

Lab 7

*School of Architecture, Civil and
Environmental Engineering*

EPFL, SS 2023-2024

http://disal.epfl.ch/teaching/signals_instruments_systems/

What this lab is about

- Mini tutorial on Webots
- Understanding sensors
 - examples of sensors on-board the e-puck
 - notion of ‘noise’
 - how Webots simulates sensors
- Understanding how to use sensor data
 - input for actuation (example: robot control)
 - manipulation of data (logging and plotting)

Reminder: Webots GUI

scene tree

world view

editor

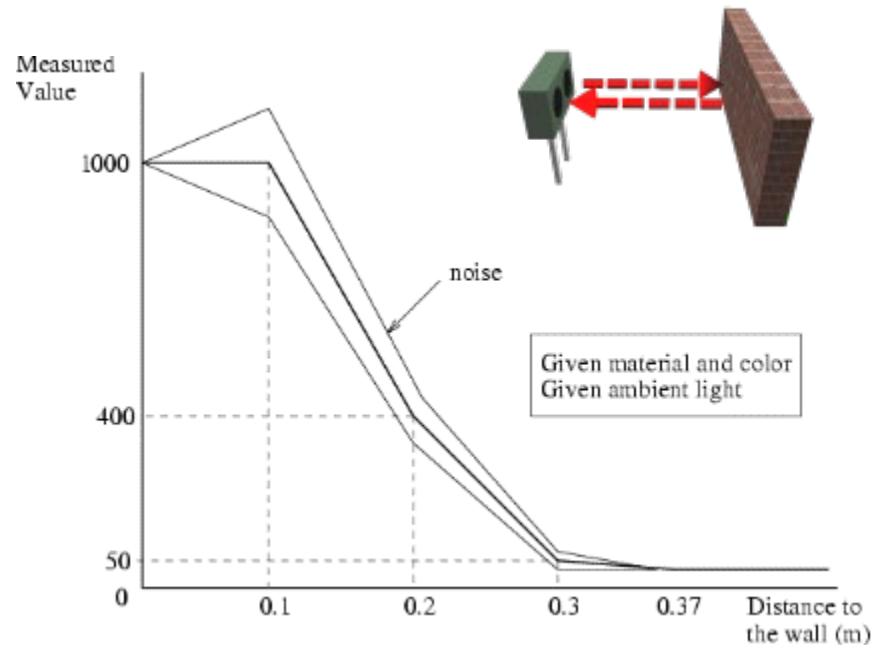
The screenshot displays the Webots GUI interface with four main components:

- Scene Tree (Left):** A hierarchical list of objects in the scene, including WorldInfo, Viewpoint, Background, PointLight, SquareArena, and several rock models (DEF short_rock_1_Solid to DEF short_rock_5_Solid) along with a DifferentialWheels component.
- World View (Center):** A 3D perspective view of a yellow arena with a robot in the center and several grey rock blocks scattered around. A small inset window shows a top-down view of the robot.
- Editor (Right):** A code editor window showing the C++ source code for the robot's controller (e-puck.c). The code includes headers for robot, differential_wheels, distance_sensor, light_sensor, camera, and accelerometer, and contains a main function that initializes and enables these sensors.
- Console (Bottom):** A terminal window showing the output of a 'make' command, indicating that the controller was successfully compiled and started.

console

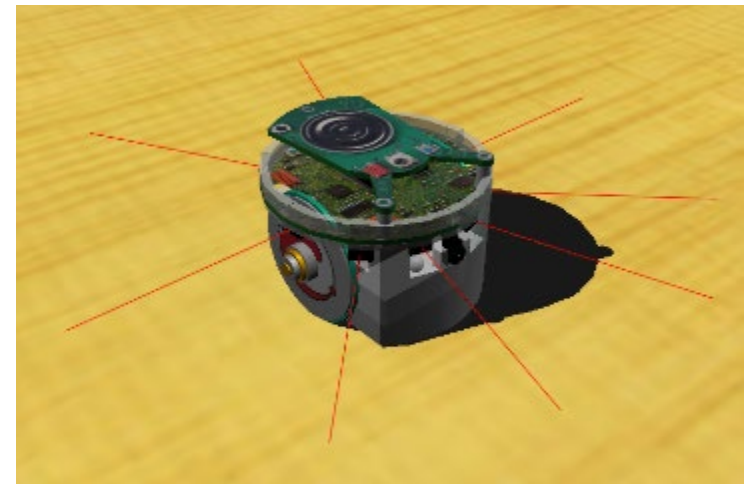
Reminder: Modeling sensors

- Capture **non-linearities** and **noise** of sensors.
- However, **calibration** is often approximative.
- Most often, sensor response is defined by a lookup table (here a proximity sensor):

