

INTRODUCTION TO ROBOT TECHNOLOGIES

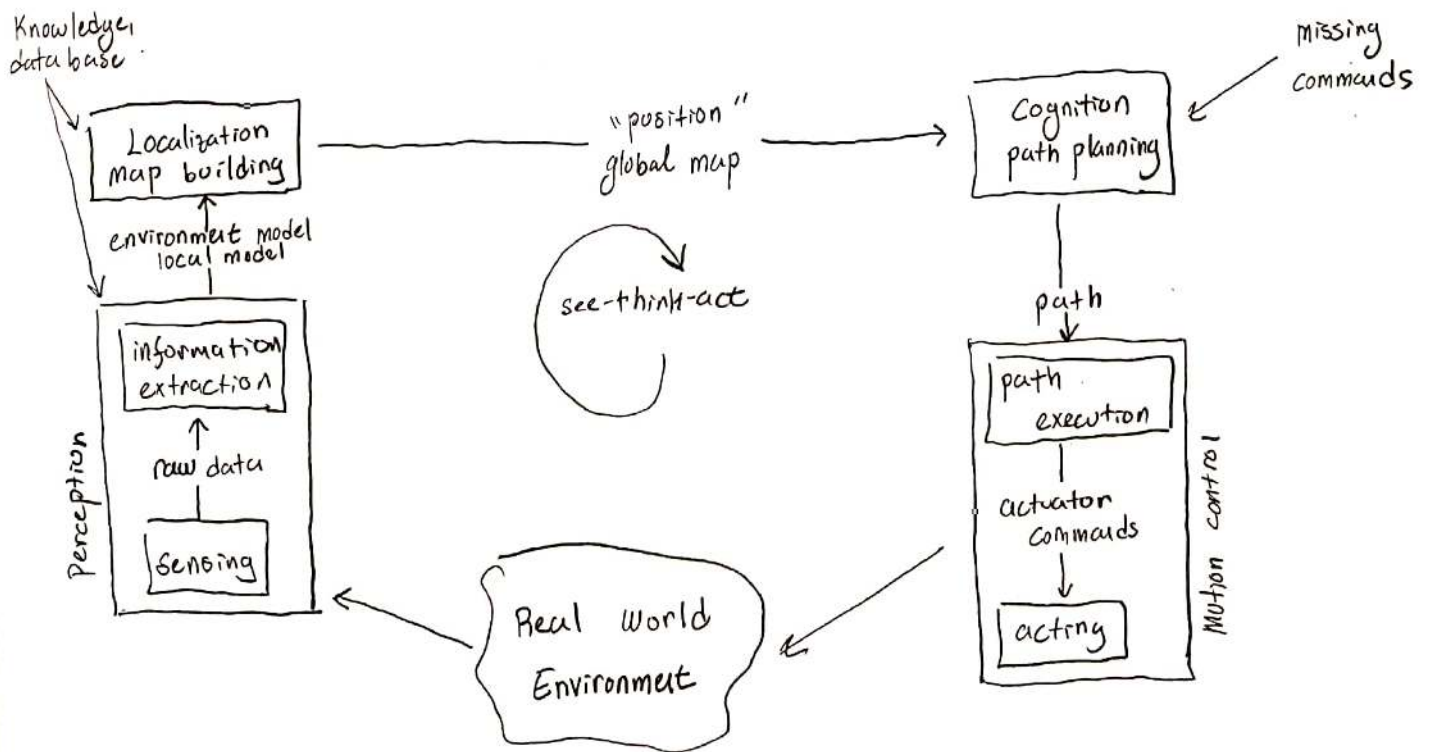
Odometry: The location of the robot according to $x=0, y=0, [x, y, \theta]$ \rightarrow odometry
Odometry is the use of data from motion sensors to estimate change in position over time.

