Semester Project

Design and evaluation of a SLAM Algorithm for a Small-Scale Multi-Robot System

Final Presentation

Student: Hugo Grall Lucas
Supervisor: Faëzeh Rahbar

Professor: Martinoli Alcherio

Contents

[1] Illustration: https://media.istockphoto.com/photos/question-mark-red-3d-interrogation-point-punctuation-mark-asking-sign-picture-id862048490?k=6&m=862048490&s=612x612&w=0&h=rLsmtXa1SMA9oZnM0Jt-BdHNm-YiA7CRihTdtu3pM7w= [visited: 29.04.2019]

Objectives:

- ❑ Find an algorithm to perform a Multi-Robot SLAM
- ❑ Find an algorithm to perform a Sonar SLAM
- ❑ Find a C/C ++ library for SLAM
- ❑ Setup a Lidar-based SLAM on simulation through Webots
- ❑ Setup a Sonar-based SLAM on simulation through Webots

[2] Illustration: https://cdn.shopify.com/s/files/1/1222/4922/products/k4-detail1_grande_be88e514-f196-45f7-94da-fdfdd413d76c_1024x1024.png?v=1459467937 [visited : 29.04.2019]

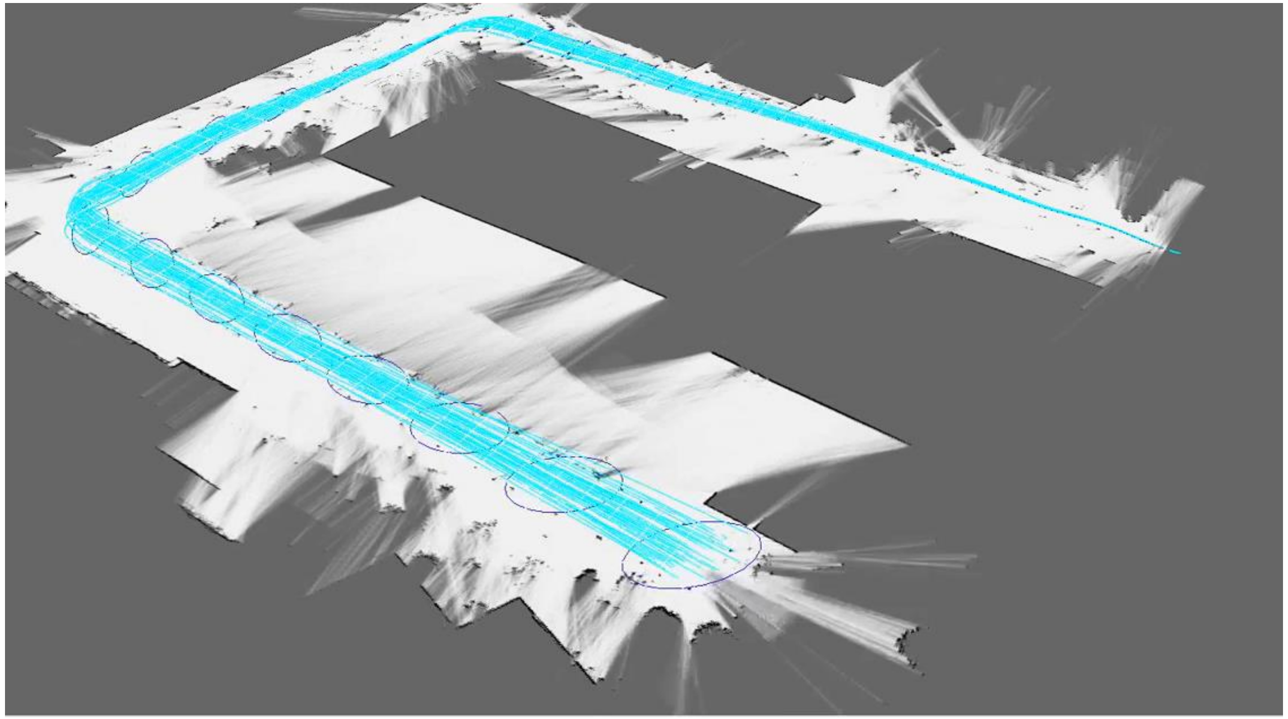
Library: MRPT v.1.9

Advantages

- Open source C/C++
- Good Doxygen documentations
- Updated regularly on GitHub
- ROS package for RBPF-SLAM

Disadvantages

- Sonar SLAM not implemented
- Need G++-7 (use C++17)
- No code example for Webots
- Not already used on K4



RBPF-SLAM performed off-line on a 2D-Lidar dataset

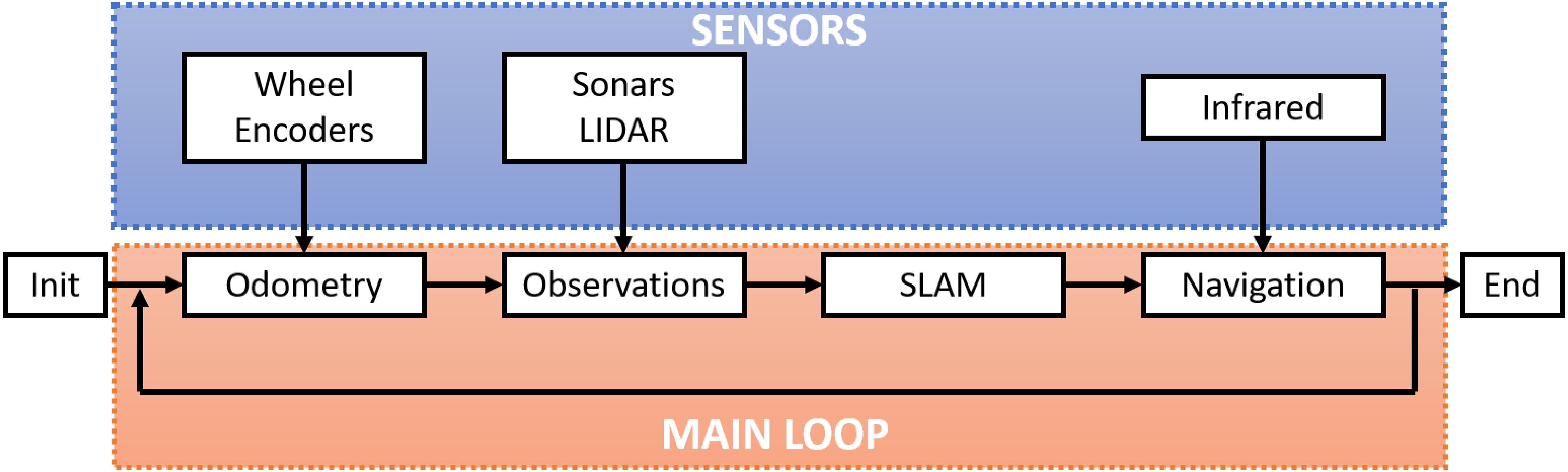
MRPT :	**M**obile **R**obot **P**rogramming **T**oolkit

[3] Video: https://www.youtube.com/watch?v=JHTOd8yRJJc&list=PLOJ3GF0x2_eX7TXG_ZyDgEIJdNcuB193X&index=13 [visited: 29.04.2019]

