

1  
2  
3  
4  
5  
6  
7  
8  
9

# An Extended Analysis on Tuning the Parameters of Adaptive Monte Carlo Localization ROS Package in an Automated Guided Vehicle

10 Wallace Pereira Neves dos Reis\* · Guilherme José da Silva · Orides Morandin Junior ·  
11 Kelen Cristiane Teixeira Vivaldini

17 Received: date / Accepted: date

20 **Abstract** With a growth tendency, the employment of the  
21 Adaptive Monte Carlo Localization (AMCL) Robot Operational  
22 System (ROS) package does not reflect a more in-depth discussion on its parameters tuning process. The au-  
23 thors usually do not describe it. This work aims to extend the  
24 analysis of the package's parameters distinct influence in an  
25 Automated Guided Vehicle (AGV) indoor localization. The  
26 experiments test parameters of the filter, the laser model,  
27 and the odometry model. Extending the previous analysis  
28 of seven parameters, the present research discusses another  
29 ten from the 22 configurable parameters of the package. An  
30 external visual vehicle pose tracking is used to compare the  
31 pose estimation from the localization package. Although the  
32 article does not propose the best parameter tuning, its results  
33 discuss how each tested parameter affects the localization.  
34 The paper's contribution is discussing the parameters variation  
35 impact on the AGV localization. It may help new re-  
36 searchers in the AMCL ROS package parameter tuning pro-  
37 cess. The results show minor changes in the default param-  
38 eters can improve the localization results, even modifying  
39 one parameter at a time.

40 **Keywords** Automated Guided Vehicle · Adaptive Monte  
41 Carlo Localization · Laser Scanner · ROS

## 1 Introduction

42 A laser scanner with an Automated Guided Vehicle (AGV)  
43 improves its navigation possibilities. With the adequate tech-  
44 niques, it turns the AGV into an autonomous vehicle. Such a  
45 characteristic also brings flexibility to a manufacturing site.  
46 Nevertheless, the localization method is decisive to the nav-  
47 igation system performance. In the literature, one can find  
48 several solutions to the localization problem. However, it is  
49 not a closed issue.

50 The Robot Operating System (ROS) has been massively  
51 used in R&D projects in the last decade, and its use in prod-  
52 uct development and commercialization has likewise expanded [1,2]. The developers' community displays a relatively  
53 large number of packages – containing specific algorithms,  
54 nodes, messages – for free use and improvement. The Adap-  
55 tive Monte Carlo Localization (AMCL) [3] is one of the ref-  
56 erence packages, very common in the literature. The pack-  
57 age contains a well-known localization algorithm, which was  
58 implemented and tested in several pieces of research. Also,  
59 being available in ROS speeds up its application due to the  
60 code reuse and the community support. Nevertheless, there  
61 is always room for improvement in the package, as Section  
62 2 shows.

63 The AMCL ROS package has 22 configurable param-  
64 eters from a total of 47. The parameter tuning is a com-  
65 binatorial problem given the high search space due to the  
66 number of configurable parameters and their interaction [4].  
67 The package parameter description does not further analyze  
68 the parameter adjustment effect, giving little information on

69 This study was financed in part by the Coordenação de  
70 Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES)  
71 - Finance Code 001. Wallace dos Reis was financed in part by the  
72 Federal Institute of Education, Science and Technology of Rio de  
73 Janeiro—IFRJ, campus Volta Redonda.

74 Wallace Pereira Neves dos Reis, correspondent author  
75 TEAR - Laboratório de Estratégias de Automação e Robótica, De-  
76 partment of Computing, Federal University of São Carlos, SP, Brazil; and  
77 Federal Institute of Rio de Janeiro, campus Volta Redonda, RJ, Brazil;  
78 E-mail: wallace.reis@ufscar.br

79 Guilherme José da Silva  
80 PET-TEAR - Laboratório de Estratégias de Automação e Robótica, De-  
81 partment of Computing, Federal University of São Carlos, SP, Brazil;  
82 E-mail: guijoseh@gmail.com

83 Orides Morandin Junior and Kelen Cristiane Teixeira Vivaldini  
84 Department of Computing, Federal University of São Carlos, SP,  
85 Brazil;  
86 E-mail: {orides, vivaldini}@ufscar.br

the default values. Also, the research that uses the package usually does not disclose the tuning process or demonstrate how the tuned parameters impact the robot localization. The discussion of parameter adjustments is relevant, given the increasing use of the localization package and the few available materials discussing the parameters tuning.

This paper expands the previous analysis, describing the methods and results to guide the utilization in another robot, and pointing out the parameters that most influence the localization for a more punctual and more direct adjustment, leaving the default tuning. It aims to present the effects of the AMCL ROS package's parameter variation on AGV localization, which are commonly not discussed in papers. The study also elaborates on the analysis using both the covariance matrix and an external measurement to compare the AGV's estimated localization.

The paper is organized as follows: Section 2 presents a review of related works and Section 3 extends the AMCL ROS package parameters description from [5], and discuss the pose covariance matrix. Section 4 describes the hardware and software used in the experiments, also defining the data presentation and the analysis metrics. Section 5 defines the experimental setup, while Section 6 presents and discusses the experiments results. Finally, Section 7 is the work conclusions and future work.

## 2 Related Work

A brief search in the literature shows the increase in publications in both the Adaptive Monte Carlo Localization Algorithm research and using the same algorithm through the ROS package. Since it is not the aim of the present work, the bibliography research is not an exhaustive search and does not intend to represent the state-of-the-art matter. Using Scopus and Web of Science research databases, Figure 1 shows the research evolution in both domains.

With a specific search string ("Adaptive Monte Carlo Localization"), one can find 251 works concerning the algorithm application in several fields [6–9], its improvement [10–12], among other employments, in the cited databases.

The graph of Figure 1 shows the increase in such publications. Restricting the application field to the ROS AMCL package domain – using the search string "ROS" and "Monte Carlo" and "Localization" – the databases returned 55 works. Moreover, the number of publications is also increasing, following the first – and more comprehensive – search.

Among the works related to the AMCL ROS package, most simply apply the tool. For instance, [13] implement a mobile robot navigation control system using the AMCL package in a mobile service robot and applying AMCL as a standard package, [14] present a ROS-based framework for navigation algorithms testing.

Other researches that use the AMCL package are [15–19]. The cited authors do not discuss the AMCL parameter tuning, despite showing robot localization in the navigation systems. Deviating from the previous authors, [20] applied the ROS navigation stack, including the AMCL package, in a Data Center environment monitoring service-robot. The paper discusses some parameters tuning, for instance, a laser-related parameter that considers glass-covered racks surfaces.

Other researchers look to improve the package. For instance, [21] present an improvement to the algorithm proposing a new laser beam observation model. The method ignores laser beam measurements with highly correlated noise. Therefore, reducing the computational cost of the algorithm. Likewise, [22] demonstrate an observation model combining the laser-based grid map localization with visual landmarks acquired by an omnidirectional camera.

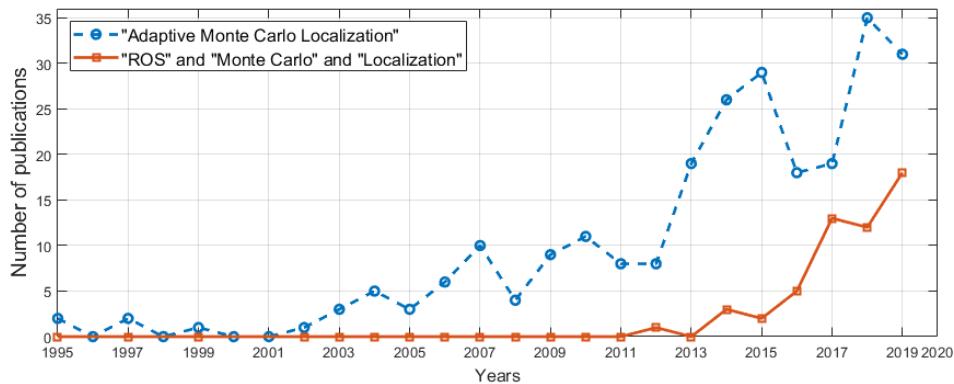
Stahl et al. [23] present a LiDAR-based localization at high speeds. Besides the resampling frequency, the work proposes a novel method for LiDAR single beam angular displacement and introduces rotational noises related to the current car speed. de Miguel et al. [24] proposed an improvement on autonomous vehicle localization enhancing the weights of the AMCL filter particles by adding Kalman filtered Global Navigation Satellite System (GNSS) information.

Concerning the parameters tuning, [25] discuss the influence of the number of AMCL filter particles in navigation accuracy. It shows that the number of particles must be adjusted to the environment since the robot end position accuracy does not increase with the number of particles increase. Still, [26] gives an overview of the ROS navigation stack parameters tuning, including AMCL, but the work performs a brief qualitative analysis on a few parameters.

The works of [27] and [4] propose automatic tuning tools of the AMCL node parameters based on optimization methods. However, the focus is not on studying the effects of the robot localization parameters but on the optimization methods' performance. The related works show that although the increasing use of the AMCL ROS package, no further attention was given to its parameters tuning and its influence study.

## 3 ROS Adaptive Monte Carlos Localization Package

The AMCL ROS package [3] is a localization algorithm based on the algorithm proposed by [28], known as Kullback-Leibler Distance Sampling Monte Carlo Localization. It adapts the number of particles used in pose estimation during its execution, depending on the system states. If the robot position is uncertain, the number of particles is wide, aiming to cover all possible robot states [4]. If the position uncertainty



**Fig. 1** Research evolution concerning the Adaptive Monte Carlo Localization Algorithm application in different domains and its application as a ROS localization package.

is low, the algorithm reduces the number of particles, which increases the algorithm efficiency.

In addition to the robot pose estimation, the AMCL ROS node also returns a covariance matrix related to it. As presented in [5], it is a 6-by-6 matrix constructed over the covariance of each pose variable related to the others. We consider  $\sigma_x^2$ ,  $\sigma_y^2$ , and  $\sigma_\theta^2$  as the metrics to measure the influence of the parameters changing, ignoring the remaining values – as explained in [5].

### 3.1 Parameters Description

Comparing to the results in [5], this paper refers to the allowed range of particles for the filter (`min_particles` and `max_particles`), the number of equally-spaced laser beams used in the likelihood function (`laser_max_beams`), and the minimum translational (`update_min_d`) and rotational (`update_min_a`) distances the robot travels before a filter update parameters. It aims to analyze the previous results with the external measurement of AGV position. Of the 47 parameters of the package, 22 of them are configurable. Together with the previous work, 17 parameters are tested and have their results compared.

The previous work [5] explored seven of them and this research extension examine other 10 parameters. The AGV control node sets initial pose parameters to a predefined value (`initial_pose_x = 0`; `initial_pose_y = 0`; and the yaw angle, `initial_pose_a = π/2`), letting the covariance values of position as `initial_cov_xx = 0.25 m`, `initial_cov_yy = 0.25 m`, and the yaw angle initial covariance value as `initial_cov_aa = 0.06 rad`, as default.

#### 3.1.1 Overall Filter Parameters

The `kld_err` parameter defines a threshold error between the maximum likelihood estimate and the actual true density. The Kullback Liebler Divergence is a measure of the

difference between two probability distributions. So, the parameter determines the maximum divergence accepted between the maximum likelihood estimate and the actual true density. In the experiments, the `kld_z` parameter was also adjusted. It has the form of  $(1-p)$ , where  $p$  is the probability that estimated distribution error is lower than the `kld_err` value [3].

The parameter `recovery_alpha_slow` “is a decay rate for the exponential filter that estimates the long-term average” [29]. The `recovery_alpha_fast` is the decay rate for the short-term average estimation, likewise. The parameters are used in deciding when to recover the pose estimation by adding random poses in the map. Yet [29] states that “if the short-term likelihood is better or equal to the long-term likelihood, no random sample is added”. However, if the first likelihood is worse than the second, then random samples are added.