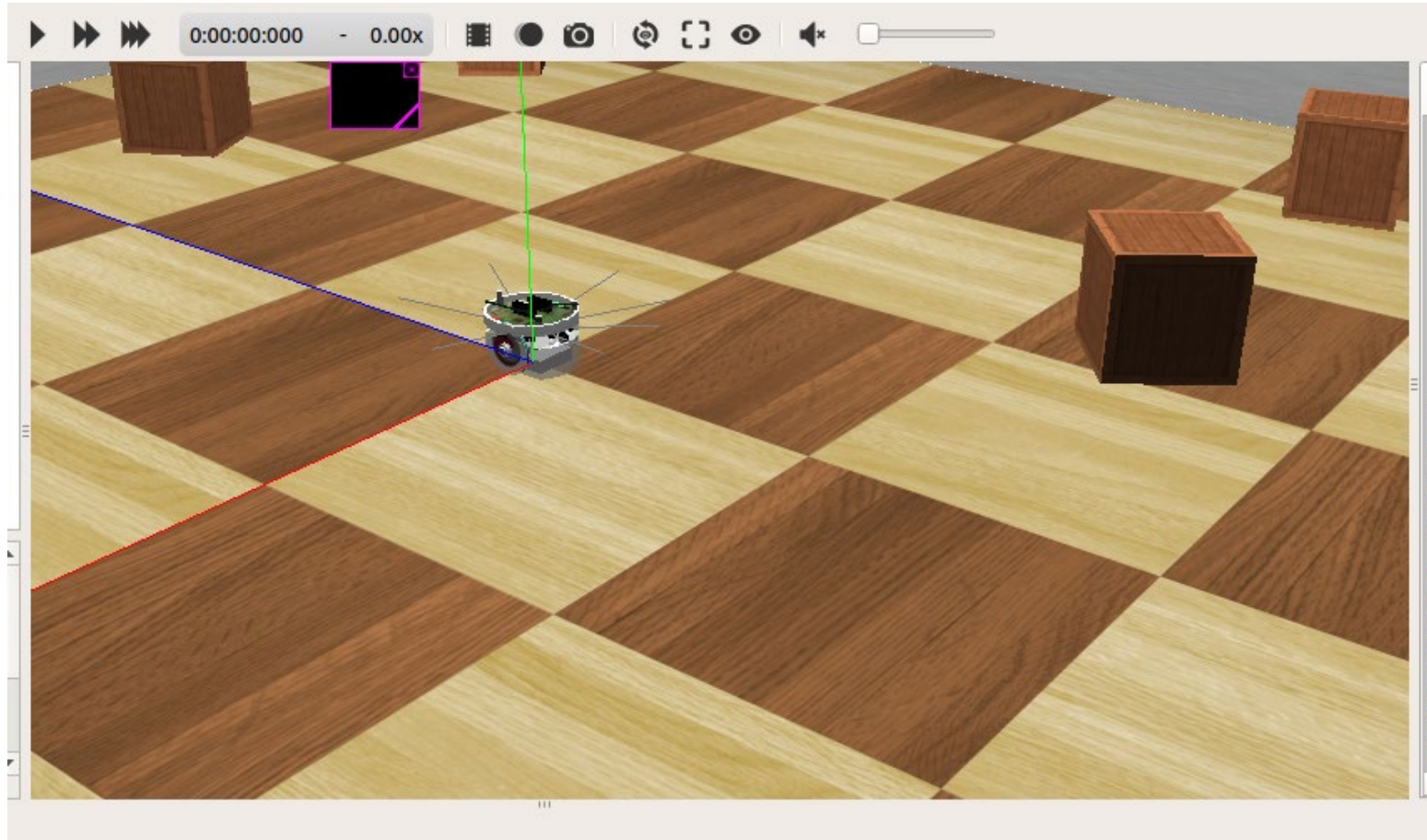


```
lookupTable [
  0
  0.1
  0.2
  0.3
  0.37
]
[
  1000
  1000
  400
  50
  30
]
[
  0,
  0.1,
  0.1,
  0.1,
  0
]
```

distance value noise



Robot control – Obstacle avoidance



Robot control – Braitenberg

