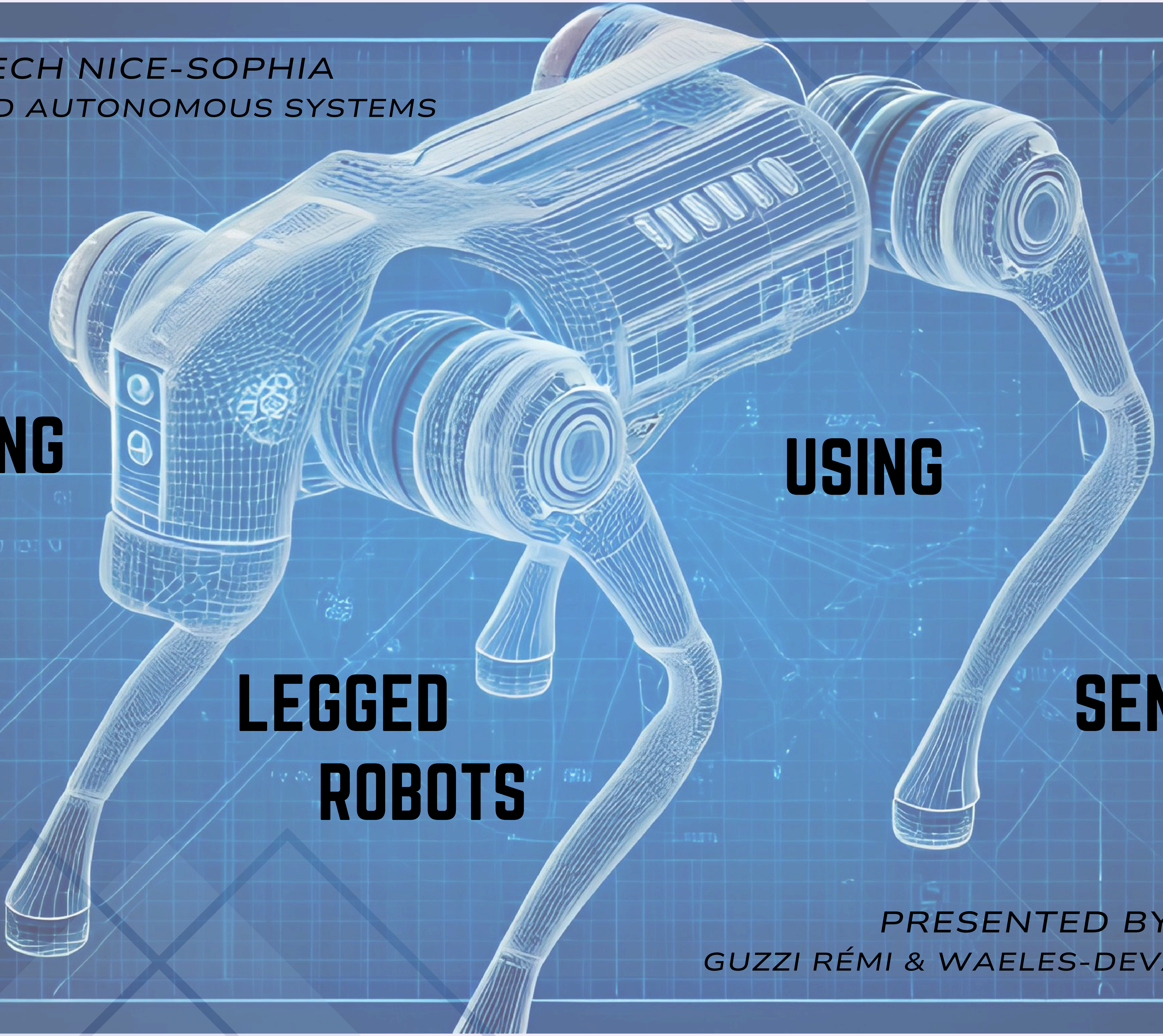
POLYTECH NICE-SOPHIA
ROBOTICS AND AUTONOMOUS SYSTEMS

