SLAM Based Autonomous Navigation of Mobile Robot

Project Members:

Bimal Paneru(069/BEX/408)

Niraj Basnet(069/BEX/422)

Rabin Giri(069/BEX/431)

Sagar Shrestha(069/BEX/436)

Project Supervisor:

Mr. Dinesh Baniya Kshatri

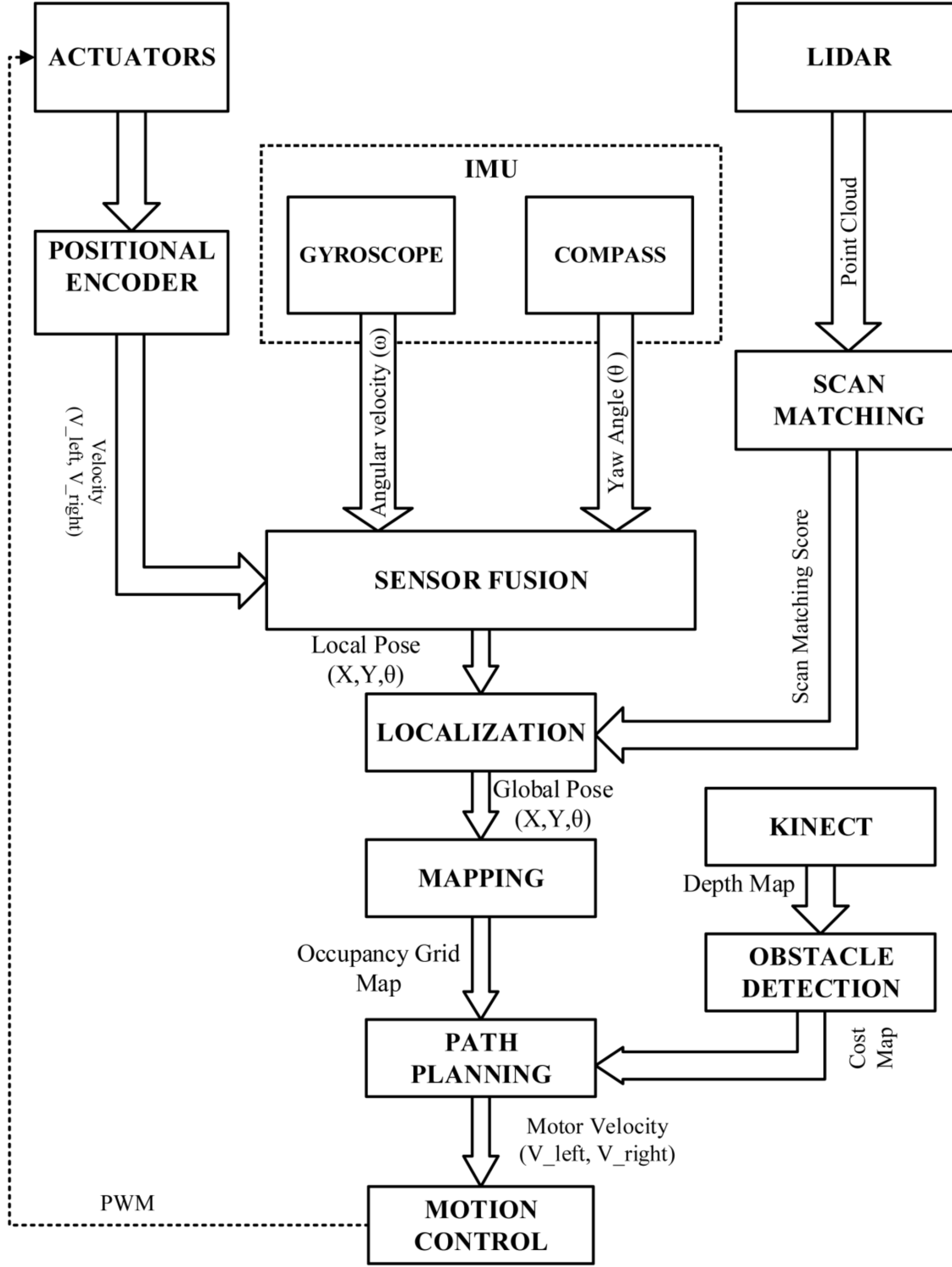
INTRODUCTION

- Autonomous Navigation
  - Navigation without human intervention
- Simultaneous Localization and Mapping (SLAM)
  - Chicken-egg problem of robotics
- Localizing, mapping and navigating from one location to another.

