Autonomous Robotics

CS - 7630

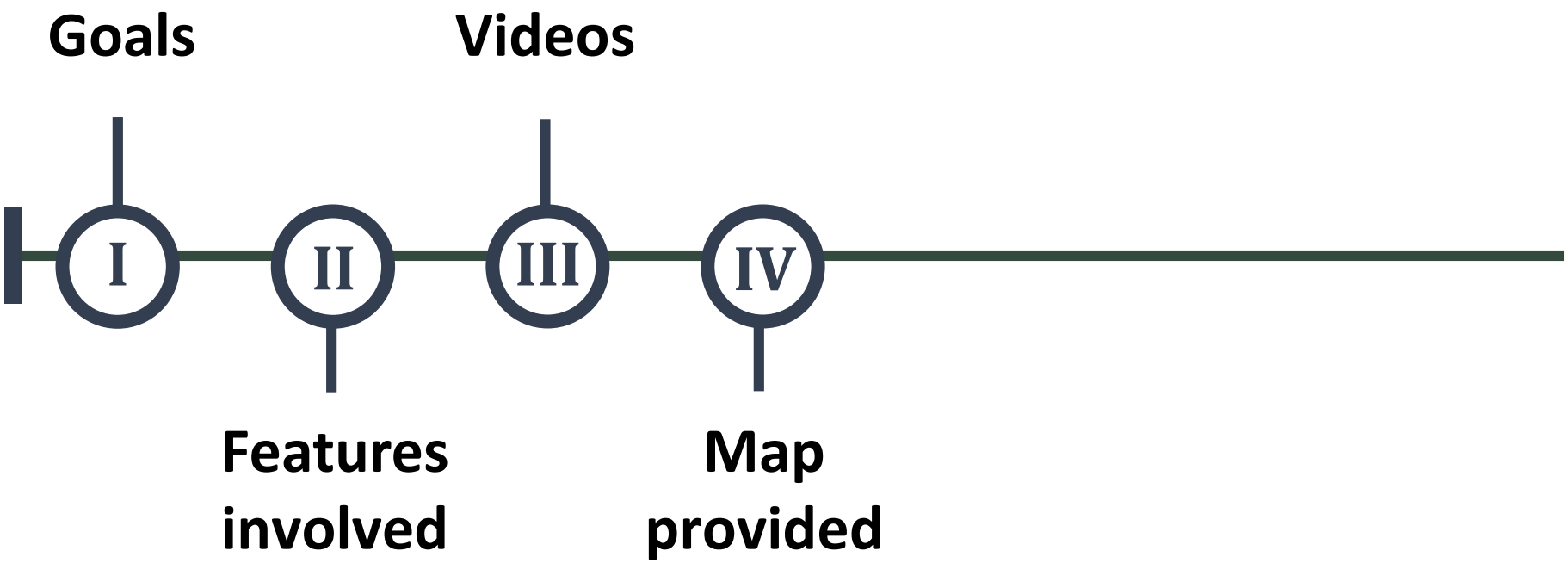
Final

By BENOIT T. , DERKAOUI H.

March 2018

Georgia Tech Lorraine

Plan :



1 Goals

The mission of our autonomous robot:

- Highest wifi intensity mapping
- Environnement mapping

2 Features

The tools designed:

- Environment mapping, AR-tag-based SLAM
- Environment mapping, occupancy grid
- Trajectory planning
- Path following, replanning, and obstacle avoidance
- Autonomous Exploration
- Wifi Mapping
- Autonomous docking/undocking and Battery management

