TUM School of Engineering and Design
Technical University of Munich

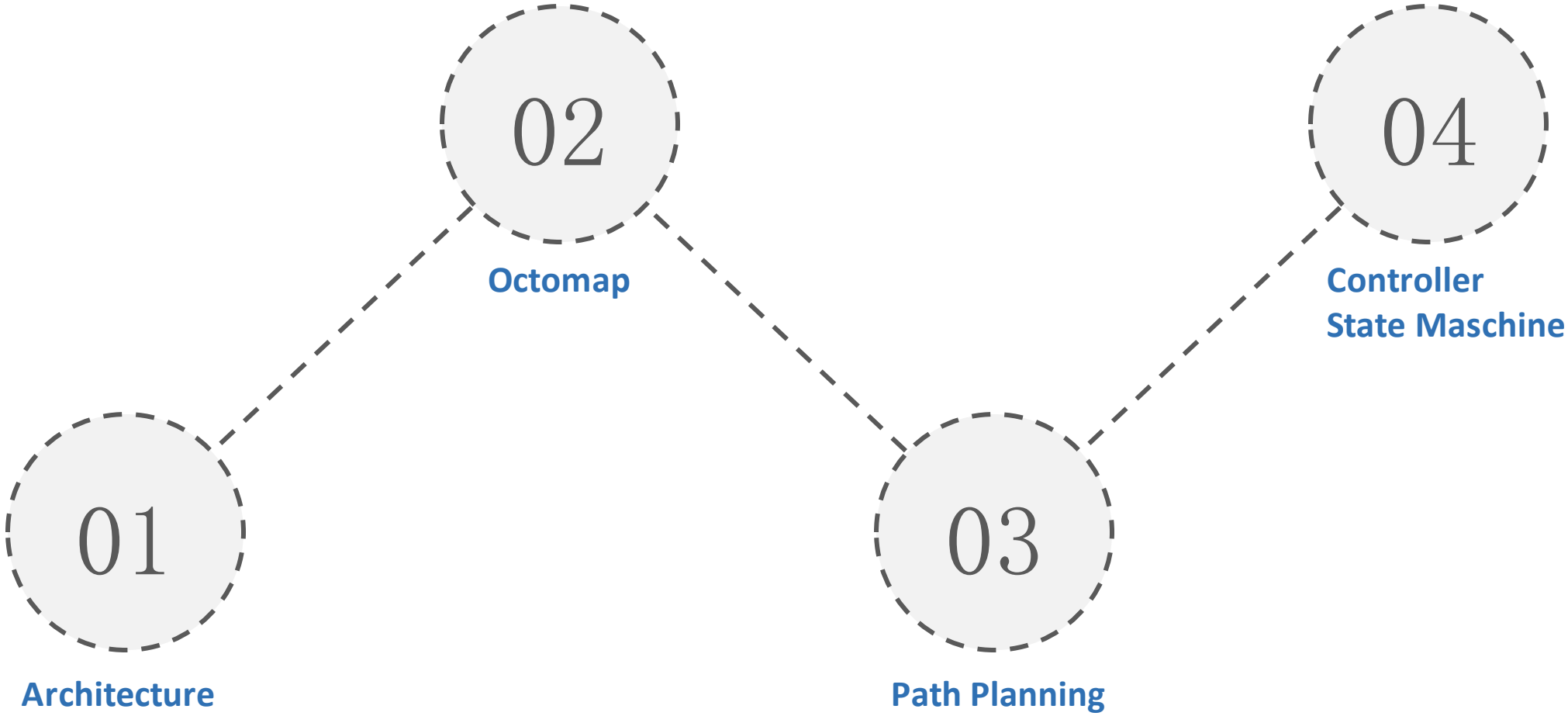
Quadruped Robot

Introduction to ROS
Endpresentation – Group 26

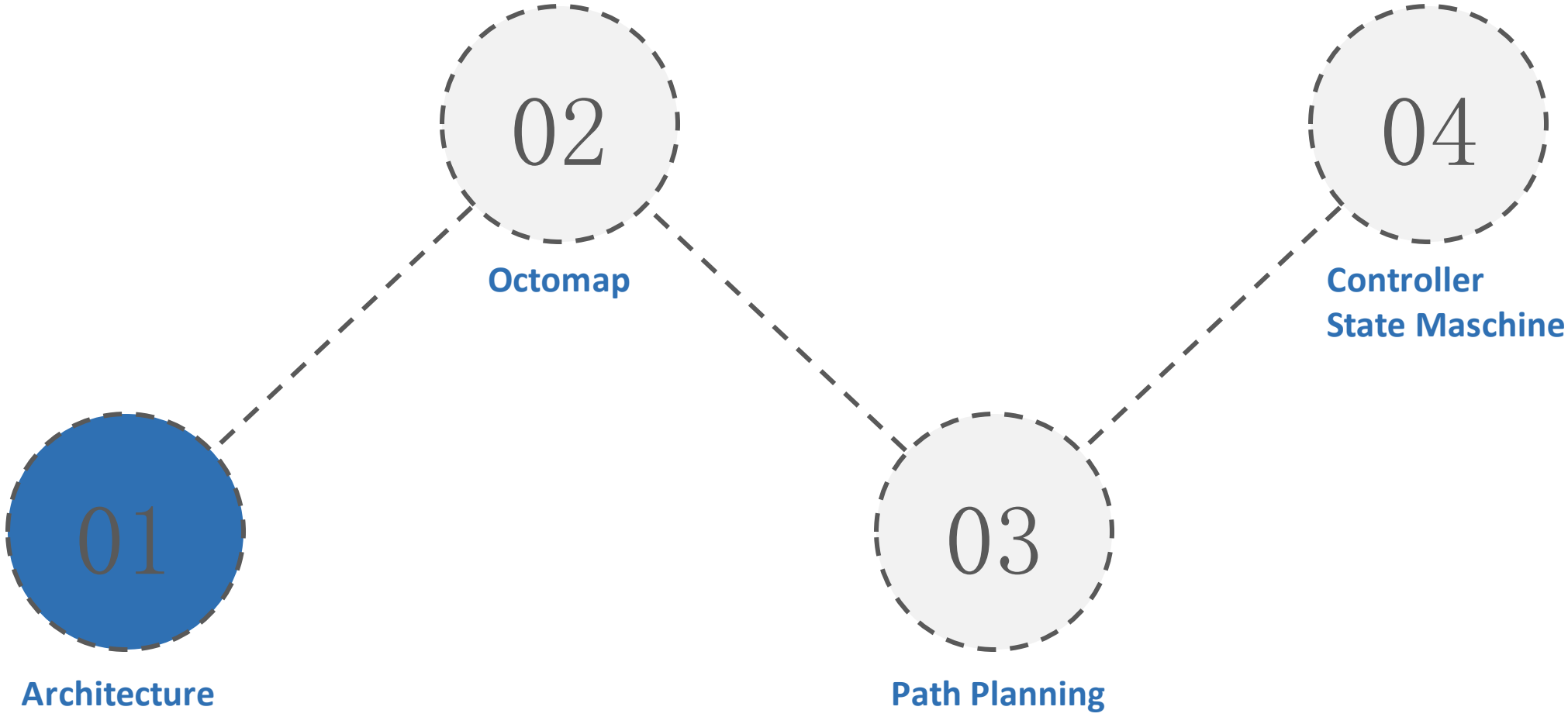
Professor: Dr. Markus Ryll

Group members: Z. Huang, C. Yao, Y. Jin, G. Lee, C. Chen

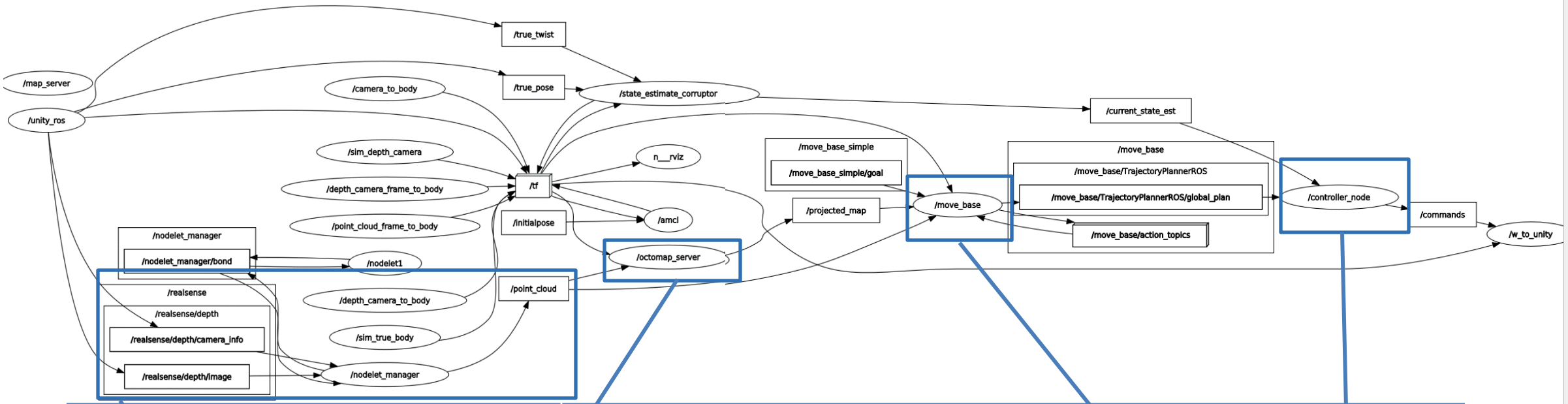
Agenda


02
Octomap
04
Controller
State Maschine
01
Architecture
03
Path Planning

Architecture

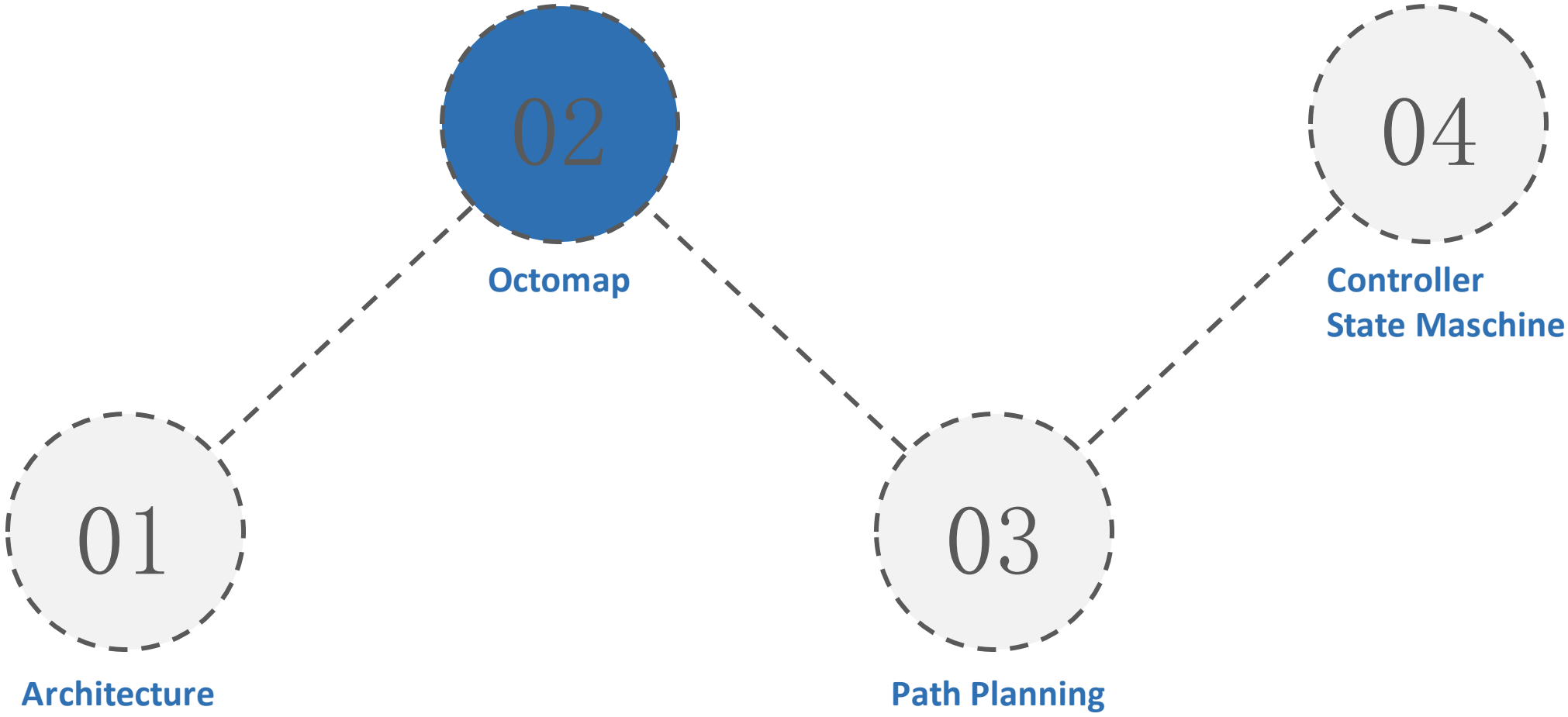


Architecture



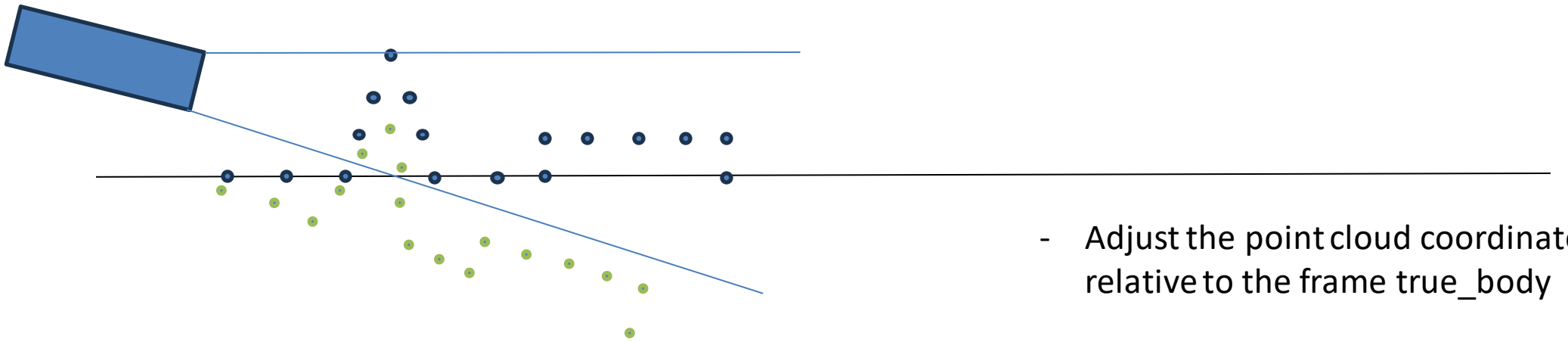
Package	Functionality
depth_image_proc	Generates point cloud
octomap	build an occupancy map based on the point cloud
move_base	nevigate the robot from its current position to the goal position
controller_pkg	control the robodog to reach the goal position by following the path plan

