Robot Teknolojisine Giriş
BLM4830

Öğr. Grv. Furkan ÇAKMAK

Ders İşleyiş Bilgilendirmesi

- Teorik Ders Saatleri: **Salı 09:00 - 11:50**
- Dersler şimdilik online ortamda olacak.
- Duyurular için: https://avesis.yildiz.edu.tr/fcakmak/dokumanlar
  - OBS Toplu Mesaj Sistemi
- Google Classroom Sınıf Kodu: **5lidsgw**
  - Dersle ilgili sorularınızı Classroom üzerinden duvar paylaşımı yaparak sorabilirsiniz.
- Zoom Personal Room:
  - https://us04web.zoom.us/j/3752287039?pwd=TTFKNittZWJTUEhHREovckl0VTVYUT09
  - Meeting ID: 375 228 7039
  - Passcode: 5AXMNd
- Dersle ilgili soru ve danışmalarınız için EN AZ 1 (BİR) GÜN ÖNCEDEN linkten randevu alınız.
  - https://fcakmak.simplybook.it/v2/
  - Mail ile randevu verilmeyecektir.
  - Alınan randevu saatinde lütfen yukarıda verilen Zoom odasında bulununuz.

Ders Tanıtım Formu ve Konular

Hafta	Tarih	Konular
1	28.02.2023	Ders Tanıtımı, ROS ve Platform Tanıtımı, Robot Çeşitleri ve Robotik Konuları Başlangıcı
2	7.03.2023	ROS Platformu Tanıtımı - 1
3	14.03.2023	ROS Platformu Tanıtımı - 2
4	21.03.2023	Kinematik - Genel Tanımlar - Diferansiyel Sürüşlü Robot İçin Hesaplama Örnekleri
5	28.03.2023	Sensörler - Çeşitleri ve Çalışma Sistematikleri ve Uygulamaları
6	4.04.2023	Odometri ve Lokalizasyon Kavramları
7	11.04.2023	Haritalama Yöntemleri ve Uygulamaları (1. Ödev Teslimi)
8	18.04.2023	1. Ödev Gösterimi
9	25.04.2023	Ara Sınav (2. Ödev Teslimi)
10	2.05.2023	2. Ödev Gösterimi
11	9.05.2023	Navigasyon ve Keşif Yaklaşımları ve Uygulamaları (3. Ödev Teslimi)
12	16.05.2023	3. Ödev Gösterimi
13	23.05.2023	Robot Üzerinden Görüntü İşleme Teknikleri
14	30.05.2023	3B Haritalama Yöntemleri

Ders Tanıtım Formu – Ders Kitabı

BLM4830
Robot
Teknolojisine
Giriş
Hafta 1

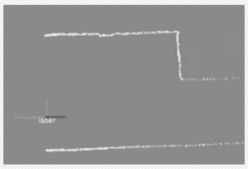
	Kitaplar
1	Introduction to Autonomous Mobile Robots. Roland Siegwart and Illah R. Nourbakhsh. A Bradford Book. The MIT Press.
2	A Gentle Introduction to ROS, JasonM. O'Kane

Öğr. Grv. Furkan ÇAKMAK

Ders Tanıtım Formu – Değerlendirme

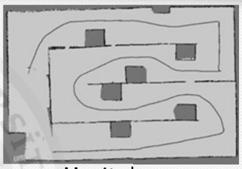
	Yöntem	Adedi	Etki Oranı (%)
Başarı Değerlendirme Sistemi	Ara Sınavlar	1	15
	Kısa Sınavlar	-	-
	Ödevler	3	45
	Projeler	1	25
	Dönem Ödevi	-	-
	Laboratuvar	-	-
	Diğer	-	-
	Final Sınavı	1	15

Ders Boyunca Neler Öğreneceğiz?

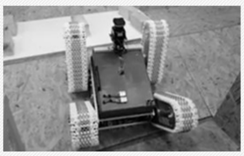

Odometri

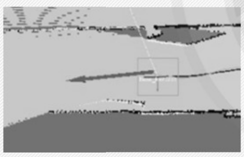

Lokalizasyon Takibi


Lokalize Olma

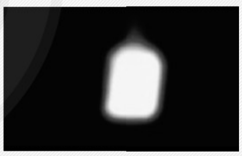

Haritalama

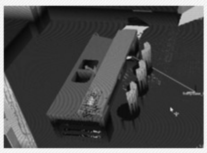

Kapsamlı Haritalama

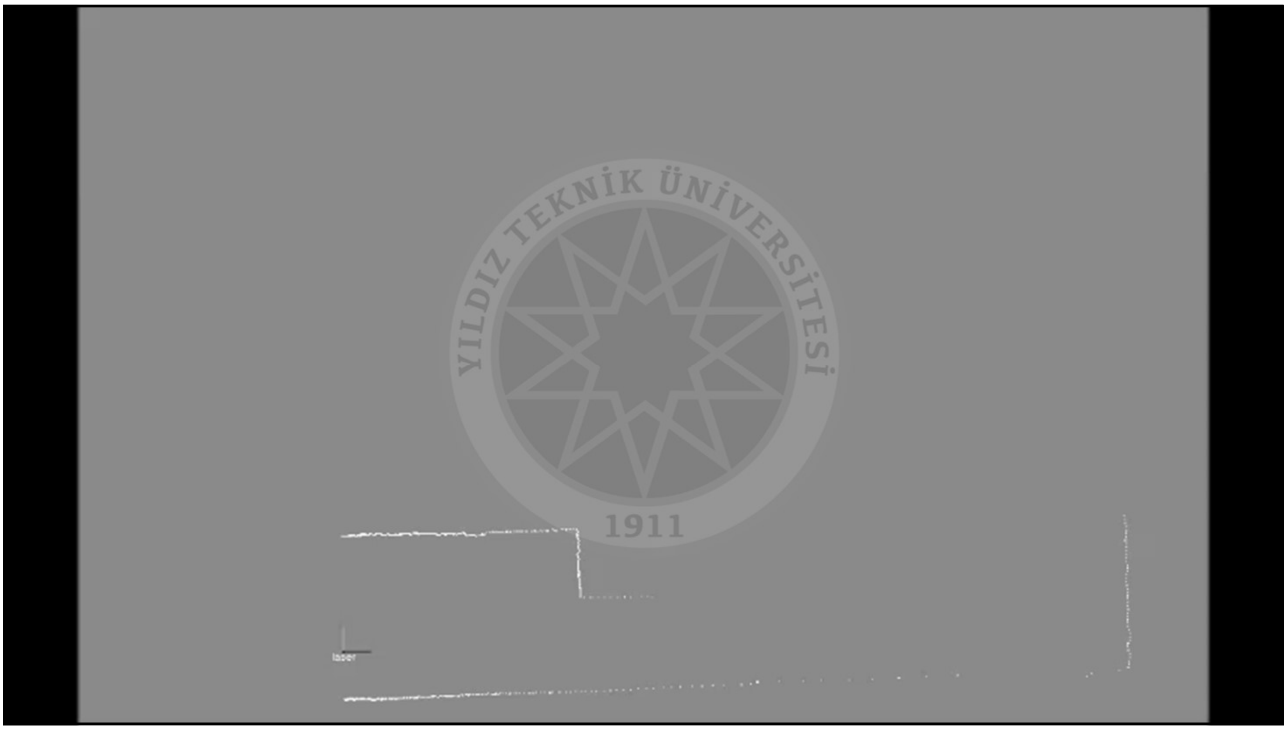

Paletli Haritalama


Navigasyon

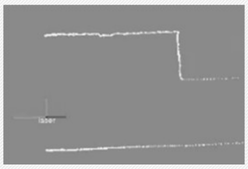

Keşif


Termal Keşif


3B Haritalama

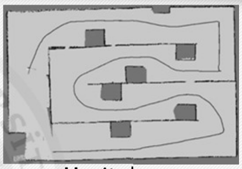


Ders Boyunca Neler Öğreneceğiz?

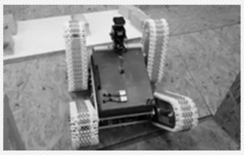

Odometri

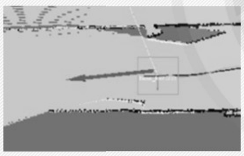

Lokalizasyon Takibi


Lokalize Olma

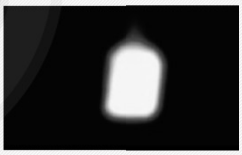

Haritalama

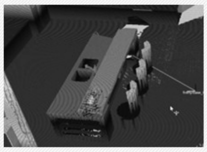

Kapsamlı Haritalama

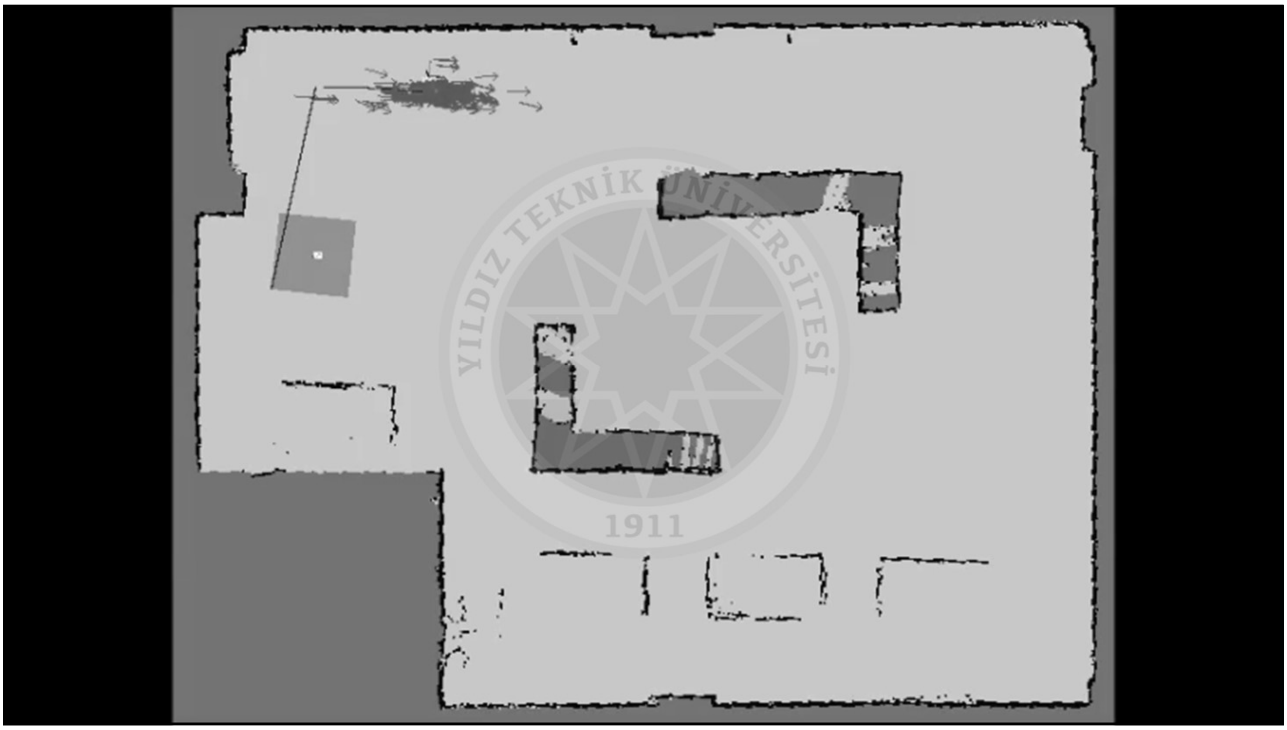

Paletli Haritalama


Navigasyon

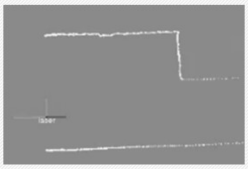

Keşif


Termal Keşif


3B Haritalama

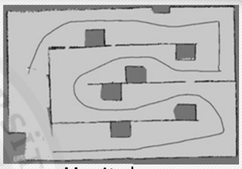


Ders Boyunca Neler Öğreneceğiz?

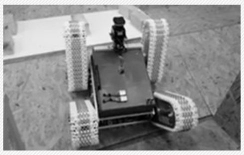

Odometri

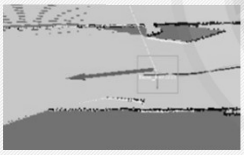

Lokalizasyon Takibi


Lokalize Olma

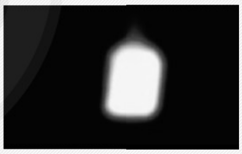

Haritalama

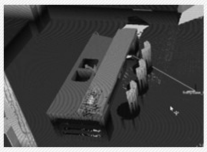

Kapsamlı Haritalama

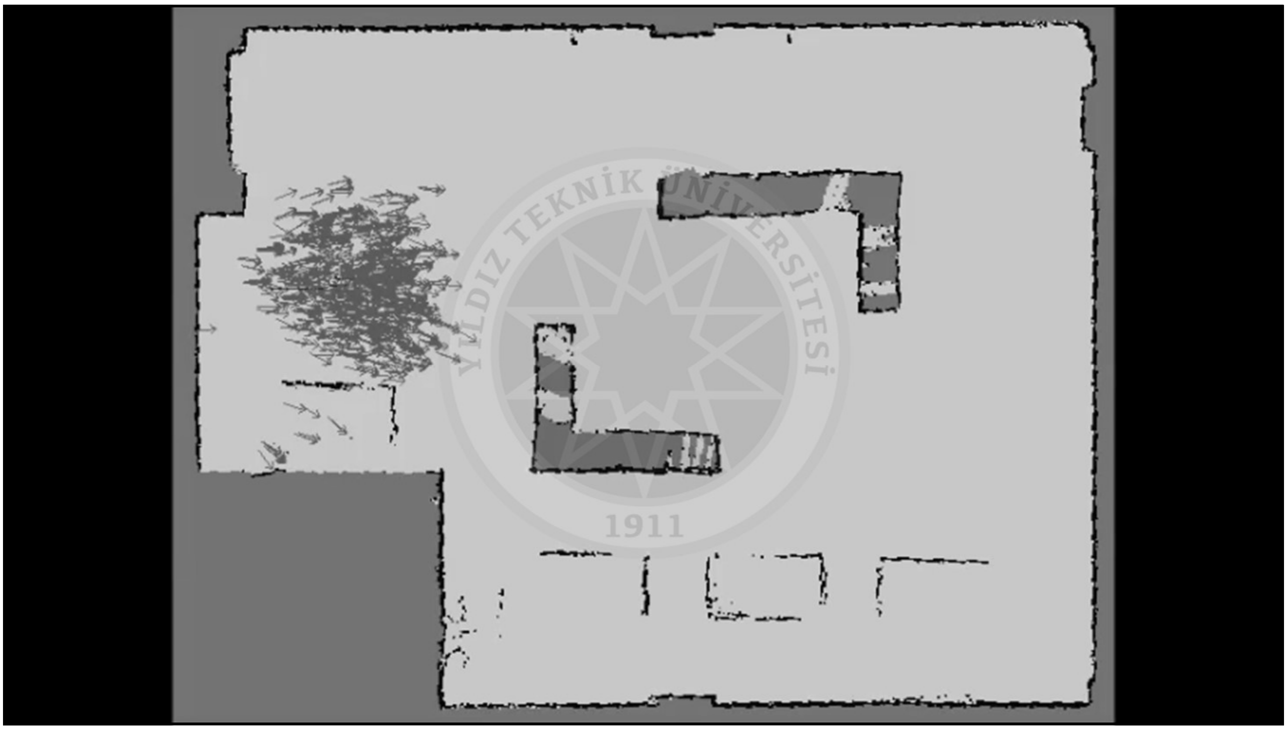

Paletli Haritalama


Navigasyon


Keşif

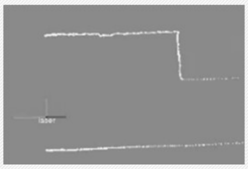

Termal Keşif


3B Haritalama



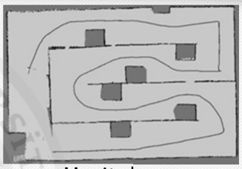
Ders Boyunca Neler Öğreneceğiz?

BLM4830
Robot
Teknolojisine
Giriş
Hafta 1

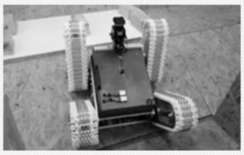

Odometri

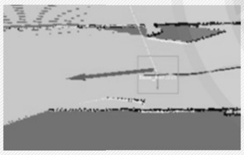

Lokalizasyon Takibi


Lokalize Olma

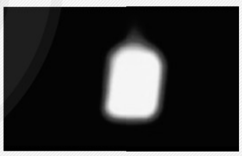

Haritalama

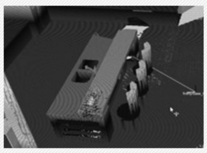

Kapsamlı Haritalama


Paletli Haritalama


Navigasyon

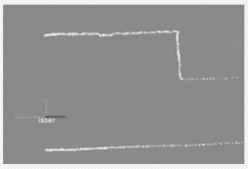

Keşif


Termal Keşif


3B Haritalama

Ders Boyunca Neler Öğreneceğiz?

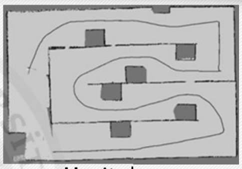
BLM4830
Robot
Teknolojisine
Giriş
Hafta 1



Odometri

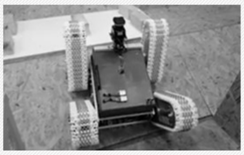
Lokalizasyon Takibi

Lokalize Olma

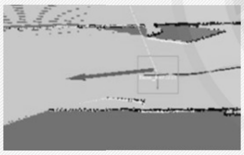


Haritalama

Kapsamlı Haritalama

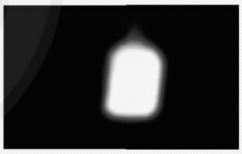


Paletli Haritalama

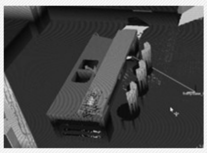


Navigasyon

Keşif



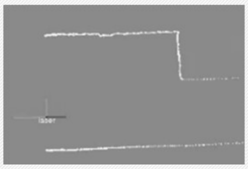
Termal Keşif



3B Haritalama

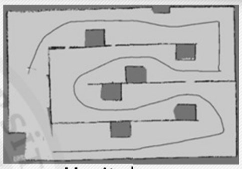
Ders Boyunca Neler Öğreneceğiz?

BLM4830
Robot
Teknolojisine
Giriş
Hafta 1

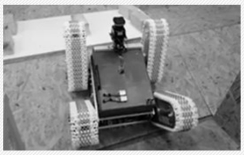

Odometri

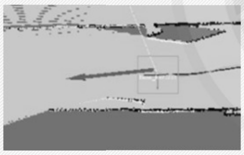

Lokalizasyon Takibi


Lokalize Olma

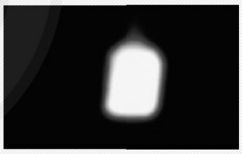

Haritalama

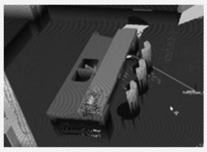

Kapsamlı Haritalama


Paletli Haritalama

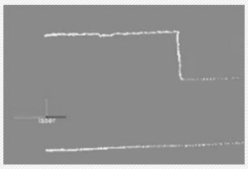

Navigasyon


Keşif


Termal Keşif


3B Haritalama

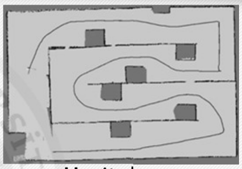
Ders Boyunca Neler Öğreneceğiz?



Odometri

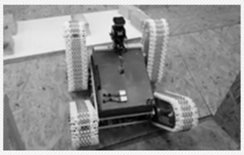
Lokalizasyon Takibi

Lokalize Olma

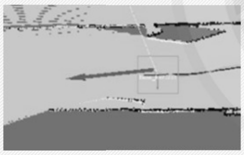


Haritalama

Kapsamlı Haritalama

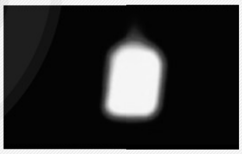


Paletli Haritalama

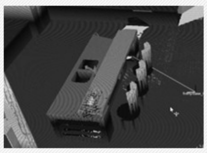


Navigasyon

Keşif

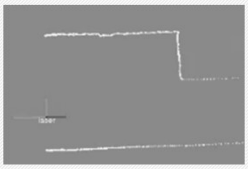


Termal Keşif



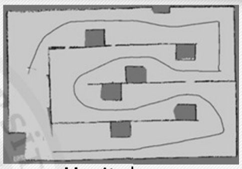
3B Haritalama

Ders Boyunca Neler Öğreneceğiz?