LOCALIZING LEGGED ROBOTS USING SENSOR FUSION



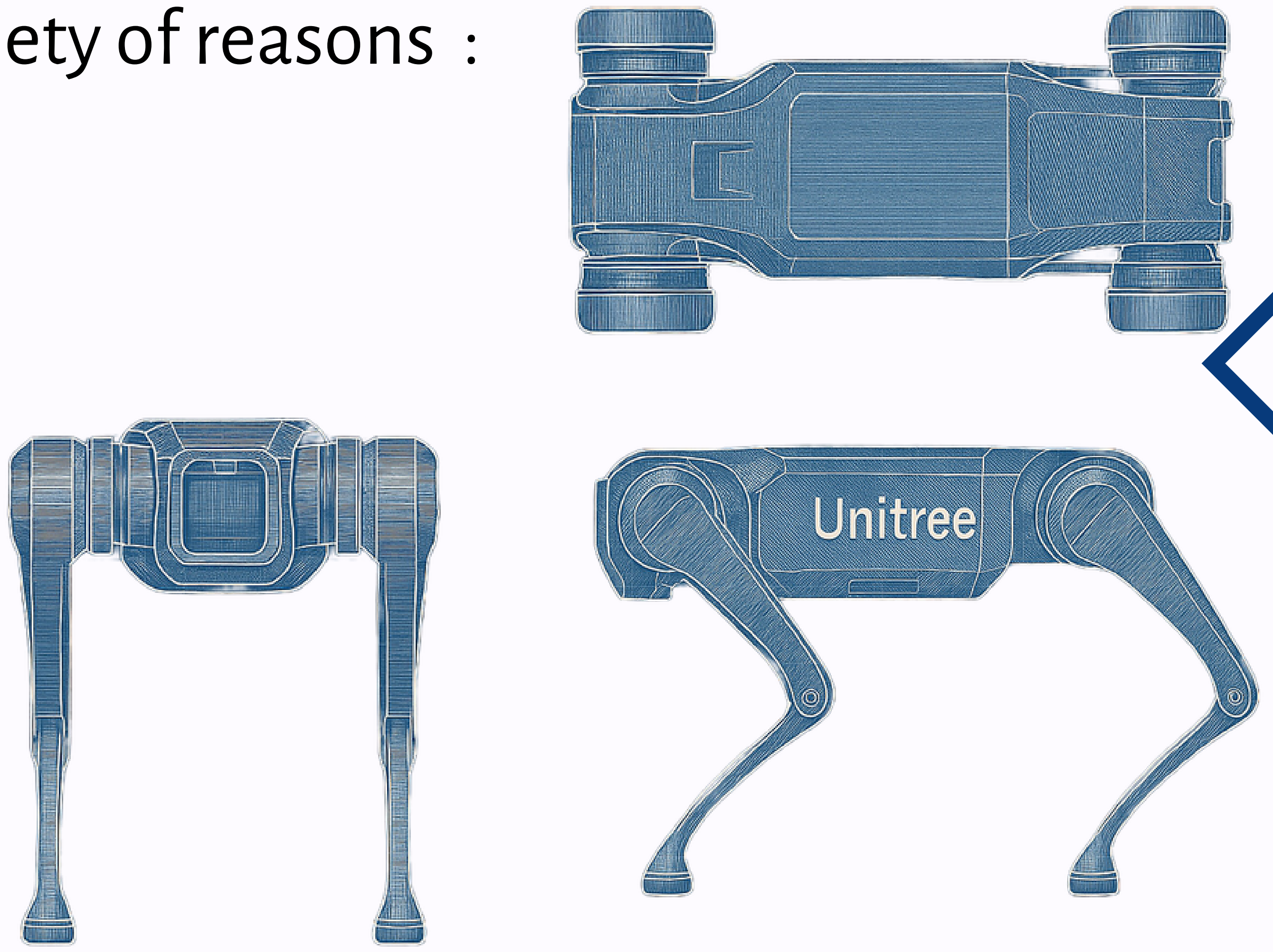
PRESENTED BY :
GUZZI RÉMI & WAELES-DEVAUX ADRIEN

OVERVIEW

PRESENTATION OF THE PROJECT

- Localizing Legged Robots is complex for a variety of reasons :

- Complex and non linear dynamics
- Foot slippage and more complex terrains
- Non continuous contact with the ground
- Quantity of sensors needed for its estimation ↗

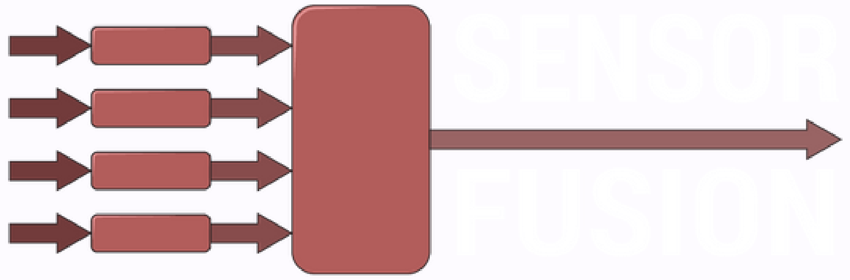


Common approach to address this problem :

