

These slides are based on material I developed while working at the University of Bern. This version has been adapted for public educational use.

Who I am

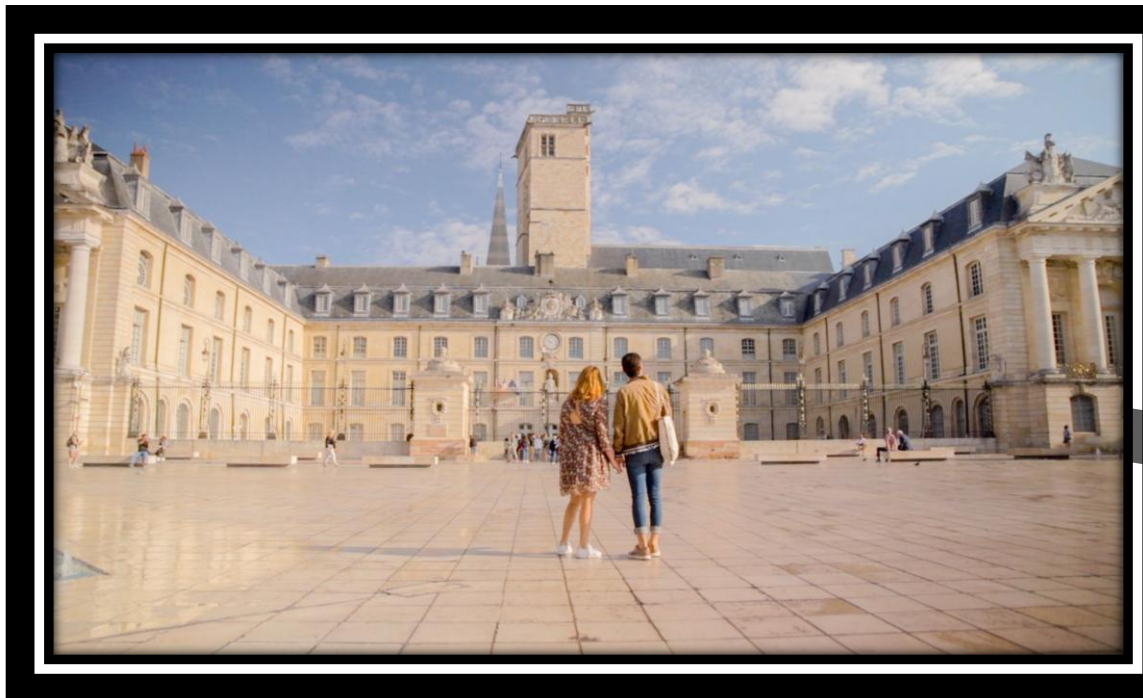


Loubet—Bonino Grégory

Research Focus

Robotic Monitoring and Self-Adaptability using AI and genetic Algorithm

Improving nonfunctional properties of robots to make the it more reliable



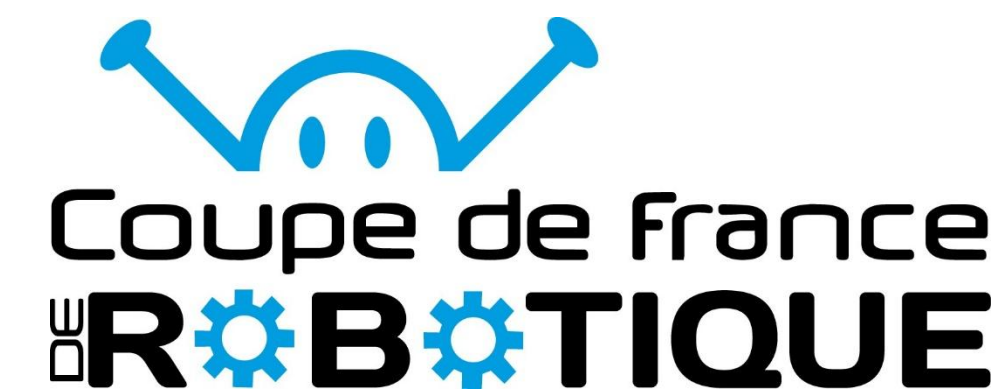
Preparatory Class, Dijon 2020-2022



Master of Embedded systems
Centrale-Lyon 2024-2025

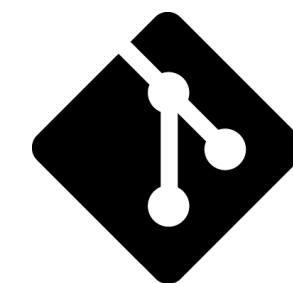


Electronics Engineering school
Telecom 2022-2025

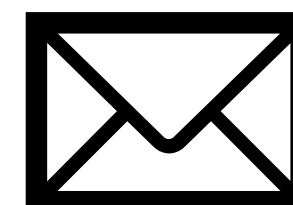


Compete in the French robotic
cup with Projet&Tech team

From 2022 to 2025



<https://github.com/dragonfuneste>



Gregory.loubet-bonino@unibe.ch

Objective

Understand
SLAM and its
Challenges



1

To master
simulation
tools



2

Be able to use
SLAM in Cyber-
Physical Systems
(CPS)



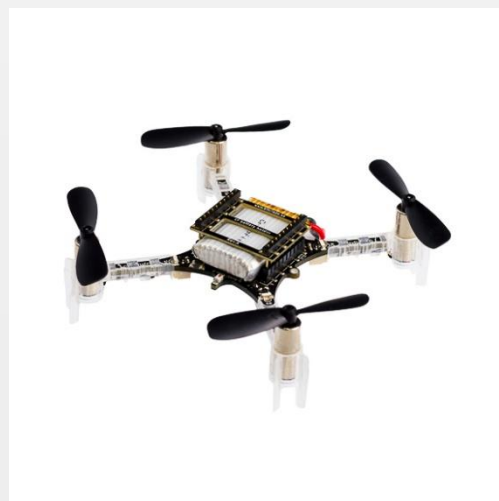
3

What is a robot ?

A robot is an embodied system that perceives its environment through sensors, makes decisions, and acts on the physical world using actuators, with a certain level of autonomy.

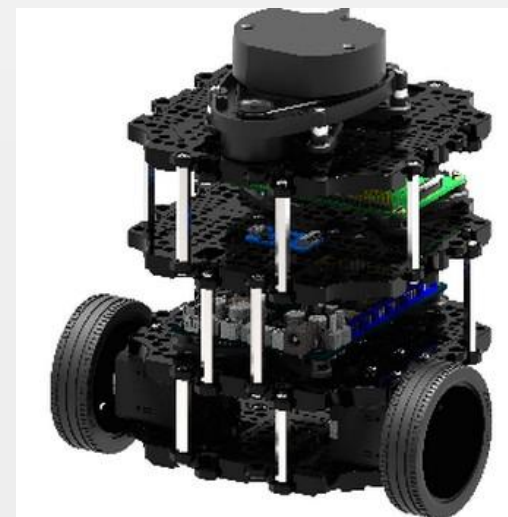
This definition includes :

UAVs
(Unmanned aerial vehicle)



Crazyflie

Ground Robot



Turtlebot3

Humanoid robots



Sparky

...

An old idea



Talos Myth in ancient Greek

It was a legend of a giant bronze man who protected Crete

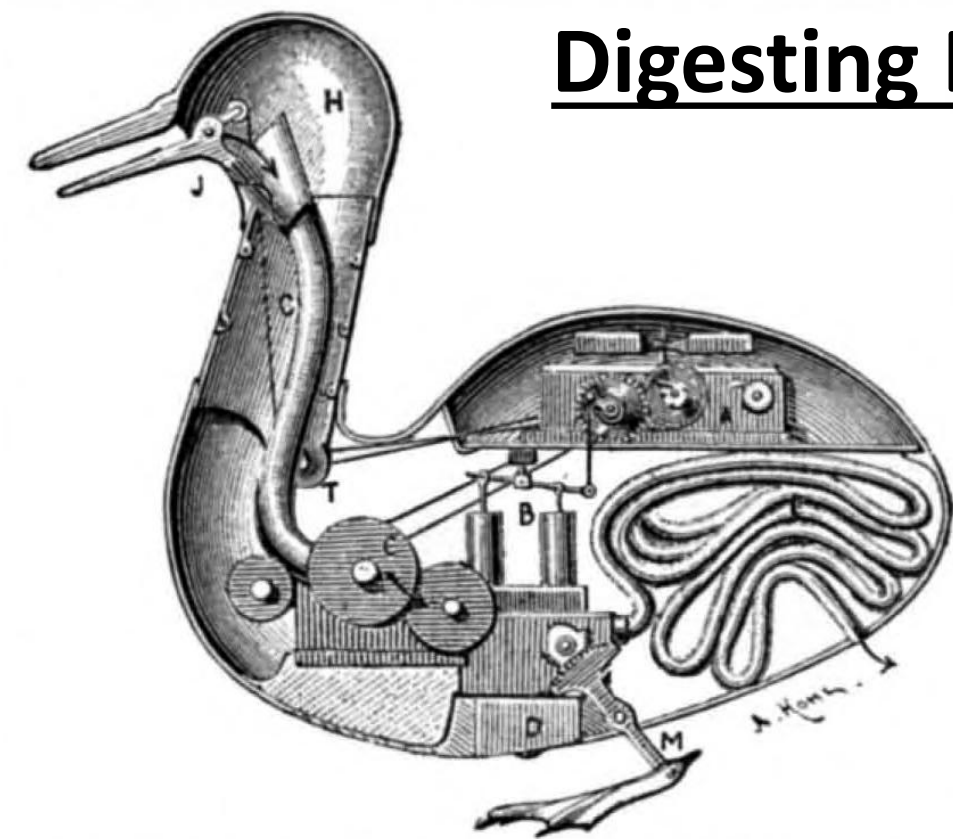


The Golem in Jewish culture

a being shaped from clay and brought to life to protect the community.



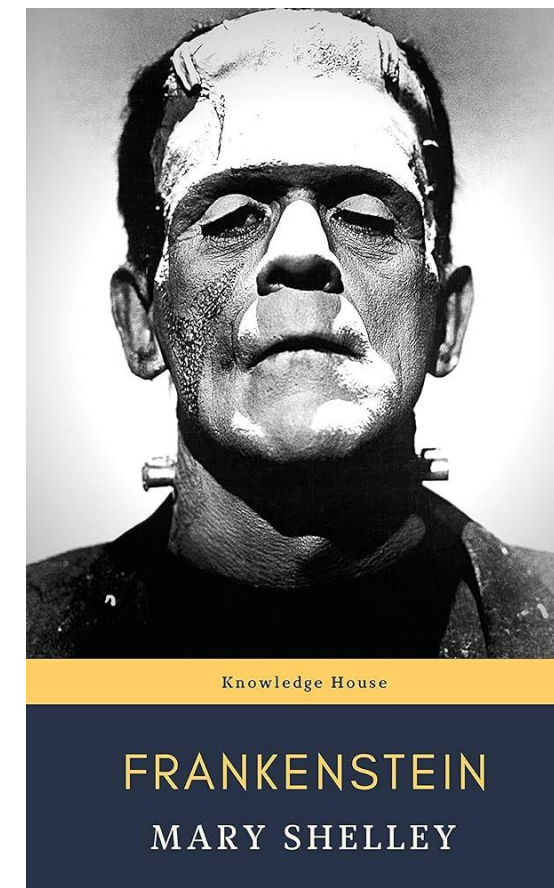
An old idea



INTERIOR OF VAUCANSON'S AUTOMATIC DUCK.
A, clockwork; B, pump; C, mill for grinding grain; F, intestinal tube;
J, bill; H, head; M, feet.

Digesting Duck of Vaucanson 1764

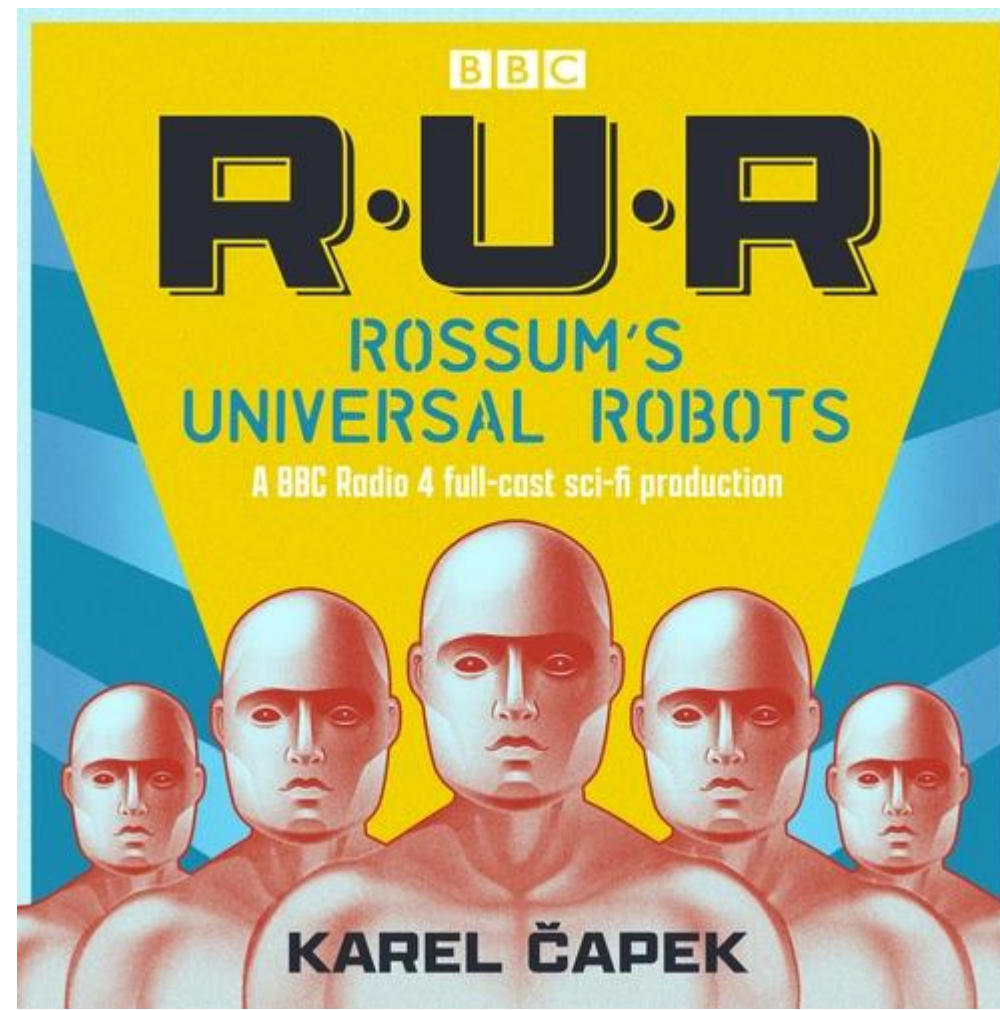
One of the first mechanical dolls ever built; this one could digest and flap its wings.



The Frankenstein monster 1818

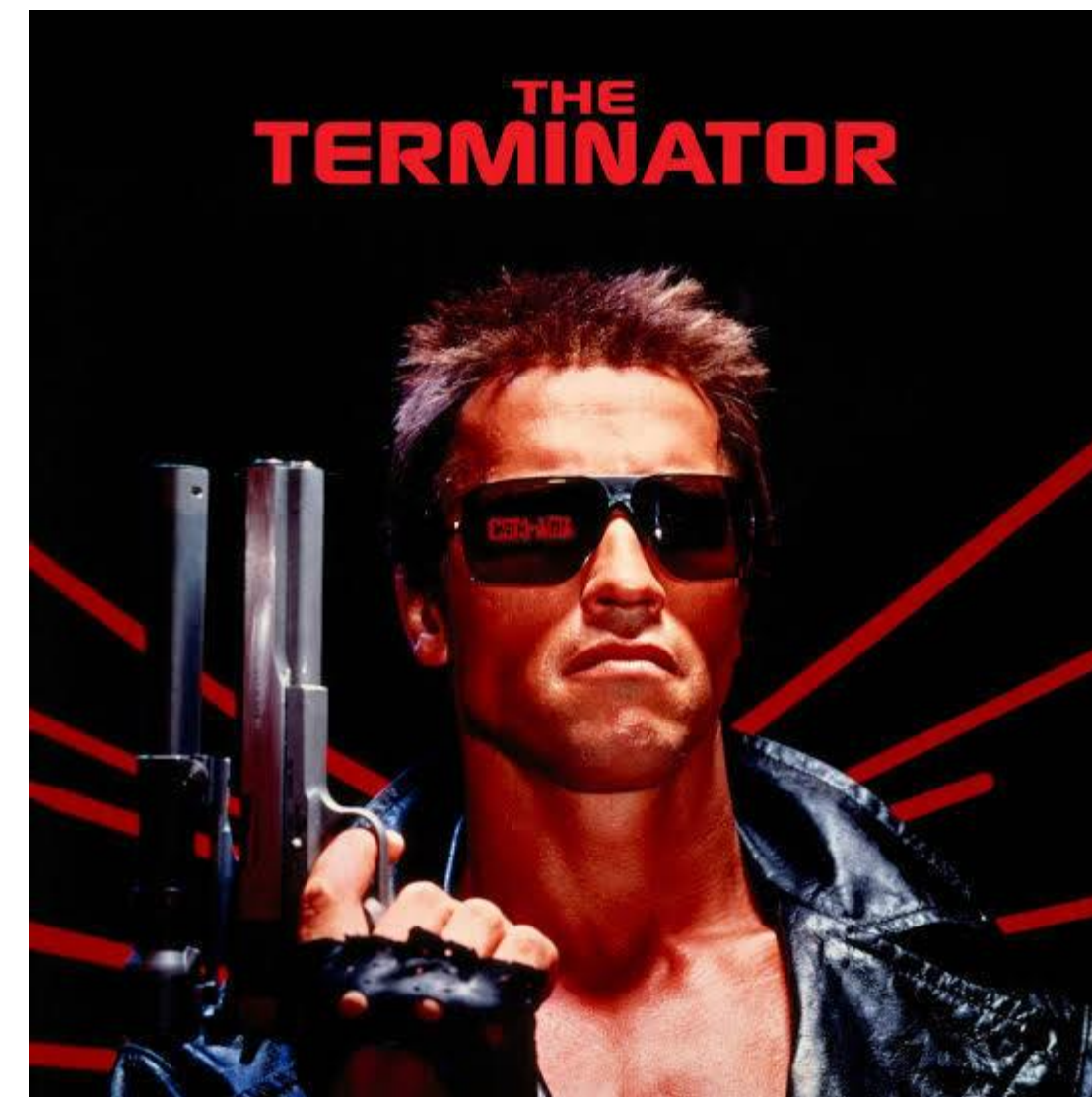
An artificial life created by humans.
It is a precursor to the idea that robots should be artificial and man-made.

An old idea



Rossum's Universal Robots 1921

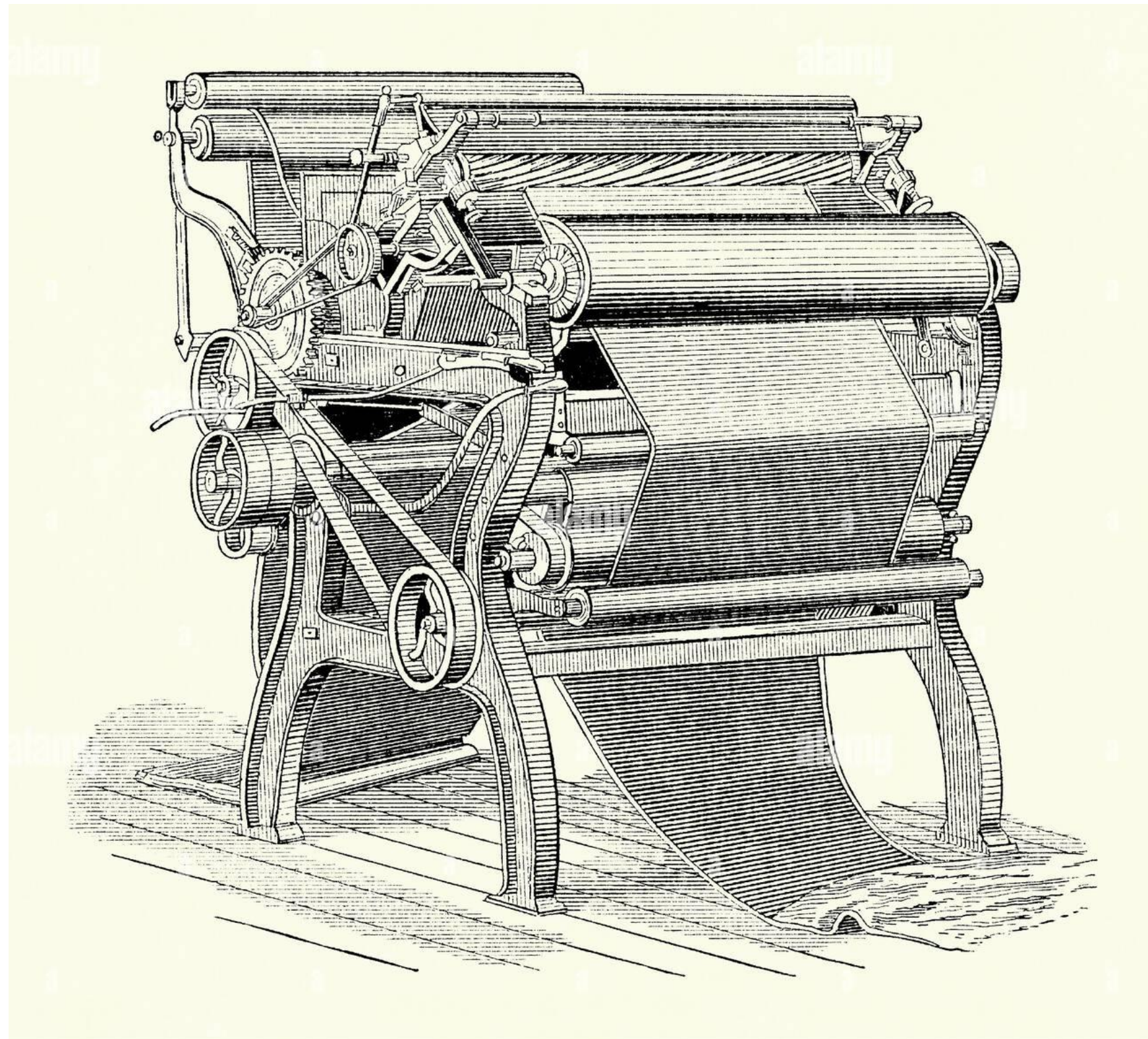
First time the word robot was put. Talk about artificial beings build like Frankenstein as working force



Terminator saga



Automation



Power loom, the first industrial machine, 1890s

increased output by a factor of 40

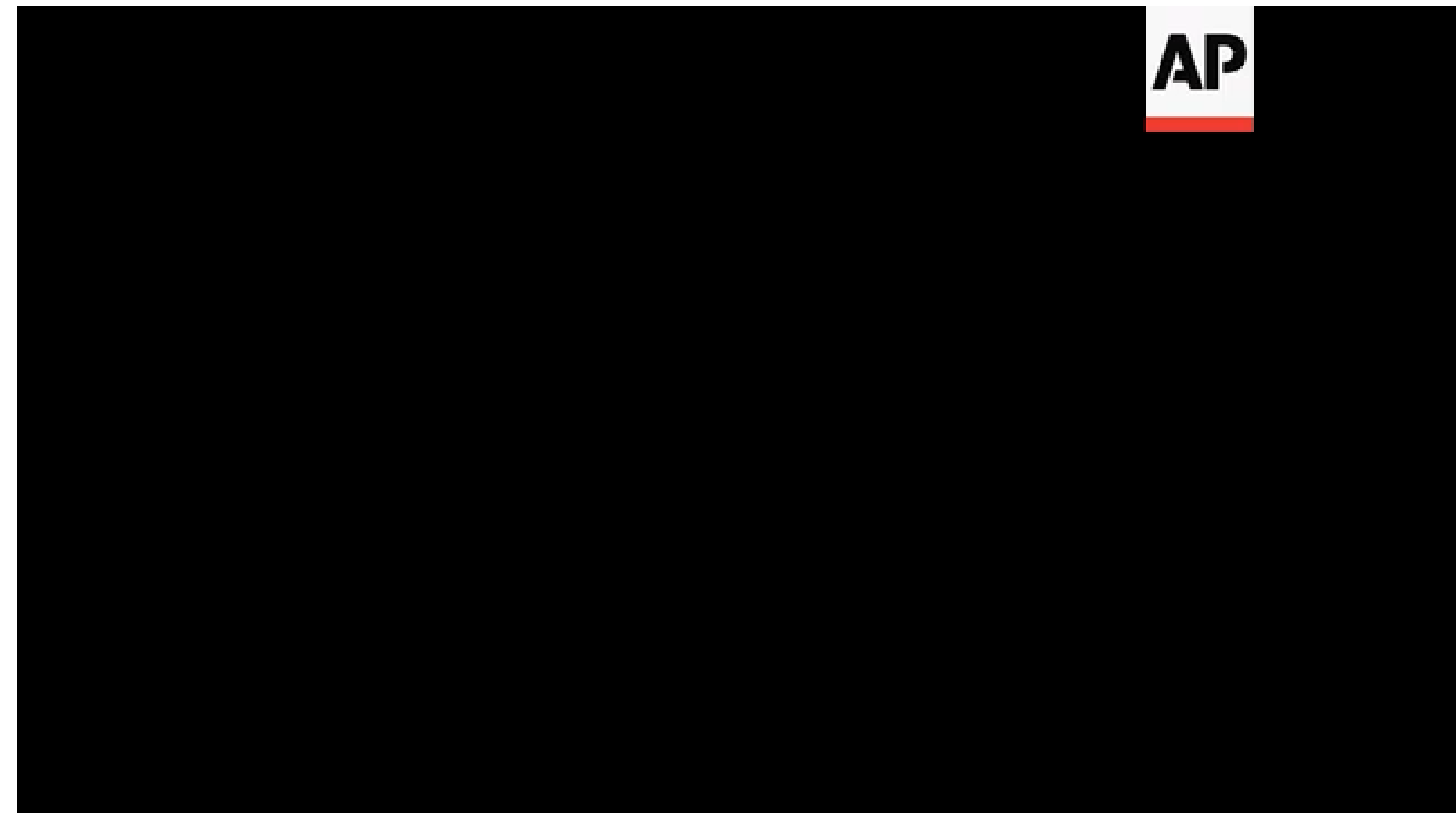
Steam engines and electricity were invented and made automation of the work possible.