Octomap


02
Octomap
04
Controller
State Maschine
01
Architecture
03
Path Planning

Octomap – Point Cloud

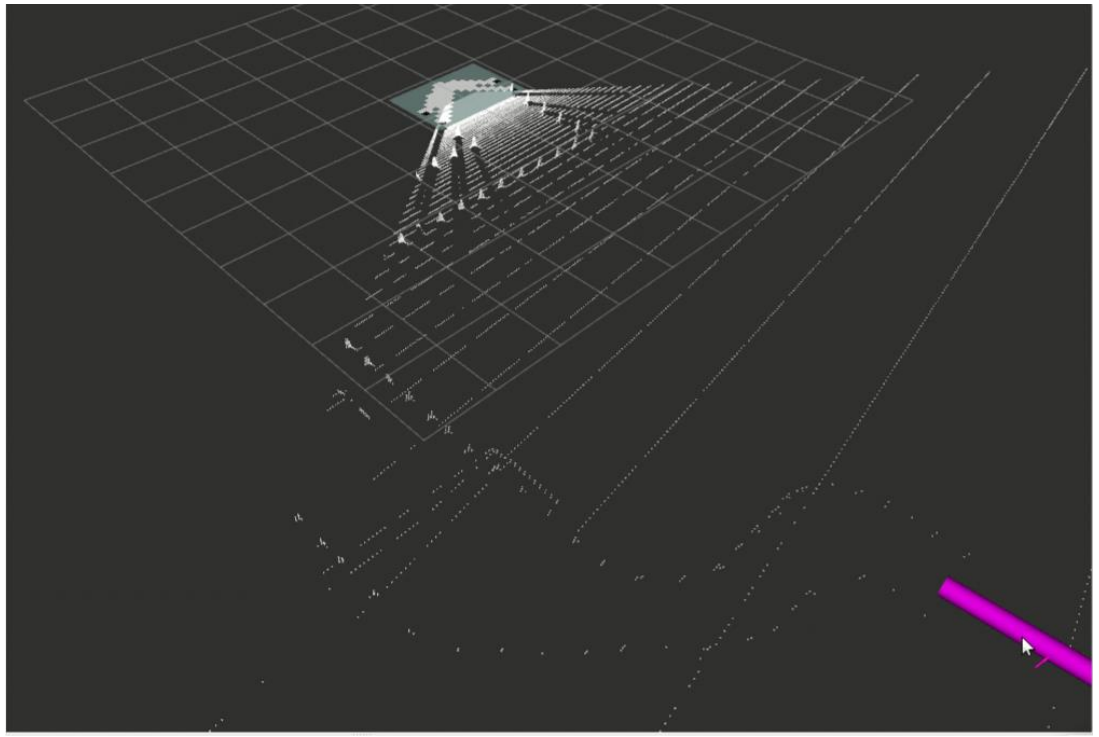
Challenge: During the movement, the camera is shaking.



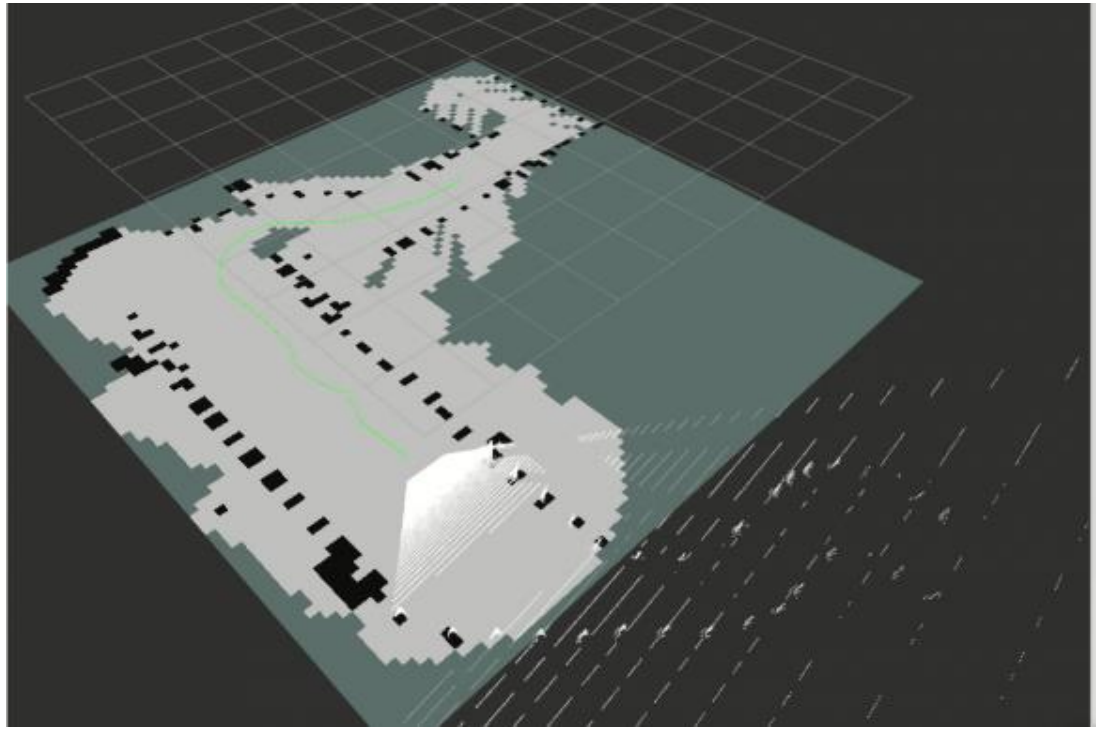
```
WARN] [1690954792.183273497]: Pointcloud in OctomapServer too small, skipping ground plane extraction
WARN] [1690954792.213091905]: Pointcloud in OctomapServer too small, skipping ground plane extraction
WARN] [1690954792.243789278]: Pointcloud in OctomapServer too small, skipping ground plane extraction
```

- Adjust the point cloud coordinates frame relative to the frame true_body
- Make ground, stairs and the ramp "sink" a little bit to filter them
- (failed) Tried to use filter_ground in pkg Octomap, but there are warnings

