INTRODUCTION TO ROBOT TECHNOLOGIES

Odometry: The location of the robot according to $x=0$, $y=0$. $[x, y, \theta] \longmapsto$ odometry

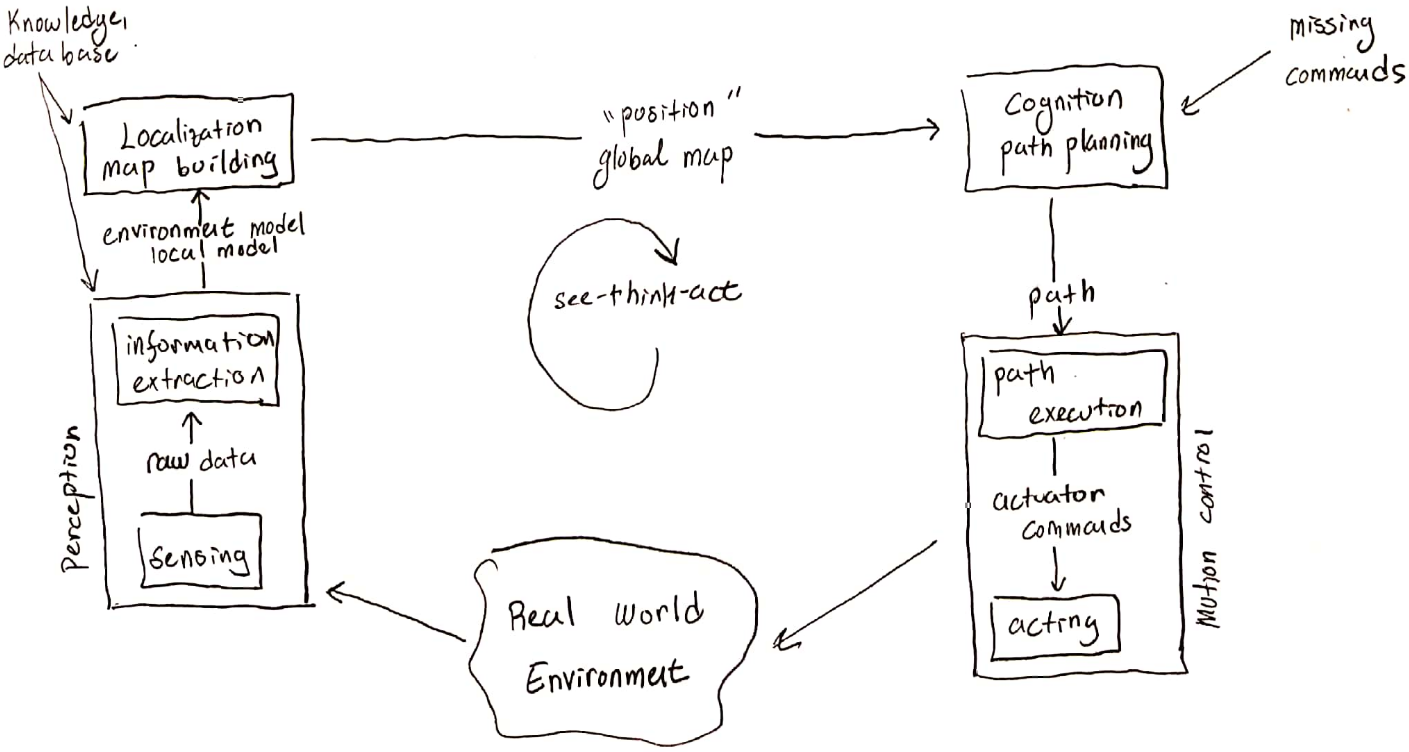
Odometry is the use of data from motion sensors to estimate change in position over time.

Trajectory: A sequential information that is collected from continuous odometry informations.

Localization: Finding the location of the robot in an environment. If the robot rotates around itself, localization sensitivity decreases.