Launched the second industrial revolution, where manual work gradually started to be replaced.

First real robot

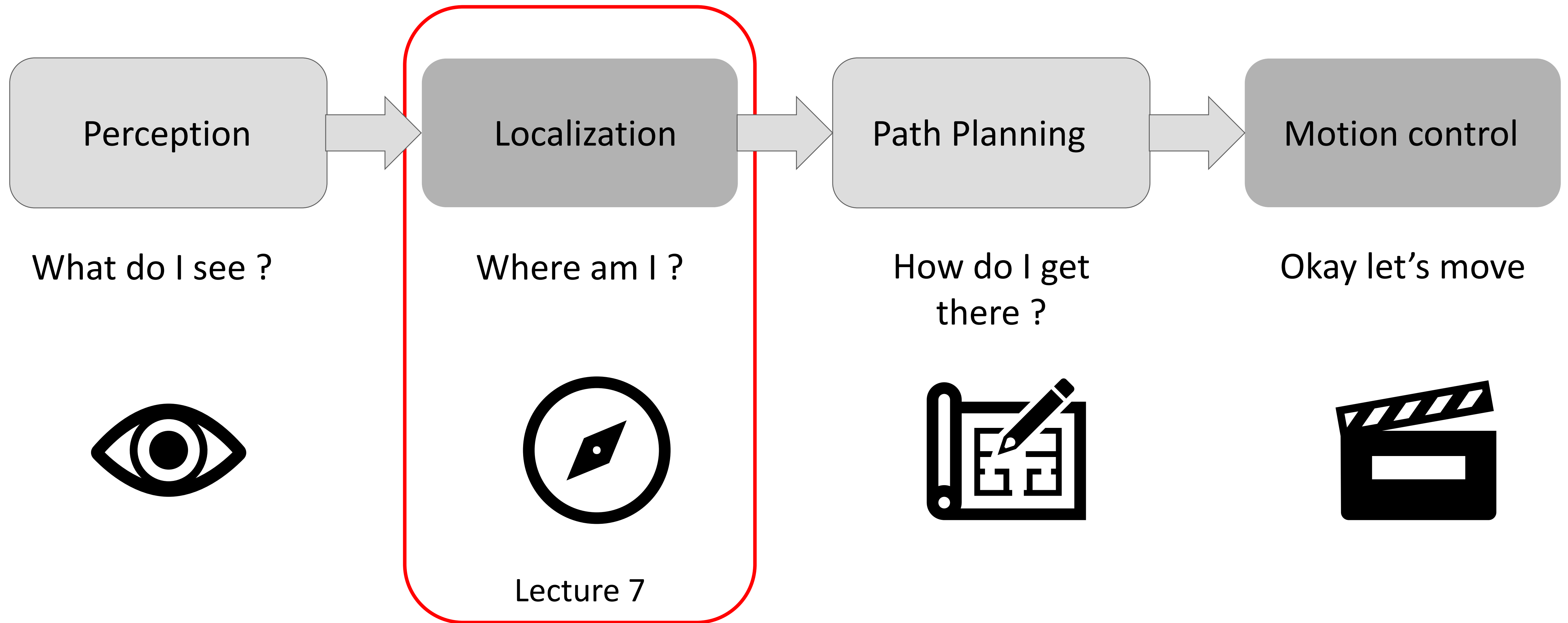
Electronics and computer science started to become more important.

Hydraulic technology became very advanced.



<https://www.youtube.com/watch?v=hxsWeVtb-JQ>
Unimate , 1961 , General Motors

How does a robot work today?



3 Key questions in robotics

What is around me ?

How do I get there ?

Where am I ?

Why localization ?

Localization allows reliable navigation.

Ability to follow a path

Knowing when you have arrived

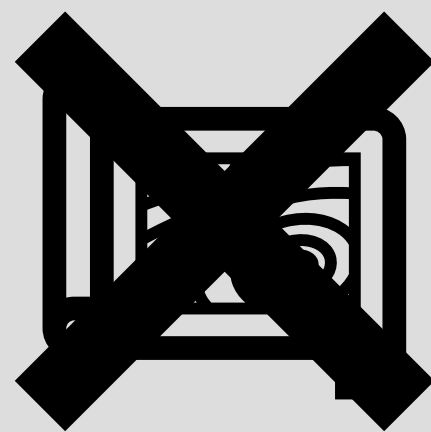
Two cases:

Localization

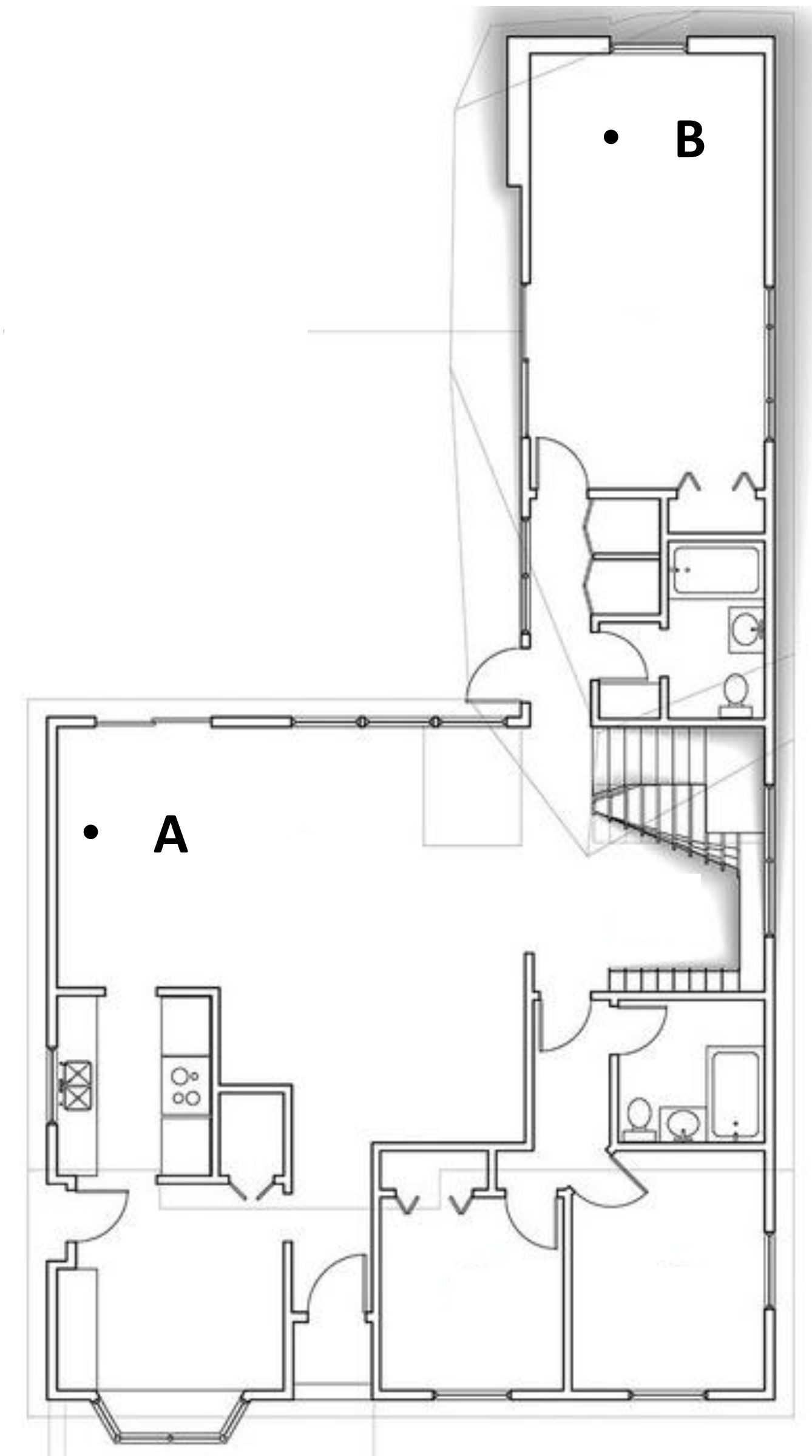


Localization with
a saved map

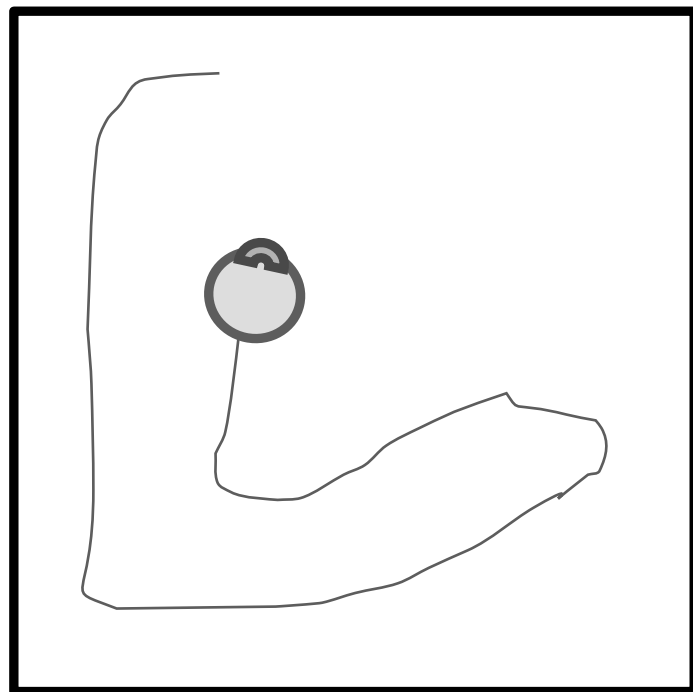
SLAM



Localization
without a saved
map (SLAM)

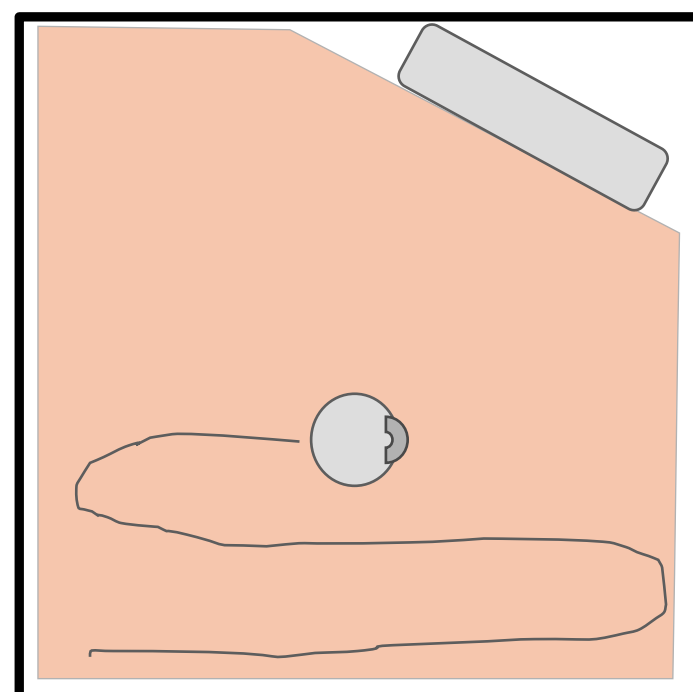


Roomba cleaner



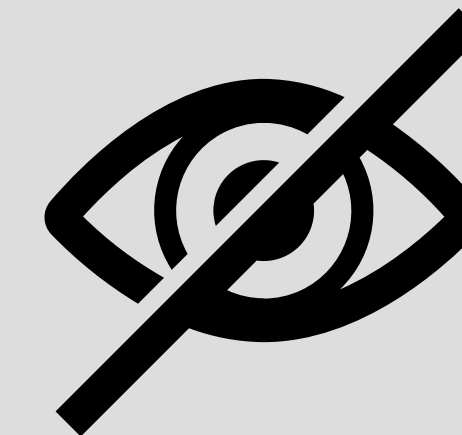
The robot does not know where it is or where the obstacles are.
It will not move well

Cleaning robot without localization module



The robot knows where it is and where are the obstacle . It can clean more efficiently the floor

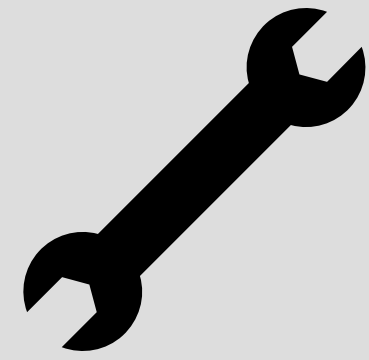
Cleaning robot with localization



If you move blindfolded **in an unknown room**, you collide with obstacles and move quite randomly

A parallel with Human

Localization methods



Sensor-based localization



Uses sensors to determine where the robot is

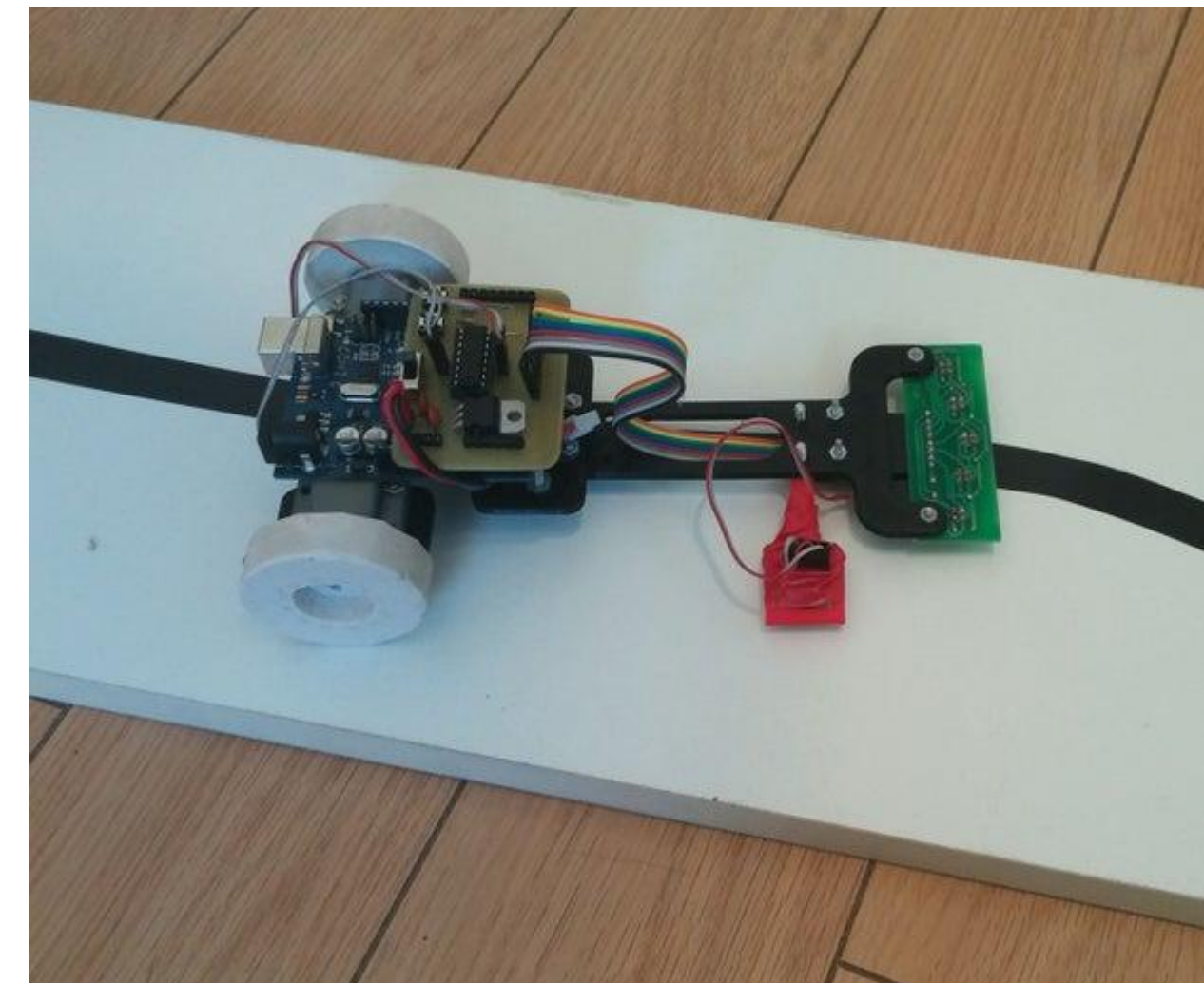


Very simple to implement



Very sensitive to sensor drift

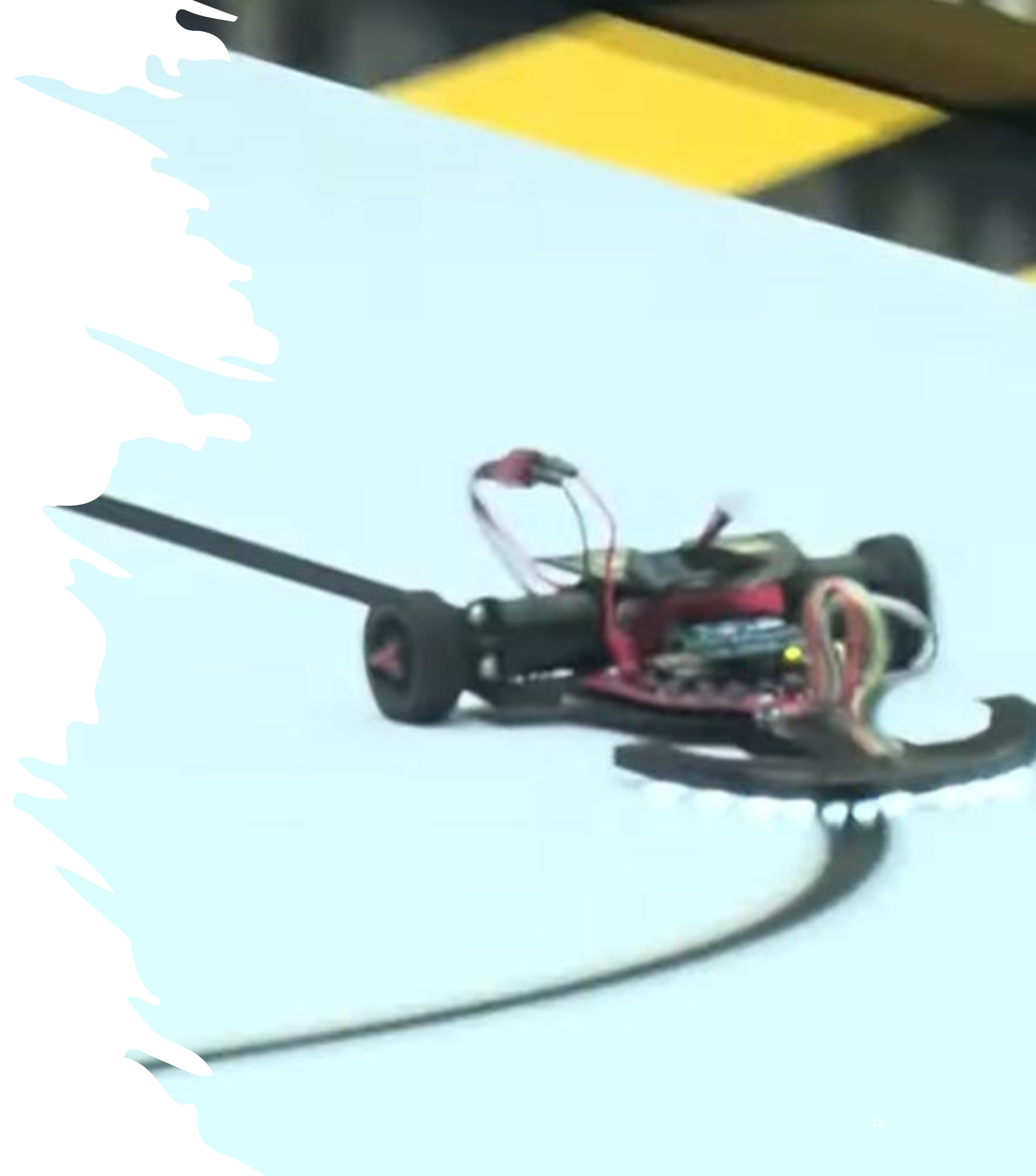
1960s – 1980s



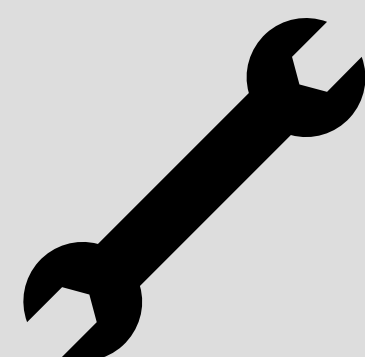
E.g. Line follower robots

Line follower robots

<https://www.youtube.com/watch?v=VwG451K62oE>



Localization methods



Beacons based localization



Use beacons put in a room to be able to tell where the robot is

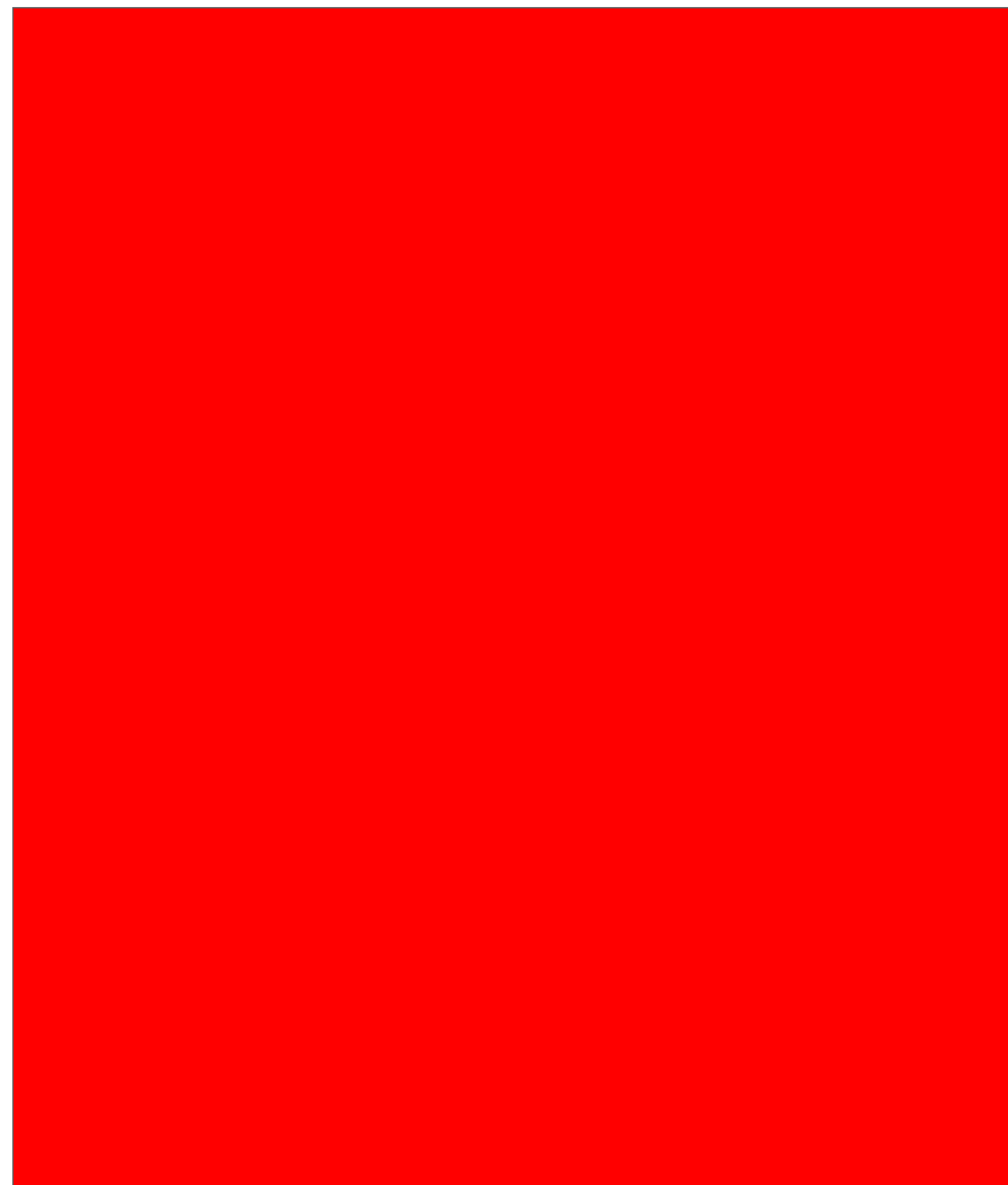


The computation is not done on the robot
(useful for swarms)