The `resample_interval` parameter defines the number of filter updates needed before resampling. This algorithm step resamples the particle from the previous particle set but is now weighted by the posterior belief function.

#### 3.1.2 Laser Model Parameters

From the laser model category, the present research explores `laser_z_hit` and `laser_z_rand` parameters, used by the likelihood field laser model, which is the default laser model of the AMCL package. The parameters are weights of a weighted average that considers two types of measures noises: a local measurement noise and the random measurements. Due to several influences, like the limited resolution of the range sensor [29], different surfaces, and atmospheric effect on the signal, a range measurement has an intrinsic error. The `laser_z_hit` parameter weights that noise, which is modeled by a narrow Gaussian with the mean in the “true” range value.

From [29], range finders sensor sometimes provide unexpected random measurements. Such measurement noise is modeled as a uniform distribution covering the entire sensor measurement range. The `laser_z_rand` parameter weight the noise distribution. Since the laser model uses a weighted average of the two noise distributions, `laser_z_hit + laser_z_rand = 1`.

### 3.1.3 Odometry Model parameters

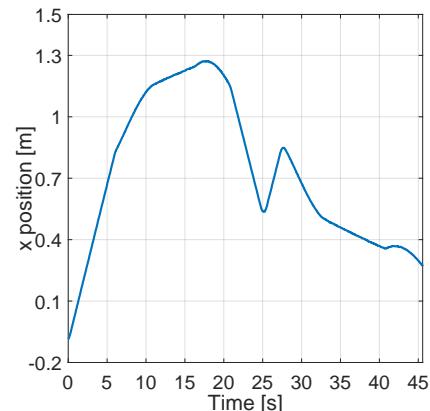
And from the odometry model category, the paper studies the impact of the odometry correction by changing the odometry model (`odom_model_type`), and the expected noise parameters concerning the robot movement `odom_alpha1`, `odom_alpha2`, `odom_alpha3`, and `odom_alpha4`. The `odom_model_type` parameter sets the odometry model regarding the robot model, differential drive, or omnidirectional drive. However, according to [3], a bug was found in both models generating new options for corrected models. Since the AGV odometry model is a differential drive, for now on, the paper only refers to that model regarding the odometry. The default value of the parameter is the uncorrected model, which was used in [5] and the tests of the present work. Using the corrected model, the odometry noise parameters' values differ from the default values of [3].

The noise parameters, `odom_alpha` from 1 to 4, represent the estimate of odometry rotational noise caused by the rotational and translational components of the robot, and the odometry translational noise caused by the translational and rotational components of the robot, respectively. These parameters influence the particles spread regarding the AGV axis.

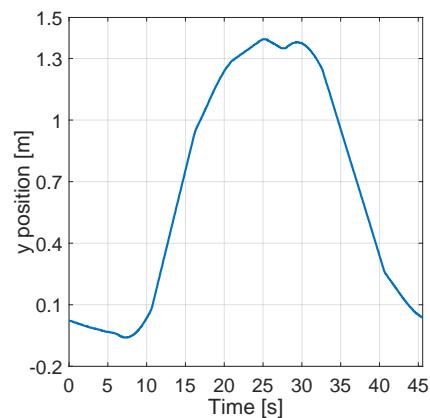
## 4 Materials and Methods

The research uses a differential drive AGV in a controlled and known environment to test the AMCL ROS package parameters' influence over the robot localization. For each experiment, at least three trials were conducted. The first step was mapping the environment using the GMapping ROS package, generating three laser-based maps with different resolutions. To extend the previous work analysis, the subsequent experiments use the lower resolution map. For each parameter, or groups of related parameters, and with the others as default, we conduct the AGV localization trials. To minimize the effect of other ROS nodes interacting with the AMCL node, we programmed the AGV with an open-loop trajectory, controlled by `kether_control` ROS node. Figure 2 shows the AGV trajectory. The path length is approximately 5.3 meters. The AGV takes 45 seconds to cover it.

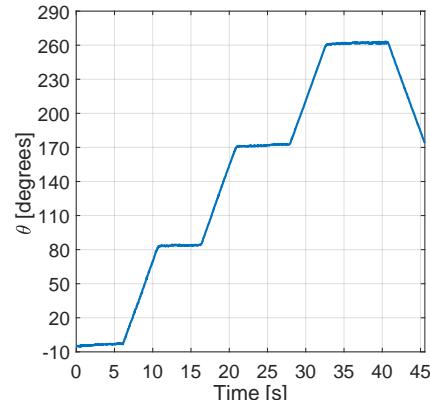
The robot also starts the trials from the same region on the map with a minor variation because of the robot posi-



(a) AGV trajectory along x-axis.



(b) AGV trajectory along y-axis.



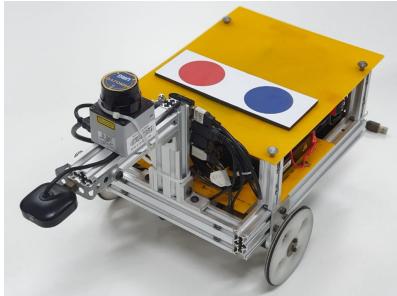
(c) AGV orientation.

**Fig. 2** The AGV trajectory overtime: the plots show how the AGV moves in the x- and y-axis, and its orientation angle gave the programmed open-loop trajectory. The course takes approximately 45 seconds covering 5.3 meters.

tioning. The trajectory control node also publishes an AGV initial pose. As the experiments use the same initial pose value, we assume a variant pose noise.

## 4.1 Hardware and Software Specifications

A differential drive AGV is the testing platform of this paper. It has a Hokuyo URG-04LX laser scanner and a 3-channel digital incremental encoder in each DC motor RE-max 29 226802, the AGV actuator. Two EPOS 24/5 dc motor driver commands the motors. Figure 3 shows the AGV. It is also equipped with a webcam, but it is not used in the experiments.



**Fig. 3** The differential drive AGV is equipped with a Hokuyo laser scanner used in the experiments. Over its platform, one can see the circle-shaped tags used to identify the AGV for external tracking.

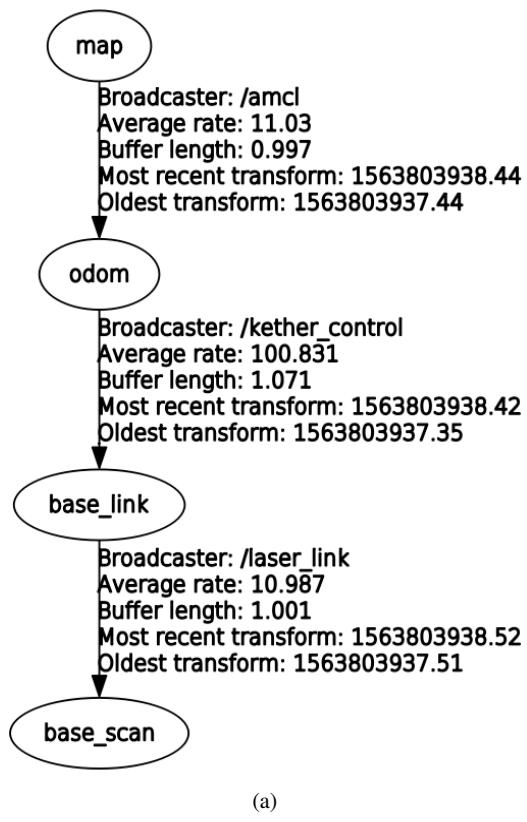
For the experiments, we used a portable computer with an Intel Core i7 7th generation, with 8 GB RAM, and an Nvidia Geforce GTX 1050Ti GPU, running Ubuntu 16.04 LTS with ROS Kinetic Kame. The computer communicates with the AGV by a USB cable, writing the speed commands to the motors drivers and reading the sensors: the laser scanner and the motors encoders.

### 4.1.1 ROS Architecture

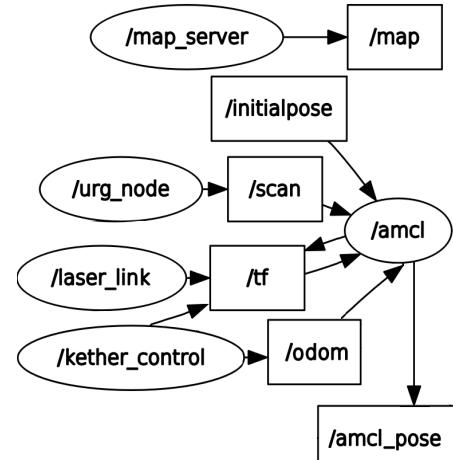
Figure 4(a) shows the transformation tree of the system coordinate frames and 4(b) the ROS nodes and ROS topics. The transformation tree allows the laser scanner measures into the map frame coordinates.

The `/map_server` node, Figure 4(b), reads the map file and publishes it on the `/map` topic. The system uses the `urg_node` to read the laser sensor data and publish the measured distances on `/scan` topic. The `/laser_link` node calculates a frame transformation from the AGV base frame to the sensor frame. Finally, the `/kether_control` node control the AGV motors and read its encoders. The AGV has a visual tag on the same coordinates of the AGV `base_link` frame to identify its movement and track its position by an external measurement system. The `/amcl` node publishes the AGV initial pose to the `/initialpose` topic. It subscribes to the `/scan` topic to read the laser scanner measurements, to the `/tf` topic to read the coordination transformations, and to the `/odom` topic to read the robot odometry used to estimate the AGV pose. A ROS *launch* file was used to run the architecture, defining each trial's parameters' values. Table 1 shows the

central parameters default values and the rationale for their consideration.



(a)



(b)

**Fig. 4** ROS architecture of the experiments. (a) The system coordinate frame transformation tree, from the map to the laser\_scan frame. (b) The system nodes and topics.

1 **Table 1** Parameters default values and rationale for their consideration.  
 2

Overall Filter Parameters		
Parameter	Default Value	Rationale
<i>min_particles</i>	100	Comparing the default values to the best tuning of the previous paper in another AGV setting.
<i>max_particles</i>	5000	Comparing the default values to the best tuning of the previous paper in another AGV setting.
<i>update_min_d</i>	0.2	Comparing the default values to the best tuning of the previous paper in another AGV setting.
<i>update_min_a</i>	0.5	Verify how the variation of the threshold error between the likelihood estimate and the actual true density ( <i>kld_err</i> ) affects the AGV localization.
<i>kld_err</i>	0.01	Verify how the variation of the probability that the error on the estimated distribution will be less than <i>kld_err</i> affects the AGV localization.
<i>kld_z</i>	0.99	Verify how the variation of the probability that the error on the estimated distribution will be less than <i>kld_err</i> affects the AGV localization.
<i>recovery_alpha_slow</i>	0.0	Verify how the inclusion of random poses affects the AGV localization and pose estimation.
<i>recovery_alpha_fast</i>	0.0	Verify how the inclusion of random poses affects the AGV localization and pose estimation.
<i>resample_interval</i>	2	Verify the influence of the resample frequency on the AGV localization and pose estimation.

Laser Model Parameters		
Parameter	Default Value	Rationale
<i>laser_model_type</i>	<i>likelihood_field</i>	The parameter was not modified. The experiments used the same laser model.
<i>laser_max_beams</i>	30	To compare the number of laser beams used and verify its impact on the AGV localization.
<i>laser_z_hit</i>	0.95	Verify how the variation of noise weighting affects the AGV localization,
<i>laser_z_rand</i>	0.05	respecting the restriction <i>laser_z_hit</i> + <i>laser_z_rand</i> = 1.