Navigation: The robot can go to a target point that we give by itself is called navigation. (autonomous)

Exploration: Navigation but the robot determines the target point instead of a human. (frontiers)



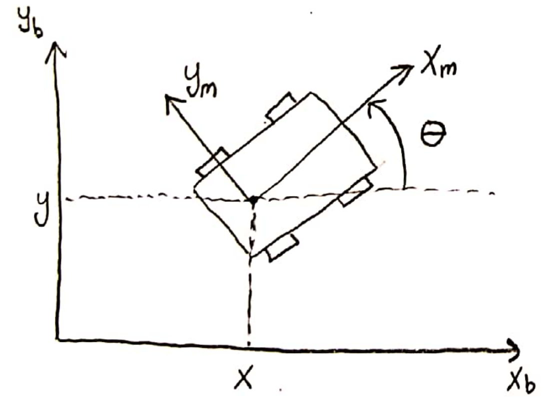
Wheeled Mobile Robots

- Combination of various physical (hardware) and computational (software) componants.

- A collection of subsystems:

    - Locomotion: how the robot moves through its environment.
    - Sensing: How the robot measures properties of itself and environments.
    - Control: How the robot generate physical actions.
    - Reasoning: How the robot maps measurements into actions.
    - Communication: How the robots communicate with each other or with an outside operator.

Notation



Pose/Posture: position (x, y) and orientation θ

- $\{x_m, y_m\} \longmapsto$ moving frame
- $\{x_b, y_b\} \longmapsto$ base frame

$$q = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix} \longmapsto \text{robot posture in base frame (odometry)}$$

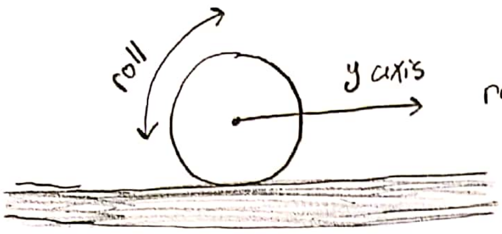
pose

pose to pose: velocity

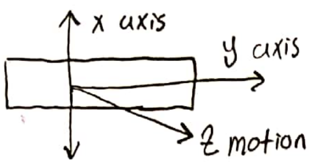
$$R(\theta) = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \longmapsto$$

rotation matrix expressing the orientation of the base frame with respect to the moving frame.

Wheels



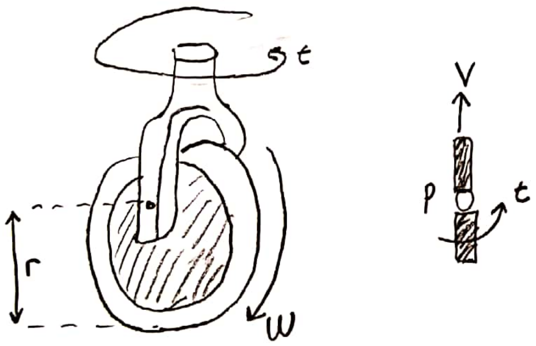
rolling motion

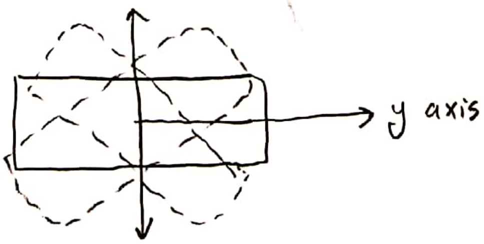


Lateral slip

Steered Wheel

The orietation of the rotation axis can be controlled.





Idealized Rolling Wheel

Assumptions:

1. The robot is built from rigid mechanisms.
2. No slip occurs in the orthogonal direction of rolling (non-slipping)
3. No translational slip occurs between the wheel and the floor (pure polling)
4. The robot contains at most one steering link per wheel.
5. All steering axes are perpendicular to floor.

Robot Wheel Parameters

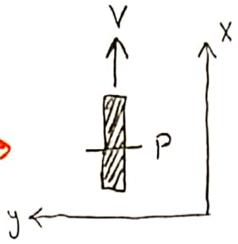
For low velocities, rolling is a reasonable wheel model. This is the model that will be considered in the kinematics models of WMR.

Wheel Parameters:

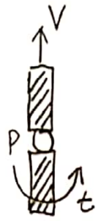
- r = wheel radius
- v = wheel linear velocity
- w = Wheel angular velocity
- t = Steering velocity.

Wheel Types

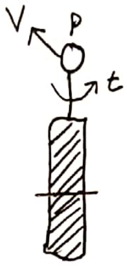
Fixed Wheel ⟼



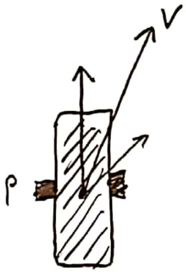
Center Orientable Wheel ⟼



Off-Centered Orientable Wheel (castor wheel) ⟼



Swedish Wheel : Omnidirectional Property ⟼



Fixed Wheel

- Velocity of point P

$$V = (r \times w) a_x$$, where a_x : A unit vector to x-axis

- Restriction to the robot mobility

point P cannot move to the direction perpendicular to plane of the wheel.