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)

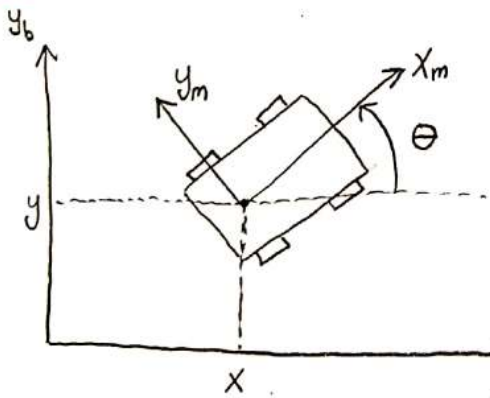
Exploration: Navigation but the robot determines the target point instead of a human. (frontiers)



Wheeled Mobile Robots

- Combination of various physical (hardware) and computational (software) components.
- A collection of subsystems:
 - **Locomotion:** How the robot moves through its environment.
 - **Sensing:** How the robot measures properties of itself and environments.
 - **Controls:** 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

• $\{x_m, y_m\} \rightarrow$ moving frame

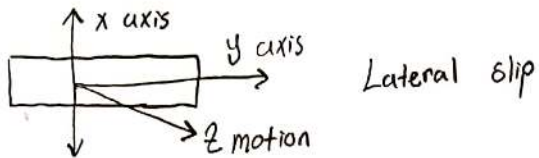
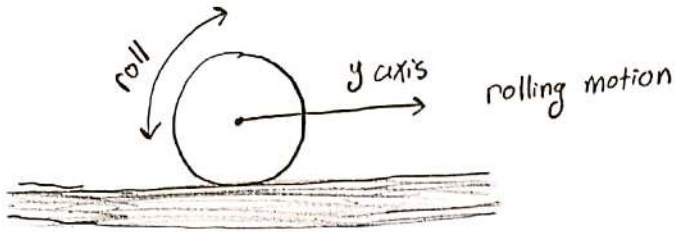
• $\{x_b, y_b\} \rightarrow$ base frame

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

$R(\theta) = \begin{bmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \rightarrow$ rotation matrix expressing the orientation of the base frame with respect to the moving frames

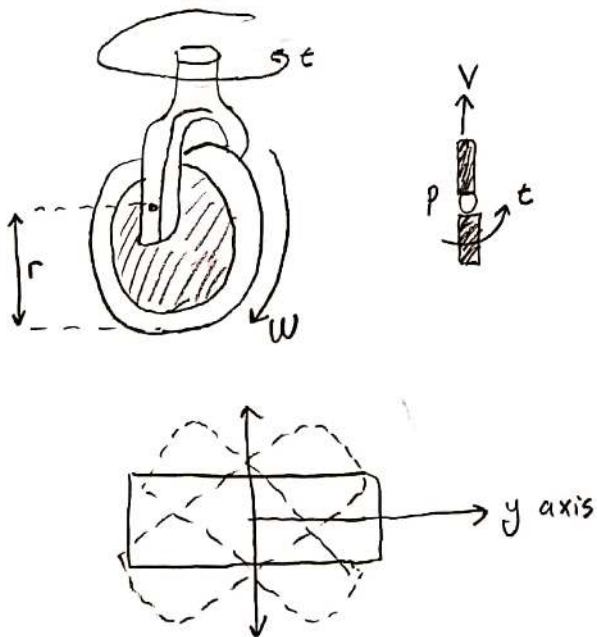
pose
pose to pose:
velocity

Wheels



Steered Wheel

The orientation 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 rolling)
4. The robot contains at most one steering link per wheel.
5. All steering axes are perpendicular to floors.

Robot Wheel Parameters

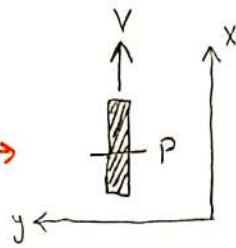
For low velocities, rolling is a reasonable wheel model. This is the model that will be considered in the kinematics models of WMR.

Wheel Parameters:

- r = wheel radius
- v = wheel linear velocity
- ω = wheel angular velocity
- $\dot{\theta}$ = steering velocity.

