Introduction to ROS SS22 Final Project

Autonomous Quadruped

Group 14 Awesome Dog

16.08.2022

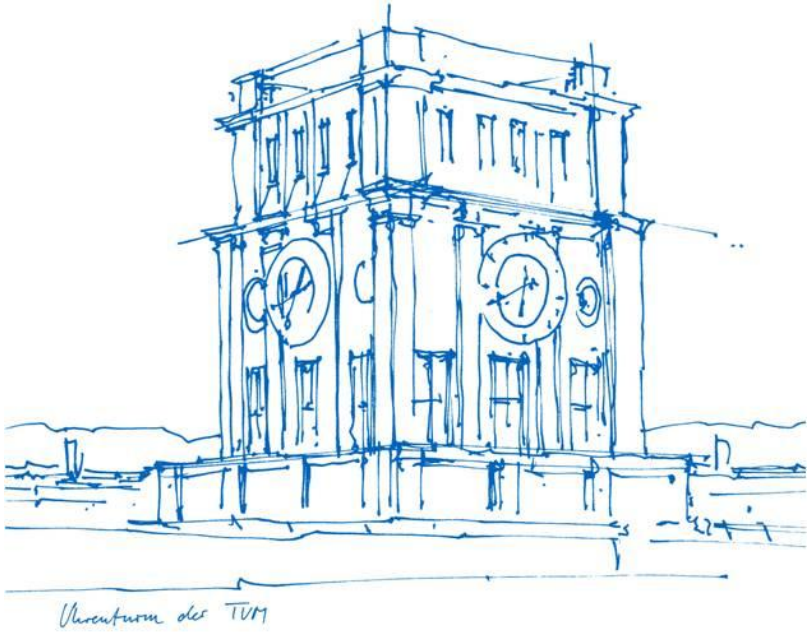
Dian Yuan

Yinglei Song

Zhelin Yang

Hang Li

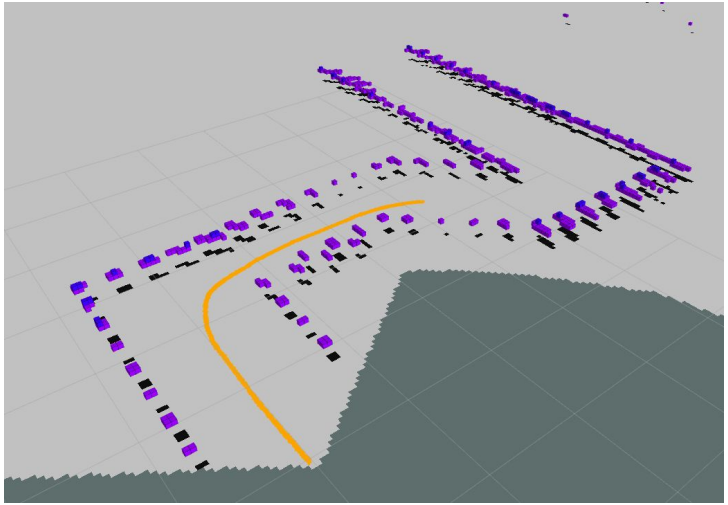
Zhaoqi Zhou



Perception & Mapping

Used external packages: depth_image_proc, octomap_server.

Transform depth image to point cloud.
Build Octomap based on point cloud.
Publish 2D projected map for further usage.