Contents

[4] Illustration: https://fixingtheeconomists.files.wordpress.com/2014/05/differentMethodtosuccess.jpeg [visited: 29.04.2019]

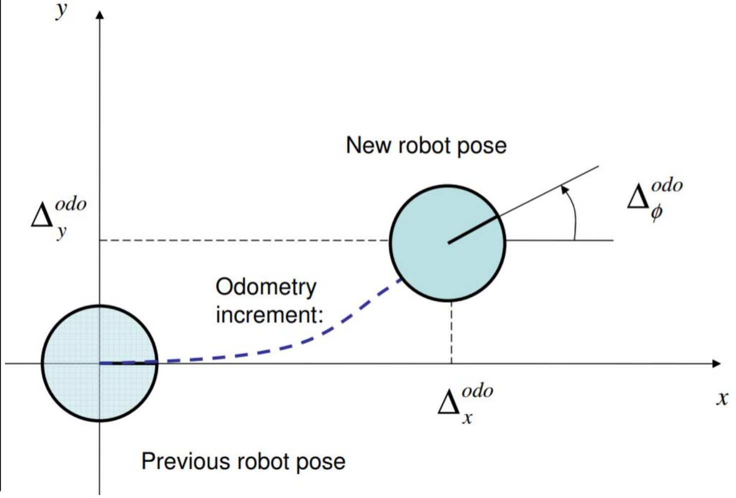
Architecture:



Odometry:

Odometry

- Read wheel encoders values
- Compute Δ_s and Δ_ϕ (from standard odometry)
- Compute odometry in local frame (eq. 3.1.2 and 3.1.3)



Odometry equations in local frame

$$R = \frac{\Delta s}{\Delta \phi} \tag{3.1.2}$$

If R $\neq$ 0   Else

$$\begin{aligned} \Delta_x^{odo} &= R \sin \Delta\phi & \Delta_x^{odo} &= \Delta s \\ \Delta_y^{odo} &= R(1 - \cos \Delta\phi) & \Delta_y^{odo} &= 0 \end{aligned} \tag{3.1.3}$$

Uncertainties

$$\sigma_{\Delta_x^{odo}} = \sigma_{\Delta_y^{odo}} = \sigma_{xy}^{\min} + \alpha_1 \sqrt{(\Delta_x^{odo})^2 + (\Delta_y^{odo})^2} + \alpha_2 \left|\Delta_\phi^{odo}\right|$$

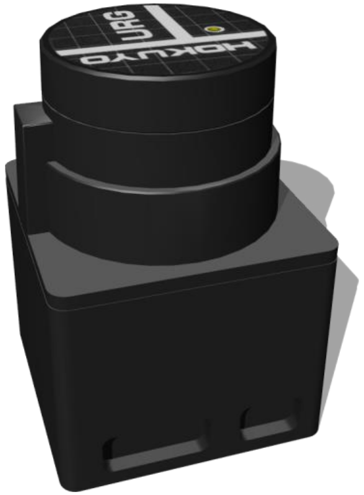
$$\sigma_{\Delta_\phi^{odo}} = \sigma_\phi^{min} + \alpha_3 \sqrt{(\Delta_x^{odo})^2 + (\Delta_y^{odo})^2} + \alpha_4 \left|\Delta_\phi^{odo}\right|$$

Standard MRPT Values

$$\alpha_1 = 0.05 \quad \alpha_2 = 0.001 \quad \alpha_3 = 5$$
$$\alpha_4 = 0.05 \quad \sigma_{xy}^{\min} = 0.01 \quad \sigma_\phi^{\min} = 0.2$$

Observations:

Lidar



Field Of View	240	[deg]
Range	0.2-5.6	[m]
Resolution	667×0.36	[deg]

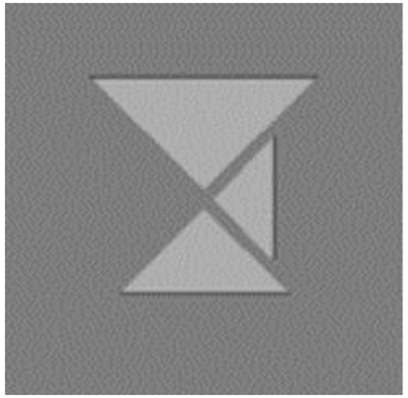
Sonar

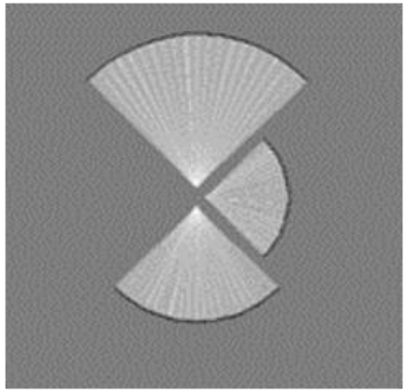
Field Of View	272	[deg]
Range	0.25-2.00	[m]
Beam angle	92	[deg]

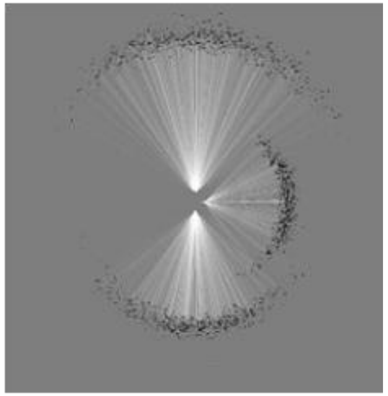
Observation cont'd:

Sonar beam models

- Single point per measurement
- Uniform distribution around a circle
- 2D-Gaussian distribution







2D-Gaussian

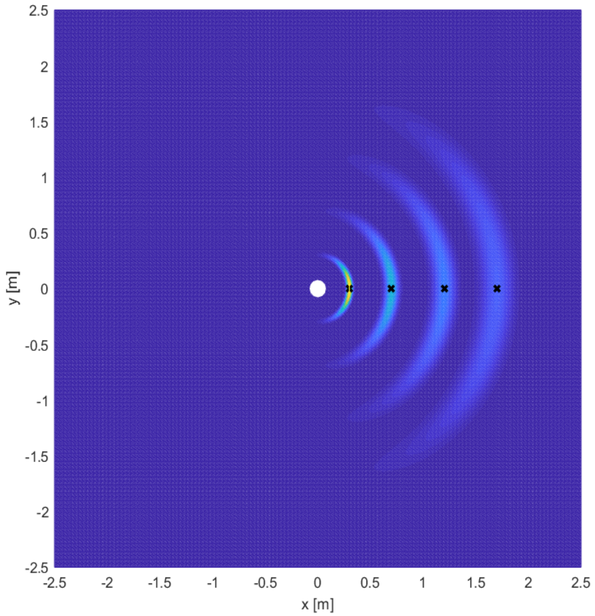
$$\phi_{i,j} = \phi_i + \phi_j \qquad (3.2.2)$$