Octomap

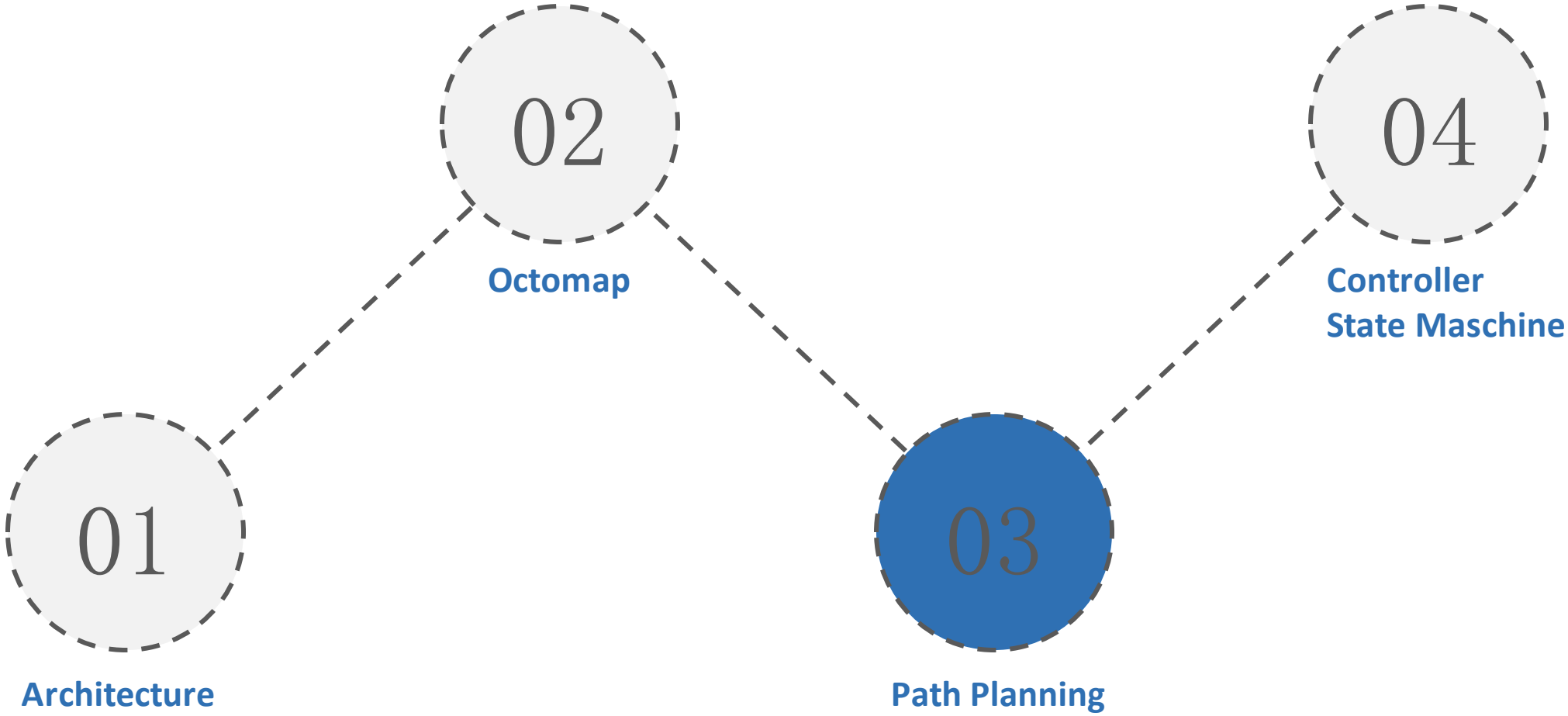


At the beginning

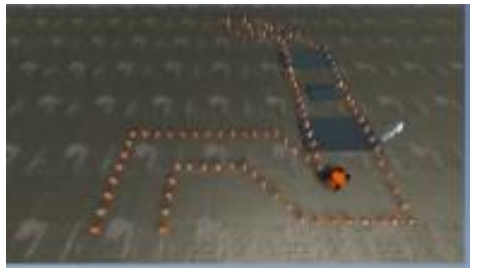


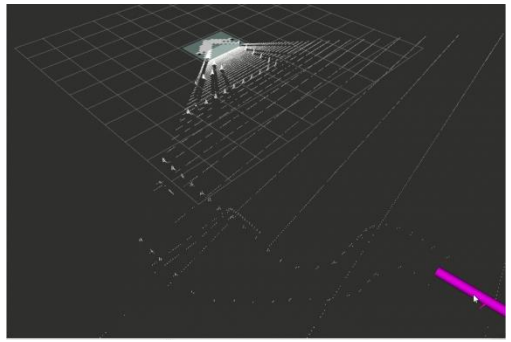
At second stair

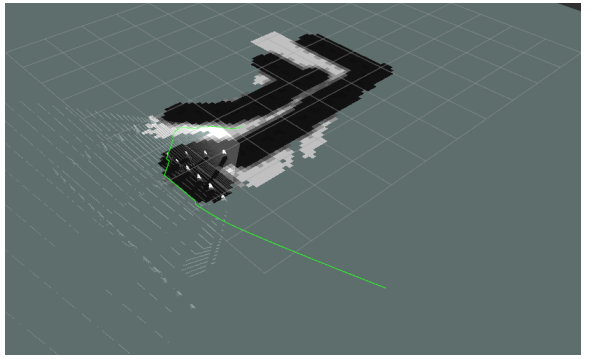
Path Planning

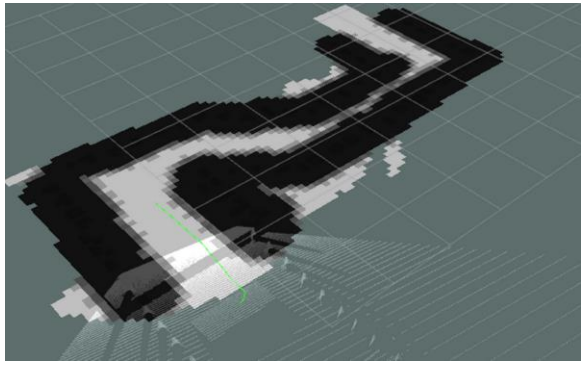


Path Planning









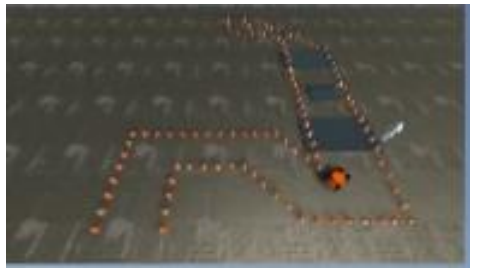
Highlight:

- Only final goal is needed to be given (One click)
- Proper robot size and inflation radius of the obstacle
- Ignore the stairs and the ramp so the robot can walk through them

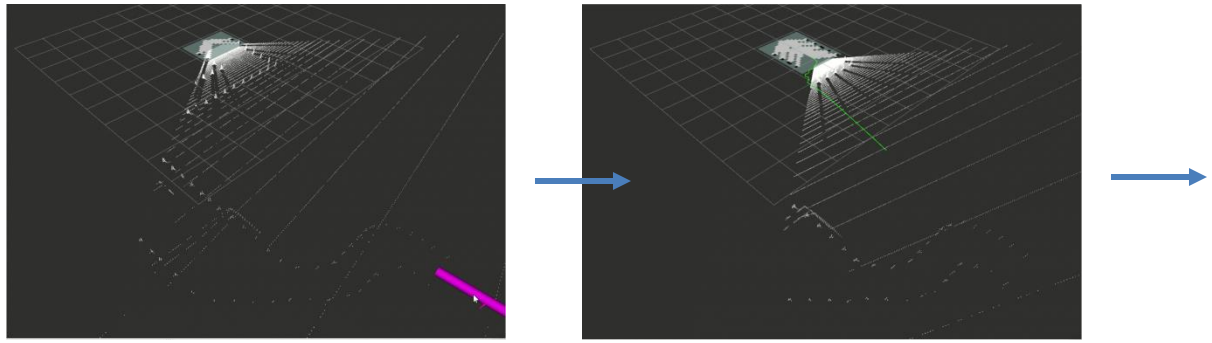
Limits:

- When go downstairs, camera towards to the ground, after walk through second stairs, path cannot be found
- (Potential) May not go through the last narrow passage (trade off with obstacle avoidance)

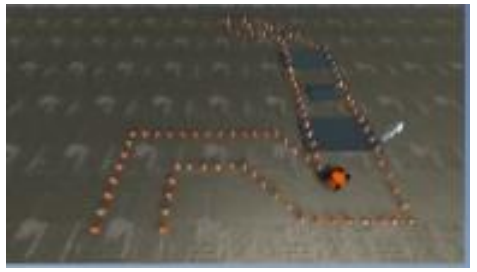
The process involves triggering the move_base node and utilizing four YAML files to activate the path planning system. These configuration options are specified to enable the robot to reach its goal.



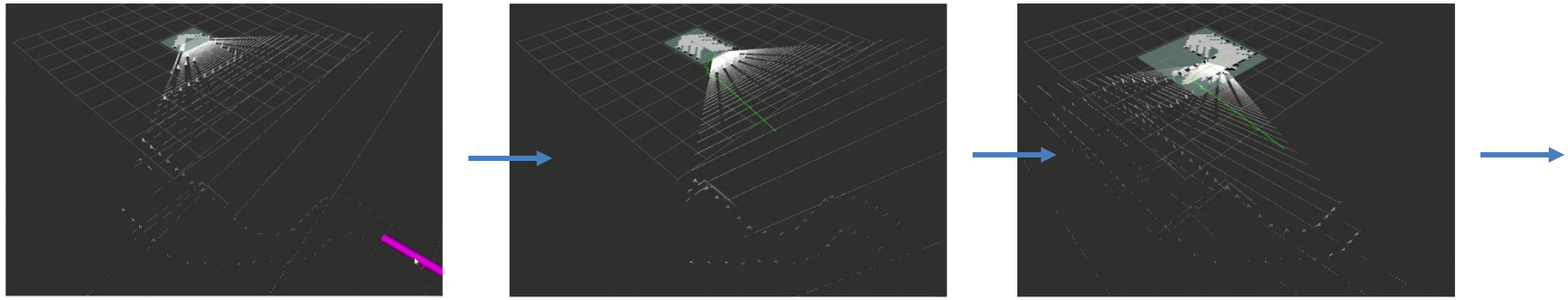
Path Planning



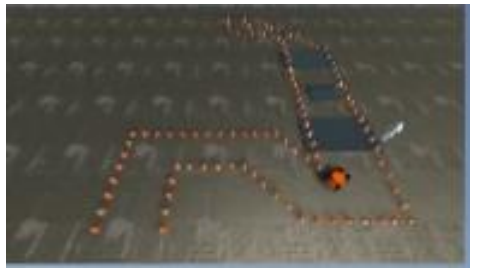
The process involves triggering the move_base node and utilizing four YAML files to activate the path planning system. These configuration options are specified to enable the robot to reach its goal.



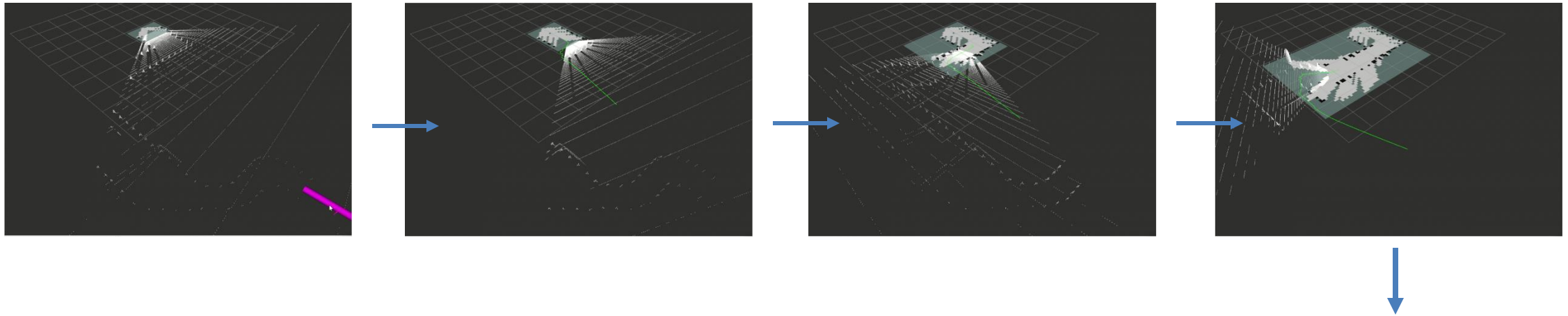
Path Planning



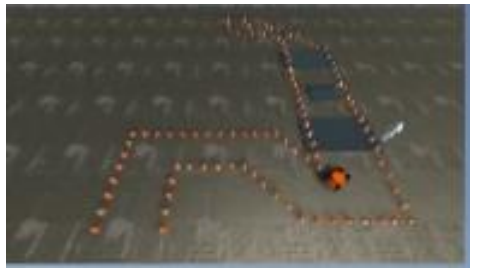
The process involves triggering the move_base node and utilizing four YAML files to activate the path planning system. These configuration options are specified to enable the robot to reach its goal.



