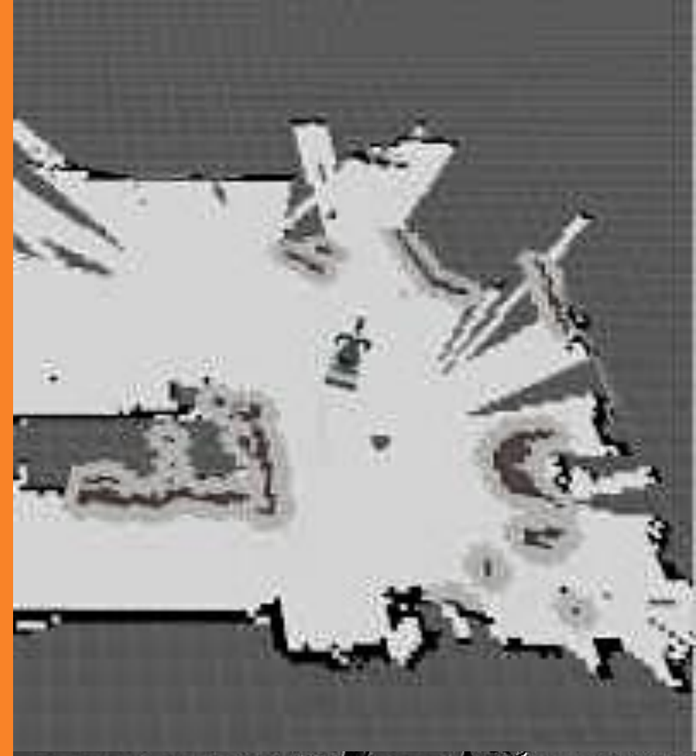
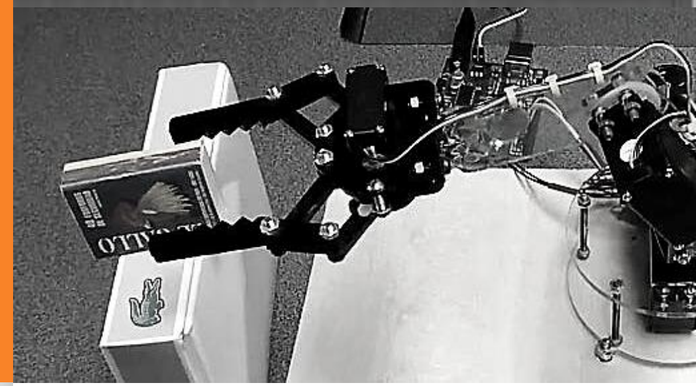
Design of a Self-Navigation Robot for Search and Rescue

Presented by Group 2





THE HONG KONG POLYTECHNIC UNIVERSITY
DEPARTMENT OF MECHANICAL ENGINEERING

FINAL YEAR PROJECT

PRESENTATION

PRESENTATION OUTLINE

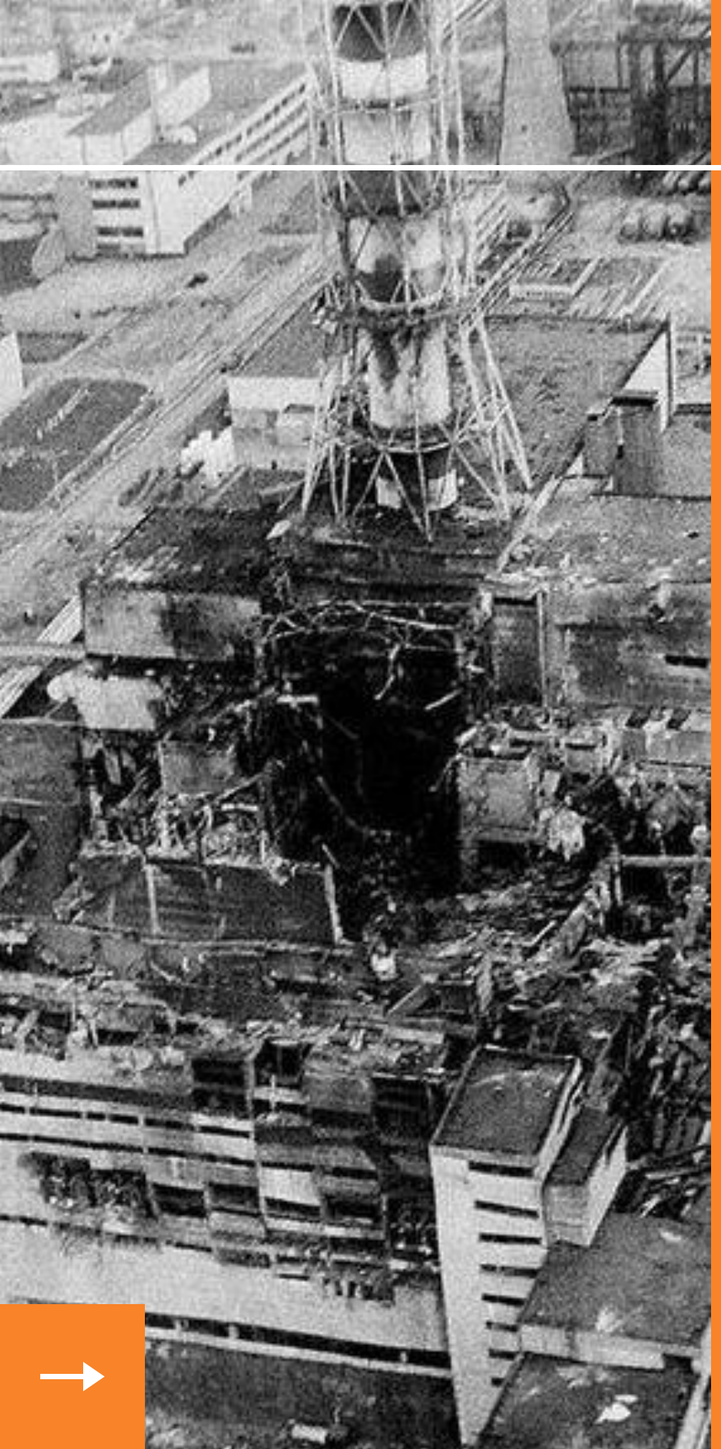
INTRODUCTION

TECHNICAL CONTENT

DISCUSSION & CONCLUSION

01

Introduction

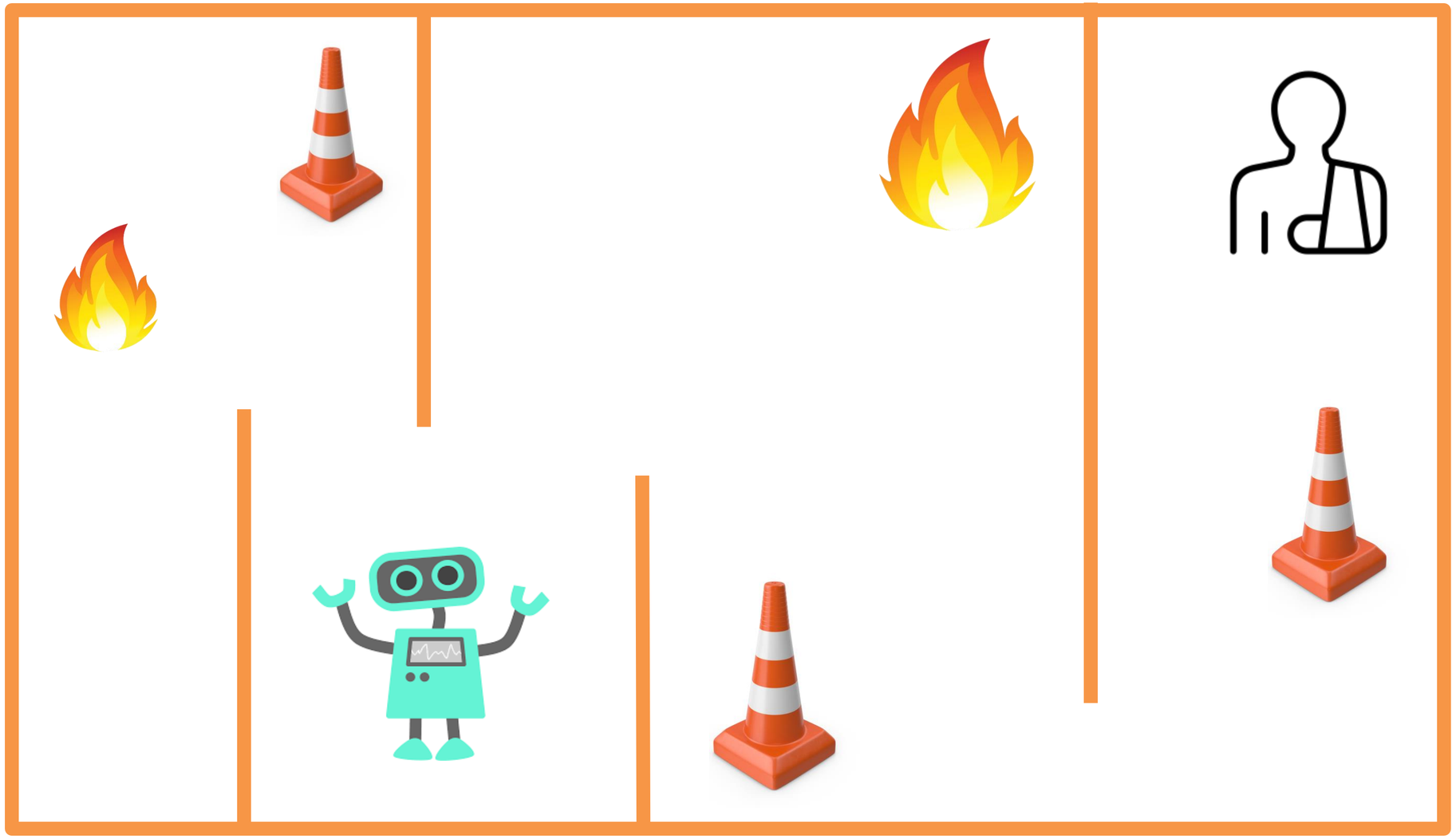


Project Background

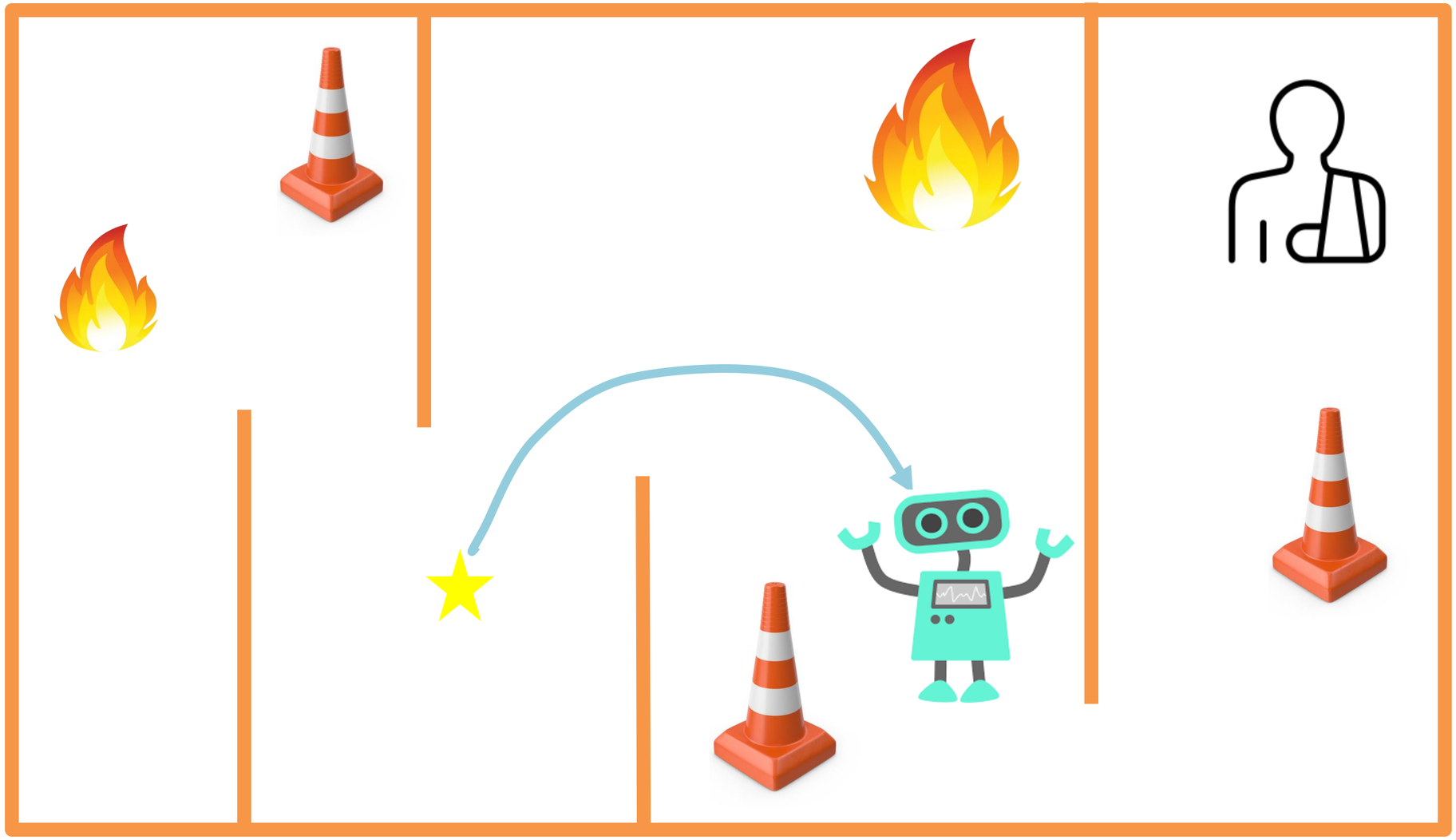
- Accident might be unpredictable
- Rescue after disaster is important

WHY RESCUE ROBOT?