2 Features

Environment mapping, AR-tag-based SLAM

AR-tag-based SLAM

Update Equations

(1) Project the state ahead

$$\hat{x}_k^- = f(\hat{x}_{k-1}, u_{k-1}, 0)$$

(2) Project the error covariance ahead

$$P_k^- = A_k P_{k-1} A_k^T + W_k Q_{k-1} W_k^T$$

Prediction Equations

(1) Compute the Kalman gain

$$K_k = P_k^- H_k^T (H_k P_k^- H_k^T + V_k R_k V_k^T)^{-1}$$

(2) Update estimate with measurement z_k

$$\hat{x}_k = \hat{x}_k^- + K_k(z_k - h(\hat{x}_k^-, 0))$$

(3) Update the error covariance

$$P_k = (I - K_k H_k) P_k^-$$

State estimate

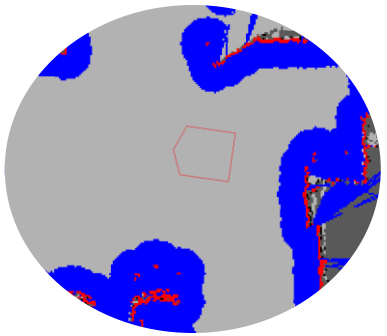
2 Features

Environment mapping, occupancy grid

Occupancy grid

Occgrid script

Cost Map



3 Hz

Laser Data →

- Free point?
- Unknown point?
- Border point?
- Occupied point?

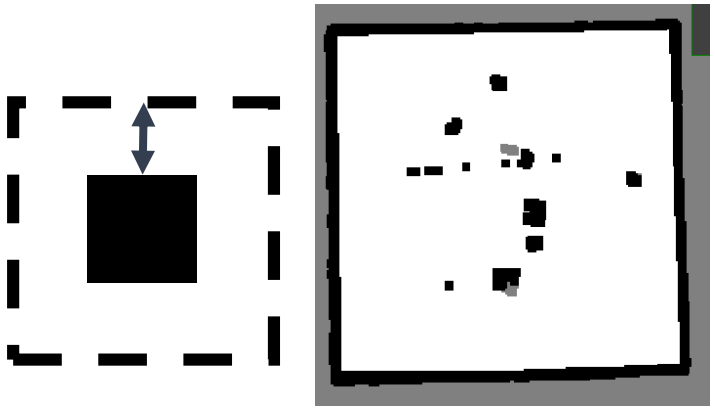
→ **2D Occupancy Grid**

2 Features

Trajectory planning

Obstacle Expansion -> White Erosion

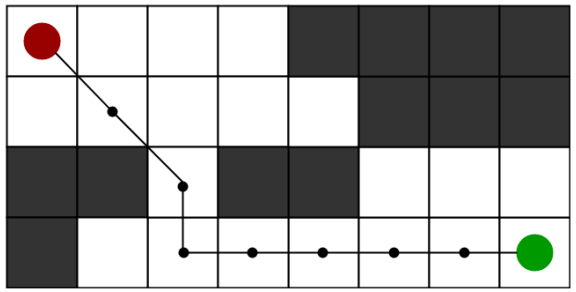
Expand the obstacles' radius



Planning -> A algorithm*

Create and publish the path thanks to the A* algorithm

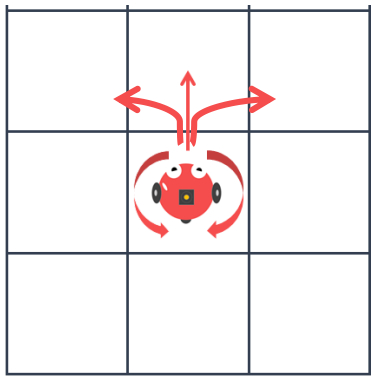
- A* uses an Euclidean distance heuristic function h(n)
- f(n) = g(n) + Ɛ h(n)



Accounting for heading

Change all features to work in 3 dimensions :

- Coordinates in the grid
- Point and matrices modifications
- Neighbours and costs