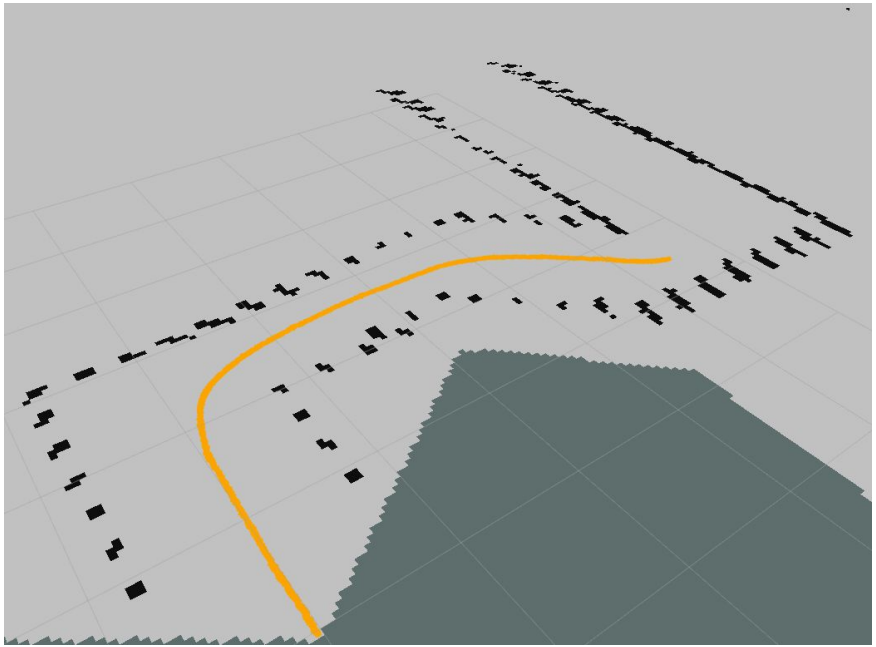


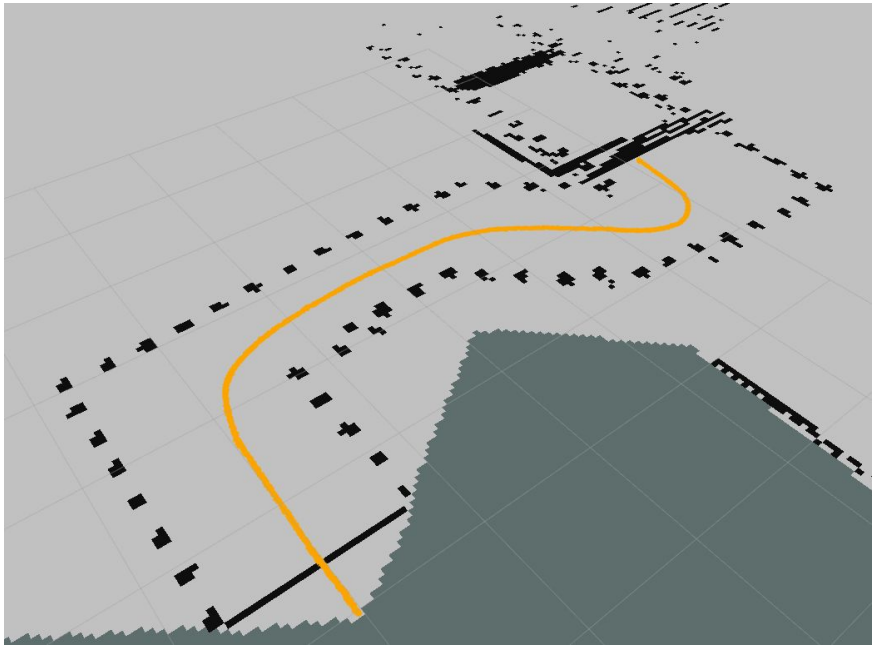
Two maps:

Projected map without step and slope: z coordinate between 0.15m and 5.0m → Path Planning

Projected map with step and slope: only filter out ground surface. → Step & Slope Detection

Map Presentation





Step & Slope Detection

Get waypoints from ***nav_msgs/path***.

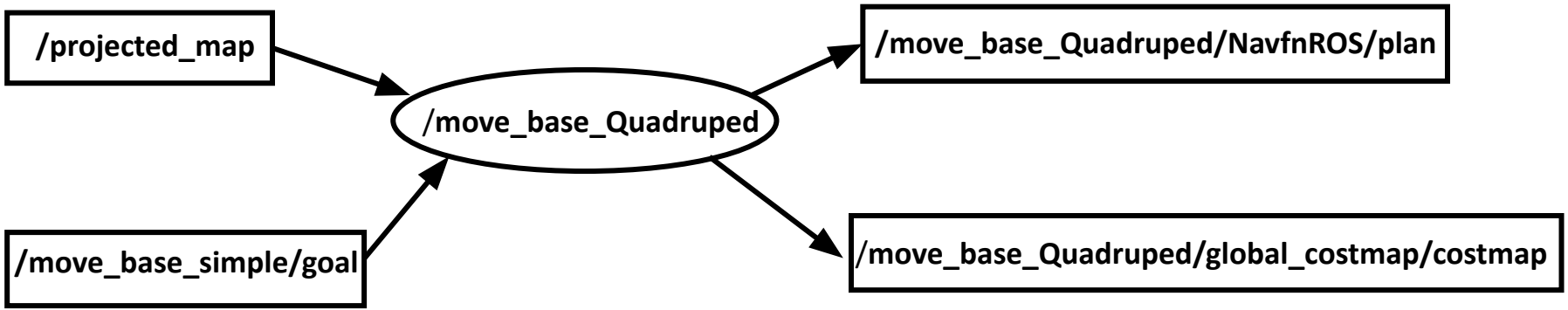
Get corresponding occupancy values from ***nav_msgs/Occupancygrid*** → Array Transformation