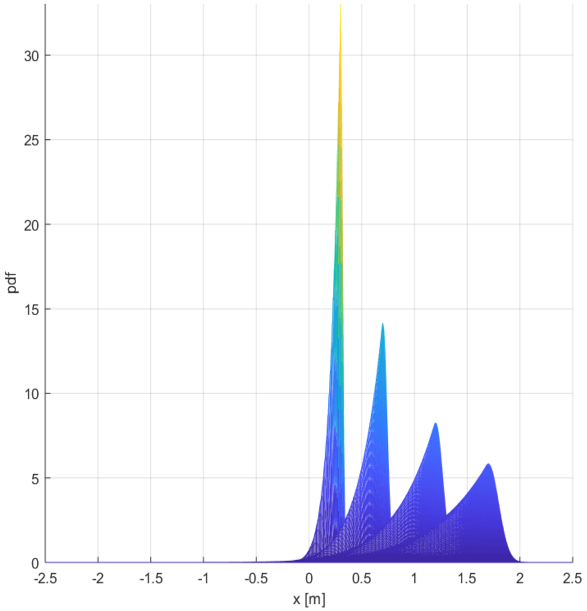
$$distance_{i,j} = distance_i + distance_j \qquad (3.2.3)$$

$$\begin{pmatrix} distance_j \\ \phi_j \end{pmatrix} \sim \mathcal{N}_N(\mu = 0, \Sigma) \qquad (3.2.4)$$

$$\Sigma = \begin{bmatrix} \sigma_{distance} & 0 \\ 0 & \frac{CONE\_APERTURE}{2} \end{bmatrix} \qquad (3.2.5)$$

$$\sigma_{distance} = \frac{distance_i}{20} \qquad (3.2.6)$$





RBPF-SLAM grid based

RBPF : **R**ao-**B**lackwellised **P**article **F**ilter

Main steps:

1) Initialize: $particle_i = \{x_i, y_i, \phi_i, w_i, Map_i\}$
2) Update 2D poses through motion model
3) If LocalizationStep:
   1. Standard Particle filter localization
   2.Update particles weight
4) If GlobalMapUpdateStep:
   1. Standard Particle filter localization
   2. Update particles weight
   3. Add new observations
5) If n_{eff} is below a threshold:
   1. Resampling the particles

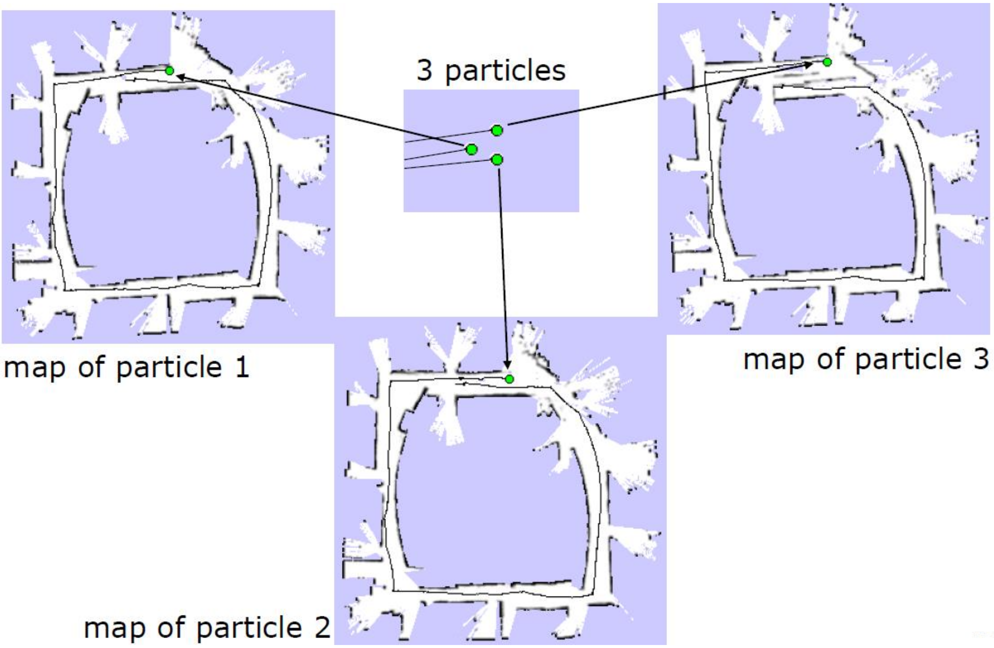
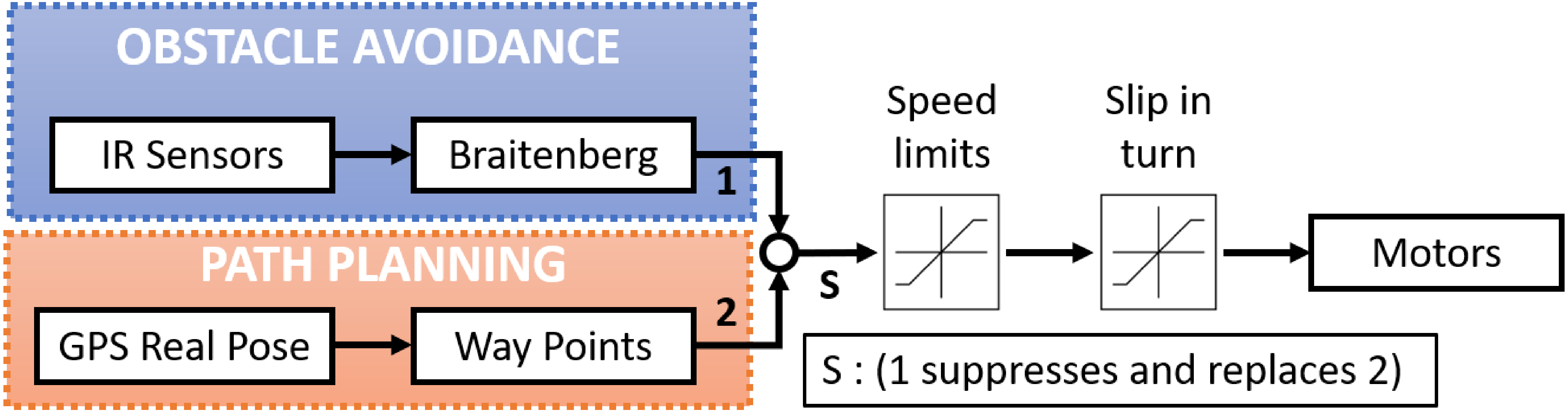