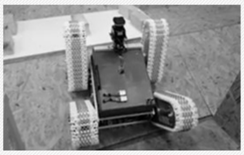

Odometri

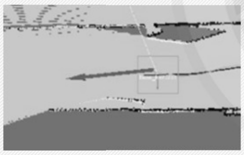

Lokalizasyon Takibi


Lokalize Olma

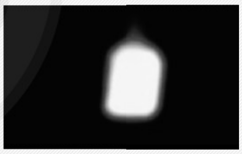

Haritalama

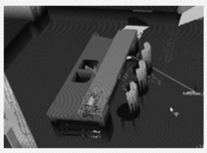

Kapsamlı Haritalama


Paletli Haritalama


Navigasyon

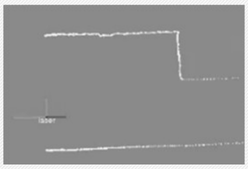

Keşif


Termal Keşif


3B Haritalama

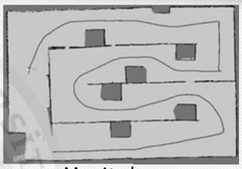
Ders Boyunca Neler Öğreneceğiz?

BLM4830
Robot
Teknolojisine
Giriş
Hafta 1

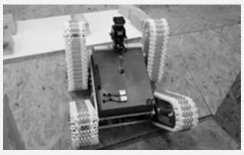

Odometri

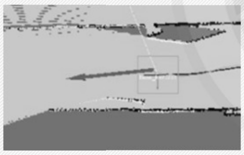

Lokalizasyon Takibi


Lokalize Olma

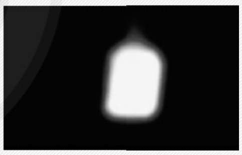

Haritalama

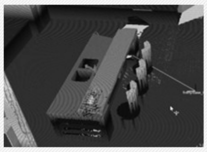

Kapsamlı Haritalama


Paletli Haritalama

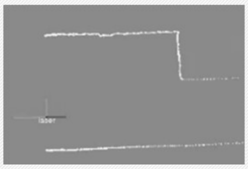

Navigasyon


Keşif


Termal Keşif

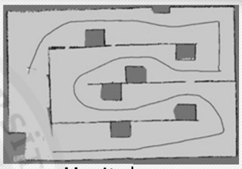

3B Haritalama

Ders Boyunca Neler Öğreneceğiz?

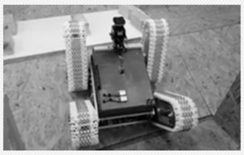

Odometri

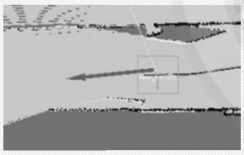

Lokalizasyon Takibi


Lokalize Olma

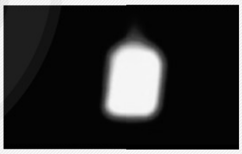

Haritalama

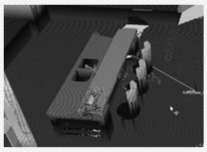

Kapsamlı Haritalama

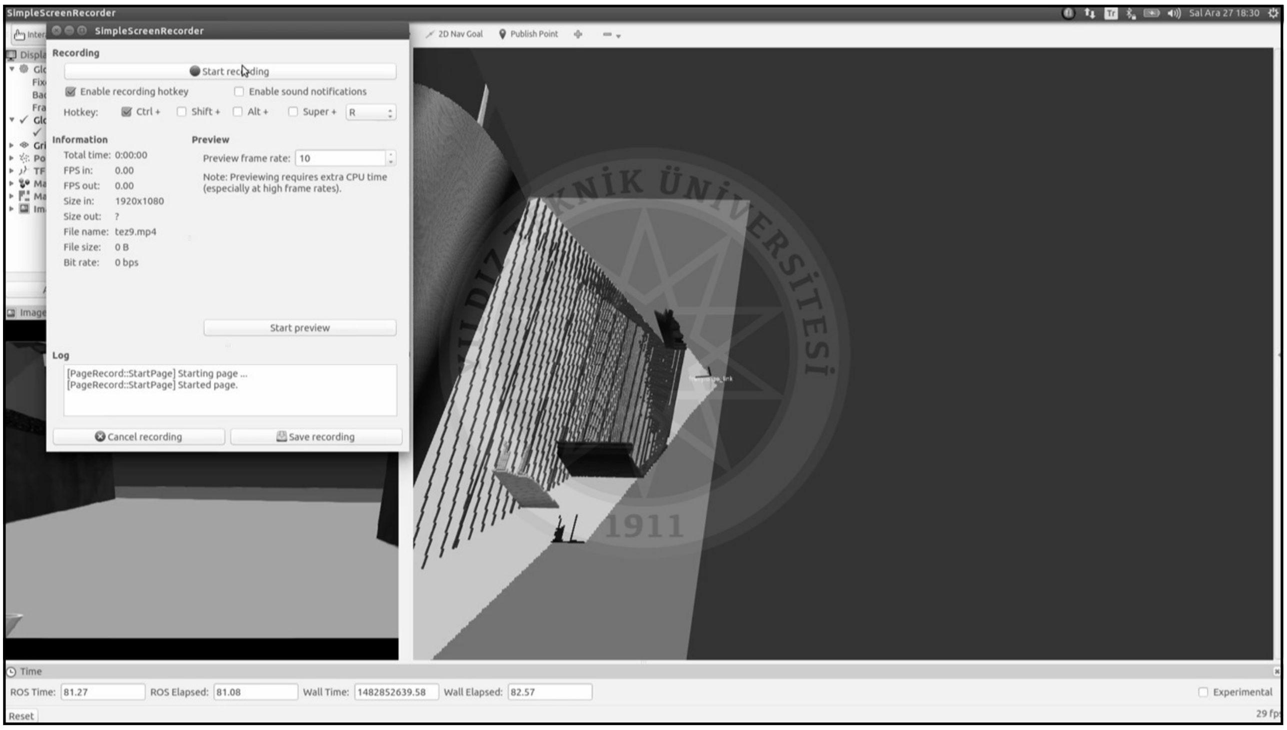

Paletli Haritalama


Navigasyon


Keşif


Termal Keşif


3B Haritalama



Geliştirme Ortalamı

BLM4830
Robot
Teknolojisine
Giriş
Hafta 1

- Ubuntu 20.04 Focal
- ROS Noetic Ninjemys
- C++
- Python

ROS/ Tutorials

Öğr. Grv. Furkan ÇAKMAK

ROS Ortam Robotları

- https://robots.ros.org/

Husky

P3AT

Sabırla Dinlediğiniz İçin Teşekkürler

Robot Teknolojisine Giriş BLM4830

Öğr. Grv. Furkan ÇAKMAK

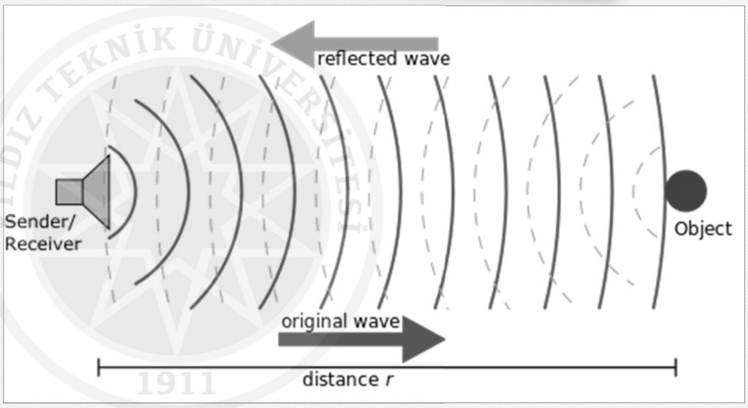
Ders Tanıtım Formu ve Konular

Hafta	Tarih	Konular
1	28.02.2023	Ders Tanıtımı, ROS ve Platform Tanıtımı, Robot Çeşitleri ve Robotik Konuları Başlangıcı
2	7.03.2023	ROS Platformu Tanıtımı - 1
3	14.03.2023	ROS Platformu Tanıtımı - 2
4	21.03.2023	Sensörler - Çeşitleri ve Çalışma Sistematikleri ve Uygulamaları
5	28.03.2023	Kinematik - Genel Tanımlar - Diferansiyel Sürüşlü Robot İçin Hesaplama Örnekleri
6	4.04.2023	Odometri ve Lokalizasyon Kavramları
7	11.04.2023	Haritalama Yöntemleri ve Uygulamaları (1. Ödev Teslimi)
8	18.04.2023	1. Ödev Gösterimi
9	25.04.2023	Ara Sınav (2. Ödev Teslimi)
10	2.05.2023	2. Ödev Gösterimi
11	9.05.2023	Navigasyon ve Keşif Yaklaşımları ve Uygulamaları (3. Ödev Teslimi)
12	16.05.2023	3. Ödev Gösterimi
13	23.05.2023	Robot Üzerinden Görüntü İşleme Teknikleri
14	30.05.2023	3B Haritalama Yöntemleri

Sensör Çeşitleri

- ToF
  - Sonar
  - Lazer
  - Radar
- Kamera
  - RGB
  - Infrared
  - RGB-D
  - Stereo Camera
  - Thermal
- IMU (Inertial Measurement Unit)
  - Accelerometer
  - Magnetometer
  - Gyroscope
- Ortam algı sensörleri
  - Işık, nem, su seviyesi, CO, CO_2, sıcaklık, dokunma vb.

ToF - Sonar, Lazer, Radar



LIDAR Sensor

BLM4830
Robot
Teknolojisine
Giriş
Hafta 4

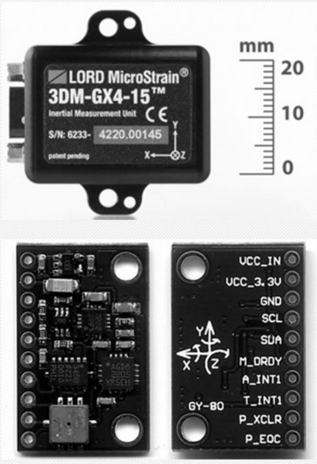
Kamera

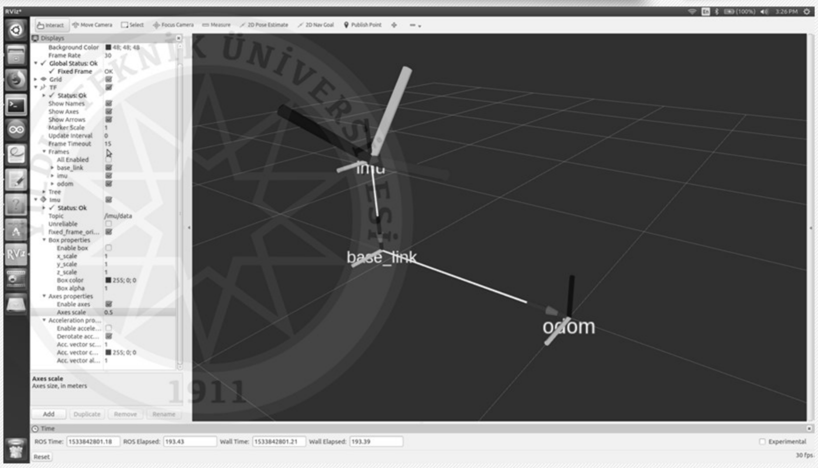
BLM4830
Robot
Teknolojisine
Giriş
Hafta 4

Öğr. Grv. Furkan ÇAKMAK

IMU

BLM4830 Robot Teknolojisine Giriş Hafta 4

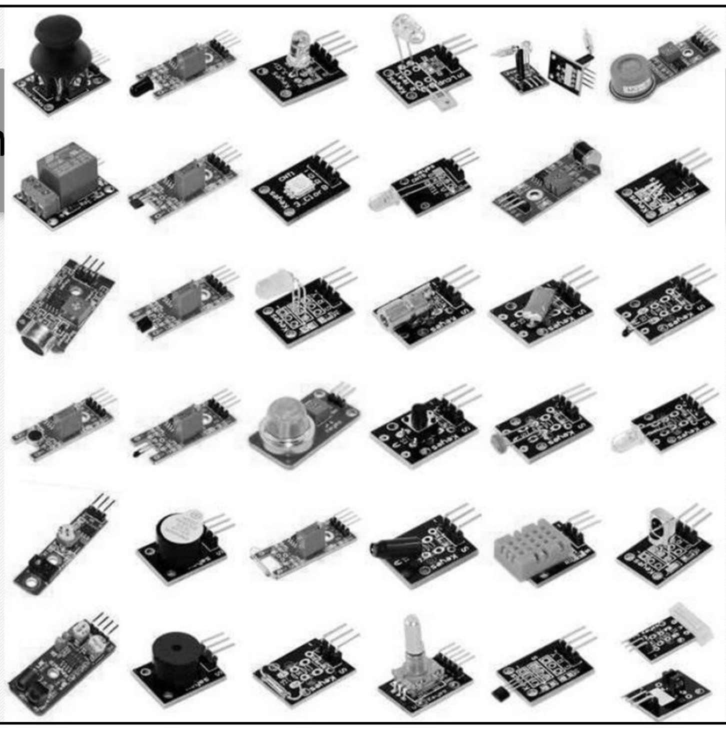




Öğr. Grv. Furkan ÇAKMAK

Diğer Sen

BLM4830 Robot Teknolojisine Giriş Hafta 4



Sabırla Dinlediğiniz İçin Teşekkürler

Robot Teknolojisine Giriş BLM4830

Öğr. Grv. Furkan ÇAKMAK

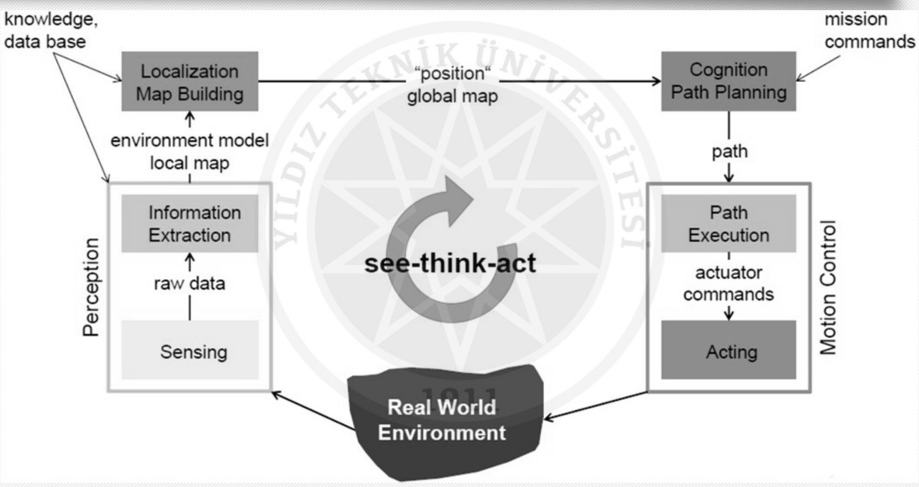
Ders Tanıtım Formu ve Konular

BLM4830 Robot Teknolojisine Giriş Hafta 5

Hafta	Tarih	Konular
1	28.02.2023	Ders Tanıtımı, ROS ve Platform Tanıtımı, Robot Çeşitleri ve Robotik Konuları Başlangıcı
2	7.03.2023	ROS Platformu Tanıtımı - 1
3	14.03.2023	ROS Platformu Tanıtımı - 2
4	21.03.2023	ROS Platformu Tanıtımı - 3
5	28.03.2023	Sensörler - Çeşitleri ve Çalışma Sistematikleri ve Uygulamaları
6	4.04.2023	Kinematik - Genel Tanımlar - Diferansiyel Sürüşlü Robot İçin Hesaplama Örnekleri
7	11.04.2023	Odometri ve Lokalizasyon Kavramları (1. Ödev Teslimi)
8	18.04.2023	Ara Sınav (1. Ödev Gösterimi)
9	25.04.2023	Haritalama Yöntemleri ve Uygulamaları (2. Ödev Teslimi)
10	2.05.2023	2. Ödev Gösterimi
11	9.05.2023	Navigasyon ve Keşif Yaklaşımları ve Uygulamaları (3. Ödev Teslimi)
12	16.05.2023	3. Ödev Gösterimi
13	23.05.2023	Robot Üzerinden Görüntü İşleme Teknikleri
14	30.05.2023	3B Haritalama Yöntemleri

See-Think-Act Cycle

BLM4830
Robot
Teknolojisine
Giriş
Hafta 5



Wheeled Mobile Robots

BLM4830
Robot
Teknolojisine
Giriş
Hafta 5

- 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 its environment.
  - **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.

Wheeled Mobile Robots

Contents

- Introduction
- Classification of wheels
  - Fixed wheel
  - Centered orientable wheel
  - Off-centered orientable wheel
  - Swedish wheel
- Mobile Robot Locomotion
  - Differential Drive
  - Tricycle
  - Synchronous Drive
  - Omni-directional
  - Ackerman Steering
- Kinematics models of WMR
- Summary

Locomotion






- **Locomotion** is the process of causing an autonomous robot to move
  - In order to produce motion, forces must be applied to the vehicle

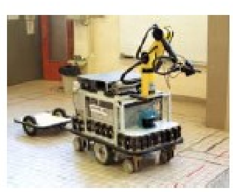
Wheeled Mobile Robots (WMR)


Yamabico


MagellanPro


Sojourner


ATRV-2


Hilare 2-Bis


Koy

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 its environment
  - **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

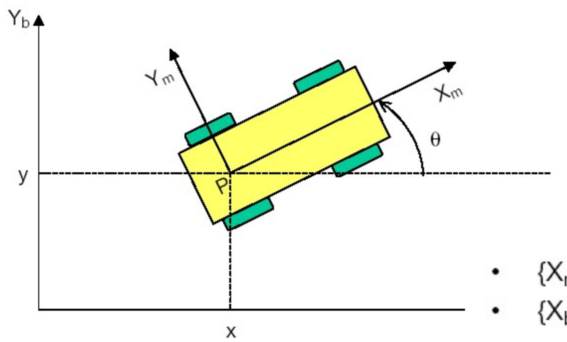
Wheeled Mobile Robots

- **Locomotion —** the process of causing an robot to move.
  - In order to produce motion, forces must be applied to the robot
  - Motor output, payload

- **Kinematics** – study of the mathematics of motion without considering the forces that affect the motion.
  - Deals with the geometric relationships that govern the system
  - Deals with the relationship between control parameters and the behavior of a system.

- **Dynamics** – study of motion in which these forces are modeled
  - Deals with the relationship between force and motions.