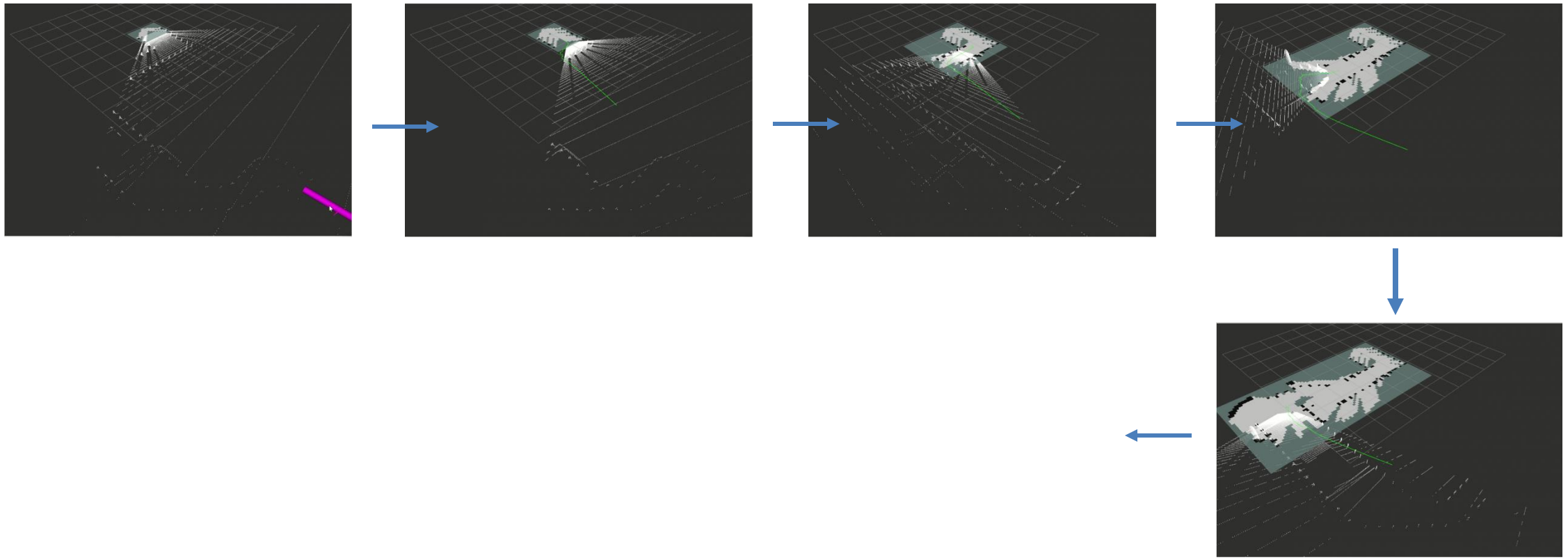
Path Planning



The process involves triggering the move_base node and utilizing four YAML files to activate the path planning system. These configuration options are specified to enable the robot to reach its goal.

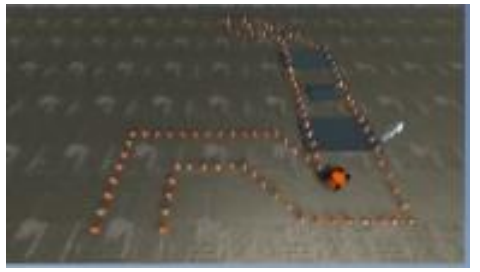


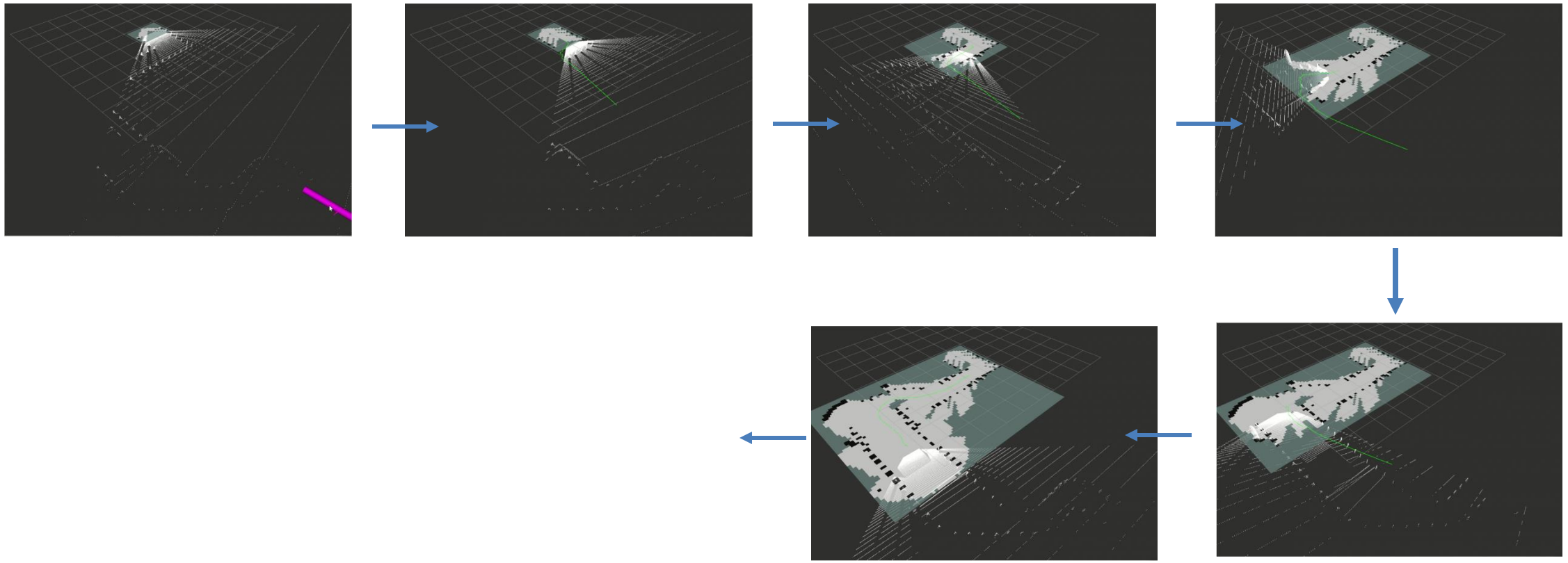
Path Planning



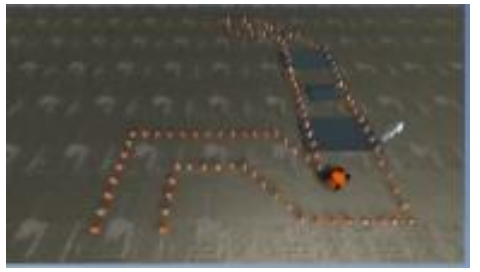
The process involves triggering the move_base node and utilizing four YAML files to activate the path planning system. These configuration options are specified to enable the robot to reach its goal.

Path Planning

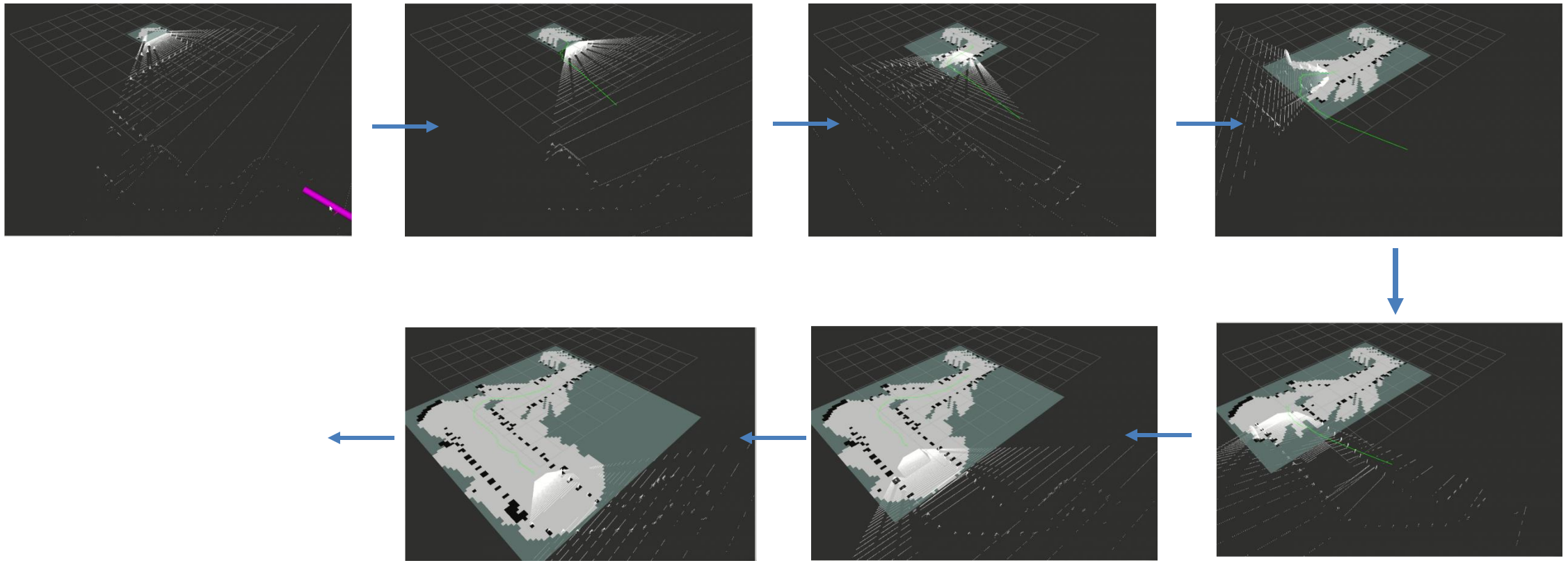




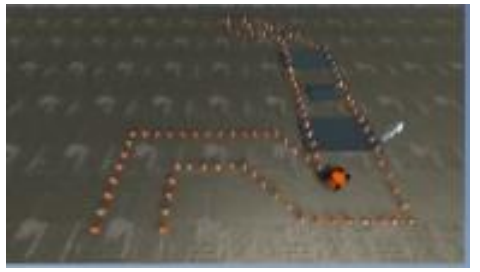
The process involves triggering the move_base node and utilizing four YAML files to activate the path planning system. These configuration options are specified to enable the robot to reach its goal.