Three different values:
100: occupied
0: free
-1: unknown

If more than 5 of following 10 waypoints are occupied → There is a step or slope.

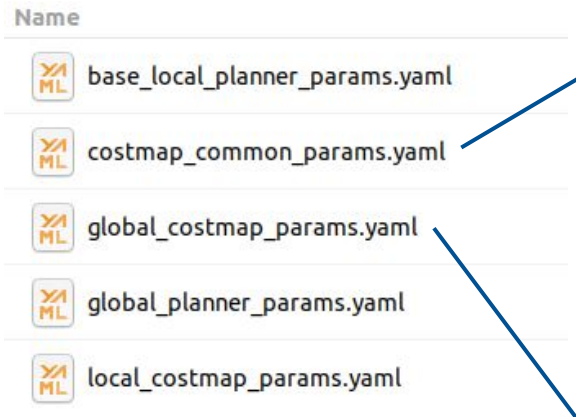
Path Planning

Move_Base Package



Path Planning

Configuration files




```yaml
footprint: [[-0.2, -0.2], [-0.2, 0.1], [0.2, 0.1], [0.2, -0.2]]

obstacle_range: 5.0
raytrace_range: 5.0

inflation_radius: 0.352
cost_scaling_factor: 0.0

map_type: costmap

observation_sources: point_cloud_sensor

point_cloud_sensor: {sensor_frame: RoboDog/base_0/Quad_Intro2ROS/Sensors/DepthCamera

static_layer:
  map_topic: /projected_map
  track_unknown_space: true
```

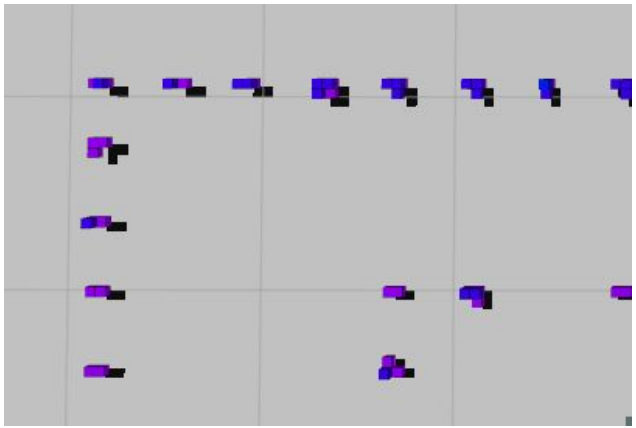
```yaml
global_costmap:
  global_frame: world
  robot_base_frame: true_body

  update_frequency: 100.0
  publish_frequency: 50.0
  transform_tolerance: 0.5

  always_send_full_costmap: true
  static_map: true
  origin_x: 0.0
  origin_y: 0.0
```