Notation



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

- $\{X_m, Y_m\}$ – moving frame
- $\{X_b, Y_b\}$ – base frame

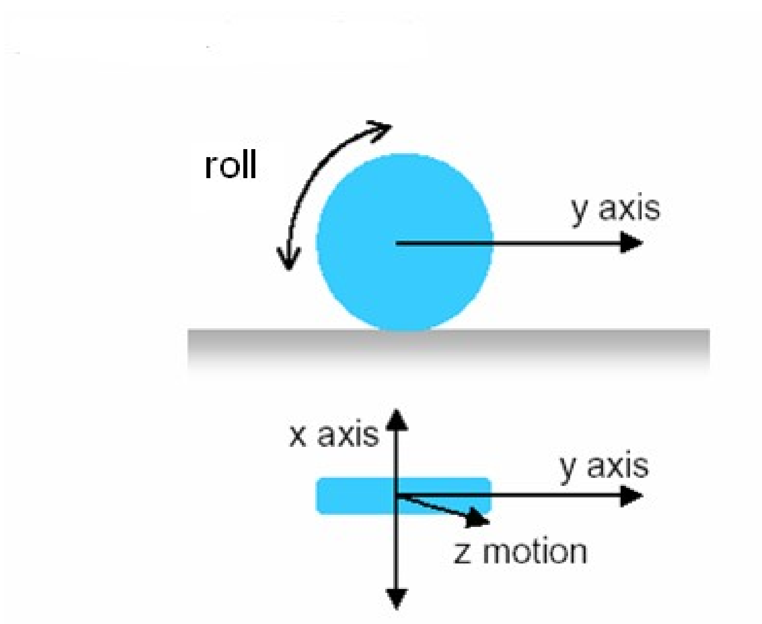
$$q = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix}$$

robot posture in base frame

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

Rotation matrix expressing the orientation of the base frame with respect to the moving frame

Wheels

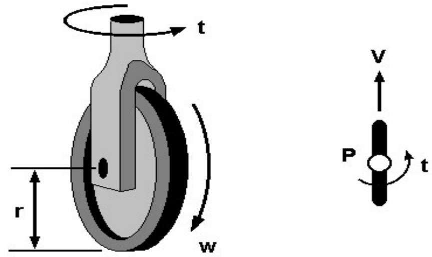


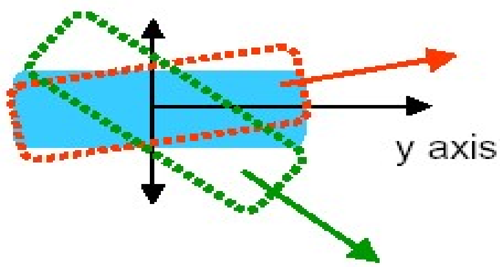
Rolling motion

Lateral slip

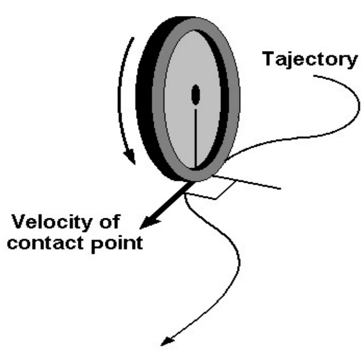
Steered Wheel

- **Steered wheel**
  - The orientation of the rotation axis can be controlled





Idealized Rolling Wheel



Non-slipping and pure rolling

- **Assumptions**

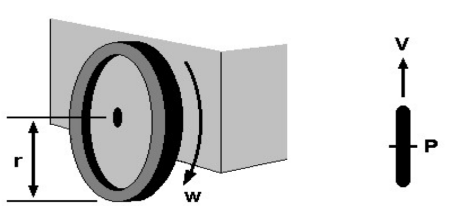
1. The robot is built from rigid mechanisms.
2. No slip occurs in the orthogonal direction of rolling (non-slipping).
3. No translational slip occurs between the wheel and the floor (pure rolling).
4. The robot contains at most one steering link per wheel.
5. All steering axes are perpendicular to the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

Robot wheel parameters

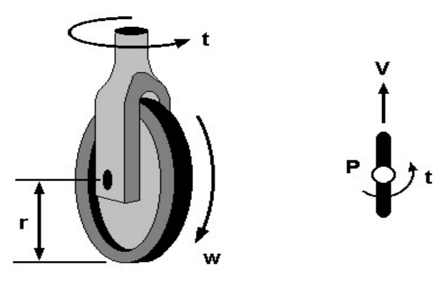
- For low velocities, rolling is a reasonable wheel model.
  - This is the model that will be considered in the kinematics models of WMR
- Wheel parameters:
  - r = wheel radius
  - v = wheel linear velocity
  - w = wheel angular velocity
  - t = steering velocity

Wheel Types

Fixed wheel

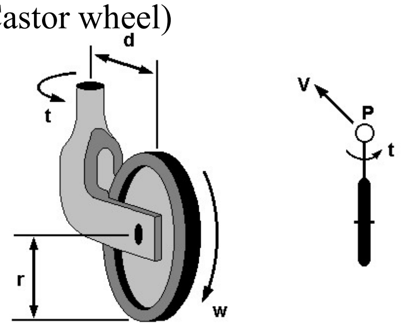


Centered orientable wheel

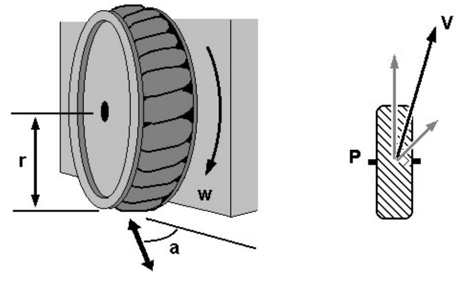


Off-centered orientable wheel

(Castor wheel)



Swedish wheel:omnidirectional property



Fixed wheel

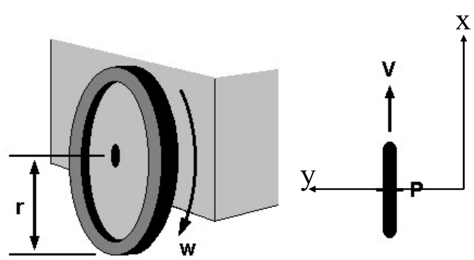
- Velocity of point **P**

$$V = (r \times w)a_x$$

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.



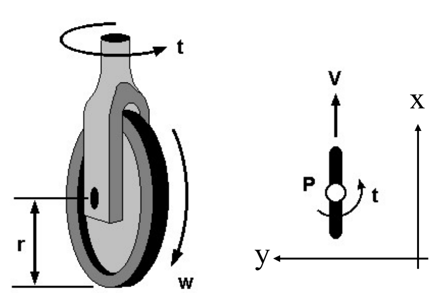
Centered orientable wheels

- Velocity of point **P**

$$V = (r \times w)a_x$$

where, a_x : A unit vector of x axis
a_y : A unit vector of y axis

- Restriction to the robot mobility



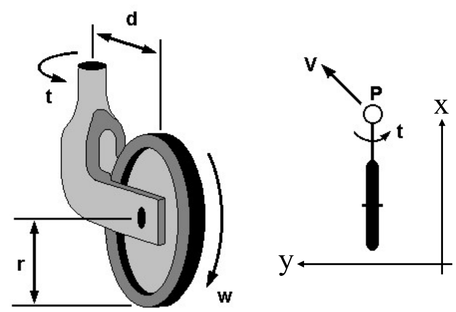
Off-Centered Orientable Wheels

– Velocity of point **P**

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

where, a_x : A unit vector of x axis
a_y : A unit vector of y axis

– Restriction to the robot mobility



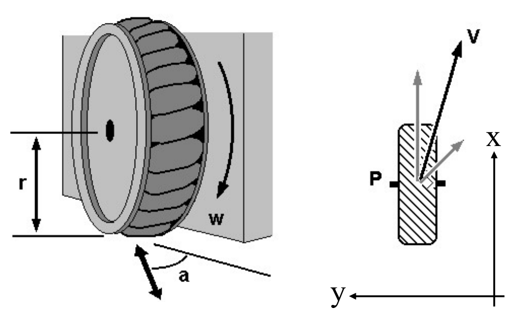
Swedish wheel

– Velocity of point **P**

$$v = (r \times w)a_x + Ua_s$$

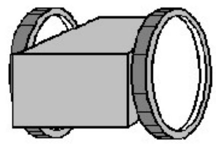
where, a_x : A unit vector of x axis
a_s : A unit vector to the motion of roller

– Omnidirectional property



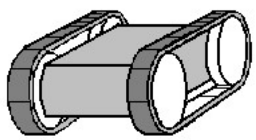
Examples of WMR

Example



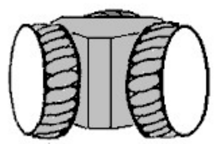
Bi-wheel type robot

- Smooth motion
- Risk of slipping
- Some times use roller-ball to make balance



Caterpillar type robot

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

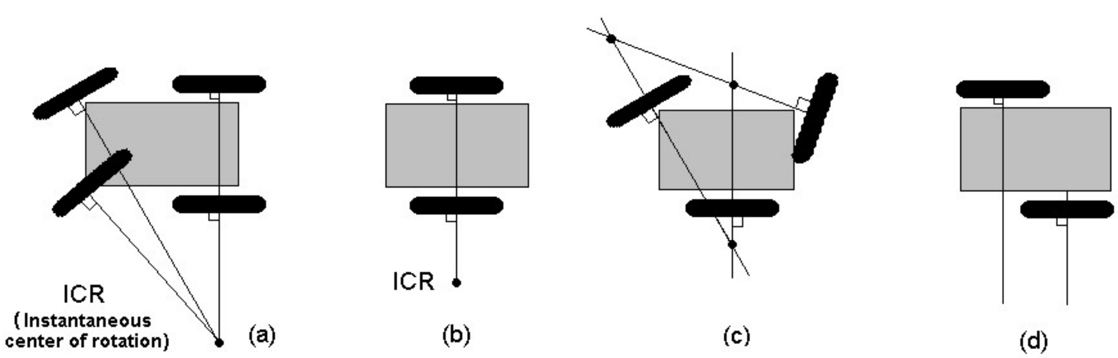


Omnidirectional robot

- 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

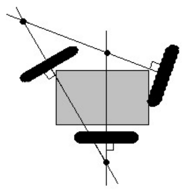


We talk about the instantaneous center, because we'll analyze this at each instant- the curve may, and probably will, change in the next moment.

Degree of Mobility

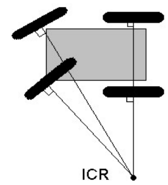
- **Degree of mobility**

  The degree of freedom of the robot motion



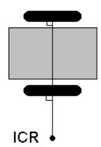
Cannot move anywhere (No ICR)

- Degree of mobility : 0



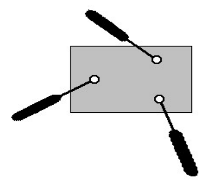
Fixed arc motion (Only one ICR)

- Degree of mobility : 1



Variable arc motion (line of ICRs)

- Degree of mobility : 2



Fully free motion

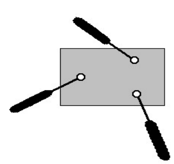
(ICR can be located at any position)

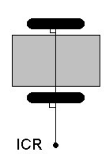
- Degree of mobility : 3

Degree of Steerability

- **Degree of steerability**

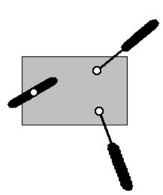
  The number of centered orientable wheels that can be steered independently in order to steer the robot



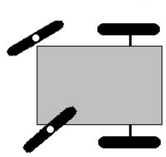


No centered orientable wheels

- Degree of steerability : 0

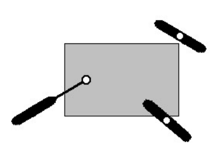


One centered orientable wheel



Two mutually dependent centered orientable wheels

- Degree of steerability : 1



Two mutually independent centered orientable wheels

- Degree of steerability : 2

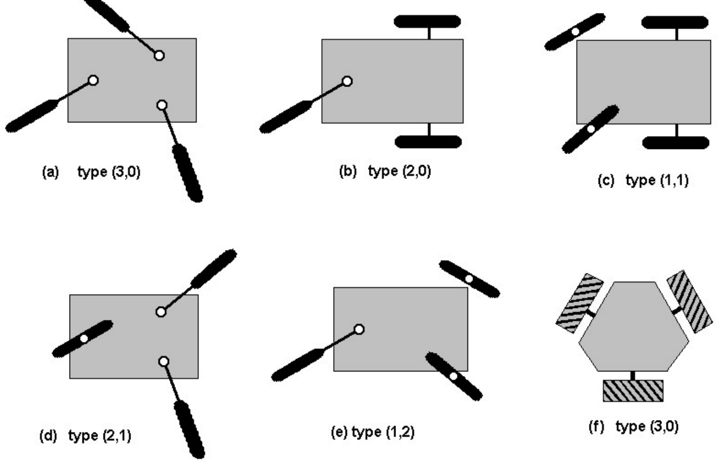
Degree of Maneuverability

- The overall degrees of freedom that a robot can manipulate:

$$\delta_M = \delta_m + \delta_s$$

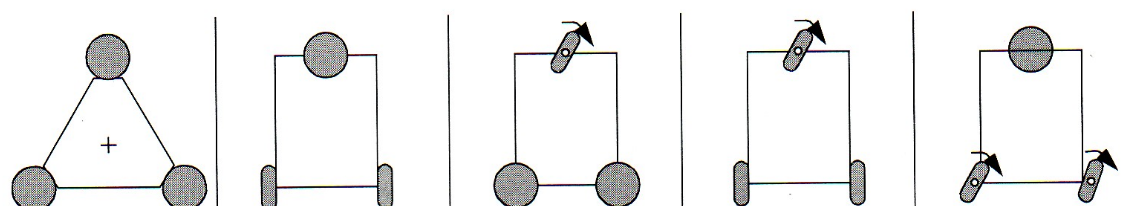
Degree of Mobility	3	2	2	1	1
Degree of Steerability	0	0	1	1	2

- Examples of robot types (degree of mobility, degree of steerability)



Degree of Maneuverability

$$\delta_M = \delta_m + \delta_s$$



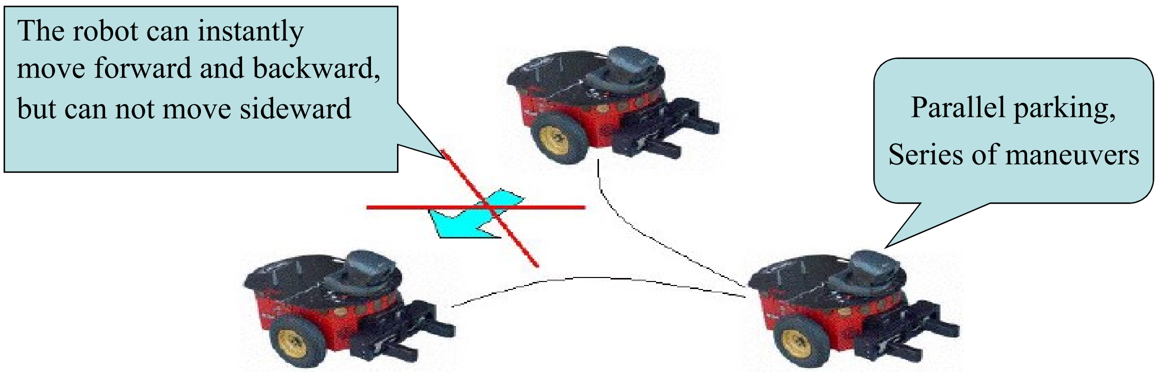
Omnidirectional	Differential	Omni-Steer	Tricycle	Two-Steer
δ_M =3	δ_M =2	δ_M =3	δ_M =2	δ_M =3
δ_m =3	δ_m =2	δ_m =2	δ_m =1	δ_m =1
δ_s =0	δ_s =0	δ_s =1	δ_s =1	δ_s =2

Non-holonomic constraint

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

So what does that mean?

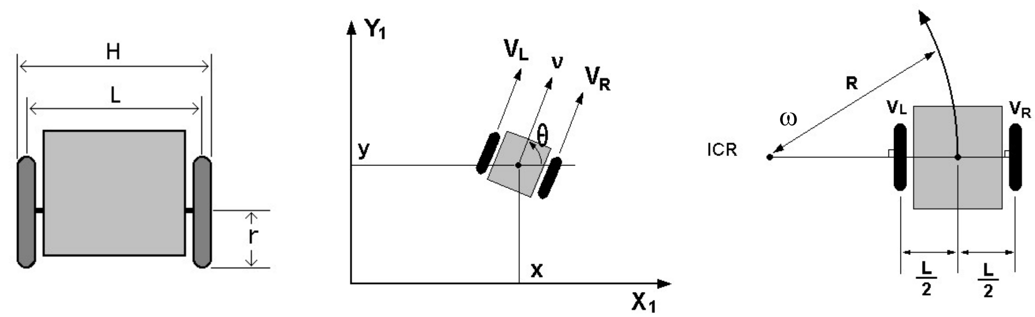
Your robot can move in some directions (forward and backward), but not others (sideward).



Mobile Robot Locomotion

- Differential Drive
  - two driving wheels (plus roller-ball for balance)
  - simplest drive mechanism
  - sensitive to the relative velocity of the two wheels (small error result in different trajectories, not just speed)
- Steered wheels (tricycle, bicycles, wagon)
  - Steering wheel + rear wheels
  - cannot turn $\pm 90^o$
  - limited radius of curvature
- Synchronous Drive
- Omni-directional
- Car Drive (Ackerman Steering)

Differential Drive



- Posture of the robot

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

(x,y) : Position of the robot

θ : Orientation of the robot

- Control input

$$U = \begin{pmatrix} v \\ w \end{pmatrix}$$

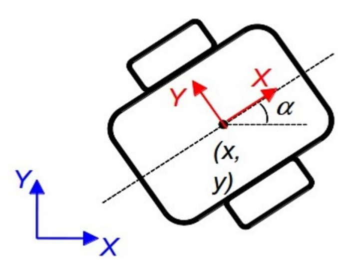
v : Linear velocity of the **robot**

w : Angular velocity of the **robot**

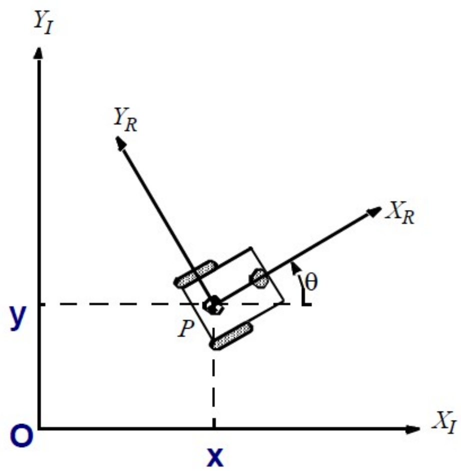
(notice: not for each wheel)

Differential Drive

- Two wheels, either side of the robot driven independently
- Steering is achieved by driving the wheels at different speeds
- Two degrees of freedom -> 2 controllable values: (V_l, V_r)



Robot Reference Frame

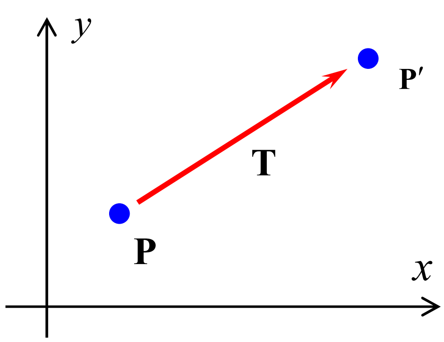


Robot position

- The robot's reference frame is three dimensional including position on the plane and the orientation, $\{X_R, Y_R, \theta\}$
- The axes $\{X_I, Y_I\}$, define inertial global reference frame with origin, **O**
- The angular difference between the global and reference frames is θ
- Point **P** on the robot chassis in the global reference frame is specified by coordinates **(x, y)**

$$\xi_I = \begin{bmatrix} x & y & \theta \end{bmatrix}^T$$

2D Translation

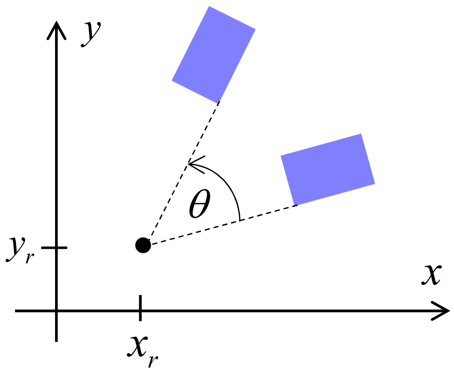


$$x' = x + t_x, \quad y' = y + t_y$$

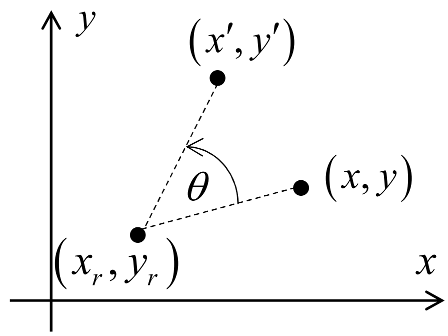
$$\mathbf{P} = \begin{bmatrix} x \\ y \end{bmatrix}, \quad \mathbf{P}' = \begin{bmatrix} x' \\ y' \end{bmatrix}, \quad \mathbf{T} = \begin{bmatrix} t_x \\ t_y \end{bmatrix}$$

$$\mathbf{P}' = \mathbf{P} + \mathbf{T}$$

2D Rotation



Rotation in angle θ about a pivot (rotation) point (x_r, y_r).



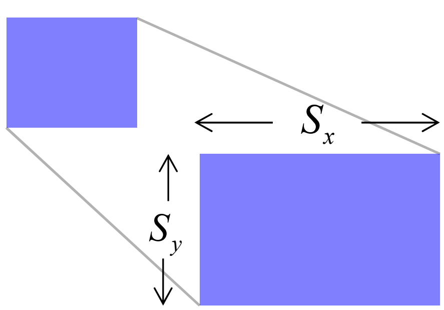
$$x' = x_r + (x - x_r)\cos\theta - (y - y_r)\sin\theta$$

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

$$\mathbf{P}' = \mathbf{P}_r + \mathbf{R} \cdot (\mathbf{P} - \mathbf{P}_r)$$

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

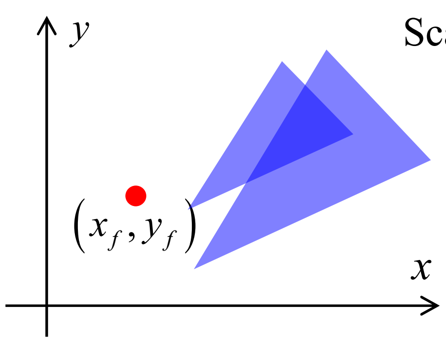
2D Scaling



$$x' = x \cdot s_x, \quad y' = y \cdot s_y$$

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

$$\mathbf{P}' = \mathbf{S} \cdot \mathbf{P}$$



Scaling about a fixed point (x_f, y_f)

$$x' = x \cdot s_x + x_f (1 - s_x)$$

$$y' = y \cdot s_y + y_f (1 - s_y)$$

$$\mathbf{P}' = \mathbf{P} \cdot \mathbf{S} + \mathbf{P}_f \cdot (\mathbf{1\text{-}S})$$

Homogeneous Coordinates

Rotate and then displace a point $\mathbf{P}$: $\mathbf{P}' = \mathbf{M}_1 \cdot \mathbf{P} + \mathbf{M}_2$

$\mathbf{M}_1$: 2×2 rotation matrix. $\mathbf{M}_2$: 2×1 displacement vector.