Sensor Fusion for Non Linear Systems

PRESENTATION OF THE PROJECT

Objective: create lab session for robotic sudents



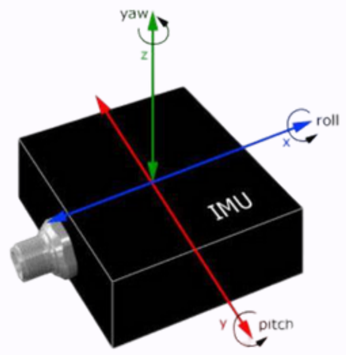
- Understand sensor fusion
- Implementing filtering techniques (complementary, kalman)
- Apply these filters on real data, on the Unitree go2

PRESENTATION OF THE PROJECT

Practical case of visualization :

- Create an obstacle avoidance
- make the robot go back to its initial position after the avoidance
- implement the filters on this program
- compare the results between the filters and non filter

CHOICE OF SENSORS FOR LOCALIZING A QUADRUPED



Proprioceptive Sensors

Inertial Measuring Unit (IMU)

Joint Encoders

Foot Force Sensors

Barometer

Magnetometer

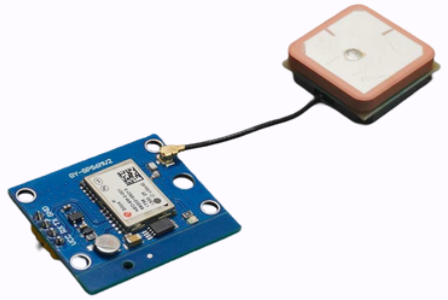
Exteroceptive Sensors

Lidar

Cameras

GPS (only for outdoor)

RFID / BLE Beacons



COULD WE USE ONLY ONE SENSOR ?

IMU

ACCELEROMETER

✅**Good estimations in non accelerating conditions**

✅Can track dynamic movements.