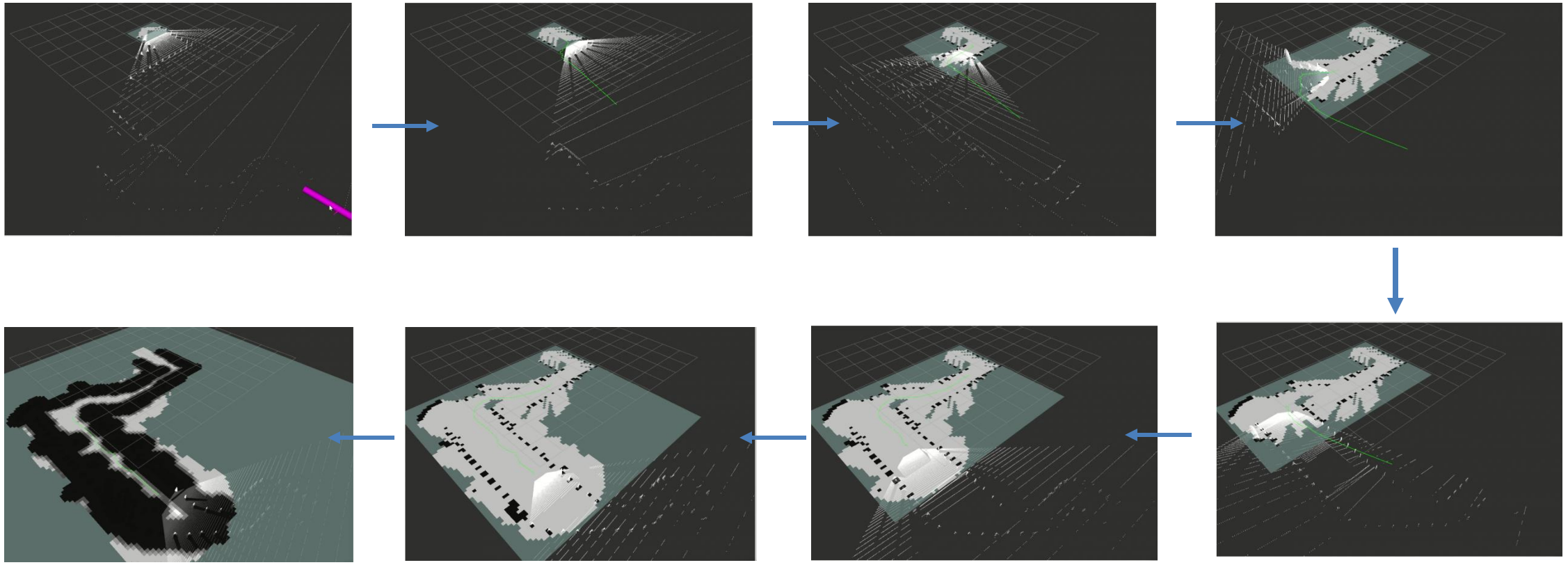
Path Planning



The process involves triggering the move_base node and utilizing four YAML files to activate the path planning system. These configuration options are specified to enable the robot to reach its goal.

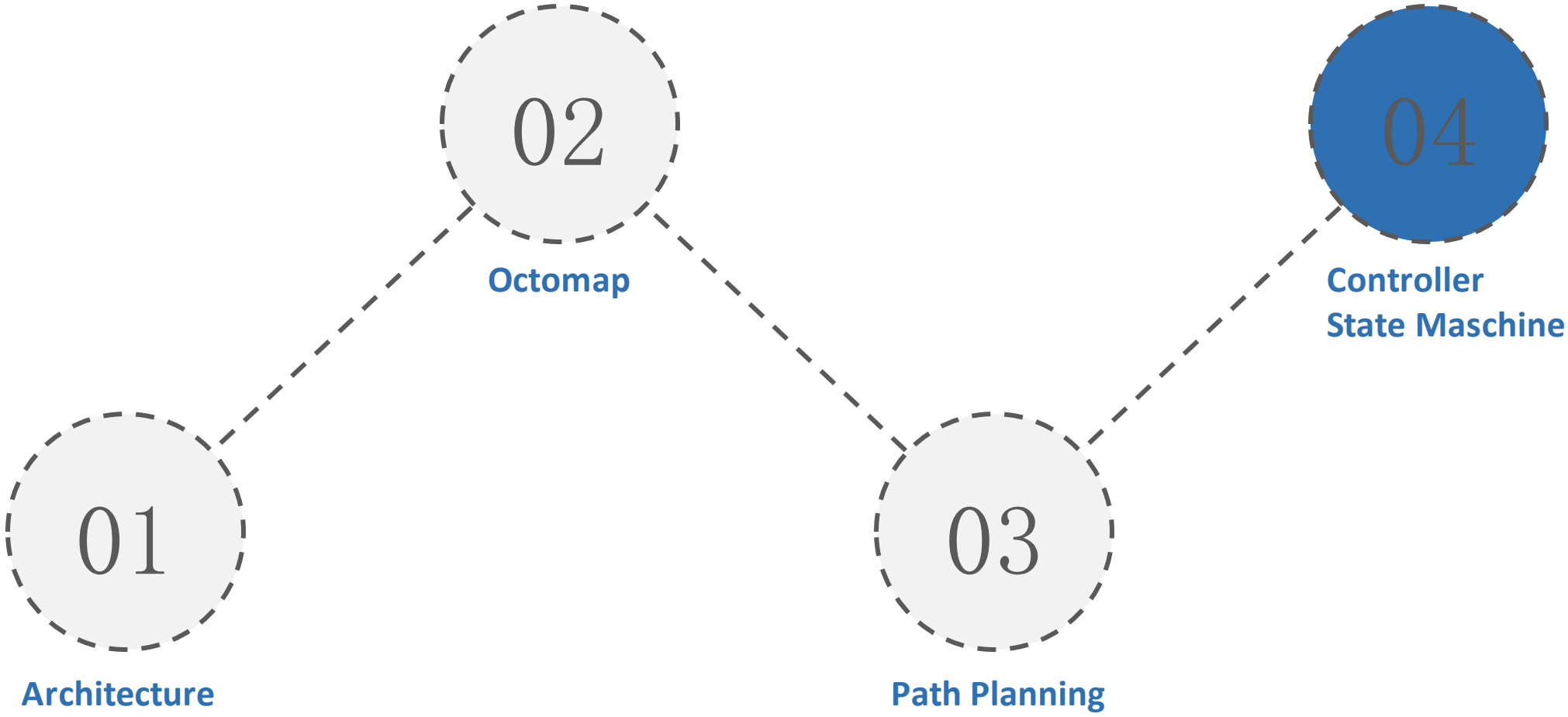


Path Planning

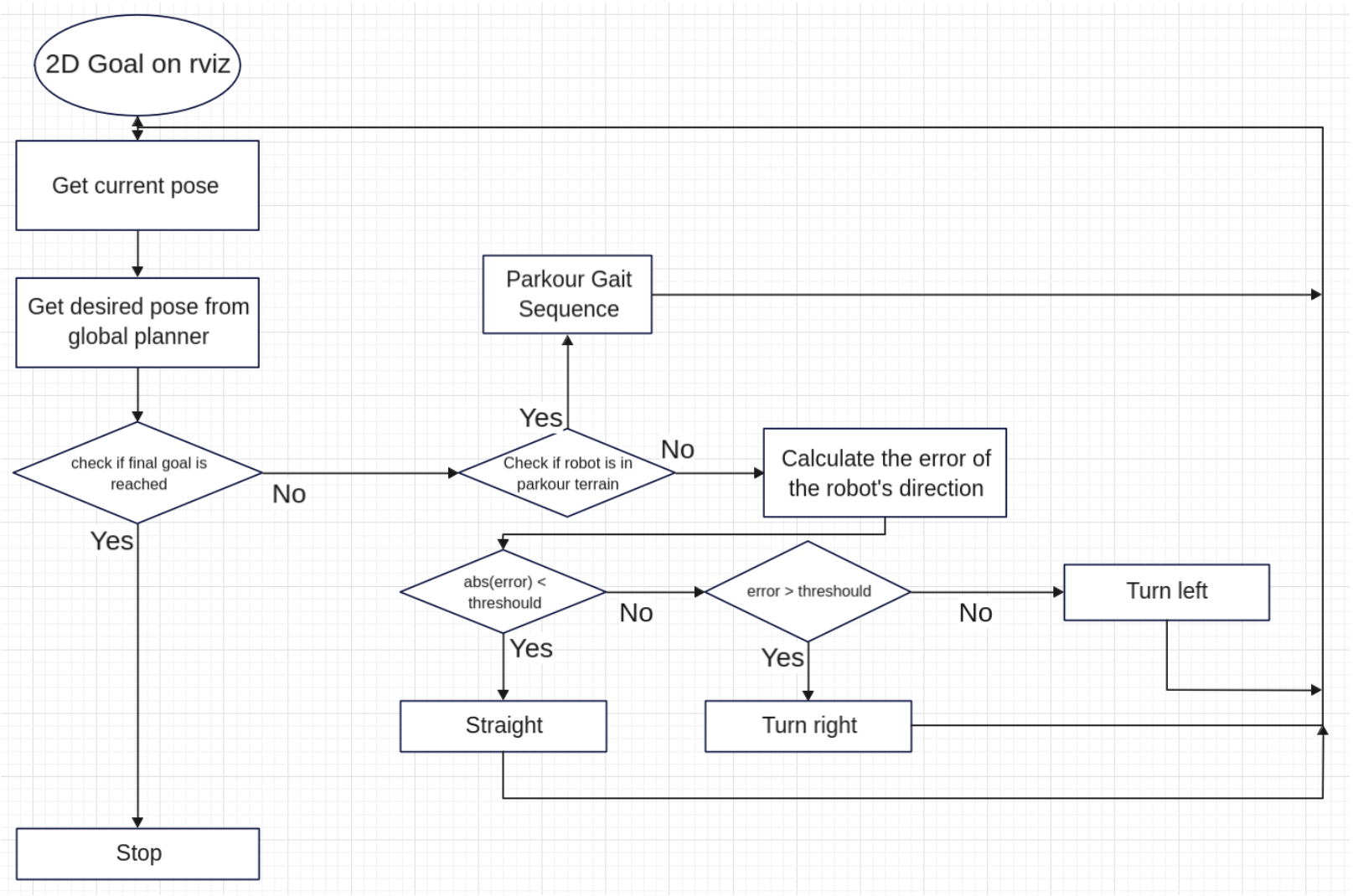


The process involves triggering the move_base node and utilizing four YAML files to activate the path planning system. These configuration options are specified to enable the robot to reach its goal.