Odometry Model Parameters		
Parameter	Default Value	Rationale
<i>odom_model_type</i>	<i>diff</i>	To verify the differential odometry corrected model's impact on the parameters tuning and the AGV localization and pose estimation.
<i>odom_alpha1</i>	0.2	To verify how the expected noise in odometry's rotation estimate from the rotational component of the robot's motion affects the AGV localization.
<i>odom_alpha2</i>	0.2	To verify how the expected noise in odometry's rotation estimate from the translational component of the robot's motion affects the AGV localization.
<i>odom_alpha3</i>	0.2	To verify how the expected noise in odometry's translational estimate from the translational component of the robot's motion affects the AGV localization.
<i>odom_alpha4</i>	0.2	To verify how the expected noise in odometry's translational estimate from the rotational component of the robot's motion affects the AGV localization.

## 4.2 Data Presentation

Figure 5 shows the behavior of the covariance values with the default localization node parameters. It is used as a pattern to compare the results of the parameter variation.

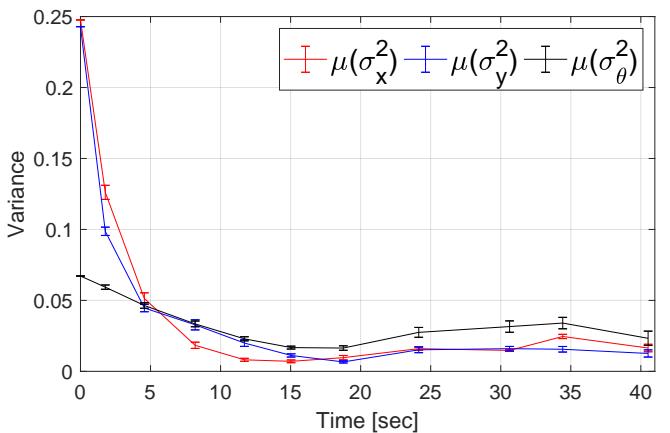
The lines denote the variances mean value  $\mu(\sigma_i^2)$ , with  $i = \{x, y, \theta\}$ . The vertical intervals represent the standard deviation of each covariance.

Likewise, the map plots include the standard deviation of the AMCL pose estimation for  $x$ - and  $y$ -axis, as Figure 6 shows. Also, Figure 6 shows the AGV vision-based external tracking pose estimation for comparison. The standard deviation of the position was suppressed to does not pollute the plot.

Section 5.1 explains the vision-based external measurement system. In the same way that Figure 5 shows the mean value of variances, Figure 6 shows the average pose estimation and average external tracking pose estimation. The same goes for the standard deviation.

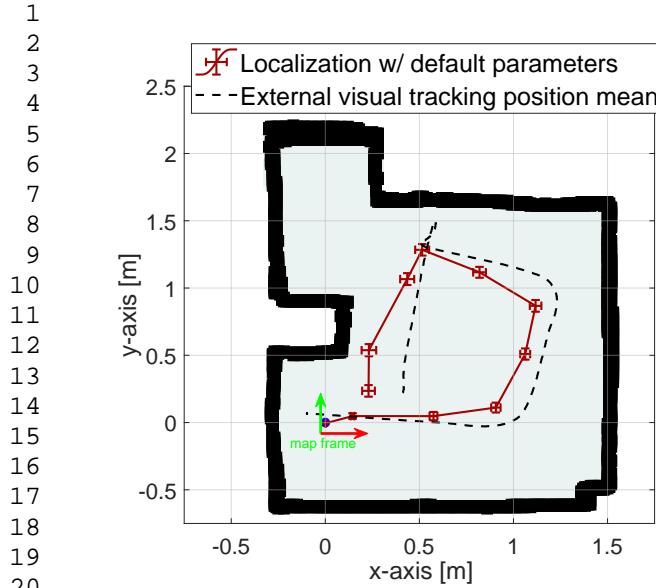
## 5 Experimental Setup

The experiments were conducted in a walled portion of our laboratory, Figure 7(a), with 2.70 by 1.80 meters, approximately, with obstacles to avoid the localization ambiguity. For each trial, we recorded a ROS *bagfile* with the published



**Fig. 5** Covariance matrix data presentation example: the average of the variance of  $x$ ,  $y$ , and  $\theta$  estimated by the AMCL node, with standard deviation, of 10 experiments with default parameters.

topics, also recorded a video from a camera in the ceiling for posterior processing, Figure 7(b), and the localization visualization in *rviz*, Figure 7(c). Each parameter configuration was tested and recorded at least three times, and the following graphs present their mean values. If the standard devi-



**Fig. 6** Map plot example: the average of the positions  $x$ , and  $y$  estimated by the AMCL node, with standard deviation, comparison with the vision-based external measurements.

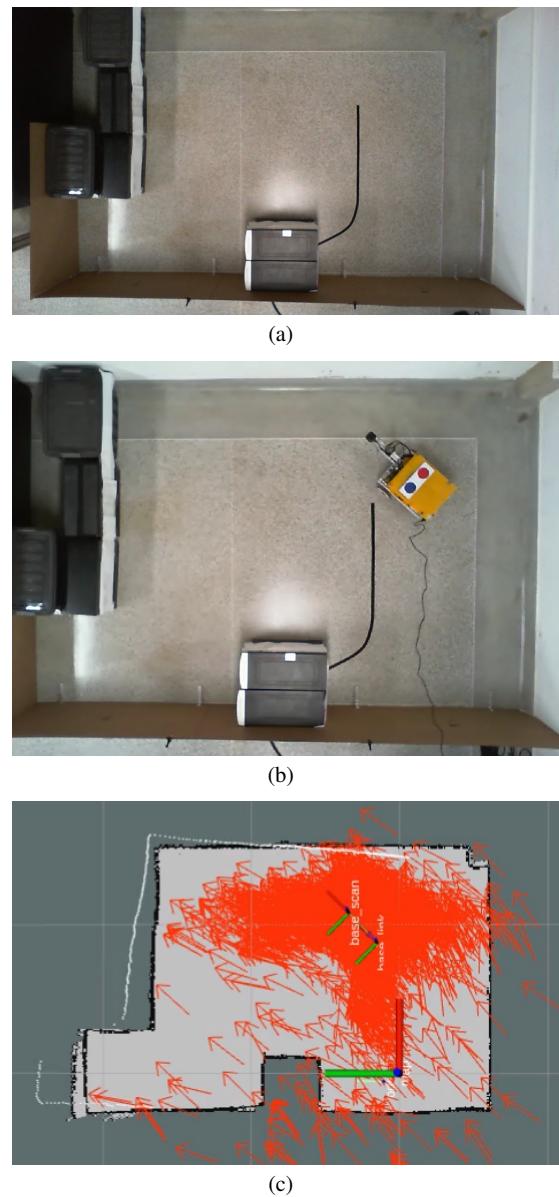
ation is small or pollutes the plot, it is suppressed without loss of information.

### 5.1 External AGV Tracking Using a Vision-based System

The tests were recorded with a Microsoft LifeCam HD-3000 fixed on the ceiling. Figure 8 shows the distance between the camera sensor and the AGV tags plane, and the camera sensor field of view angle, as well as the size of the generated image, in pixels. With this information and using the approach of [30], one can calculate the equivalent distance value, in meters, per image pixel. Thus, the vision-based external tracking system measures the AGV pose in the environment. This work considers the vision-based external pose tracking the most accurate localization measurement to compare the AMCL ROS package pose estimations.

The external tracking system employs python 2.7 code language, and OpenCV 4.0.0, NumPy, among others, libraries to post-processing the video recording. So, it is an offline localization. The algorithm processes a video frame from the AGV environment, measuring the vehicle position and  $\theta$  angle related to the origin point of the image, the superior right corner, as Figure 9 shows. To speed up the image processing and correct the frame perspective, the system user sets four points collinear pairwise. In our case, these are the floor corner points of the walled arena. After setting these points, the algorithm corrects the image perspective and crop it, leaving only the interest area.

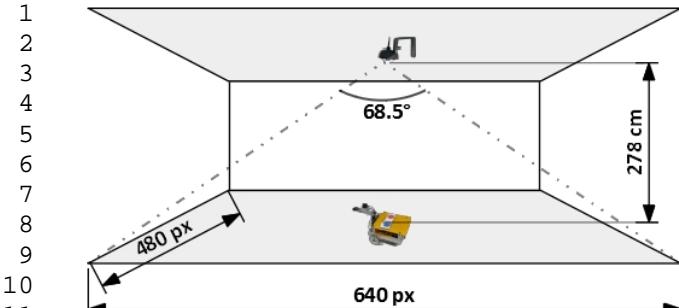
The image processing method uses the HSV color space to identify the tags fixed on the AGV platform. This color system shows advantages when working with images that



**Fig. 7** Experimental setup and data acquisition: (a) Superior view of the experiment scenario without the AGV. (b) Running a localization trial with the AGV and recording the video for posterior AGV position detection. (c) The AGV localization on *rviz*. For better visualization, the images are rotated 90 degrees counter-clockwise from the original.

suffer from the environment lighting variation. The algorithm creates an image mask over the pixels with the chosen color tags to detect it. After the detection, the tags' positions are determined. Note that it is still possible to define to AGV position with a single tag, but not its orientation. That is why the AGV has a blue and a red circle tag.

Finding the center of each circle tag, considering the red circle on the AGV right side when it drives forward, the system measures its  $\theta$  angle. The AGV position is the median point of the line linking the tags centers. The  $\theta$  angle is given by the perpendicular line that crosses the position



**Fig. 8** The AGV external pose tracking system: the setup of the vision-based external pose tracking system, with the webcam fixed ate the ceiling and pointed downwards.



**Fig. 9** The AGV external pose tracking system: AGV tags identification and the outputs of the vision-based external tracking. Compare the cropped area with Figure 7(b).

point. Thus, for each frame, the algorithm returns the AGV position and angle from the origin of the image, as shown in Figure 9.

Observing Figures 5, 6, 7(c) and 9, one can note the Cartesian coordinate system of the laser-based maps diverges from the one of AGV vision-based external tracking system. GMmapping fixes the map's origin at the beginning of the AGV mapping when the odometry is null. So, a linear transformation was necessary to converge the AMCL estimated pose and the external AGV tracking for comparison. This transformation requires a rotation of  $\theta_p$  degrees, a mirroring of the y-axis, a translation to the origin  $o_x$  and  $o_y$ , and a multiplication by a scale factor  $d_p$ .

Given the matrix D, Equation 1, and the rotation matrix R, Equation 2, Equation 3 shows the linear transformations to obtain the pose values on the same Cartesian coordinate system of the map created using GMmapping. The transformed pose coordinates are  $x'$ ,  $y'$ , in meters, and  $\theta'$ , in degrees;  $d_p = 0.0047 \text{ m/pixel}$  is the equivalent distance per image pixel, which was found according to [30];  $o_x$ ,  $o_y$ , and  $o_\theta$  are the original coordinates of the mapping step;  $x_p$

and  $y_p$  are the AGV position in pixels, and  $\theta_p$  is the  $\theta$  angle related to the frame coordinate axis.

$$D = \begin{bmatrix} d_p & 0 & 0 \\ 0 & d_p & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (1)$$

$$R = \begin{bmatrix} \cos(o_\theta) & -\sin(o_\theta) & 0 \\ \sin(o_\theta) & \cos(o_\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2)$$

$$\begin{bmatrix} x' \\ y' \\ \theta' \end{bmatrix} = D \cdot \left\{ \left[ \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \left( R \cdot \begin{bmatrix} x_p \\ y_p \\ \theta_p \end{bmatrix} \right) \right] + \begin{bmatrix} o_x \\ o_y \\ o_\theta \end{bmatrix} \right\} \quad (3)$$

## 6 Results and Discussions

The section presents and discusses the results of the experiments, separated by the parameter category. To extend the analysis of [5], it provides some new experiments with the best previous results. In the following graphs and map plots, we suppressed the standard deviation of the AMCL pose estimation and covariance matrix values when its values were too low, without lacking information. The standard deviation of the external AGV tracking pose average pollutes the plot due to the number of data points, so it is also suppressed. Table 2 presents a summary of the experiments' parameter adjustments — the next sections present only the variation of the correspondent parameter or set of parameters. The remaining parameters maintain the default value.