Requires beacons to be installed

1980s – 1990s



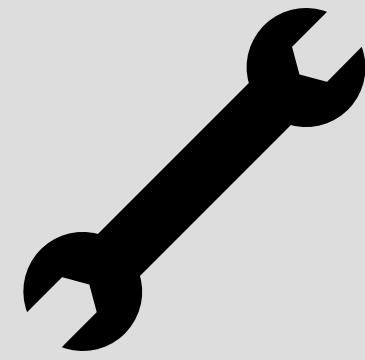
<https://www.youtube.com/watch?v=NrKrCPo16Y8>

amazon warehouse storing robot

<https://www.youtube.com/watch?v=IWsMdN7HMuA>



Localization methods



GPS based
localization



Use GPS to localize
yourself in the world



Very precise and reliable



Mostly works outdoors
Can be costly



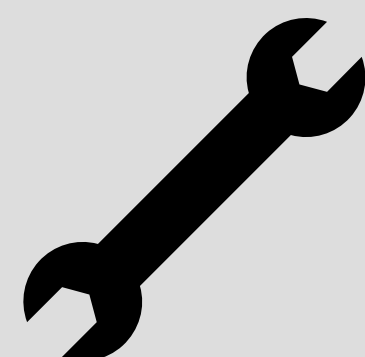
E.g. Google Maps

AT crawler heavy duty robot

<https://www.youtube.com/watch?v=G9RXRht6SaA>



Localization methods



Maps based localization



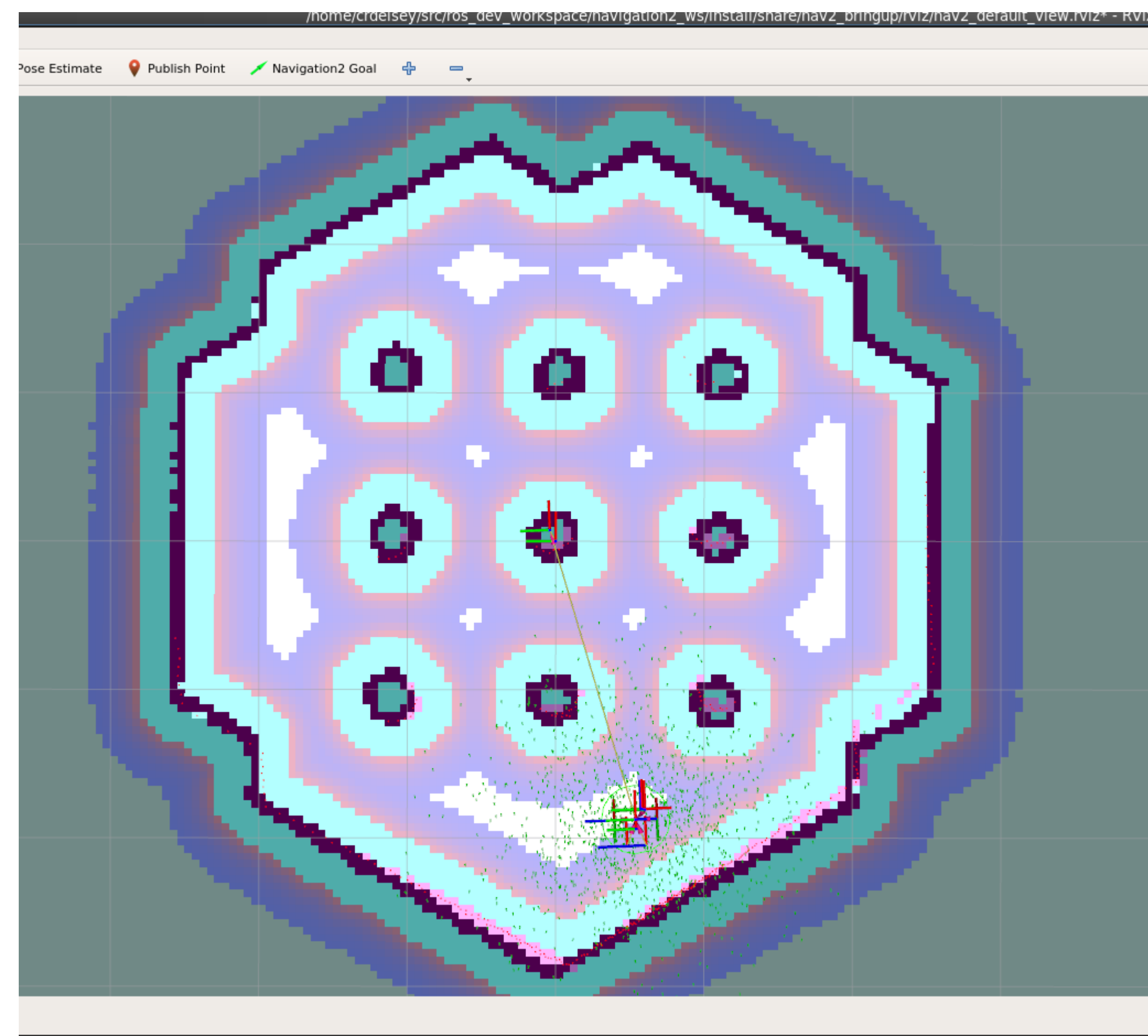
Use a saved map to know where you are



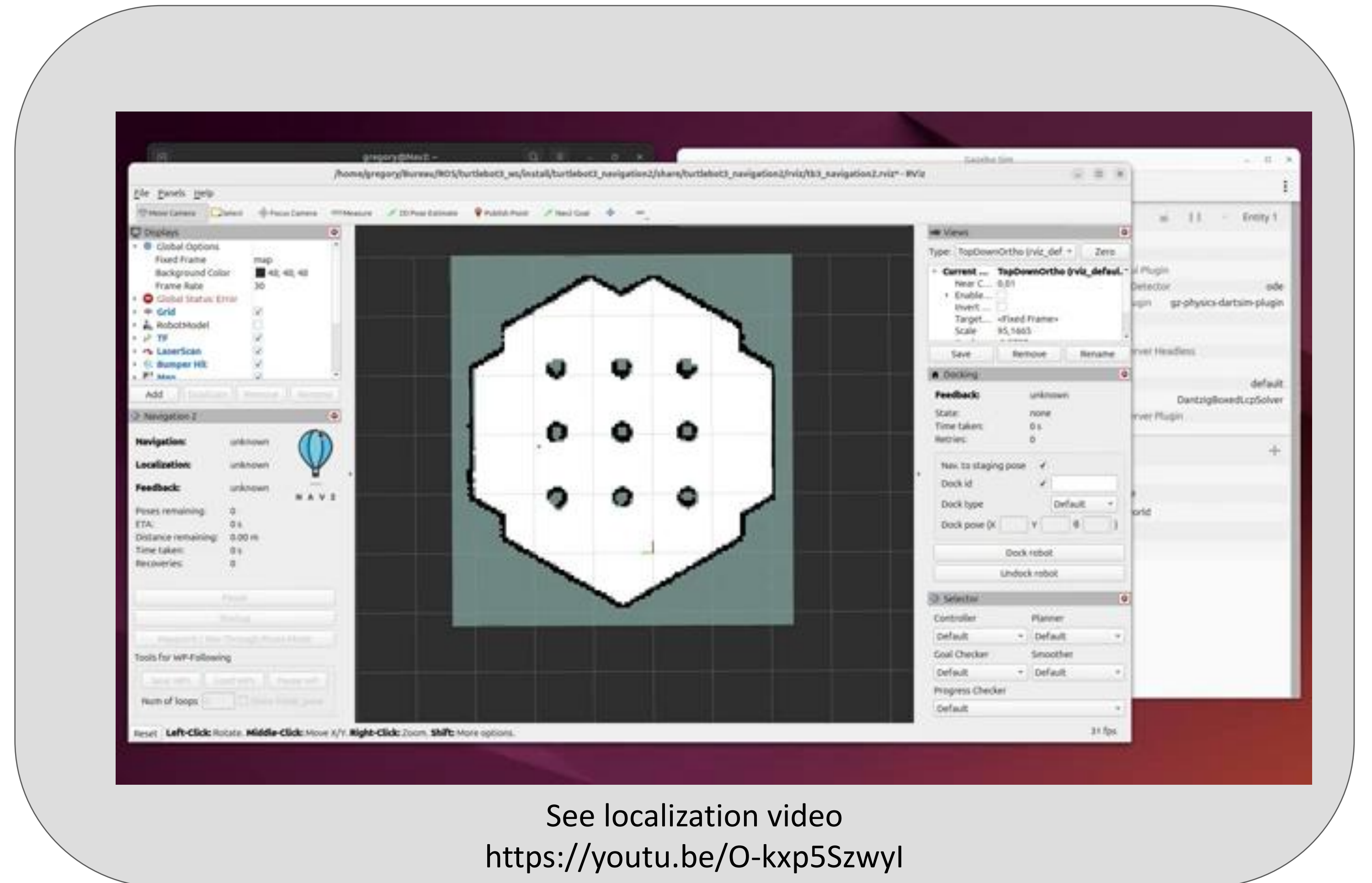
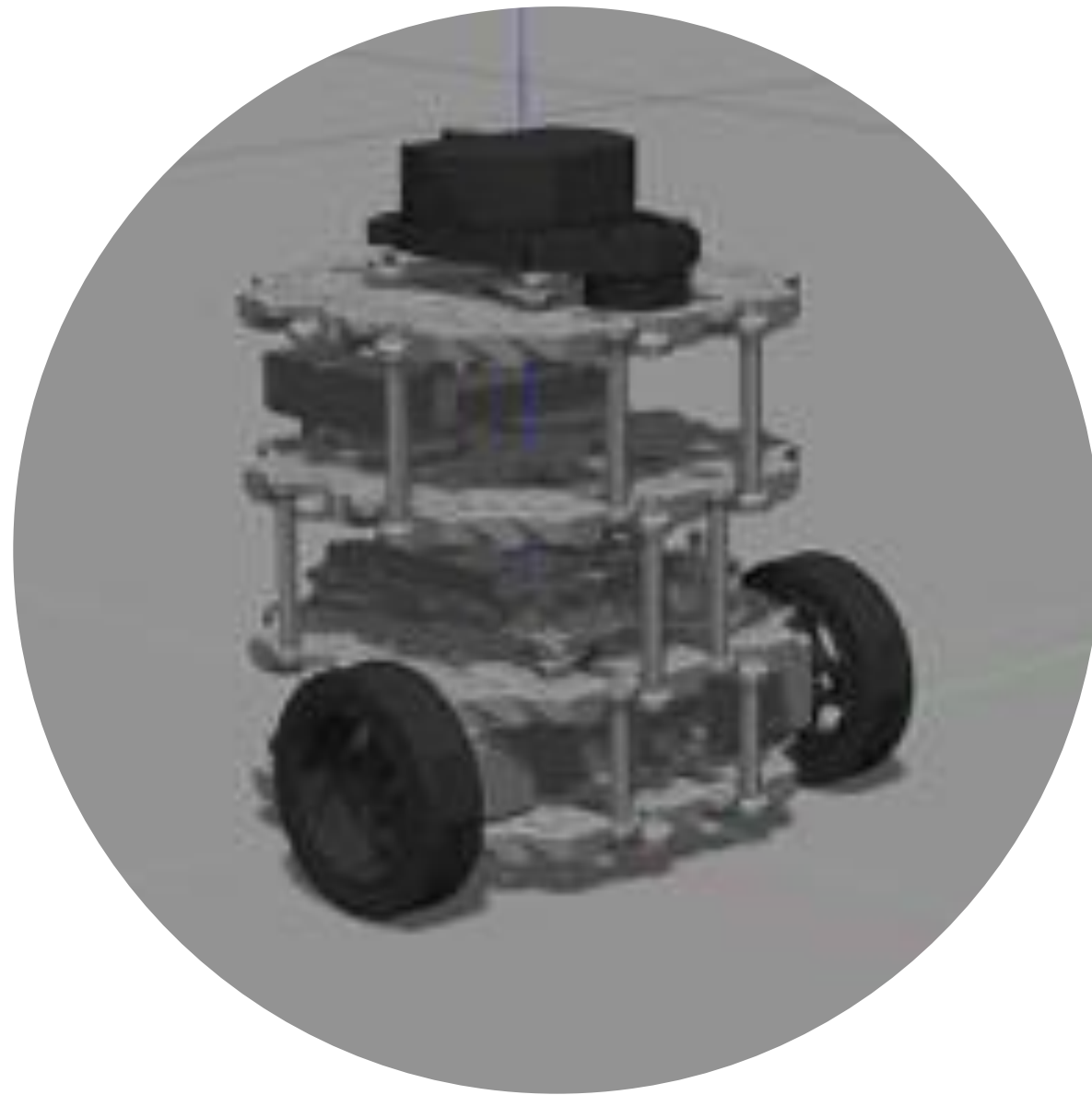
Huge step forward in robotics
(2001-2004)
Most reliable method



Need a lot of computation



Example : TurtleBot robot

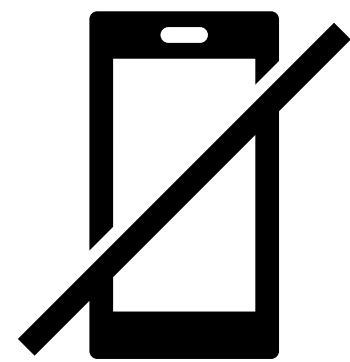


See localization video
<https://youtu.be/O-kxp5Szwyl>

Example “Being lost with a Map”

Imagine you are lost without GPS or phone, and you need to go to a destination

How will you go to the destination ? How will you know where you are ?



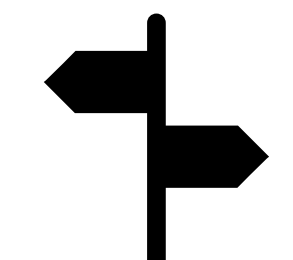
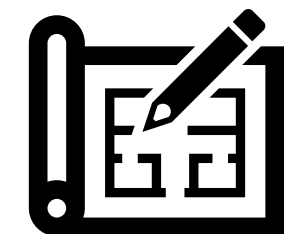
*No GPS
No phone
Only a map*

Find yourself searching for similarity between the map and where you are

Look at the best road possible to go to the position

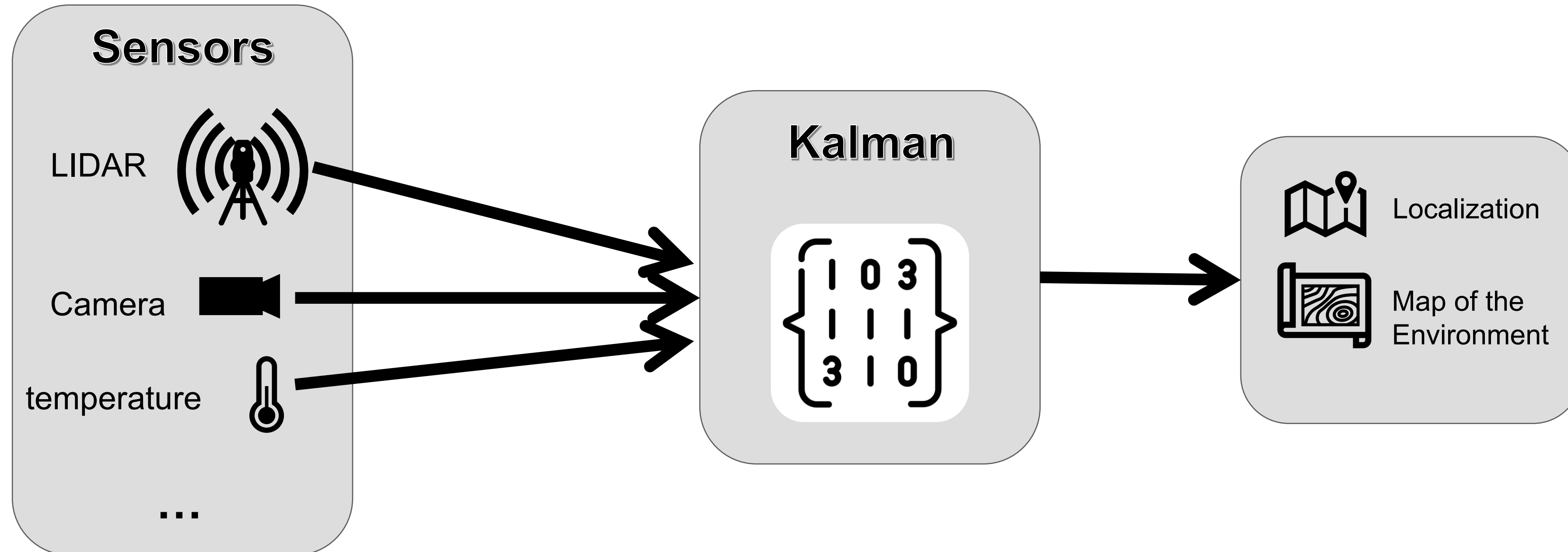
You follow your travel plan but, but the road, you wanted to take is blocked

You change your travel plan to a better one



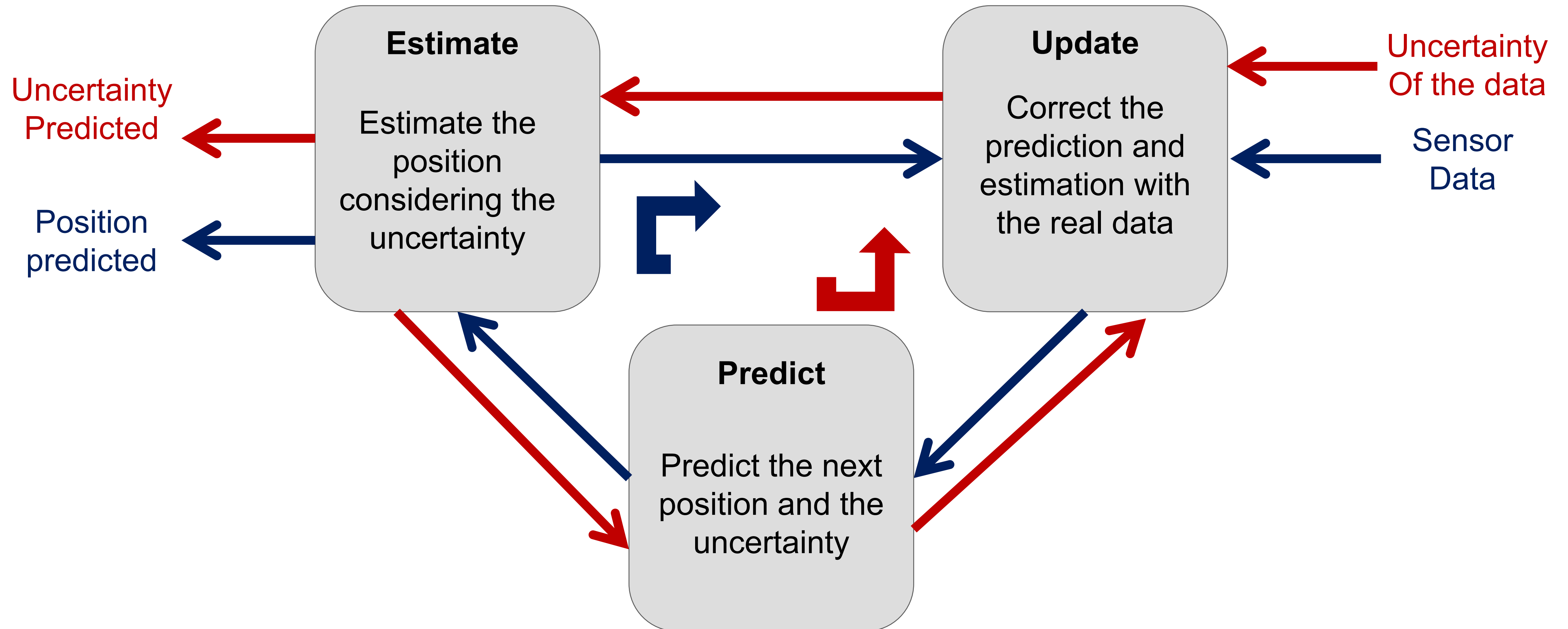
But you need to know again where you are ...

Kalman filter



Stores previous sensor data and uses it to make a prediction of future data that will occur to know the localization

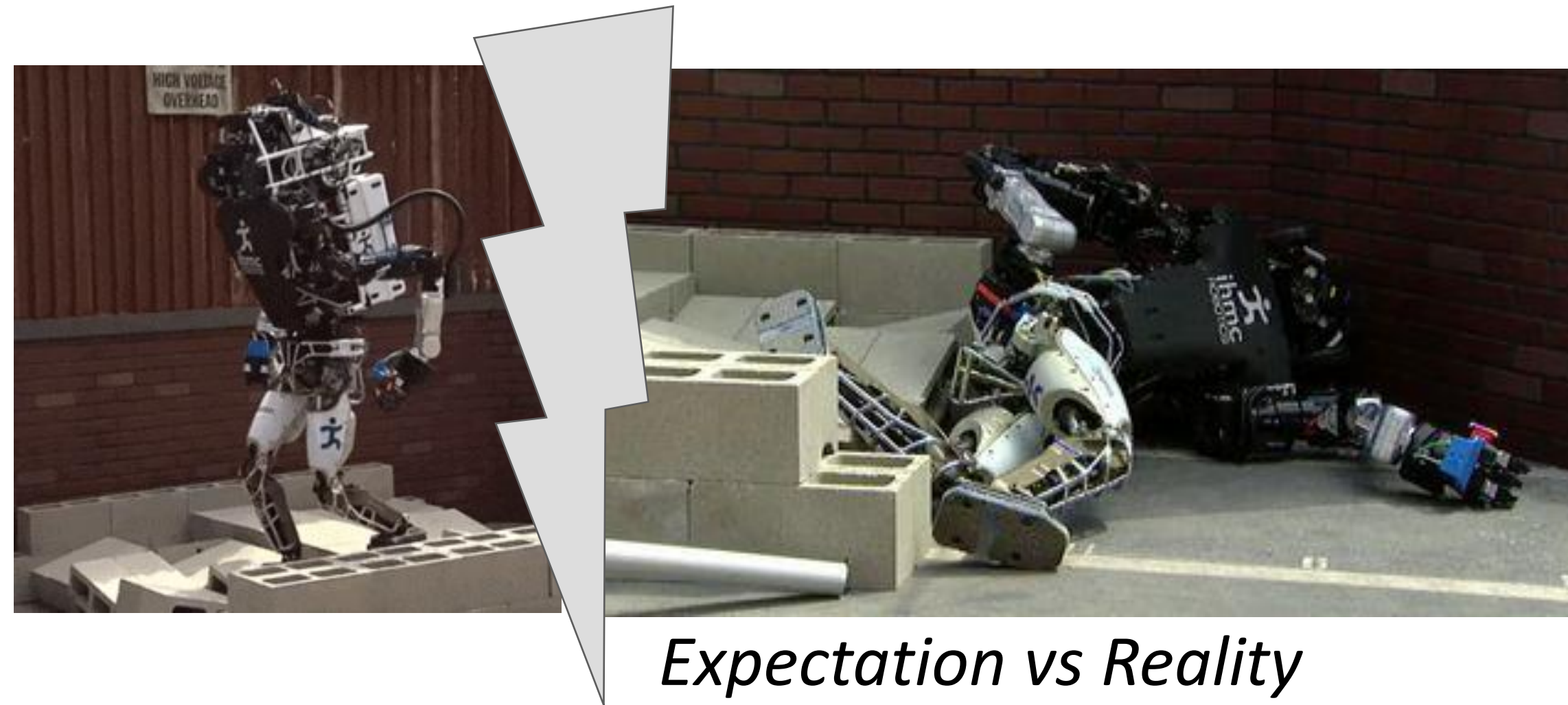
Kalman filter

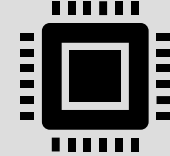


Kalman filter issues

The filter is obtained from a mathematical model of the sensors. 

Model are just a models and don't represent every case

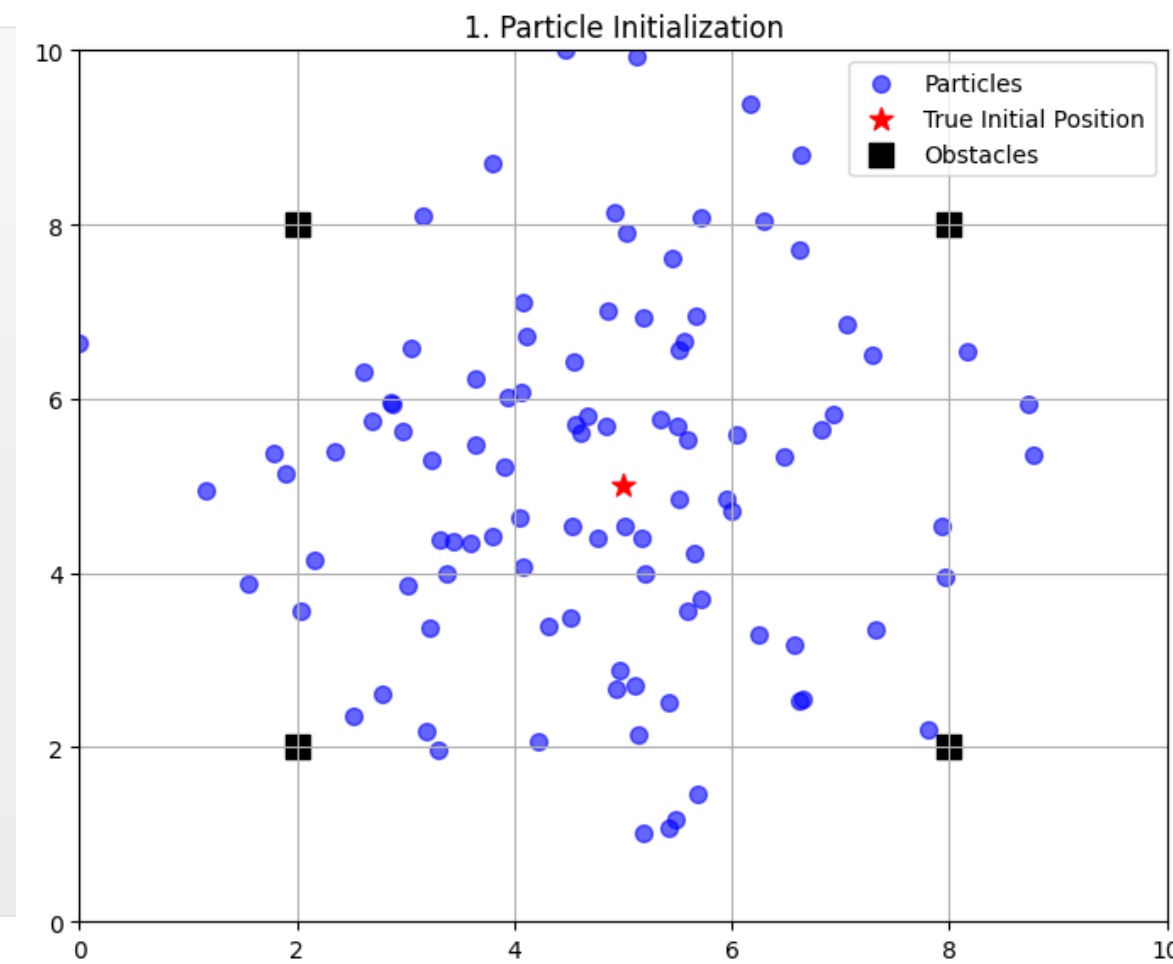


Requires matrix computations, which demand significant computing power. 