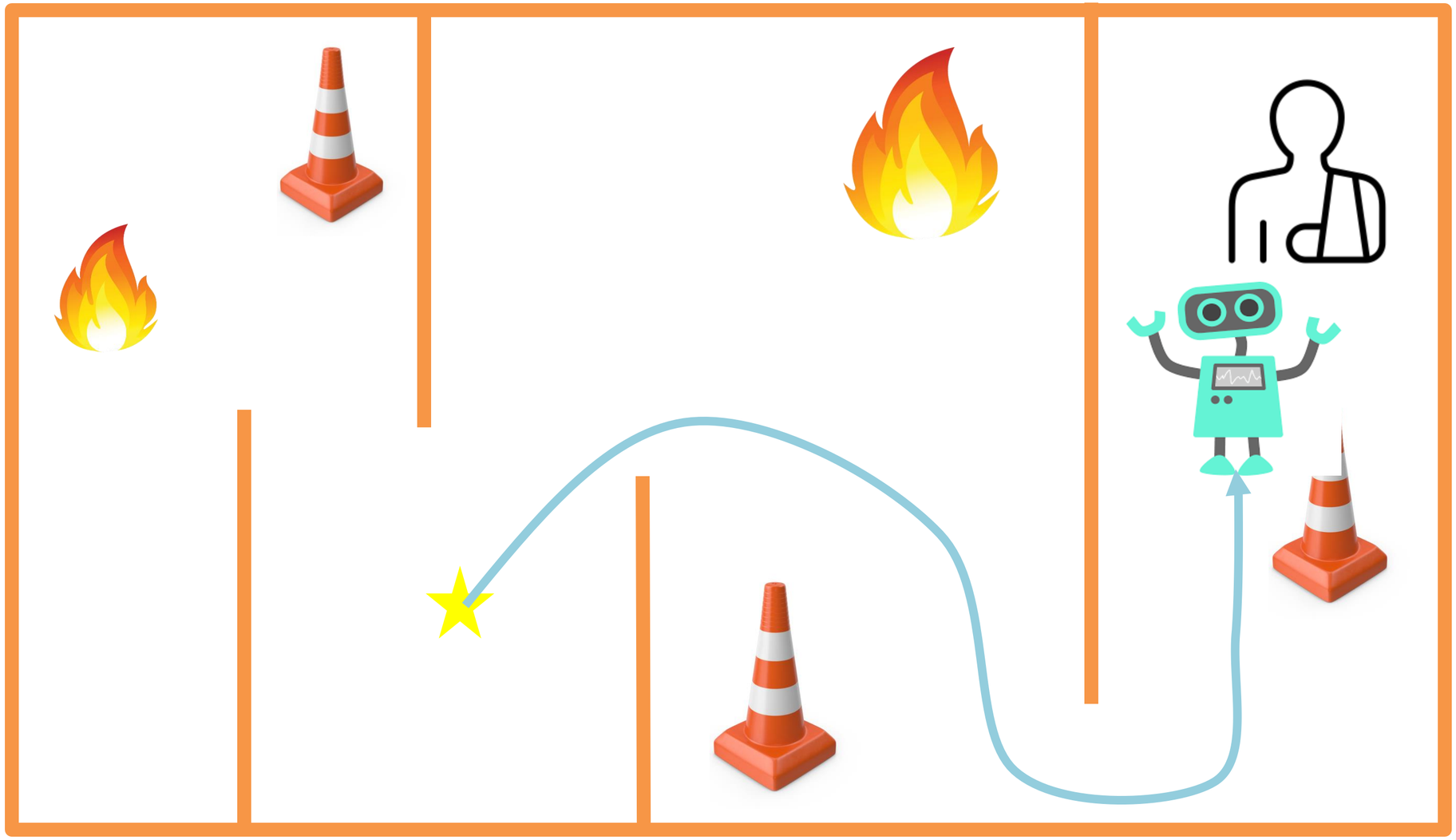
Step 1
Mapping



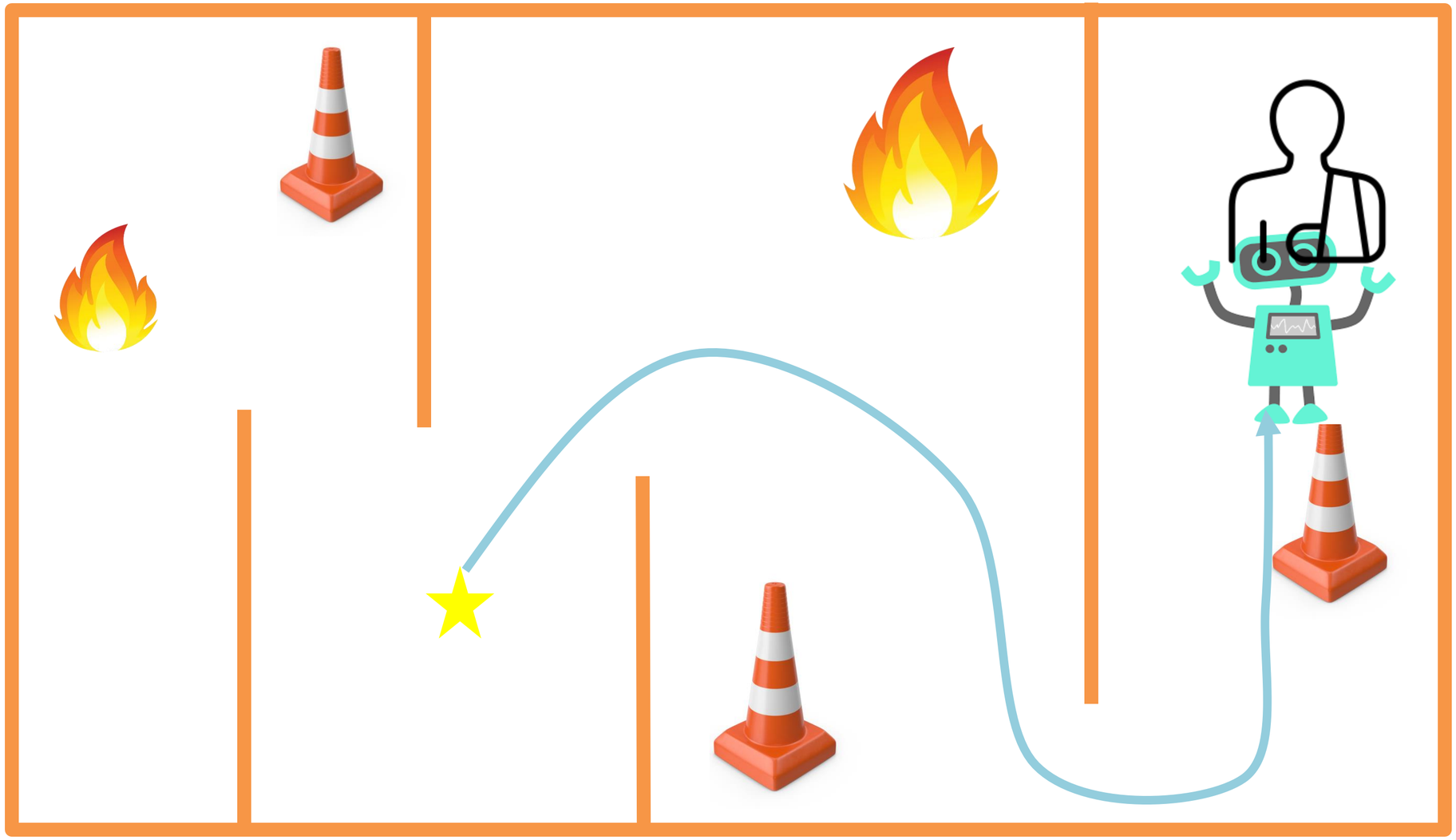
Step 2
Navigation



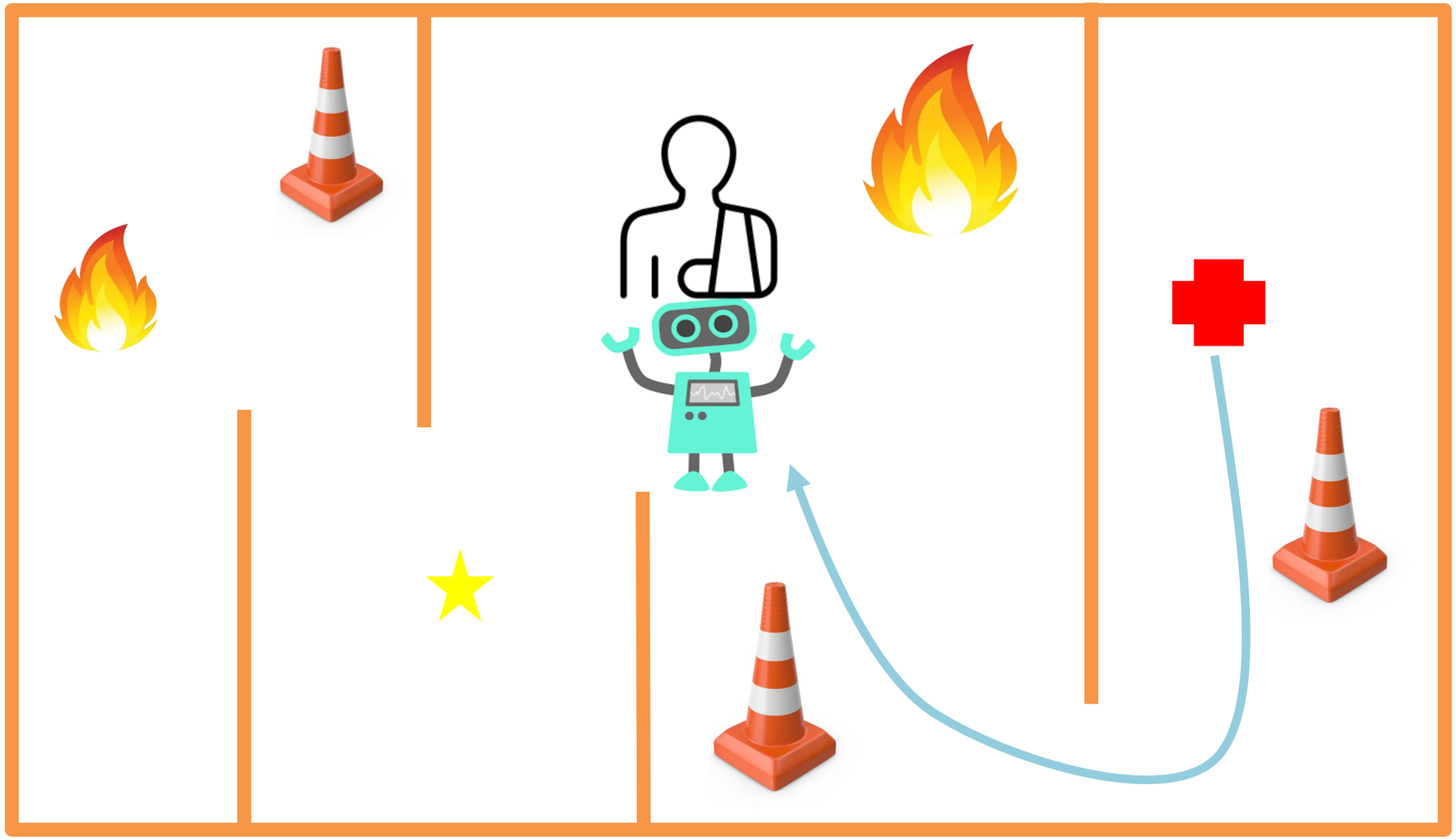
Step 3
Find Target



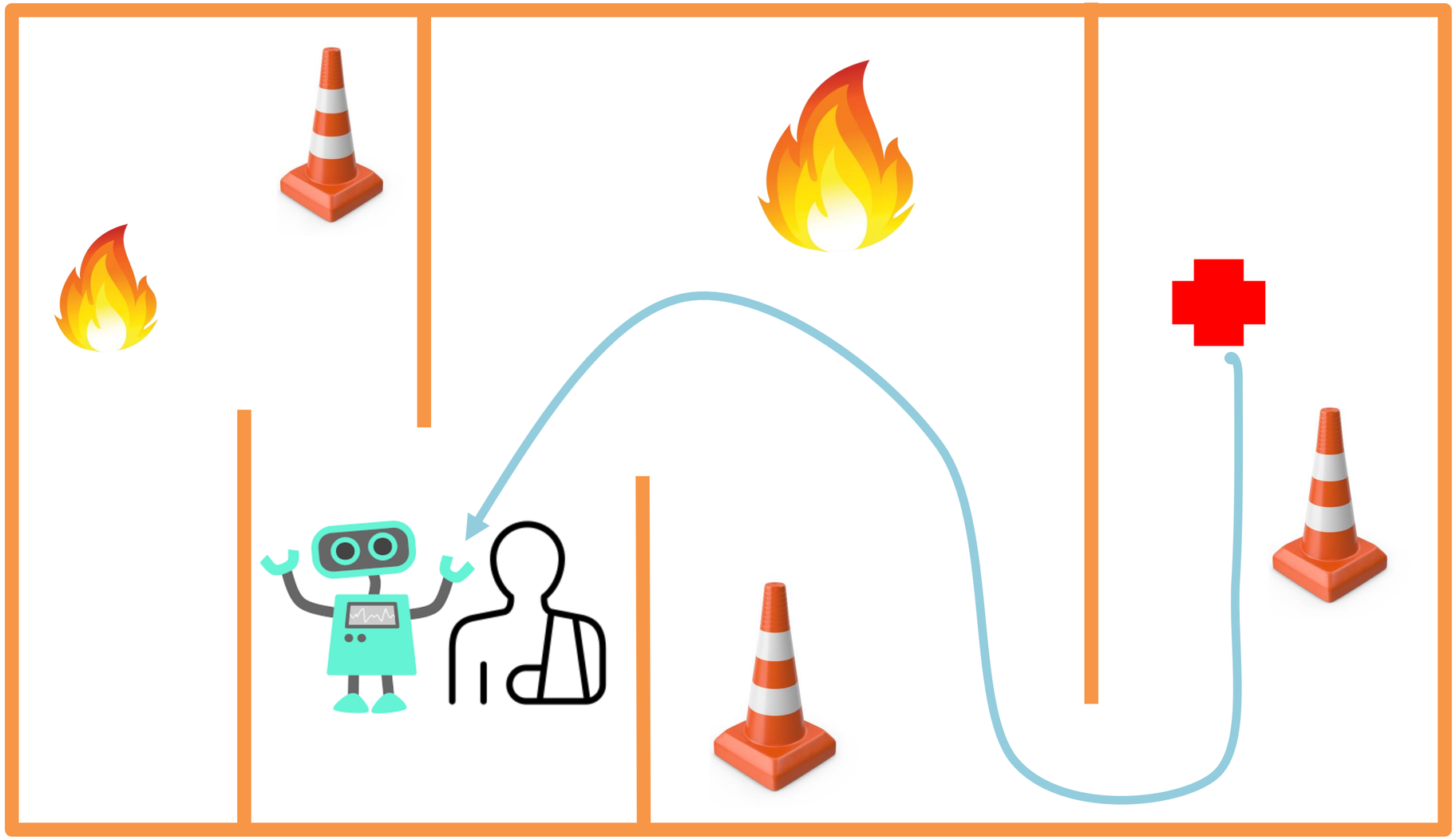
Step 4
Carry Target



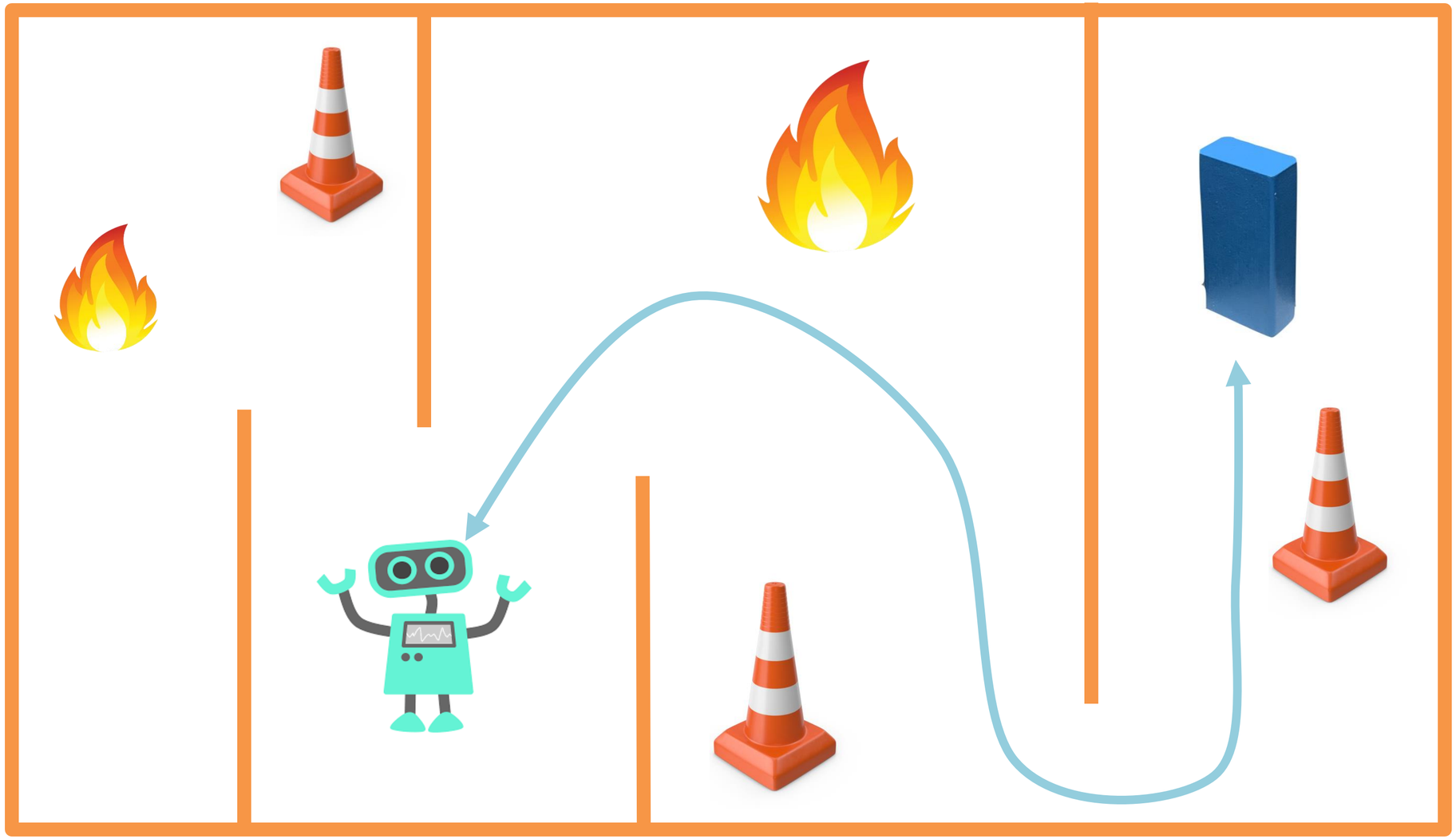
Step 5
Return



Step 6
Put down

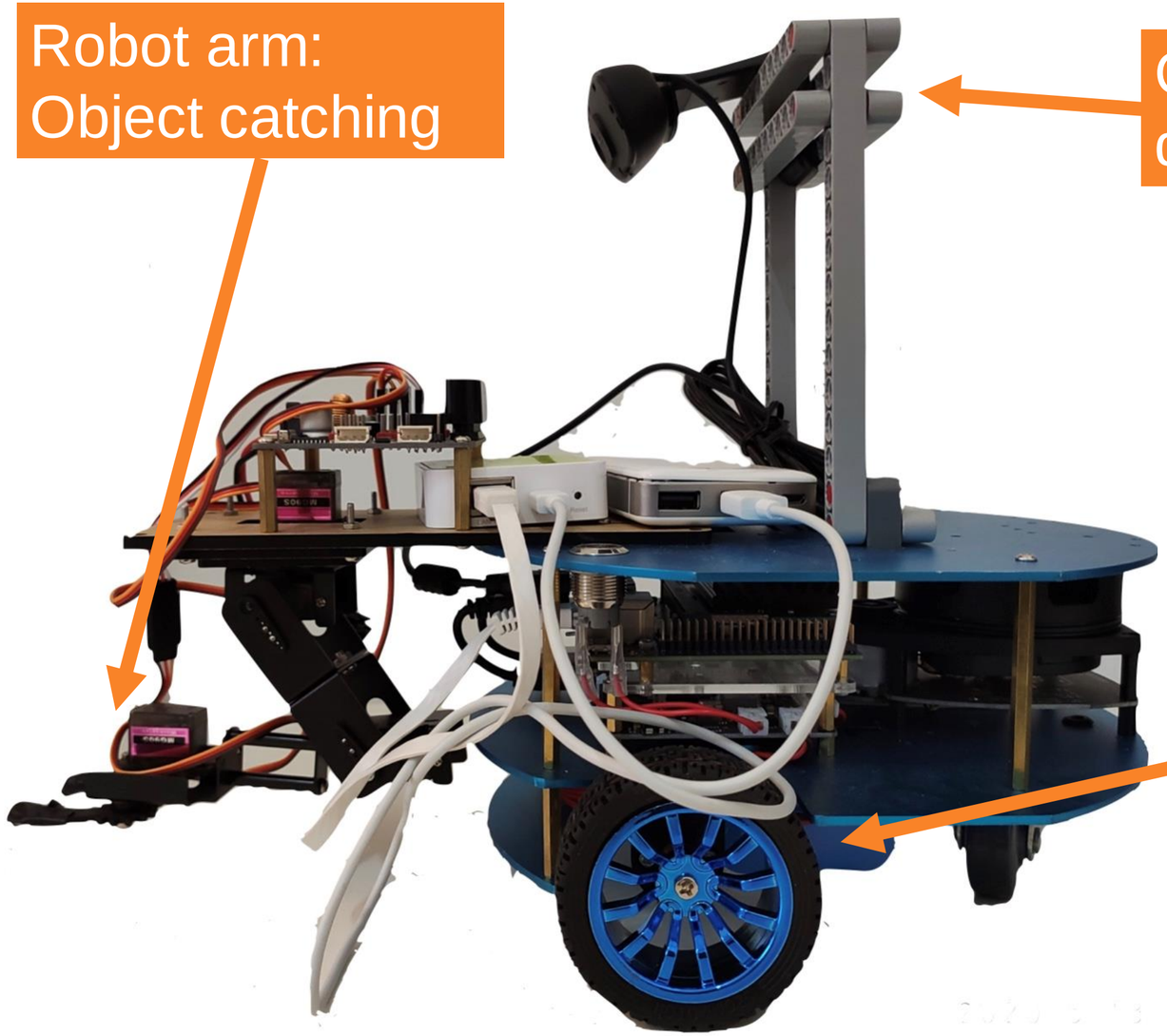


Simplification



RESCUE ROBOT SENARIO

Component


Robot arm:
Object catching
Camera: Object
detection and tracking
Robot base with