### 6.1 Overall Filter Parameters

#### 6.1.1 Comparison to the Previous Results

The best results for the minimum and the maximum number of filter particles in [5] were the range of 2000 to 5000 particles. Figure 10(a) shows the results in the positions and  $\theta$  covariance averages. This range did not present significantly better results for the current experiment scenario in terms of convergence and covariance final values. However, when compared to the default values outcome, the experiment result reveals a position estimation closer to the external AGV position tracking estimation, as shown in Figure 10(b).

In [5], the best results for the minimum translational and rotational movement before a filter update was 0.05 and 0.125 meters, respectively. Observing Figure 11, when compared to the default values outcome, it shows a quick convergence with lower average values and lower standard deviation values compared to the default values outcome. Since the lower values of distance increase the filter update frequency, the pose estimation in Figure 12 shows a smooth trajectory, related to the default estimation. When contrasting to the external AGV tracking, Figure 12, the pose estimation shows a systematic position error until the moment the AGV maneuver, in the left upper corner of the map, but a

1 **Table 2** Summary of default parameter values and the adjusted parameters used in the experiments.  
 2

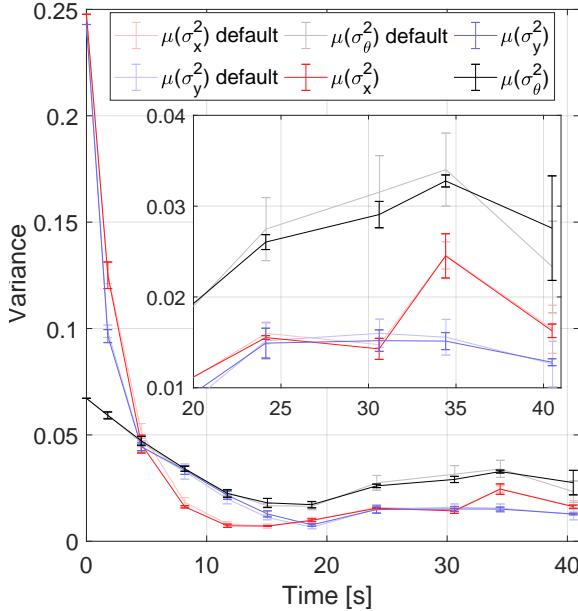
Overall Filter Parameters									
Parameter	Default Value	Parameter tuning values used in experiments							
<code>min_particles</code>	100	2000							
<code>max_particles</code>	5000	5000							
<code>update_min_d</code>	0.2	0.05							
<code>update_min_a</code>	0.5	0.125							
<code>kld_err</code>	0.01	0.001	0.1	0.5	0.001	0.1	0.5	0.001	0.1
<code>kld_z</code>	0.99	0.99	0.99	0.99	0.80	0.80	0.80	0.60	0.60
<code>recovery_alpha_slow</code>	0.0	0.001	0.001	0.01	0.01	0.1	0.1		
<code>recovery_alpha_fast</code>	0.0	0.001	0.01	0.001	0.01	0.001	0.01		
<code>resample_interval</code>	2	1	4	8					

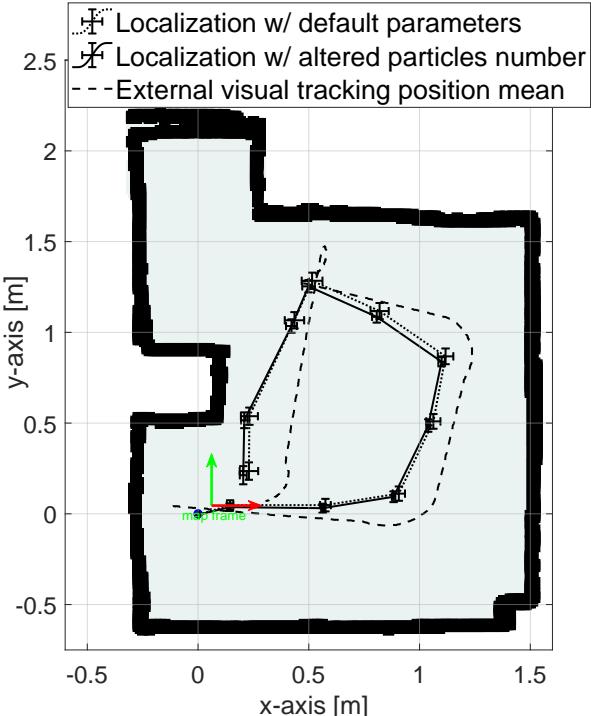
Laser Model Parameters									
Parameter	Default Value	Parameter tuning values used in experiments							
<code>laser_model_type</code>	<code>likelihood_field</code>	Not modified.							
<code>laser_max_beams</code>	30	512							
<code>laser_z_hit</code>	0.95	0.99	0.90	0.80	0.70				
<code>laser_z_rand</code>	0.05	0.01	0.10	0.20	0.30				

Odometry Model Parameters									
Parameter	Default Value	Parameter tuning values used in experiments							
<code>odom_model_type</code>	<code>diff</code>	<code>diff-corrected</code>	<code>diff-corrected</code>	<code>diff-corrected</code>	<code>diff-corrected</code>	<code>diff-corrected</code>	<code>diff-corrected</code>	<code>diff-corrected</code>	<code>diff-corrected</code>
<code>odom_alpha1</code>	0.2	0.005	0.05	0.05	0.05	0.05	0.05	0.05	0.5
<code>odom_alpha2</code>	0.2	0.005	0.005	0.005	0.005	0.005	0.005	0.005	0.5
<code>odom_alpha3</code>	0.2	0.010	0.005	0.005	0.005	0.005	0.005	0.005	0.5
<code>odom_alpha4</code>	0.2	0.005	0.005	0.005	0.005	0.005	0.005	0.005	0.5



49 (a) Results of the covariance matrix of the pose estimation compared with the default parameters one.  
 50

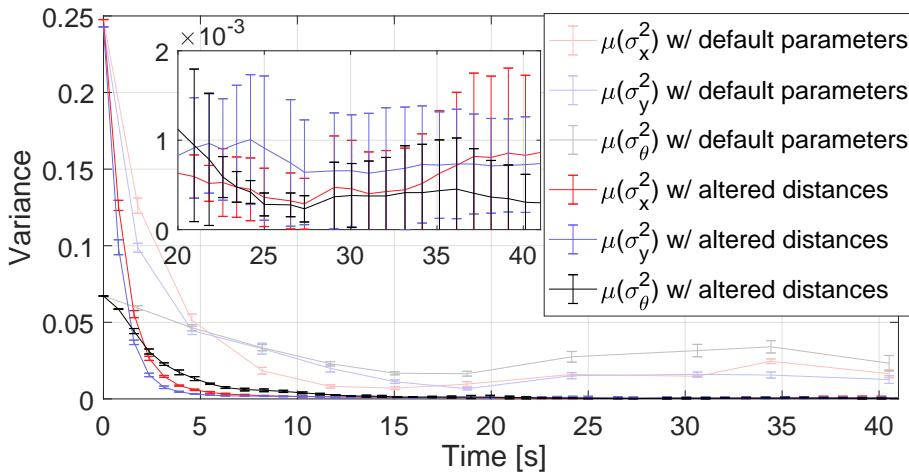


51 (b) AMCL pose estimation compared with the default parameters estimation and external tracking.  
 52

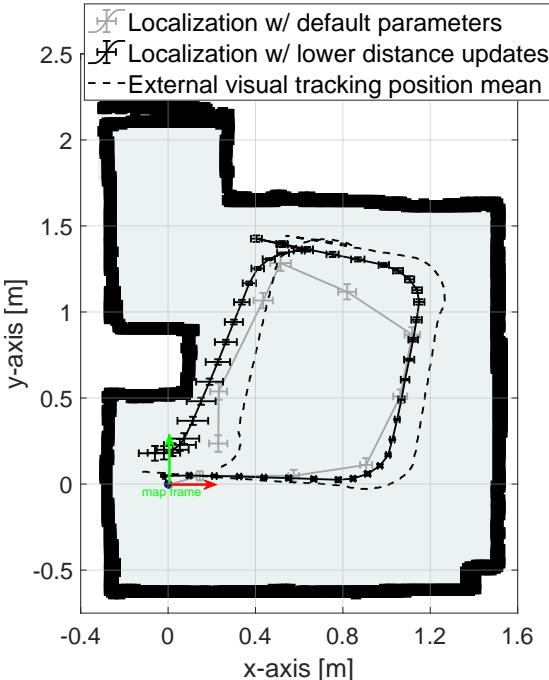
53 **Fig. 10** Minimum and maximum number of filter particles parameters, using the best results from [5].  
 54

55 close estimation to the AGV localization. After the maneuver,  
 56 the difference between the default values estimation, the  
 57 external pose tracking, and the experiment pose estimation  
 58 increases due to an accumulated error in pose estimation.  
 59

60 The odometry correction parameters tuning might help to  
 61 reduce this kind of error.  
 62



**Fig. 11** Covariance matrix results of minimum translational and rotation movements before filter update parameters, using the best results from [5].

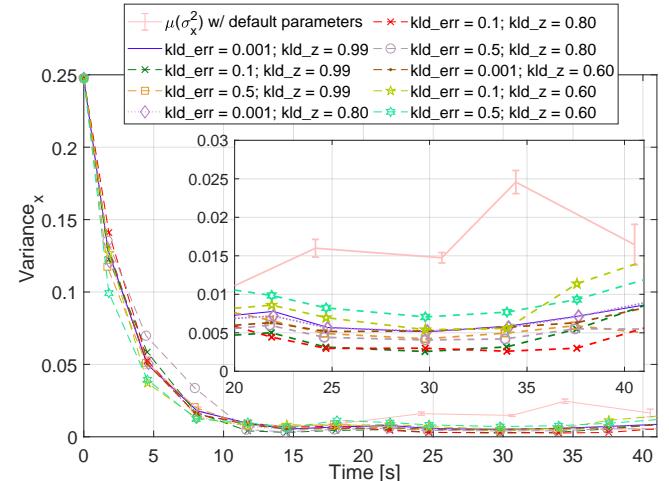


**Fig. 12** AMCL pose estimation results of minimum translational and rotation movements before a filter update parameters, compared with the default parameters result and the external pose tracking, using the best results from [5].

#### 6.1.2 Kullback Liebler Divergence Parameters

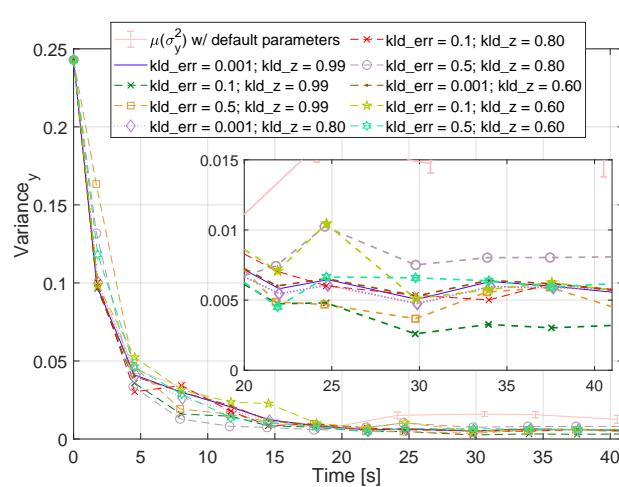
The previous investigations show that despite higher values of  $kld\_err$ , parameter led to an oscillation behavior along the AGV path, their covariance curves converge to lower values. Also, it exposes the lower values of the same parameter, 0.001 and 0.005, have similar behavior as the default, but always above the mean. In this case, the  $kld\_z$  parameter has the default value of 0.99. The current experiment outcomes reestablish the above affirmation. Nevertheless, with

the varying  $kld\_z$  parameter, new observations arise. With the lower value of  $kld\_err$ , the  $kld\_z$  variation has a minor influence, as one can see at the detailed graph of Figure 13, for instance.