It is also hard to understand the choice of algorithm.

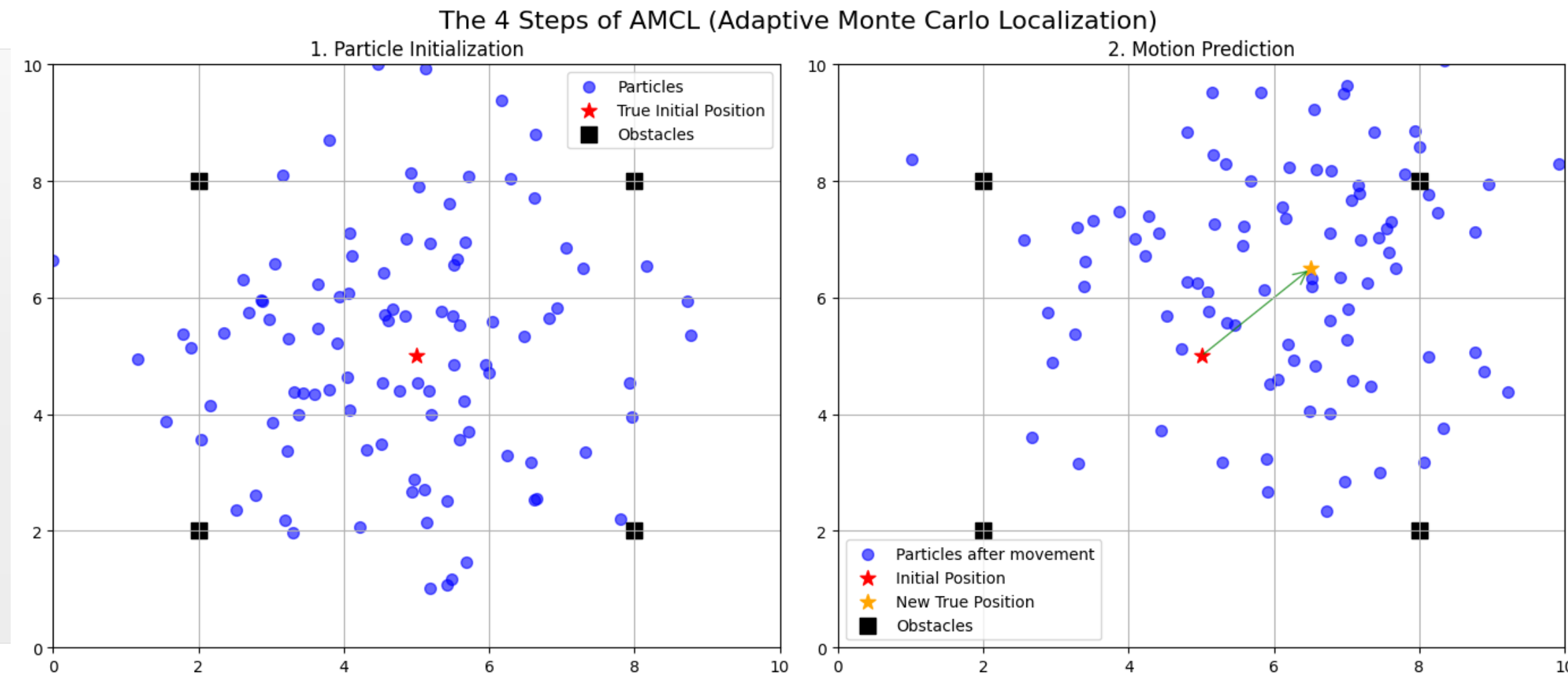
AMCL (Adaptive Monte Carlo Localization)

We initialize with all the potential points where the robot could be



AMCL (Adaptive Monte Carlo Localization)

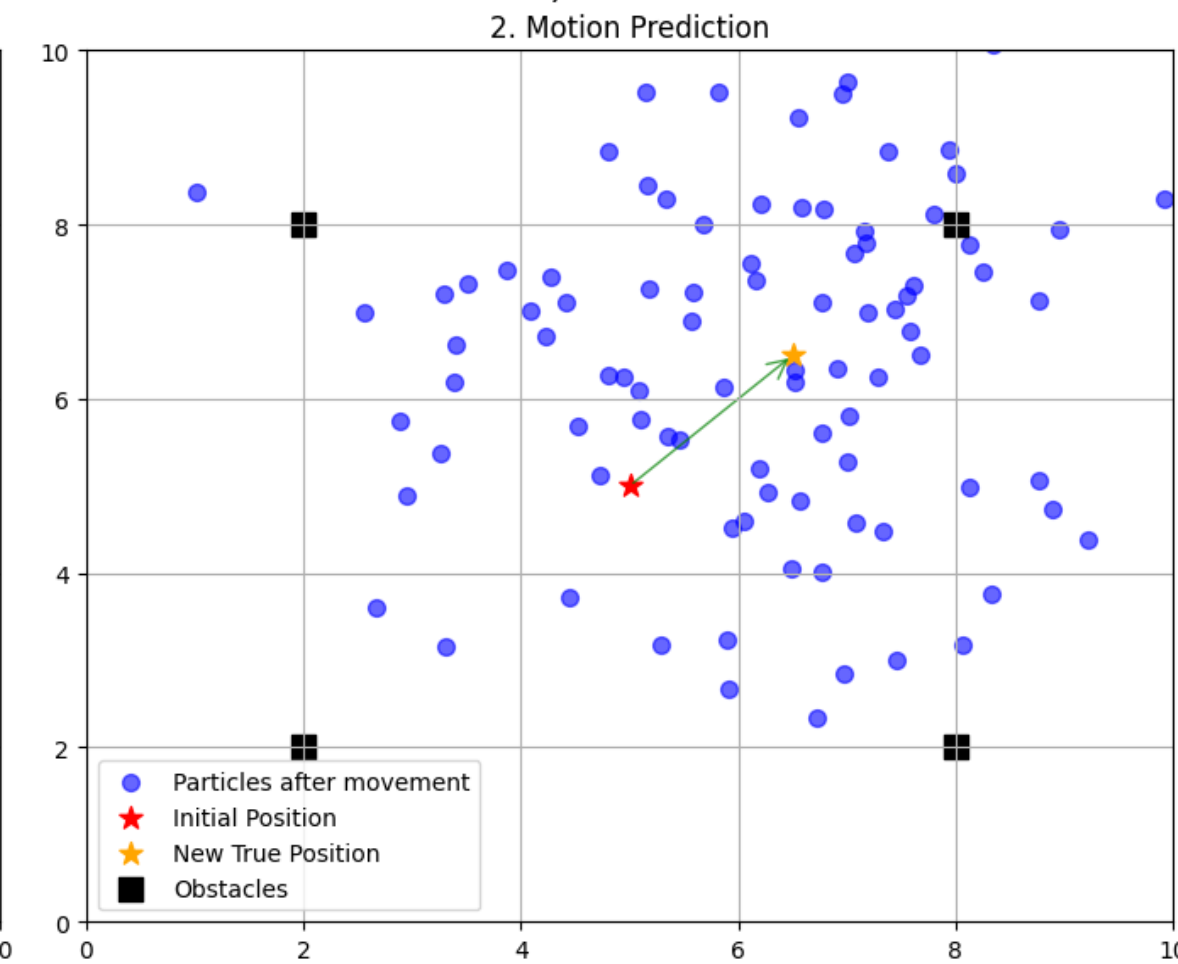
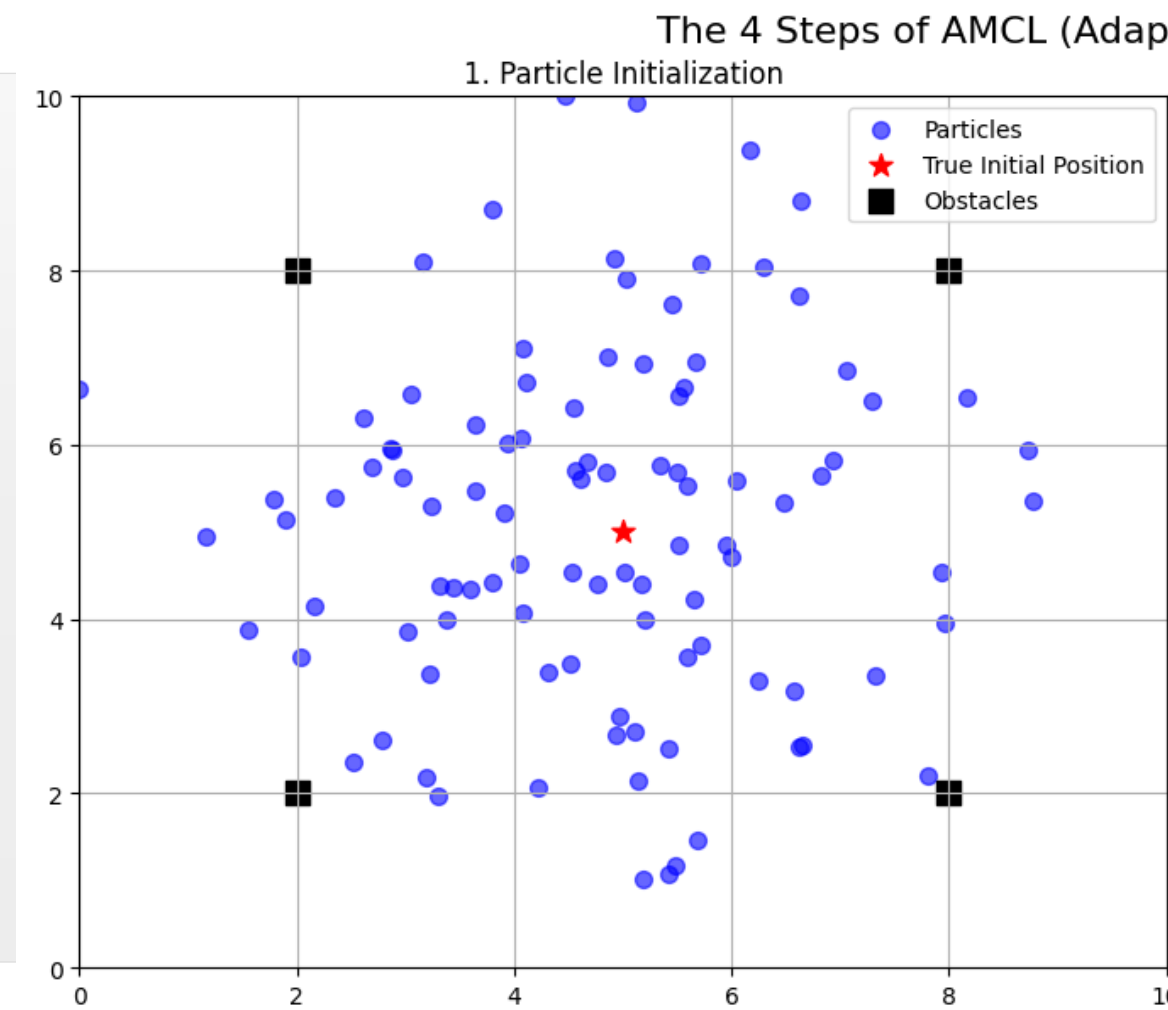
We initialize with all the potential points where the robot could be



We know how the robot moves, so we move the potential positions

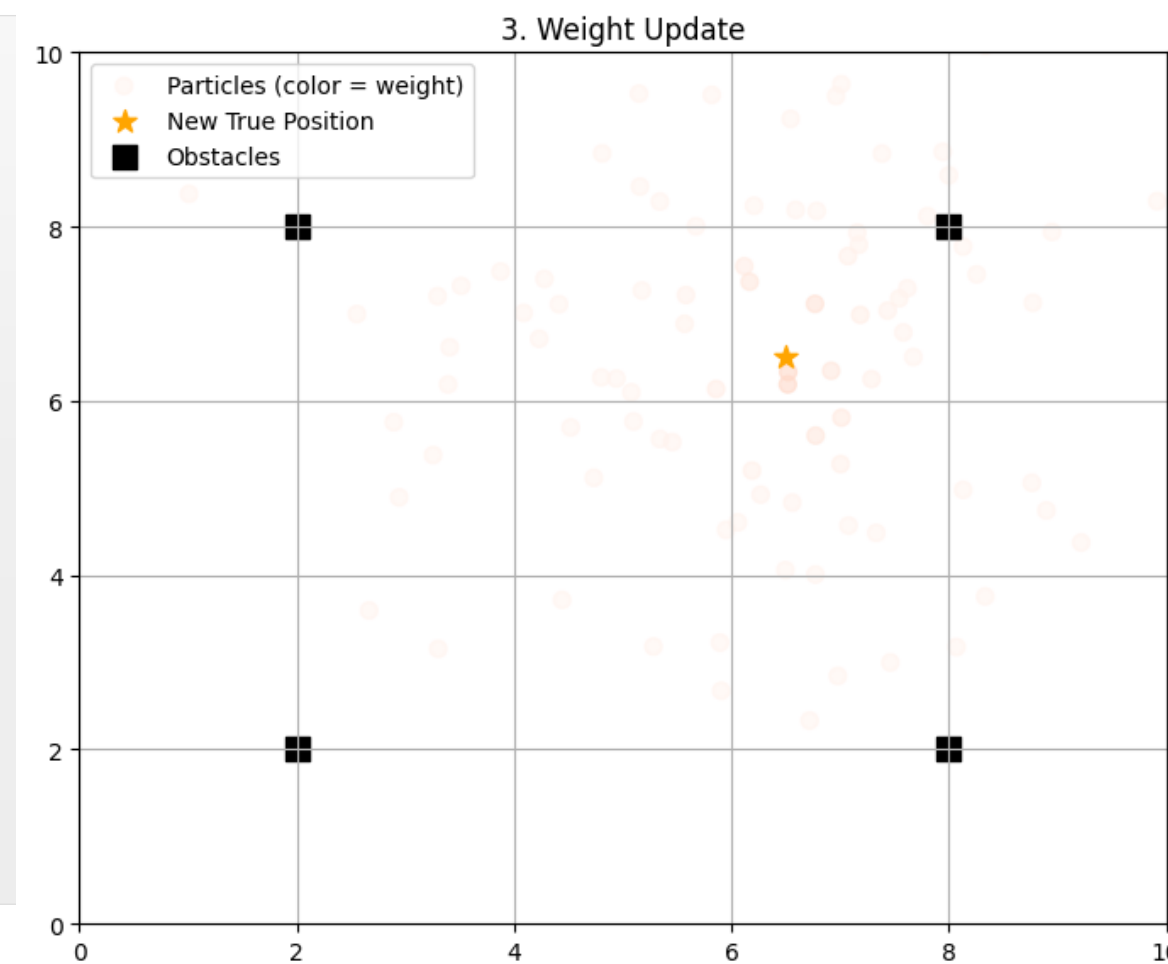
AMCL (Adaptive Monte Carlo Localization)

We initialize with all the potential points where the robot could be



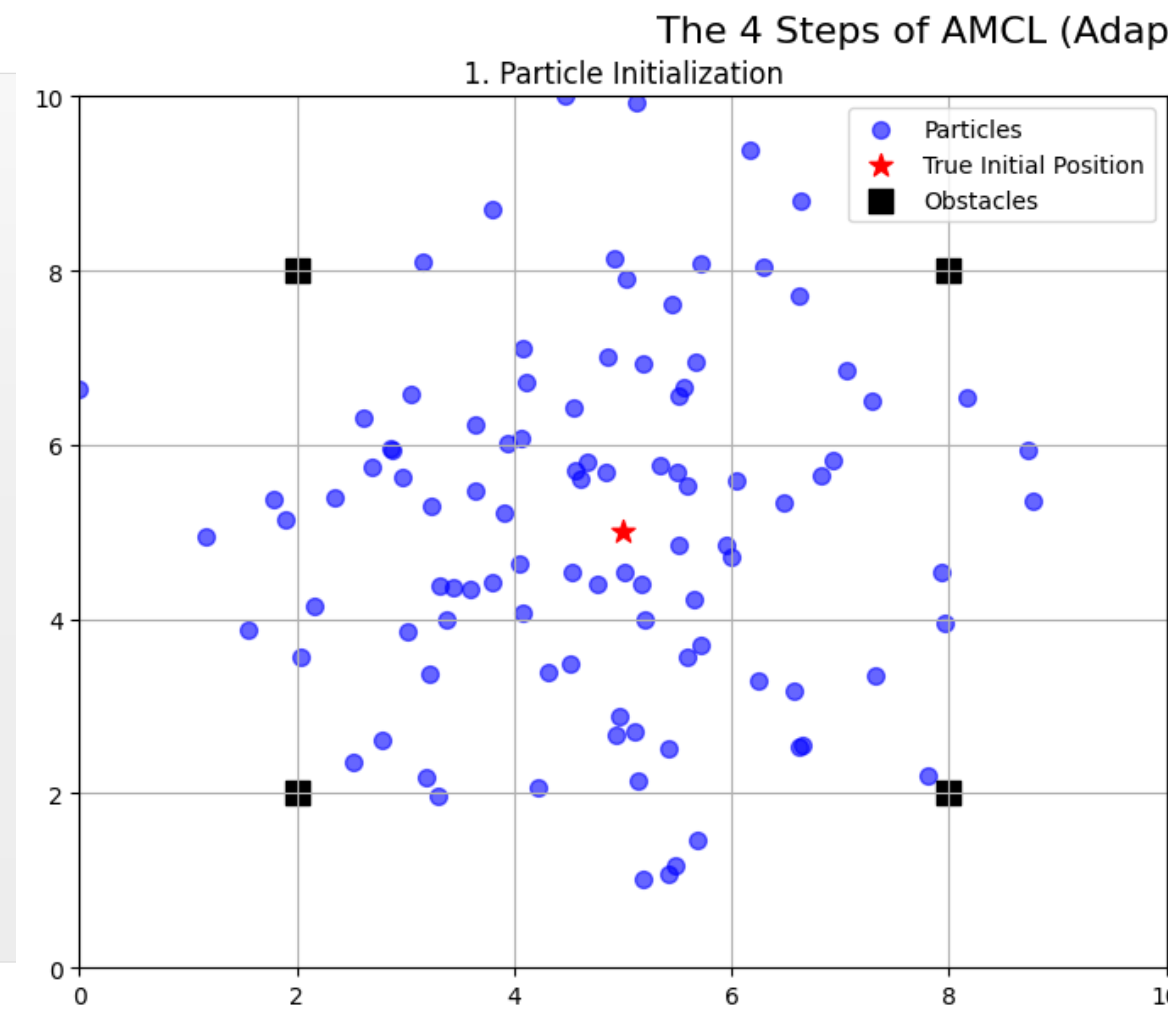
We know how the robot moves, so we move the potential positions

We update the probability

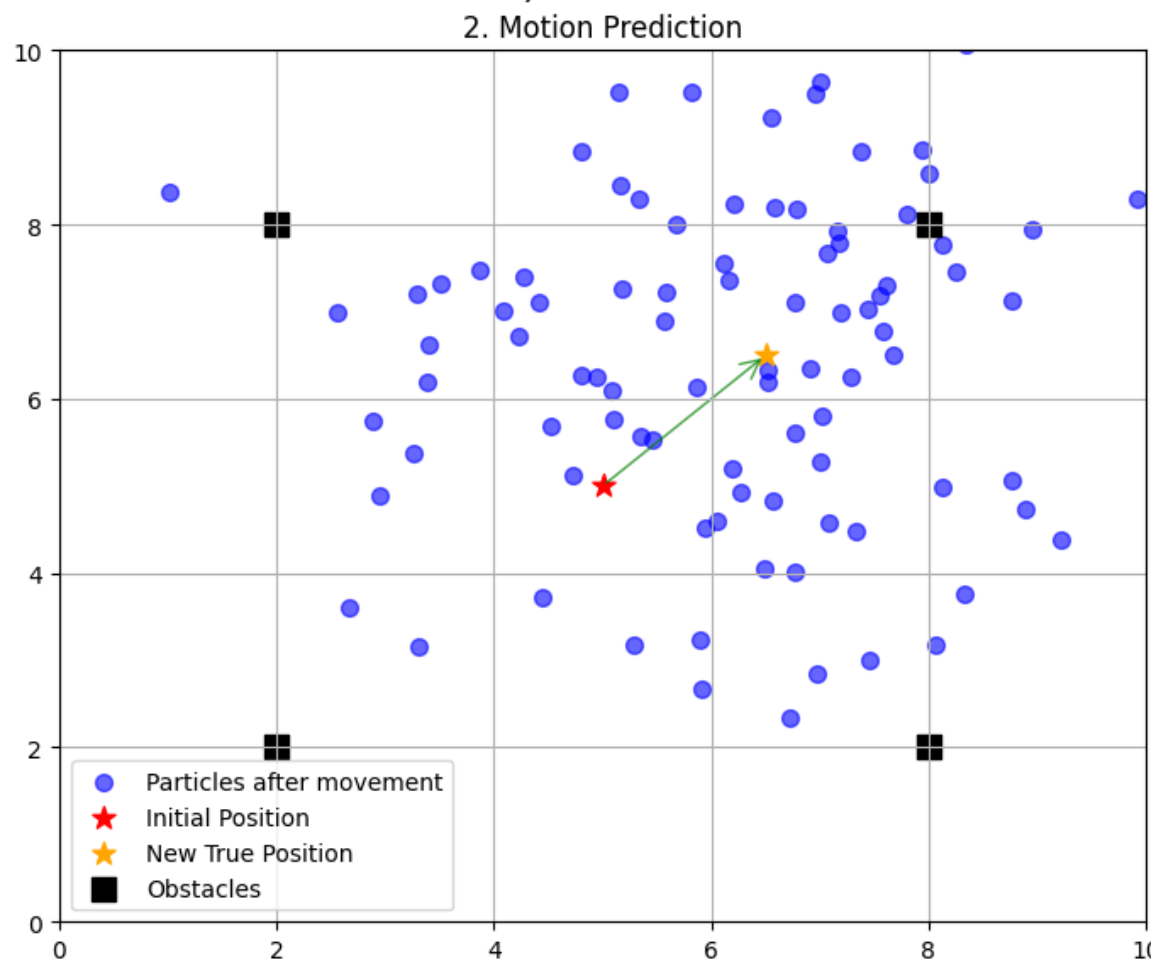


AMCL (Adaptive Monte Carlo Localization)

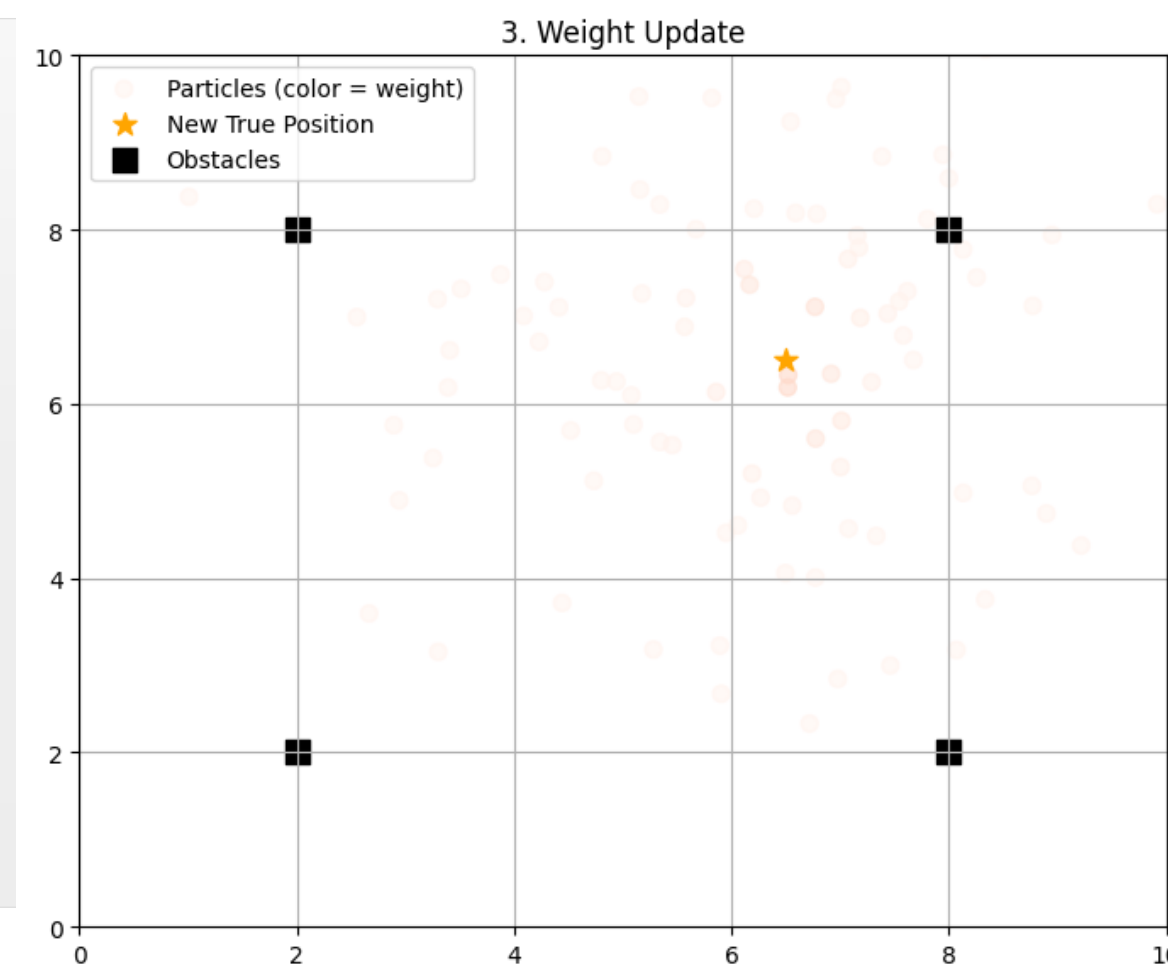
We initialize with all the potential points where the robot could be