NAVIGATION SYSTEM

- SENSOR FUSION
- LOCALIZATION
- MAPPING
- SLAM
- PATH PLANNING
- MOTION CONTROL

SYSTEM BLOCK DIAGRAM



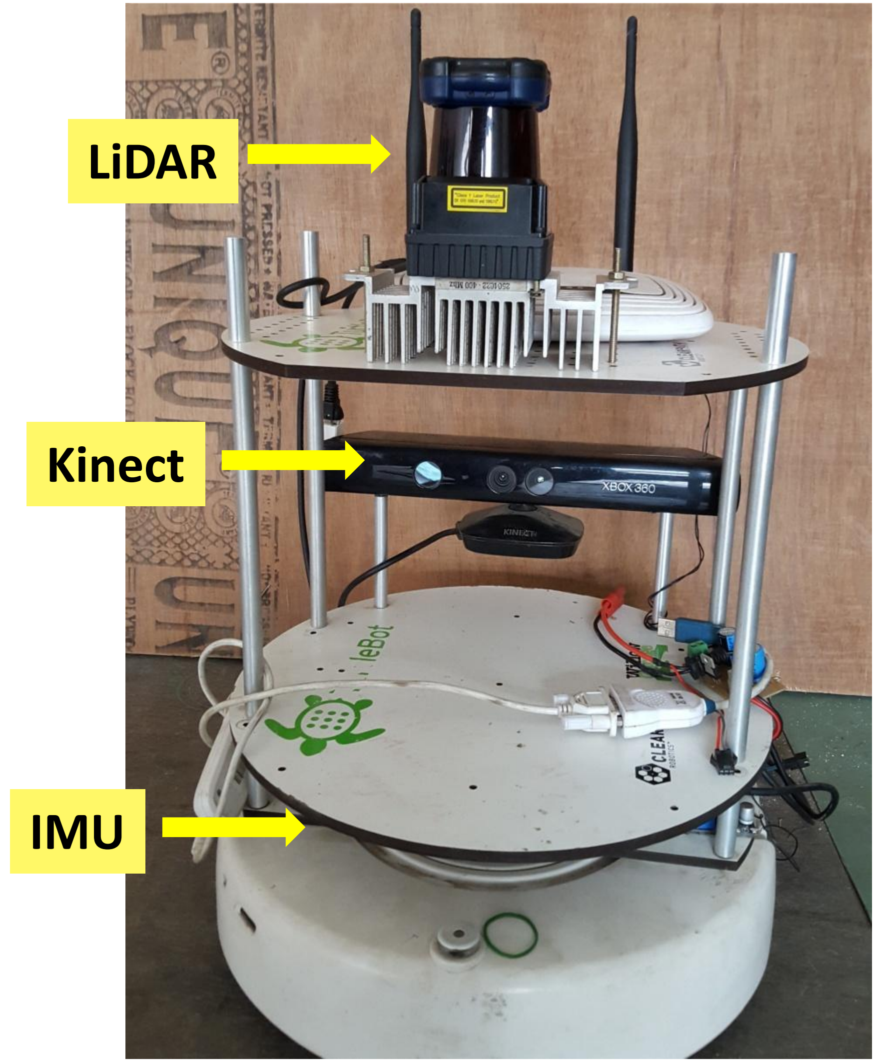
APPLICATIONS

Cleaning with Navigation (left)

Automatic Transportation of load (right)

TOOLS AND PLATFORMS

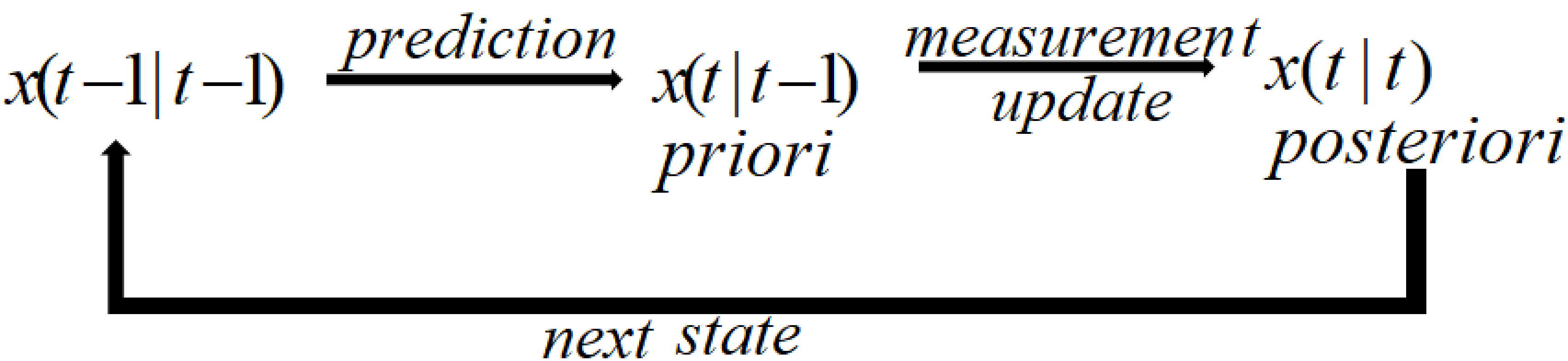
- **Hardware**
  - Inertial Measurement Unit (IMU)
  - Encoder
  - LiDAR
  - ARM
  - Kinect
- **Software**
  - Robot Operating System (ROS)
  - STM HAL
  - Keil uVision IDE