❌ No orientation information

❌No distinction between gravity and acceleration

❌Double integration to obtain position, leads to **drift quickly**

GYROSCOPE

✅ Excellent accuracy for estimating orientation over short periods of time

❌**Drifts on long-term**

❌**No Position Information**

JOINT ENCODERS

✅ Highly Accurate

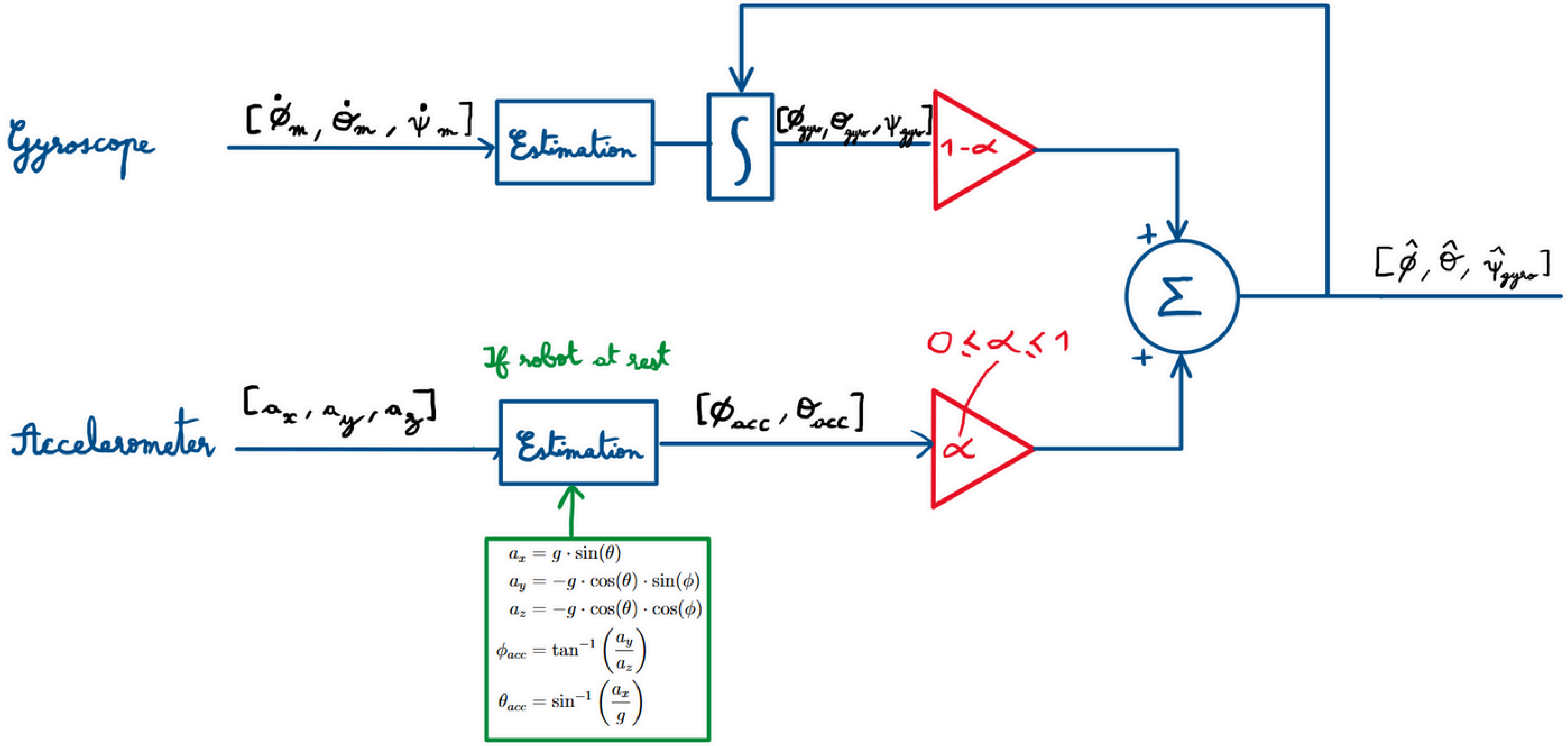
✅ **No drift**

❌ **Only meaningful when foot is in contact with ground**

❌ Sensitive to mechanical issues

❌**Slippage invalidates data**

COMPLEMENTARY FILTER : COMBINING ACCELEROMETER AND GYROSCOPE FOR POSE ESTIMATION



INTERFACING WITH THE ROBOT

SDK

- Protocol: DDS (ddsc, ddscxx)
- Ethernet (via UDP/TCP)
- Language: C++ (or python)
- Command types: both Low and High Level
- Cmake

FILTER ON & OFF EXPERIMENT

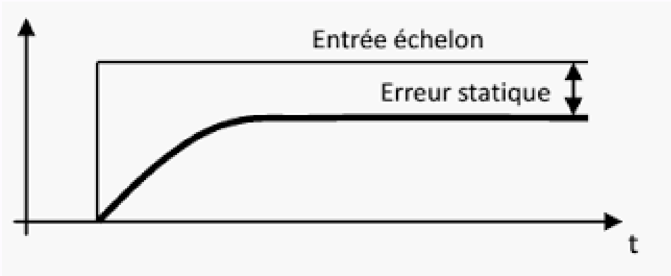
Starting at a certain position

Avoid obstacles

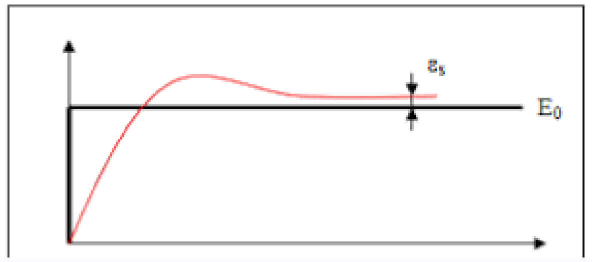
Got out of its position

Returning to its initial position:

WITHOUT FILTER



WITH FILTER



KALMAN FILTER

STEP 1
PREDICTION

Uses a mathematical model to forecast the next state based on previous data and system dynamics.

STEP 2
UPDATE

Adjusts prediction by incorporating sensor measurements, weighting them based on their uncertainty.

WHY IS IT NOT EFFICIENT WITH LEGGED ROBOTS ?

- Dynamics are nonlinear (jumping, rotating, swinging legs).
- Not resilient to unpredictable disturbances
- The terrain is often uncertain or uneven, causing issues like foot slippage.

