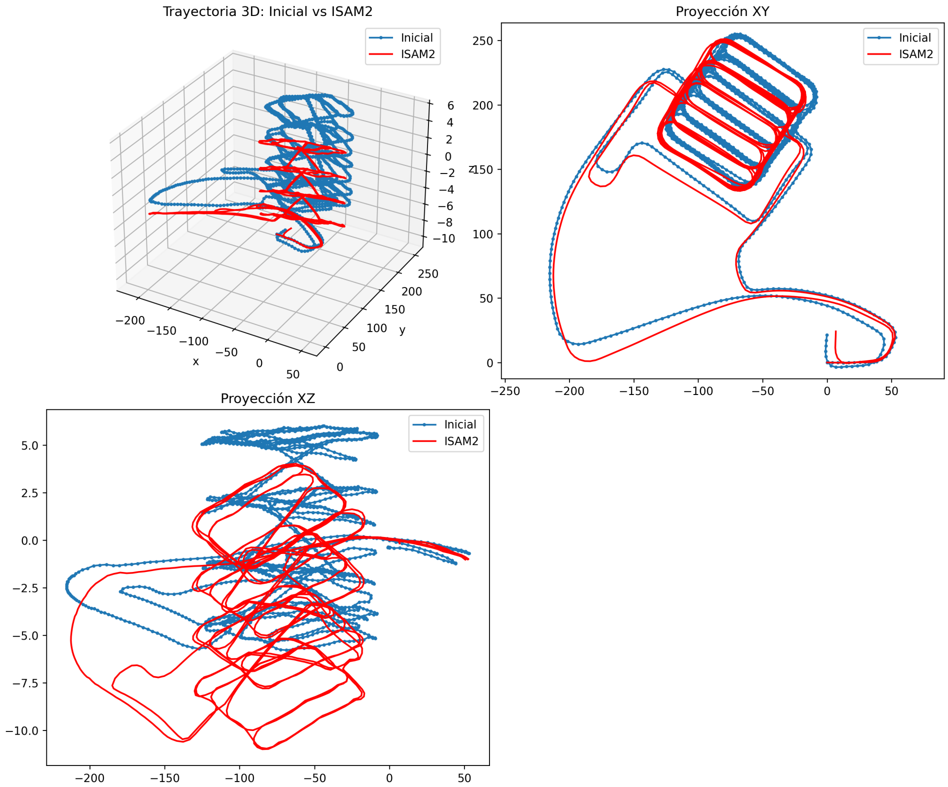

Figura 8: Optimización de trayectoria 3D utilizando iSAM

El caso incremental 3D se aborda de igual manera que el 2D. Cada nueva optimización se realiza sobre el grafo anterior previamente optimizado y solo se añade la actualización y factorización de cada nueva pose a la matriz global. Para la primera pose se utiliza la primer pose del archivo g2o como el prior de manera de, otra vez, evitar el problema de `Gauge Freedom`.

Referencias

[1] https://github.com/geronimogonzalez/Robotica-movil/tree/main