Lab 8

School of Architecture, Civil and Environmental Engineering

EPFL, WS 2021-2022

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

Lab 8 outline

- Contents
  - Positioning Systems and Frames
  - 1D Odometry with accelerometer
  - 2D Odometry with wheel encoders

} Noise-free Sensors

- Tools
  - Webots
  - C compiler
  - MATLAB

Getting Started

- Installation
  - Go to moodle and download:

    Lab08 (.zip) and the assignment
- Starting with Webots
  - Launch Webots (from terminal)
  - Open World file *odometry.wbt*
  - Clean + Build robot controller

Robot Pose

- In 2D : Ground Robot
  - Needs 3 variables

  (e.g. 2 positions + heading)

  $p_{2D} = [x, y, \theta]^T$

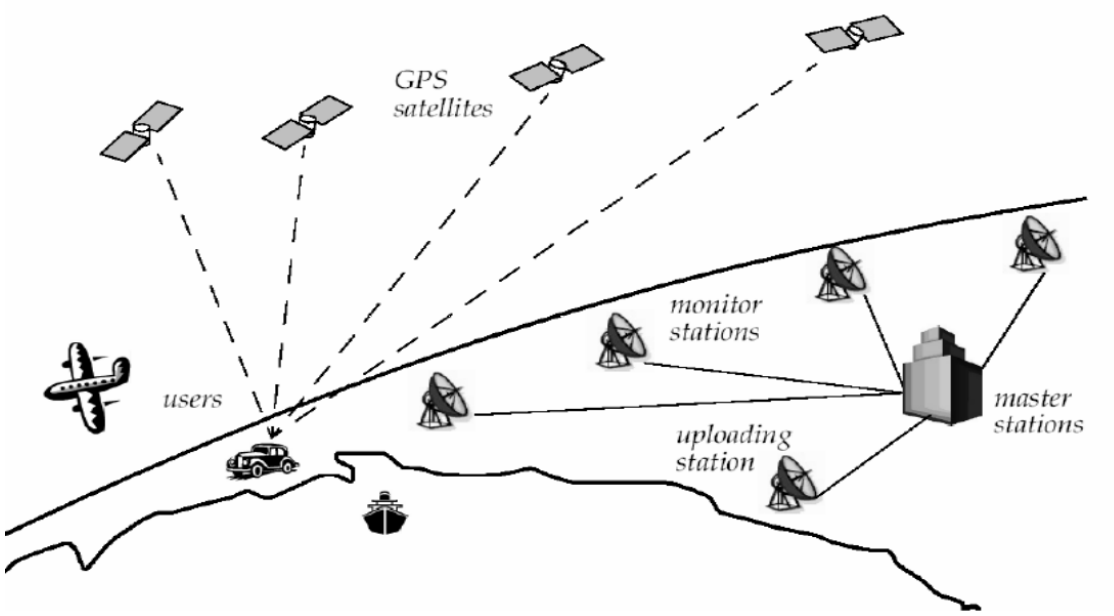
- In 3D : Aerial robots
  - Needs 6 variables

  (e.g. 3 positions + 3 angles)