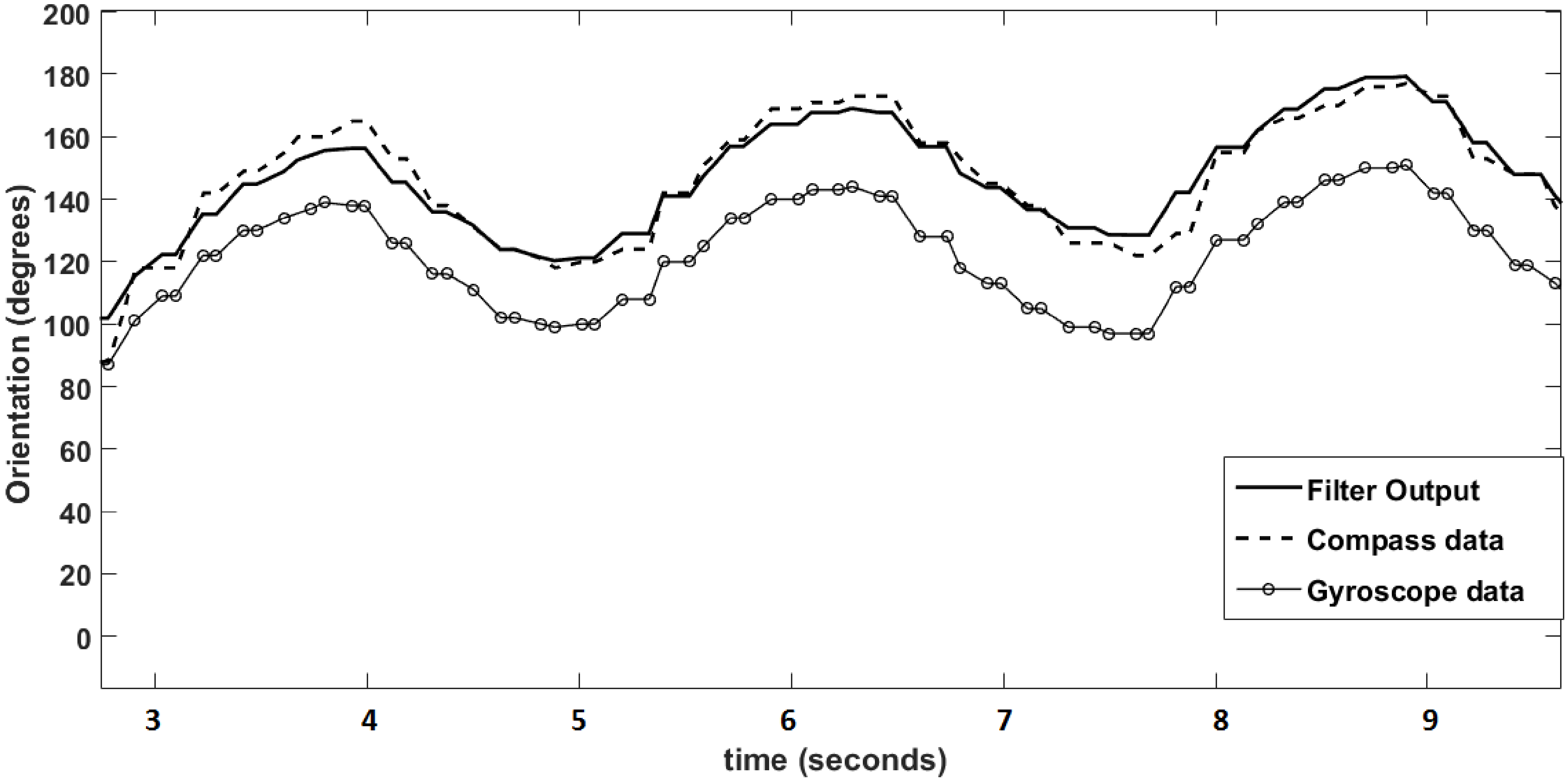
SENSOR FUSION

- Sources of odometry
  - Encoder + IMU (Gyroscope + Compass)
- Fusion output is corrected orientation and position

Algorithm : Kalman filter



SENSOR FUSION : Kalman Filter



LOCALIZATION

- Positioning the robot in world space.
- Appropriate algorithms to localize using multiple sensors data

Algorithms

- Monte Carlo Localization (MCL)
- Adaptive Monte Carlo Localization(AMCL)

$$p(x_{0:t} \mid u_{1:t}, z_{1:t}, m)$$

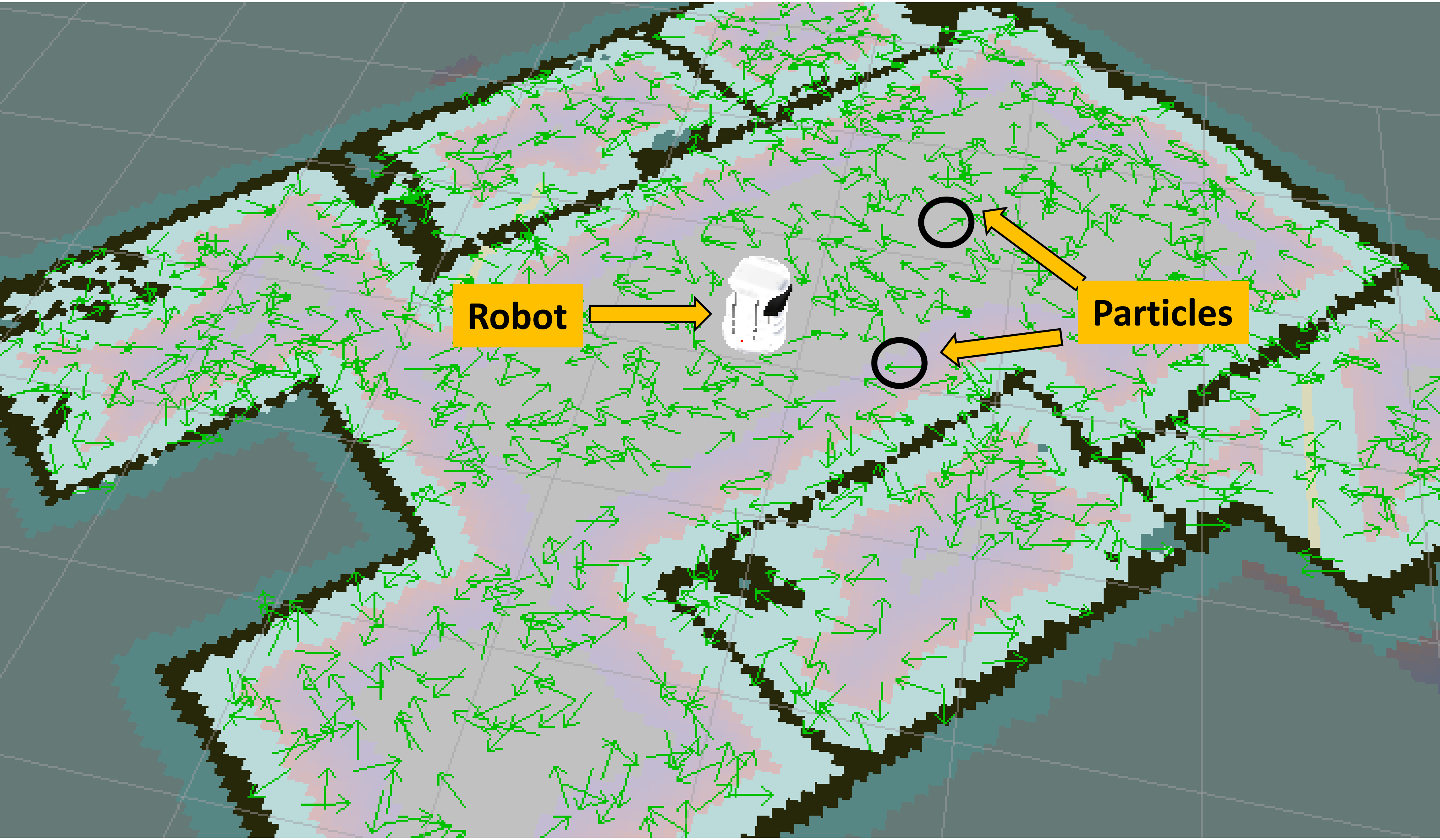
Where, $u_{1:t} = \{u_1, u_2, \ldots u_t\}$ ➡ Control input