lidar: SLAM &
Object tracking

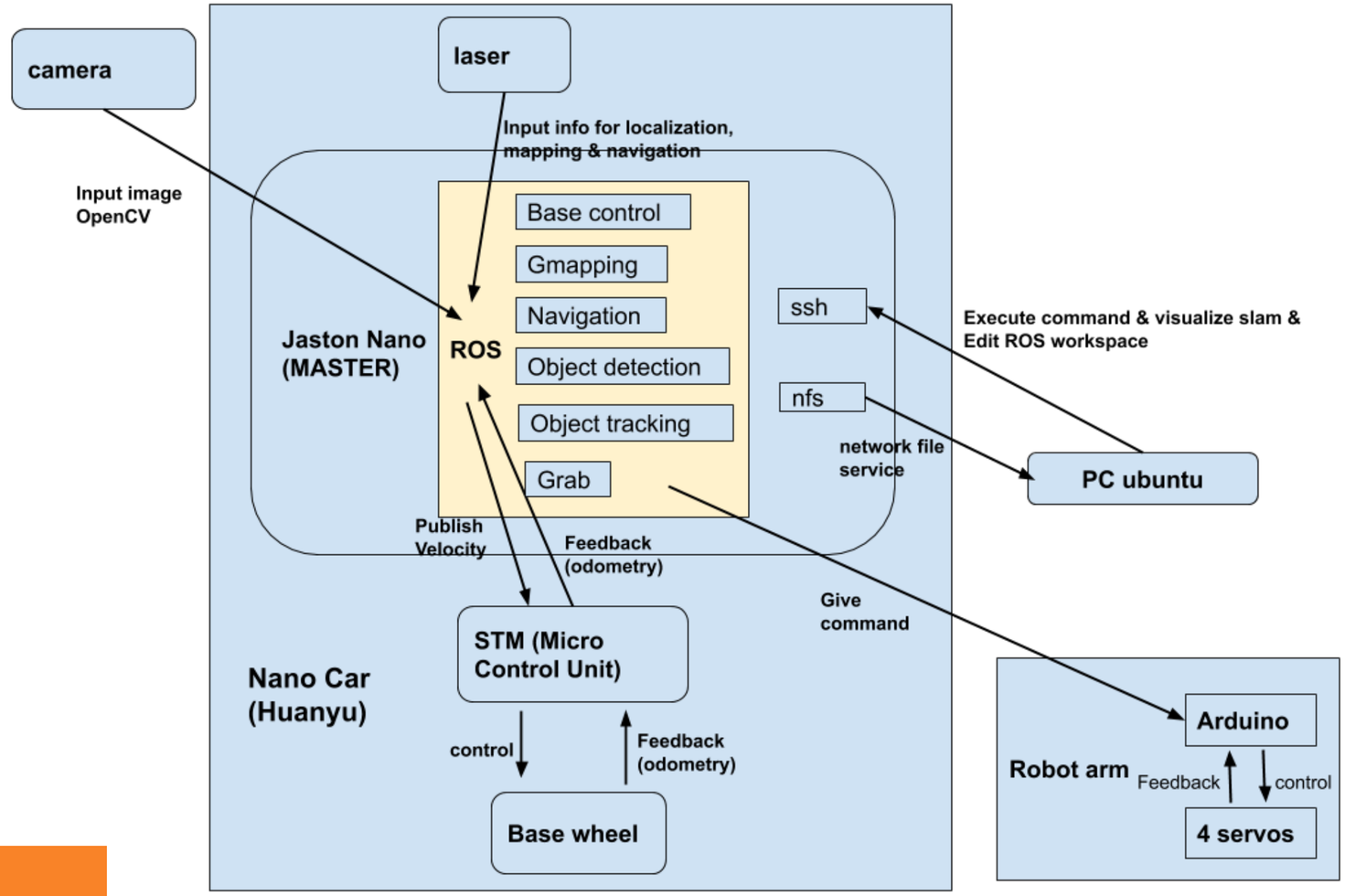
Objectives

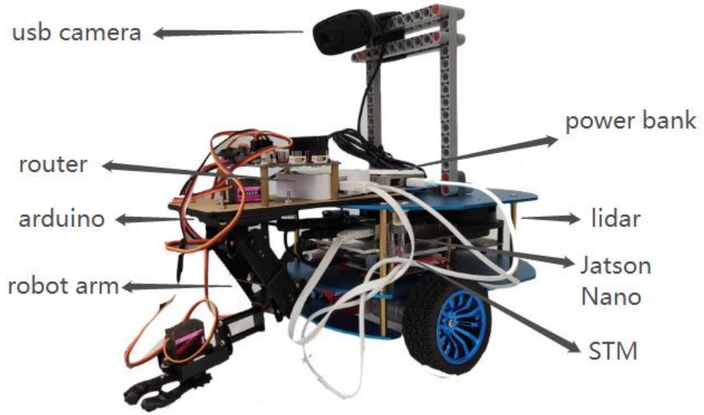
- SLAM:
  Mapping and Navigation
- Object Detection and Tracking
- Control of Robot Arm
- Integration

02

Technical content

Project Framework





PHYSICAL DESIGN

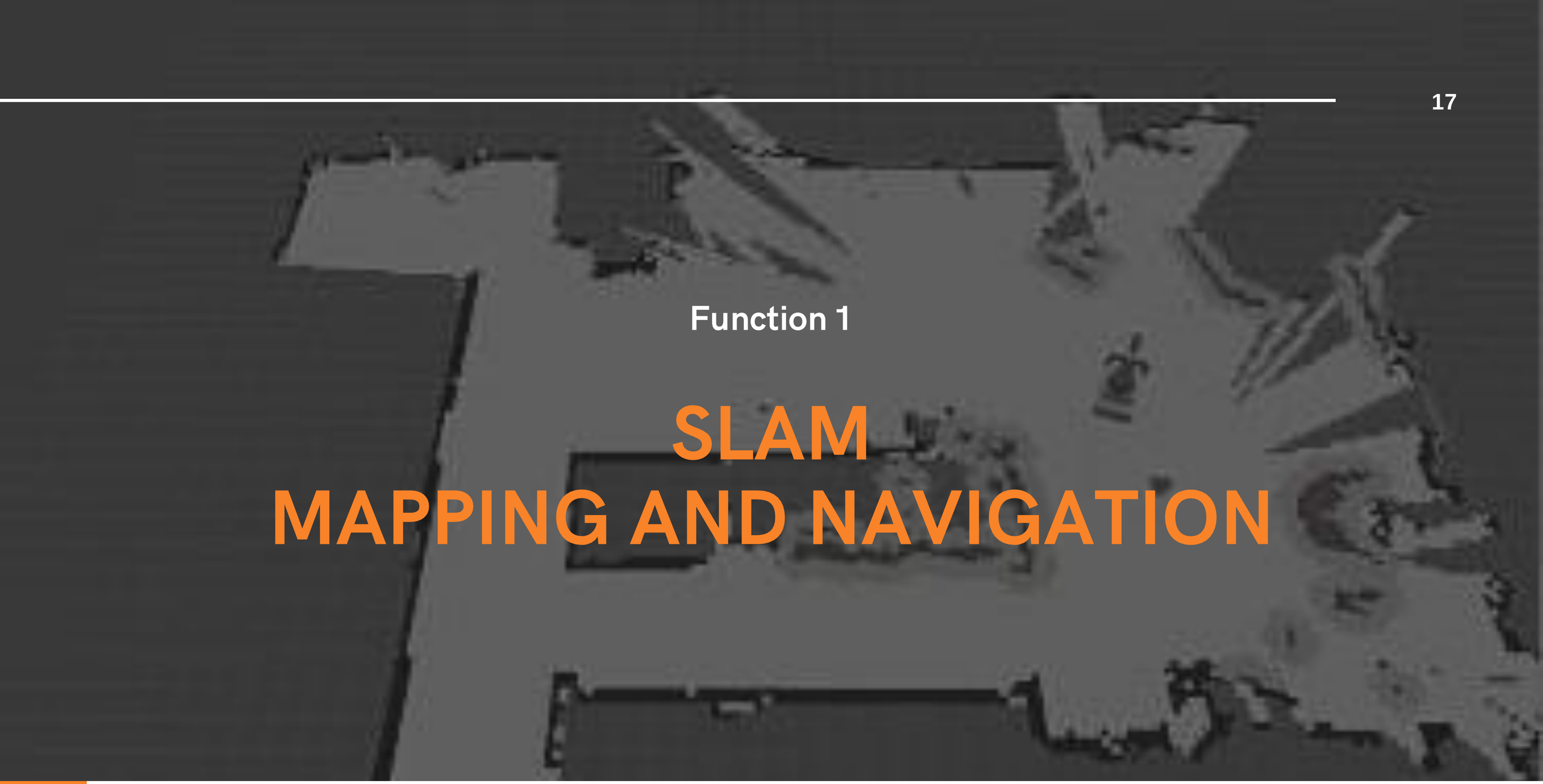
Communication in ROS

Message

```
geometry_msgs/Vector3 linear
  float64 x
  float64 y
  float64 z
geometry_msgs/Vector3 angular
  float64 x
  float64 y
  float64 z
```