Illustration of the different local maps

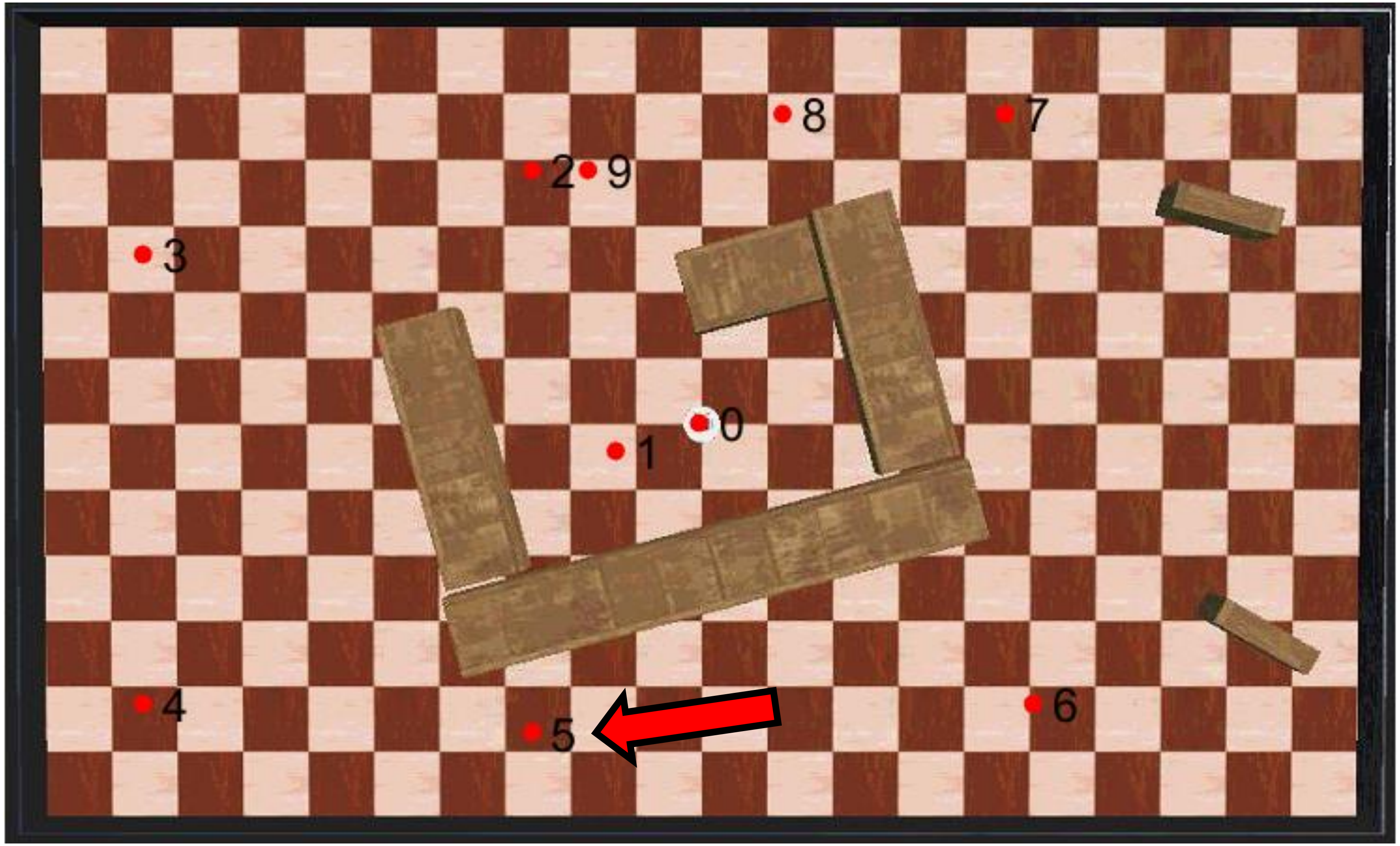
$$n_{eff} = \frac{1}{\sum_i \left(w_t^{[i]}\right)^2}$$

[5] Illustration: [Tipladi and Burgard, 2018] course

Navigation:

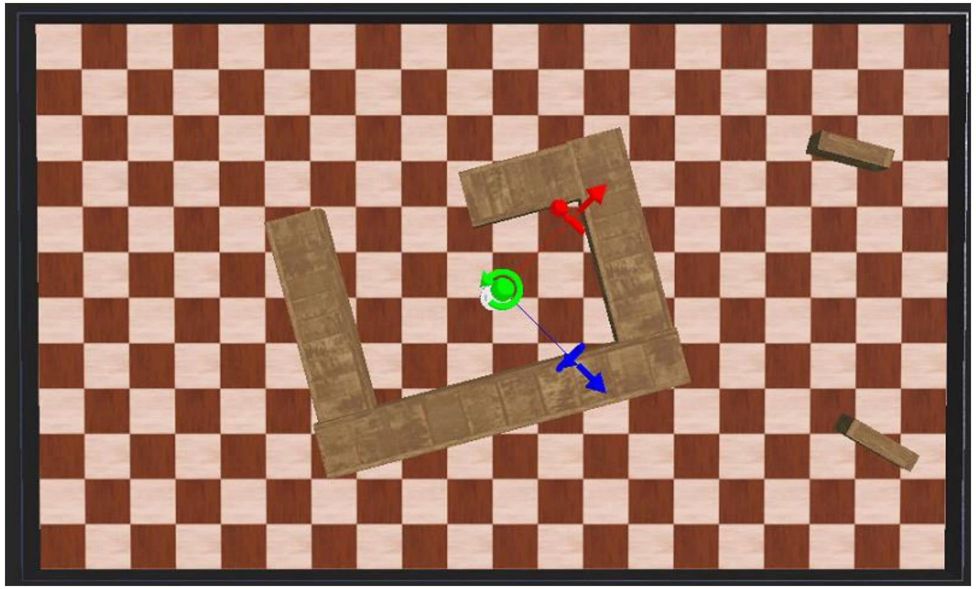

OBSTACLE AVOIDANCE
IR Sensors
Braitenberg
1
PATH PLANNING
GPS Real Pose
Way Points
2
S
Speed limits
Slip in turn
Motors
S : (1 suppresses and replaces 2)