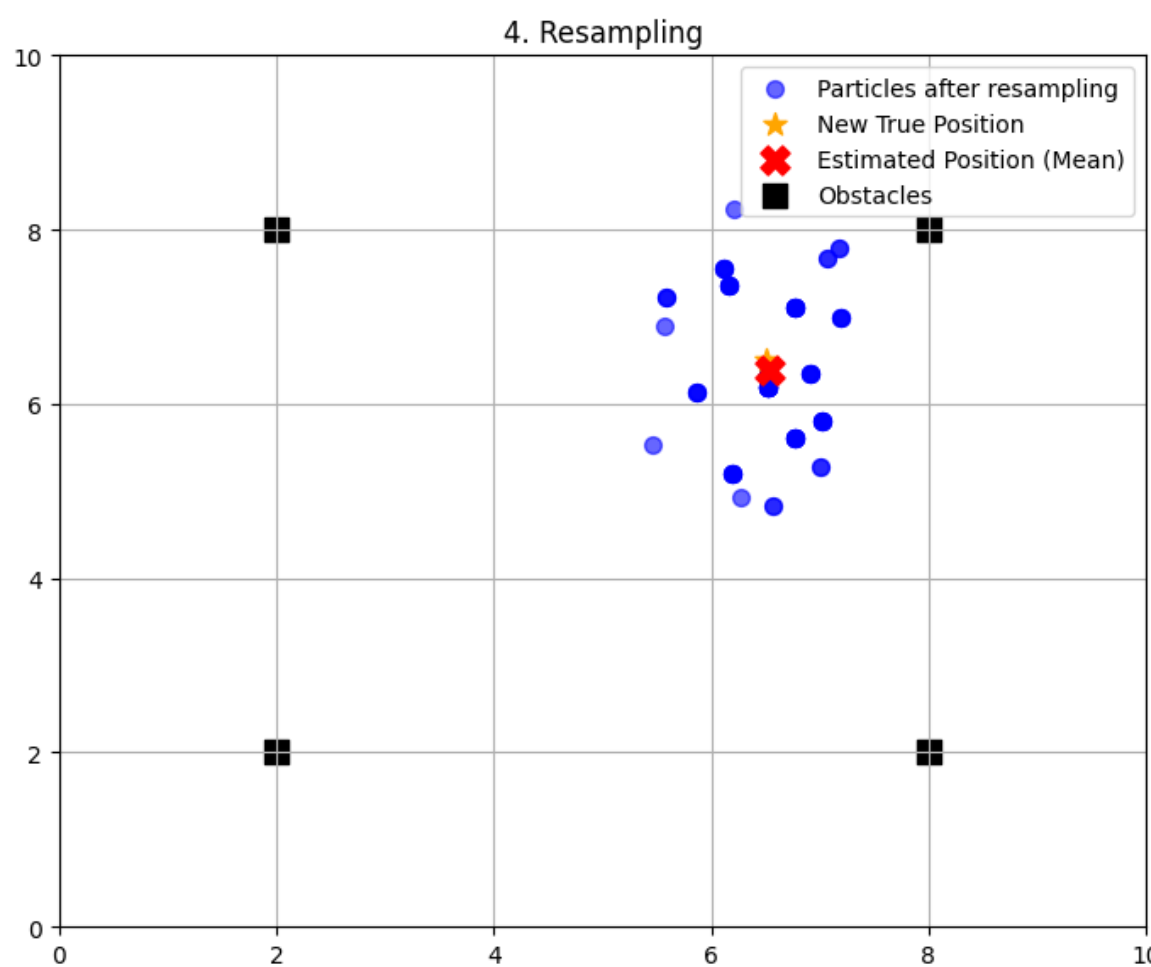
We know how the robot moves, so we move the potential positions



We update the probability



We calculate the mean probability to know where the robot is, and we resample to redo 2 and 3



Comparison between Kalman and AMCL

Kalman Filter	AMCL
Gaussian	Particle-based
Linear assumption	Non-linear
Fast	More robust
Hard to tune	Easier intuitively

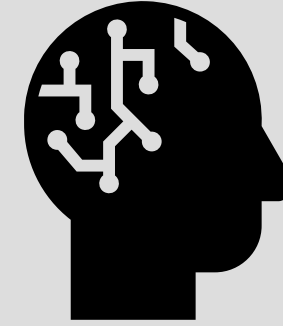
Localization methods

methods	pros	cons
Beacons based	Position externally computed	Cannot determine the position outside the beacons Need beacons to be installed Need communication between the robot and the beacons
Sensor Based	The robot knows its position internally	Sensor can drift
GPS based	Very precise and very reliable	Cannot work indoors Can be costly
Map	Most reliable method Understandable by humans	<i>Requires significant computational power</i>

Not a problem anymore
because of the
technology
advancement

Still, It's not enough

Yearning for autonomous robots and human-robot environments



Often doesn't know the map, or the map changes



SLAM a map-based method

Simultaneous

Localization

And

Mapping

Goals

Mapping

Map unknown environment



Localization

Know the position without the need of external data

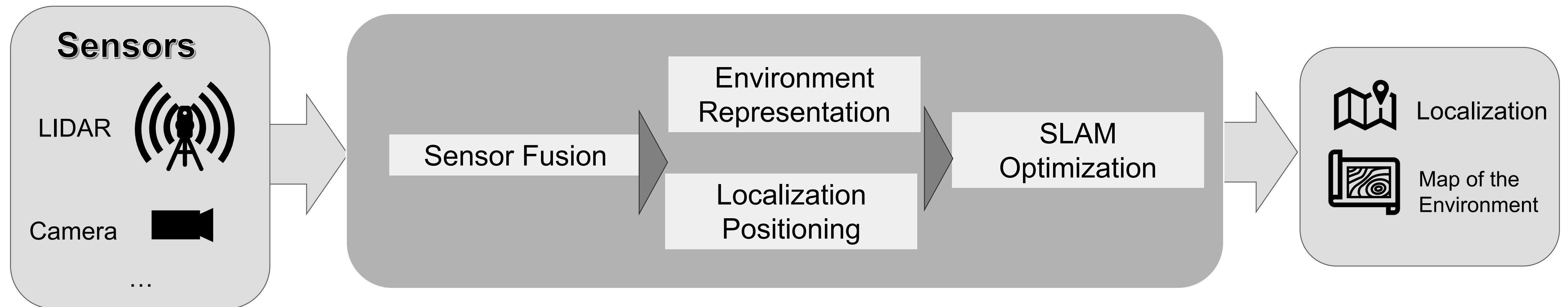


Simultaneous

Being able to perform both tasks at the same time



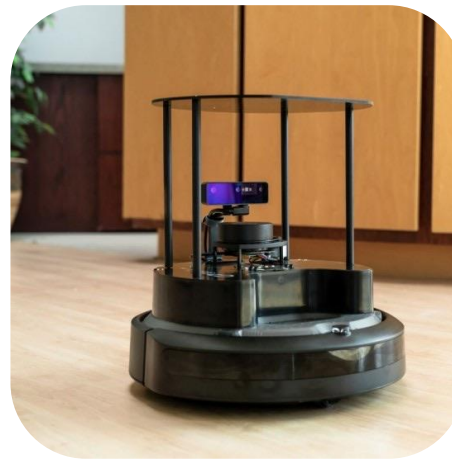
How SLAM works



SLAM algorithm

Example

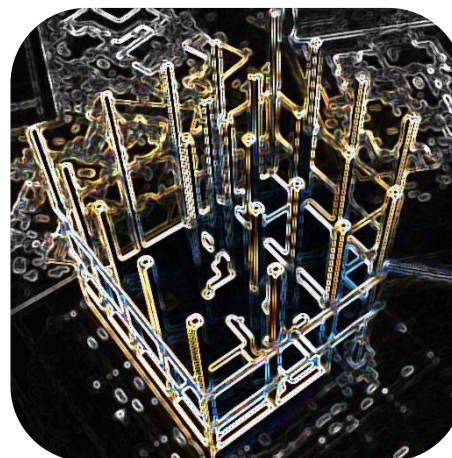
Use example



Robotic



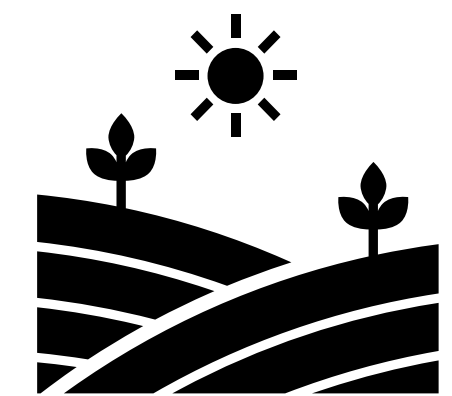
AR / VR



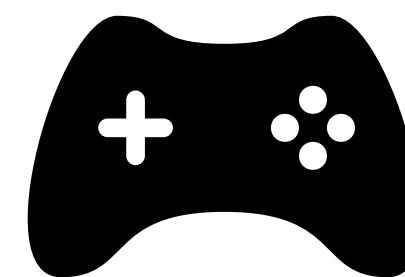
Architecture



Self-driving cars



Agriculture

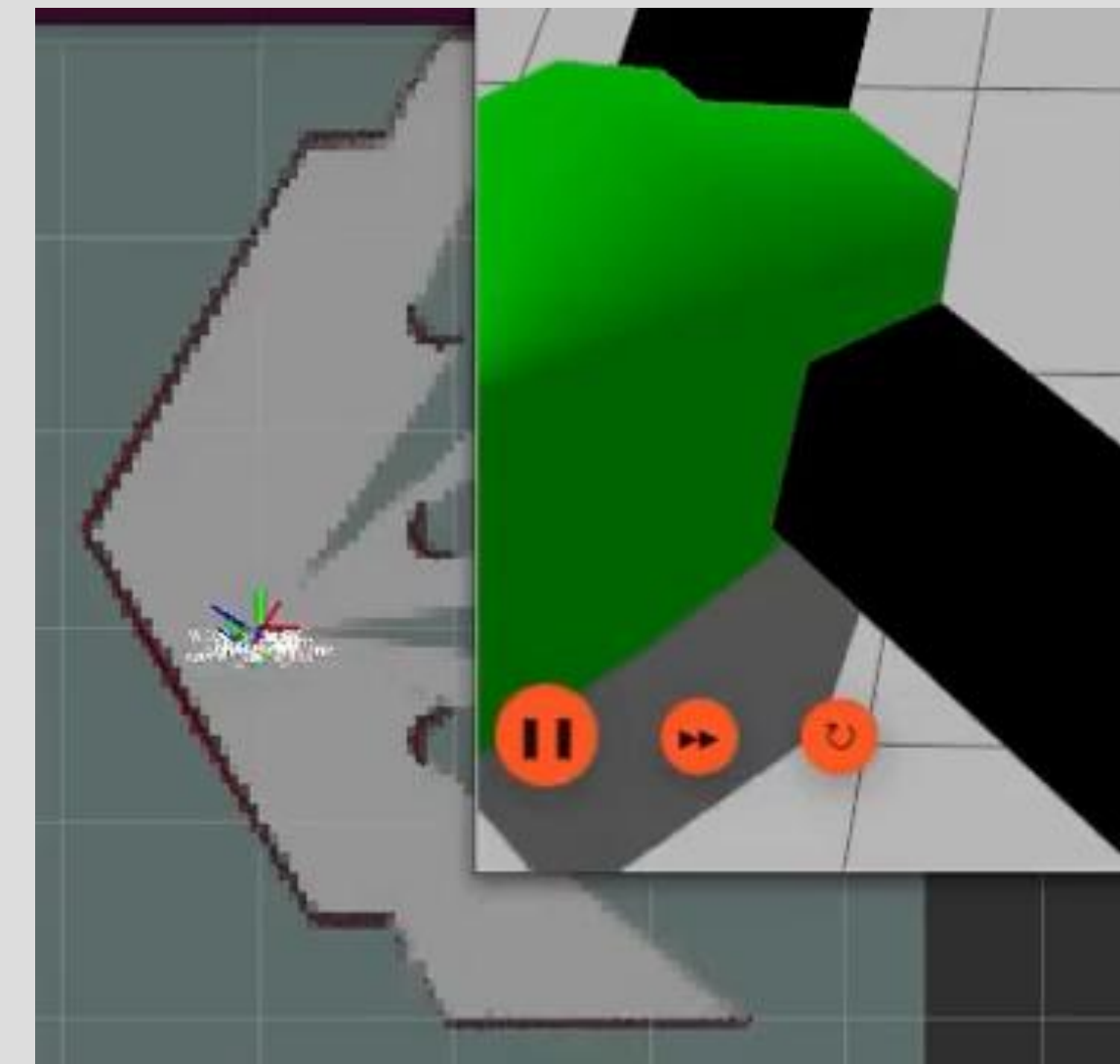


Video games

...

Almost everywhere
nowadays

Demonstration



See SLAM video
<https://youtu.be/4tnFLh795jk>

Pro and cons

Pros

Very accurate

Can work with any kind of sensor

Allow autonomous navigation

Permits real-time mapping

Cons

Computationally intensive

Sensor limitation

Environment constraints

Complex implementation

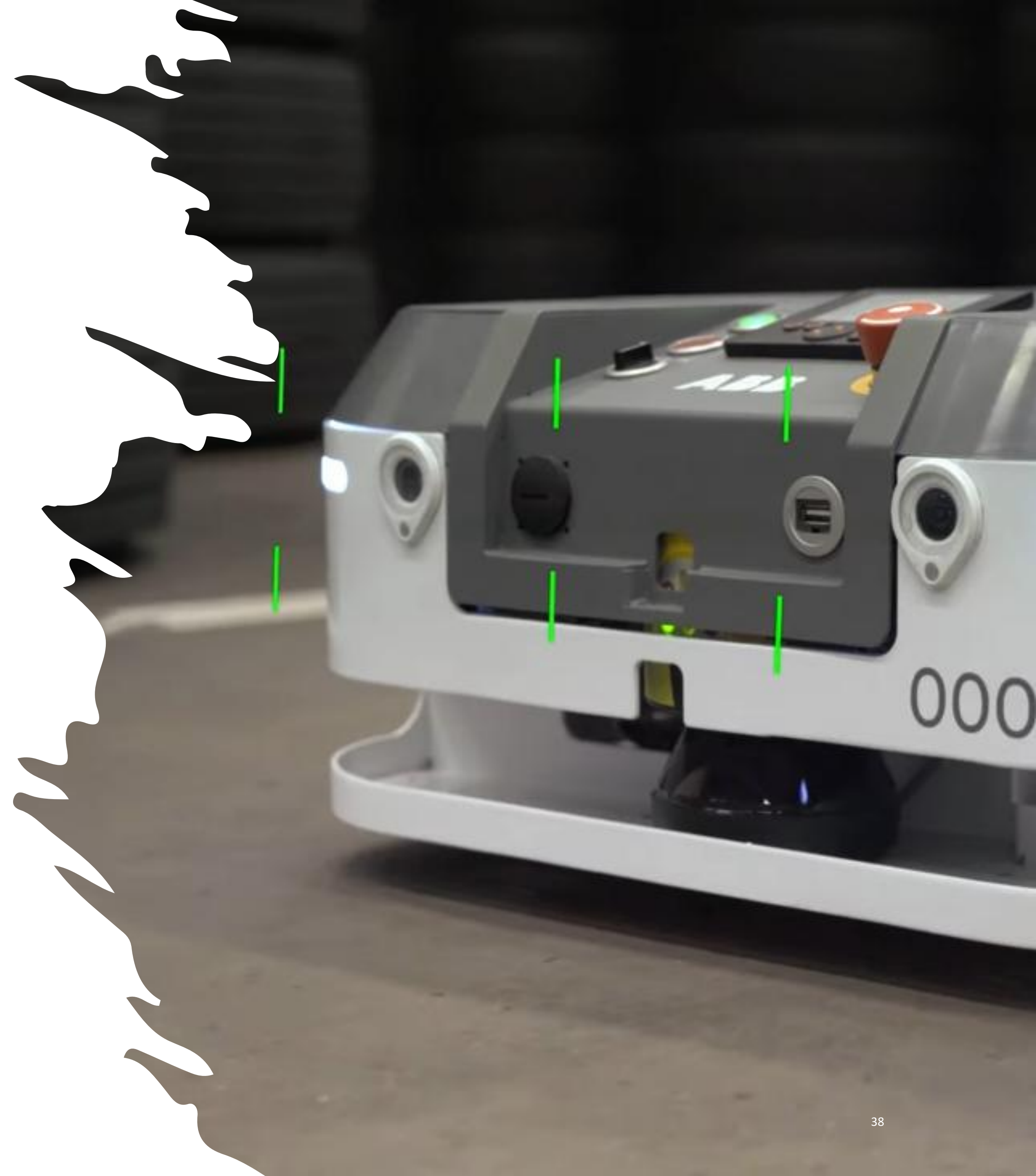
SLAM against localization

Method	Known map	Localization	Mapping
Localization	✓	✓	✗
SLAM	✗	✓	✓

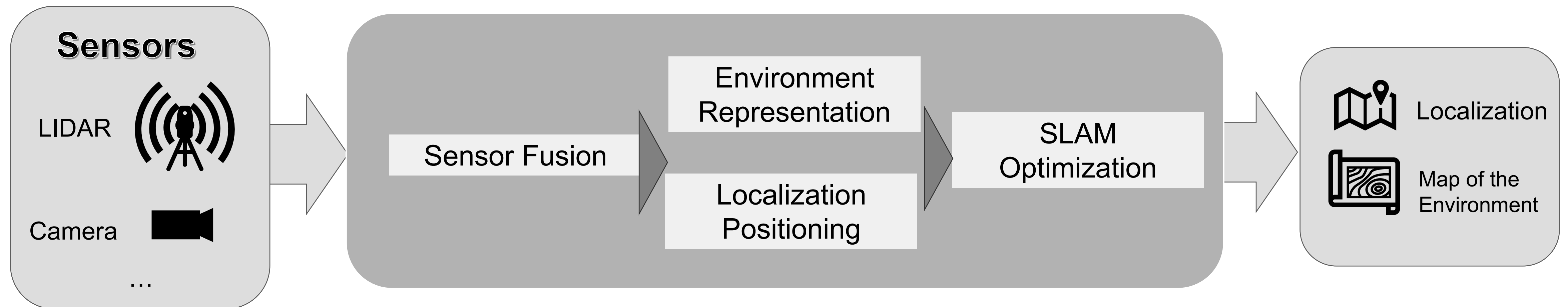
AMR T702

Datacenter robot

<https://www.youtube.com/watch?v=HmHqqa0Ryn8&t=83s>



How SLAM works



SLAM algorithm

Environment Representation

CPS need something to represent the environment to be able to do something

Multiple kinds of representation

Occupancy Grid

2D representation of the environment where each cell can be :

- Occupied
- Free
- Unknown

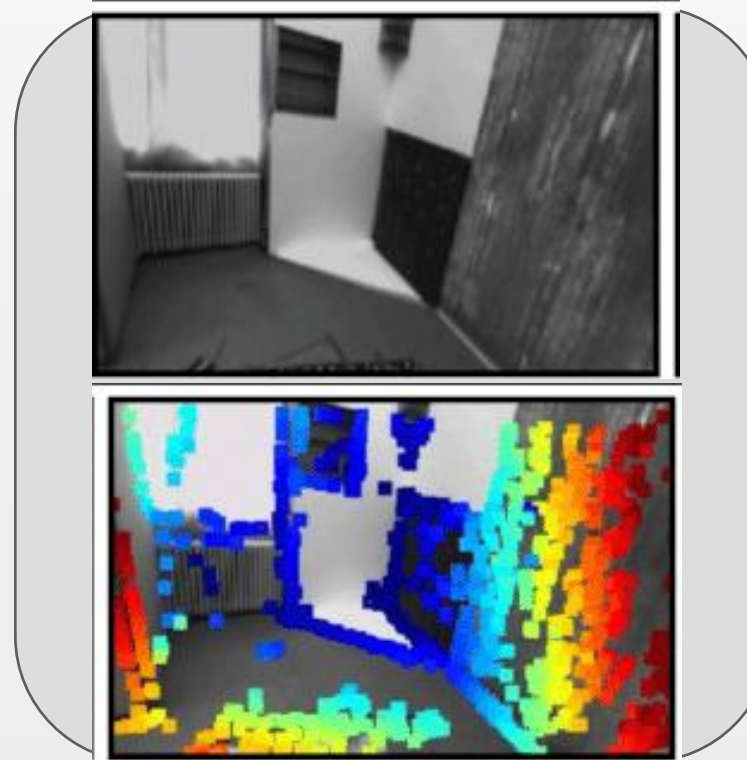


- + Simple
- + Robust
- Heavy memory usage

Environment Representation

Feature based

Uses feature-matching techniques to determine where we are

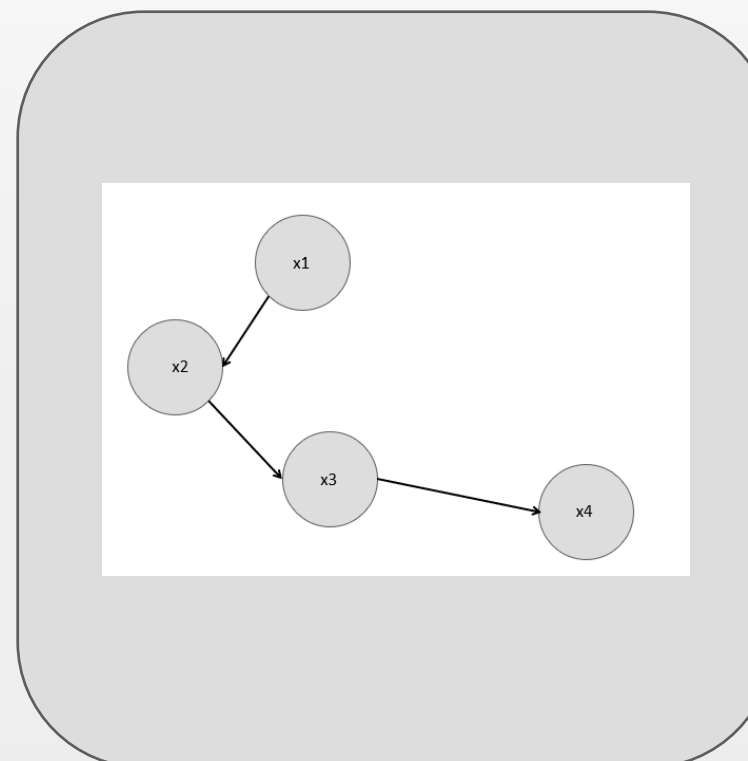


+ Efficient

- Requires a camera
- Feature dependent

Pose graph based

Representation of the position during a time interval considering the constraint during each interval



+ very accurate

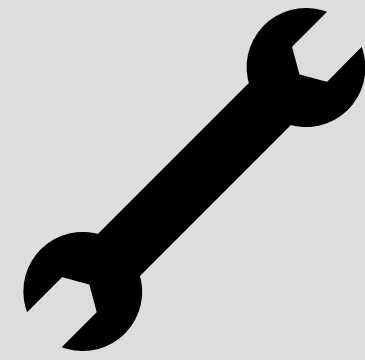
+ Scalable

- Need a closure loop

Comparison between representation

Representation	Dimension	Use for	Pros	Cons
Occupancy Grid	2D/3D	Mapping	Simple, Robust	Memory heavy
Pose Graph	Abstract	Optimization	Accurate, scalable	Need a closure loop
Feature-based	Sparse	Visual SLAM	Efficient	Feature dependent

Useful Tools for mapping



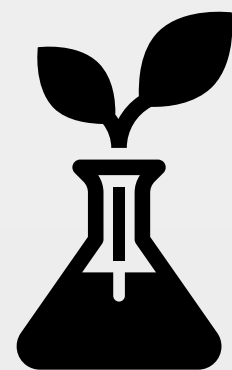
Slam Toolbox



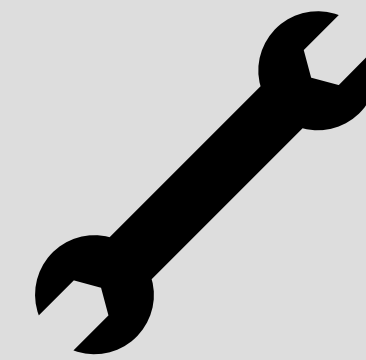
Made by the community



Uses the Occupancy Grid algorithm and a Kalman filter



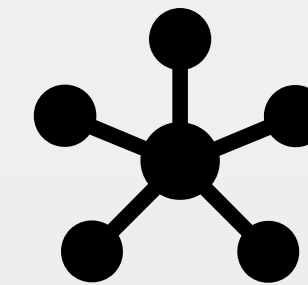
very easy to use
Mostly used for education