2 Features

Path following, Replanning and Obstacle avoidance

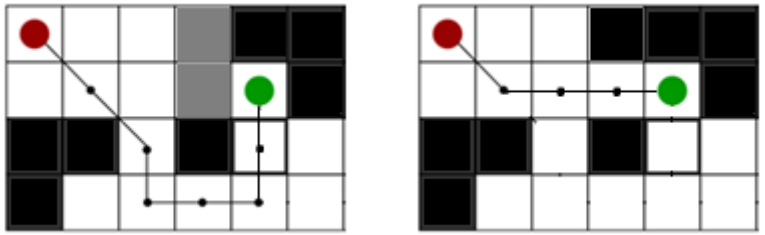
Path following

Compute Error between robot and goal

Publish linear and angular Velocity until error is low

Replanning

Every 5 seconds, the path is replanned to find a more suitable one if part of the grid was still unknown

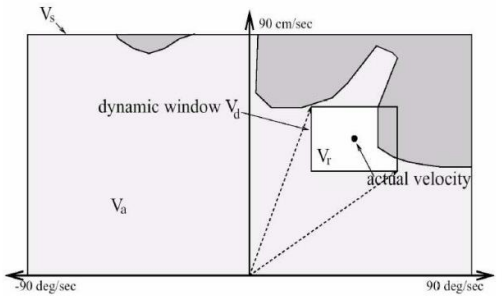


Obstacle Avoidance

Implementation of the Dynamic Window obstacle

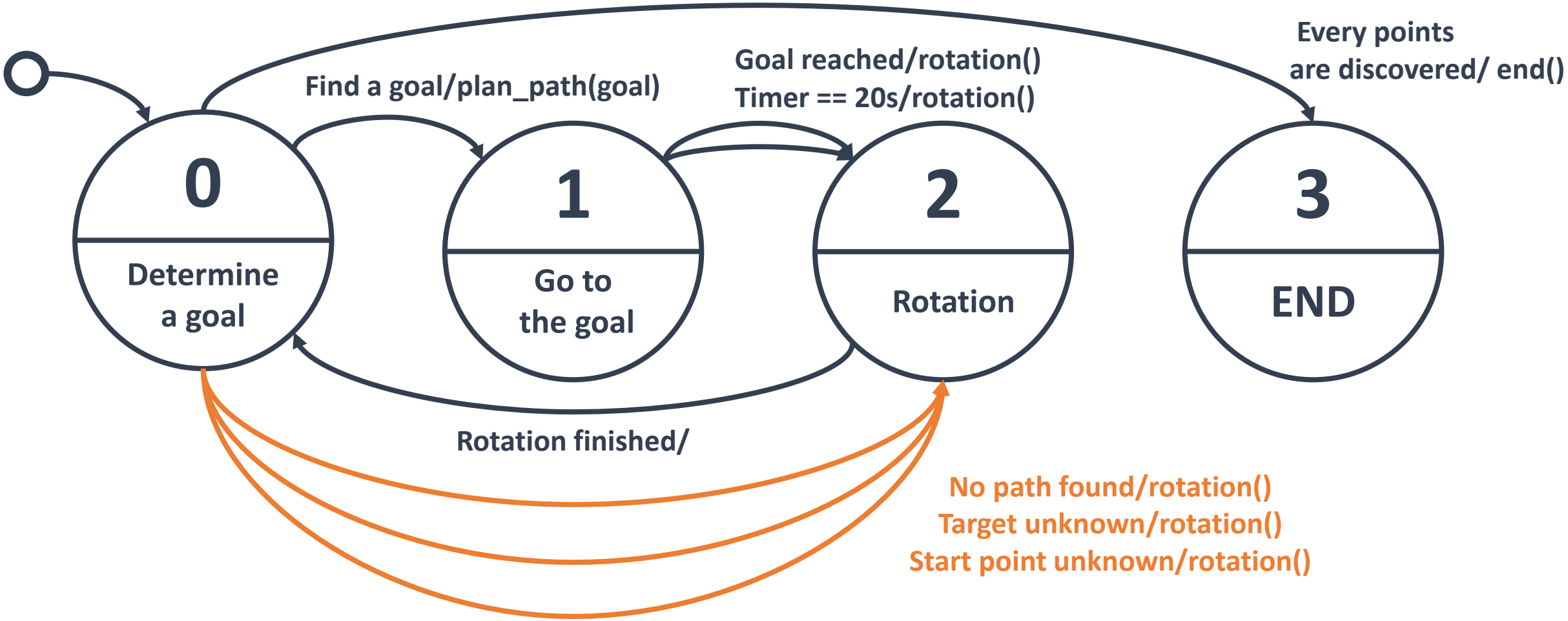
$V_r = V_s \cap V_a \cap V_d$

V_s: static limits in velocity
V_d: dynamic limits in change in velocity
V_a: limits due to nearby obstacles