Navigation: Way Points

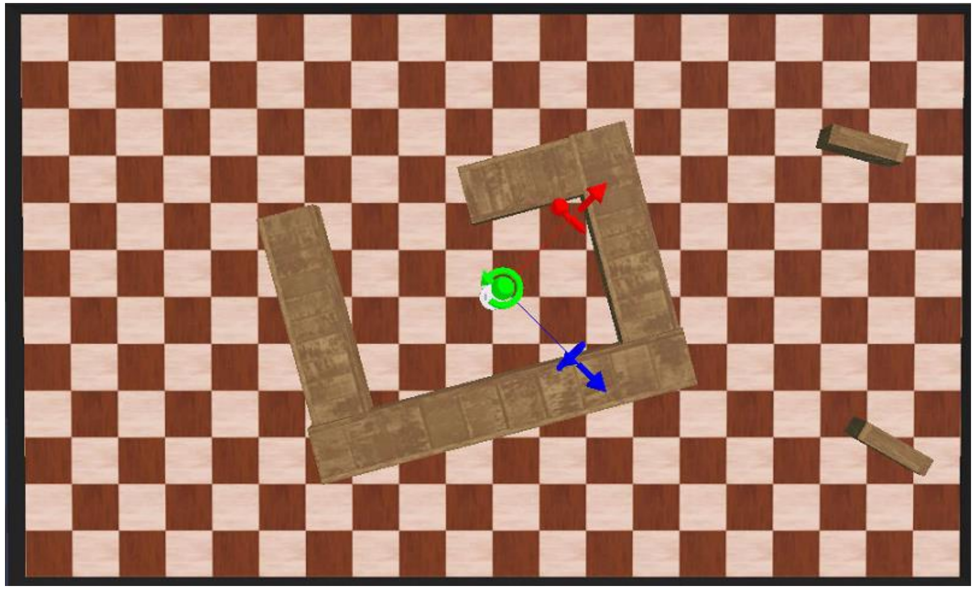


Navigation: Speed and Acceleration

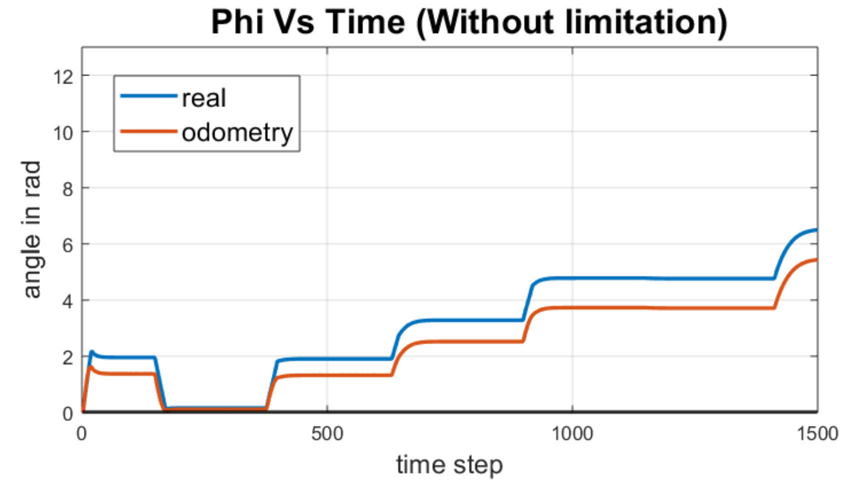
Without Limitation

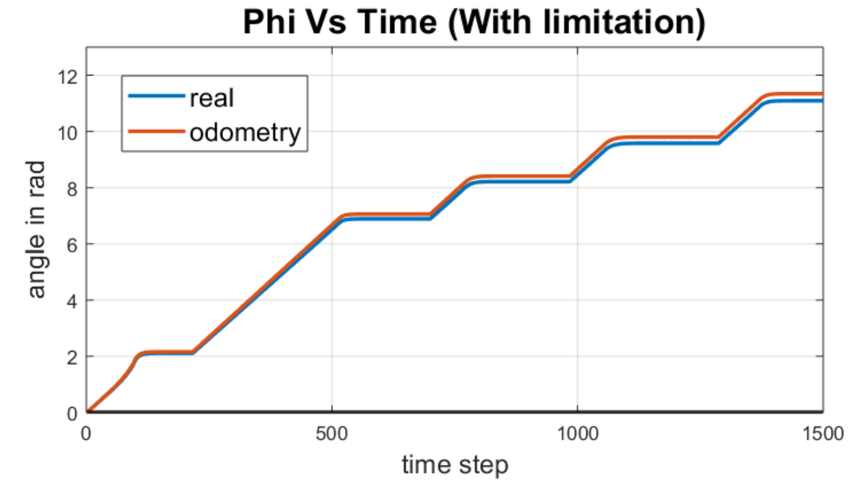


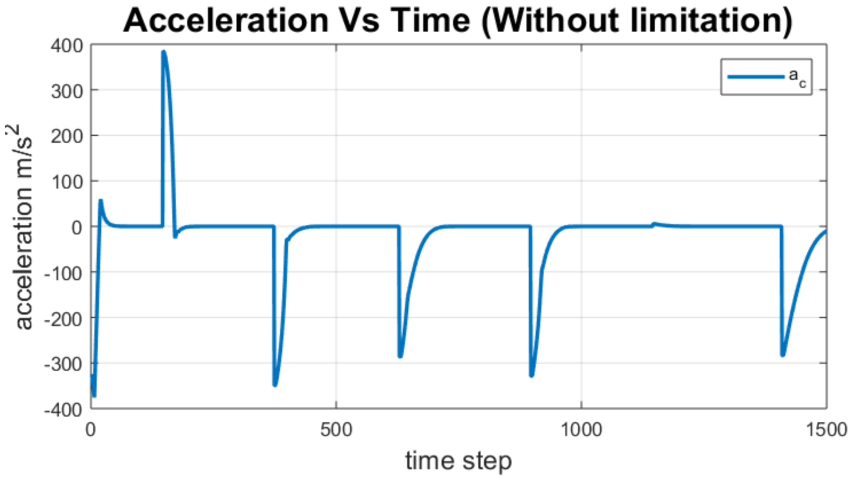
With Limitation

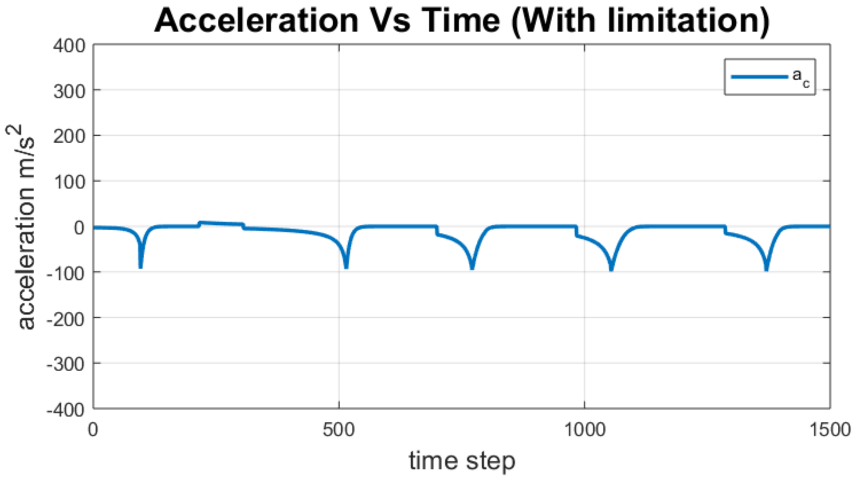


Navigation: Speed and Acceleration









Contents

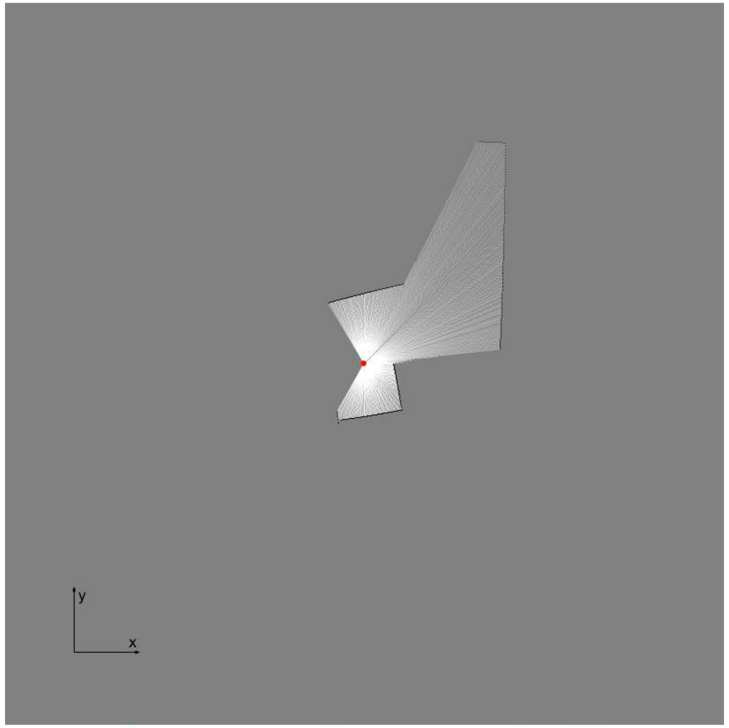
[7] Illustration: https://us.123rf.com/450wm/ribah/ribah1602/ribah160200034/54062669-stock-illustration-3d-illustration-of-man-sitting-on-word-text-result-3d-rendering-of-human-people-character.jpg?ver=6 [visited: 29.04.2019]