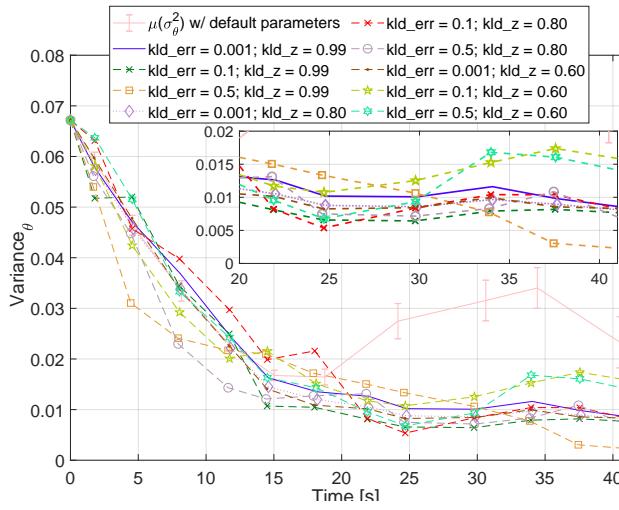


**Fig. 13** The  $x$  position estimation covariance results under Kullback Liebler parameters variation.

Despite the  $kld\_z$  variation, the variance means converge to similar values. As we increase the probability of the error on the estimated distribution is lower than  $kld\_err$  by decreasing  $kld\_z$ , it outcomes lower covariance values. For the  $kld\_err = 0.1$ , lower  $kld\_z$  values show better pose covariance values. And the same applies to  $kld\_err = 0.5$  outcomes. The lower the number of  $kld\_z$  is, the lower the required number of samples for each sample bin of the particle filter. Thus, the particles resample is less spread and more concentrated around the mean. Figures 14 and 15 show the results for the  $y$  position and the orientation angle  $\theta$  covariance, respectively.



**Fig. 14** The  $y$  position estimation covariance results under Kullback Liebler parameters variation.



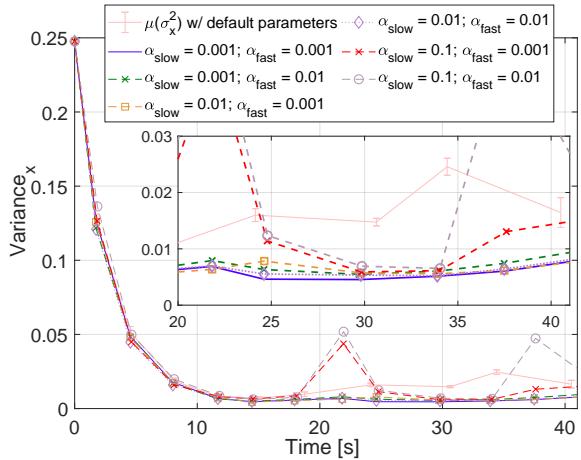
**Fig. 15** Results of  $\theta$  covariance under Kullback Liebler parameters variation.

However, despite the difference in covariances convergence, the pose estimation with different parameter values does not significantly differ, although the results are better than the default setting.

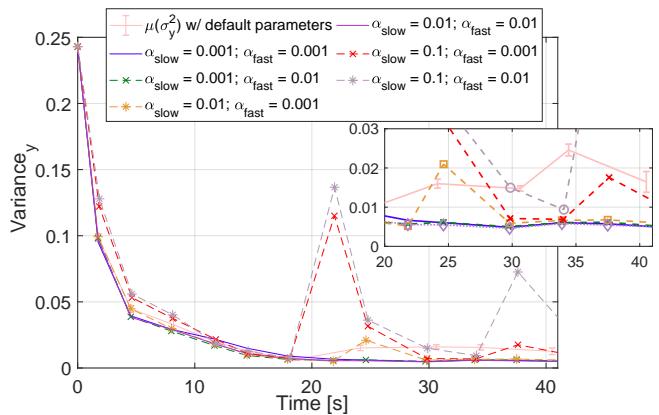
#### 6.1.3 Filter Recovery Parameters

To prevent the localization algorithm from failing by getting stuck on a wrong global localization – or recovering after a robot kidnapping – it incorporates two recovery parameters [29]. These parameters are exponential decay rates for the average weight filter used to decide when to recover by adding particles in random poses. “The probability of adding a random sample considers the divergence between the short- and the long-term average of the measurement likelihood” [29]. As default, both `recovery_alpha_slow` ( $\alpha_{slow}$ ) and `recovery_alpha_fast` ( $\alpha_{fast}$ ) are disabled.

The higher the  $\alpha_{slow}$  and  $\alpha_{fast}$  values, the more likely the algorithm is to include random particles. The alpha parameters multiply a divergence of the weighted average of the measurement model and the short- and long-term average of the measurement likelihood,  $w_{fast}$  and  $w_{slow}$ , respectively, resulting in an update in the last two averages values. The rate of the short-term and long-term determines the probability of adding particles in random poses, in the form of  $\max(0.0, 1 - w_{fast}/w_{slow})$  [29]. Figures from 16 to 18 show the covariance matrix results of the  $\alpha$  parameters variation.

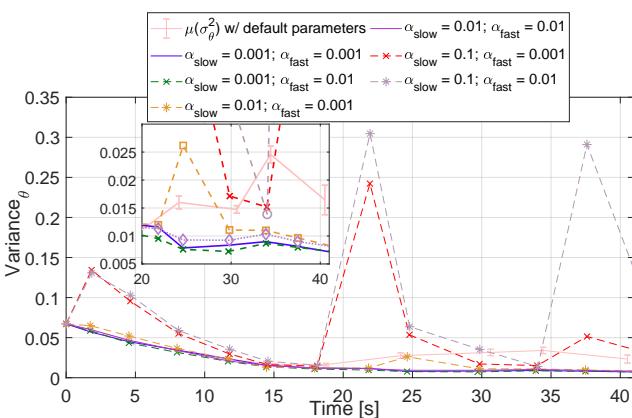


**Fig. 16** The  $x$  position estimation covariance results under filter recovery parameters variation.



**Fig. 17** The  $y$  position estimation covariance results under filter recovery parameters variation.

Lower values of both parameters show better results of convergence and covariance values. With  $\alpha_{slow} = 0.001$  and  $\alpha_{fast} = 0.01$ , the results also confirm the algorithm need of  $0 \leq \alpha_{slow} \ll \alpha_{fast}$ , when both parameters are enable. Figures 19, 20, and 21 compare the externally mea-



**Fig. 18** Results of  $\theta$  covariance under filter recovery parameters variation.

sured AGV pose with the estimated one. The first one compares the AGV angle estimates. According to the relative  $\theta$  error, calculated by the difference between the AMCL pose estimation and the externally measured angle, one can conclude greater values of the alpha parameters result in better angle estimates when the AGV is turning – when the angle graph is an increasing or decreasing slope. When the AGV draws a straight-line movement, with minimal angle variation, the parameter settings show a similar estimation.

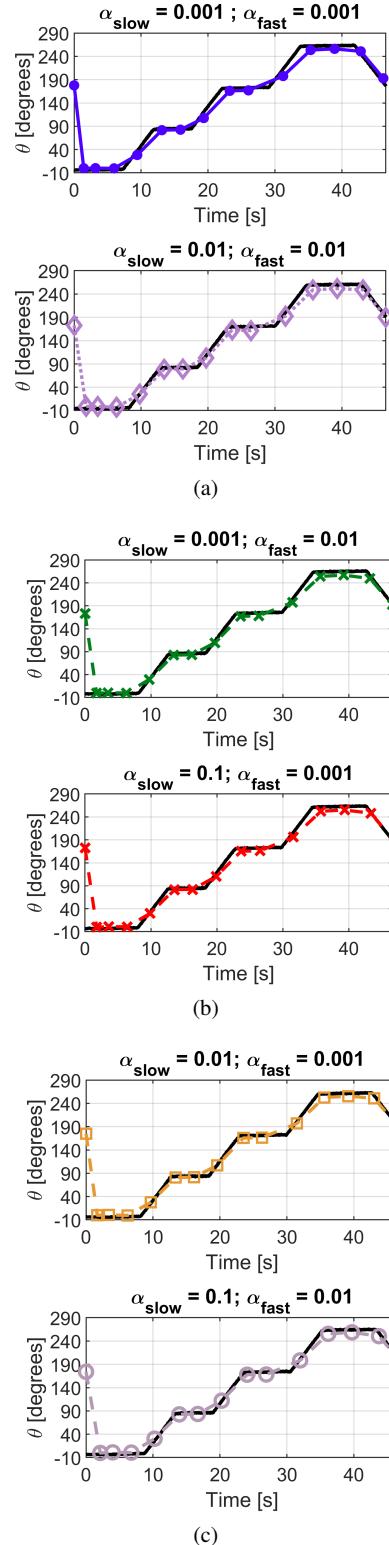
Figure 21 shows the map plots that compare the default parameter values positions x and y average and the standard deviation of each axis, the AGV external position tracking, and the AMCL position estimation.

Although the plots show similar results, in visual inspection, it is possible to observe the influence of the increasing number of particles in random poses in trials with greater alpha parameters values. It degrades the pose estimation as Figure 21(e) and (f) show, for instance, compared to Figure 21(a), (b), (c), and (d). The estimated localization in the last two maps, (e) and (f), present a greater error related to the external measurement. Observe the three estimated points at the top of the map to see their difference.

#### 6.1.4 Resample Interval Parameter

Changing the resample interval parameter implies changing the number of needed filter updates before resampling and revising the estimated pose outcome. Figures from 22 to 24 display the behavior of the covariance matrix value to the resample interval change. Although the tested values lead the covariance values to lower values if compared to the default parameters, the higher is the resampling frequency, the quicker the covariances converge.

For `resample_interval = 1`, for every filter update, the algorithm also resamples the particles. It generates the maximum number of pose estimates. Observe the blue solid line in Figures 22, 23 and 24 compared to the dashed orange



**Fig. 19** Comparison between the externally detected  $\theta$  using the vision-based system, the straight black line, and the AMCL estimated  $\theta$ , for each parameter setting.

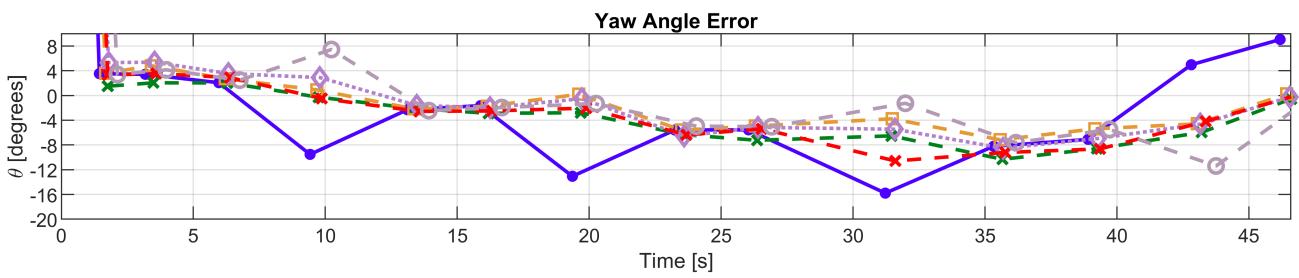


Fig. 20 The relative error, for each parameter setting, ignoring the first estimate, which is from a previous pose estimate.