~~Fixed Wheel~~ Centered Orientable Wheel

- Velocity of point P

$$V = (r \times w) a_x$$, where a_x : A unit vector to x-axis

a_y : A unit vector to y-axis

Off-Centered Orientable Wheels

$$V = (r \times \omega) a_x + (d \times t) a_y$$

Sweedish wheel

$$V = (r \times \omega) a_x + U a_s$$

where a_s : A unit vector to the motion of roller.

Examples of WMR

Bi-wheel Robot

- Smooth motion
- Risk of slipping
- Sometimes use roller-bar to make balance

Caterpillar Type Robot

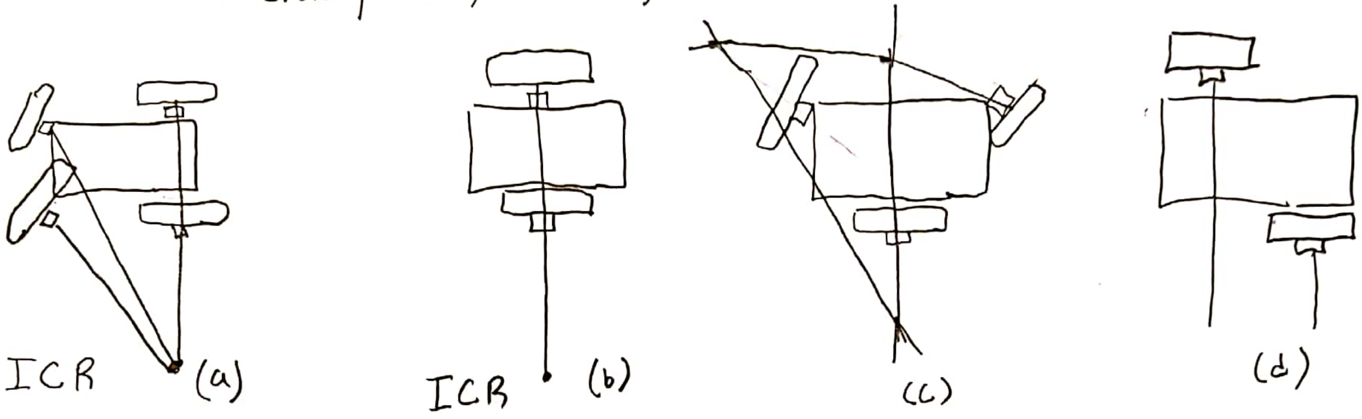
- Exact straight motion
- Robust to slipping.
- Inexact modeling of turning.

Omnidirectional Robot

- Free motion.
- Complex structure.
- Weakness of the frame.

Mobile Robot Locomotion

- instantaneous center of rotation (ICR) or instantaneous center of curvature (ICC)
  - a cross point of all axes of the wheels

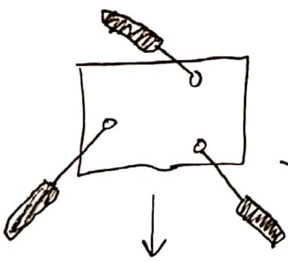


(a) → degree mobility 1 (fixed arc motion) → degree of steerability : 1

(b) → degree mobility 2 (variable arc motion)

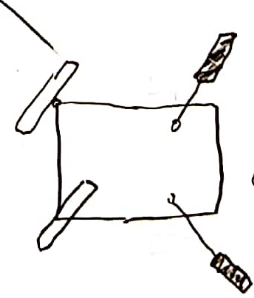
(c) → degree mobility 0 (cannot move anywhere)

degree of steerability : 1 ↦ two mutually depedent centered orientable wheels

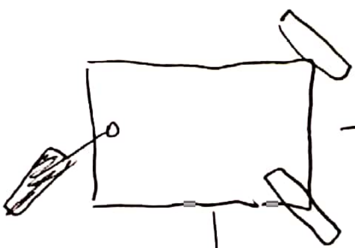


→ degree mobility 3 (fully free motion)

→ degree of steerability : 0 (no center oriented wheels)



one centered orientable wheel → degree of steerability : 1



→ Two mutually independent centered orientable wheels.