Parameters

	L1	S1	S2
insertionLinDistance	0.5m	0.1m	0.1m
insertionAngDisatance_deg	20°	10°	10°
localizationLinDistance	0.5m	0.07m	0.07m
localizationAngDistance_deg	20°	5°	5°
motionModelParameters	default	default	default
PF_algorithm	pfAuxiliaryPFOptimal (Scan Matching)	pfStandardProposal	pfStandardProposal
adaptiveSampleSize	Fixed	Fixed	Fixed
resamplingMethod	prMultinomial (standard)	prMultinomial (standard)	prMultinomial (standard)
sampleSize	25	10	10
Beta	0.5	0.5	0.5
min_x/y	x:-4 y:-4	x:-4 y:-5	x:-4 y:-5
max_x/y	x:4 y:4	x:4 y:5	x:4 y:5
resolution	0.01	0.01	0.01
maxOccupancyUpdateCertainty	0.60	0.57	0.57
maxFreenessUpdateCertainty	0.60	0.3	0.3
nbSample	637	20	200
type	LIDAR	Sonar CIRCLE	Sonar 2D Gaussian
maxSpeed	20	10	10
maxAcceleration	150	100	100

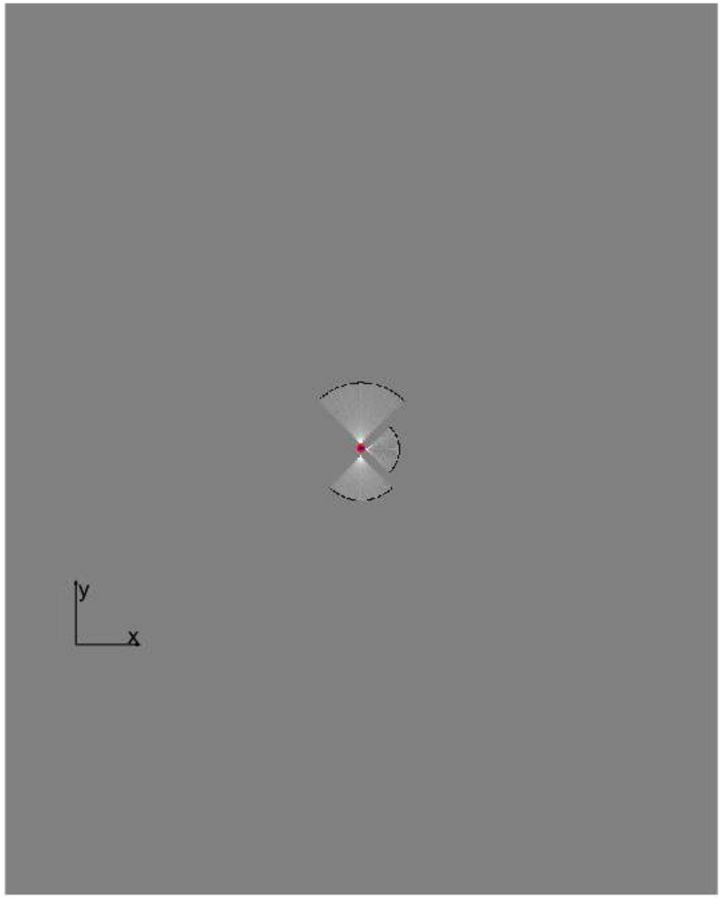
Mapping + particles

Lidar (L1)



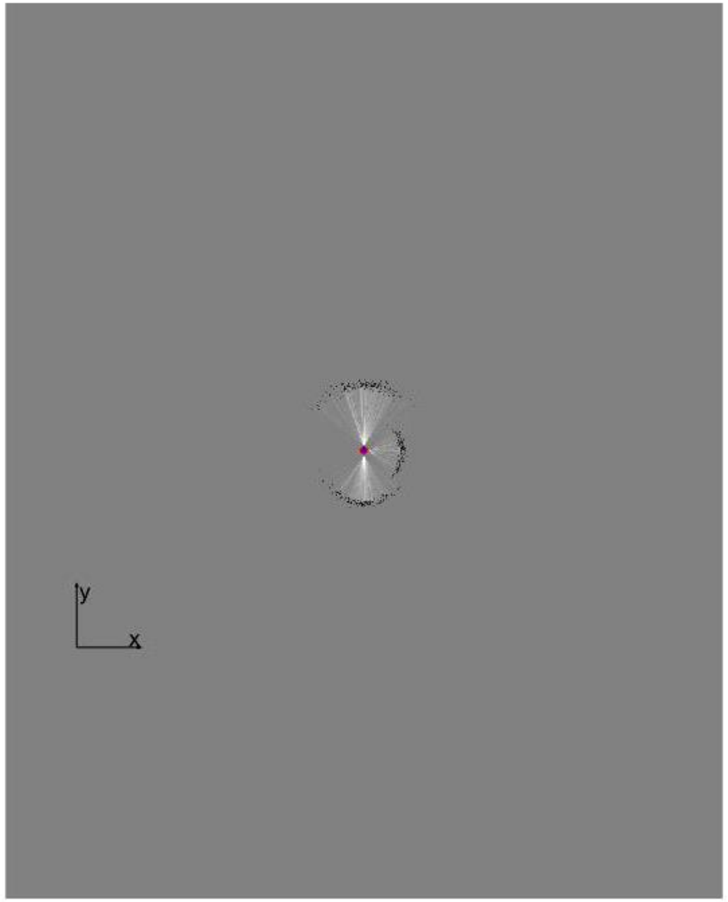
Slam time step : 1 - Simulation time : 00:0:0.032

Sonar (S1)



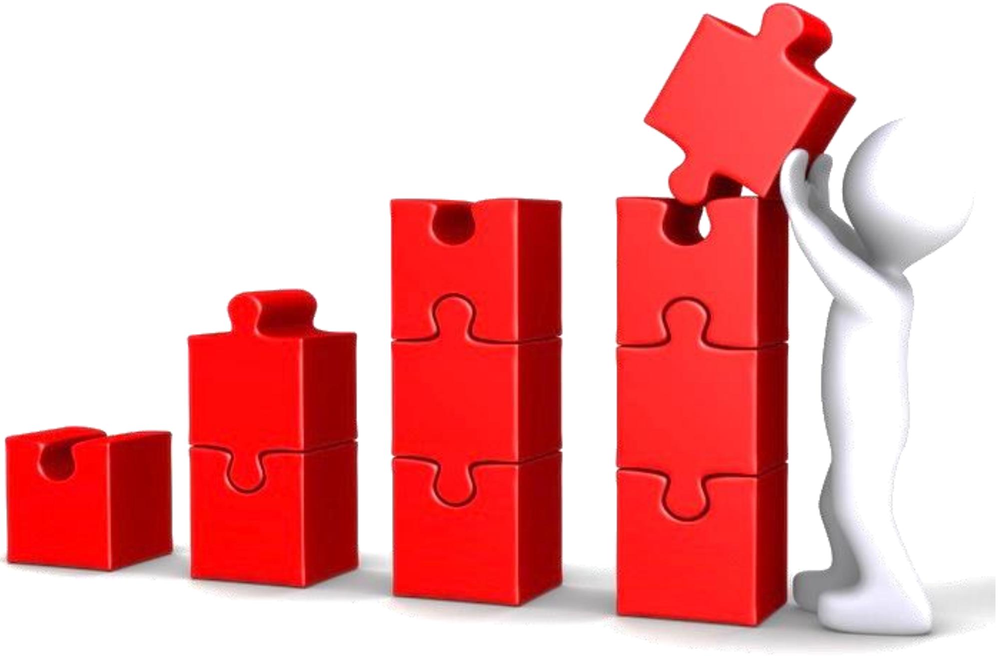
Slam time step : 1 - Simulation time : 00:0:0.032