  $p_{3D} = [x, y, z, \phi, \theta, \psi\ ]^T$

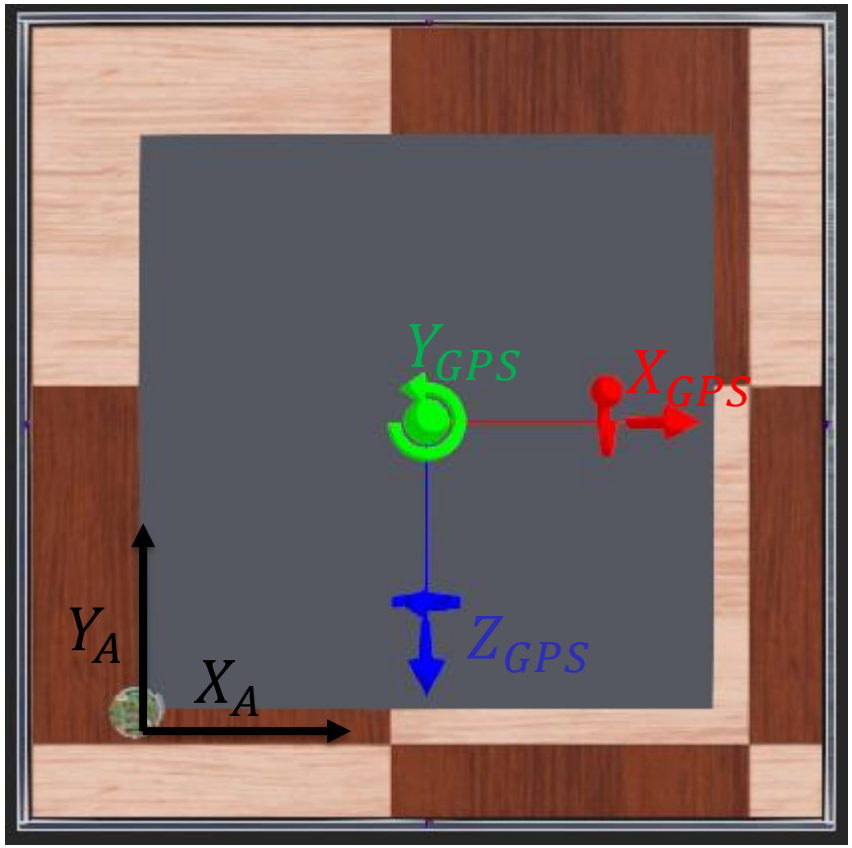
Positioning Systems

- Example : GNSS, MCS, IR-UWB
- Goals : Get positions + orientation (sometimes)



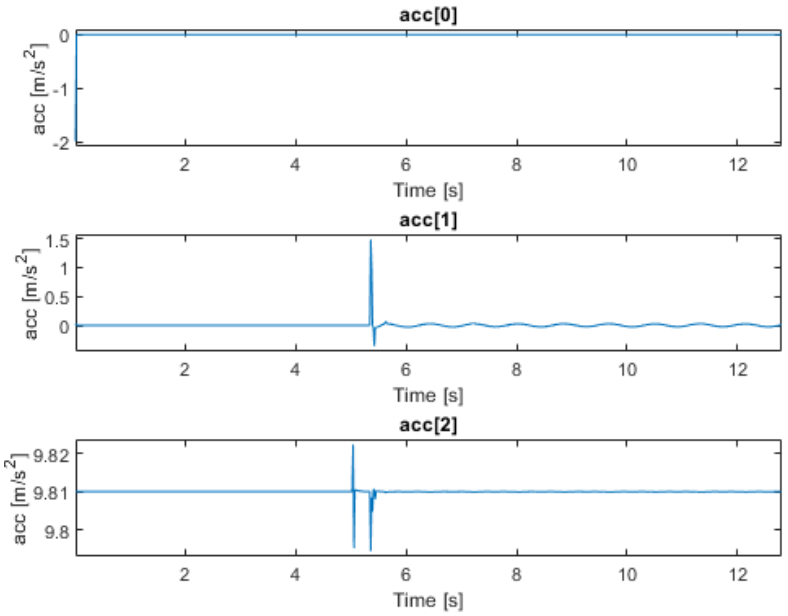
Positioning Systems (Webots)

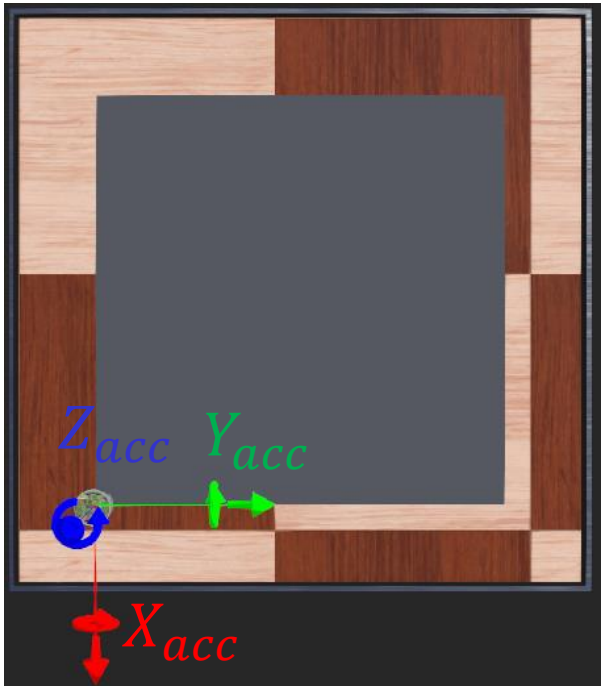
- Example : GNSS
  - GPS (USA)
  - Galileo (EU)
  - Beidou-2 (China)
- Frames :
  - GPS frame (inertial)
  - A frame (inertial)



Check and test your frame conversion !