→ degree of steerability : 2

Scanned with CamScanner

Non-holonomic Constraint

A non-holonomic constraint is a constraint on the feasible velocities of a body.

Your robot can move in some directions (forward and backward), but not others (sideward).

Differential Drive:

- two driving wheels (plus roller-bar for balance)
- simplest drive mechanism.
- sesitive to relative velocity of the two wheels (small error result in different trajectories, not just speed)

Steered Wheels:

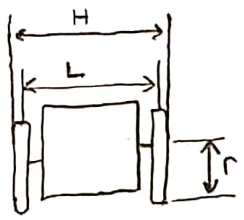
- Steering wheel + rear wheels
- cannot turn $\pm 90°$
- limited radius of curvature

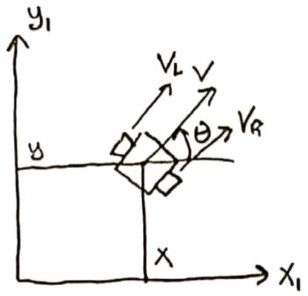
Synchronous Drive

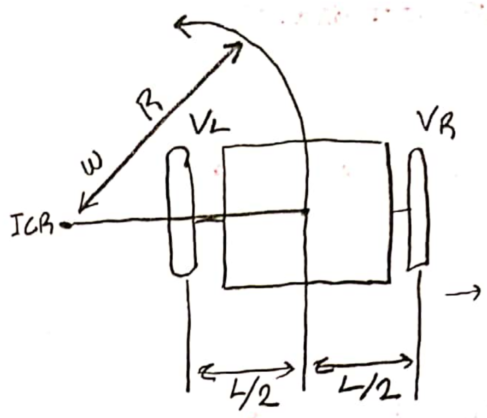
Omni-directional

Car Drive

Differential Drive







→ W is same for each wheel

- Posture of the robot

$$P = \begin{pmatrix} x \\ y \\ \theta \end{pmatrix}$$

$(x, y) \mapsto$ position of the robot

$\theta \mapsto$ orientation of the robot

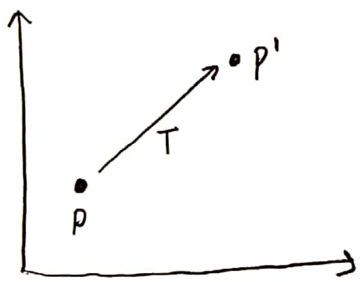
- Control input

$$U = \begin{pmatrix} V \\ \omega \end{pmatrix}$$

$V \mapsto$ Linear velocity of the robot

$\omega \mapsto$ Angular velocity of the robot (notice: not for each wheel)

2D Translation

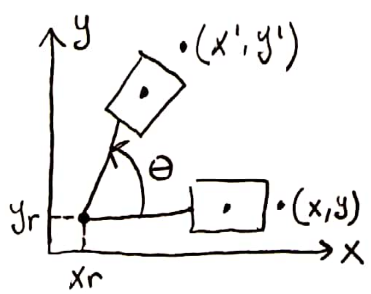


$$x' = x + t_x, \quad y' = y + t_y$$

$$P = \begin{bmatrix} x \\ y \end{bmatrix}, \quad P' = \begin{bmatrix} x' \\ y' \end{bmatrix}, \quad T = \begin{bmatrix} t_x \\ t_y \end{bmatrix}$$

$$P' = P + T$$

2D Rotation



$$x' = x_r + (x - x_r)\cos\theta - (y - y_r)\cdot\sin\theta$$

$$y' = y_r + (x - x_r)\sin\theta + (y - y_r)\cdot\cos\theta$$

$$P' = P_r + R\cdot(P - P_r)$$

$$R = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$$

Homogeneus Coordinates

2D Translation $\longmapsto$ $\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$, $P' = T(t_x, t_y)\, P$

2D Rotation $\longmapsto$ $\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$, $P' = R(\theta)\, P$

Sensor Types

- Time of flight (ToF)
  - Sonar
  - lazer
  - radar
- Camera
  - RGB
  - Infrared → salt and pepper
  - RGB-D → point cloud
  - Stereo camera
  - Thermal
- Inertial Measurement Unit (IMU)
  - Accelerometer
  - Magnetometer
  - Gyroscope

  (orientation, where is north, robots trausformation, robots acceleration)
- Ortam algı sensörleri (CO_2, ışık, nem etc)

- LIDAR sensor: 270°'lik okuma.
  0.25'lik hassasiyet. (derecenin dörtte biri)
  2D-3D

Localization

Determination of the pose (= position + orientation) of a mobile robot in a known environment in order to successfully perform a given task.