Displacement is unfortunately a non linear operation.

Make displacement linear with **Homoheneous Coordinates**.

$(x, y) \Rightarrow (x, y, 1)$. Transformations turn into 3×3 matrices.

Very big advantage. All transformations are concatenated by matrix multiplication.

2D Translation

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}, \quad \mathbf{P}' = \mathbf{T}\left(t_x, t_y\right) \cdot \mathbf{P}$$

2D Rotation

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}, \quad \mathbf{P}' = \mathbf{R}\left(\theta\right) \cdot \mathbf{P}$$

2D Scaling

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}, \quad \mathbf{P}' = \mathbf{S}\left(S_x, S_y\right) \cdot \mathbf{P}$$

Orthogonal Rotation Matrix

The ***orthogonal rotation matrix*** is used to map motion in the global reference **$\{X_I, Y_I\}$** frame to motion in the robot's local reference frame **$\{X_R, Y_R\}$**

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

The orthogonal rotation matrix is used to convert robot velocity in the global reference frame to components of motion along the robot's local axes **$\{X_R, Y_R\}$**

$$\dot{\xi}_R = R(\theta)\dot{\xi}_I = R(\theta)\cdot\begin{bmatrix} \dot{x} & \dot{y} & \dot{\theta} \end{bmatrix}^T$$

How to convert Degrees to Radians

- One degree is equal 0.01745329252 radians:

1° = π/180° = 0.005555556π = 0.01745329252 rad

- The angle α in radians is equal to the angle α in degrees times pi constant divided by 180 degrees:

α(radians) = α(degrees) × π / 180°

- or radians = degrees × π / 180°

Sabırla Dinlediğiniz İçin Teşekkürler

Robot Teknolojisine Giriş BLM4830

Öğr. Grv. Furkan ÇAKMAK

Ders Tanıtım Formu ve Konular

Hafta	Tarih	Konular
1	28.02.2023	Ders Tanıtımı, ROS ve Platform Tanıtımı, Robot Çeşitleri ve Robotik Konuları Başlangıcı
2	7.03.2023	ROS Platformu Tanıtımı - 1
3	14.03.2023	ROS Platformu Tanıtımı - 2
4	21.03.2023	ROS Platformu Tanıtımı - 3
5	28.03.2023	Sensörler - Çeşitleri ve Çalışma Sistematikleri ve Uygulamaları
6	4.04.2023	Kinematik - Genel Tanımlar - Diferansiyel Sürüşlü Robot İçin Hesaplama Örnekleri
7	11.04.2023	Odometri ve Lokalizasyon Kavramları
8	18.04.2023	Haritalama Yöntemleri ve Uygulamaları
9	25.04.2023	Ara Sınav
10	2.05.2023	(1. Ödev Teslimi) 1. Ödev Gösterimi
11	9.05.2023	Navigasyon ve Keşif Uygulamaları ve Robot Üzerinden Görüntü İşleme Teknikleri
12	16.05.2023	(2. Ödev Teslimi) 2. Ödev Gösterimi
13	23.05.2023	3B Haritalama Yöntemleri
14	30.05.2023	(3. Ödev Teslimi) 3. Ödev Gösterimi

Localization:

Determination of the pose (= position + orientation) of a mobile robot in a known environment in order to succesfully 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.)

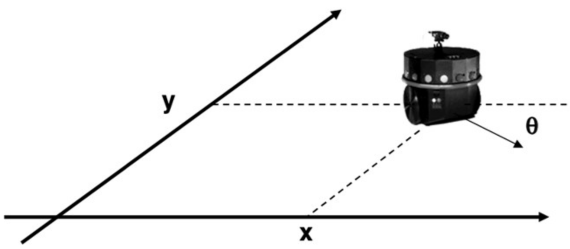
- Environment Measurements

  → Exteroceptive sensors (e.g. laser, sonar, IR, GPS, camera, RFID, etc.)

A filtering approach is required to fuse all information

Notation

$$x_r = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix}$$ Robot pose



$x_{r,0:t} = \{x_{r,0}, x_{r,1}, \ldots, x_{r,t}\}$ Robot poses from time 0 to time t

$z_{1:t} = \{z_1, z_2, \ldots, z_t\}$ Robot exteroceptive measurements from time 1 to time t

$u_{0:t} = \{u_0, u_1, \ldots, u_t\}$ Motion commands (or proprioceptive measurements) from time 0 to time t

<u>Belief of the robot at time t</u>: probability density function (pdf) describing the information the robot has regarding its pose at time t, based on all available data (exteroceptive measurements and motion commands):

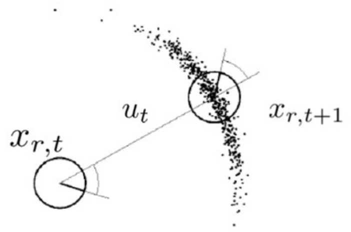
$$bel_t(x_r) = p(x_{r,t} = x_r | z_{1:t}, u_{0:t-1})$$

<u>Prior belief of the robot at time t</u>: pdf before acquiring the last measurement z_t:

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

The robot motion model is the pdf of the robot pose at time t+1 given the robot pose and the 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:

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

Robot Localization Algorithms

Genel olarak lokalizasyon algoritmaları Bayes Filtresine dayanıyor:

1. Adım Tahmin (t anında x state inde iken u kontrol işareti uygulandığında t+1 anında yeni state olasılığı)

Motion model

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

2. Adım Güncelleme (t anında state x iken bir z ölçümünü alma olasılığı)

Measurement model

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

Typical Motion Models

- In practice, one often finds two types of motion models:
  - **Odometry-based**
  - **Velocity-based** (**dead reckoning**)
- Odometry-based models are used when systems are equipped with wheel encoders.
- Velocity-based models have to be applied when no wheel encoders are given.
- They calculate the new pose based on the velocities and the time elapsed.

Dead Reckoning

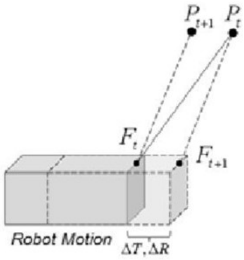
- Derived from "deduced reckoning."
- Mathematical procedure for determining the present location of a vehicle.
- Achieved by calculating the current pose of the vehicle based on its velocities and the time elapsed.

- Odometry tends to be more accurate than velocity model,
- But, Odometry is only available after executing a motion command, cannot be used for motion planning

Wheel Odometry

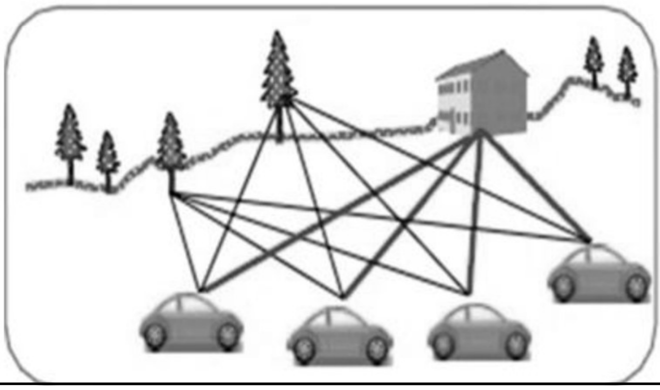
- **Odometry** is the use of data from motion sensors to estimate change in position over time. → Calculate the resulting robot position and orientation from wheel encoder measurements (Use the forward kinematics equations)

Visual Odometry

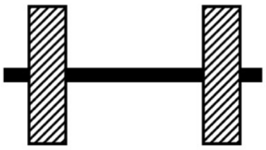


The visual odometry concept. The vehicle ego-motion is estimated from the apparent motion of the features in the image space

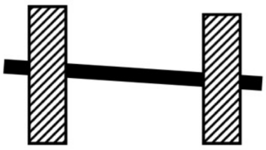
** LSM laser scan matching kamera yerine laser verilerini ilişkilendirme



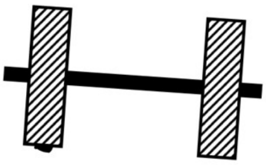
Possible sources of noise



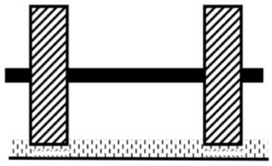
Ideal case



Different wheel
diameters (systematic source of noise)



Bump (casual source of noise)



Carpet (casual source of noise)

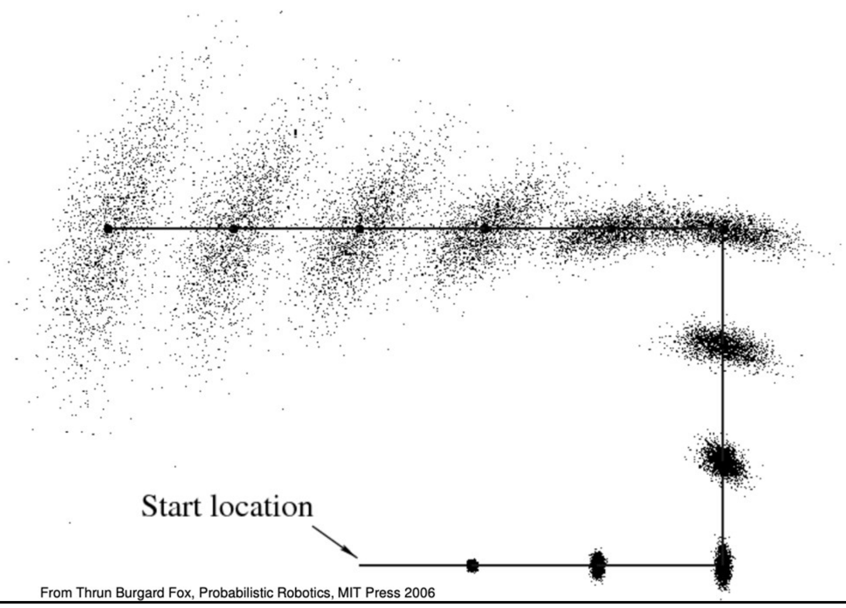
and many more (e.g. distance between the wheels) …

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.

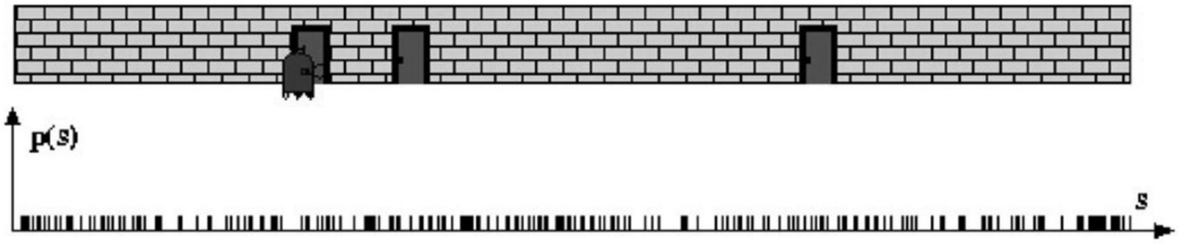
Errors are integrated, unbounded !!

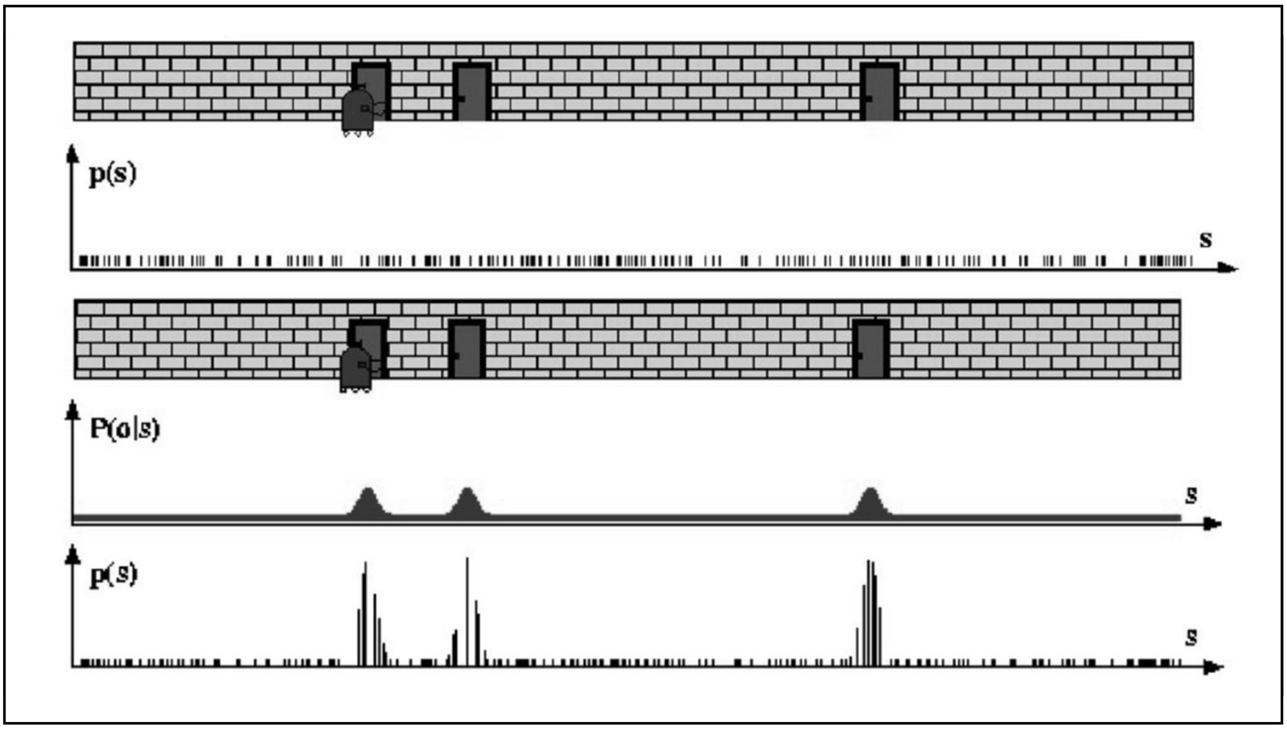
Accumulation of the pose estimation error under the robot motion
(only proprioceptive measurements)



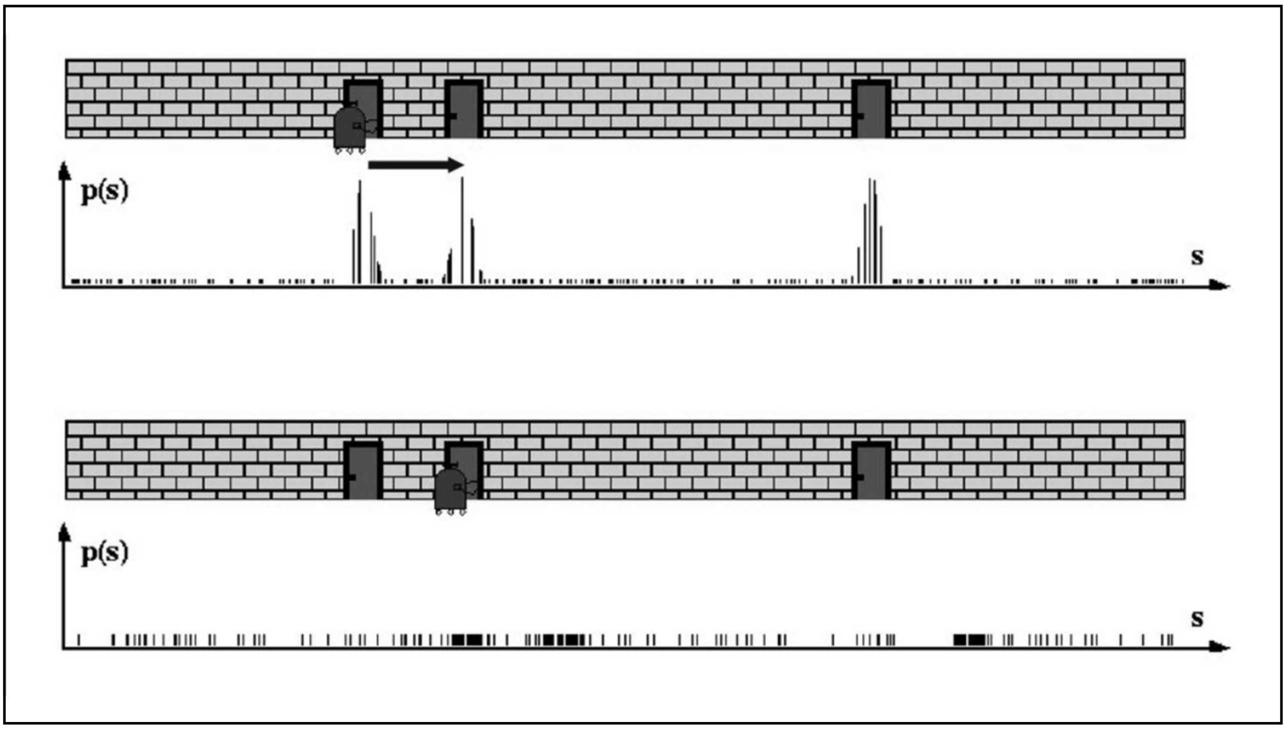
From Thrun Burgard Fox, Probabilistic Robotics, MIT Press 2006