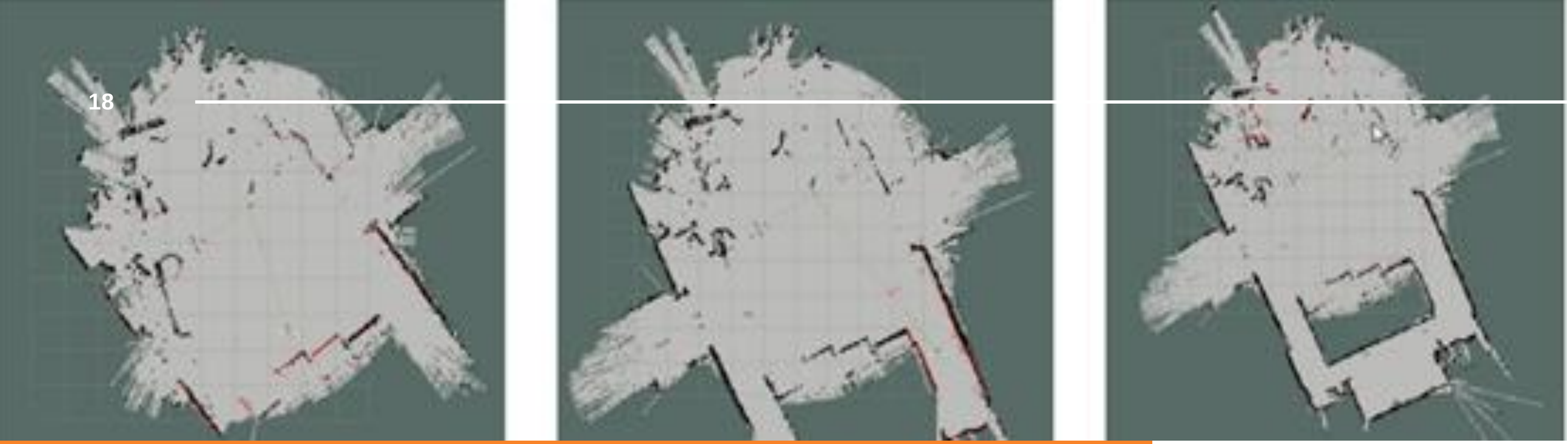
Functions:

1. SLAM: Mapping & Navigation
2. Object Detection & Tracking
3. Robot Arm control



Function 1

SLAM
MAPPING AND NAVIGATION

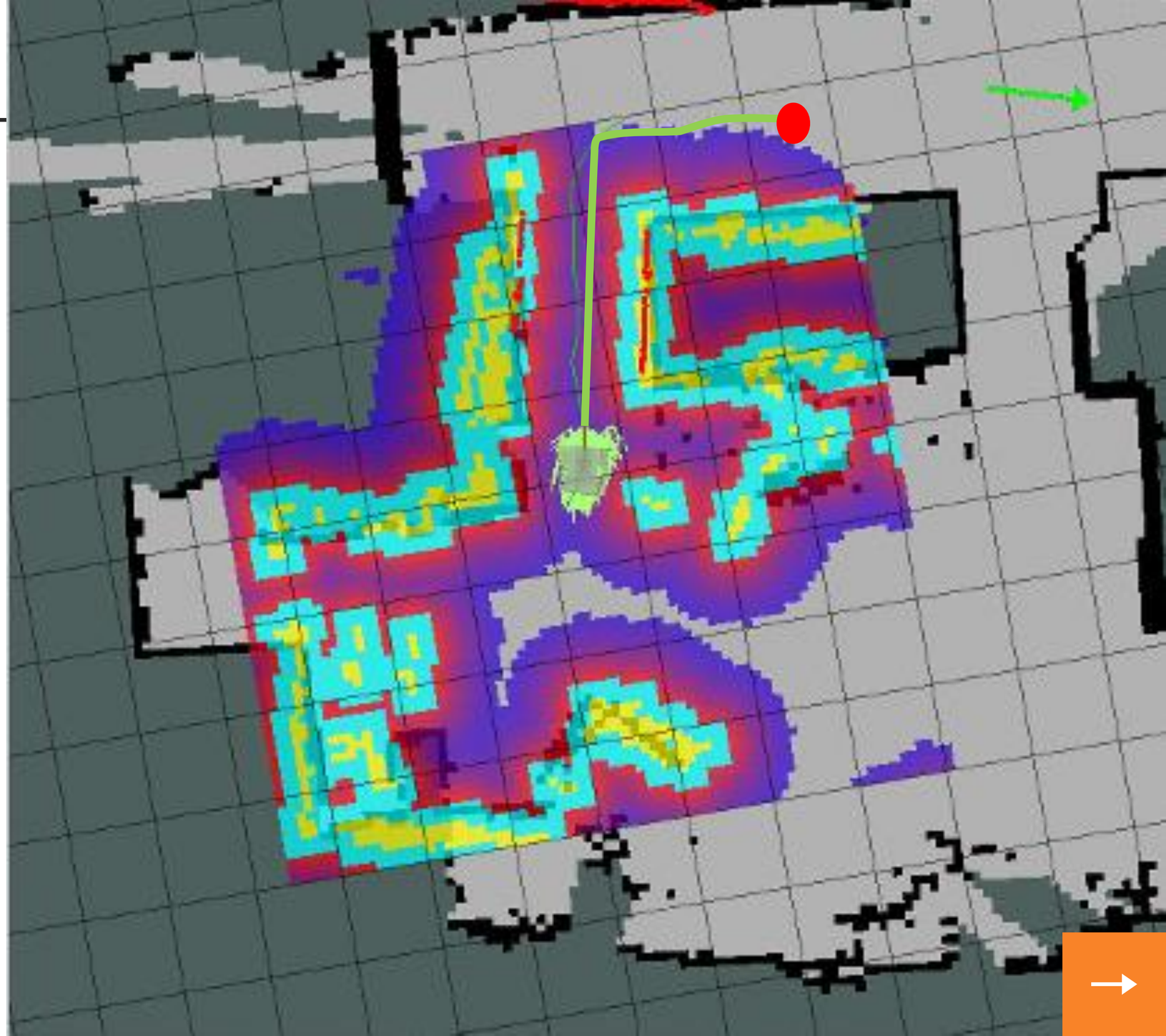


Mapping (Gmapping)