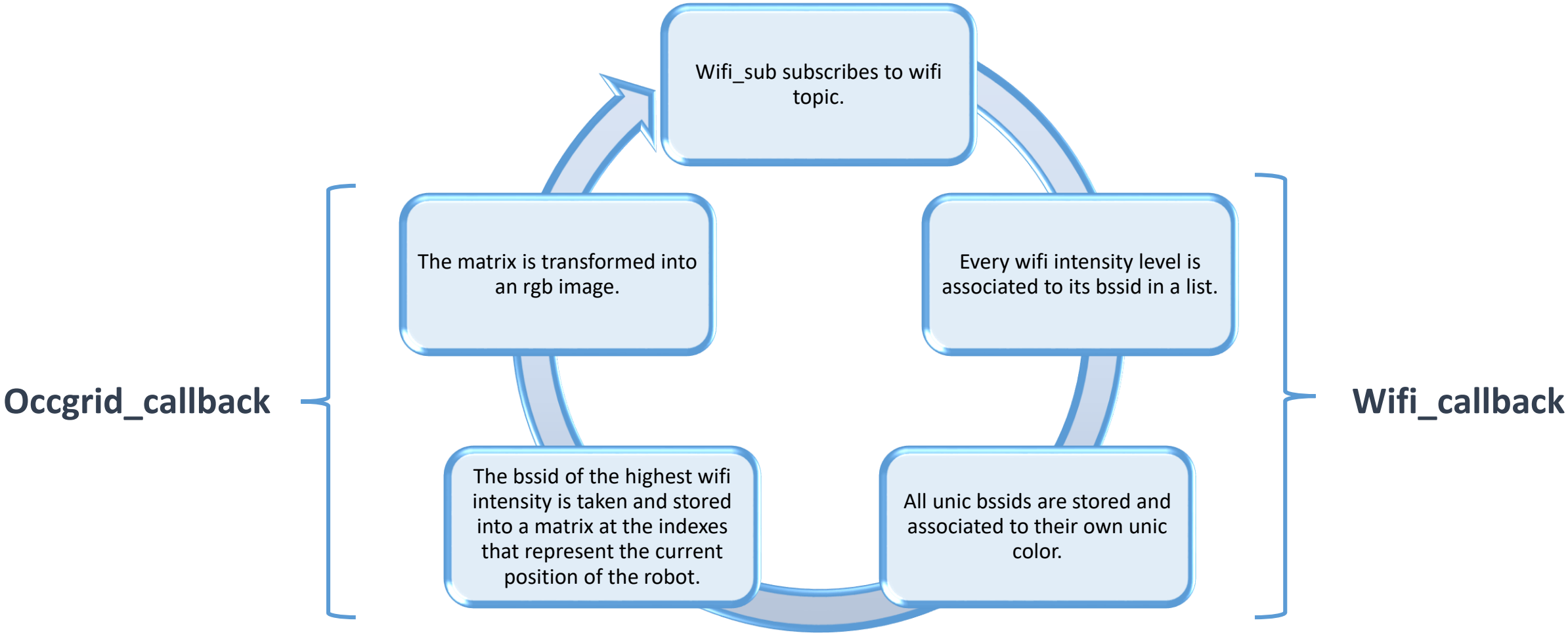
2 Features

Autonomous Exploration



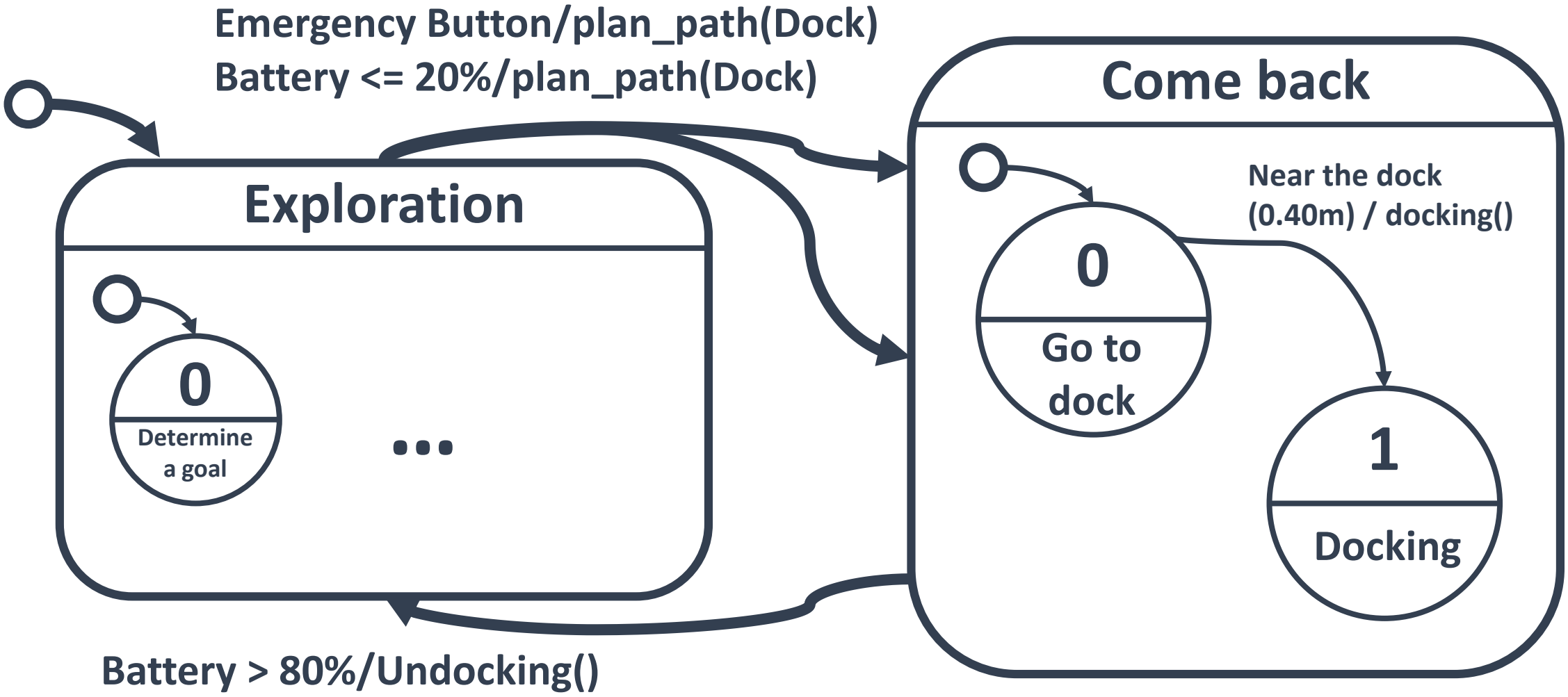
2 Features

Wifi Mapping



2 Features