Particle Filters

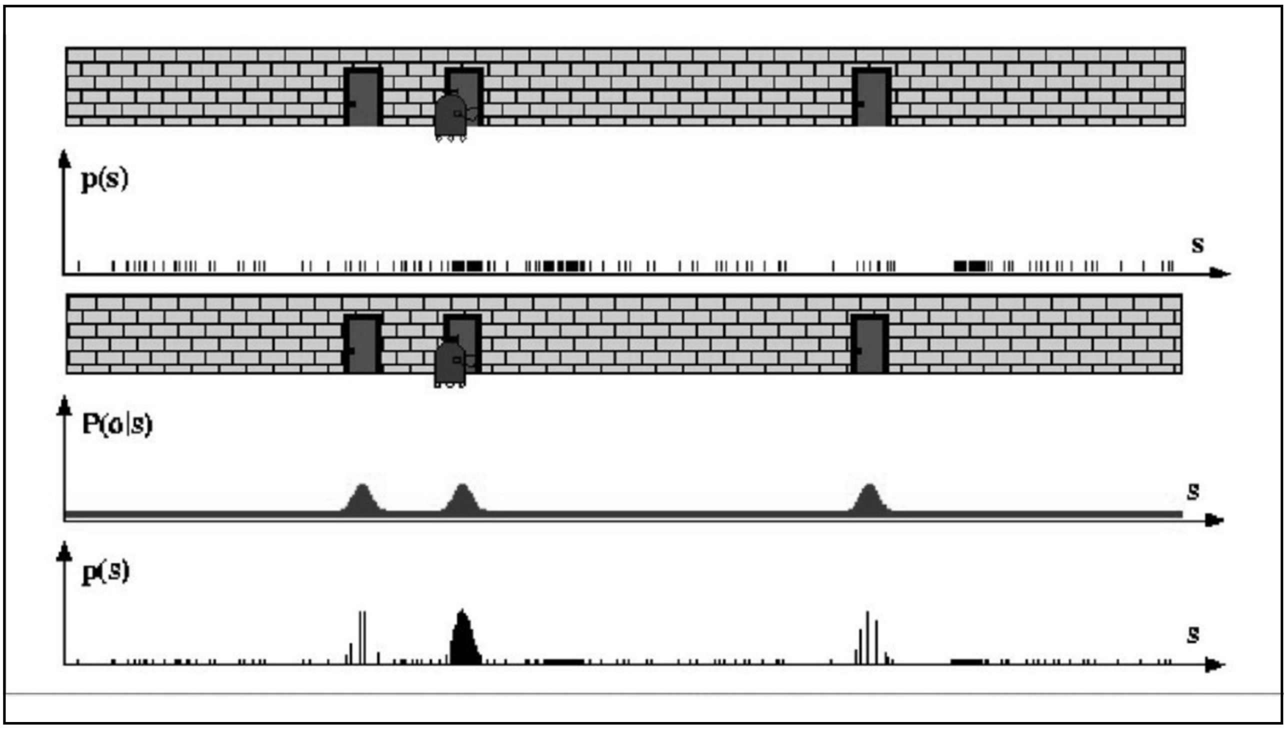


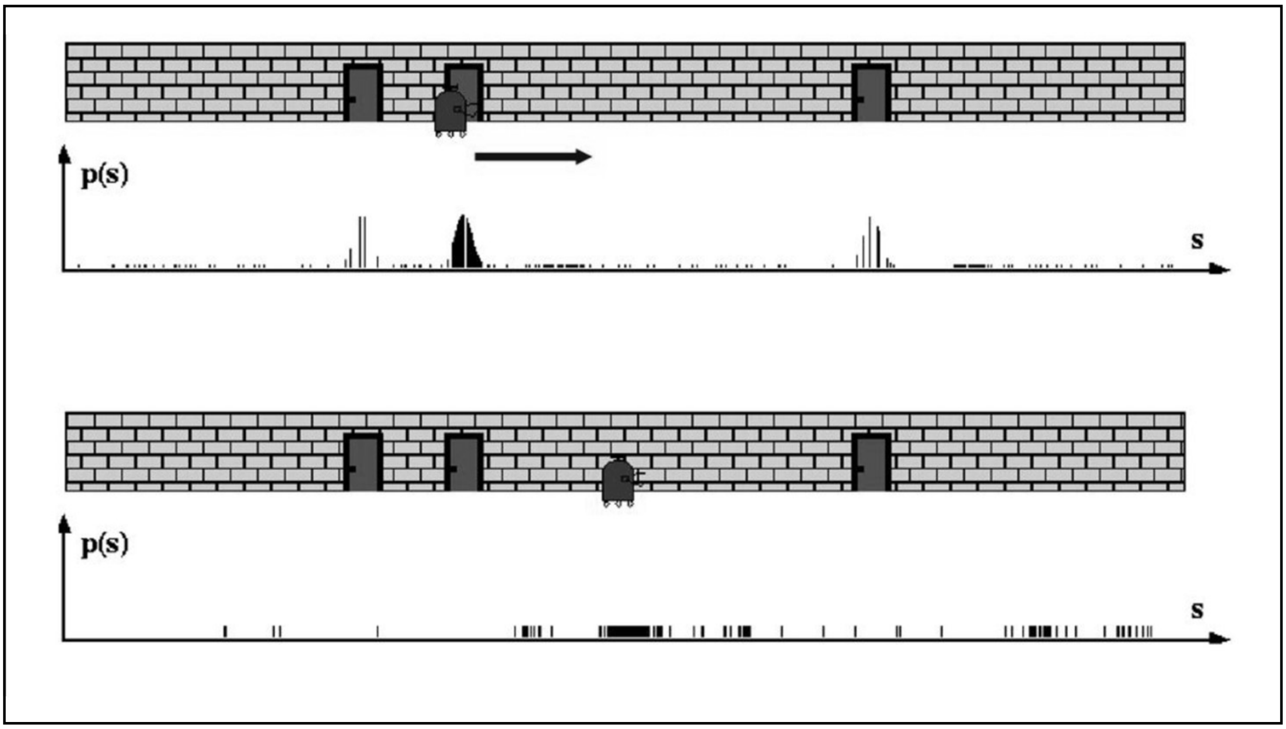


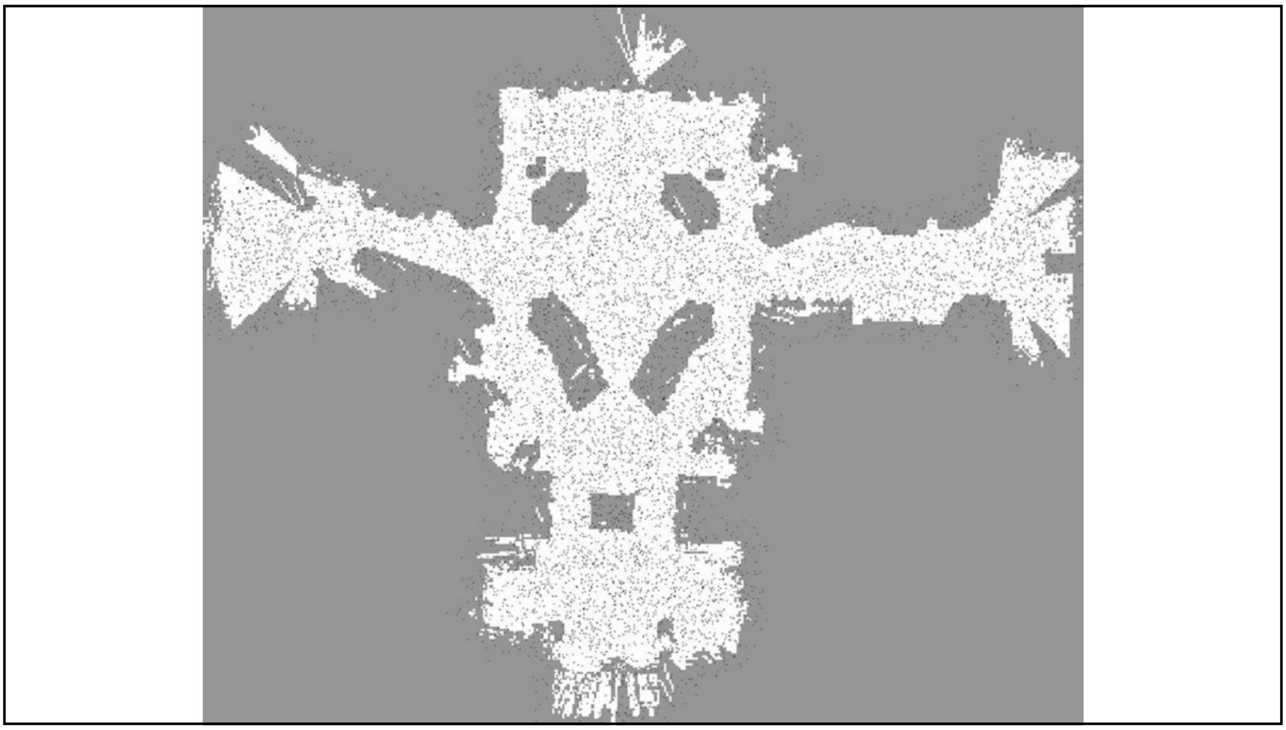
p(s)
s
P(o|s)
s
p(s)
s

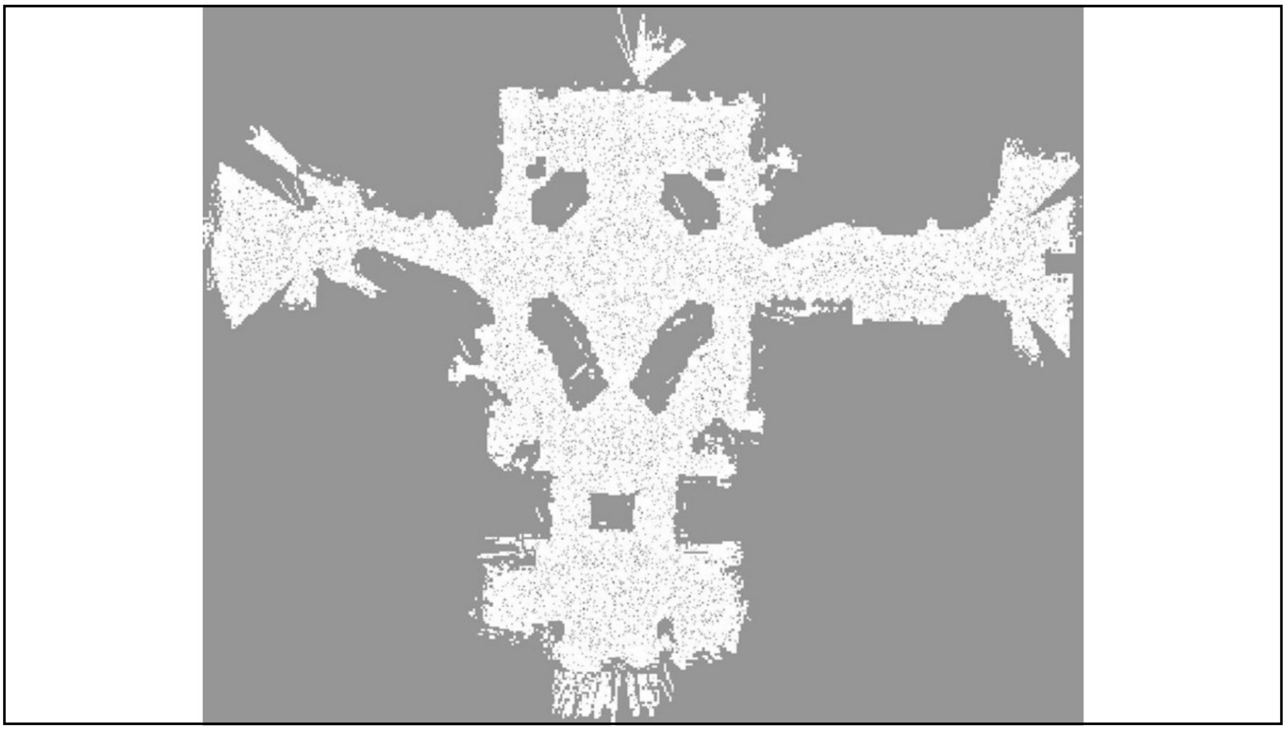


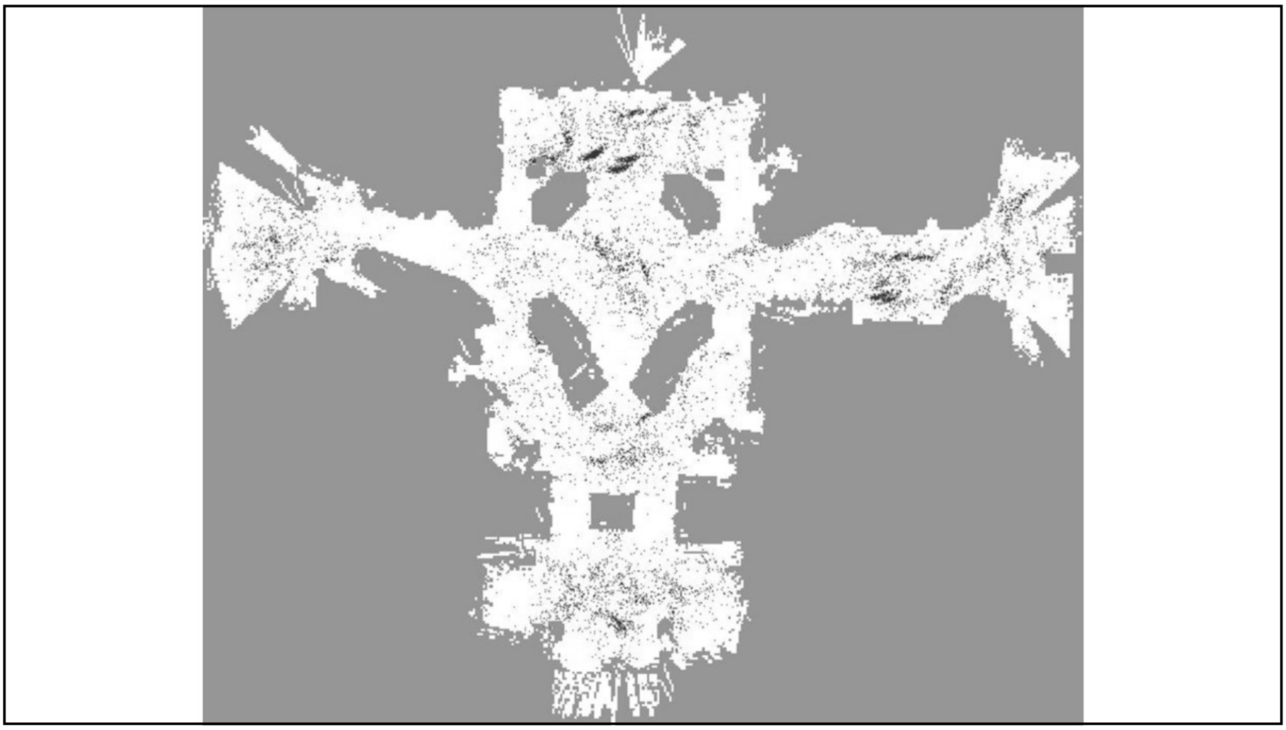
p(s)
s
p(s)
s

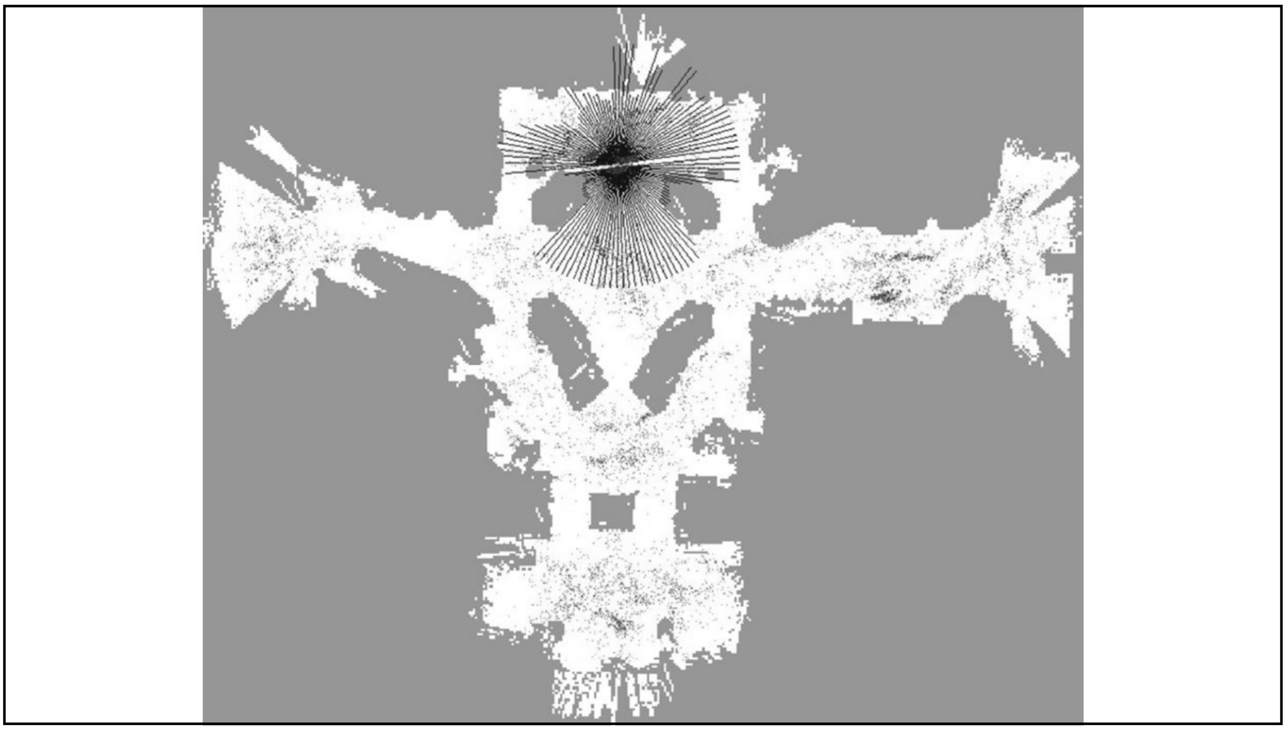

p(s)
s
P(o|s)
s
p(s)
s

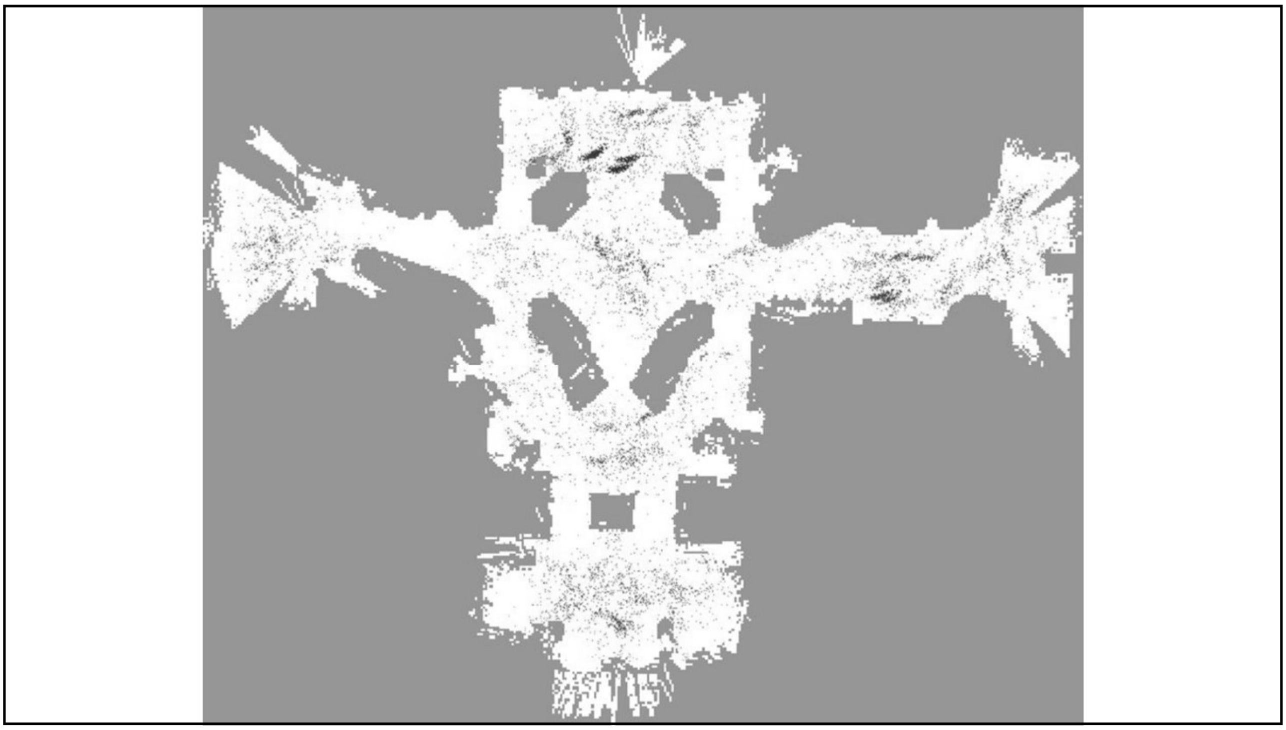

p(s)
s
p(s)
s

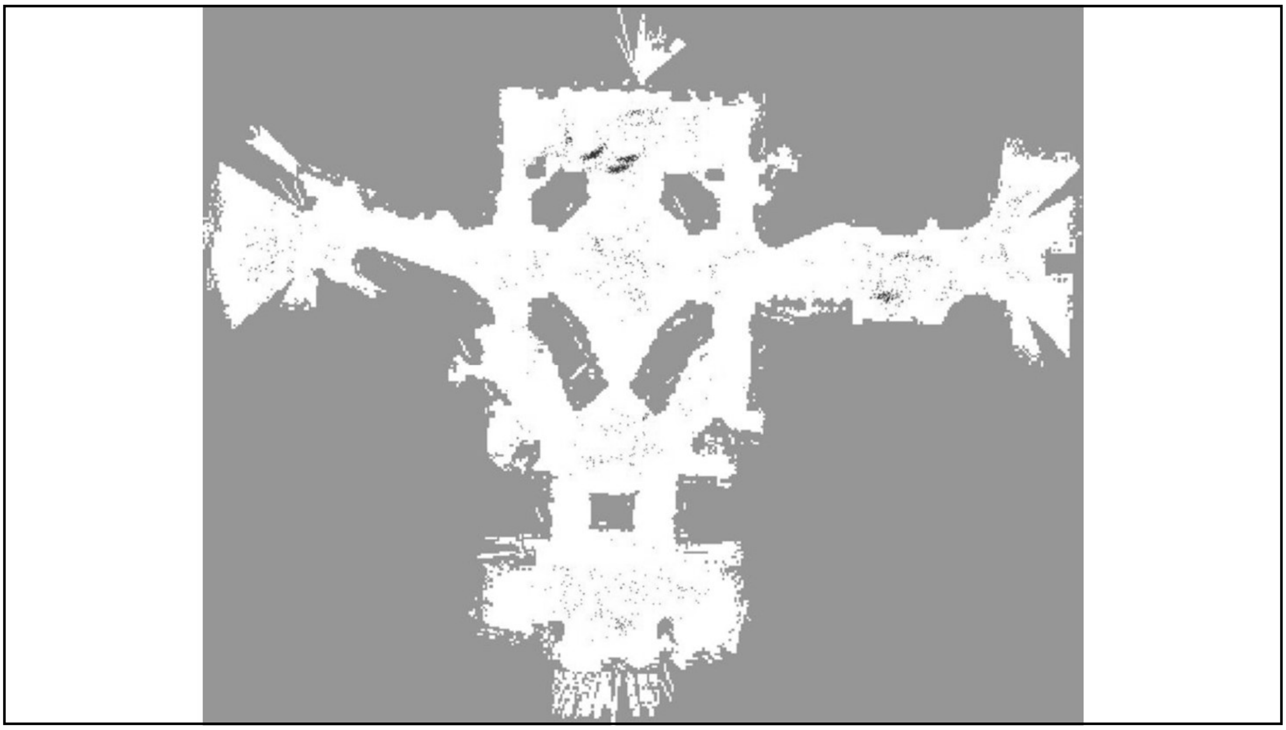


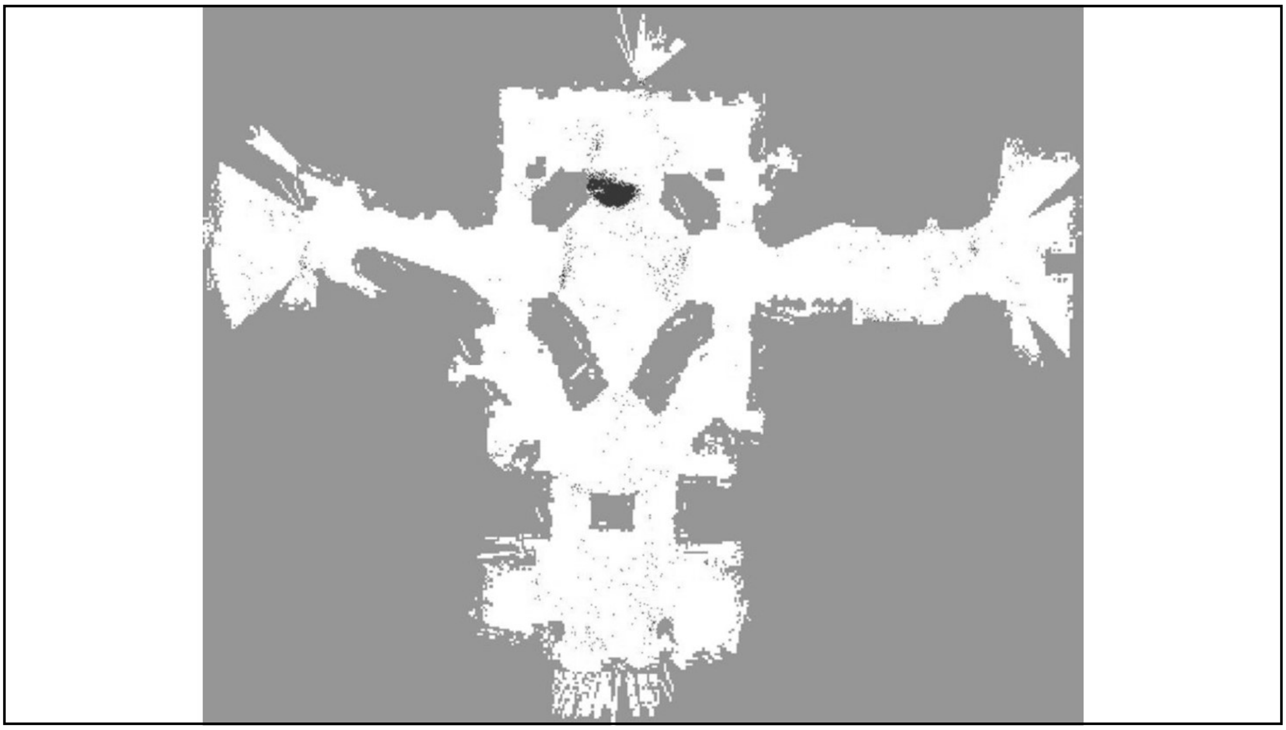


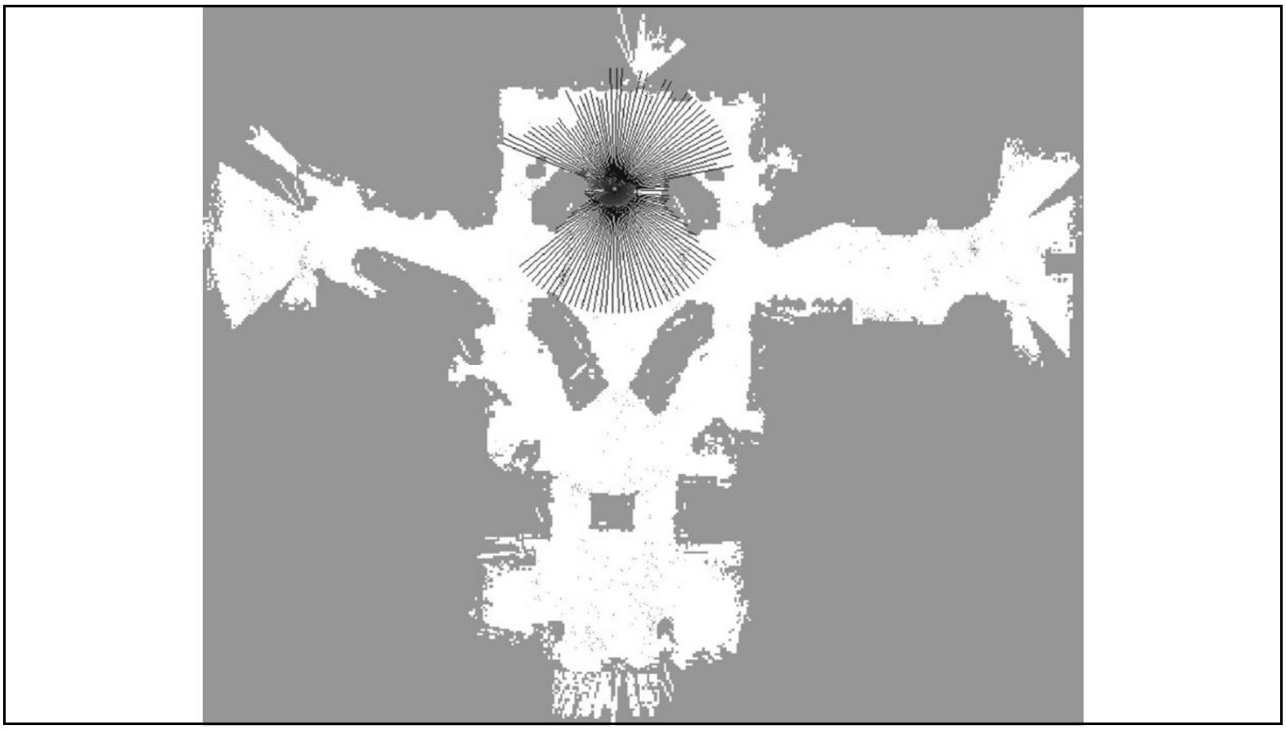


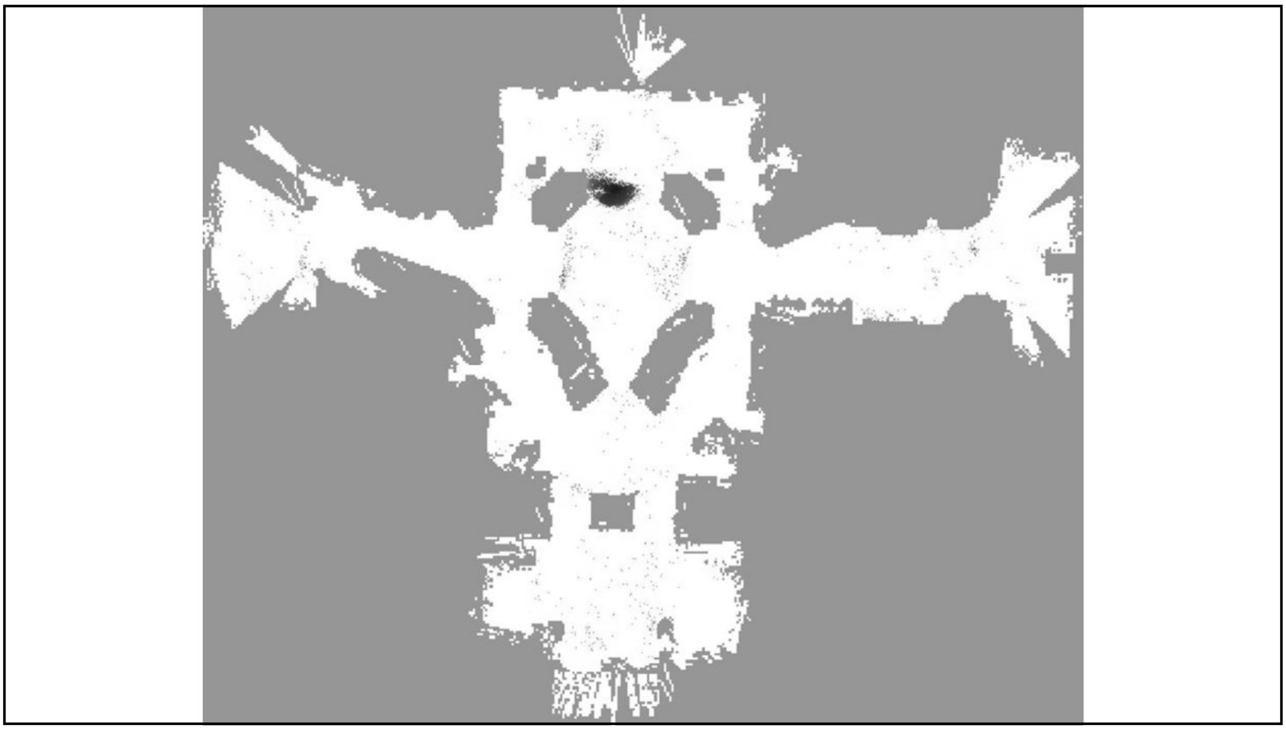


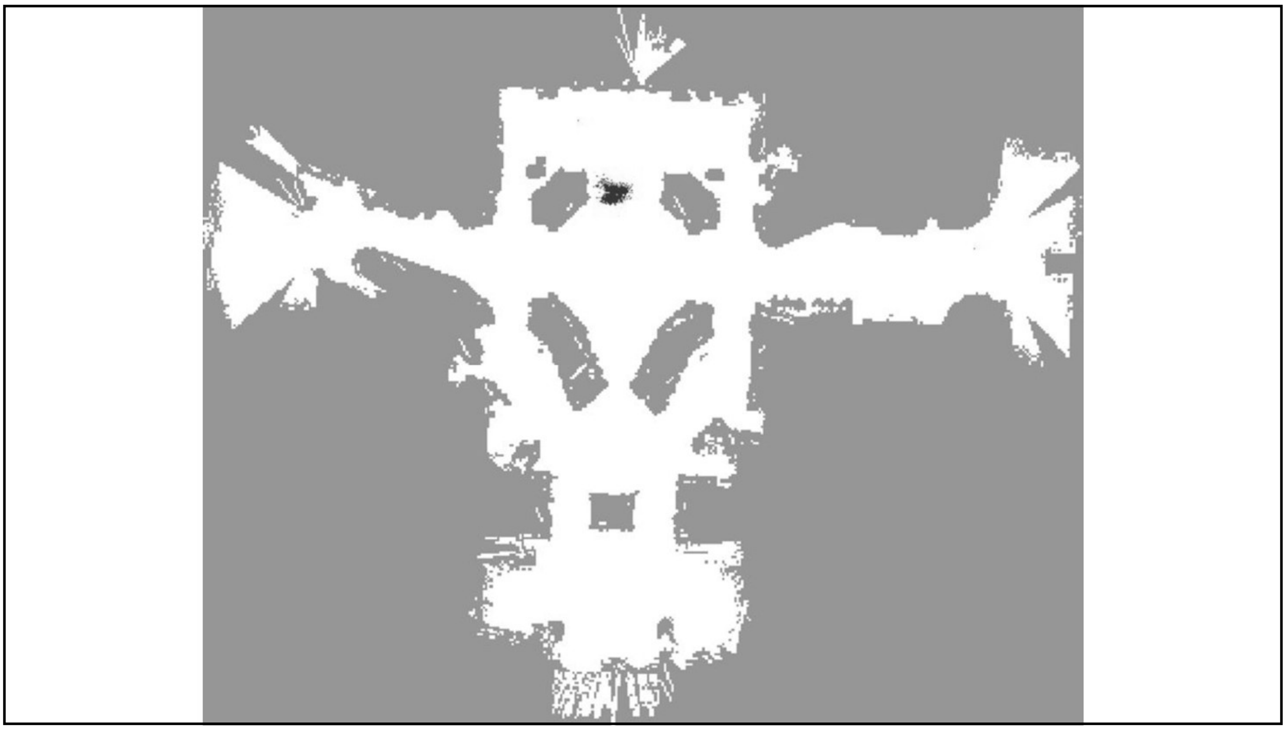


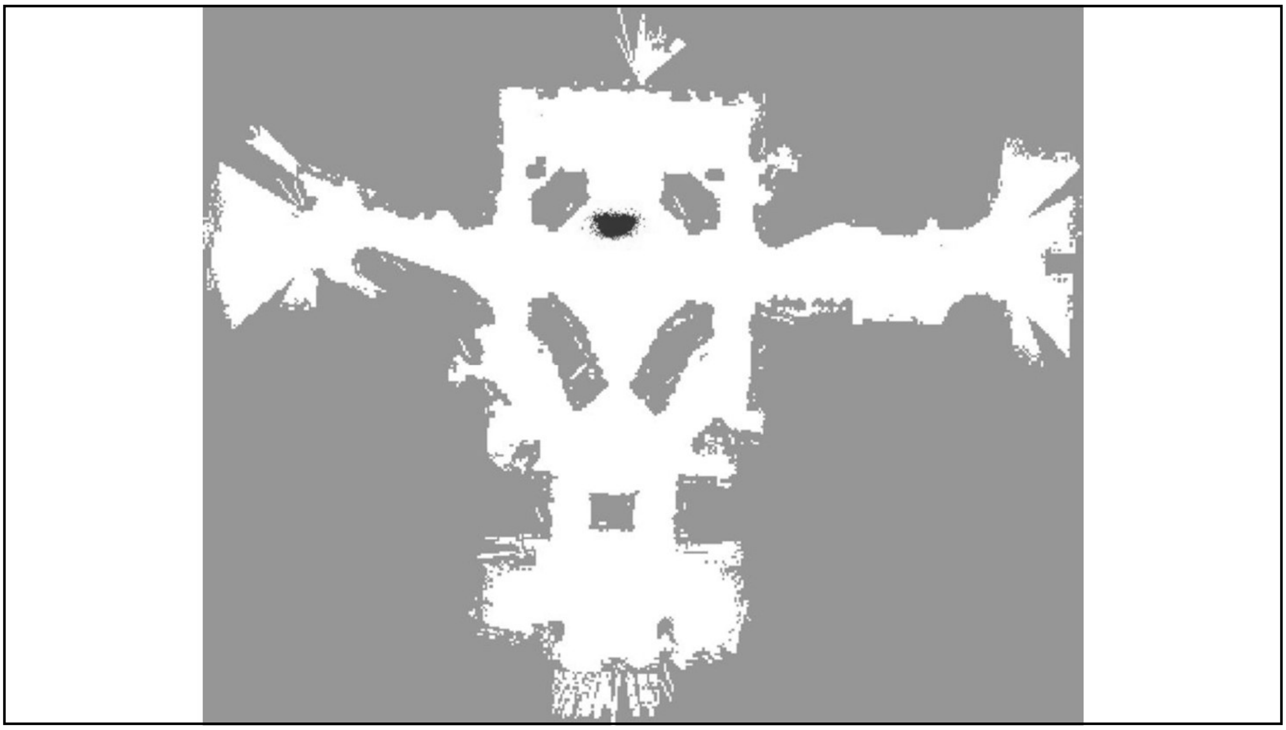


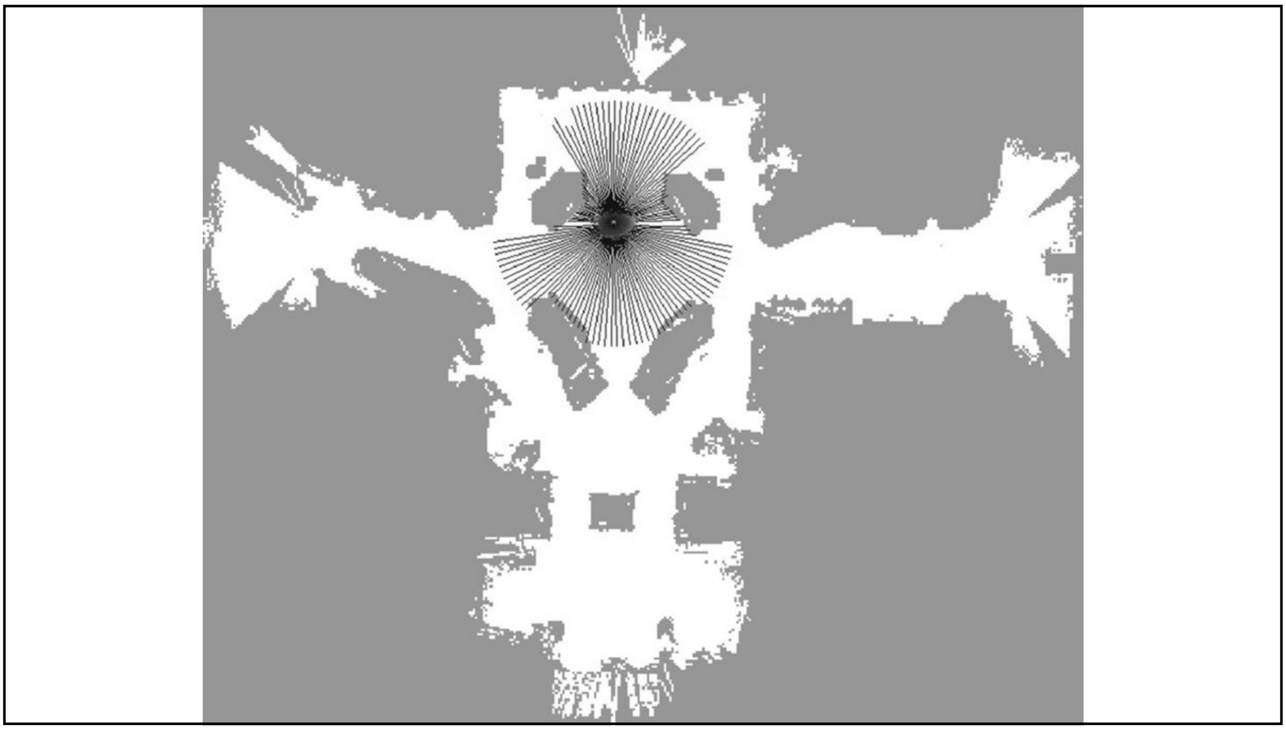


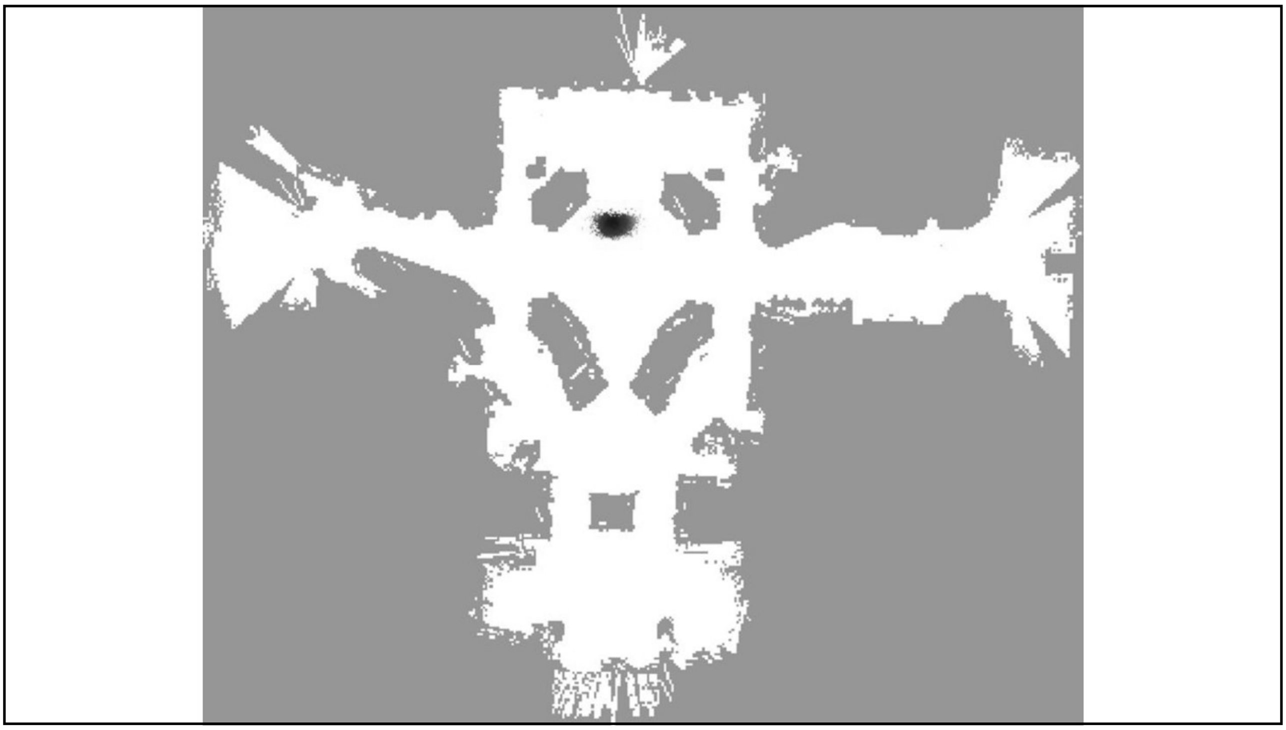


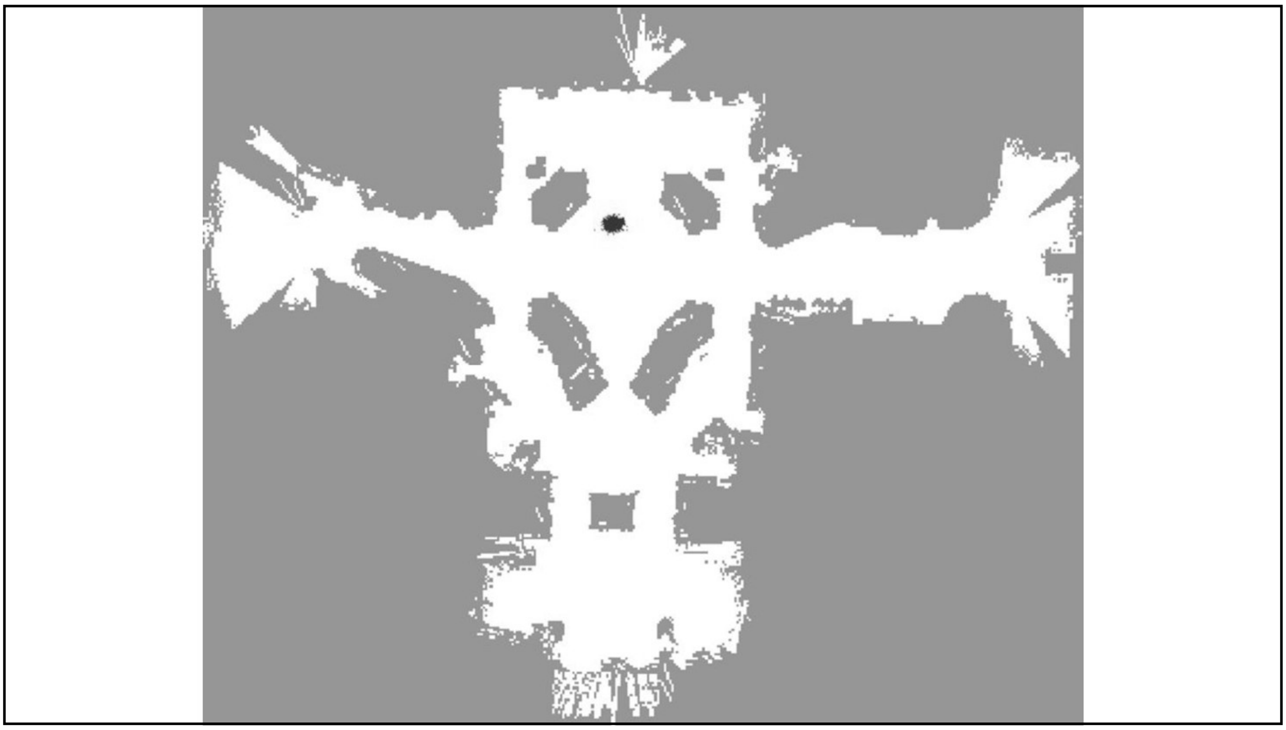


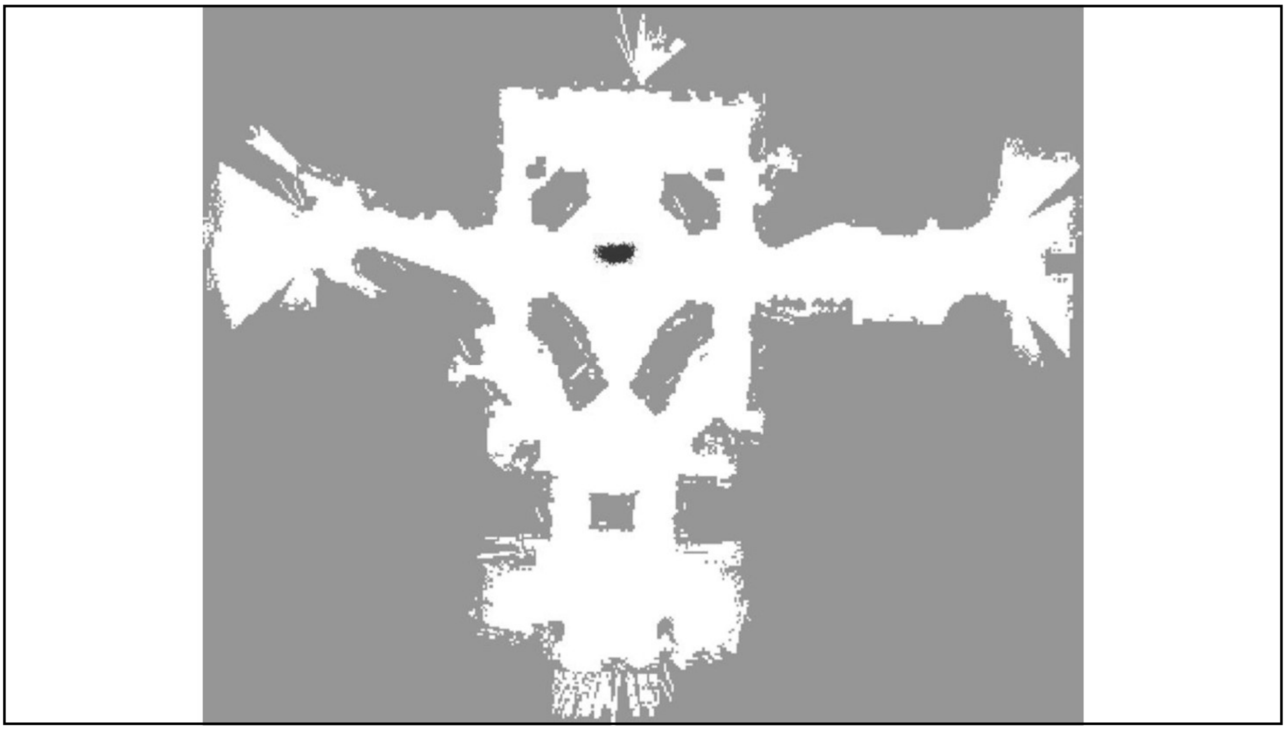












the ROS robot_pose_ekf Package

- *The Robot Pose EKF package is used to estimate the 3D pose of a robot, based on (partial) pose measurements coming from different sources.*
- *It uses an extended Kalman filter with a 6D model (3D position and 3D orientation) to combine measurements from wheel odometry, IMU sensor and visual odometry.*
- *The basic idea is to offer loosely coupled integration with different sensors, where sensor signals are received as ROS messages*

Robot_Pose_EKF (RPE)

Kombine bir odom üretir

- LSM odometrisi
- IMU odometrisi
- Görsel odometri
- Tekerlek odometrisi
- GPS

- Robotun pozu en son t_0 anında güncellendiyse;
- Abone olunan her bir sensörden t_0 anından sonra en az 1 ölçüm gelene kadar beklenir.
- t_0'dan sonra gelen bir t_1 anında odometri bilgisi alınırsa ve bundan sonra t_2 anında IMU bilgisi ulaşırsa (yani her bir sensörden bilgi almak mümkün olduğunda) kombine odom bilgisi güncellenebilmektedir.
- Her bir mesajın t_2 anında içerdiği bilgi farklı time_stamp barındırıyorsa, lineer interpolasyon ile dönüşüm yapılır.