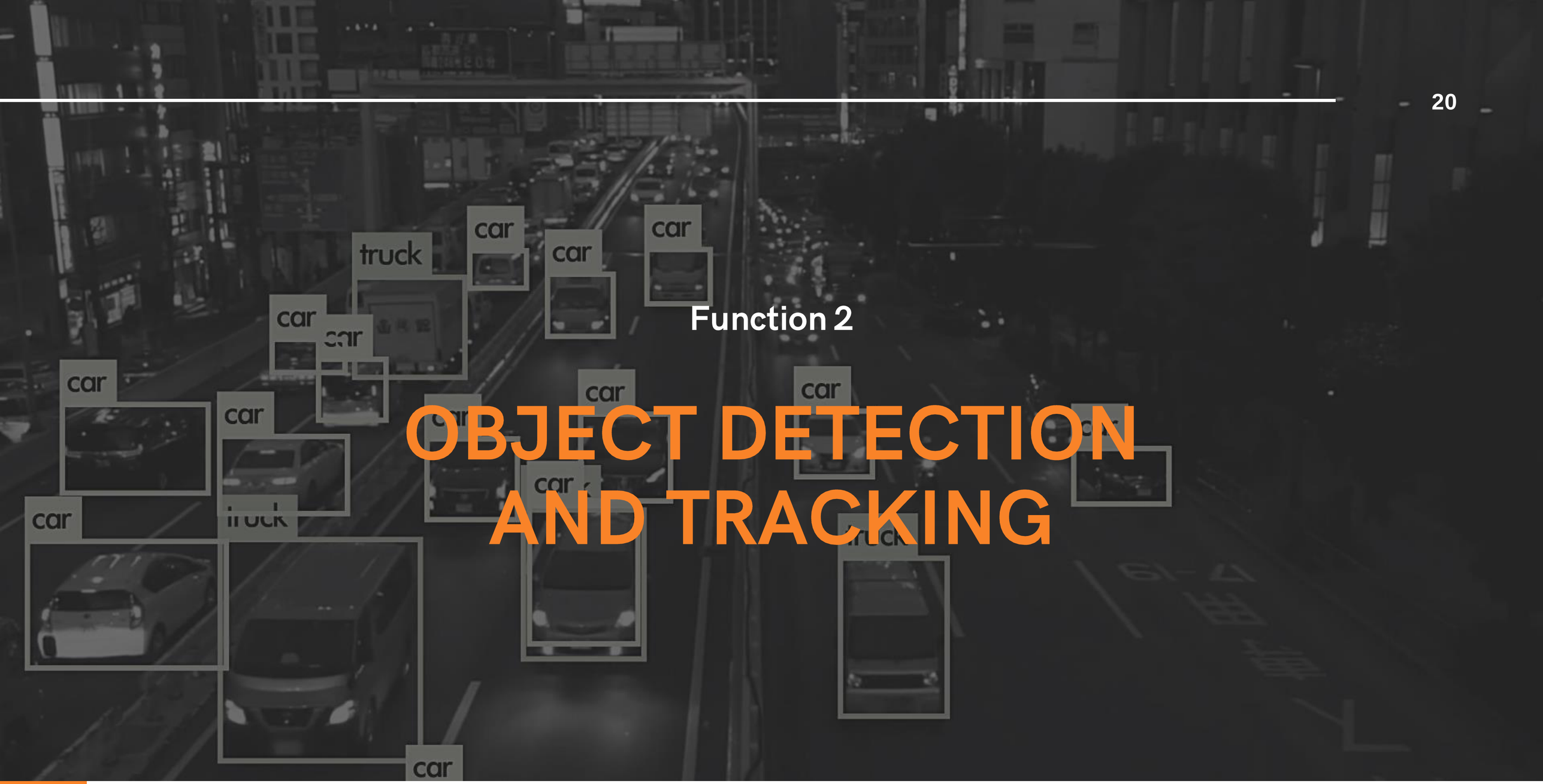
Visualization in Rviz

Navigation

1. Receive goals for navigation
2. Publish velocity to base wheel
3. Receive request to pause navigation

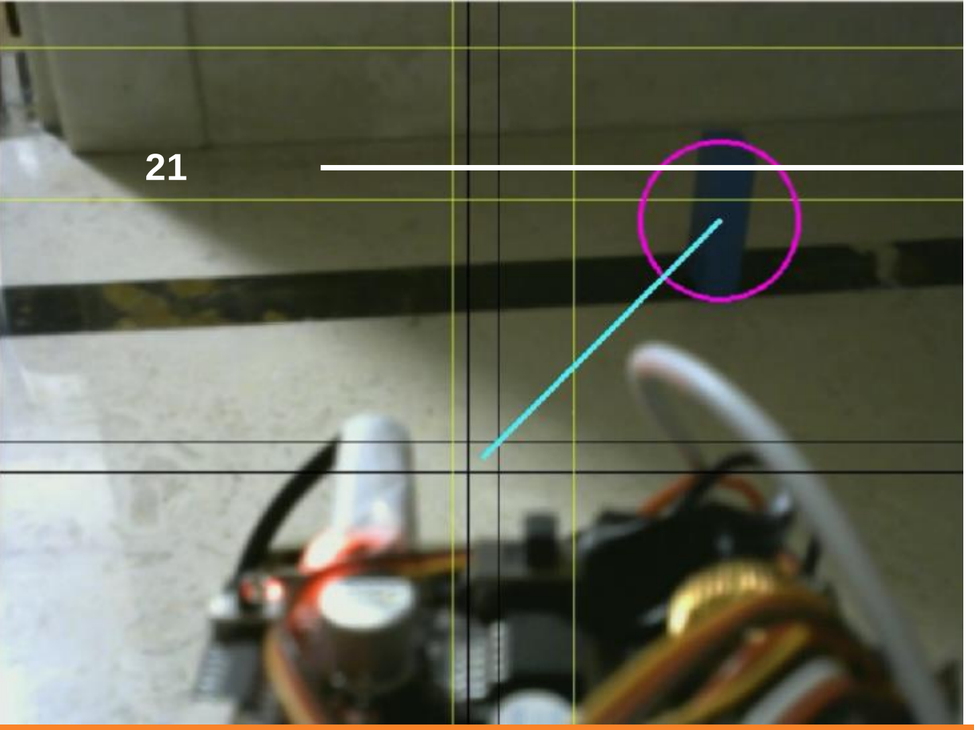


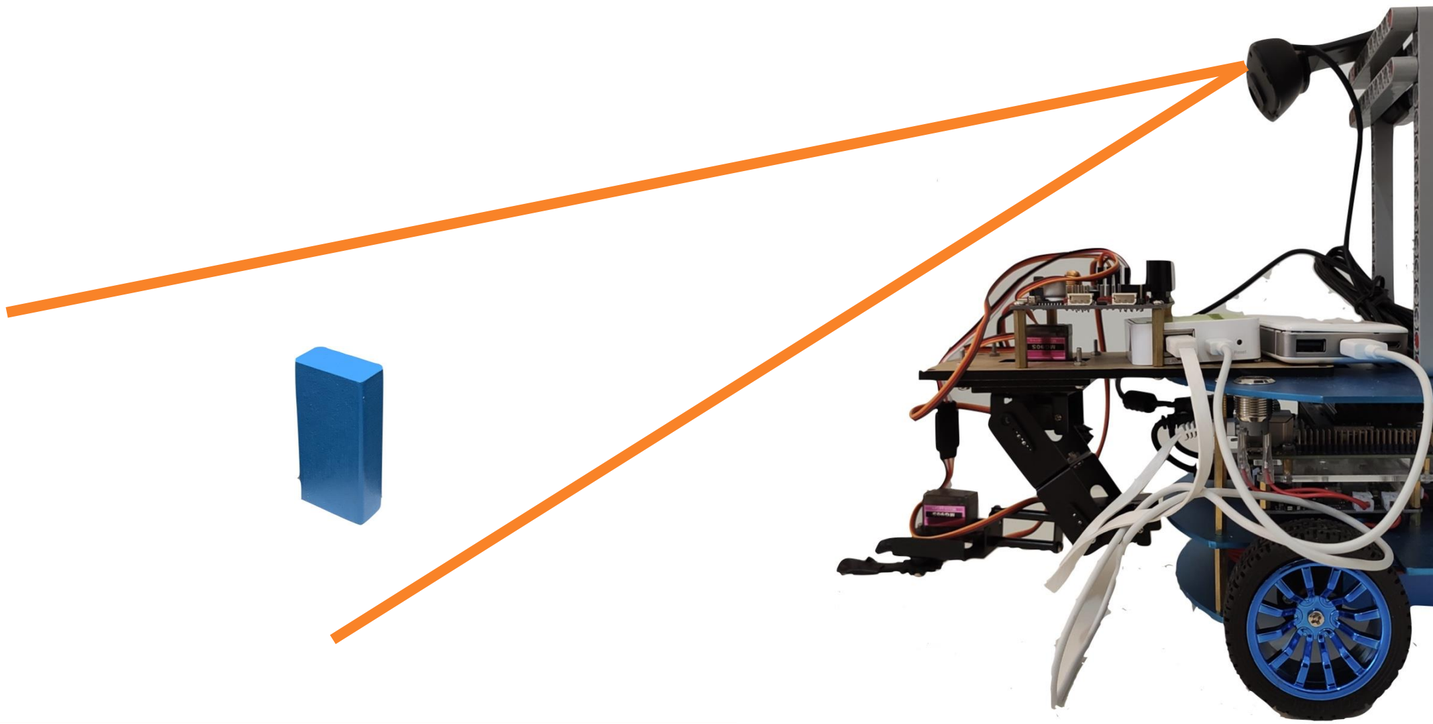
PACKAGE MOVE_BASE



Function 2

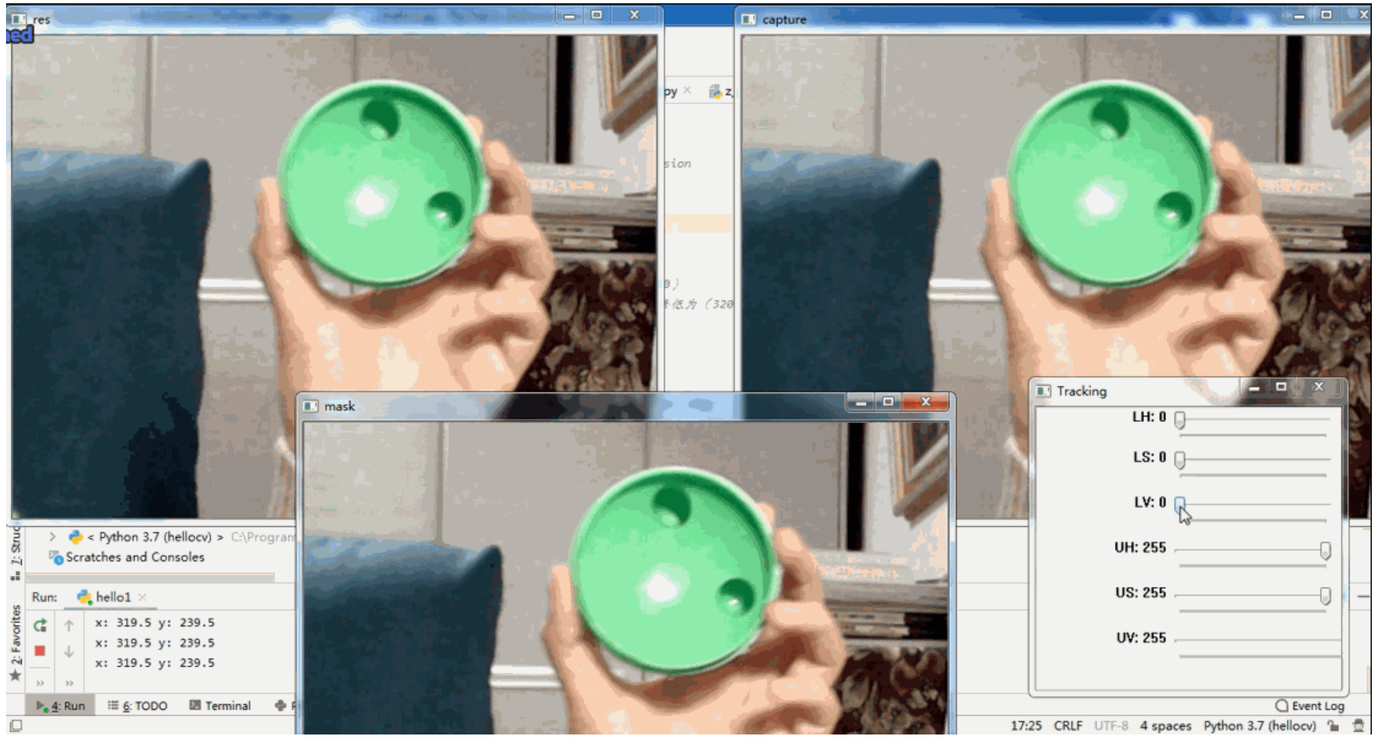
OBJECT DETECTION AND TRACKING





Object Detection and Tracking

Find the target?