$z_{1:t} = \{z_1, z_2, \ldots, z_t\}$ ➡ Observations

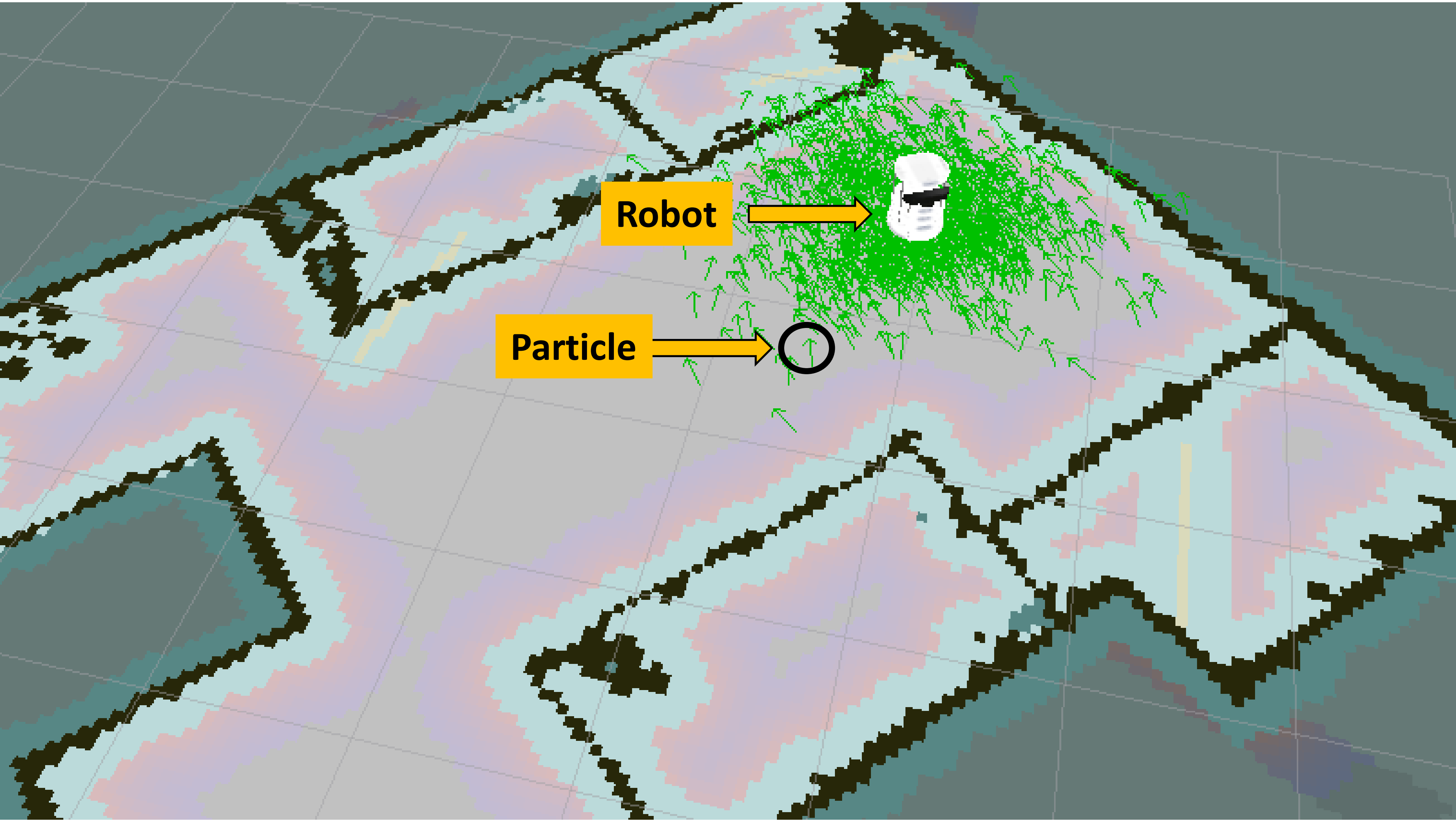
$x_{0:t} = \{x_0, x_1, \ldots, x_t\}$ ➡ Path of the robot

m ➡ Map of the environment

LOCALIZATION : Random Particle Injection



LOCALIZATION : Particle Converges



MAPPING

- Map Development of surrounding
- LiDAR fused with odometry data for reliability