Localization As An Estimation Problem

The robot must infer its pose from available data

Data (noisy):

- Motion Information:
  - → proprioceptive sensors (e.g., encoders, accelerometers, etc.)
- Environment Measurement
  - → Exteroceptive Sensors (e.g. laser, sonar etc)

$x_{r,0:t} = \{x_{r,0}, x_{r,1}, \dots, x_{r,t}\} \mapsto$ pose

$z_{1:t} = \{z_1, z_2, \dots, z_t\} \mapsto$ exteroceptive measurement

$u_{0:t} = \{u_0, \dots, u_t\} \mapsto$ motion commands (proprioceptive measurement)

Belief of the robot at time t: PDF describing the information the robot has regarding its pose at time t, based on all available data.

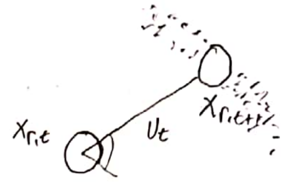
$$bel_t(x_r) = p(x_{r,t} = x_r \mid z_{1:t}, u_{0:t-1})$$

Prior belief of the robot at time t: pdf before acquiring the last measurement z_t:

$$\overline{bel_t}(x_r) = p(x_{r,t} = x_r \mid z_{1:t-1}, u_{0:t-1})$$

The **robot motion model** is the pdf of the robot at time $t+1$ given the robot pose and motion action at time t. It takes into account the noise characterizing the proprioceptive sensors:

$$p(X_{r,t+1} \mid X_{r,t}, U_t)$$



The measurement model describes the probability of observing at time t a given measurement Z_t when the robot pose is $X_{r,t}$. It takes into account the noise characterizing the exteroceptive sensors.

$$p(z_t \mid X_{r,t})$$

$$\overline{bel_t}(X_r) = \int_{\Omega} p(X_r \mid X_{r,t-1} = y, U_{t-1}) \cdot bel_{t-1}(y)\, dy$$

$$bel_t(X_r) = \eta\, p(z_t \mid X_{r,t} = X_r)\, \overline{bel_t}(X_r)$$

Typical Motion Models

- Odometry based ↦ wheel encoders
- Velocity based (dead reckoning) ↦ no wheel encoders given
  - ↳ calculates the new pose based on the velocities and the time elapsed.
  - ↳ lower performance

- Odometry is the use of data from motion sensors to estimate change in position over time ↦ calculate the resulting robot position and orientation from wheel encoder measurements.

Visual Odometry

The vehicle ego-motion is estimated from the apparent motion of the features in the image space.

possible sources of noise: different wheel diameters, bump, carpet.

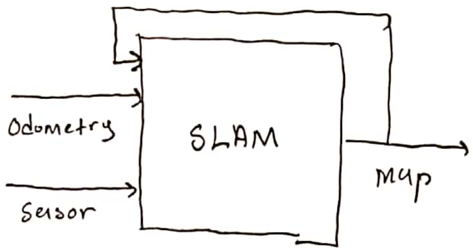
Deterministic (systematic) errors can be eliminated through proper calibration.

Non-deterministic errors have to be described by error models, and will always lead to uncertain position estimate.

Errors are integrated, unbounded!!

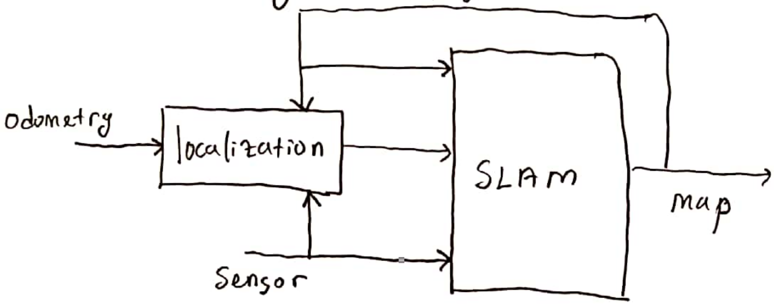
SLAM

- Simultaneous localization and mapping
- if odometry is perfect



gmapping → full slam

- Since odometry is not perfect



Online slam: $p(x_t, m \mid z_{1:t}, u_{1:t})$

full slam: $p(x_{1:t}, m \mid z_{1:t}, u_{1:t})$

$$p(x_t, m \mid z_{1:t}, u_{1:t}) = \int \cdots \int p(x_{1:t}, m \mid z_{1:t}, u_{1:t})\, dx_1 \cdots dx_{t-1}$$