Odometry 1D





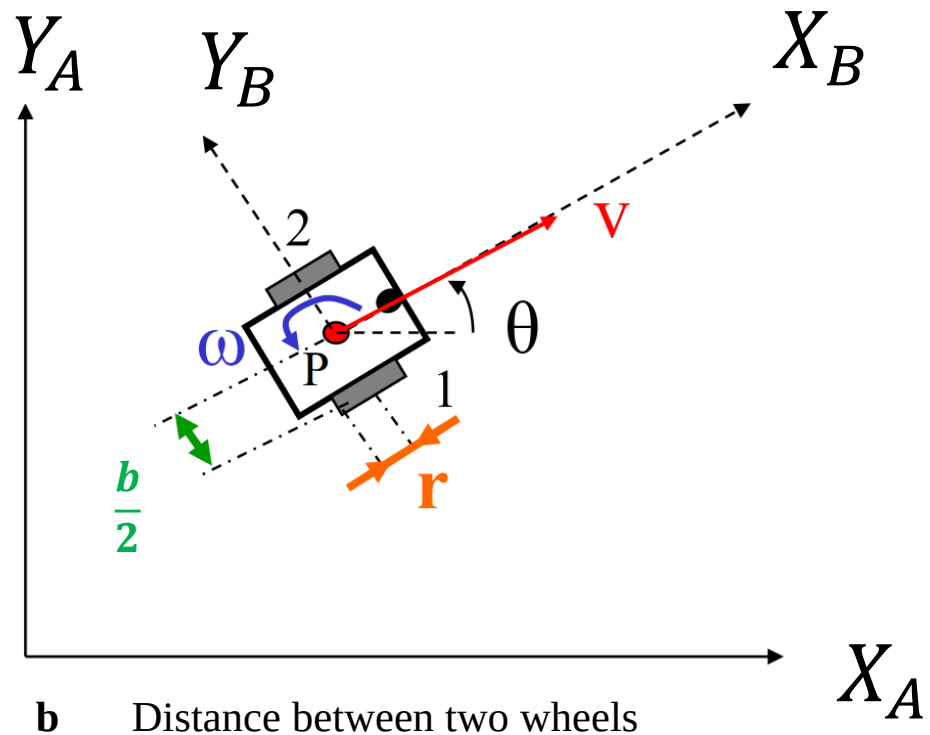
- Goals :
  - Understand the accelerometer values
  - Derive motion model based on accelerometer
  - Implement + Test your equations against GPS

Check and test your frame conversion !

Odometry 2D : Recipe

1) Compute Delta wheel encoders
2) Compute forward (**v**) and rotational (**ω**) speed in body frame {B}.
3) Transform **v** and **ω** to frame {A}
4) Derive the continuous time equations for the motion model
5) Discretization (integration) of your equations
6) Implement, test and improve your odometry

Useful slides :
23-35 from lecture notes (Week9)



b Distance between two wheels
r Wheel radius
{A} Frame A , static in time
{B} Frame B (Body), move with the robot

Code complements

```
(void *)memset(void *str, int c, size_t n)
```

1) Set array `double acc[3]` to 0 :

```
memset(arr, 0 , sizeof(arr));
```

2) Set structure `pose_t _pose` to 0 :

```
memset(&_pose, 0 , sizeof(pose_t));
```

```
(void *)memcpy(void *dest, const void * src, size_t n)
```

1) Copy array `double* gps_position` to array `double _meas.gps[3]`:

```
memcpy(_meas.gps, gps_position, sizeof(_meas.gps));
```

2) Copy structure `pose_t odo_pose_enc` to struct `pose_t odo` :

```
memcpy(odo, &_odo_pose_enc, sizeof(pose_t));
```

General Information

- Code :
  - Use the same controller + world for all the parts
  - Control the robot with your keyboard

Table 2: Keyboard keys and corresponding actions

Keyboard key	Simulation actions
R	The robot start to move
S	The robot stop
U	Increase the speed
D	Decrease the speed
Up Arrow	The robot move forward
Down Arrow	The robot move backward
Left Arrow	The robot turn left
Right Arrow	The robot turn right

Click on the world window and then use the keyboard!

Feedback for Lab 8

Please help us improve the labs by giving us feedback.

Thank you and enjoy the lab!