Algorithm : Occupancy Grid Mapping

$$p(m \mid x_{1:t}, z_{1:t})$$

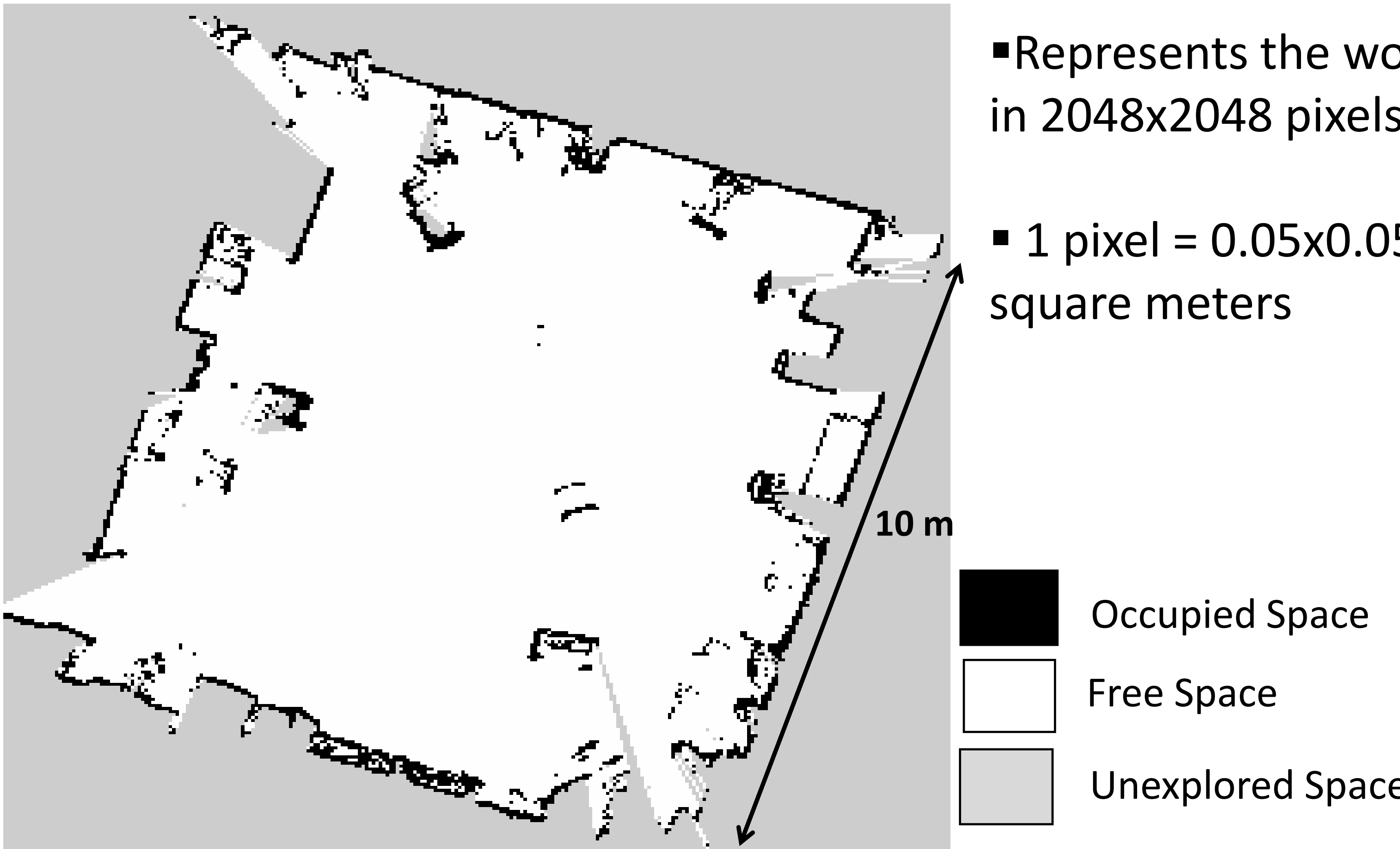
Where,

$z_{1:t} = \{z_1, z_2, \ldots, z_t\}$ ➡ Observations from time 1 to t

$x_{1:t} = \{x_1, x_2, \ldots, x_t\}$ ➡ Positions of the robot

m ➡ Map of the environment

MAPPING : Occupancy Grid Map



- Represents the world in 2048x2048 pixels
- 1 pixel = 0.05x0.05 square meters

PATH PLANNING : Global

- Optimum path planning between source and destination
- Graph Search for finding routes