Wheel Types

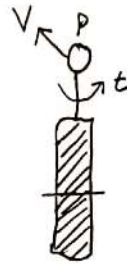
Fixed Wheel \rightarrow



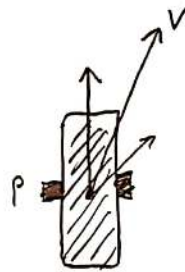
Center Orientable Wheel \rightarrow



Off-Centered Orientable Wheel
(Caster wheel) \rightarrow



Swedish Wheel: Omnidirectional
Property \rightarrow



Fixed Wheel

- Velocity of point P

$$V = (r \times \omega) a_x, \text{ where } a_x: \text{A unit vector to x-axis}$$

- 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

$$V = (r \times \omega) a_x, \text{ where}$$

a_x : A unit vector to x-axis

a_y : A unit vector to y-axis

Off-Centered Orientable Wheels

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

Swedish wheel

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

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.

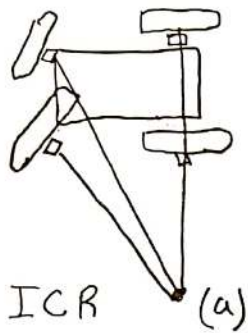
Omnidirectional Robot

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

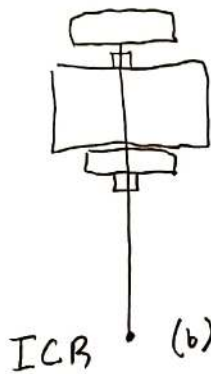
Mobile Robot Locomotion

- instantaneous center of rotation (ICR) or instantaneous center of curvature (ICC)

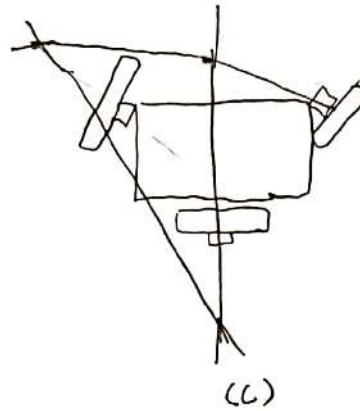
- a cross point of all axes of the wheels



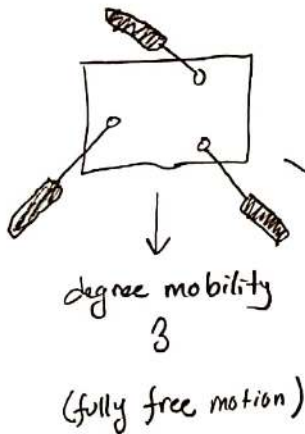
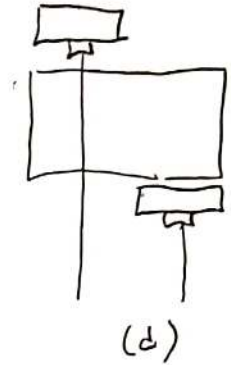
↓
degree mobility 1
(fixed arc motion)



↓
degree mobility 2
(variable arc motion)



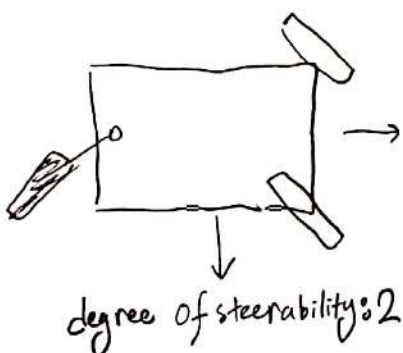
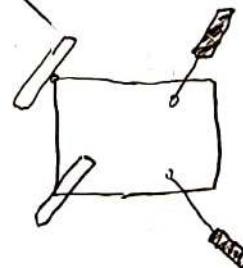
↓
degree mobility 0
(cannot move anywhere)



degree of steerability: 1

→ two mutually dependent centered orientable wheels

degree of steerability: 0
(no center oriented wheels)



Two mutually independent centered orientable wheels.