Laser_Scan_Matcher (LSM)

- /scan mesajına abone olur
- /pose2D mesajı üretir
- Iterative Closest Point – ICP temelli PLICP kullanır.
- /odom üretilmektedir

Algoritma PLICP

Girdi: y_{t-1}, y_t, q_0

$S^{ref} \leftarrow y_{t-1}$ den oluşturulan parçalı doğru yüzeyi,

$k \leftarrow 0$

repeat

$p_i^w \leftarrow p_i \oplus q_k$

$j_1^i, j_2^i \leftarrow p_i^w$'ye en yakın iki nokta $(j_1^i, j_2^i \in y_{t-1})$

$C_k \leftarrow$ tüm $\langle i, j_1^i, j_2^i \rangle$ üçlüleri

C_k'dan aykırı değerleri temizle

$J(q_{k+1}, C_k) \leftarrow \sum_i \left(n_i^T \left[R(\theta_{k+1}) p_i + t_{k+1} - p_{j_1^i} \right] \right)^2$

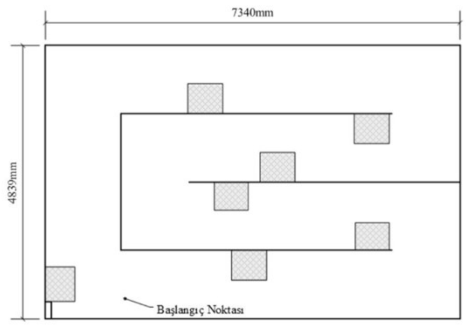
J'yi en küçükleyen q_{k+1} değerini bul

$k \leftarrow k$+1

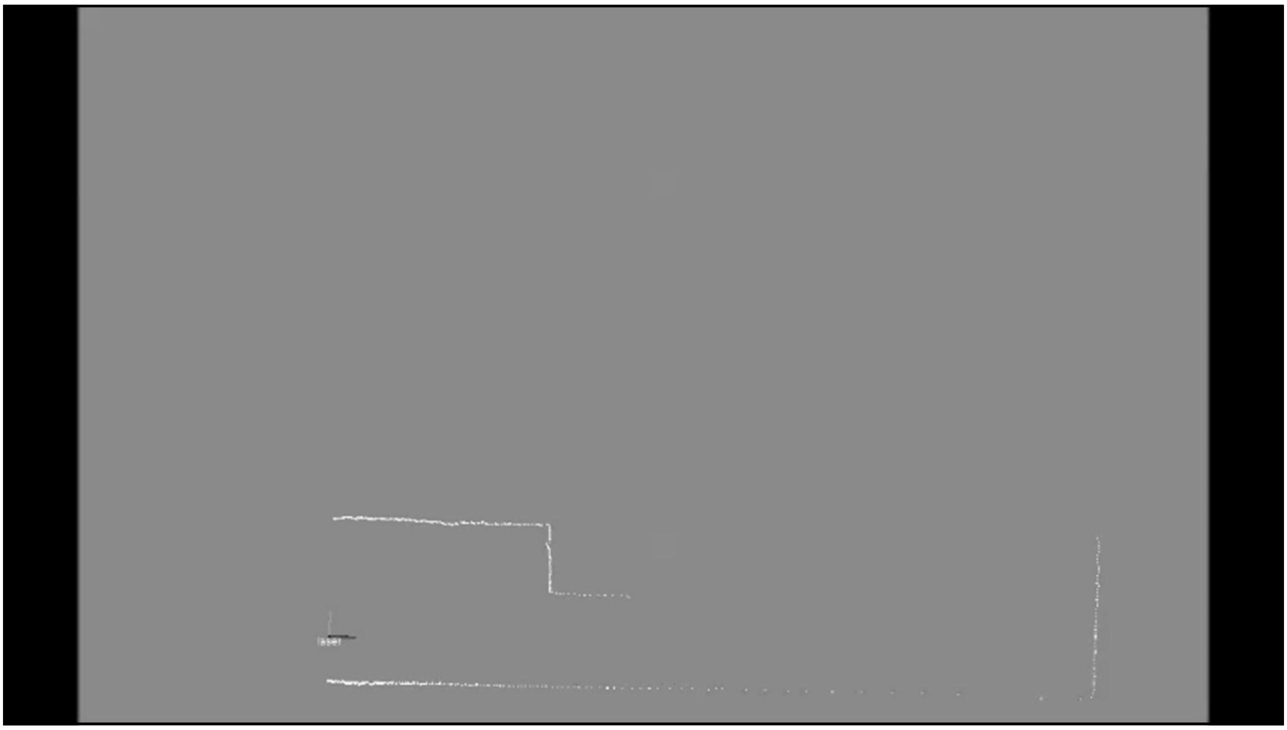
until (max _iterasyon_sayısı) or (yakınsama)

Çıktı: q

LSM Deneyleri – Deney Ortamı




laser


laser

Sabırla Dinlediğiniz İçin Teşekkürler

Robot Teknolojisine Giriş
BLM4830

Öğr. Grv. Furkan ÇAKMAK

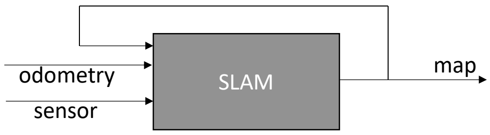
Ders Tanıtım Formu ve Konular

BLM4830
Robot
Teknolojisine
Giriş
Hafta 8

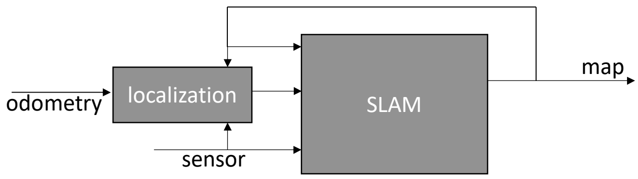
Hafta	Tarih	Konular
1	28.02.2023	Ders Tanıtımı, ROS ve Platform Tanıtımı, Robot Çeşitleri ve Robotik Konuları Başlangıcı
2	7.03.2023	ROS Platformu Tanıtımı - 1
3	14.03.2023	ROS Platformu Tanıtımı - 2
4	21.03.2023	ROS Platformu Tanıtımı - 3
5	28.03.2023	Sensörler - Çeşitleri ve Çalışma Sistematikleri ve Uygulamaları
6	4.04.2023	Kinematik - Genel Tanımlar - Diferansiyel Sürüşlü Robot İçin Hesaplama Örnekleri
7	11.04.2023	Odometri ve Lokalizasyon Kavramları
8	18.04.2023	Haritalama Yöntemleri ve Uygulamaları
9	25.04.2023	Ara Sınav
10	2.05.2023	(1. Ödev Teslimi) 1. Ödev Gösterimi
11	9.05.2023	Navigasyon ve Keşif Uygulamaları ve Robot Üzerinden Görüntü İşleme Teknikleri
12	16.05.2023	(2. Ödev Teslimi) 2. Ödev Gösterimi