Algorithm : A*

$$f(n) = g(n) + h(n)$$

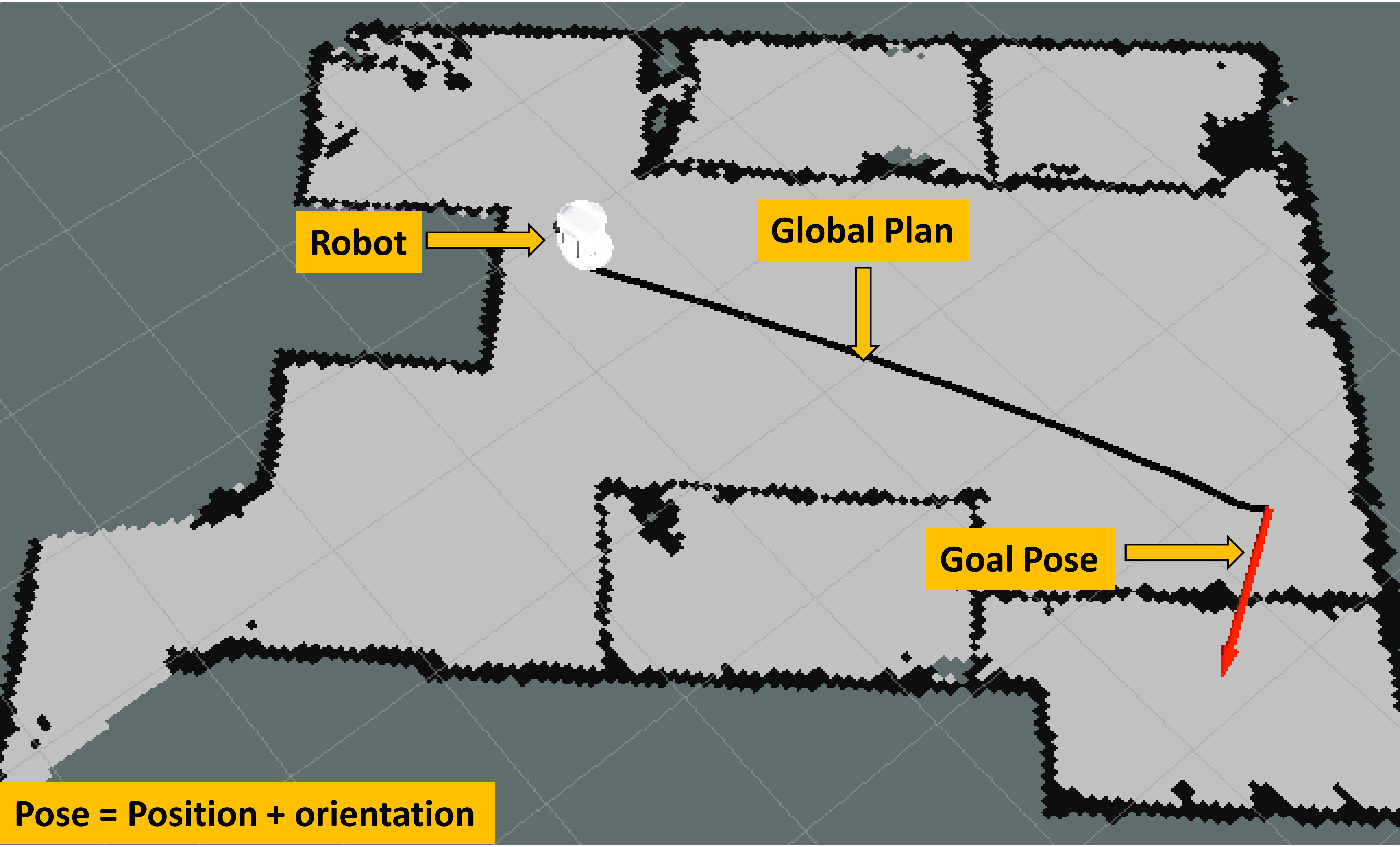
Where,

$f(n) \rightarrow$ Total cost of node n

$g(n) \rightarrow$ Cost of reaching node n from starting location

$h(n) \rightarrow$ Value of heuristic function at node n

PATH PLANNING : Global



PATH PLANNING : Local

- Execute segments of the Global plan sequentially

Algorithm : Dynamic Window Approach (DWA)

$$O(v,w) = a * heading\ (v,w) + b * velocity\ (v,w) + c * clearance\ (v,w)$$

Where,

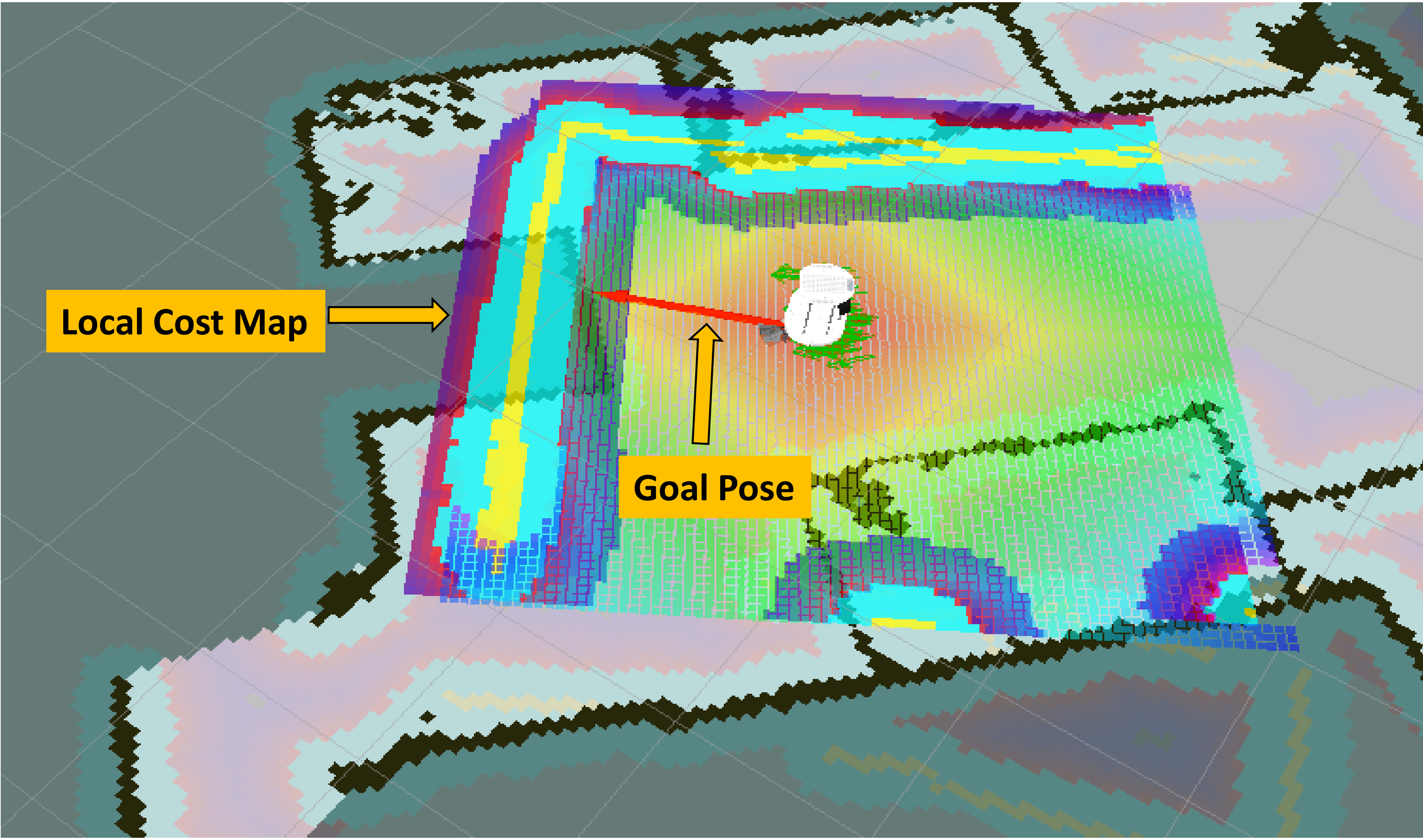
$O(v,w)$ ⟶ Objective function for linear velocity *v* and angular velocity *w*