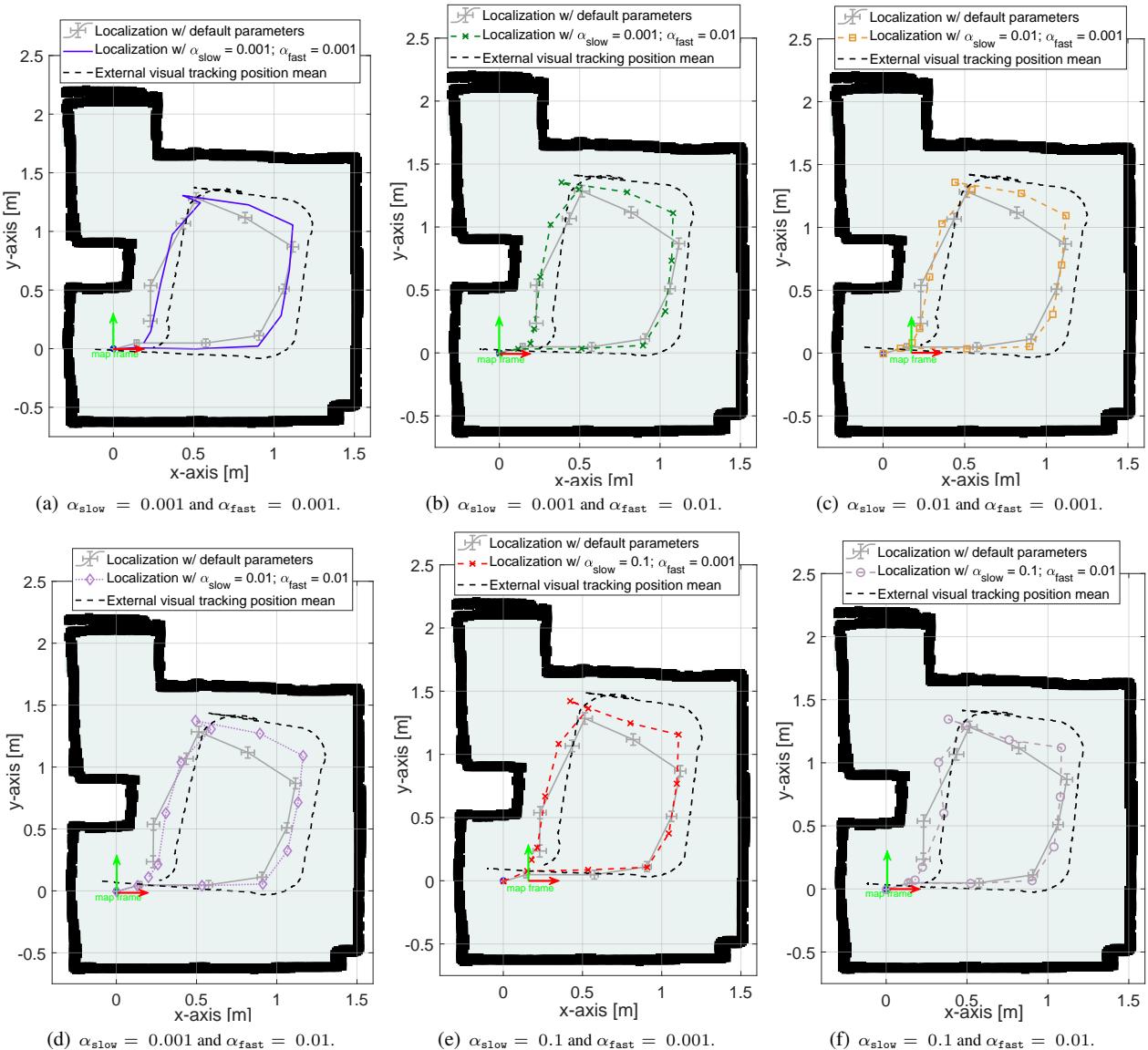
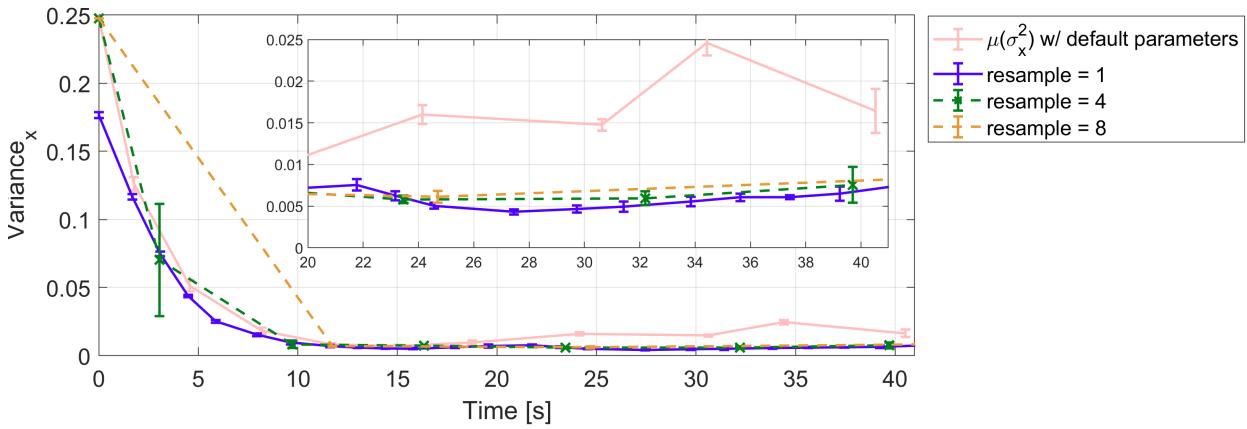


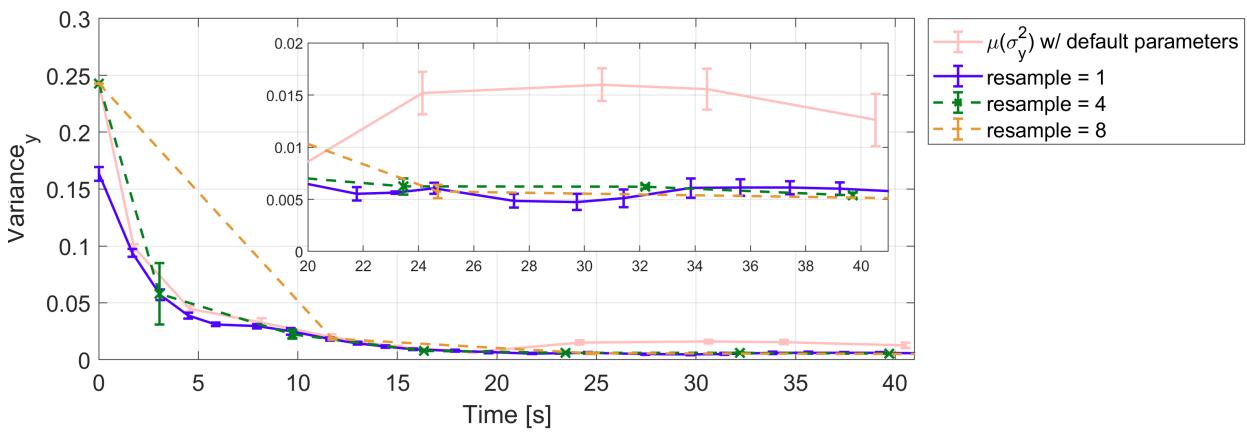
Fig. 21 Results of estimate localization under  $\alpha_{slow}$  and  $\alpha_{fast}$  parameters variation.

one. Note the filter update also depends on other parameters, such as minimum translation or rotational movement. For a high resample frequency, i.e., `resample_interval = 1`, the covariance values converge to lower values.

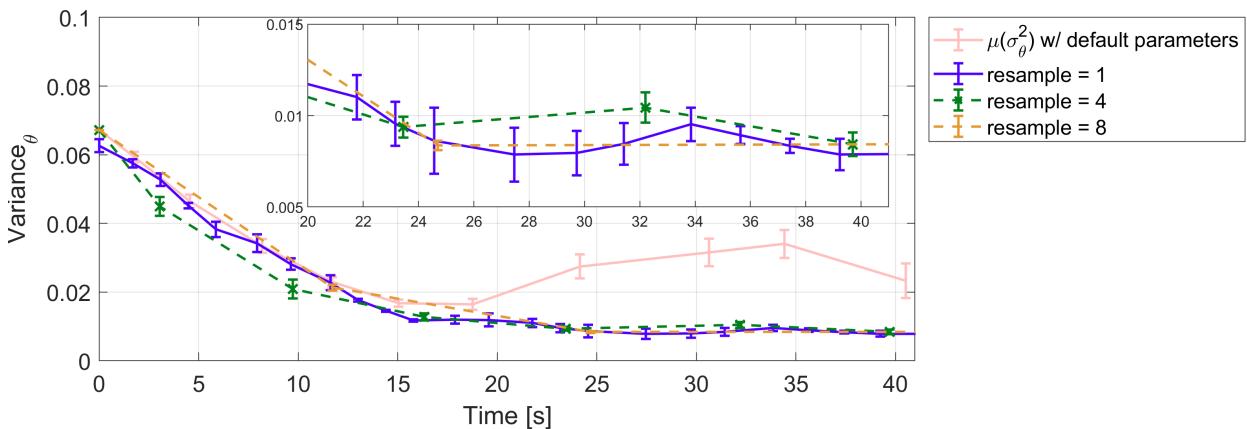
Observing Figure 25(a), the result shows a systematic error of the localization estimation compared to the external measurement. Such error is equivalent to the error found with lower values of minimum translational and rotational



**Fig. 22** The  $x$  position estimation covariance results under resample interval parameter variation.



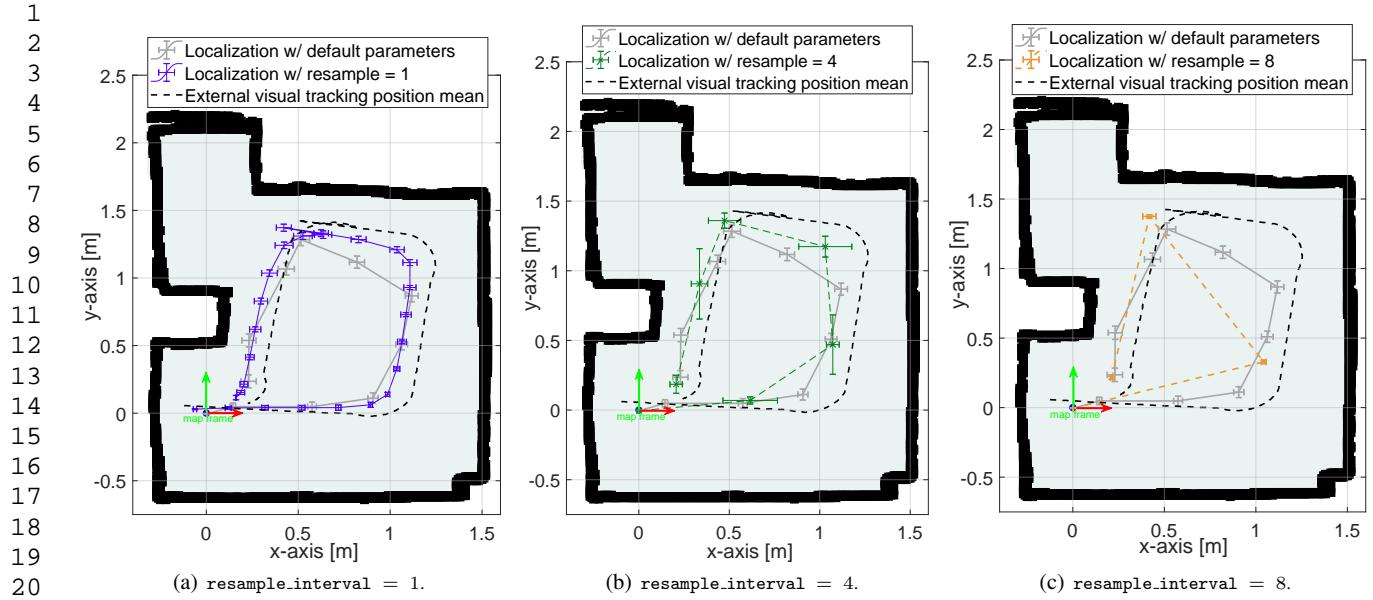
**Fig. 23** The  $y$  position estimation covariance results under resample interval parameter variation.