Object Detection

- Good Compatibility
- Simple

Step 1: Transfer RGB image to HSV image

Step 2: Color filter

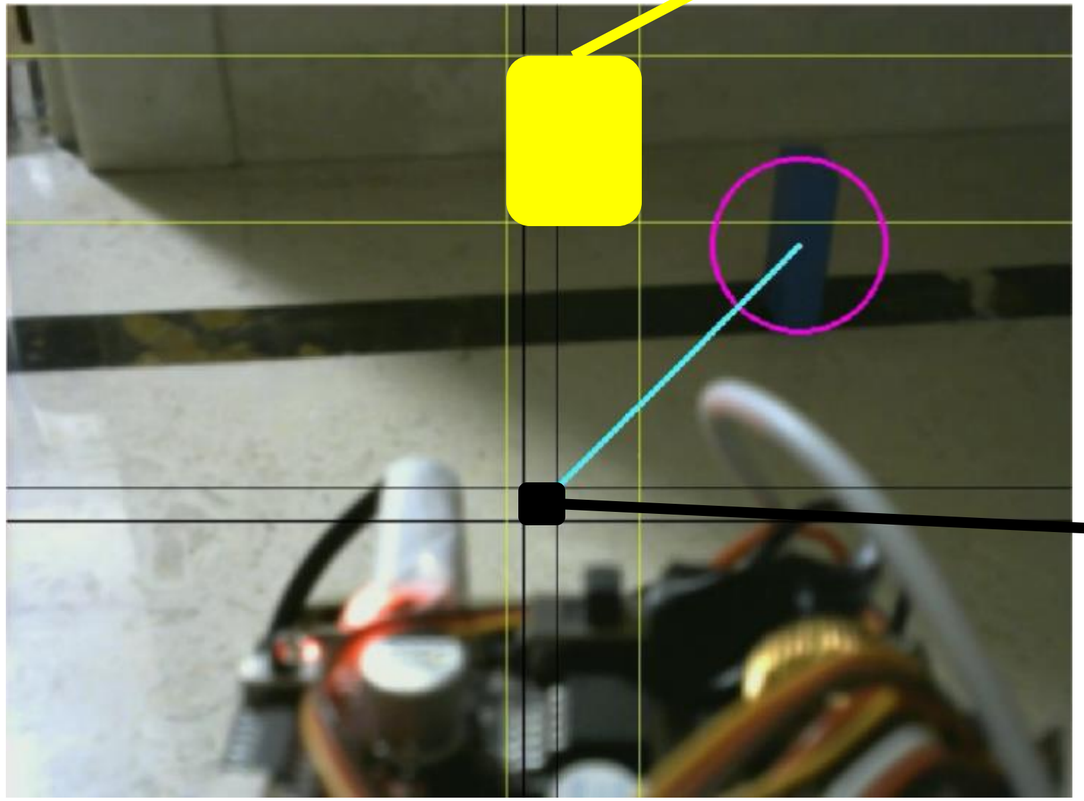
Step 3: Eliminate noise

Step 4: Get target center (x,y) location

(x,y) → go_to_center → velocity

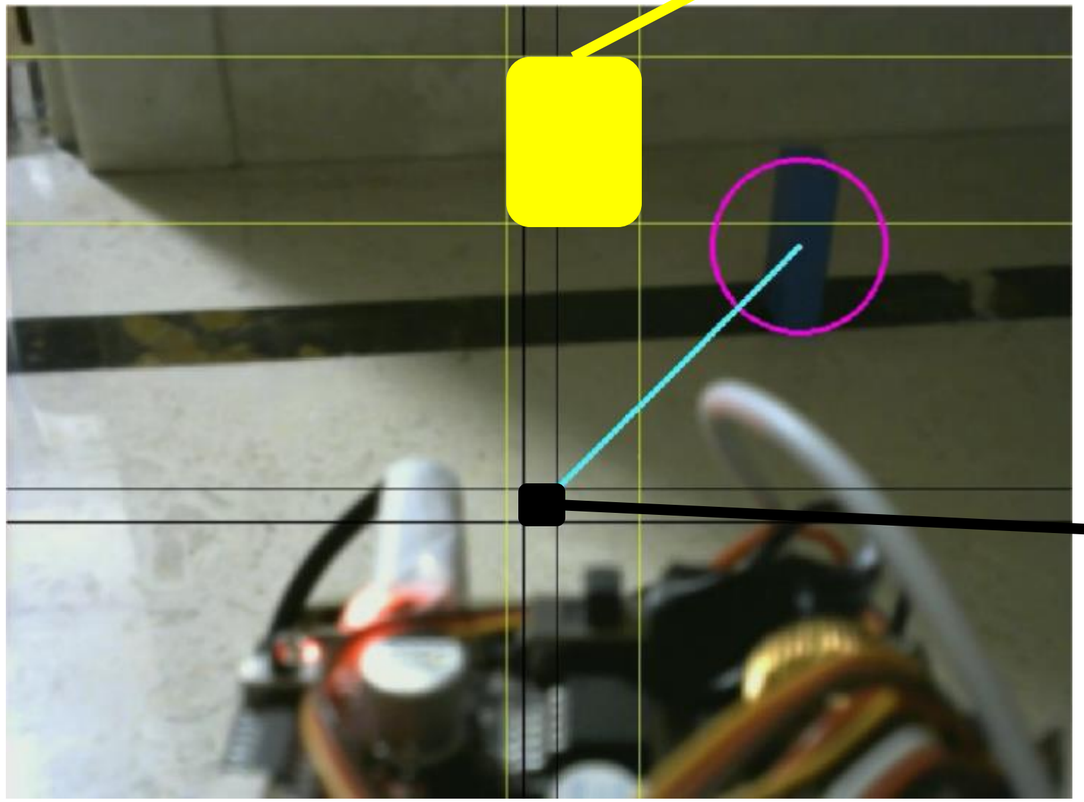
Object Tracking

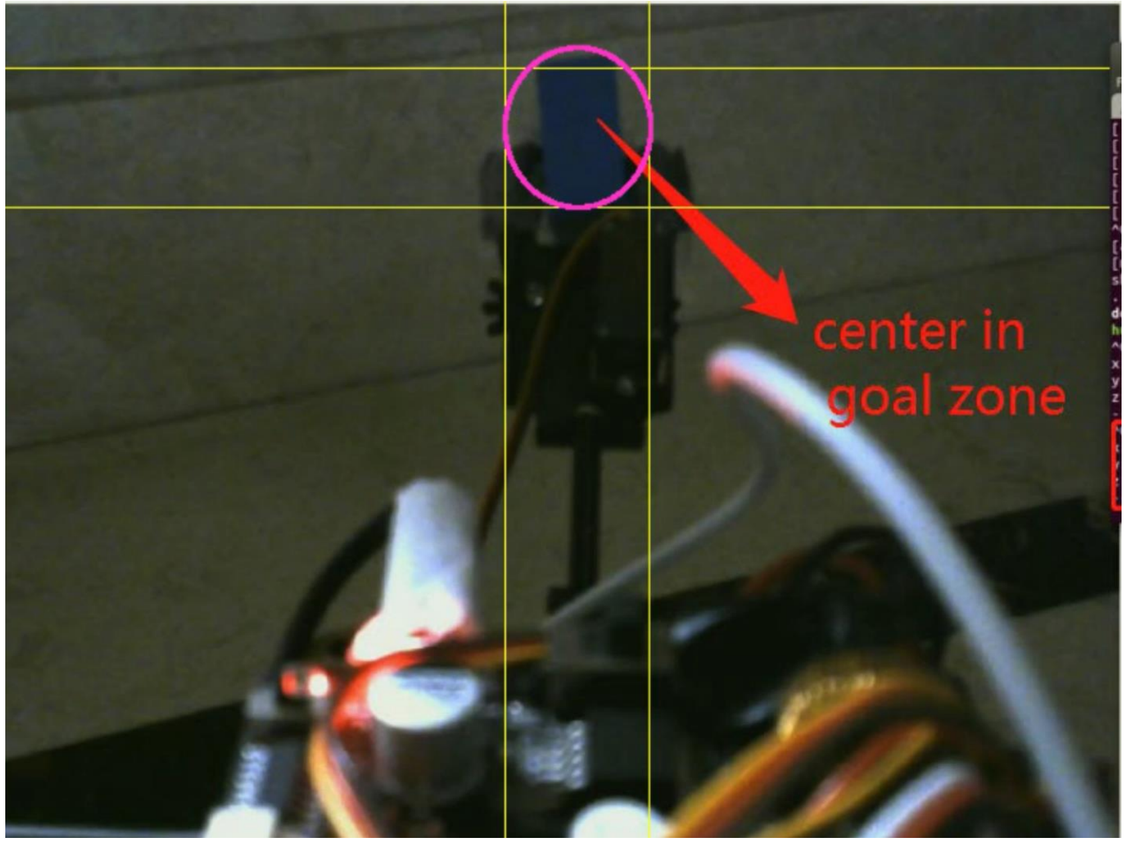
(x,y) → go_to_center → velocity



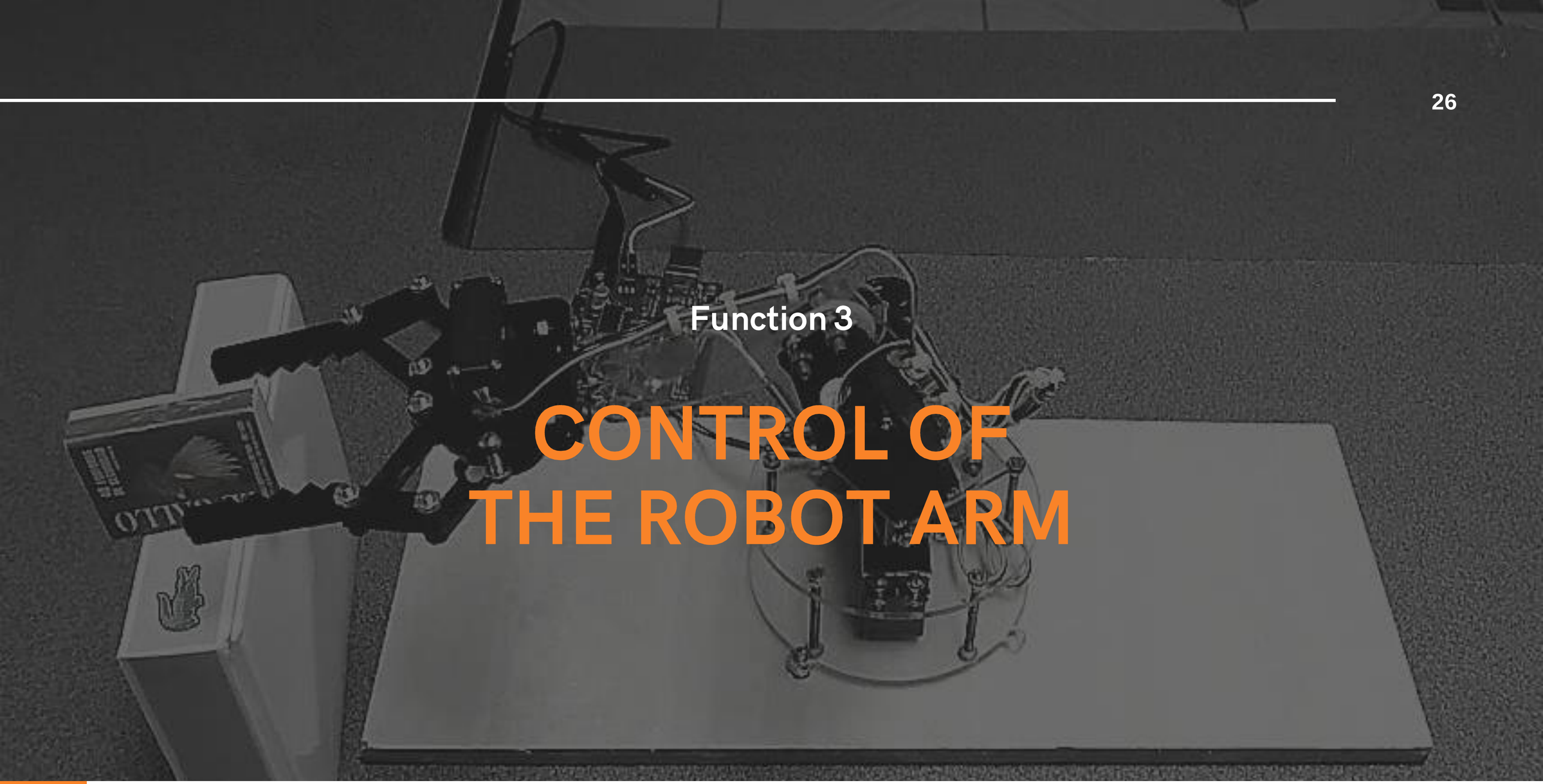
Object Tracking

(x,y) → go_to_center → velocity





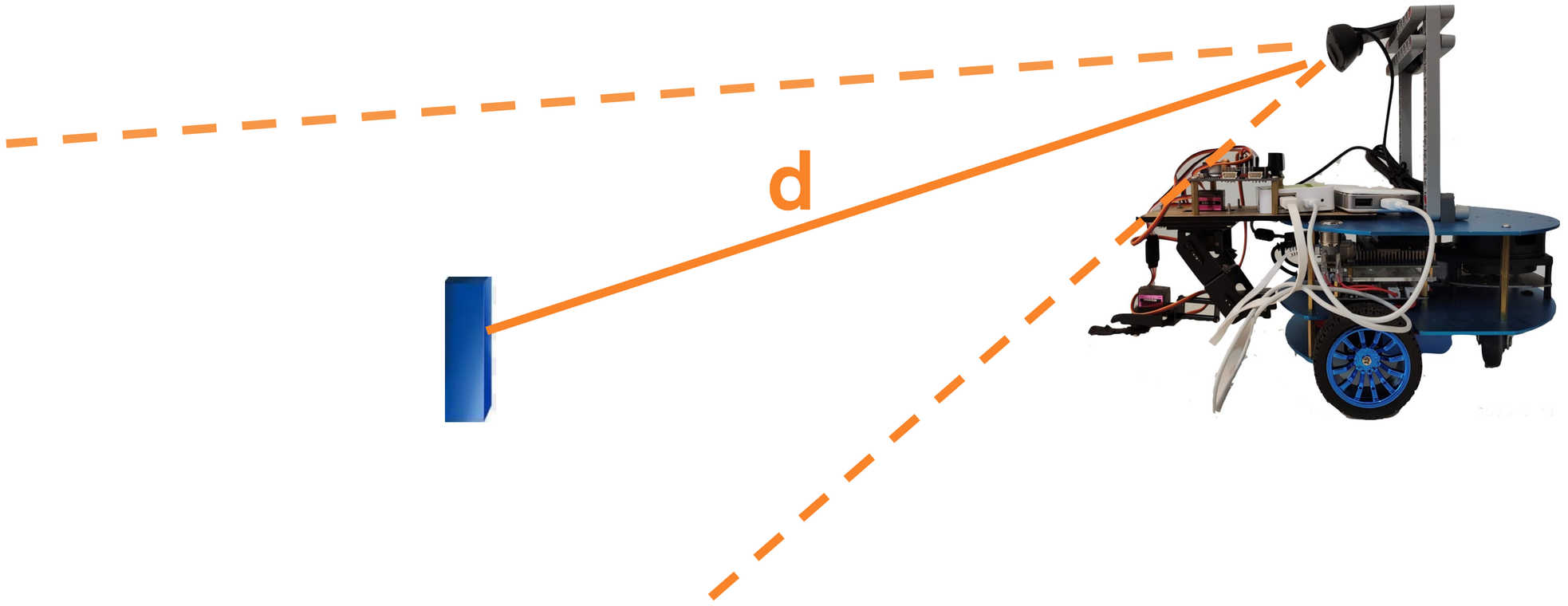
Object Tracking

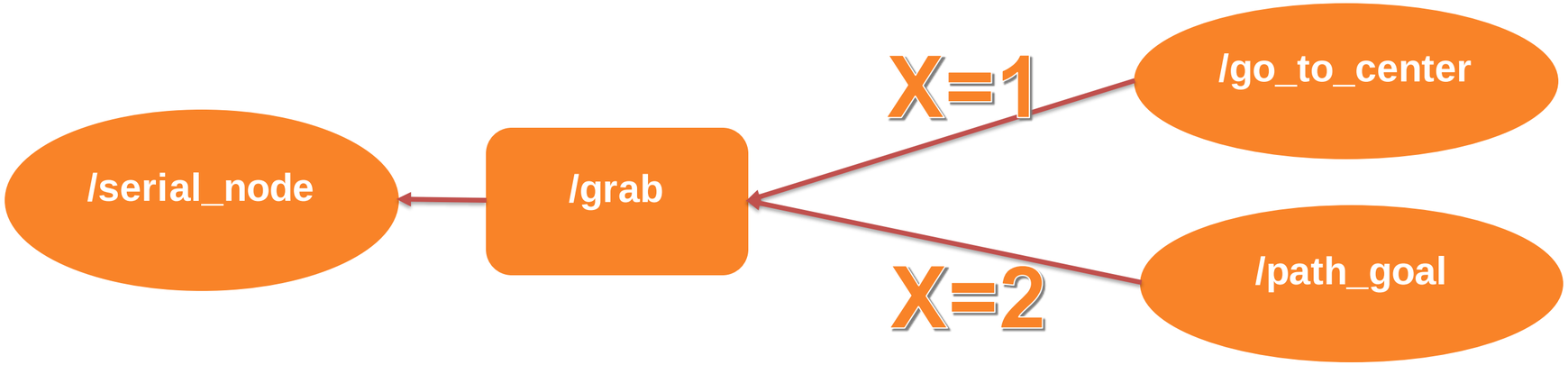


Function 3

CONTROL OF THE ROBOT ARM

The distance between the object to the camera d is fixed





Rosserial

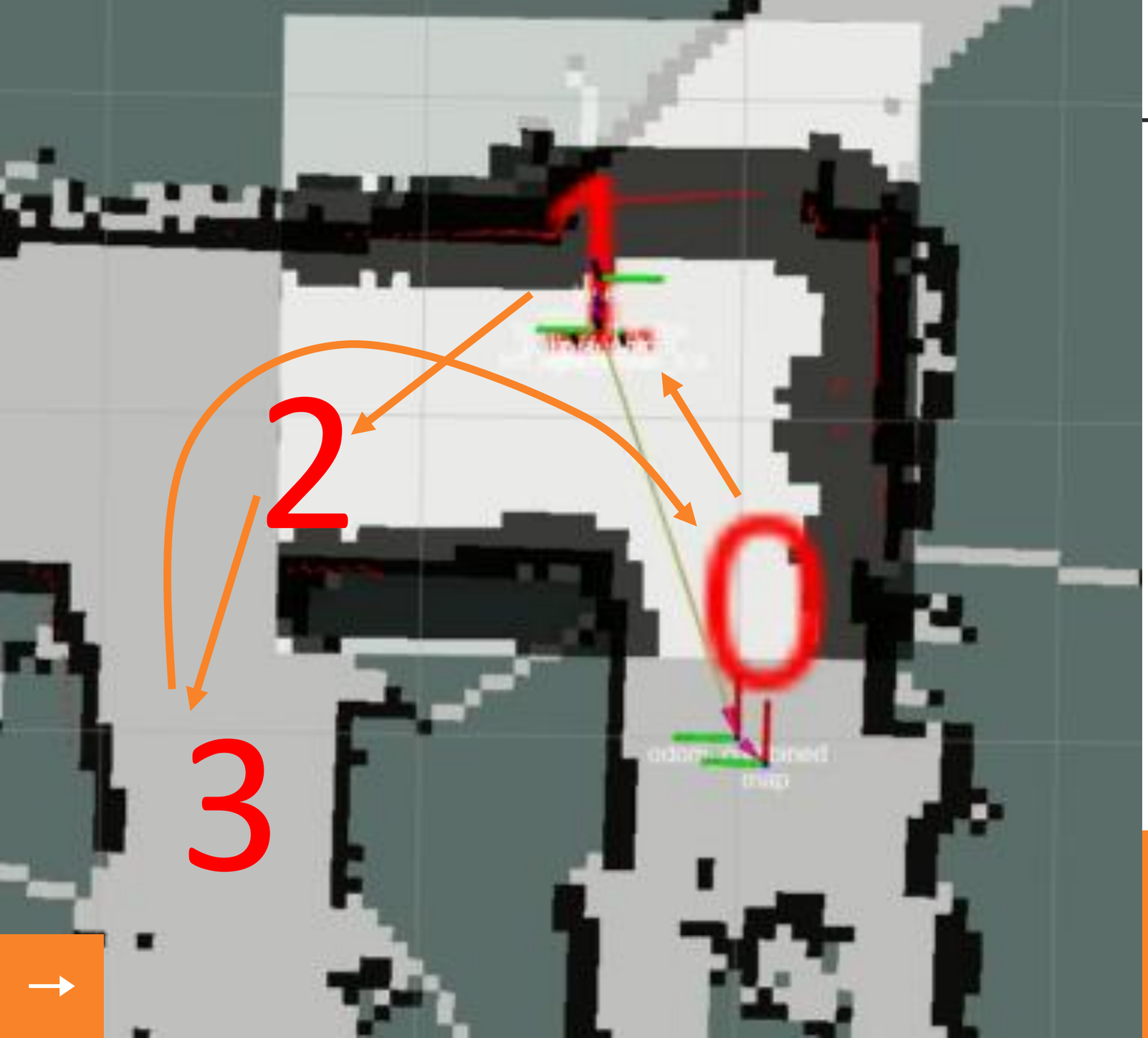
```
ros::Subscriber<geometry_msgs::Point> sub("grab", move_or_not);
void move_or_not(const geometry_msgs::Point &indicator){

  if(indicator.x == 1){
    severalServoControl();
    delay(5000);
  }
  if(indicator.x ==2){
    putdownServoControl();
    delay(5000);
  }
}
```

CONNECT ARDUINO TO ROS

RESCUE MISSION

INTEGRATION

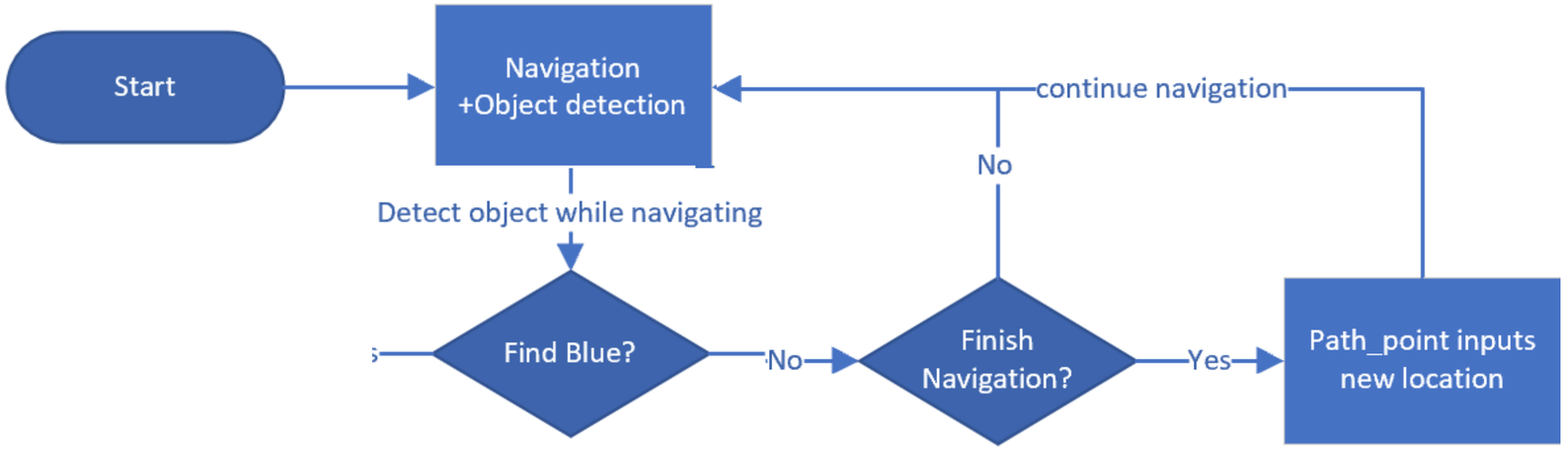


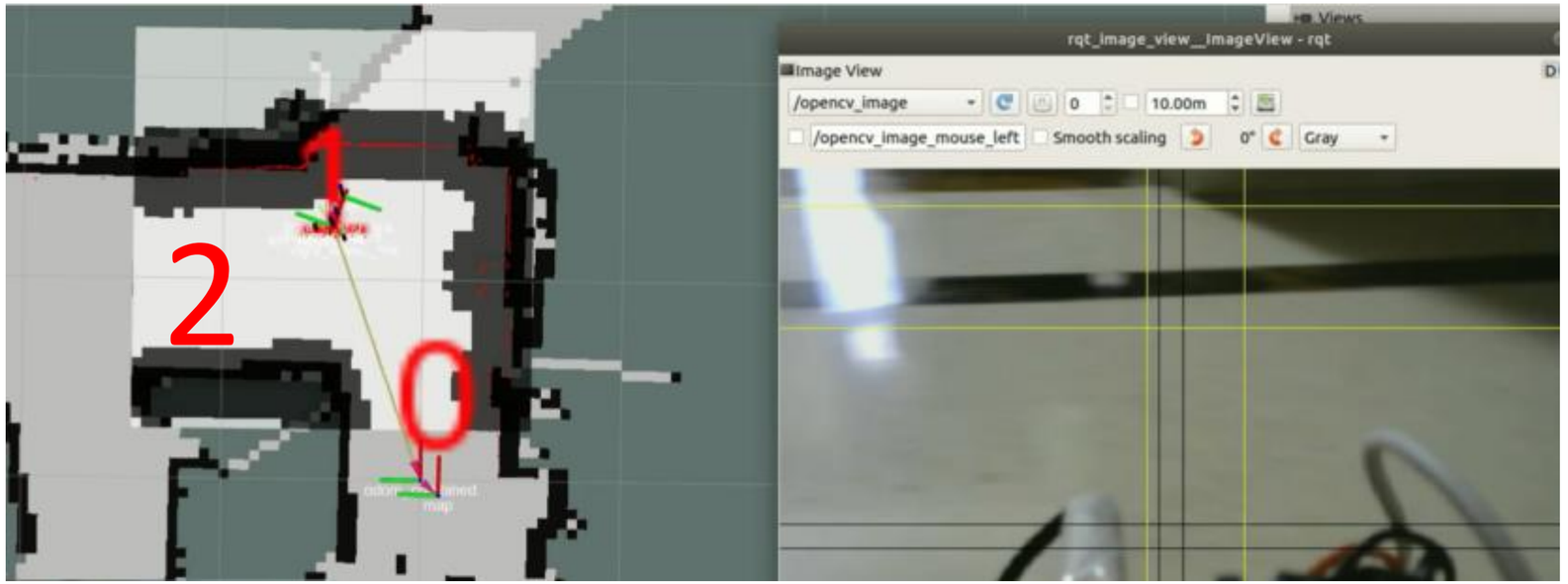
Path goal

1. Record initial point (0) and search points (1,2,3…) in Rviz
2. Publish these search points
3. Publish initial position if grab finish
4. Put down the object after return