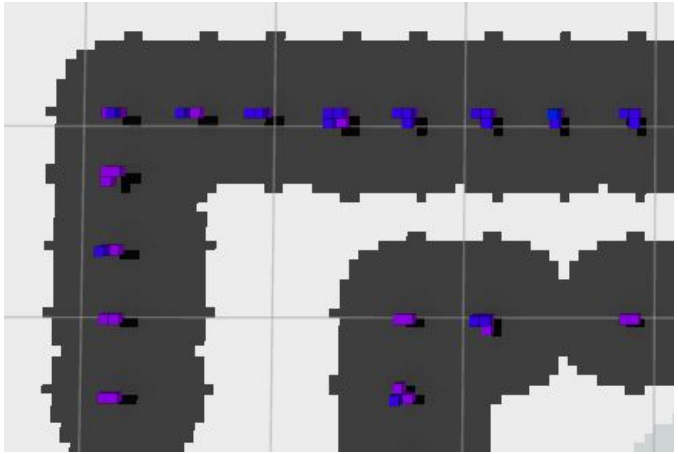
Path Planning

octomap



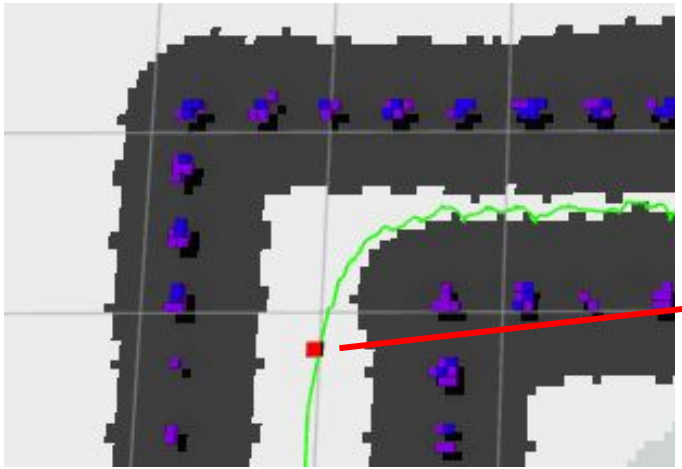
Move_base

costmap



Path Planning

path(green line)