Controller & State Machine



Controller


2D Goal on rviz
Get current pose
Get desired pose from global planner
check if final goal is reached
No
Yes
Check if robot is in parkour terrain
Yes
No
Parkour Gait Sequence
Calculate the error of the robot's direction
abs(error) < threshould
No
Yes
error > threshould
No
Yes
Turn left
Straight
Turn right
Stop

State Machine

	0-90	90-180	180-270	270-360
Front Left				
Front Right				
Back Right				
Back Left				

Sequence diagram of trot gait (Straight)

Stance Phase

Swing Phase

```
msg.angular_velocities[0] = 0;
msg.angular_velocities[1] = 90;
msg.angular_velocities[2] = 0;
msg.angular_velocities[3] = 0;
msg.angular_velocities[4] = 8;
```

State Machine

	0-45	45-90	90-135	135-180
Front Left				
Front Right				
Back Right				
Back Left				

Sequence diagram of turn right/left

Stance Phase

Swing Phase

```
msg.angular_velocities[0] = 0;
msg.angular_velocities[1] = 45;
msg.angular_velocities[2] = 0;
msg.angular_velocities[3] = 0;
msg.angular_velocities[4] = 8;
```

State Machine

	0-90	90-180	180-270	270-360
Front Left	Low Amp	Low Amp	Low Amp	Low Amp
Front Right	Low Amp	Low Amp	Low Amp	Low Amp
Back Left	High Amp	High Amp	High Amp	High Amp
Back Right	High Amp	High Amp	High Amp	High Amp

Sequence diagram of climb

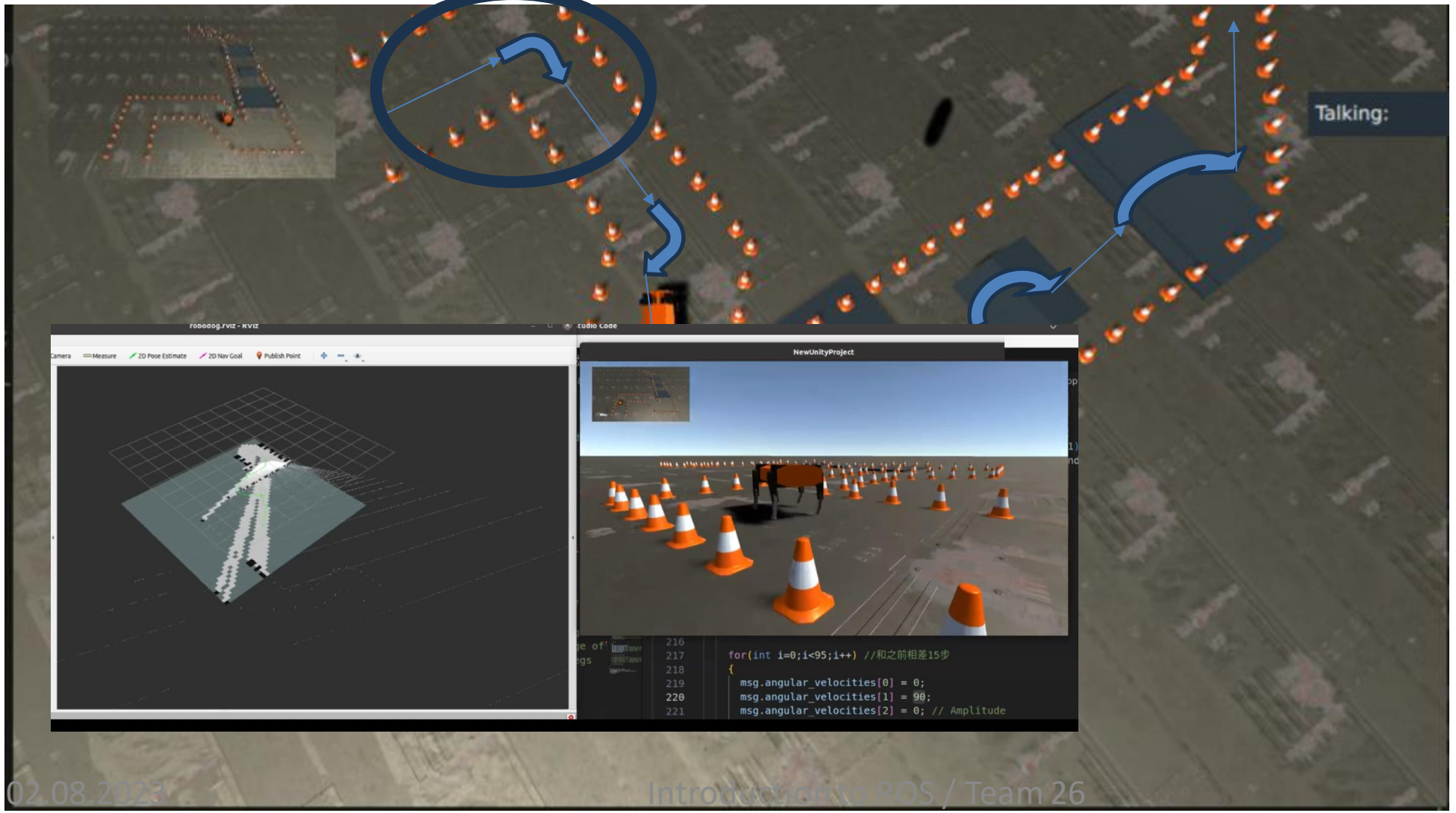
```
msg.angular_velocities[0] = 0;
msg.angular_velocities[1] = 0;
msg.angular_velocities[2] = 3; /
msg.angular_velocities[3] = 25;
msg.angular_velocities[4] = 12;
```