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 Drives

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

Steered Wheels:

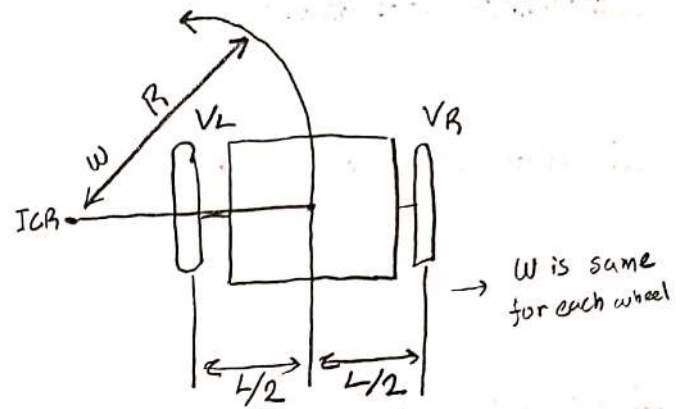
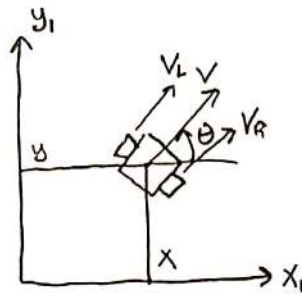
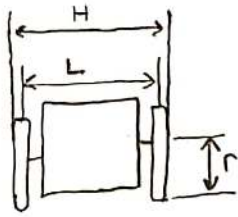
- Steering wheel + rear wheels
- Cannot turn $\pm 90^\circ$
- limited radius of curvature

Synchronous Drive

Omnidirectional

Car Drive

Differential Drive



- Posture of the robot

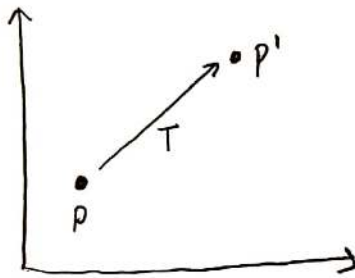
$$p = \begin{pmatrix} x \\ y \\ \theta \end{pmatrix} \quad \begin{array}{l} (x, y) \mapsto \text{position of the robot} \\ \theta \mapsto \text{orientation of the robot} \end{array}$$

- 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

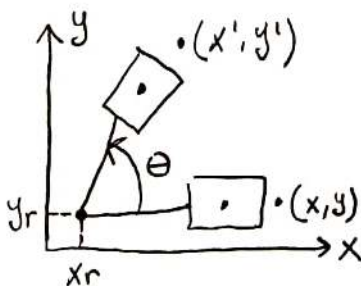


$$x' = x + t_x, \quad y' = y + t_y$$

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

$$p' = p + T$$

2D Rotation



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

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

$$p' = p_r + R \cdot (p - p_r)$$

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

Homogeneous Coordinates

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

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

Sensor Types

- Time of flight (ToF)
 - Sonar
 - laser
 - radar
- Camera
 - RGB
 - Infrared \rightarrow salt and pepper
 - RGB-D \rightarrow point cloud
 - Stereo camera
 - Thermal
- LIDAR sensor : 270°'lik okuma.
0.25°'lik hassasiyet. (derecenin dörtte biri)
2D-3D
- Inertial Measurement Unit (IMU)
 - Accelerometer (orientation, where is north, robots transformation, robots acceleration)
 - Magnetometer
 - Gyroscope
- Ortam algı sensörleri (CO₂, ısı, nem etc)

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.)

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

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

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

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.

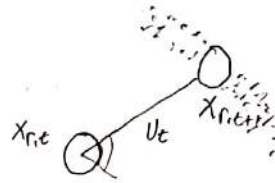
$$\text{bel}_t(x_r) = p(x_{r,t} = x_r | z_{1:t}, u_{0:t-1})$$

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

$$\overline{\text{bel}}_t(x_r) = p(x_{r,t} = x_r | z_{1:t-1}, u_{0:t-1})$$

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:

$$p(x_{r,t+1} | x_{r,t}, u_t)$$



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.

$$p(z_t | x_{r,t})$$

$$\overline{bel}_t(x_r) = \int_{\Omega} p(x_r | x_{r,t-1} = y, u_{t-1}) \cdot bel_{t-1}(y) dy$$

$$bel_t(x_r) = \eta p(z_t | x_{r,t} = x_r) \overline{bel}_t(x_r)$$

Typical Motion Models

- Odometry based \rightarrow wheel encoders
- Velocity based (dead reckoning) \rightarrow no wheel encoders given
 - \rightarrow calculates the new pose based on the velocities and the time elapsed.
 - \rightarrow lower performance
- Odometry is the use of data from motion sensors to estimate change in position over time \rightarrow 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.