$heading\ (v,w)$ ⟶ Heading towards goal

$clearance\ (v,w)$ ⟶ Clearance from obstacle

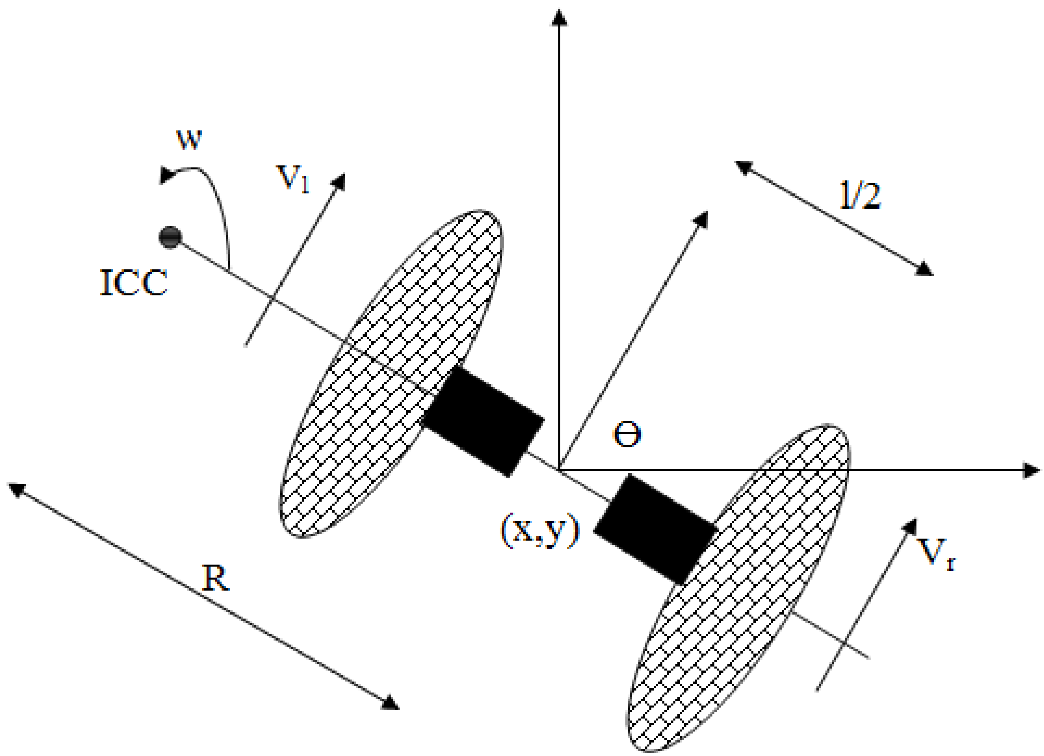
$velocity\ (v,w)$ ⟶ Forward velocity of the robot