Stance Phase

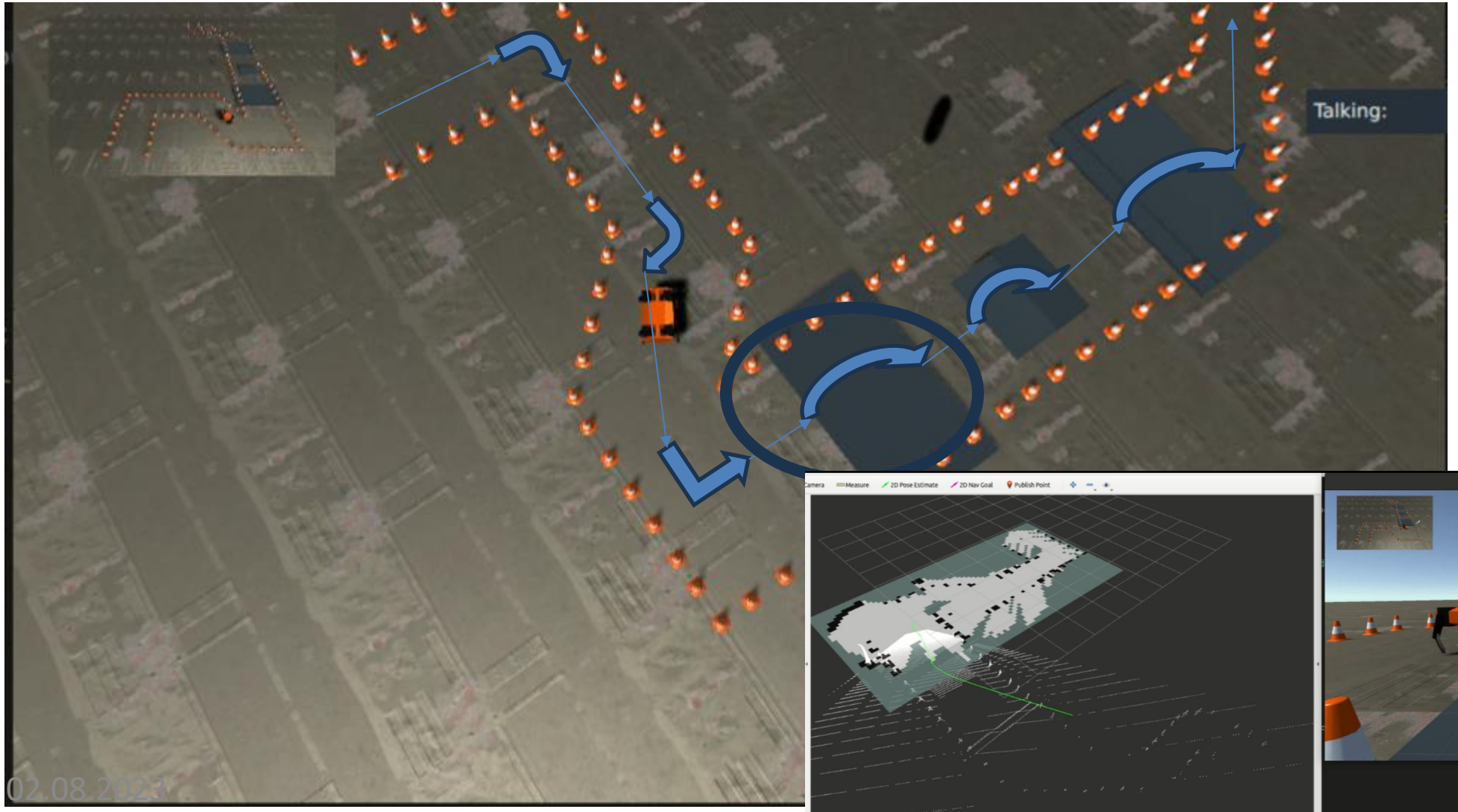
Swing Phase

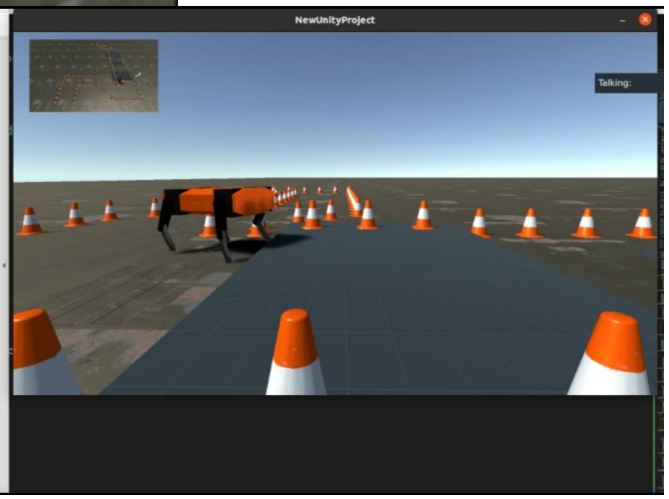
Controller

Controller & State Machine

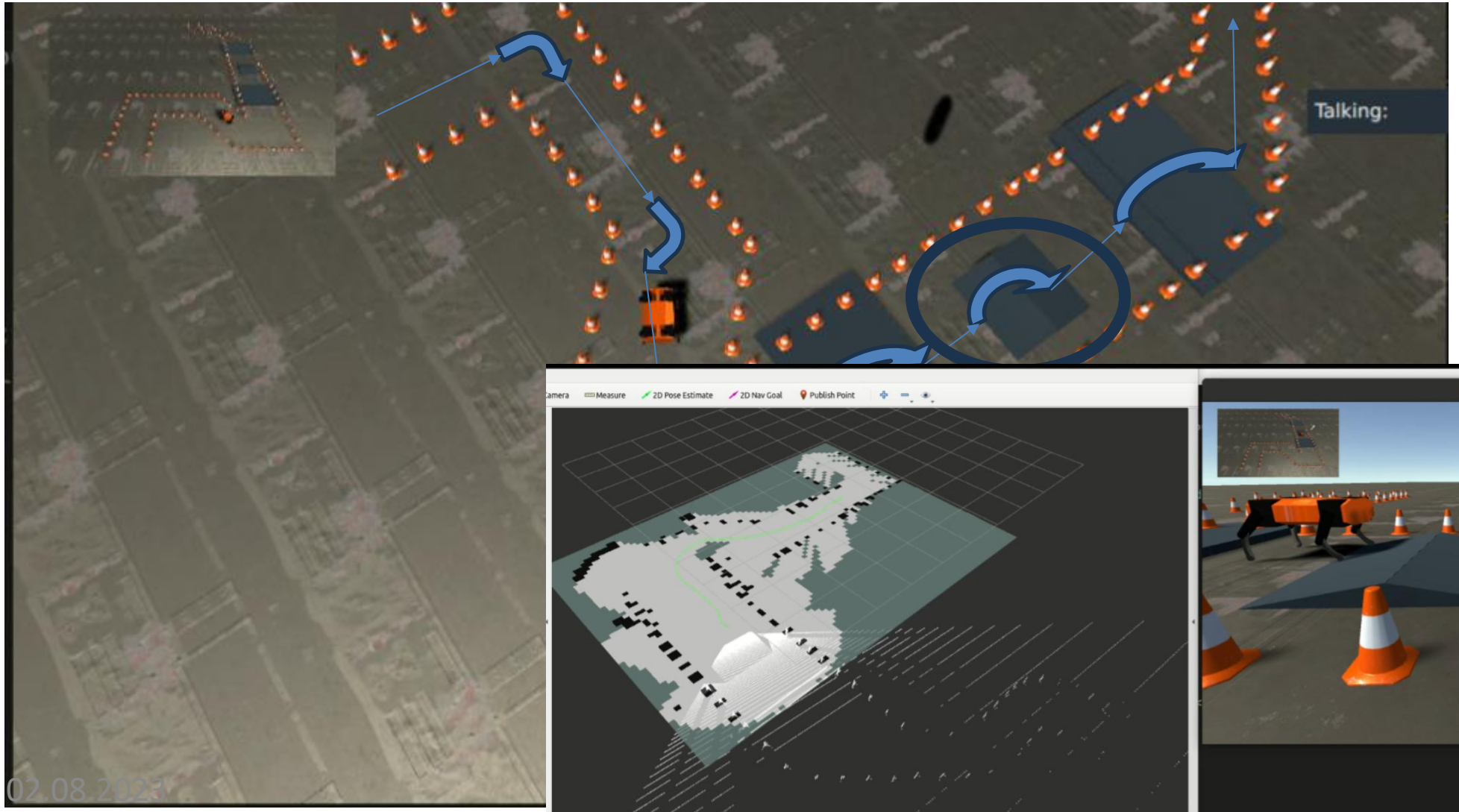


Controller & State Machine

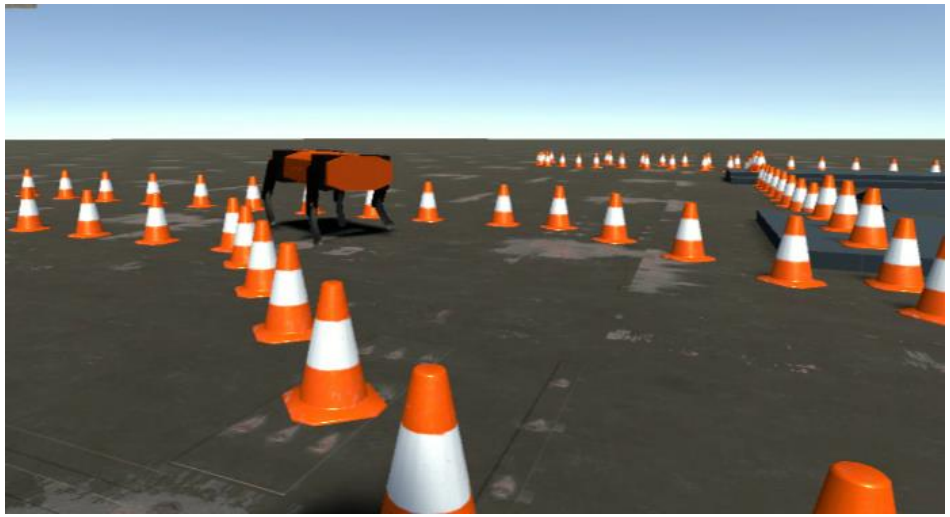




Controller & State Machine



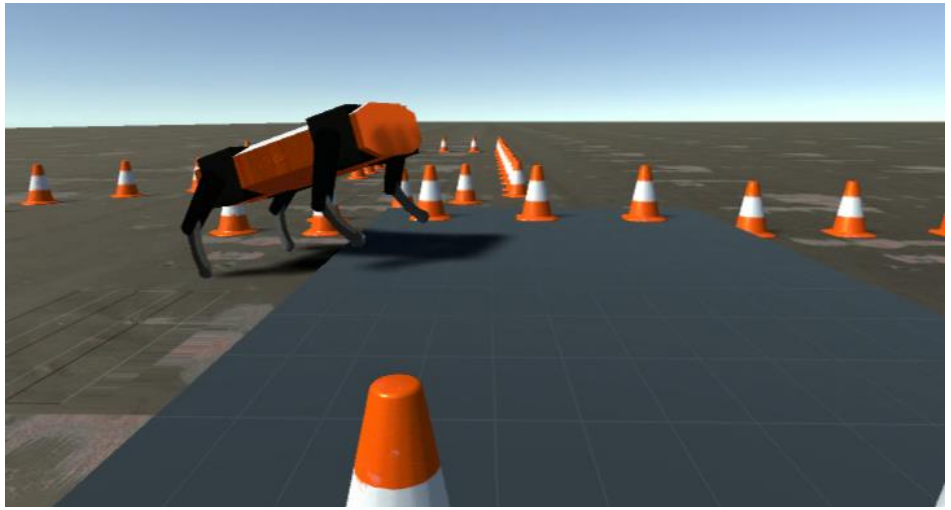
Controller & State Machine



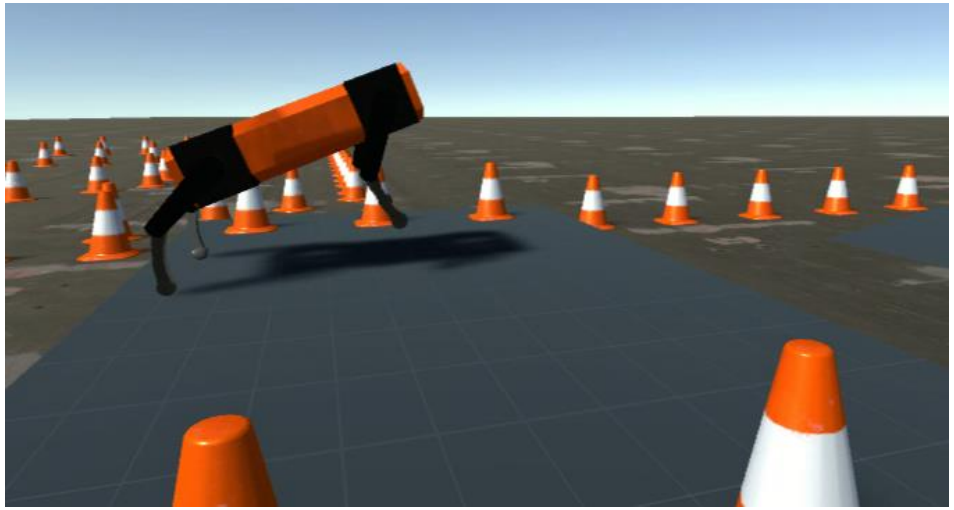
straight	
angular velocities[0]	0
angular velocities[1]	90
angular velocities[2]	0
angular velocities[3]	0
angular velocities[4]	8

Turn left or right	
angular velocities[0]	0
angular velocities[1]	-45 or 45
angular velocities[2]	0
angular velocities[3]	0
angular velocities[4]	8

Overview of Project and Results - Controller, state machine

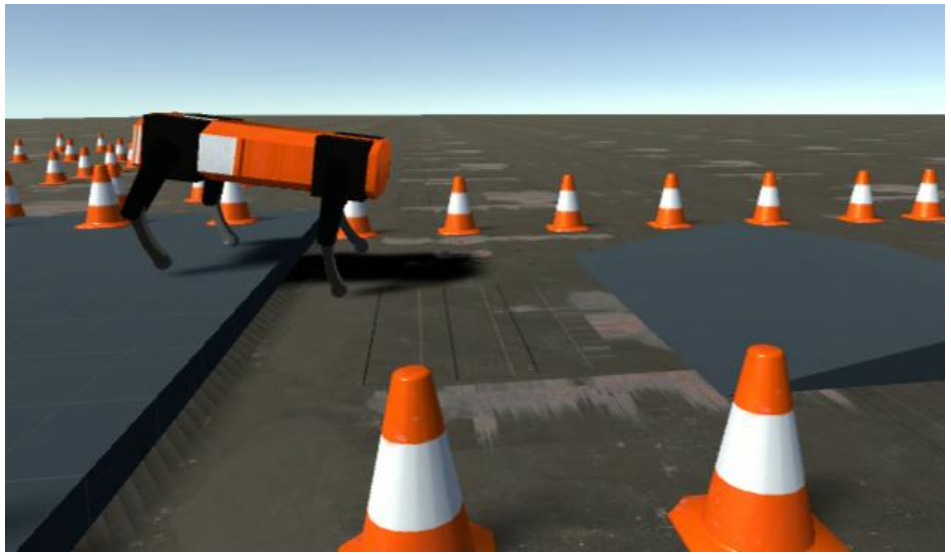


jump up to one-step staircase(the front legs)	
angular velocities[0]	0
angular velocities[1]	0
angular velocities[2]	3
angular velocities[3]	25
angular velocities[4]	12

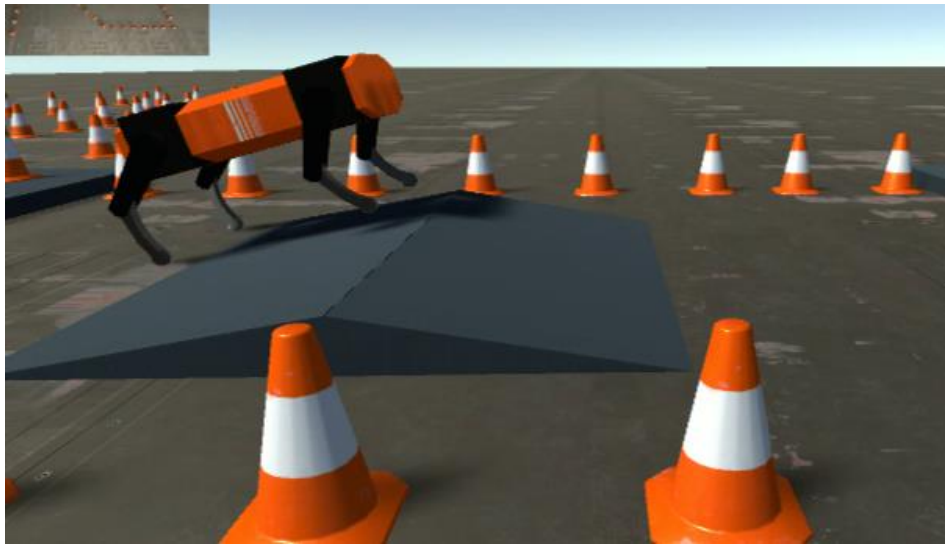


jump up to **one-step staircase (the back legs)**	
angular velocities[0]	0
angular velocities[1]	0
angular velocities[2]	3
angular velocities[3]	38
angular velocities[4]	10