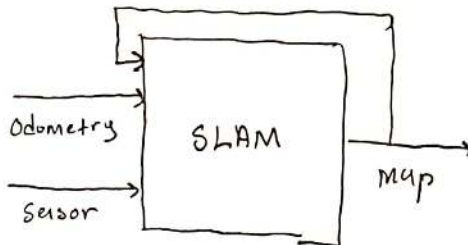
Deterministic (systematic) errors can be eliminated through proper calibration.

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

Errors are integrated, unbounded!!

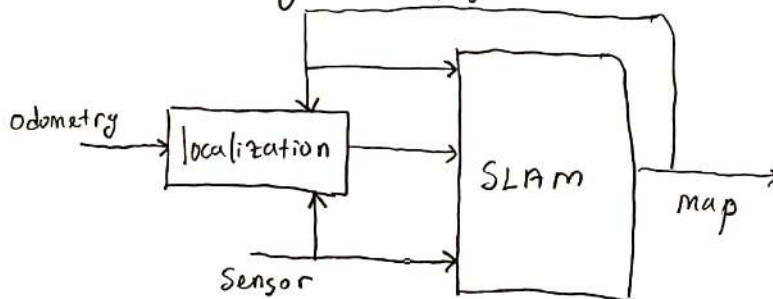
SLAM

- Simultaneous localization and mapping
- if odometry is perfect



mapping \rightarrow full slam

- Since odometry is not perfect



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

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

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

input: odometry, sensor data, previous map

output: updated map

Quaternion - Orientation Format

$$[w, x, y, z] = \left[\cos \frac{\theta}{2}, \sin \frac{\theta}{2} \cdot nx, \sin \frac{\theta}{2} \cdot ny, \sin \frac{\theta}{2} \cdot nz \right]$$

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

- Loop closing

↳ scan matching

↳ deferred validation

↳ Search strategies

↳ Bias:

- Anisotropic point sampling
- Differing sensor fields of view

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.

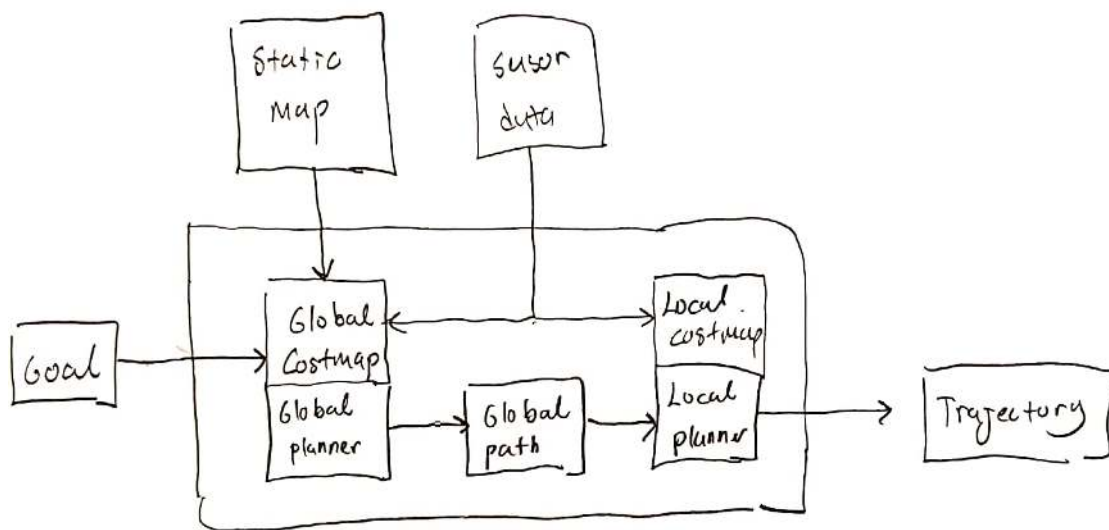
YAML holds additional informations about the map.

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 bots.

↳ 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 \mapsto paths for a goal

Local planner \mapsto paths in the nearby distances
 \mapsto appropriate velocity command

Yakin frontier \rightarrow A* duhd uygun