PATH PLANNING : Local



MOTION CONTROL : Kinematics

- Differential Drive Kinematics

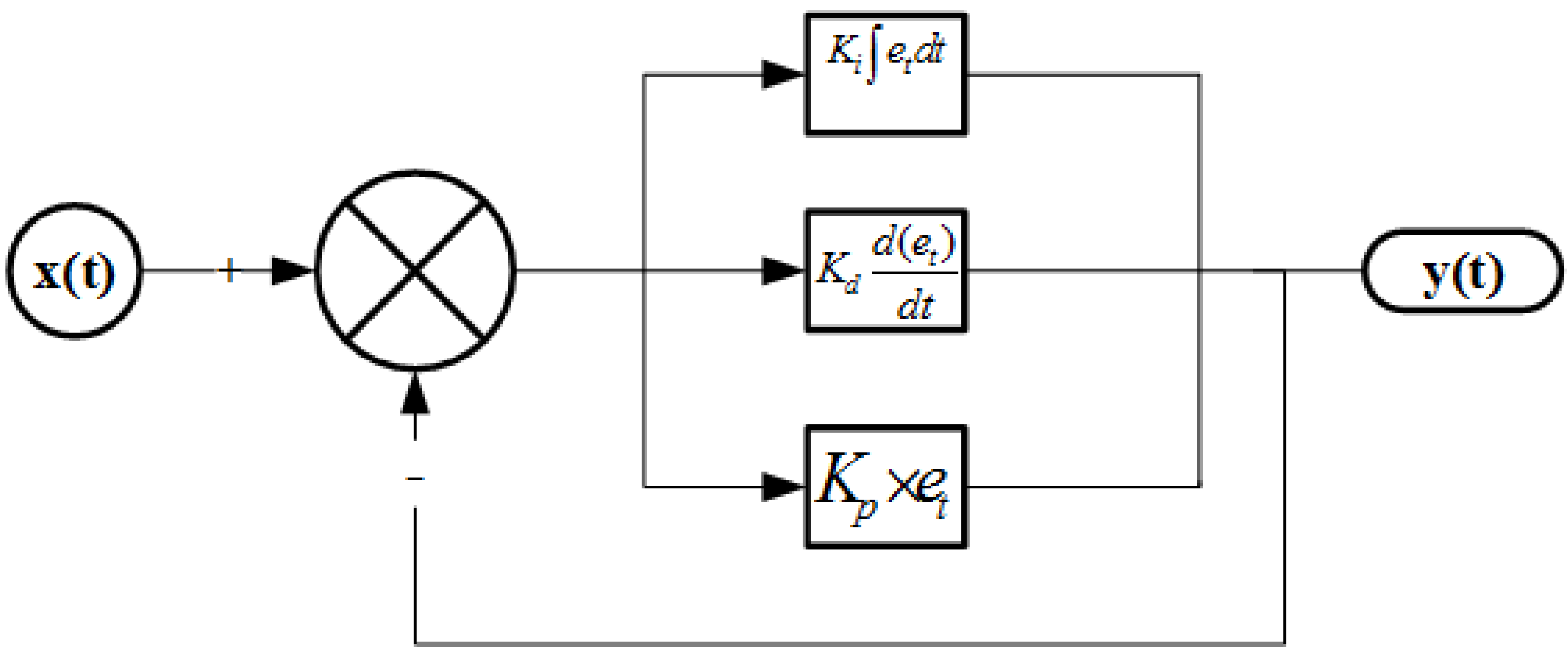


$$w(R+l/2)=V_r$$

$$w(R-l/2)=V_l$$

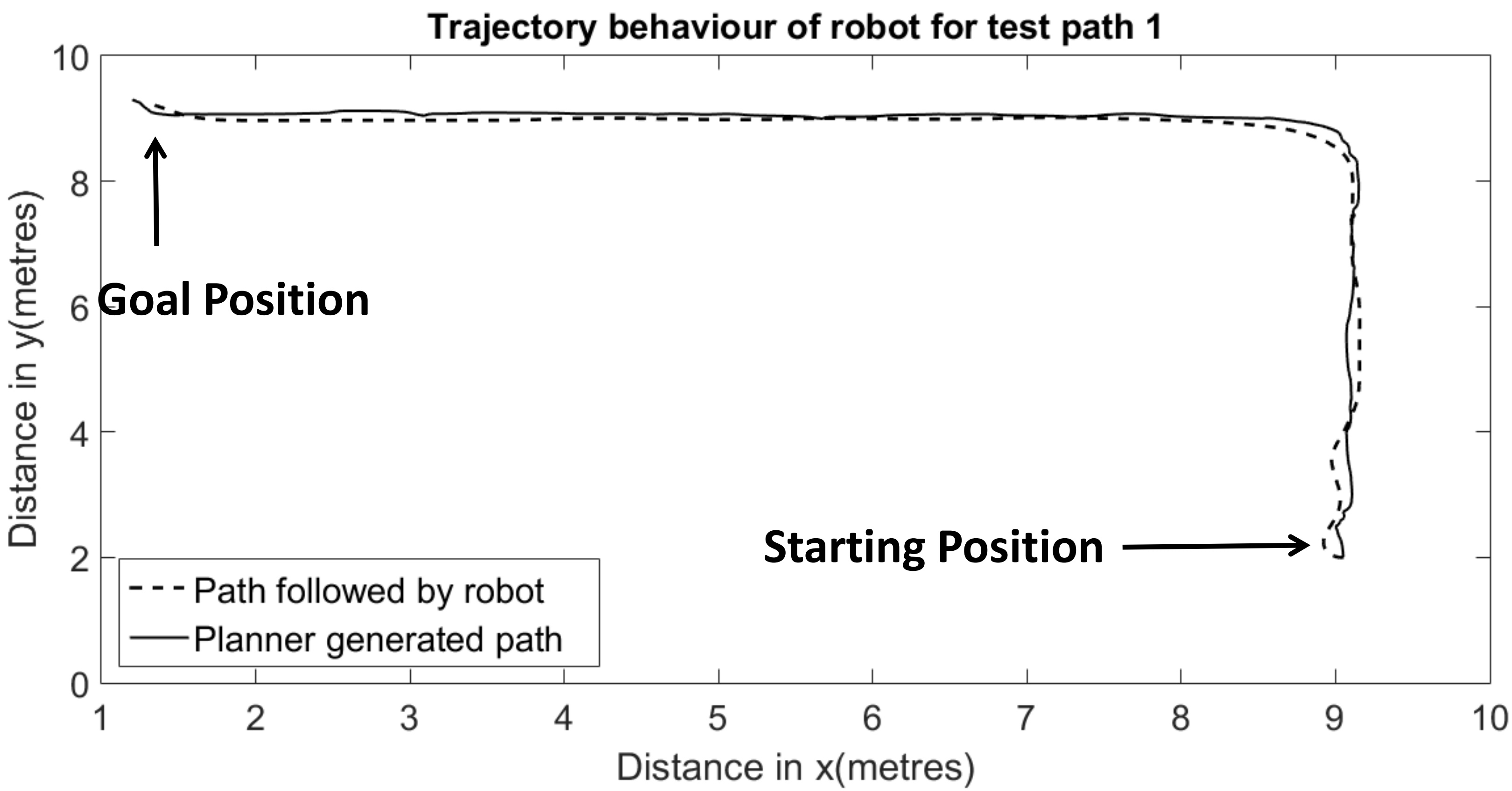
MOTION CONTROL : PID

- PID control on the motors to attain desired velocities



COMPLETE NAVIGATION

- Path Planned and actual trajectory followed by robot



COMPLETE NAVIGATION

Test path	Final Plan position(x_p,y_p)(m)	Final postion of robot(x_r,y_r)(m)	Deviation at goal(m)
1	(1.367,9.194)	(1.346,9.211)	0.027
2	(4.296,1.984)	(4.45,2.05)	0.1675
3	(4.037,3.1)	(3.985,3.054)	0.0694
4	(9.15,4.5)	(9.25,4.38)	0.1562

- Maximum deviation from goal in four experiments = 16.75 cm

CONCLUSION

- Sensor Fusion of IMU data for orientation with 2 degree of maximum error
- Localization with MCL using 500 particles
- Mapping with grid size 2048x2048 and scale: 1 pixel equivalent to 0.05 meters
- Path planning with less than 0.2 meters deviation on final location during navigation

Thank you!...