input: odometry, sensor data, previous map

output: updated map

Quaternion - Orientation Format

$$[w, x, y, z] = \left[\cos\frac{\theta}{2}, \sin\frac{\theta}{2}\cdot n_x, \sin\frac{\theta}{2}\cdot n_y, \sin\frac{\theta}{2}\cdot n_z\right]$$

For mobile robot travelling on xy-plane

$$[w, x, y, z] = \left[\cos\frac{\theta}{2}, 0, 0, \sin\frac{\theta}{2}\right]$$

With only dead reckoning, vehicle pose uncertainty grows without bound.

- SLAM with laser scanning
- Observations
- Local Mapping
  - ↳ iterated closest point → Scan matching $O(n^2)$
    - ↳ False minima:
      - if icp starts far from true alignment
      - if scans exhibit repeated local structure
    - ↳ Bias:
      - Anisotropic point sampling
      - Differing sensor fields of view
- Loop closing
  - ↳ scan matching
  - ↳ deferred validation
  - ↳ Search strategies

↓ full-SLAM

↓ icp solves small-scale short-duration SLAM fairly well

But now consider:

- Large scale
- High uncertainty

Map files as images PNG, JPG

Although, RGB images are converted to grayscale images to being interpreted by ROS.

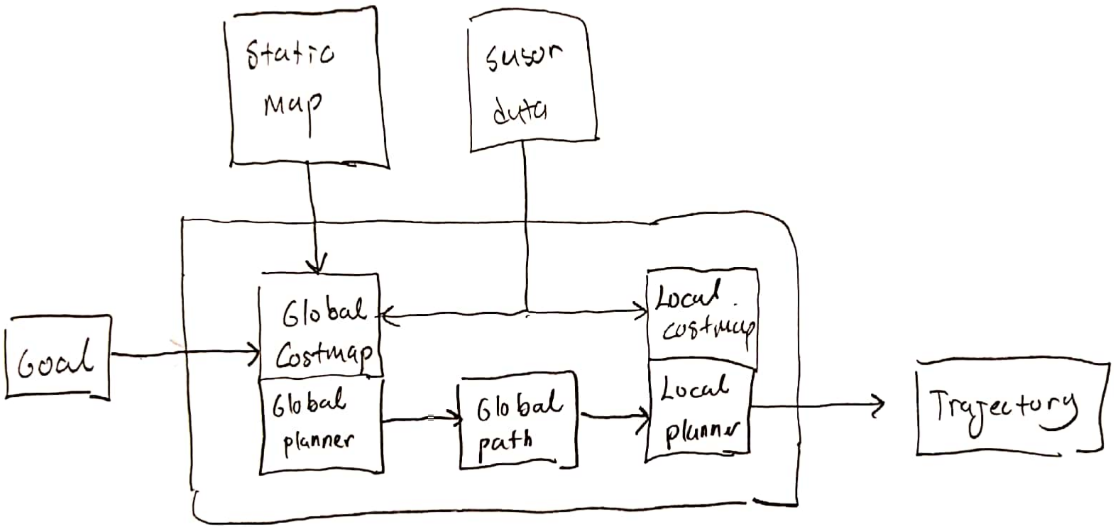
YAML holds additional informations about the map.

ROBOT NAVIGATION

Give a map, starting location and a goal point.

↳ Navigation

↳ Takes information from odometry and sensors, and a goal pose and outputs safe velocity commands that are sent to robot.



- The navigation stack can only handle differential drive and holonomic wheeled botos.
    - ↳ it can also do certain things with biped robots such as localization, as long as the robot does not move sideways
    - ↳ A planar laser must be mounted on the mobile base of the robot to create map and localization
    - ↳ perform best on robots that are nearly square or circular,

Navigation Planners

Global planner ⟼ paths for a goal

Local planner ⟼ paths in the nearby distances

↳ appropriate velocity command

Yakin frontier → A^* daha uygun