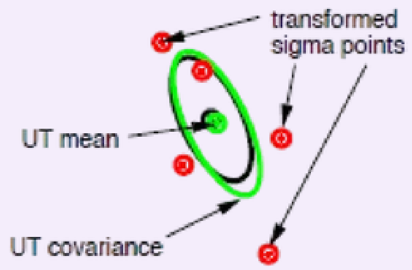
UNSCENTED KALMAN FILTER

Generating Deterministic Samples from Gaussian States

Multivariate Gaussian “cloud”.

Generation of sigma points (possible states, spread around the mean).

These sigma points are then propagated through the nonlinear system model.



Prediction Step

Using IMU

IMU provides acceleration and angular velocity.

We substract gravity vector.

UKF integrates values over time to predict changes in velocity, orientation, and position.

Correction Step

Using Leg Odometry

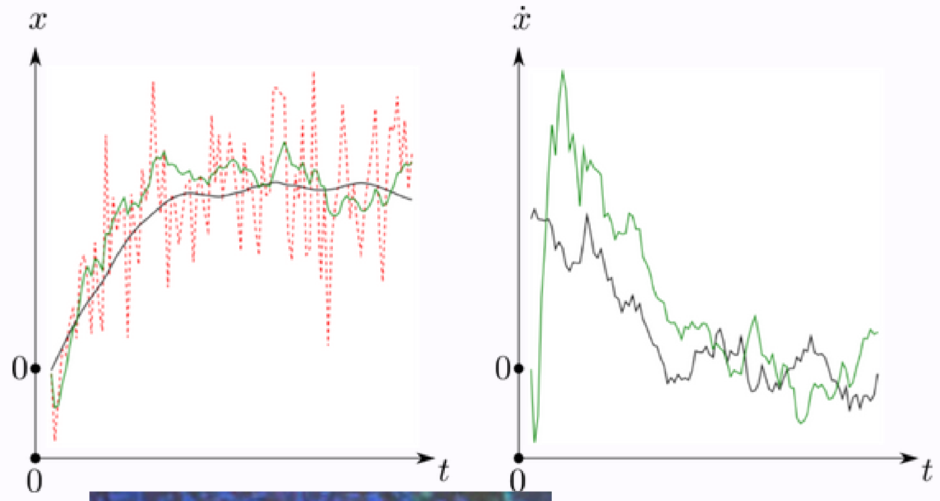
Compute **Forward Kinematics** for foot relative position estimation.

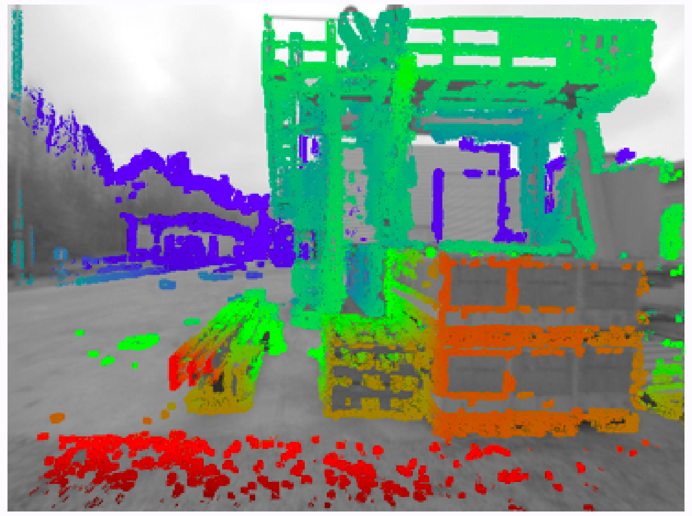
Applies only each time a foot is in contact with the ground.

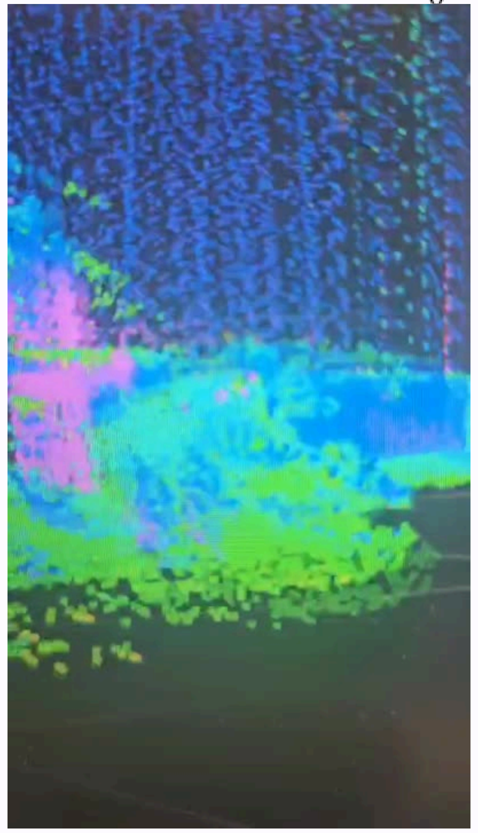
Update our belief with difference between prediction (from IMU) and actual motion (from legs kinematics).