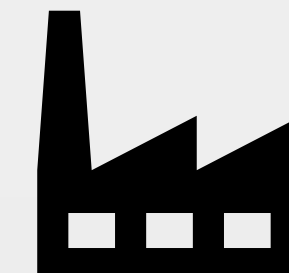
Cartographer



Made by Google



Use Pose graph algorithm
And graph theory



very robust
Mostly use in Industry

SLAM optimization

Sensor drift

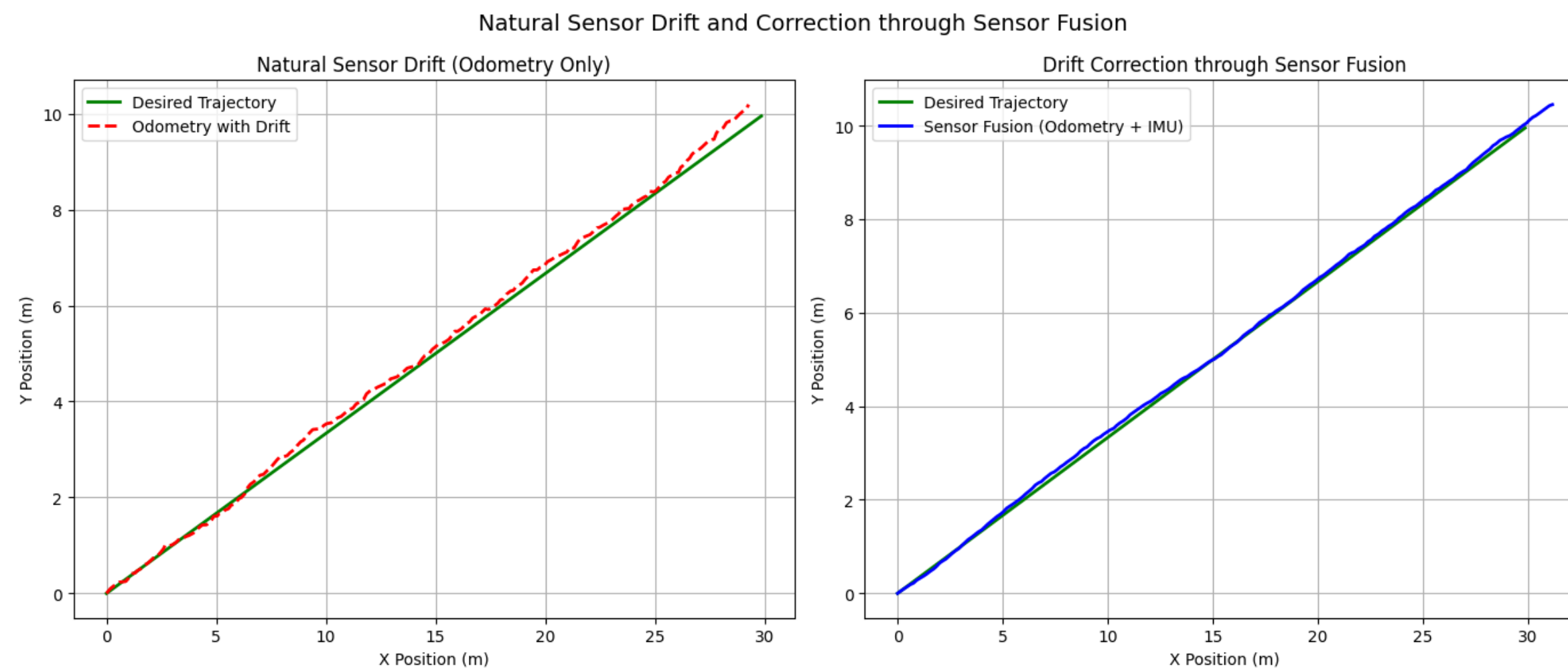


Accumulation of localization error over time due to sensor noise and drift

Drift Prevention

Solution :

- Sensor fusion
- Use Pose Graph optimization
- Adjust the map



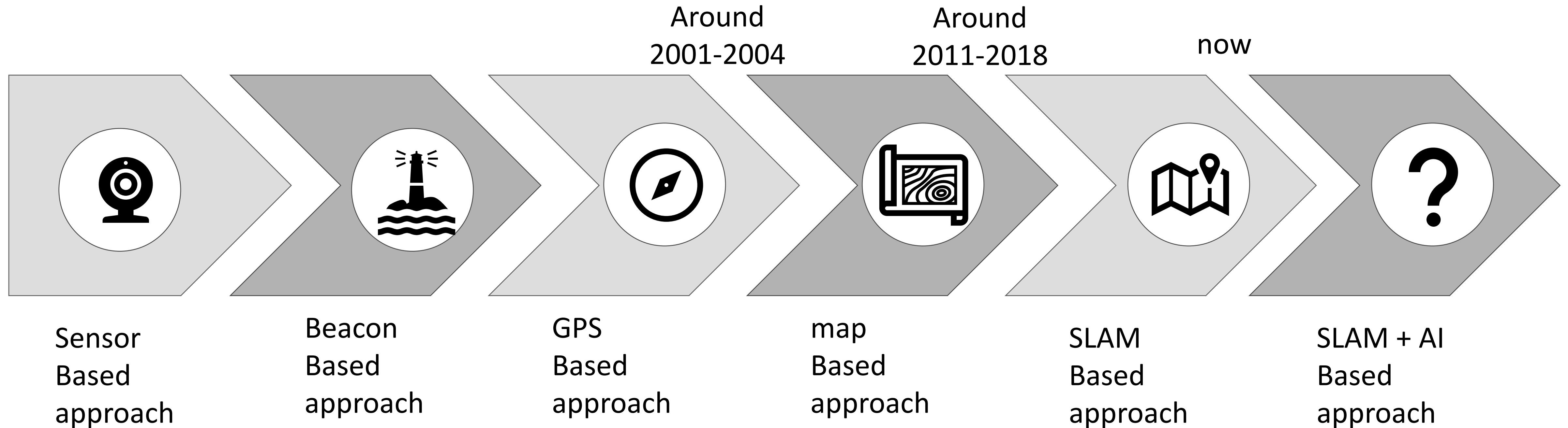
Loop Closure

recognizing previously visited locations and using this information to correct the accumulated drift.

Solution :

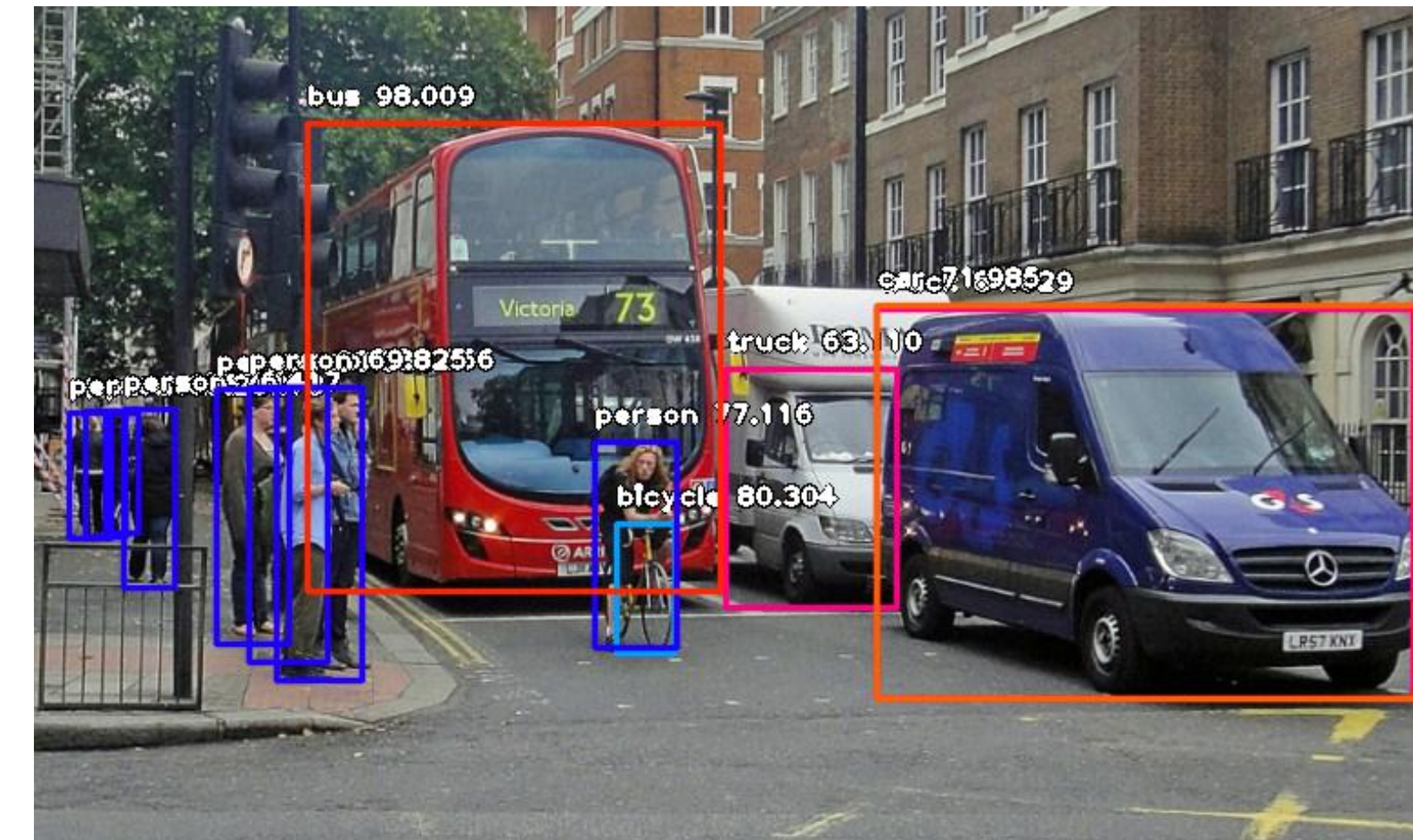
- Feature Matching: SIFT, SURF, ORB.
- Bag of Words

Full timeline of localization



AI Based approach for localization and mapping

AI and feature recognition become more and more important in robotic



AI also permit the robots to adapt himself more easily to unexpected scenario

Research Directions:
Cyber-Physical Systems

www.cambridge.org/cbp

Impact Paper

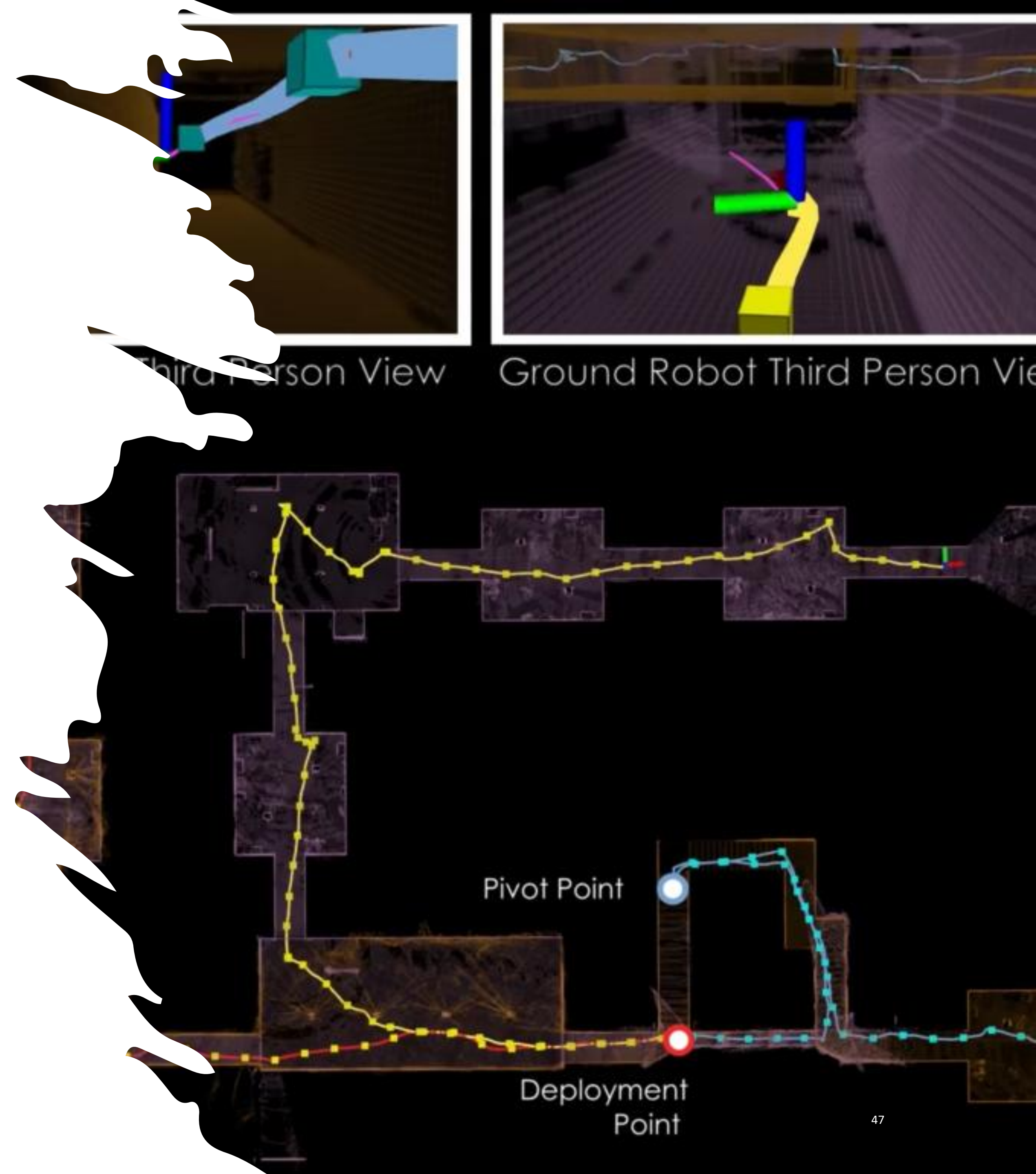
Robotic safe adaptation in unprecedented situations: the RoboSAPIENS project

Peter G. Larsen¹, Shaukat Ali², Roland Behrens³, Ana Cavalcanti⁴, Claudio Gomes¹,
Guoyuan Li⁵, Paul De Meulenaere⁶, Mikkel L. Olsen⁷, Nikolaos Passalis⁸,
Thomas Peyrucain⁹, Jesús Tapia¹⁰, Anastasios Tefas⁸ and Houxiang Zhang⁵



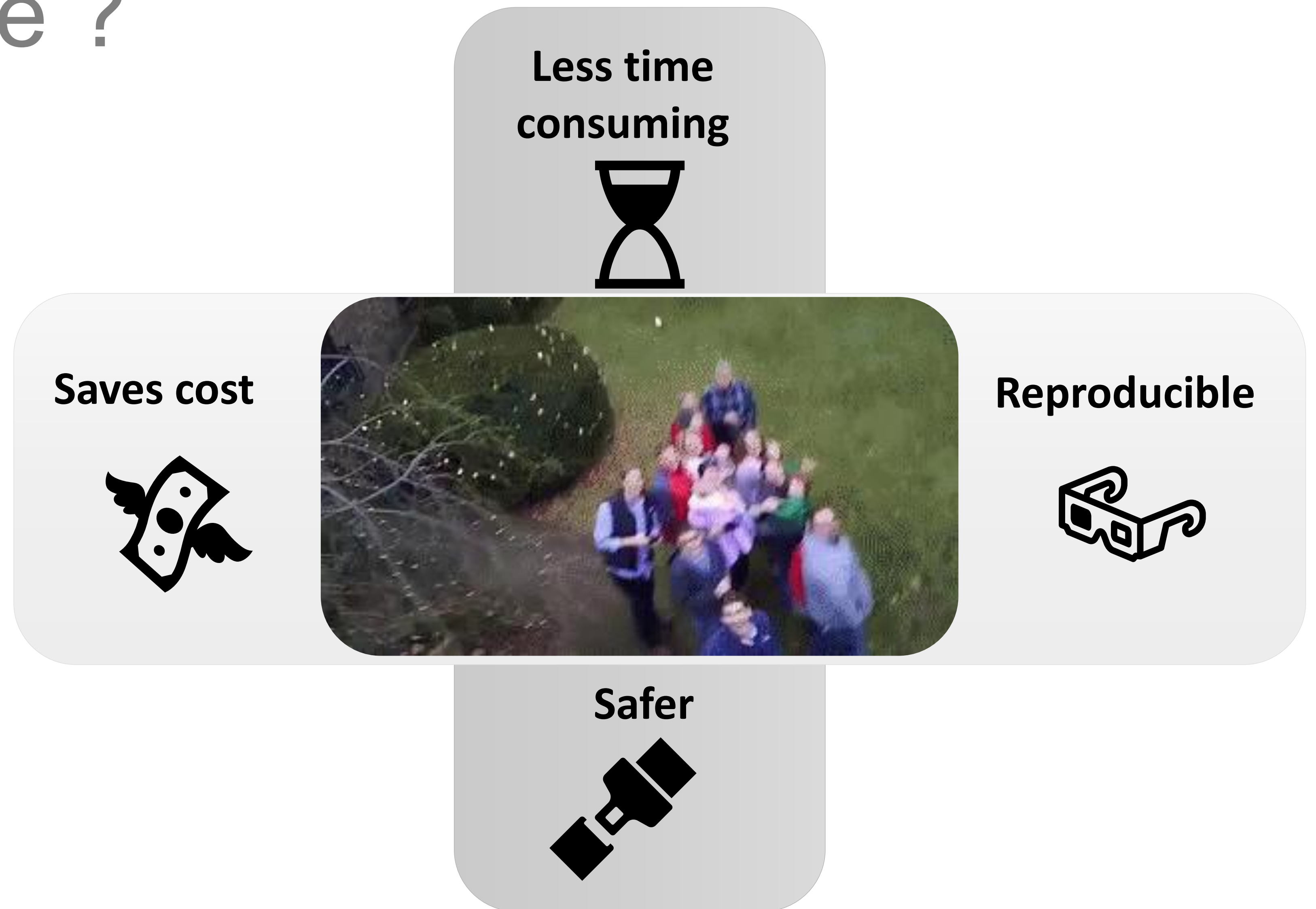
Multiple drone collaborative mapping

<https://www.youtube.com/watch?v=VEYS5BjmZP8>



Simulation environment

Why simulate ?



Practical case
where
simulation
could avoid the
problem

SWI swissinfo.ch

The Swiss voice in the world since 1935

GEOPOLITICS DEMOCRACY SCIENCE SWISS IDENTITY ECONOMY SWISS ABROAD

News >

swissinfo.ch

The Swiss voice in the world since 1935

CS DEMOCRACY SCIENCE SWISS IDENTITY ECONOMY SWISS ABROAD

Swiss Post refines safety processes after drone crash



Swiss Post's drone for the delivery of parcels, as presented in 2017 Ti-press

June 28, 2019 - 15:17 2 minut

Keystone-SDA/Swiss Post/swissinfo.ch/filj

An interim report into the most recent Swiss Post drone crash recommended improvements to the parachute and warning system. Swiss Post has announced that it has already taken measures to improve drone safety.

A drone belonging to a Swiss Post medical deliveries programme crashed on May 9 in a Zurich forest – the second crash in four months. It was not carrying any supplies but the national postal service has decided to suspend all drone flights.

+ Read more about the May crash here

Swiss Post drone crashes in Zurich – again



A drone belonging to a Swiss Post medical deliveries programme crashed on Thursday morning in a forest – the second crash in four months. It was not carrying any supplies but the national postal service has decided to suspend all drone flights.

Player project

1998-2010

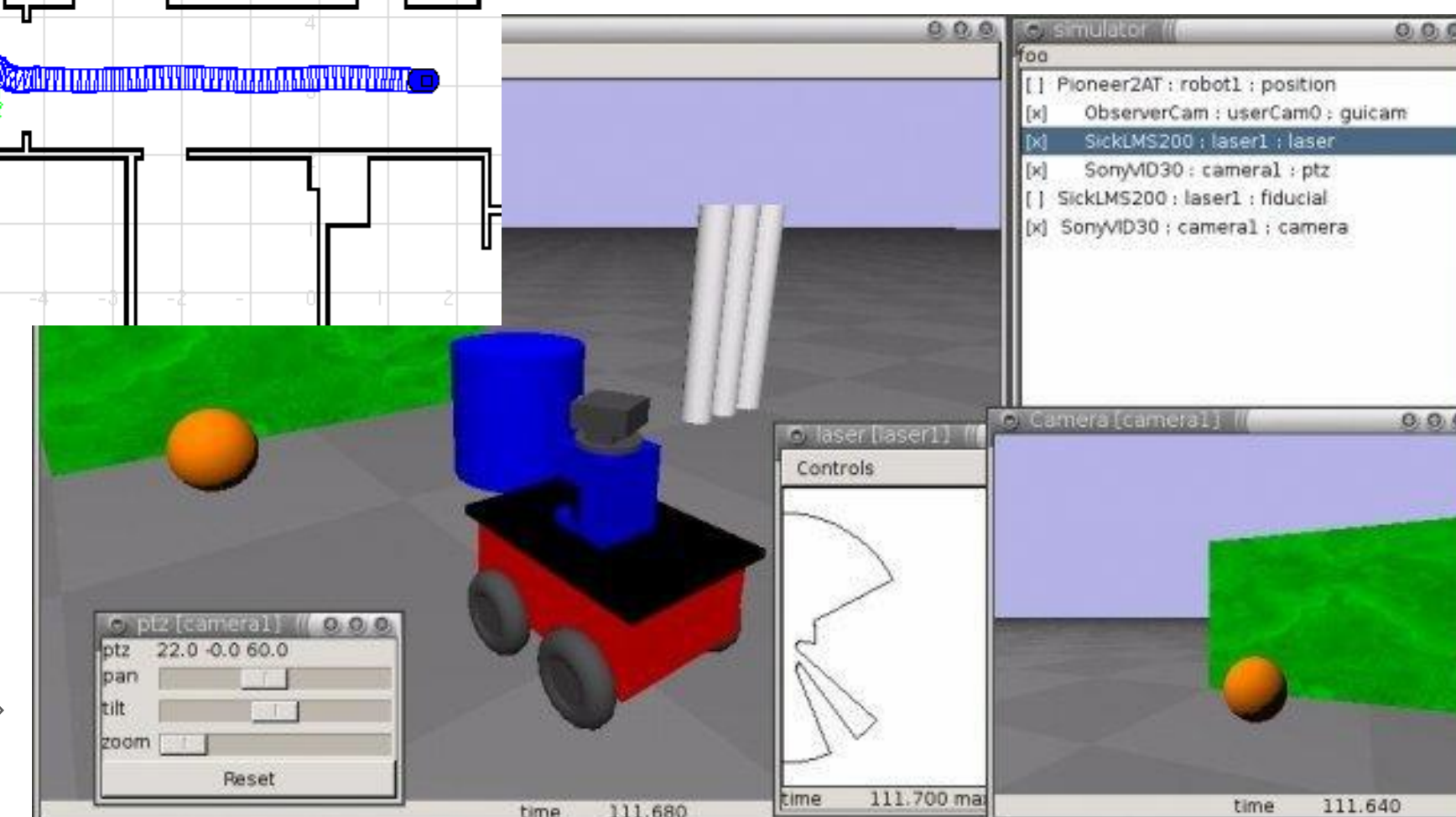
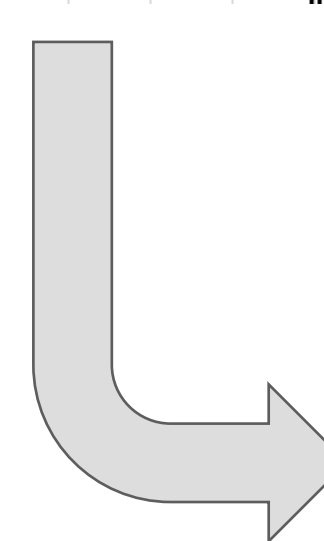
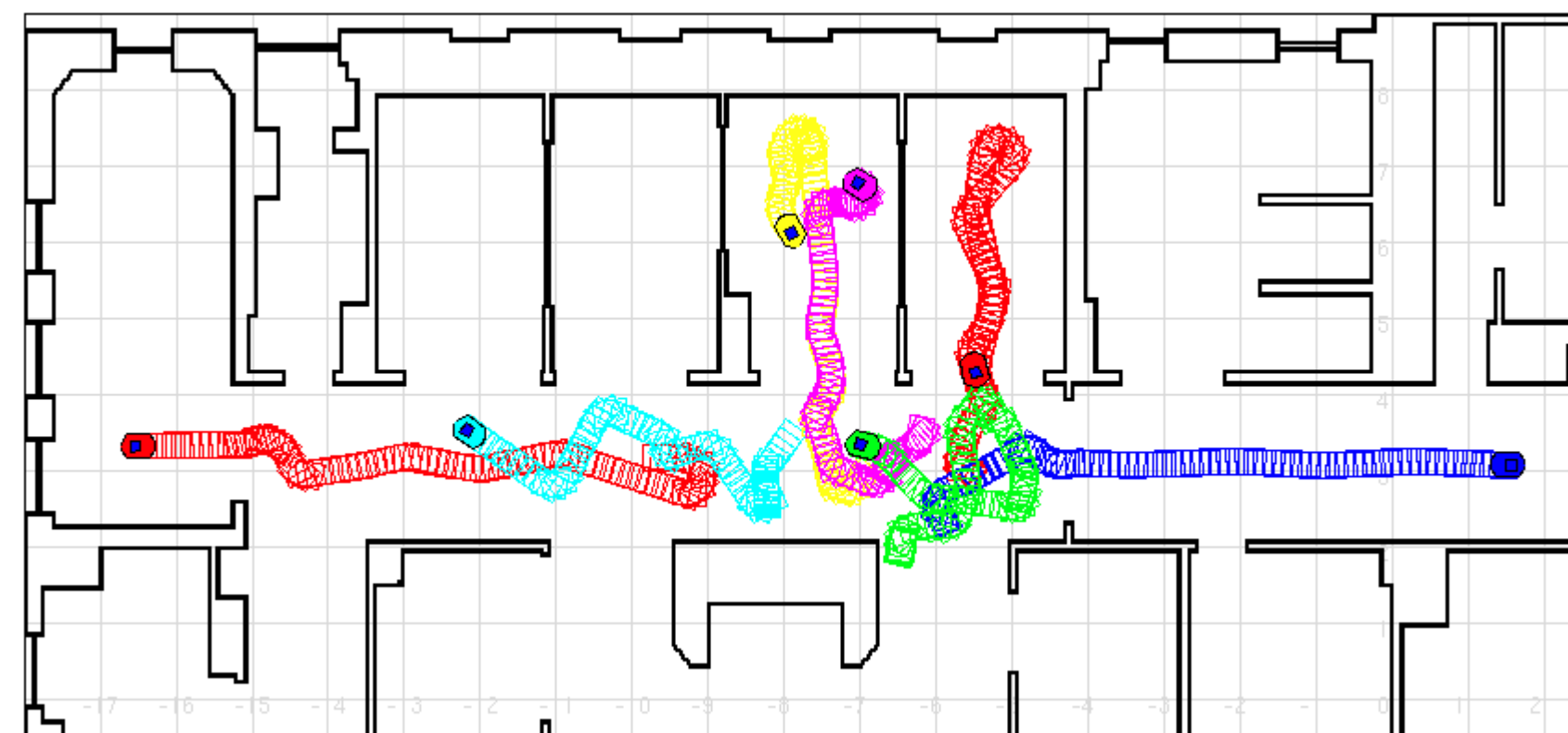
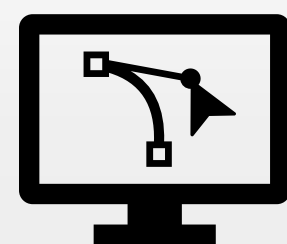