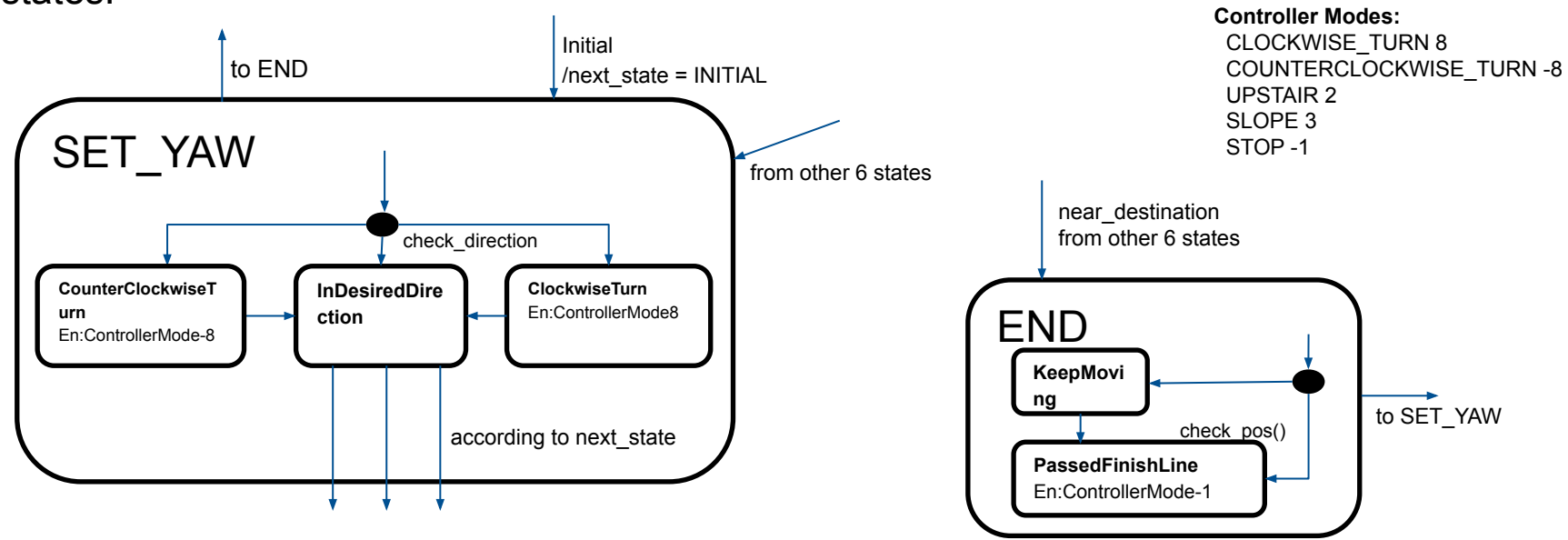


Data of a point

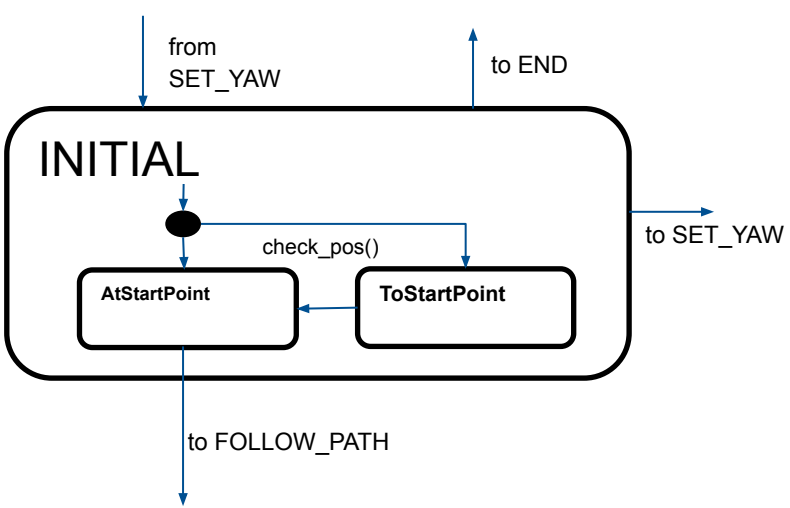
```
-
  header:
    seq: 0
    stamp:
      secs: 1660479426
      nsecs: 236030248
    frame_id: "world"
  pose:
    position:
      x: 5.500000266730785
      y: 6.000000128895044
      z: 0.0
    orientation:
      x: 0.0
      y: 0.0
      z: 0.0
      w: 1.0
```

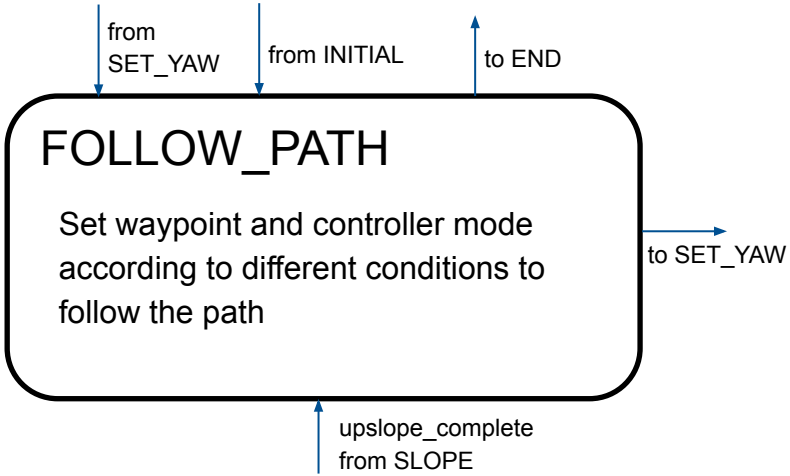
State Machine

The states experienced by the robot are described by a hierarchical state machine with 7 main states:

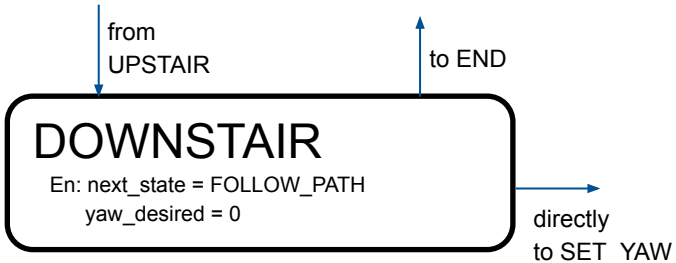


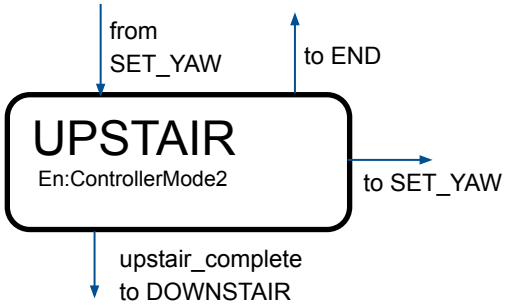
State Machine

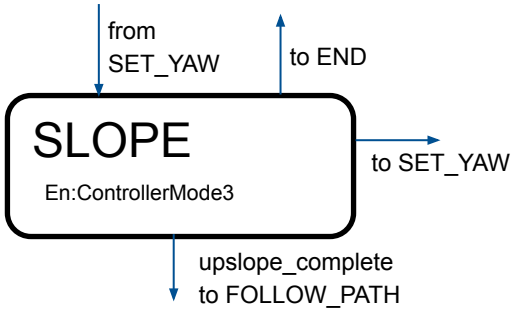




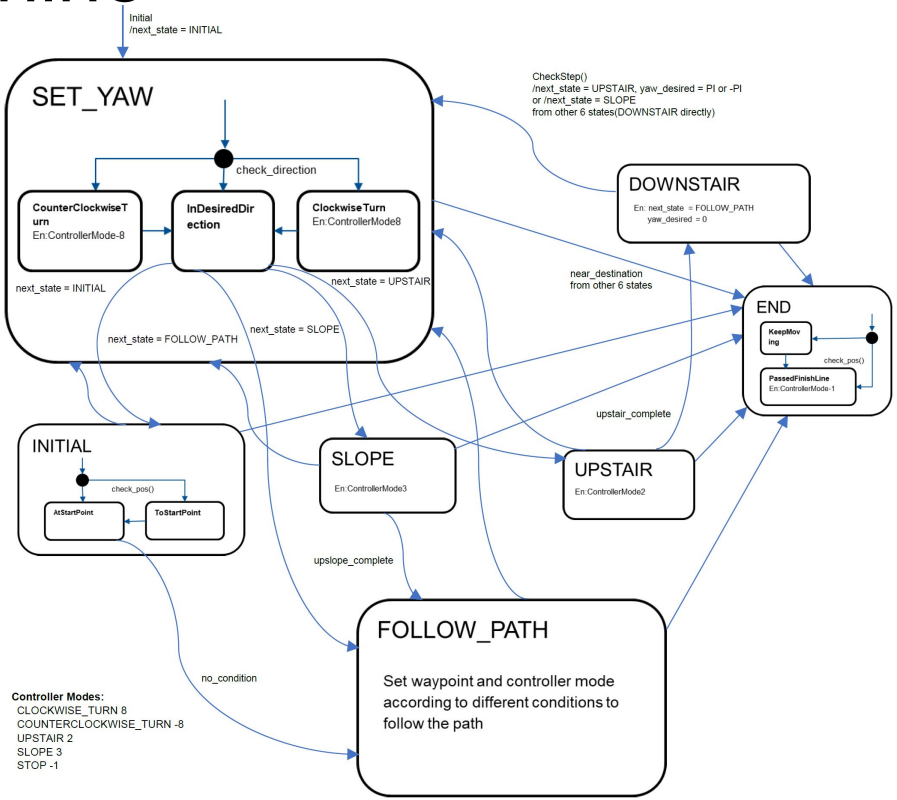
State Machine







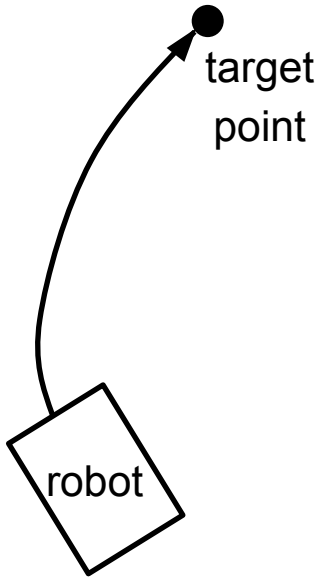
State Machine



Controller

Function of the controller

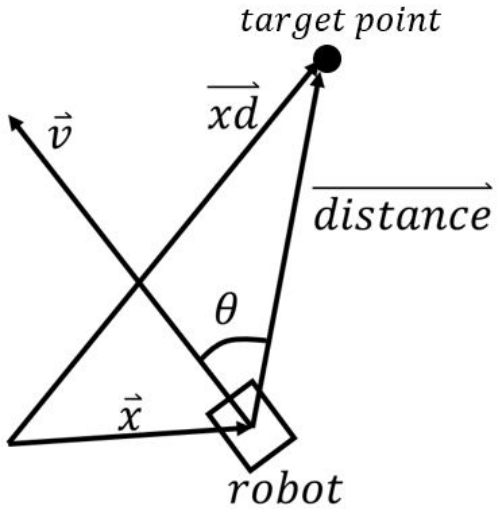
Given a target point, the controller adjusts the forward orientation of the robot towards the point and drive the robot to reach the point