13	23.05.2023	3B Haritalama Yöntemleri
14	30.05.2023	(3. Ödev Teslimi) 3. Ödev Gösterimi

SLAM

- Simultaneous localization and mapping
- If odometry is perfect



- Since odometry is not perfect



Online – Full SLAM

- Online SLAM

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

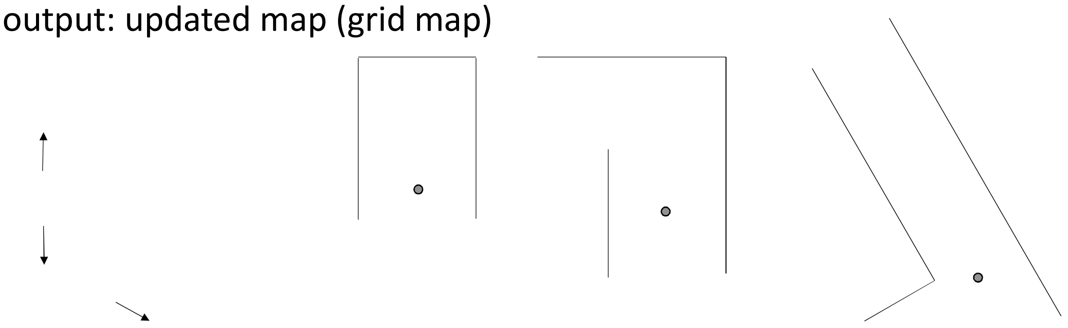
- Full SLAM

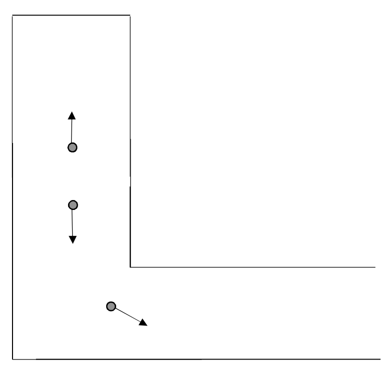
$$p(x_{1:t}, m|z_{1:t}, u_{1:t})$$

$$p(x_t, m|z_{1:t}, u_{1:t}) = \int \dots \int p(x_{1:t}, m|z_{1:t}, u_{1:t})\, d_{x_1} \dots d_{x_{t-1}}$$

Mapping with Perfect Odometry

- input: odometry (position, orinetation), sensor data (polar laser range), previous map (grid map)
- output: updated map (grid map)





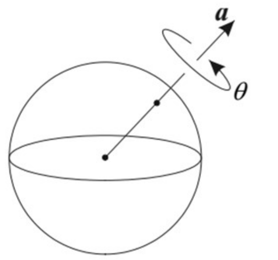
Quaternion – Orientation format

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

(a is an arbitrary vector of rotation, nx, ny, nz components of normal vector a)

- For mobile robot travelling on XY-plane

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



SLAM Problem Statement

- Inputs:
  - No external coordinate reference
  - Time sequence of proprioceptive and exteroceptive measurements* made as robot moves through an initially unknown environment
- Outputs:
  - A *map** of the environment
  - A robot *pose estimate* associated with each measurement, in the coordinate system in which the map is defined

*Not yet fully defined

SLAM Problem -- Incremental

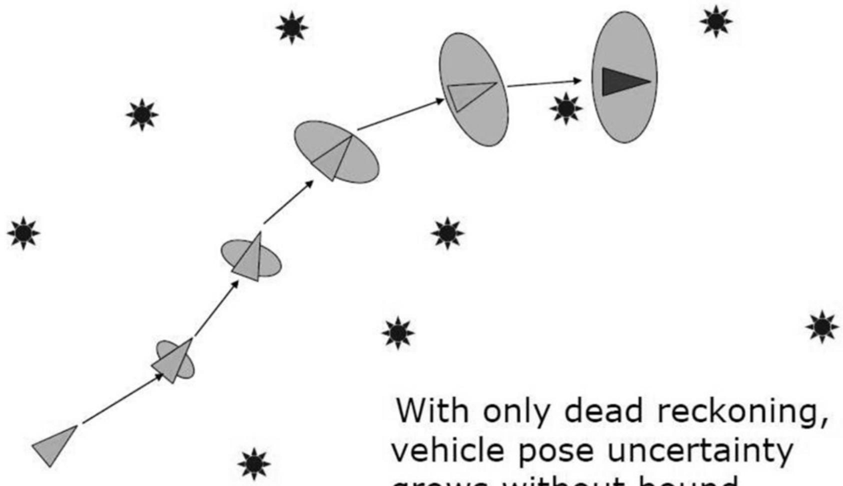
- State/Output:
  - Map of env’t observed “so far”
  - Robot pose estimate w.r.t. map
- Action/Input:
  - Move to a new position/orientation
  - Acquire additional observation(s)
- Update State:
  - Re-estimate the robot’s pose
  - Revise the map appropriately

SLAM Aspects

- What is a measurement?
- What is a map?
- How are map, pose coupled?
- How should robot move?
- What is hard about SLAM?