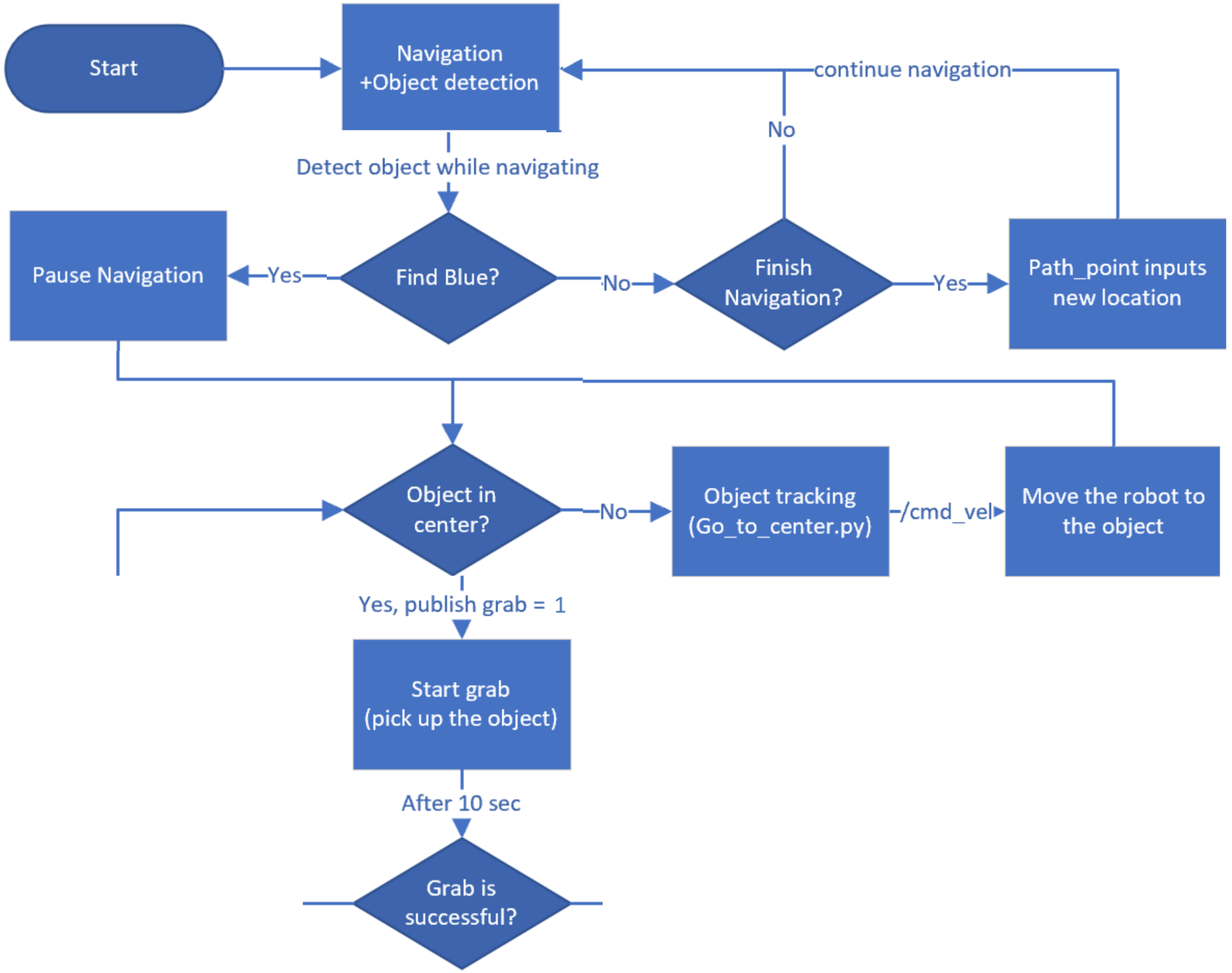
CONNECT NAVIGATION WITH OBJECT TRACKING

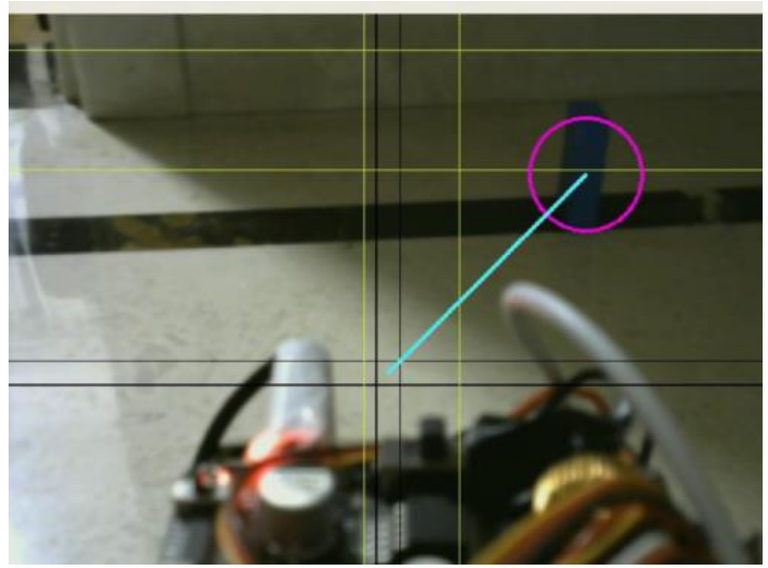
Logic Flow


Start
Navigation
+Object detection
continue navigation
No
Detect object while navigating
Find Blue?
No
Finish
Navigation?
Yes
Path_point inputs
new location

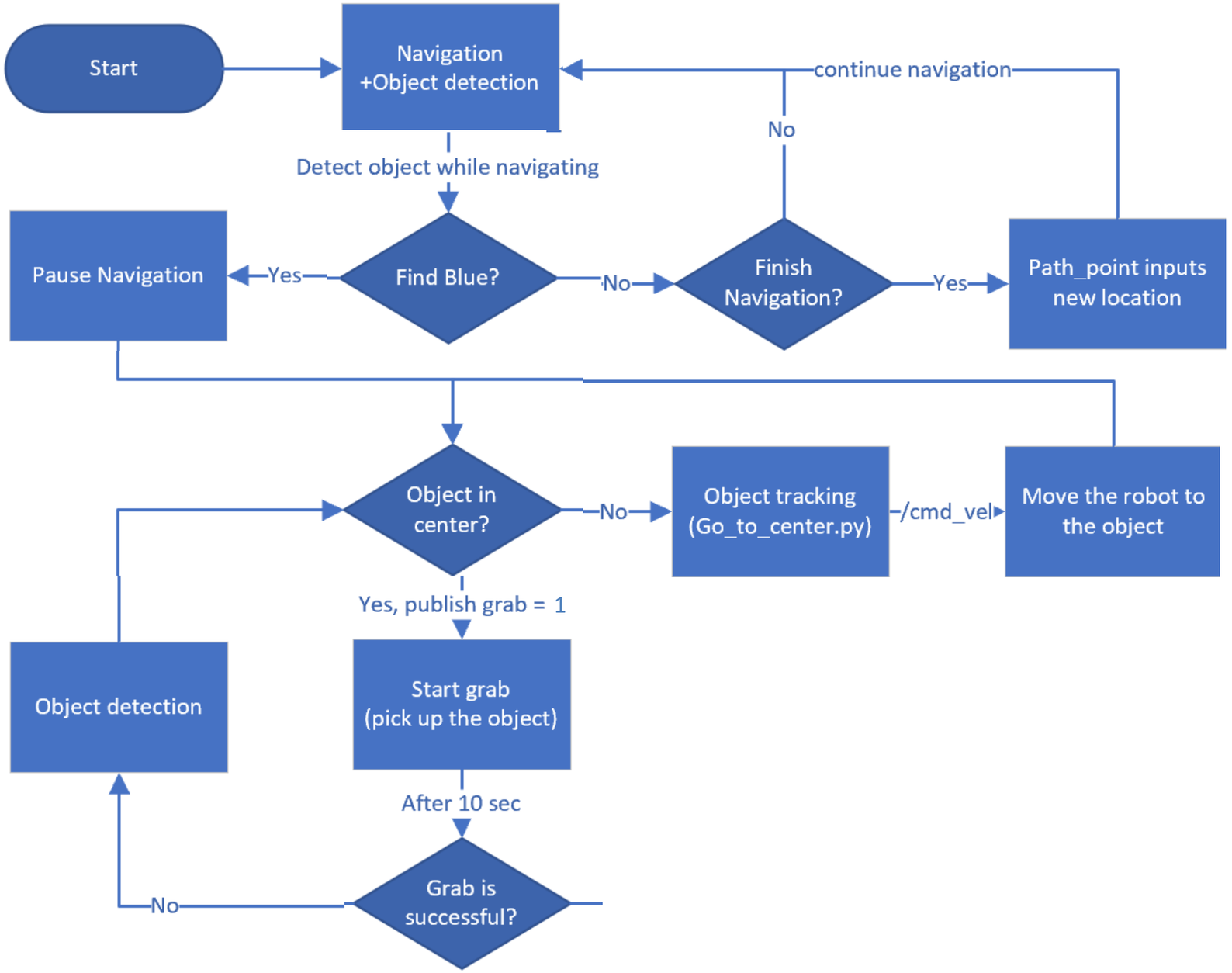

rqt_image_view__imageView - rqt
Image View
/opencv_image
/opencv_image_mouse_left
Smooth scaling
Gray
1
2
0

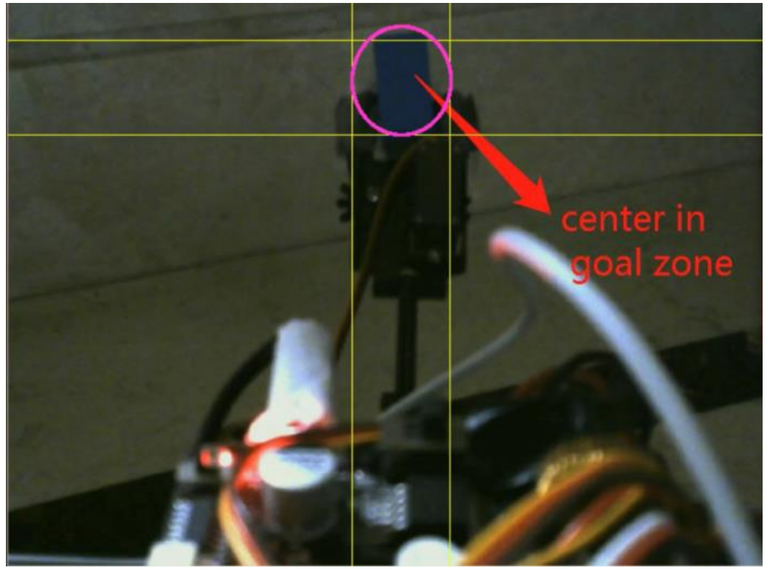
Logic Flow


Start
Navigation
+Object detection
continue navigation
No
Detect object while navigating
Pause Navigation
Yes
Find Blue?
No
Finish
Navigation?
Yes
Path_point inputs
new location
Object in
center?
No
Object tracking
(Go_to_center.py)
/cmd_vel
Move the robot to
the object
Yes, publish grab = 1
Start grab
(pick up the object)
After 10 sec
Grab is
successful?