Docking/Undocking & battery management

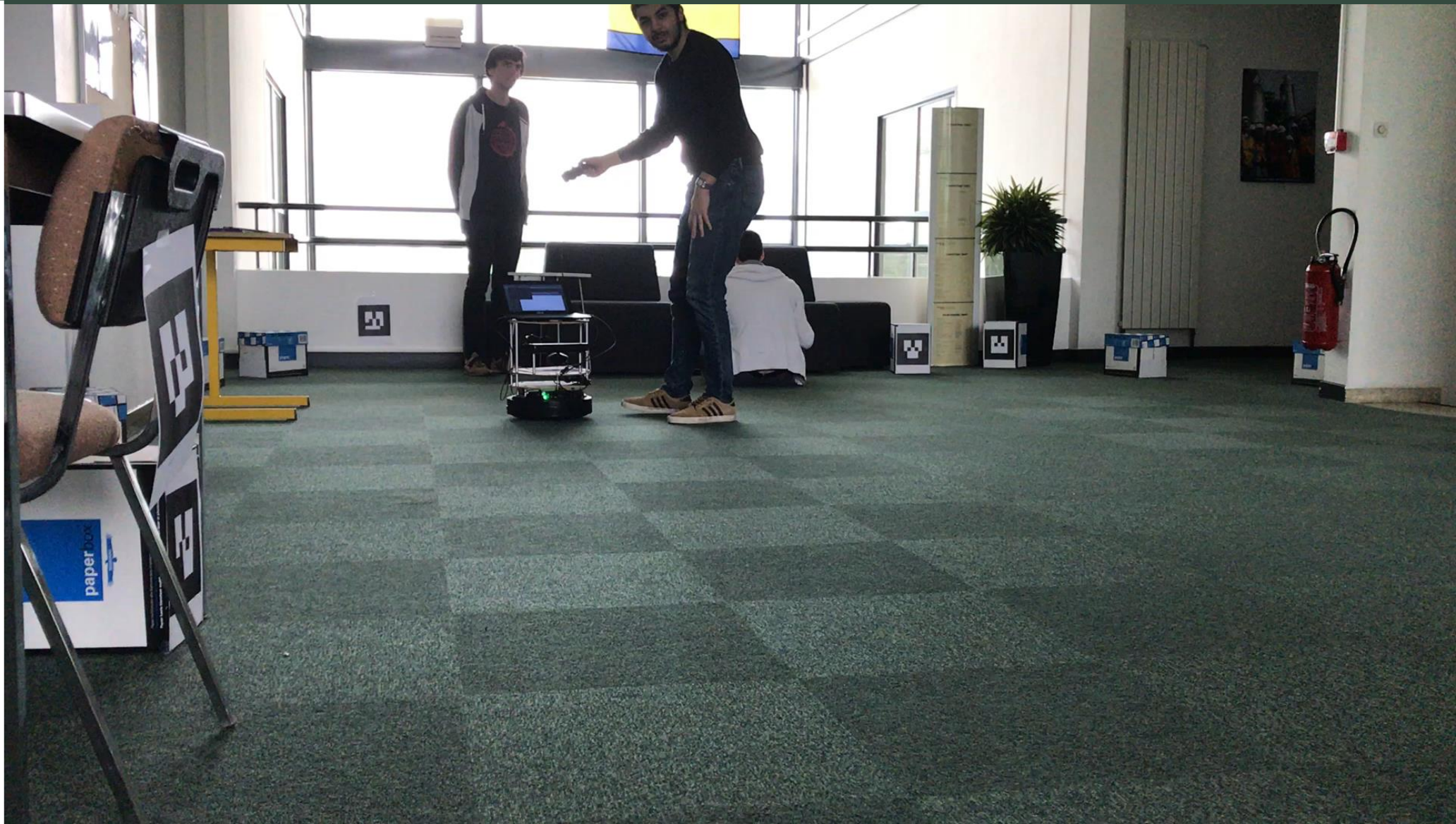


3 Videos

The Dual Mapping

3 Videos

The Docking



4 Maps provided

WIFI Map

GTL Robot

crestron

GTL Robot (relay)

ECE6612

HP-Print-fb-Color

Pic... [S... gdb Oc... DA

Environment

... Va Sc... Oc... wifi wifi

Questions