**Fig. 24** The results of  $\theta$  covariance under resample interval parameter variation.

movement before a filter update parameters in Figure 12. With the resample interval parameter equals to four, the AGV position uncertainty increases. See Figure 25(b). The same behavior does not occur to the resample interval equals eight. However, with higher resample values, the estimated position values do not recover the path traveled nor show a good approximation, as Figure 25(c) shows.

The results show a need for a trade-off between the resample interval (and other parameters that increase the filter update rate) and the filter processing computational cost depending on the AGV application. Further, the resample parameter affects the resolution of the estimated trajectory using the localization estimation points. If the AGV works in small spaces or travels short distances, using a high value of resampling interval may lead to information loss.



**Fig. 25** Results of estimated position under `resample_interval` parameter variation.

## 6.2 Laser Model Parameters

### 6.2.1 Extended Comparison

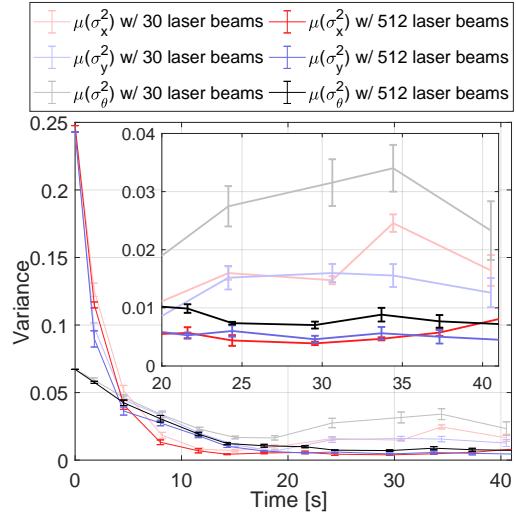
In [5], the best result for the number of laser beams parameter was 512 – the maximum value of the AGV laser scanner – beams. The parameter defines how many equally spaced beams of laser scanner the measurement model of the AMCL algorithm uses. In calculating the samples weight step of the AMCL ROS package, the algorithm determines the range of each laser beam to fit it into the knowing map, testing if it is a random measurement, or an unexpected obstacle, or a failure in a sensor measurement. The experiment results reaffirm the previous ones, as Figure 26 show the increase of the number of beams reduces the covariance value in the AGV position regarding the default values.

Comparing the results with the AGV external position tracking, using a higher number of laser beams improves the position estimates, as Figure 27 shows.

Figure 27 shows the comparison among the AMCL pose estimation with default parameters, the pose estimation using 512 laser beams of the sensor, and the result of the AGV external pose tracking system. Figure 27(a) shows the localization improvements compared to the default parameter localization, resulting in an estimation closer to the external measurements. Also, the  $\theta$  estimation, shown in Figure 27(b), has a lower error related to the external pose tracking system measurement if contrasted to the default estimation.

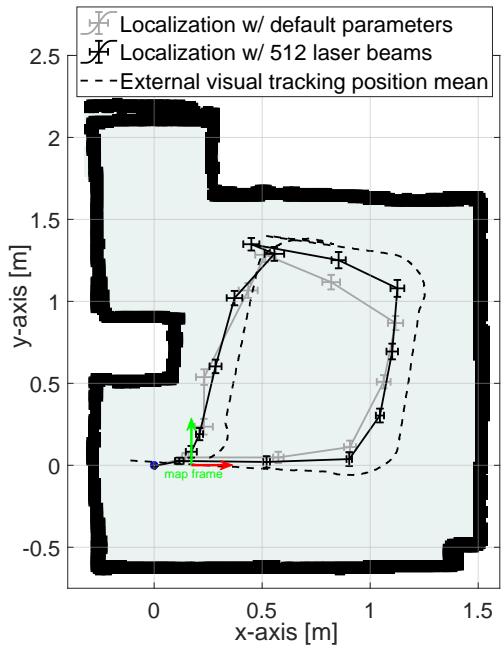
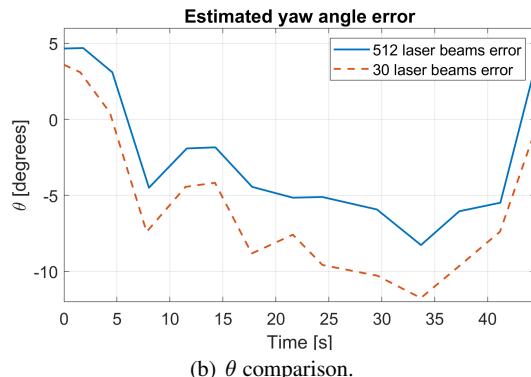
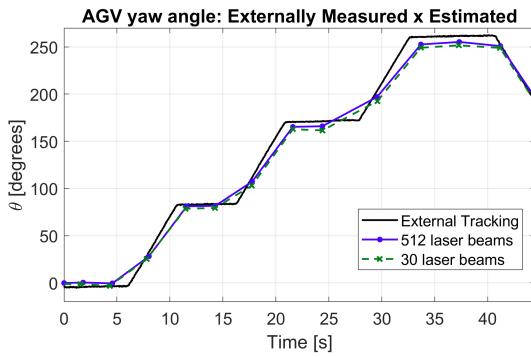
### 6.2.2 Laser z\_hit and z\_rand parameters

The `laser_z_hit` and `laser_z_rand` parameters adjustment represent the weight of each noise has in the final sen-



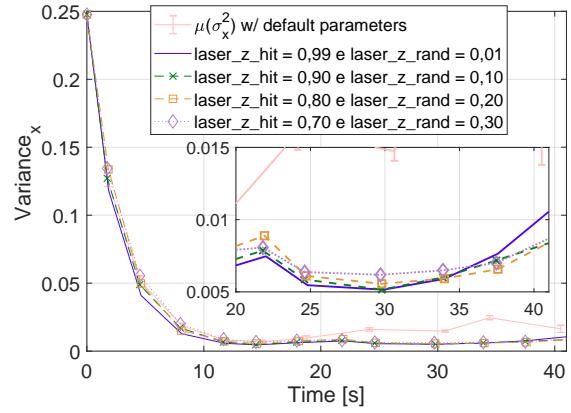
**Fig. 26** The pose estimation covariance results of the number of used laser beams parameter, using the best results from [5].

sor measures set since its sum must be equal to 1. So, if `laser_z_hit` is greater than `laser_z_rand`, the weight of the sensor measurement noise is greater than the weight of the random measurement noises. When tuning the parameters, considering the surface's material and characteristics that affect the laser beams measurements is important, as cited by [20]. In the experiments, a gradual increase in the weight of the random measurements and, consequently, a decrease in the sensor measurements noise weight is shown. The parameters default values are `laser_z_hit` = 0.95 and `laser_z_rand` = 0.05, meaning the algorithm considers a lower contribution of random measures noise. As a result,

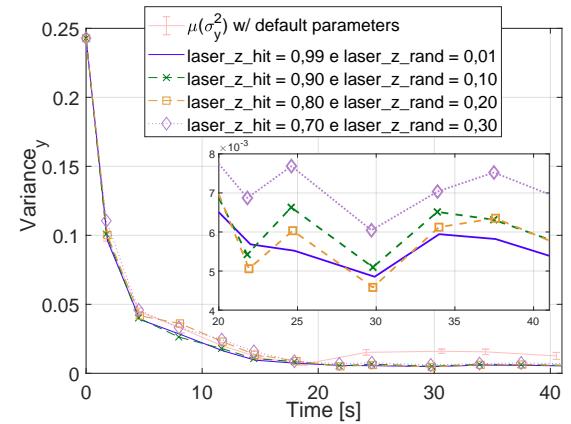
(a) Estimated  $x$  and  $y$  positions comparison.(b)  $\theta$  comparison.

**Fig. 27** Results comparison for 512 laser beams. (a) Position estimation results of the number of used laser beams parameter, using the best results from [5], compared with the default parameters result and the external tracking. (b) Comparison among the externally measured AGV  $\theta$  angle, the angle estimation results using the best results from [5], and the default parameters result.

Figures 28 to 30, as `laser_z_rand` increases the covariance of the pose estimation becomes more susceptible to oscillations, e.g., as occurs in the  $y$  position and  $\theta$  estimates, see Figures 29 and 30. It occurs because the random measurement noise parameter contributes more to the weighted average, contributing to the estimated localization oscillation. However, the increase of the `laser_z_rand` parameter may be beneficial depending on the surface's material of the environment [20], which reflects the laser beams. So it must be considered in the tuning process.

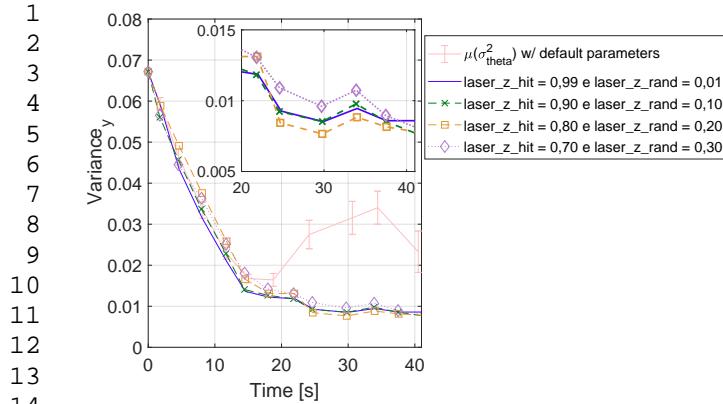


**Fig. 28** The  $x$  position estimation covariance results under `laser_z` parameters variation.



**Fig. 29** The  $y$  position estimation covariance results under `laser_z` parameters variation.

One can observe the covariance results do not follow a pattern, indicating the decrease of the laser hit parameter does not mean the proportional increase or decrease of covariance values. It shows the dependence of the environment regarding those parameters tuning. For instance, `laser_z_hit` = 0.80 and `laser_z_rand` = 0.20 lead to lower theta covariance value and similar values for the position estimation.



**Fig. 30** The  $\theta$  estimation covariance results under `laser_z` parameters variation.

However, despite the difference in covariances convergence, the pose estimation with different parameter values does not significantly differ, although the results are better than the default setting.

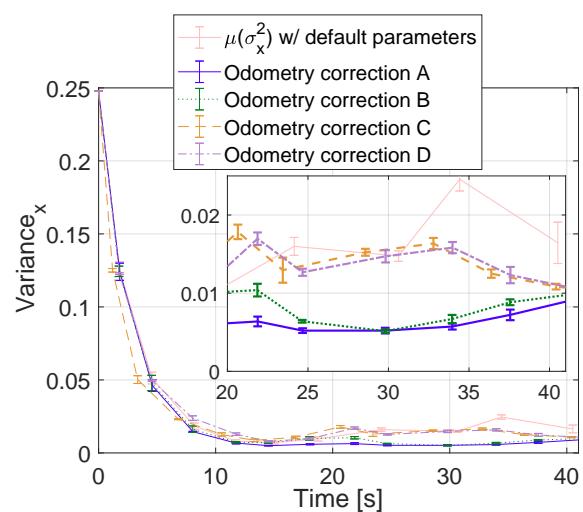
### 6.3 Odometry Model Parameters: Odometry Model Correction

As cited in Section 3.1.3, the `odom_model_type` has a corrected version. Since several robots and other applications were configured and tuned to the `diff` model, a new parameter value was included, `diff-corrected` [3]. Table 3 shows the odometry parameters values used in this work trials.