Controller

Variables

x: current position (subscribed)
xd: target position (subscribed)
v: current speed (subscribed)
distance: vector pointing from the robot to the target position (xd - x)
θ: angle between v and distance



Controller

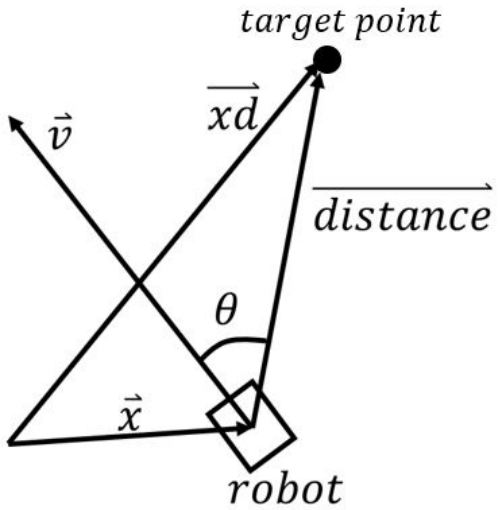
Size of the speed

$|speed| \propto$ |distance|

$0.2 < |speed| < 1$

If |distance|<threshold

|speed|=0



Controller

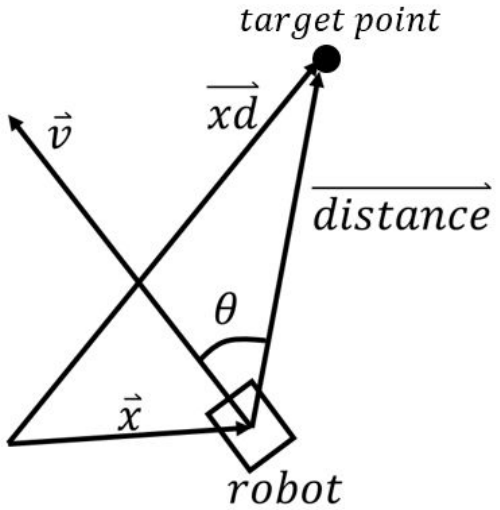
Orientation & Rotational direction

Orientation: the same direction as v

Rotational direction:

$$\vec{v} \times \overrightarrow{distance}$$

if ≤ 0, turn right
else turn left



Controller

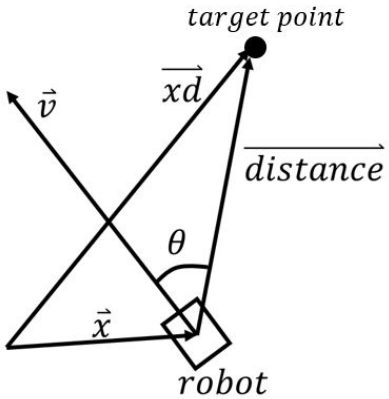
Rotational speed

$$\pm\ (4 + \underbrace{\min\left(4|distance|^2, 3\right)}_{\text{takes effect when the robot is far from the target point}} + \underbrace{\frac{0.05}{|distance|^2 + 0.01}}_{\text{takes effect when the robot is near to the target point}})$$

bias

$$\overbrace{scaler_{rotate} * (1.3 - cos\theta)}^{scaler_{distance2rot}}$$

the part reflecting influence from θ



Demo Video

Normal speed: https://drive.google.com/file/d/1SIjrDl93RdoaB7c93GNVHmsPp5w93V0a/view?usp=sharing

8x speed: https://drive.google.com/file/d/1Ev4BGYuUePXea4E4gV0mWkTVYtVBBhBW/view?usp=sharing

Thanks for watching!

Autonomous Quadruped

Group 14 Awesome Dog

16.08.2022

Dian Yuan

Yinglei Song

Zhelin Yang

Hang Li

Zhaoqi Zhou