Sonar (S2)



Slam time step : 1 - Simulation time : 00:0:0.032

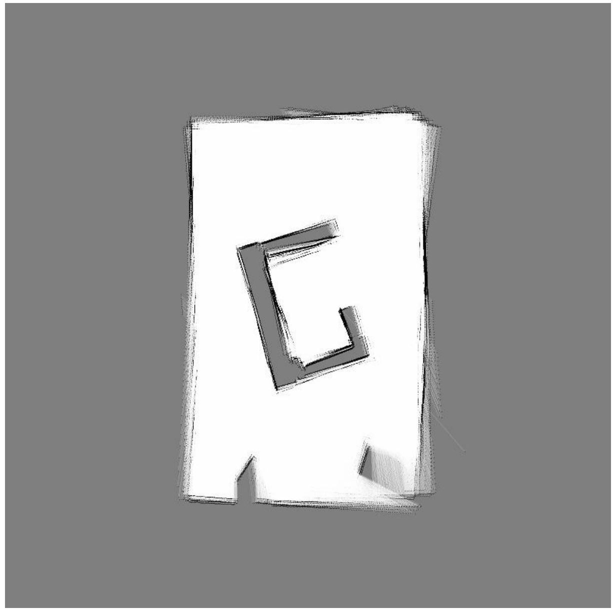
Contents



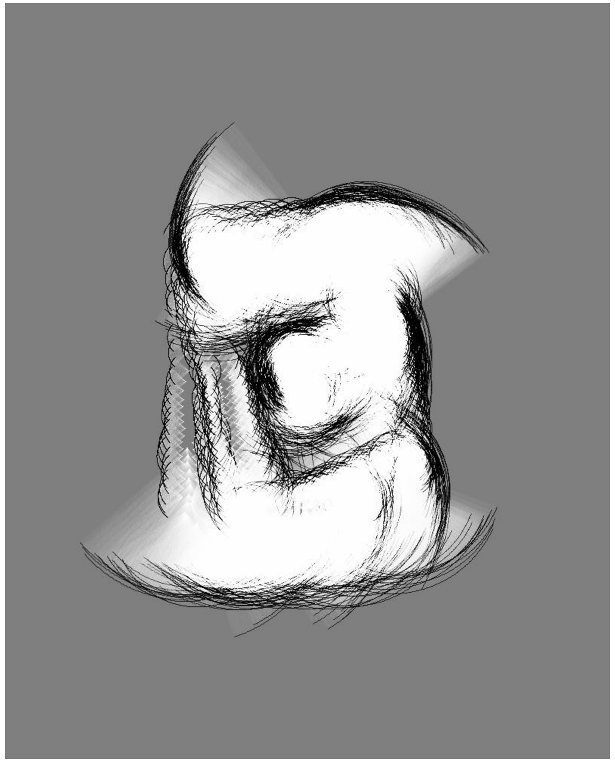
[8] Illustration: http://cavventures.in/wp-content/uploads/2018/01/Build-business-2.jpg [visited: 29.04.2019]

Final maps

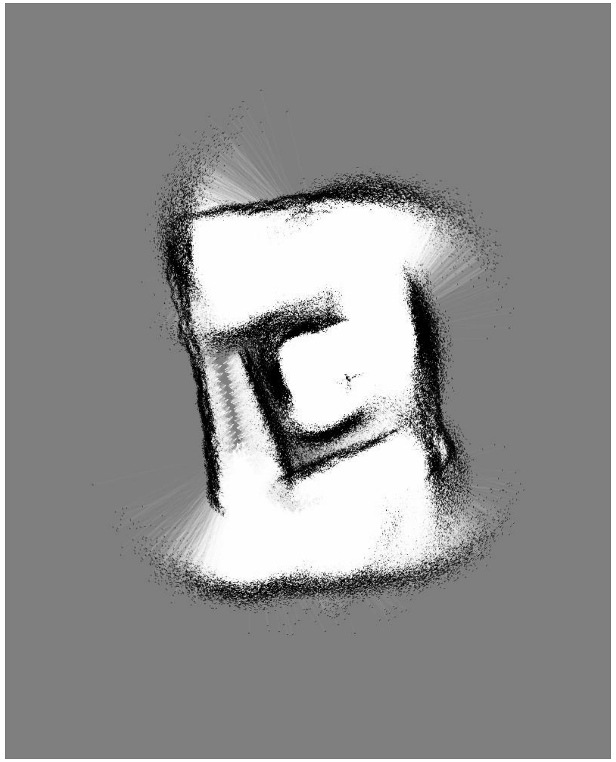
Lidar (L1)



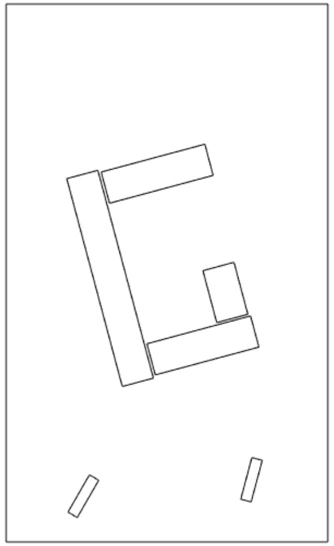
Sonar (S1)