FUTURE IMPROVEMENT

- Finishing implementing Unscented Kalman
- Using filters for general use of the robot
- Take more time to tune the filter parameters
- SLAM







CREDITS

- ## UNITREE GO2 INTERFACING

Unitree Go2 Documentation : support.unitree.com/home/en/developer/

- ## COMPLEMENTARY FILTER

Phil's Lab - Sensor Fusion : https://youtu.be/RZd6XDx5VXo?si=_POkmP69mj-m1VXt

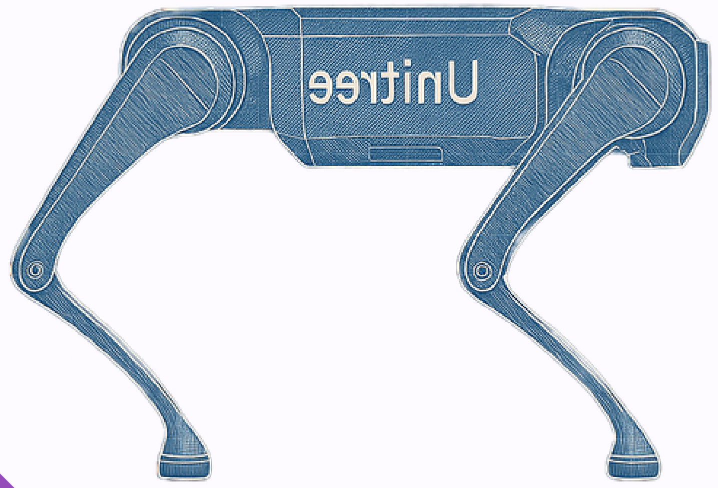
- ## KALMAN AND NON LINEAR FILTERING

Kalman and Bayesian Filters in Python by *Roger R. Labbe*
github.com/rlabbe/Kalman-and-Bayesian-Filters-in-Python

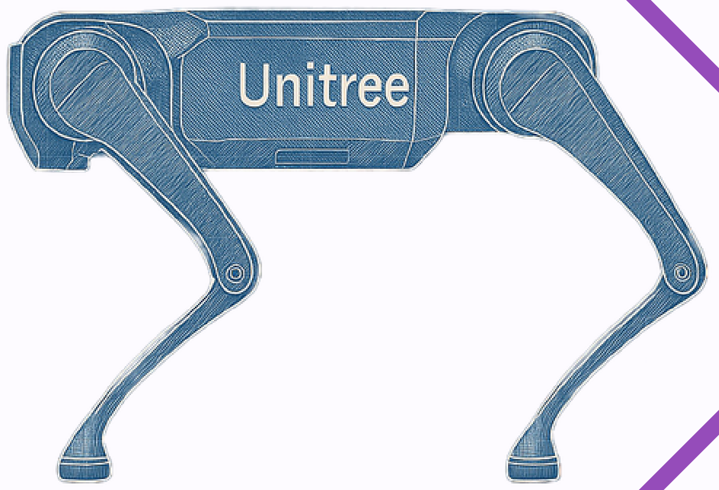
Sensor Fusion and Non-linear Filtering for Automotive Systems by *ChalmersX University*
edx.org/learn/mechanical-engineering/chalmers-university-of-technology-sensor-fusion-and-non-linear-filtering-for-automotive-systems

Robust State Estimation for Legged Robots with Dual Beta Kalman Filter
by *Tianyi Zhang, Wenhan Cao, Chang Liu, Tao Zhang, Jiangtao Li, Shengbo Eben Li*
arxiv.org/abs/2411.11483

Polytech Nice Sophia



THANK YOU



Presented by :

GUZZI Remi

WAELES-DEVAUX Adrien

Robotics And Autonomous Systems Master | 2024-2025