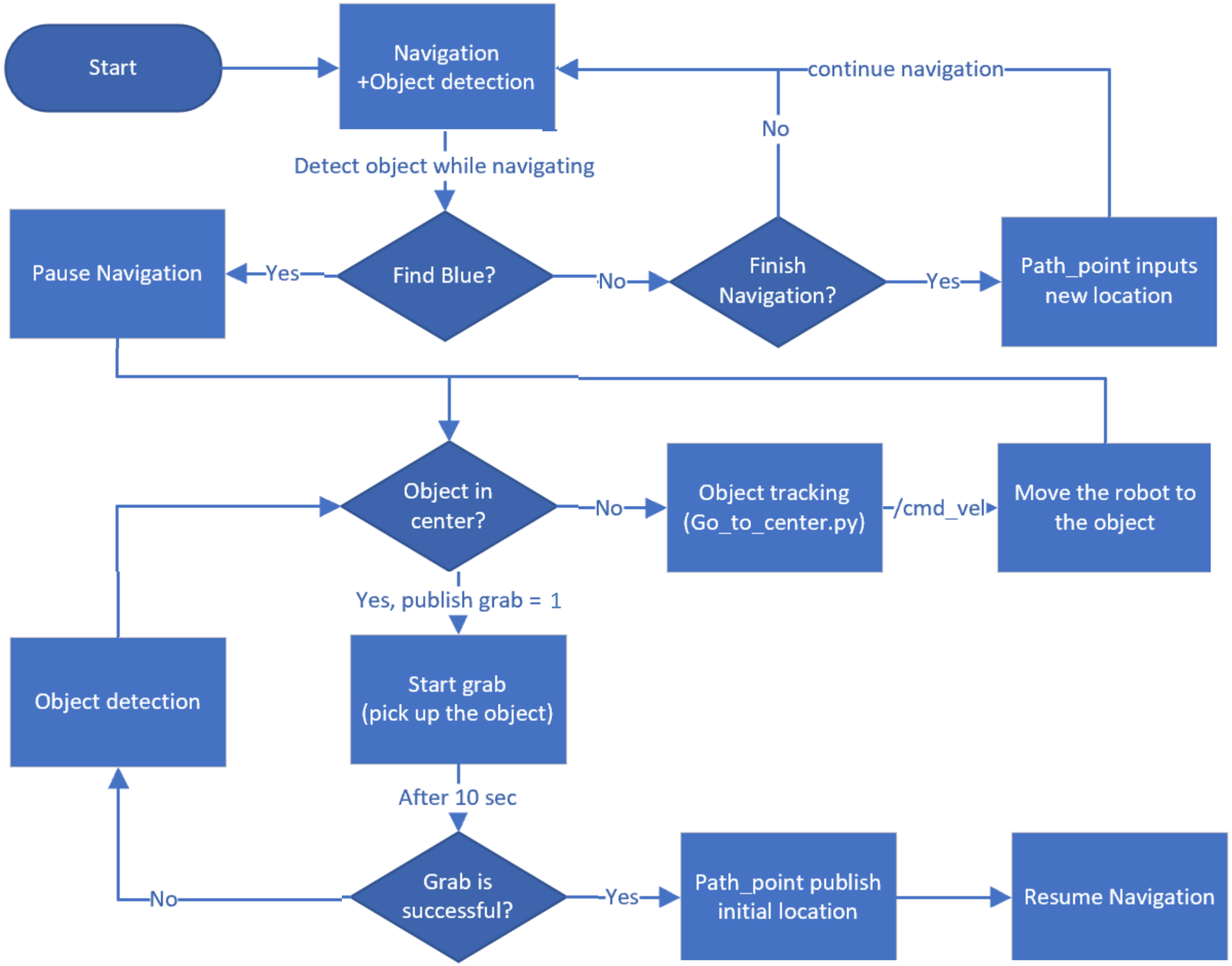


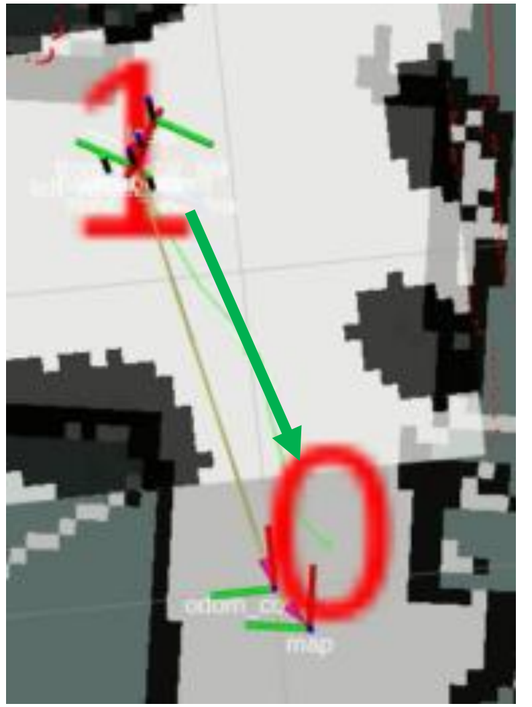
Logic Flow


Start
Navigation +Object detection
continue navigation
No
Detect object while navigating
Pause Navigation
Yes
Find Blue?
No
Finish Navigation?
Yes
Path_point inputs new location
Object in center?
No
Object tracking (Go_to_center.py)
/cmd_vel
Move the robot to the object
Yes, publish grab = 1
Object detection
Start grab (pick up the object)
After 10 sec
No
Grab is successful?

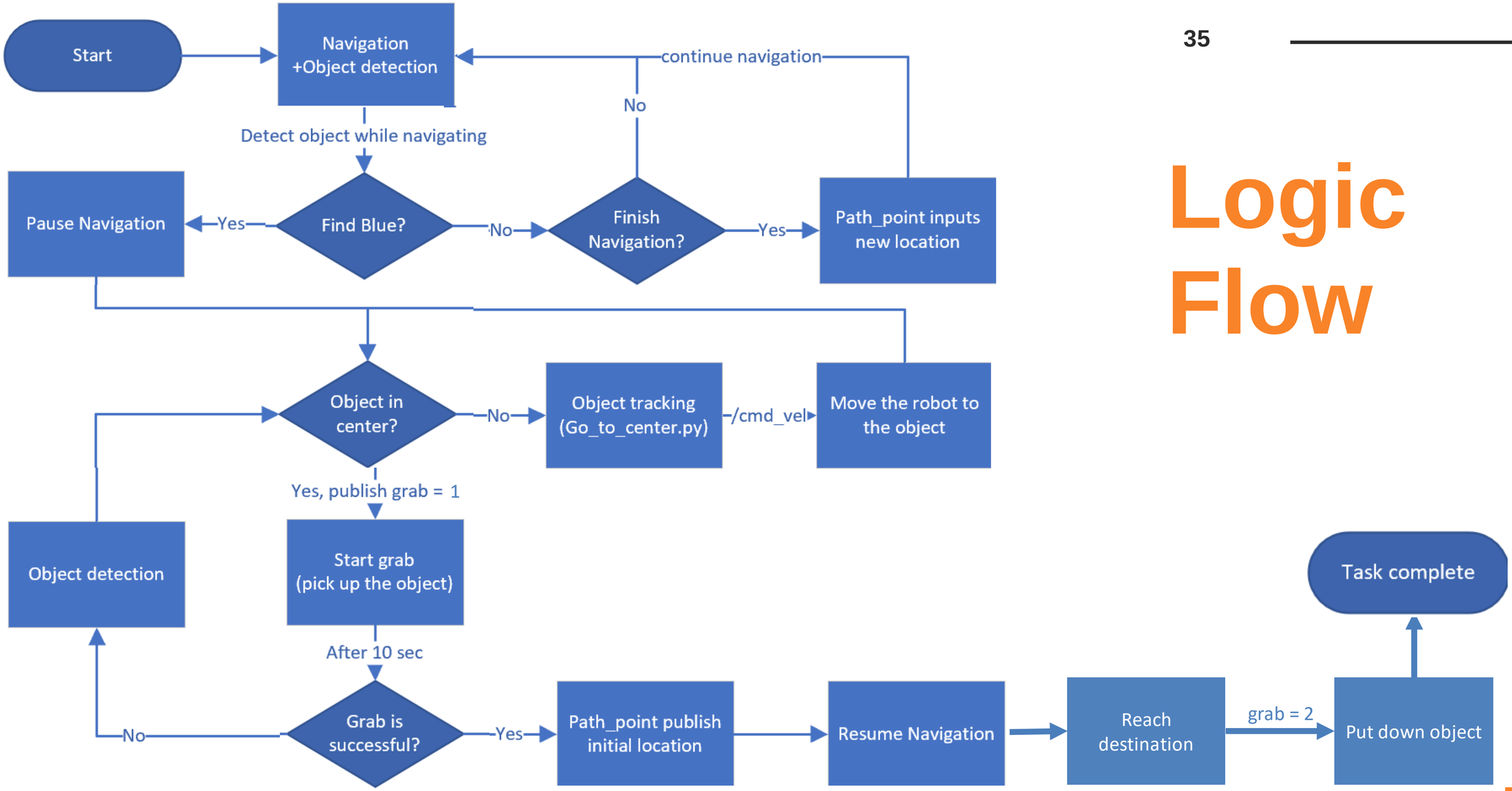

center in goal zone

Logic Flow


Start
Navigation +Object detection
continue navigation
No
Detect object while navigating
Pause Navigation
Yes
Find Blue?
No
Finish Navigation?
Yes
Path_point inputs new location
Object in center?
No
Object tracking (Go_to_center.py)
/cmd_vel
Move the robot to the object
Yes, publish grab = 1
Object detection
Start grab (pick up the object)
After 10 sec
No
Grab is successful?
Yes
Path_point publish initial location
Resume Navigation


1
0

Logic Flow


Start
Navigation +Object detection
continue navigation
No
Detect object while navigating
Pause Navigation
Yes
Find Blue?
No
Finish Navigation?
Yes
Path_point inputs new location
Object in center?
No
Object tracking (Go_to_center.py)
/cmd_vel
Move the robot to the object
Yes, publish grab = 1
Object detection
Start grab (pick up the object)
After 10 sec
Task complete
No
Grab is successful?
Yes
Path_point publish initial location
Resume Navigation
Reach destination
grab = 2
Put down object

Demo Video – Rescue mission

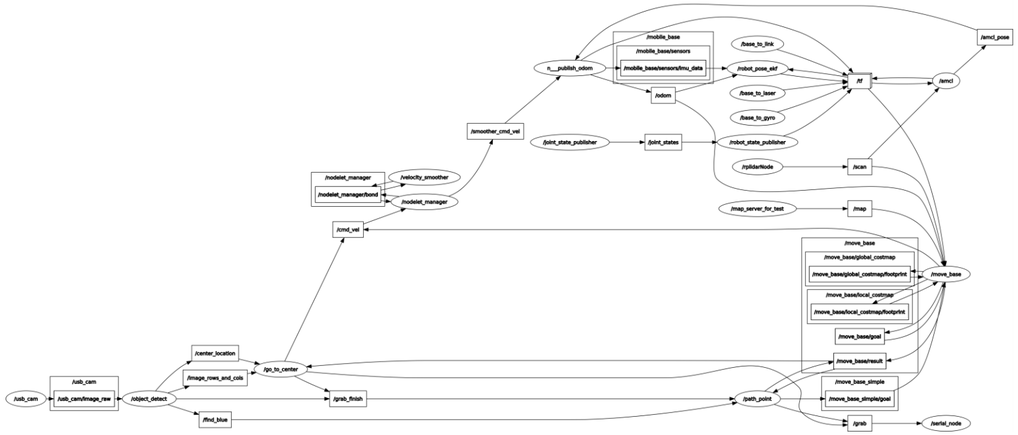
03

Discussion & Conclusion

Result Analysis

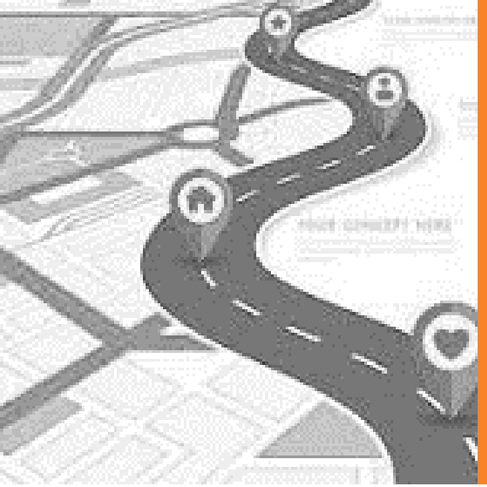
WEAKNESS:

- Architecture is big
- Not stable



SOLUTION:

Manually control



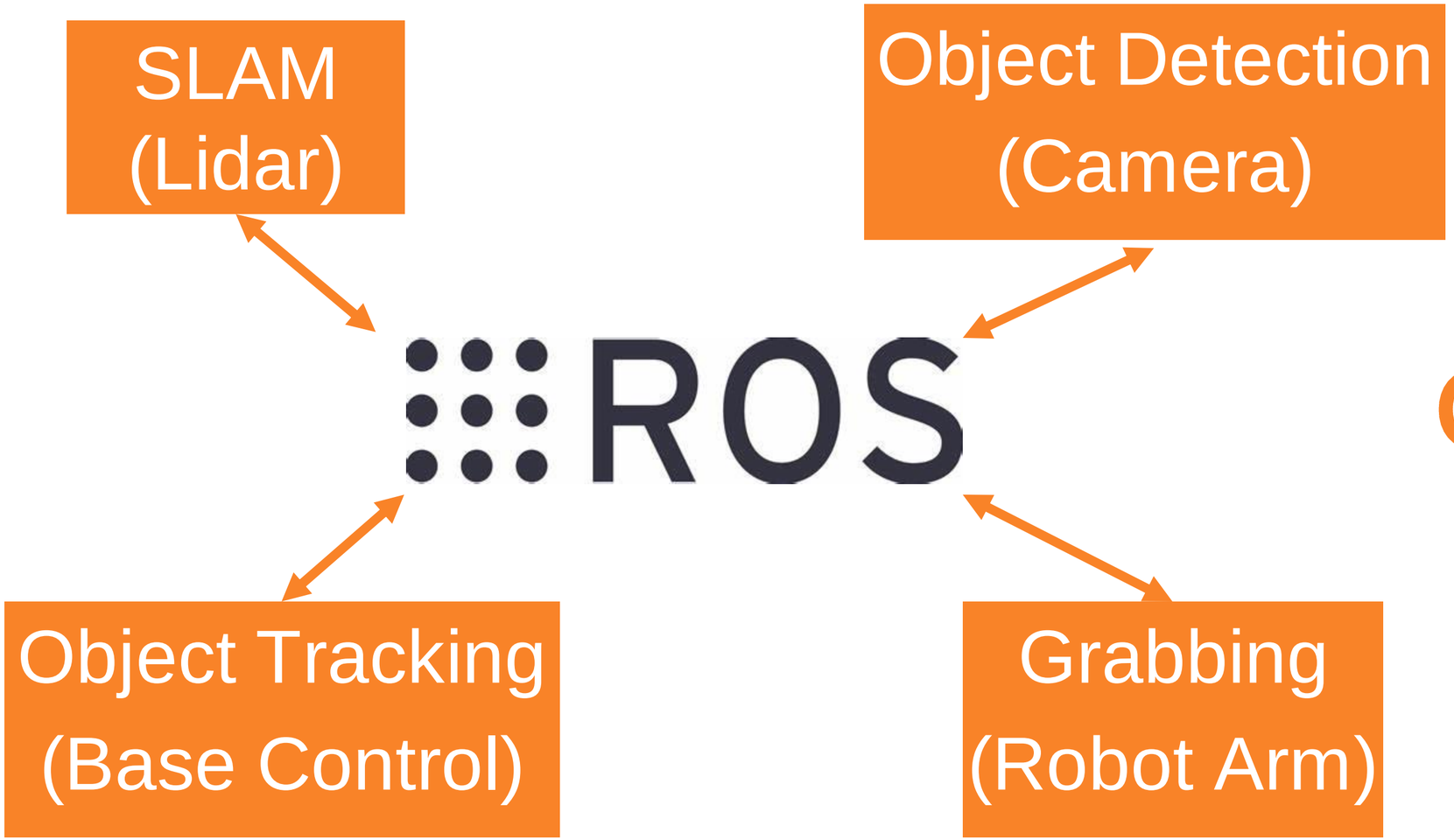
Self-Navigation

Better grabbing feedback

Improved object detection method

Application in logistic

Improvement & Future Development


SLAM
(Lidar)
Object Detection
(Camera)
ROS
Object Tracking
(Base Control)
Grabbing
(Robot Arm)

CONCLUSION

Thank you

Q & A