<https://playerstage.sourceforge.net>

One of the first simulator tool
to make simulation in robotics



Was in his first version using
2D representation



In his latest version introduced the
3D visualization but was too late

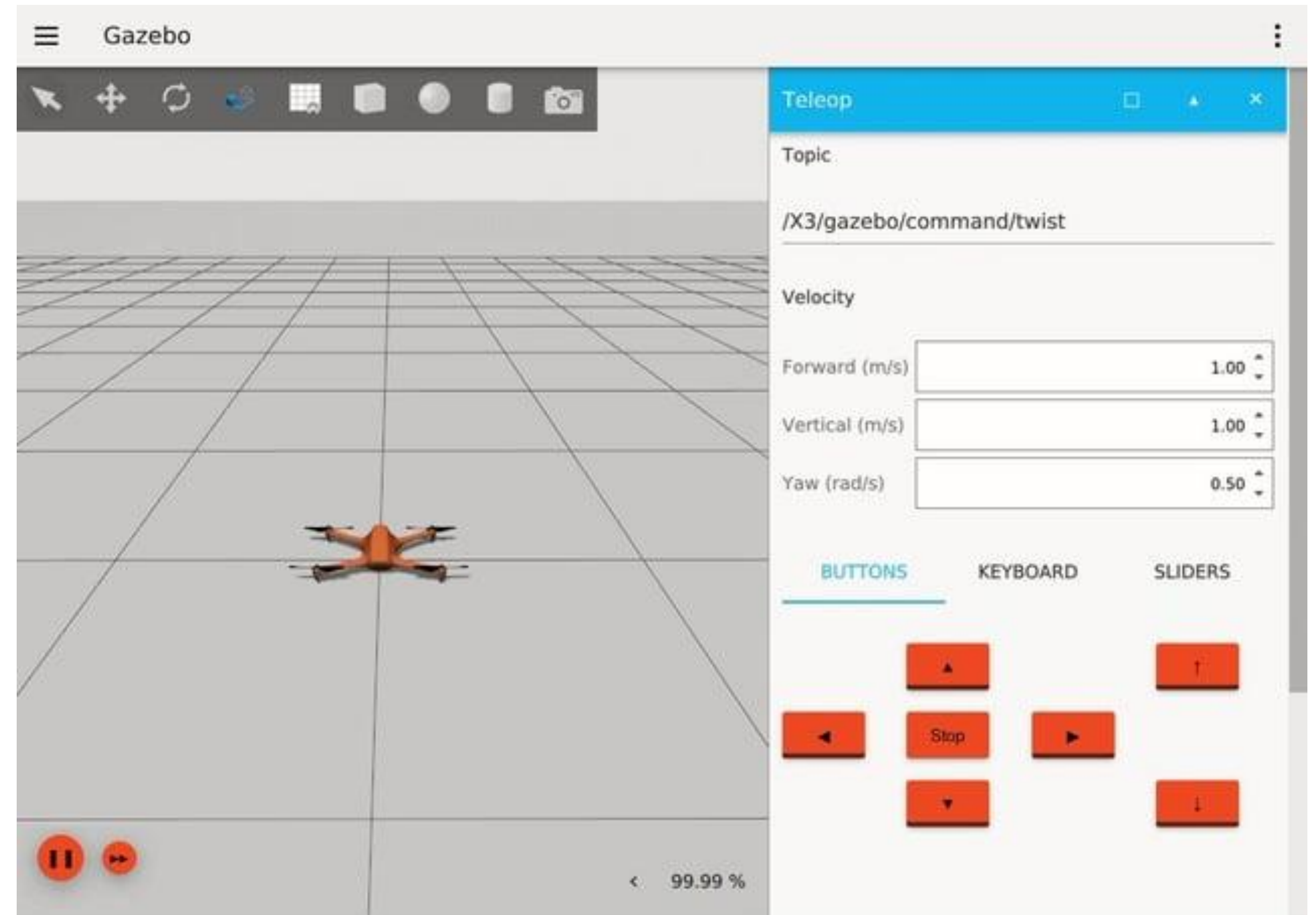


Gazebo

2004-2010



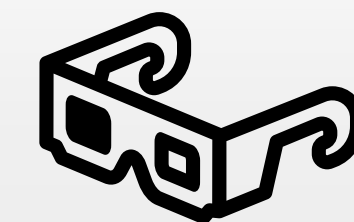
<https://gazebosim.org/docs/fortress/install/>



The most used simulation software in research
and industry



Gave realistic environment and sensors and use 3D
visualization



The concept behind Gazebo

Born by the same developer than ROS



Willow Garage, creator of ROS and Gazebo

“Could we actually create something that would enable people to build on each other's results rather than continuing this cycle of 90 percent duplicating what someone else has already done, with a little bit at the end of something new, if you're lucky.”

10 anniversary of ROS speech

Before the replication of result were very hard

Everyone had his own proprietary software

In its different language

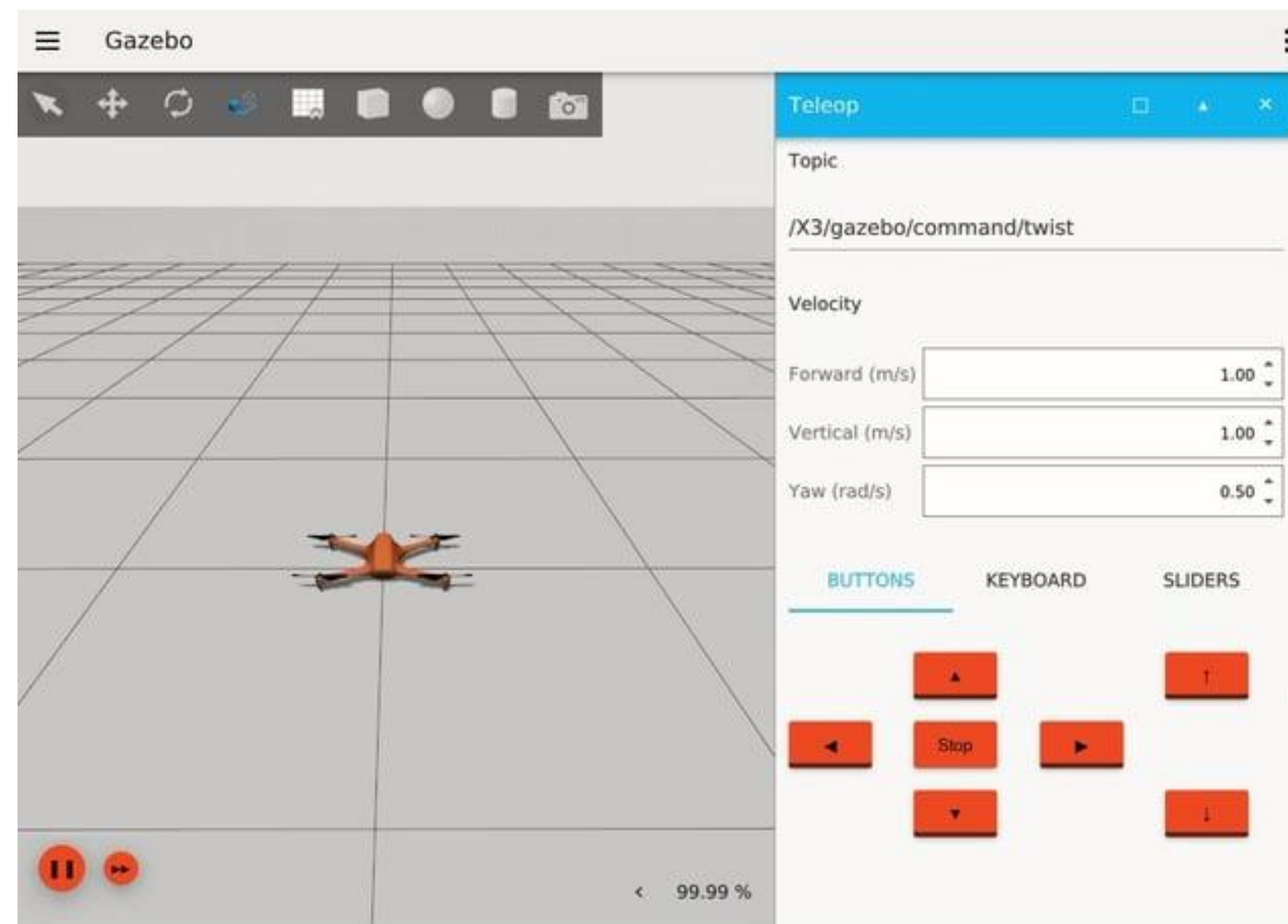
The open source one use mostly 2D in their simulation tools

The simulator where not connected to the middleware

By responding to this problems, it became the norm use everywhere

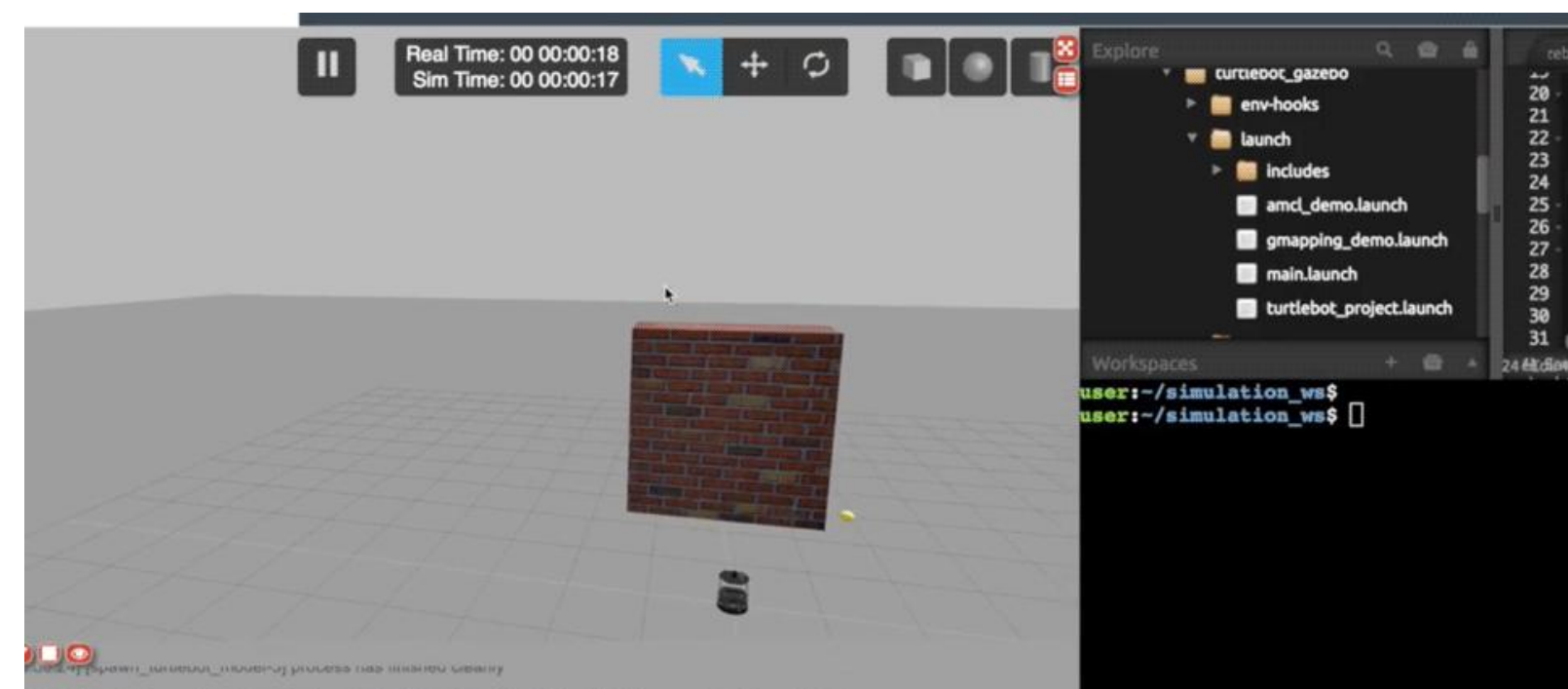
Gazebo

2004-2010



New recent version
With the engine
reworked

Still the world around the robot where not
realist enough to do computer vision on robots



Photorealism software

2015-now

Main idea :

Use gaming creation software to be able to generate more realistic environment to test robots

+

It permitted the rise of computer vision in robotics

-

The physics and algorithm inside the robot became more approximate in simulation environments



Realistic physics $\neq$ realistic behavior

And with the AI rising it became a problem in robotics

AI software

2021-now

Robots now need to learn

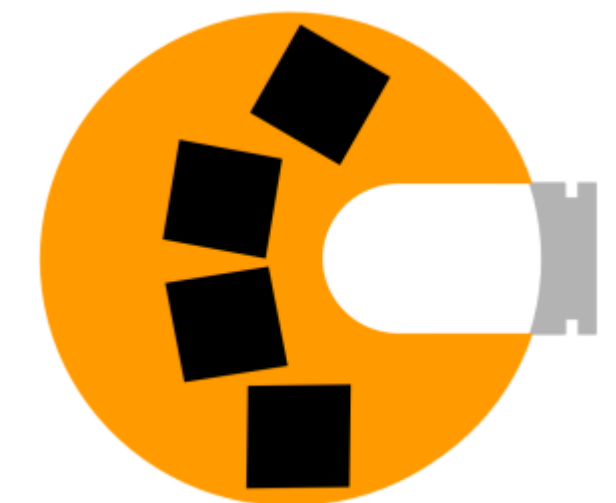
Requires thousands of iterations

Realistic sensor noise

Large-scale data generation

Fast computation on GPUs but need specific hardware

Isaac Sim



Py bullet

Software tools comparison

Simulator	Developer	Physics Engine	Typical Use	Strengths	Limitations
Gazebo (Classic / Gazebo Sim)	Open Source / Open Robotics	ODE, Bullet, DART, Simbody	ROS / ROS2 simulation	Native ROS integration, large community, free & open source	Graphics less realistic, limited photorealism
Isaac Sim	NVIDIA	PhysX (GPU-accelerated)	AI, RL, perception, synthetic data	Photorealistic rendering, GPU acceleration, domain randomization	High hardware requirements, NVIDIA ecosystem
Webots	Cyberbotics	ODE	Education, research	Easy to use, good documentation, fast setup	Less scalable for large environments
CoppeliaSim (V-REP)	Coppelia Robotics	Bullet, ODE, Vortex	Prototyping, manipulation	Very flexible, multi-physics engines	UI complexity, less ROS-native
MORSE	Open Source	Blender Game Engine	Academic research	High-level abstraction, scripting	No longer actively developed
PyBullet	Open Source (Google)	Bullet	RL, control, research	Lightweight, Python-friendly	Limited visualization
Unity + Robotics Hub	Unity Technologies	Unity Physics	Simulation + visualization	High-quality graphics, large ecosystem	Physics less accurate for robotics
AirSim	Microsoft	Unreal Engine	Autonomous vehicles, drones	Realistic environments, sensors	Focused on aerial/vehicle robotics
Stage	Open Source	2D physics	Multi-robot simulation	Fast, lightweight	2D only, very simple

Ros in industry, avidbots example

https://www.youtube.com/watch?v=WmWW_yMupSs



Why we use Gazebo

Created by the same developers as ROS



Willow Garage, creator of ROS and Gazebo

“Could we actually create something that would enable people to build on each other's results rather than continuing this cycle of 90 percent duplicating what someone else has already done, with a little bit at the end of something new, if you're lucky.”

10 anniversary of ROS speech

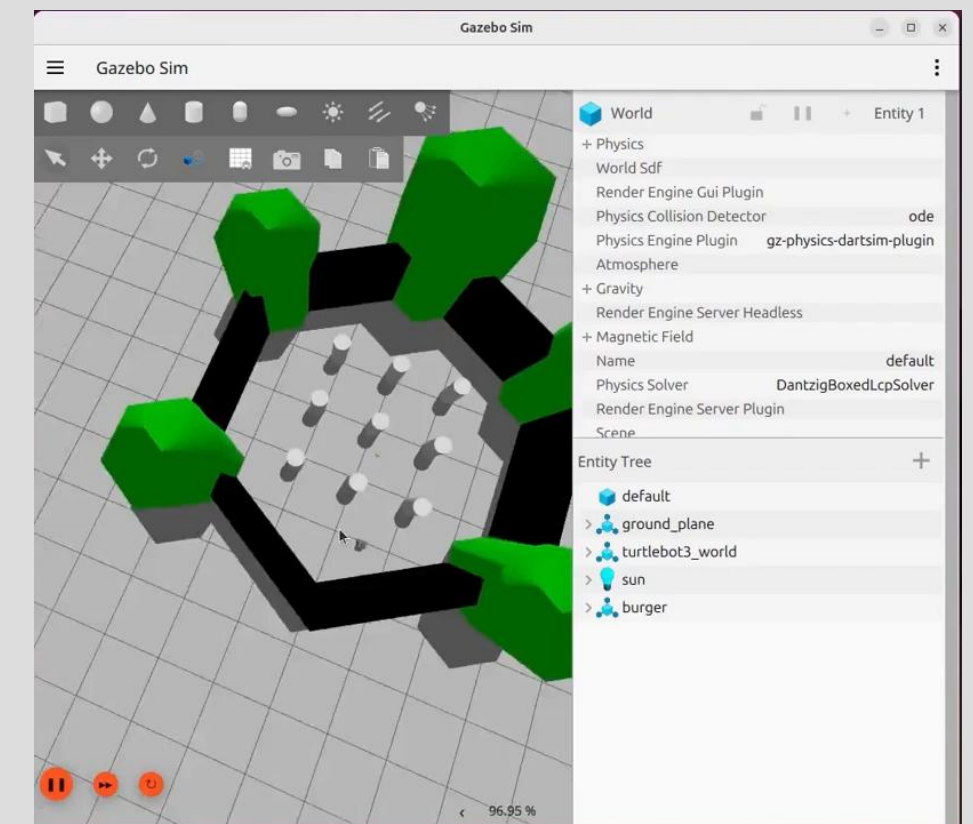
Native ROS integration

Physics-first simulation

Realistic sensor modeling

Very Modular

Enables reproducible robotics experiments



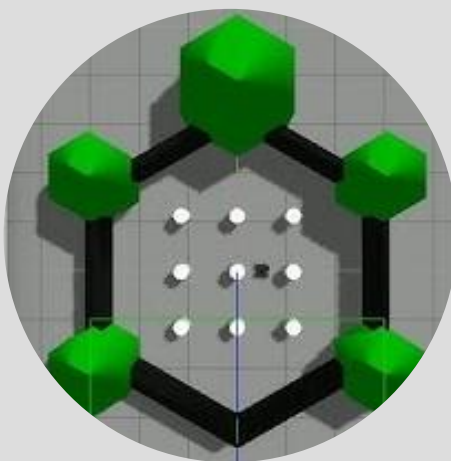
See Gazebo video
<https://youtu.be/dM1EGtEgOHs>

Why Gazebo is modular ?

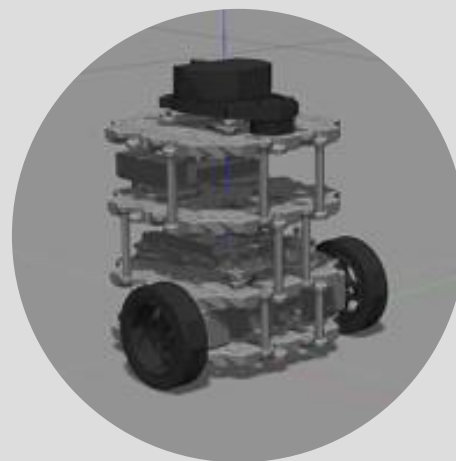
Use three main models :

Worlds

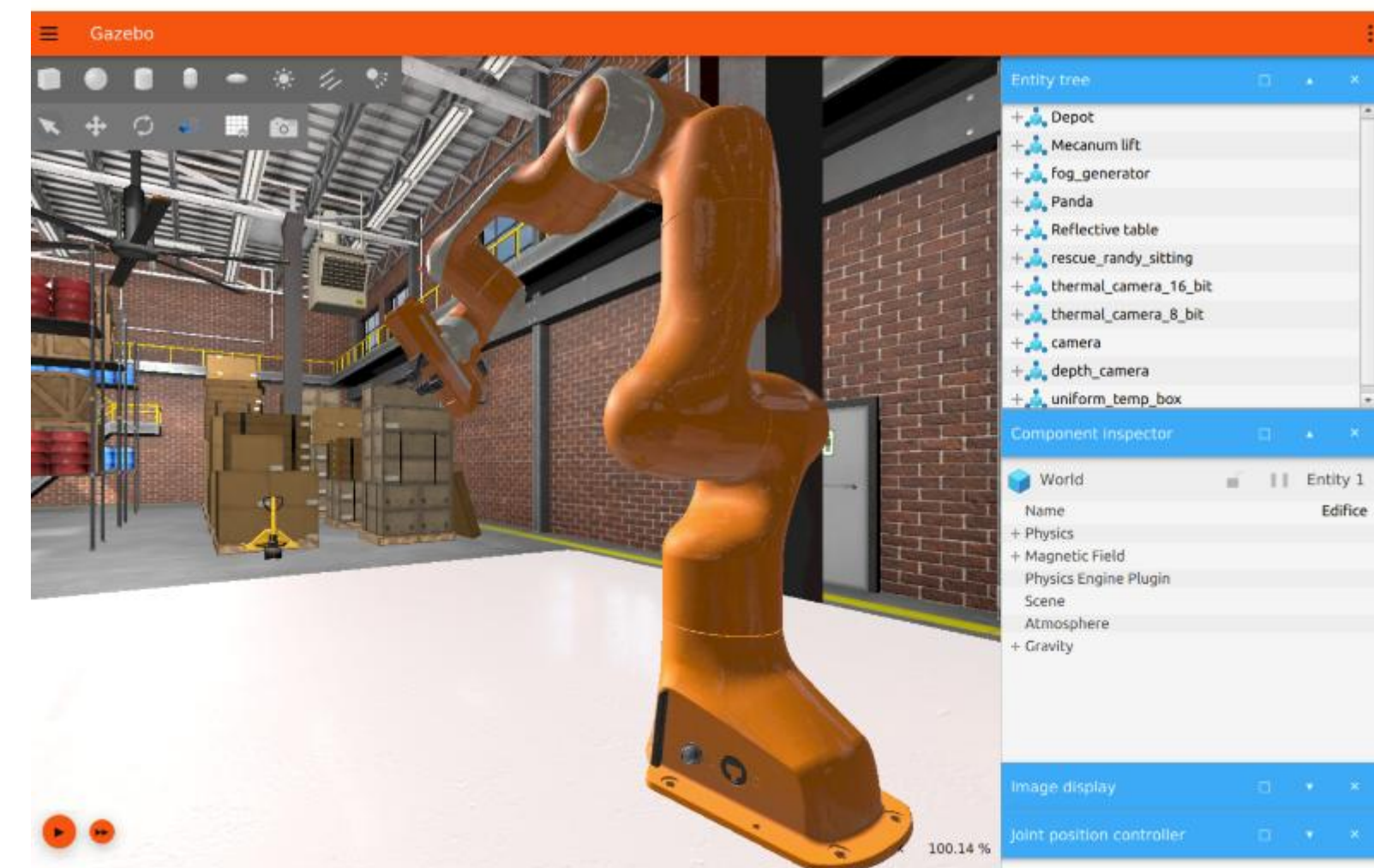
Environment



Models



Plugins



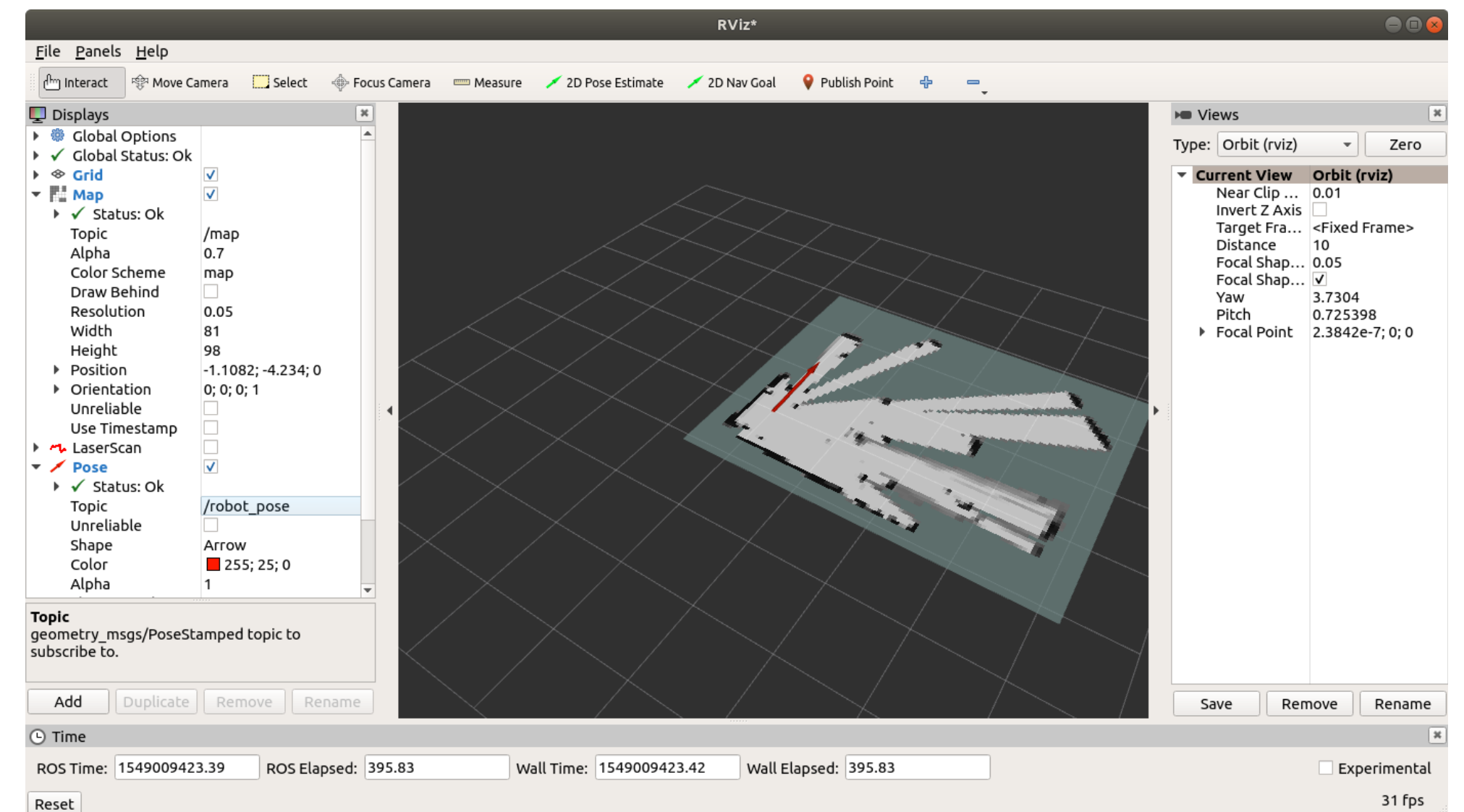
Each of these elements uses the SDF file format to describe correctly the visual, size,...
The physic, behavior and packages are coded with C++,Python and C language