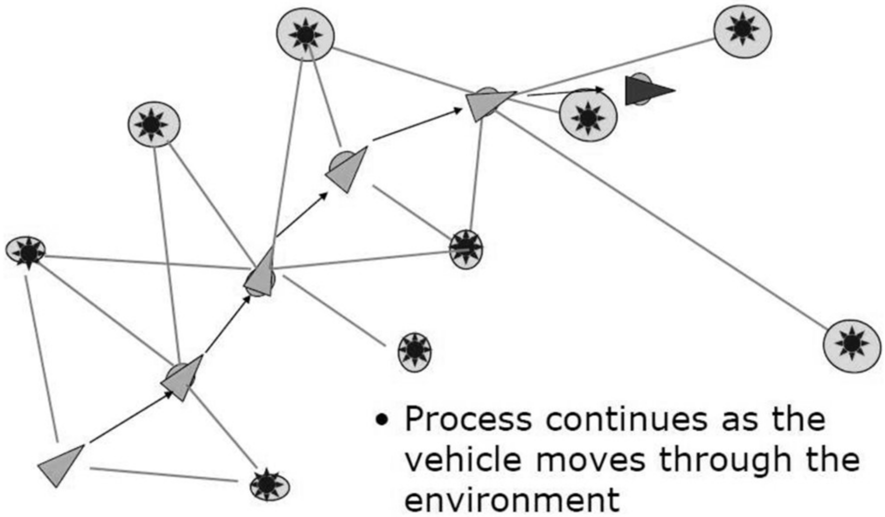
- But first: some intuition

Illustration of SLAM without Landmarks



With only dead reckoning, vehicle pose uncertainty grows without bound

Illustration of SLAM with Landmarks



- Process continues as the vehicle moves through the environment

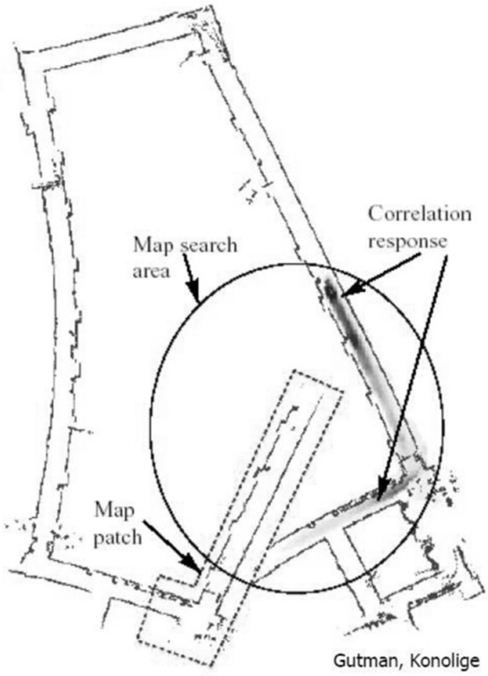
Why is SLAM Hard?

- "Grand challenge"-level robotics problem
  - Autonomous, persistent, collaborative robots mapping multi-scale, generic environments
- Map-making = learning
  - Difficult even for humans
  - Even skilled humans make mapping mistakes
- Scaling issues
  - Space: Large extent (combinatorial growth)
  - Time: Persistent autonomous operation
- "Chicken and Egg" nature of problem
  - If robot had a map, localization would be easier
  - If robot could localize, mapping would be easier
  - … But robot has neither; starts from blank slate
  - Must also execute an *exploration strategy*
- Uncertainty at every level of problem

Uncertainty in Robotic Mapping

Uncertainty: / Scale:	Continuous	Discrete
Local	Sensor noise	Data association
Global	Navigation drift	Loop closing

Loop Closing

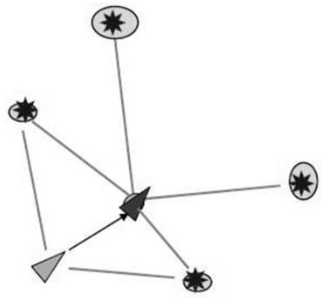


What is a map?

- Collection of *features* with some *relationship* to one another
- What is a *feature*? ← Uncertainty
  - Occupancy grid cell
  - Line segment
  - Surface patch
- What is a feature *relationship*?
  - Rigid-body transform (metrical mapping)
  - Topological path (chain of co-visibility)
  - Hybrid representation (geom. + topo.)

What is *pose* w.r.t. a map?

- Pose estimate that is (maximally) *consistent* with the estimated features observed from vicinity
- Consistency can be evaluated locally, semi-locally, or globally

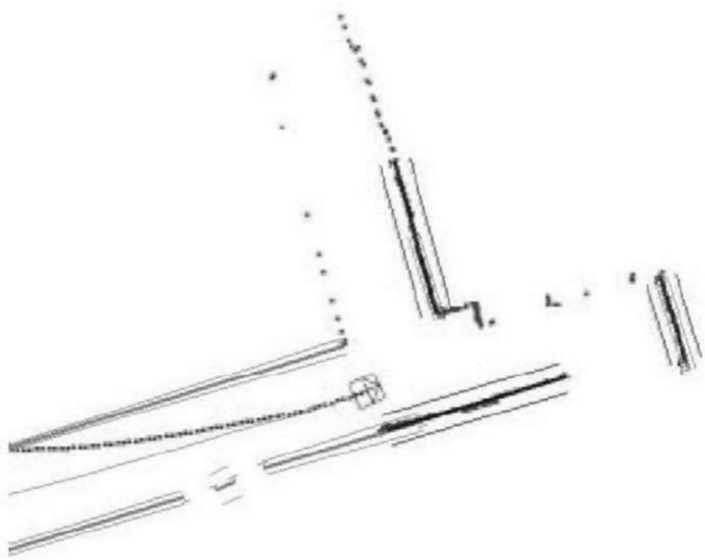


- Note tension between estimation *precision* and solution *consistency*

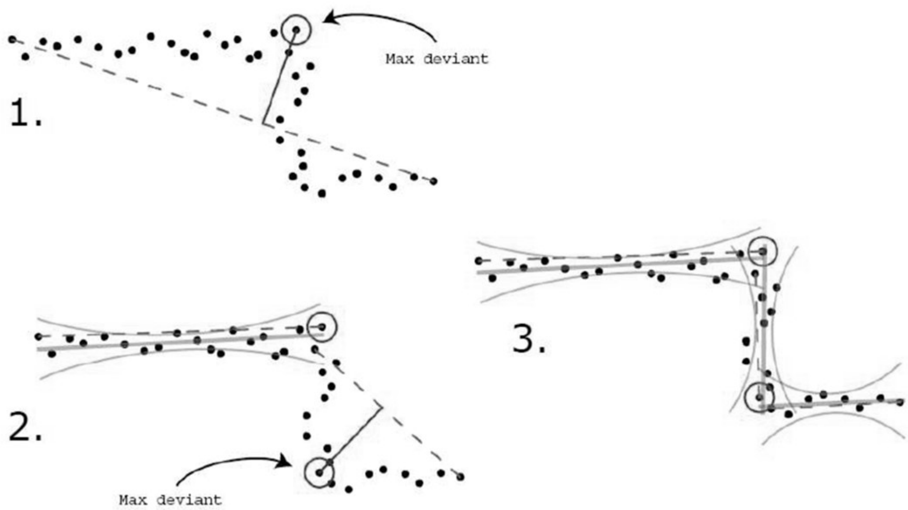
Example

- SLAM with laser scanning
- Observations
- Local mapping
  - Iterated closest point
- Loop closing
  - Scan matching
  - Deferred validation
  - Search strategies

Observations

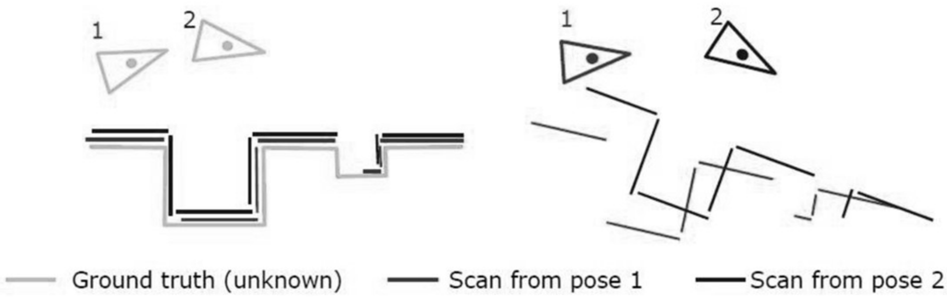


Observations



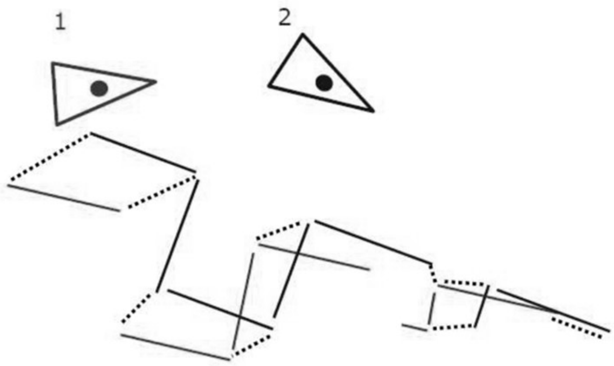
Scan Matching

- Robot scans, moves, scans again
- Short-term odometry/IMU error causes misregistration of scans
- *Scan matching* is the process of bringing scan data into alignment



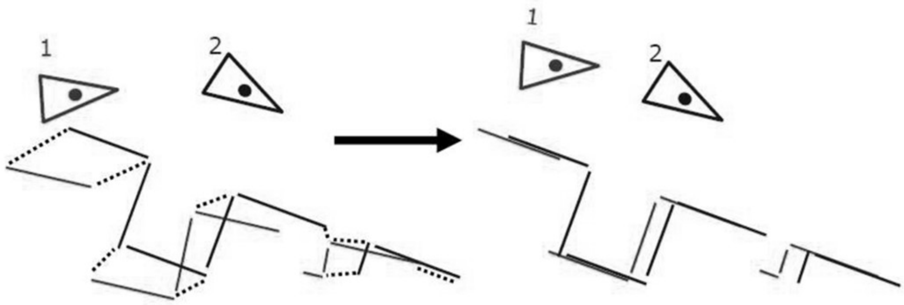
Iterated Closest Point

- For each point in scan 1
  - Find closest point in scan 2



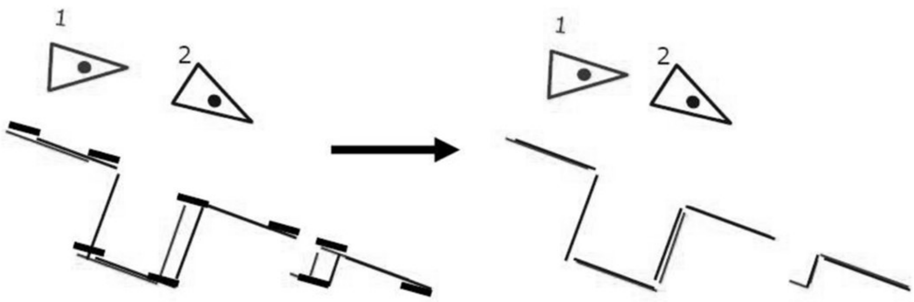
Iterated Closest Point

- Find the transformation that best aligns the matching sets of points



Iterated Closest Point

- … Repeat until convergence



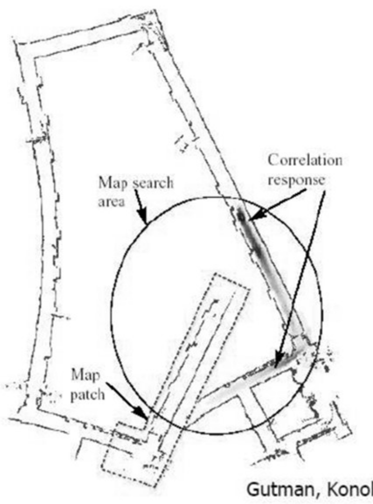
- Can ICP across scans, across a scan and a (sub)map, or even across submaps!

Limitations / failure modes

- Computational cost (two scans of size *n*)
  - Naively, $O(n^2)$ plus cost of alignment step
- False minima
  - If ICP starts far from true alignment
  - If scans exhibit repeated local structure
- Bias
  - Anisotropic point sampling
  - Differing sensor fields of view (occlusion)
- Lots of research on improved ICP methods (see, e.g., Rusinkiewicz)

Loop Closing

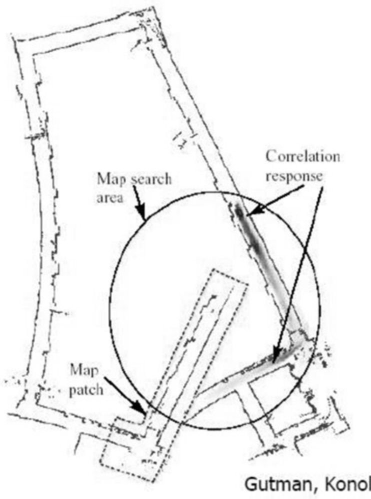
- ICP solves small-scale, short-duration SLAM fairly well
- But now, consider:
  - Large scale
  - High uncertainty



Gutman, Konolige

Loop Closing

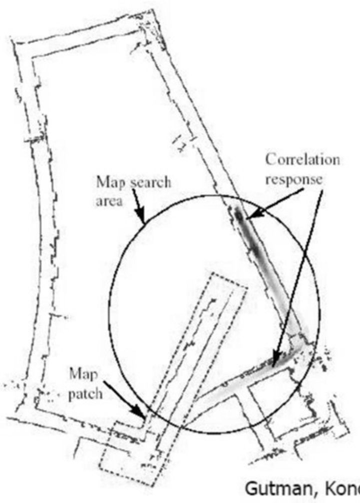
- Naive ICP ruled out:
  - Too CPU-intensive
- Assume we have a pose *uncertainty bound*
- This limits portion of existing map that must be searched
- Still have to face the problem of matching two partial scans that are far from aligned



Gutman, Konolige

Scan Matching Strategies

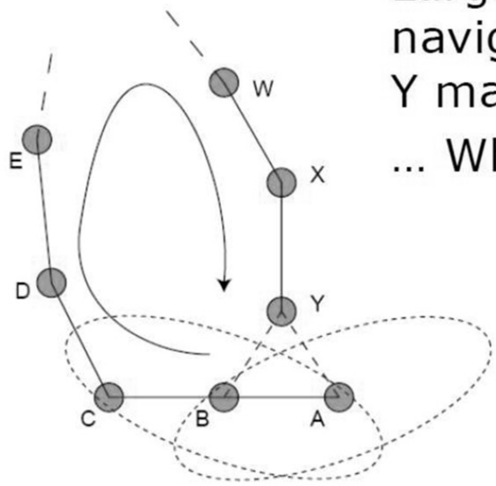
- Exhaustive search
  - Discretize robot poses
  - Find implied alignments
  - Assign score to each
  - Choose highest score
  - Pros, Cons?
- Randomized search
  - Choose minimal suff-icient match, at random
  - Align and score
  - Choose highest score
  - RANSAC (1981)
  - Pros, Cons?



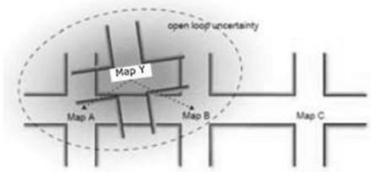
Gutman, Konolige

Loop Closing Ambiguity

- Consider SLAM state after ABC … XY

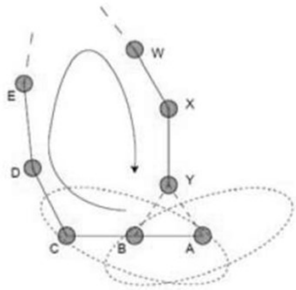


Large open-loop navigation uncertainty
Y matches *both* A & B
… What to do?



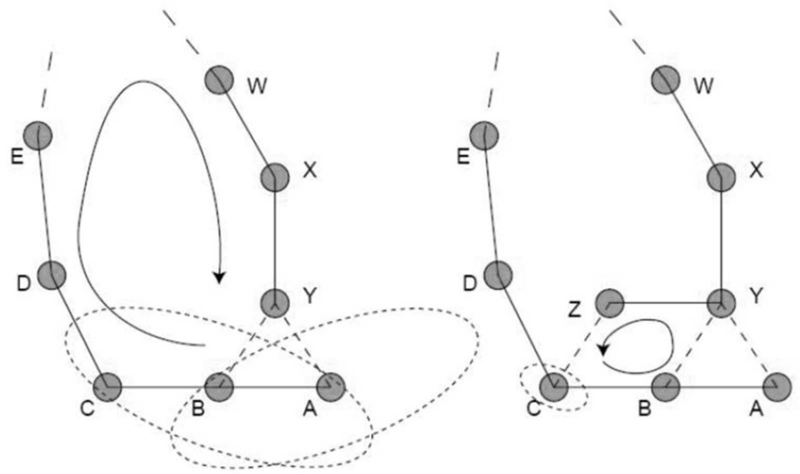
Loop Closing Choices

- Choose neither match
  - Pros, cons?
- Choose one match
  - Pros, cons?
- Choose both matches
  - Pros, cons?



Deferred Loop Validation

- Continue SLAM until Z matches C
- Examine graph for ~*identity cycle*



Occupancy Grid Map (OGM)

Maps the environment as a grid of cells

– Cell sizes typically range from 5 to 50 cm

Each cell holds a probability value that the cell is occupied in the range [0,100]

Unknown is indicated by -1

– Usually unknown areas are areas that the robot sensors cannot detect (beyond obstacles)

Occupancy Grid Map

White pixels represent free cells
Black pixels represent occupied cells
Gray pixels are in unknown state

Occupancy Grid Maps

Pros:

– Simple representation

– Speed

Cons:

– Not accurate - if an object falls inside a portion of a grid cell, the whole cell is marked occupied

– Wasted space

Maps in ROS

Map files are stored as images, with a variety of common formats being supported (such as PNG, JPG, and PGM)

Although color images can be used, they are converted to grayscale images before being interpreted by ROS

Associated with each map is a YAML file that holds additional information about the map

Map YAML File

```
image: map.pgm
resolution: 0.050000
origin: [-100.000000, -100.000000, 0.000000]
negate: 0
occupied_thresh: 0.65
free_thresh: 0.196
```

resolution: Resolution of the map, meters / pixel

origin: 2D pose of the lower-left pixel as (x, y, yaw)

occupied_thresh: Pixels with occupancy probability greater than this threshold are considered completely occupied

free_thresh: Pixels with occupancy probability less than this threshold are considered completely free

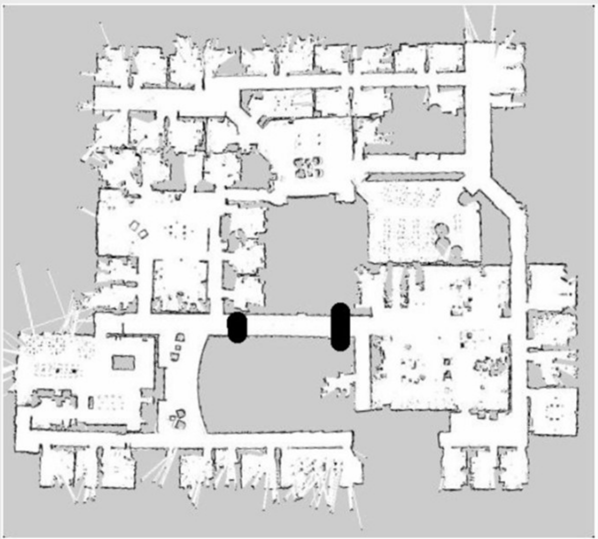
Editing Map Files

Since maps are represented as image files, you can edit them in your favorite image editor

This allows you to tidy up any maps that you create from sensor data, removing things that shouldn't be there, or adding in fake obstacles to influence path planning

For example, you can stop the robot from planning paths through certain areas of the map by drawing a line across a corridor you don't want to the robot to drive through

Editing Map Files



SLAM

Simultaneous localization and mapping (**SLAM**) is a technique used by robots to build up a map within an unknown environment while at the same time keeping track of their current location

A chicken or egg problem: An unbiased map is needed for localization while an accurate pose estimate is needed to build that map

Particle Filter – FastSLAM

Represent probability distribution as a set of discrete particles which occupy the state space

Main steps of the algorithm:

- Start with a random distribution of particles
- Compare particle's prediction of measurements with actual measurements
- Assign each particle a weight depending on how well its estimate of the state agrees with the measurements
- Randomly draw particles from previous distribution based on weights creating a new distribution

Efficient: scales logarithmically with the number of landmarks in the map

gmapping

http://wiki.ros.org/gmapping

The gmapping package provides laser-based SLAM as a ROS node called **slam_gmapping**

Uses the FastSLAM algorithm

It takes the laser scans and the odometry and builds a 2D occupancy grid map

It updates the map state while the robot moves

ROS with gmapping video

Sabırla Dinlediğiniz İçin Teşekkürler

BLM4830
Robot
Teknolojisine
Giriş
Hafta 8