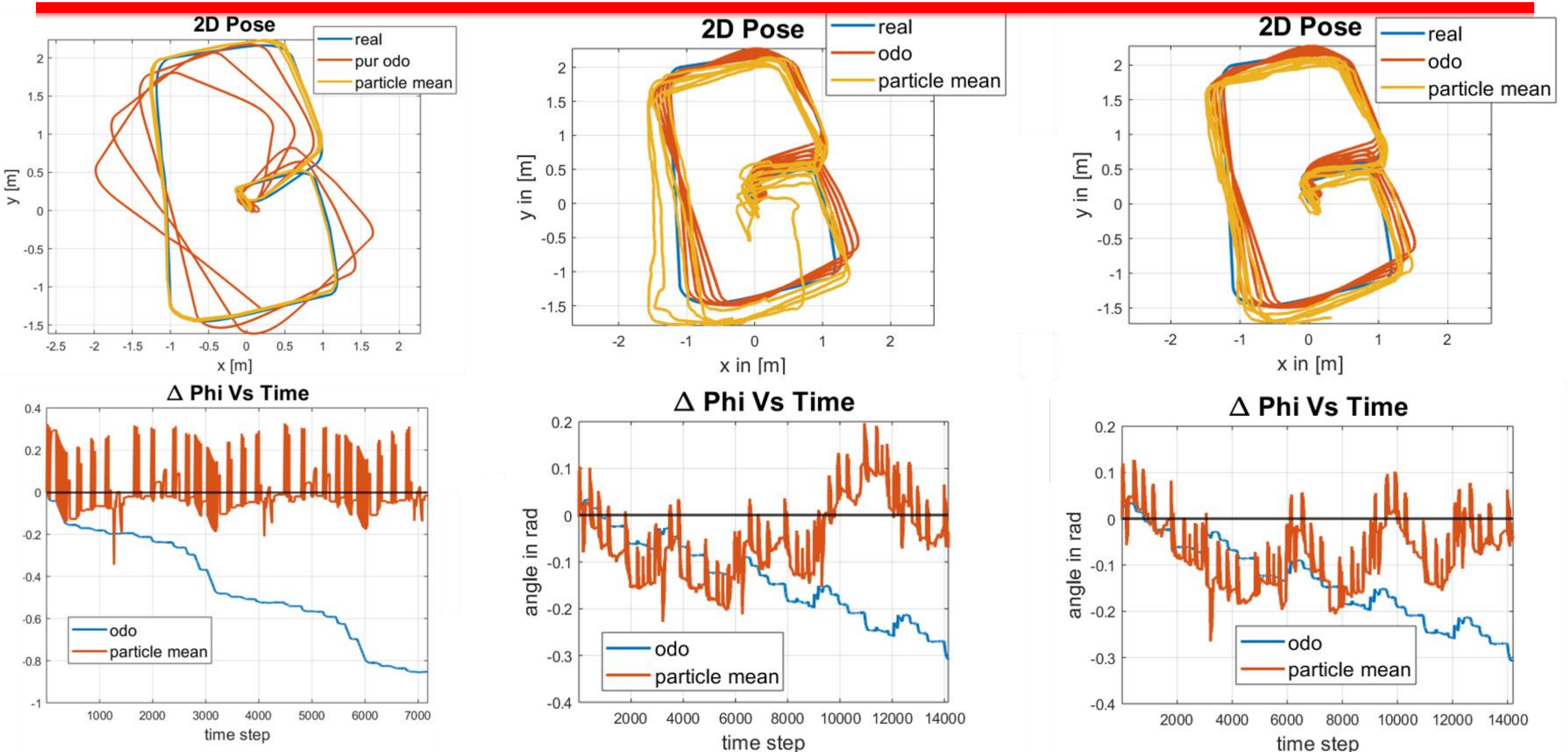
Sonar (S2)



Real



Pose Estimations



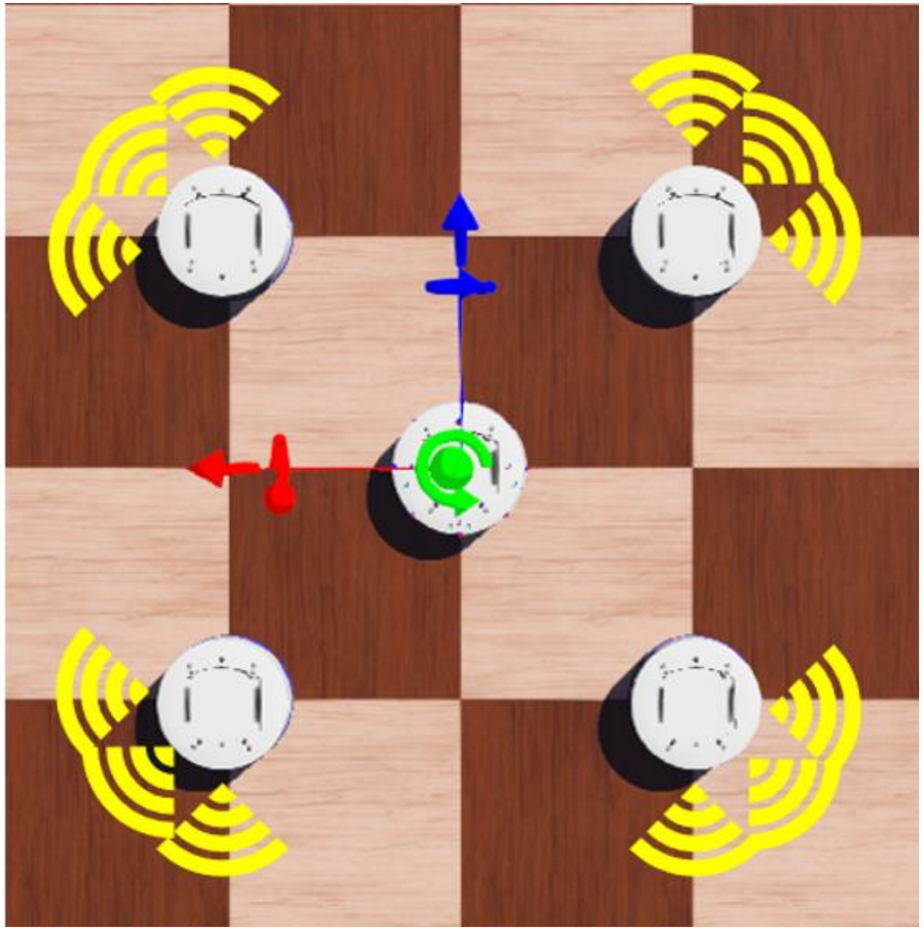
Discussion: Sonar-SLAM+ MR-SLAM

Sonar-SLAM: key points

- Odometry
- Sonar sampling scheme
- Path planning for good sonar measurements
- SPLAM

MR-SLAM with K4: idea

- Centralized (explorers + leader)
- Place the explorers to improve the measurements
- Network of moving sensors



Contents

[9] Illustration: http://permisautomoto.com/wp-content/uploads/sites/88/2016/05/Fotolia_10083216_Subscription_XL.jpg [visited: 29.04.2019]

Conclusion

- ☑ Find an algorithm to perform a Multi-Robot SLAM
- ☑ Find an algorithm to perform a Sonar SLAM
- ☑ Find a C/C ++ library for SLAM
- ☑ Setup a Lidar-based SLAM on simulation through Webots
- ☑ Setup a Sonar-based SLAM on simulation through Webots

References

[Simoes Martins,2013] Simões Martins J. A. (2013), *MRSLAM -Multi-Robot Simulaneous Localization and Maping*, Universidade de Coimbra (Portugal).

[Tipladi and Burgard, 2018] Tipaldi G.D. and Burgard W. (2018), *FastSLAM – Feature-Based SLAM with Particle Filters*, from the “Robot Mapping” course in Albert-Ludwigs-Universität Freiburg (Germany).

[Tipladi, Burgard and Spinello, 2018] Tipaldi G. D., Spinello L., Burgard W. (2018), *Grid-Based FastSLAM*, from the “Robot Mapping” course in Albert-Ludwigs-Universität Freiburg (Germany).

[Blanco Claraco, 2010] Blanco Claraco J. L. (2010), *Development of Scientific Applications with the Mobile Robot Programming Toolkit - The MRPT reference book*, Machine Perception and Intelligent Robotics Laboratory University of Malaga (Spain).

Questions

[10] Illustration: https://www.lecoindesentrepreneurs.fr/wp-content/uploads/2015/06/liste-de-questions-%C3%A0-poser-au-franchiseur.png [visited: 29.04.2019]