Controller & State Machine

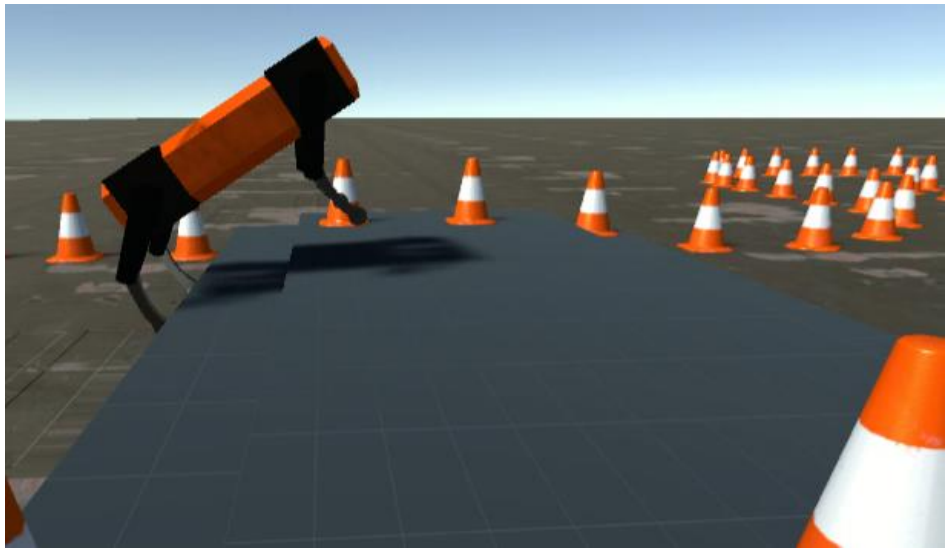


Descending stairs	
angular velocities[0]	0
angular velocities[1]	90
angular velocities[2]	0
angular velocities[3]	4
angular velocities[4]	8

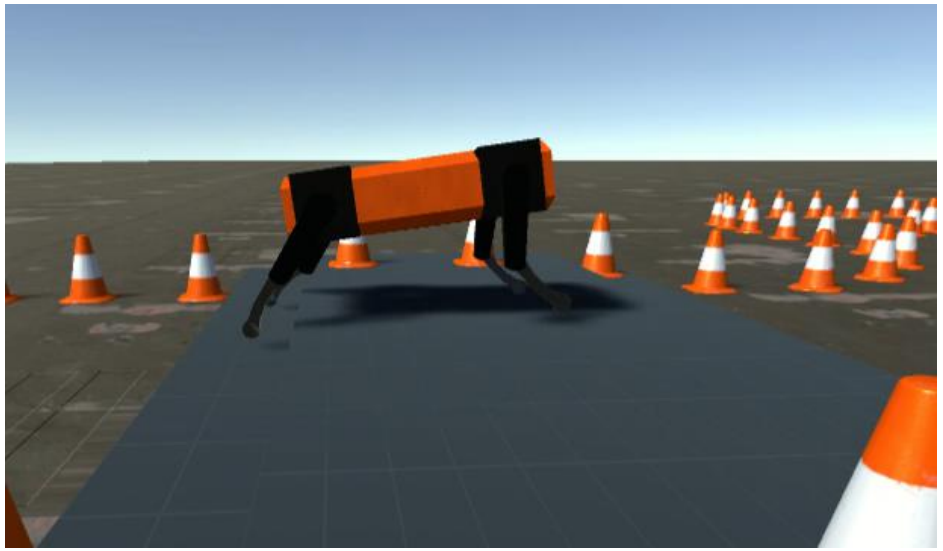


Ascending a slope	
angular velocities[0]	0
angular velocities[1]	0
angular velocities[2]	3
angular velocities[3]	25
angular velocities[4]	10

Controller & State Machine

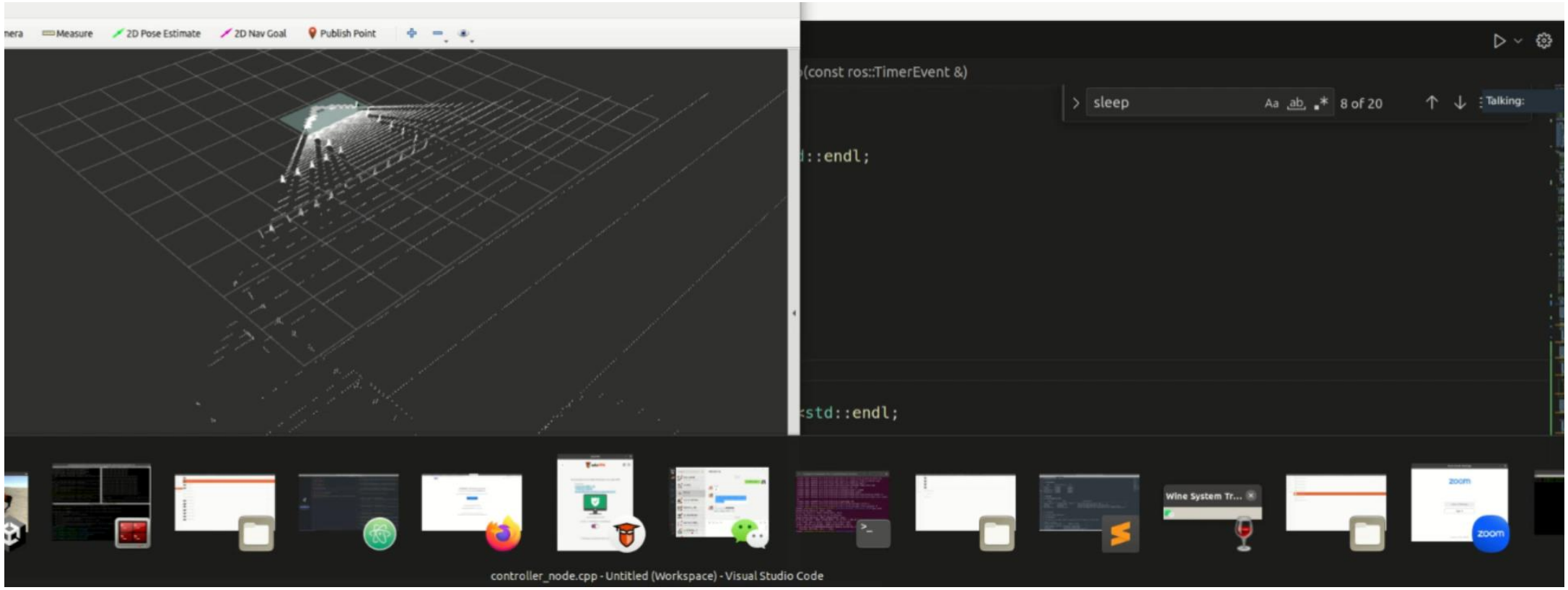


jump up to two-step staircase(the front legs)	
angular velocities[0]	0
angular velocities[1]	0
angular velocities[2]	3
angular velocities[3]	25
angular velocities[4]	10



jump up to two-step staircase (the back legs)	
angular velocities[0]	10
angular velocities[1]	10
angular velocities[2]	-4
angular velocities[3]	40
angular velocities[4]	10

Live Demo


Measure
2D Pose Estimate
2D Nav Goal
Publish Point
(const ros::TimerEvent &)
sleep
8 of 20
Talking:
::endl;
std::endl;
Wine System Tr...
zoom
controller_node.cpp - Untitled (Workspace) - Visual Studio Code

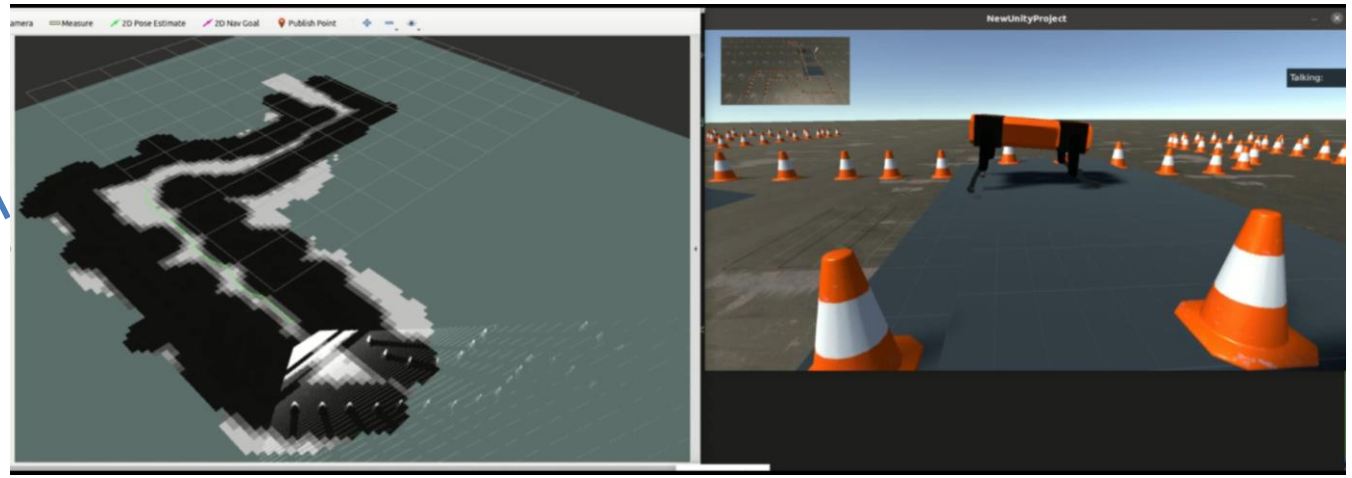
Limitations and issues

1. Finding an optimal value for “inflation_radius” is challenging. **In narrow passages**, the robot's large size and a large inflation_radius cause it to consider the passage impassable.

2. When descending stairs, the camera angle facing the ground is too large, causing the ground to be perceived as an obstacle. As a result, after descending the stairs, the robot cannot find a suitable path to reach the goal.

3. When using DWA as the local planner, it cannot find a path to reach the 2D goal. It is unclear whether the problem lies with the DWA planner itself or if there are errors in “base_local_planner.yaml” parameters.

4. Finding a perfect solution for the gait during ascending and descending stairs is indeed challenging. This is because the height and slope of the stairs can lead to gait instability or difficulties in balancing for the robot.



Feedback about the project work

- Good practical understanding of ROS.
- Effective teaching approach with theory followed by projects.
- Challenges to address for a more robust and reliable robot performance.
- Continuous testing, iteration, and parameter fine-tuning are crucial for optimal results.
- Problems lead to knowledge

TUM School of Engineering and Design
Technical University of Munich

THANKS