How to investigate the sensor Data

In ROS2 the main choice is to use Rviz

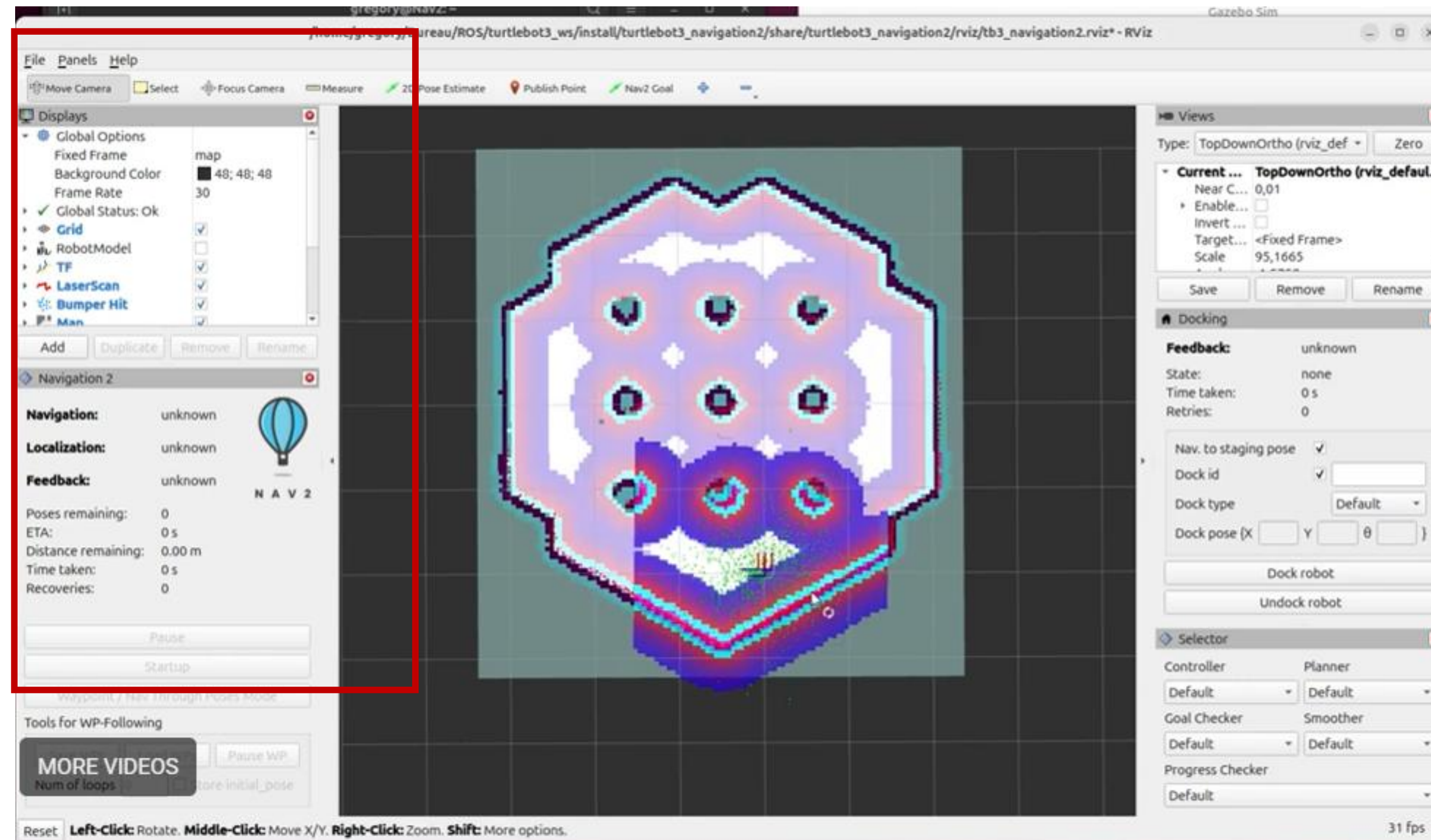
It places us inside the robot's point of view.

It's a debug tool to see why the robot does not work



Rviz

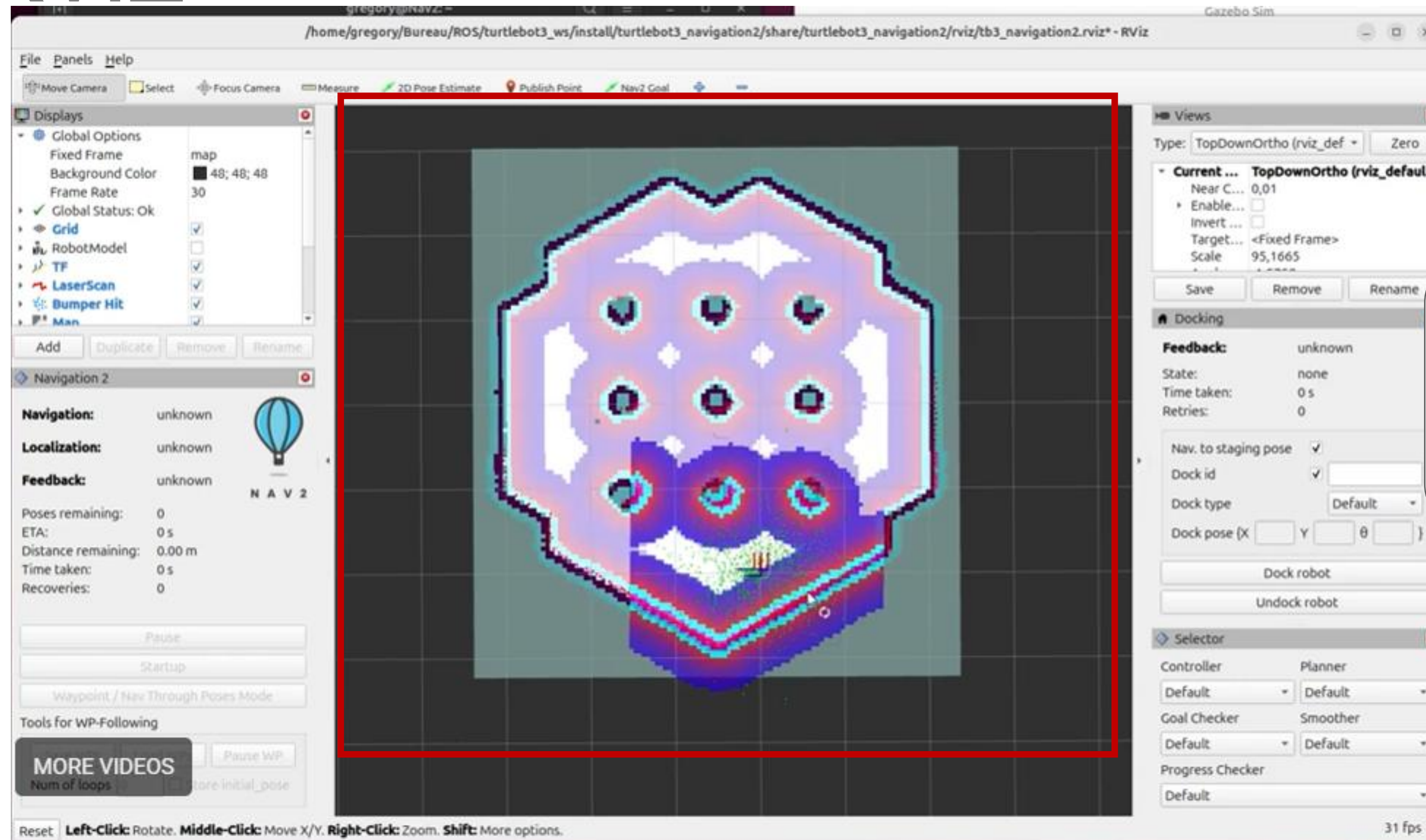
TF
three



Shows the robot's nodes to clearly understand the localization

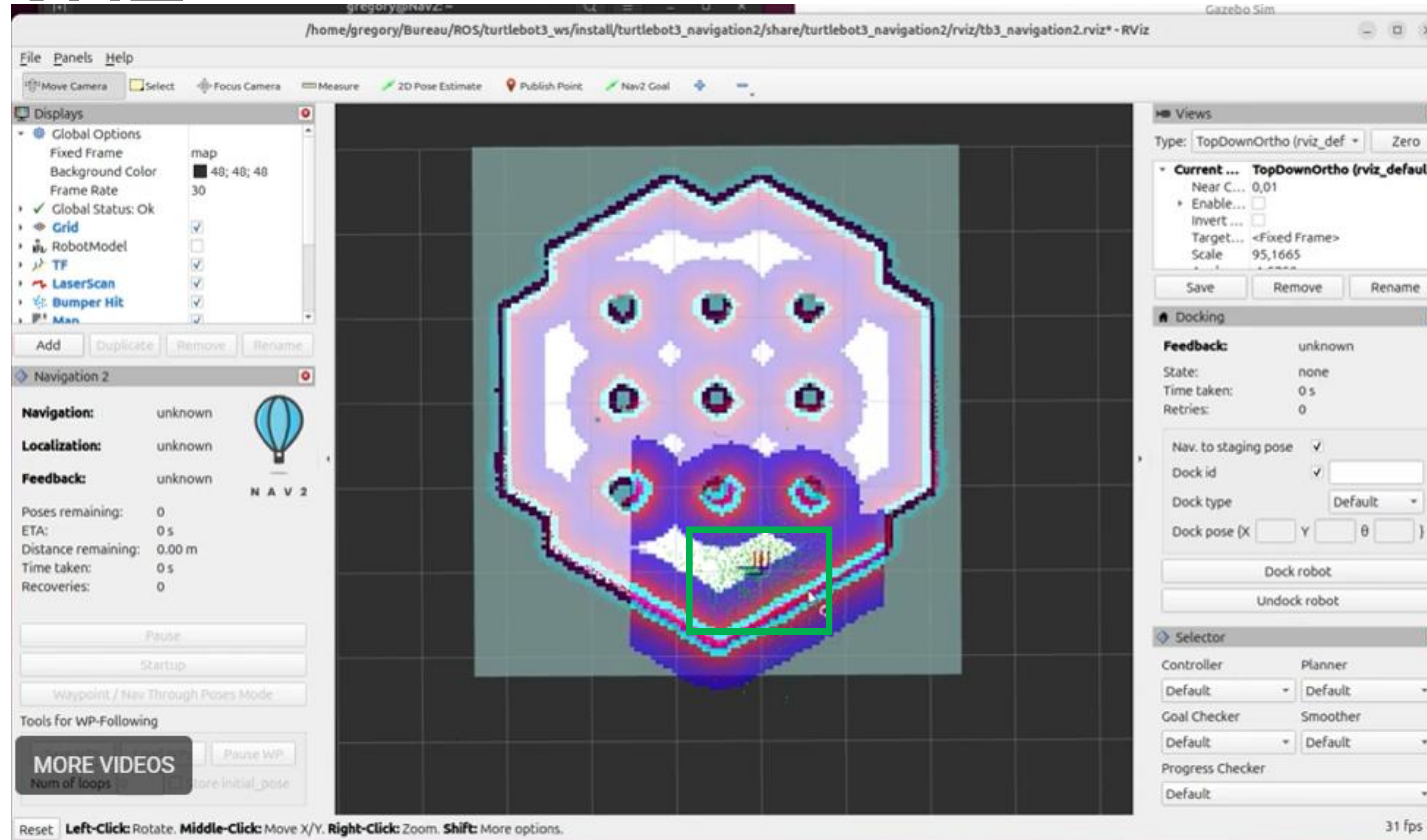
Rviz

The
maps



With the occupancy grid and
the obstacles

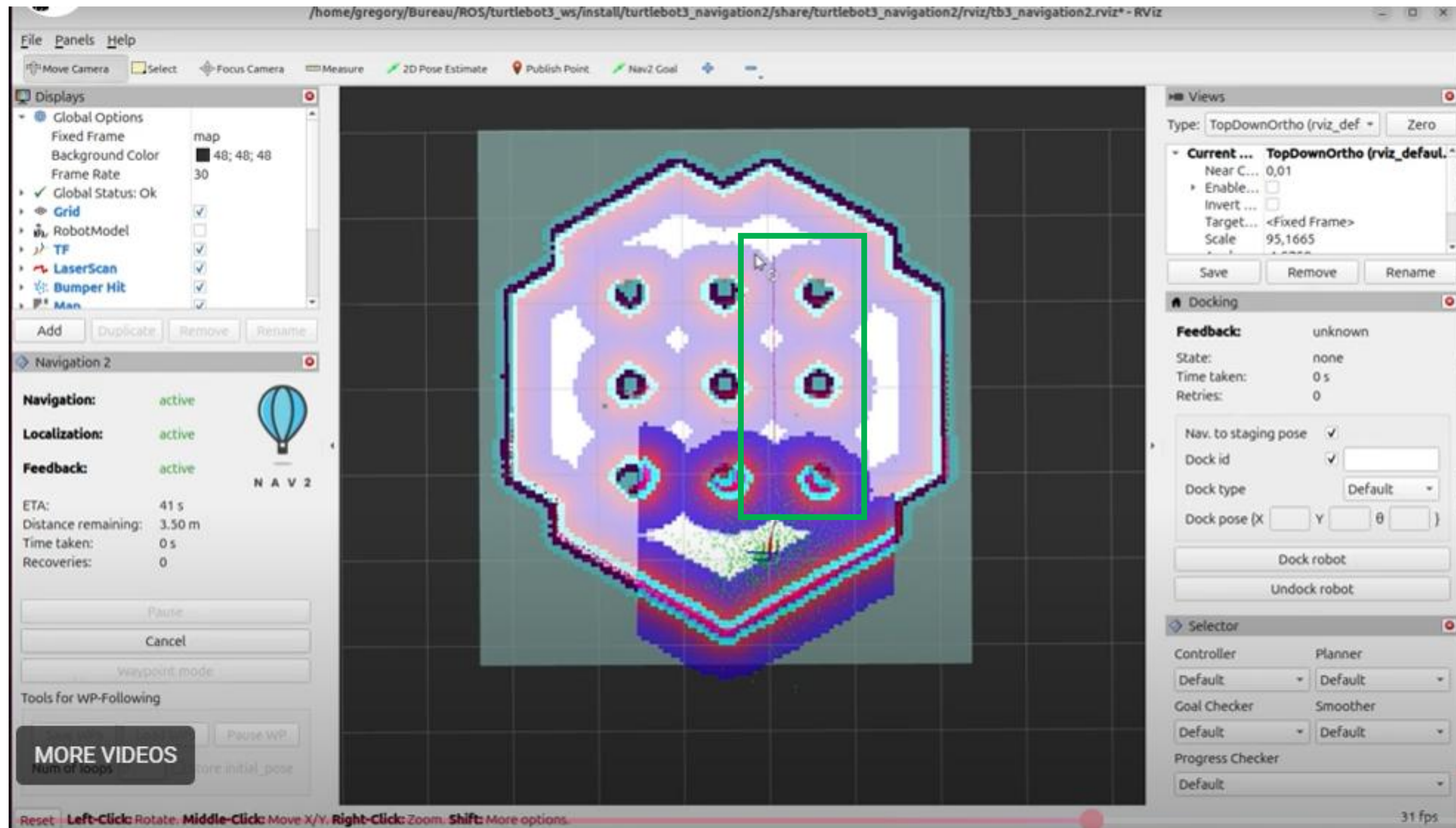
Rviz



The robot
models

Where we can see the axis of
the robots and the articulation
of the model

Rviz



To see the travel that the simulation estimate for going to the desired location

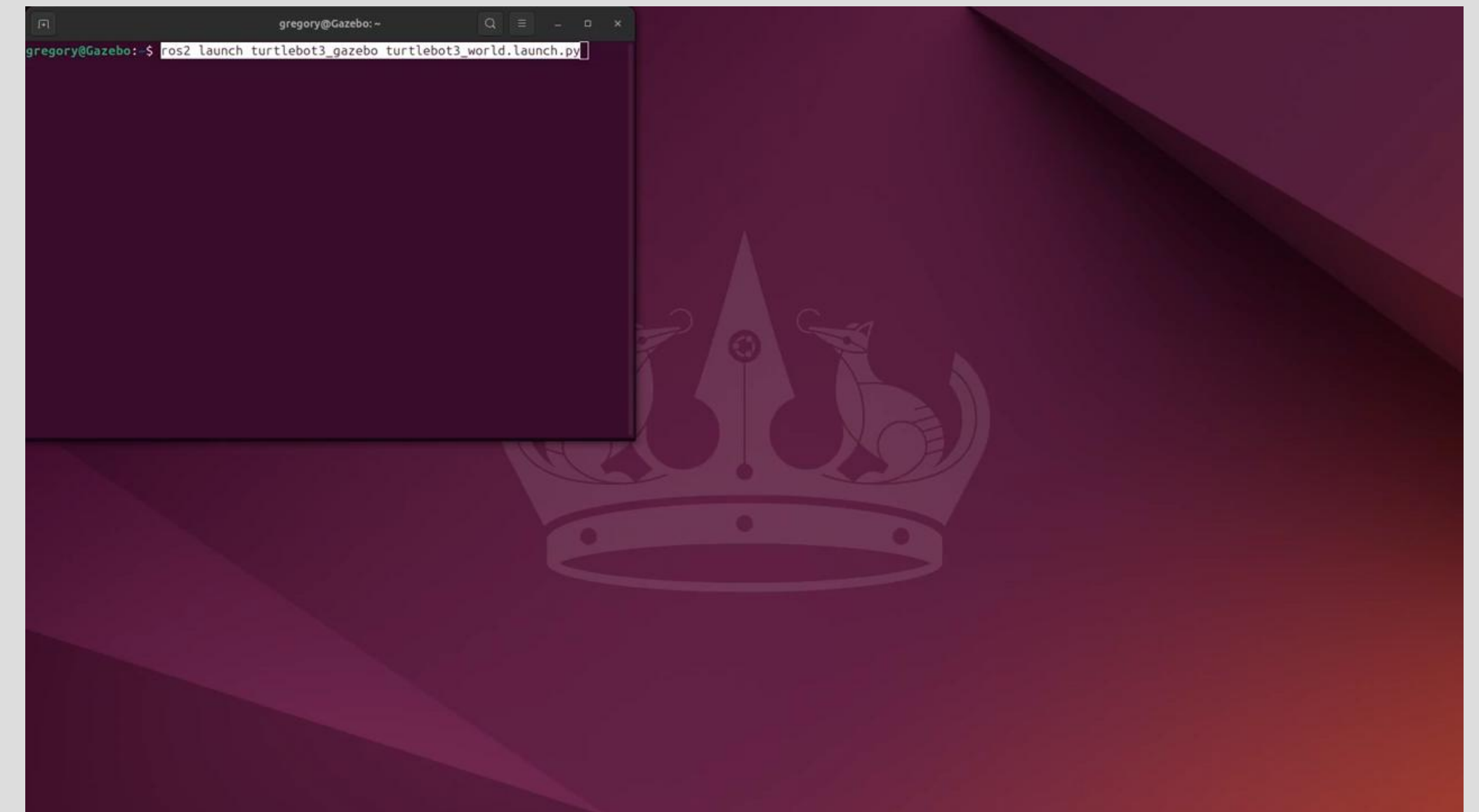
The path to go to the waypoint

Full Demonstration

Command on the demo

```
ros2 launch turtlebot3_gazebo turtlebot3_world.launch.py  
ros2 launch turtlebot3_cartographer cartographer.launch.py  
use_sim_time:=True ros2  
run turtlebot3_teleop teleop_keyboard  
ros2 run nav2_map_server map_saver_cli -f ~/map
```

Map creation tutorial :



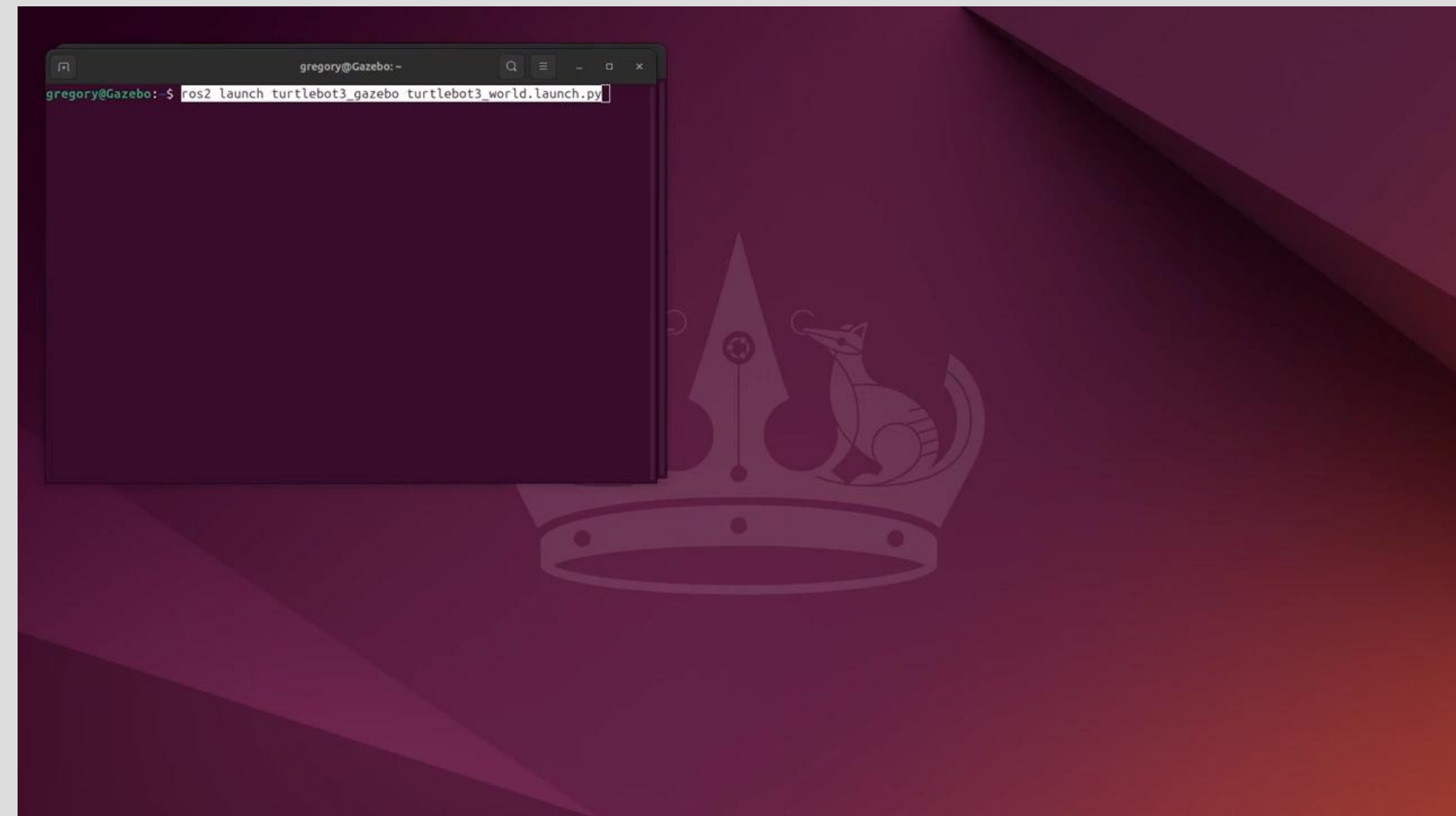
See Mapping full video
<https://youtu.be/Eem8vaHdsol>

Full Demonstration

Command on the demo

```
ros2 launch turtlebot3_gazebo turtlebot3_world.launch.py  
ros2 launch turtlebot3_navigation2 navigation2.launch.py  
use_sim_time:=True map:=$HOME/map.yaml
```

Autonomous tutorial



See [Autonomous full video](https://youtu.be/1jcSVPTf9cY)
<https://youtu.be/1jcSVPTf9cY>

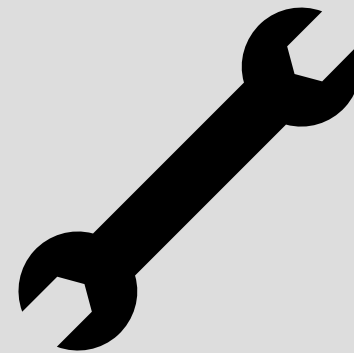
What was the lecture about

How SLAM works



- Environment
- Localization
- Correction

Tool to use SLAM on ROS2



- Gazebo
- Rviz
- Tf Tree

The limits of SLAM



Sensor Drifting
Needs high computational power