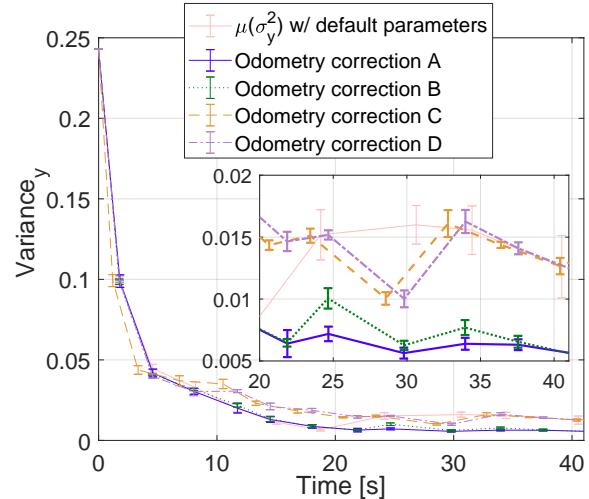
**Table 3** Parameters values for the Odometry Model experiments.

Parameter	Odometry Correction A	Odometry Correction B	Odometry Correction C	Odometry Correction D
<code>odom_model_type</code>	<code>diff-corrected</code>	<code>diff-corrected</code>	<code>diff-corrected</code>	<code>diff-corrected</code>
<code>odom_alpha1</code>	0.005	0.05	0.05	0.5
<code>odom_alpha2</code>	0.005	0.005	0.05	0.5
<code>odom_alpha3</code>	0.010	0.005	0.05	0.5
<code>odom_alpha4</code>	0.005	0.005	0.05	0.5

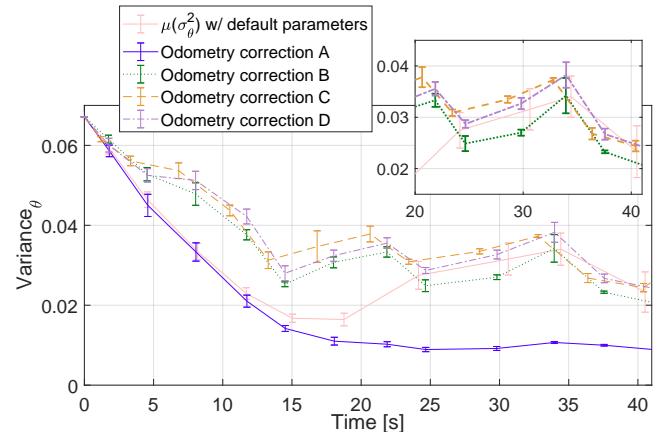
The Odometry Correction A values are the default ones for the corrected model. Comparing its results with the default diff model results, the covariances of both positions and the angle decrease significantly, as shown in Figures 31, 32, and 33. From Odometry Correction B to D, the `odom_alpha1` to `odom_alpha4` parameters (odometry noise values) were increased. The positions and angle covariance values follow the parameters increasing, almost in the same ascending order in these situations. It also affects each covariance outcome's standard deviation value and influences the most  $\theta$  covariance results, Figure 33. Figure 34 shows the map plots of the corrected odometry.



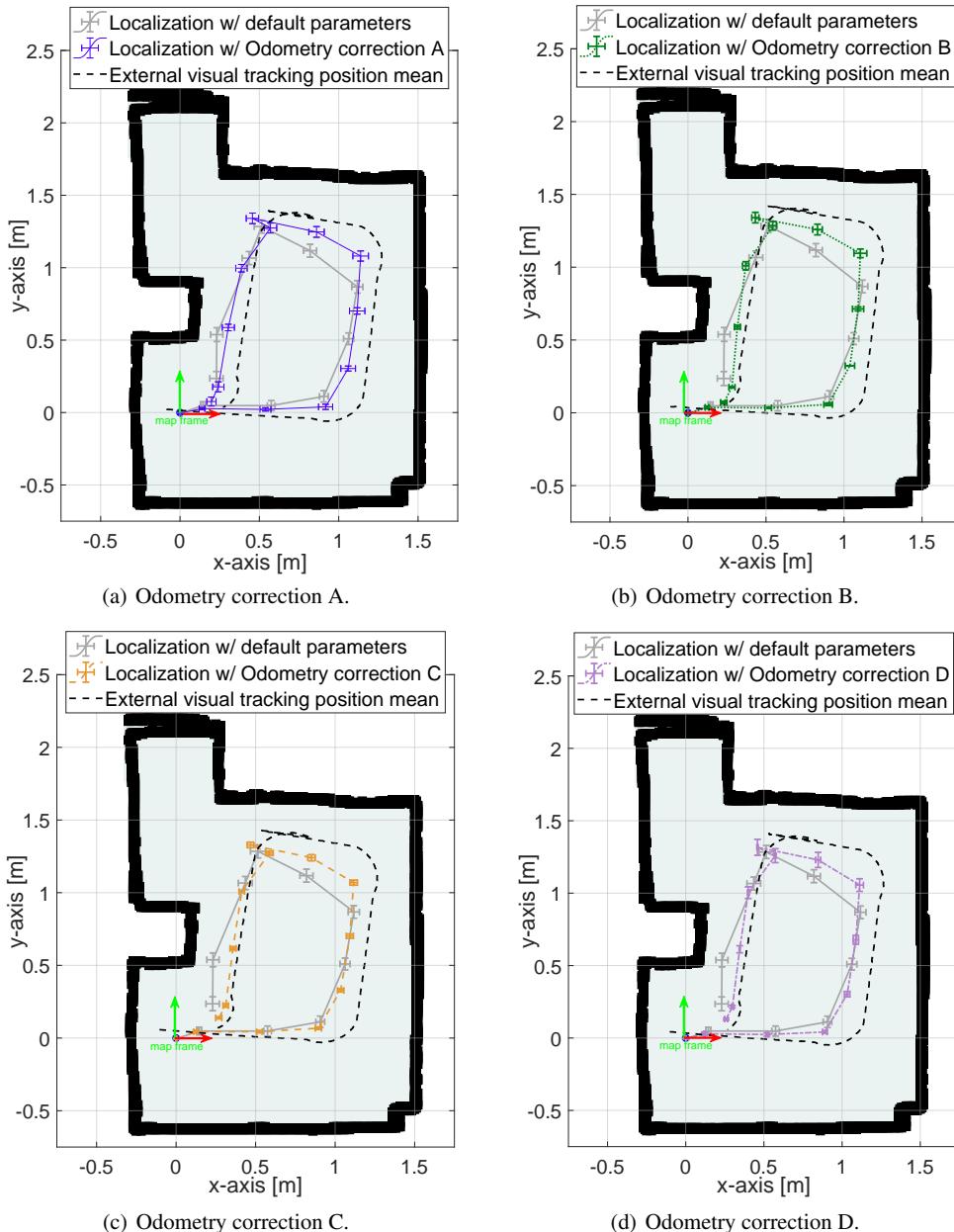
**Fig. 31** The  $x$  position estimation covariance results under odometry noise parameters variation.



**Fig. 32** The  $y$  position estimation covariance results under odometry noise parameters variation.



**Fig. 33** The results of  $\theta$  covariance under odometry noise parameters variation.



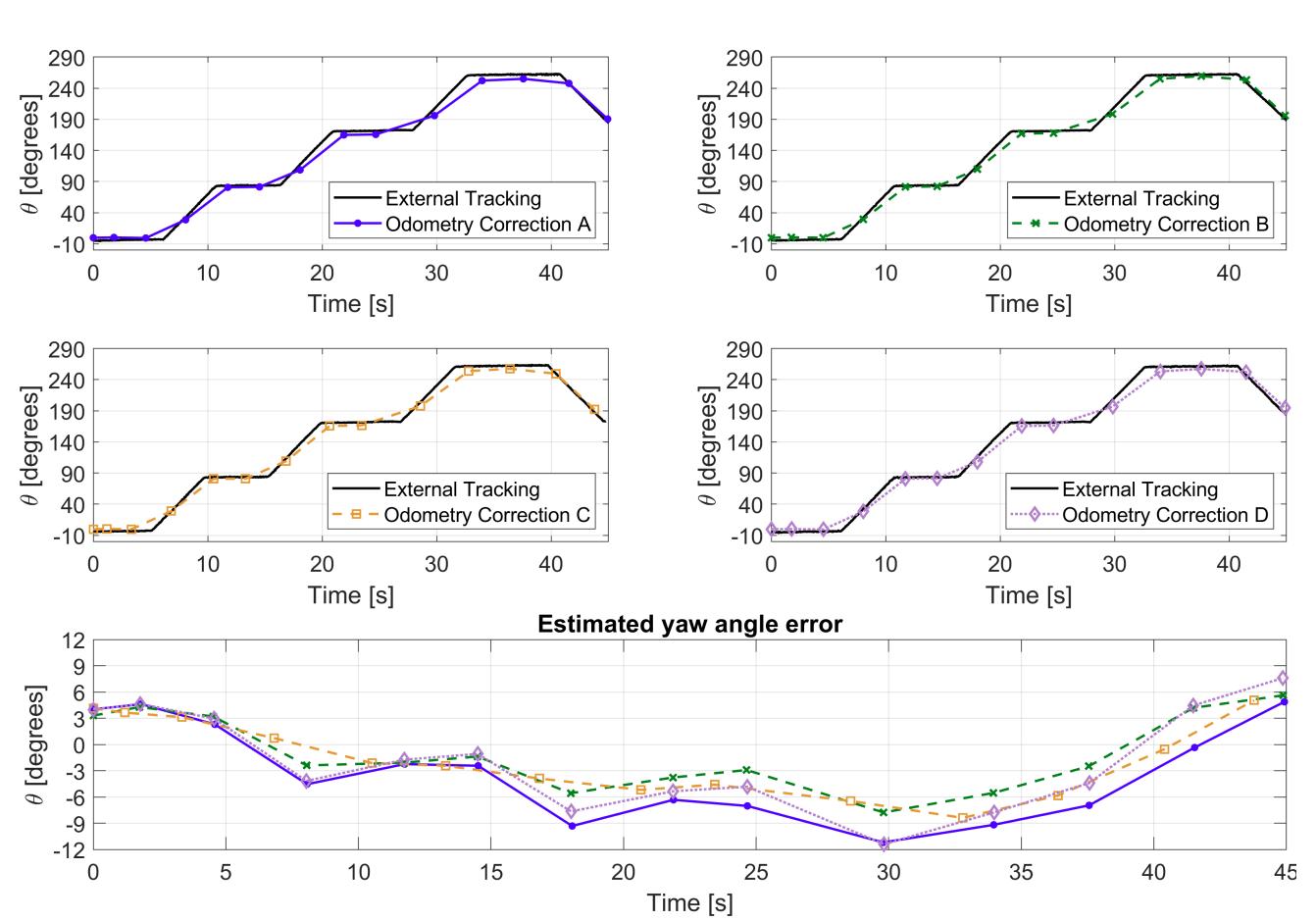
**Fig. 34** The position results of the odometry model and noise parameters adjustments, and comparison with default values result and the external tracking system position.

Analyzing the map plots in Figure 34, one can observe that the odometry corrections positively affect the AMCL pose estimation compared to the default estimation and the external tracking system measurements. From the other experiments' outcomes, the last part of the AGV path – the last left turn, straight line, and a right turn – accumulates an error on the pose estimation repeatedly. The corrected model improved the localization estimate, given similar results to the external tracking outcomes. Also, observe the increase of odometry correction parameters decreased the uncertainty of pose estimation. Figures 34(b), 34(c), and 34(d) present

smaller uncertainty crosses on estimated localization points compared to Figure 34(a). Moreover, it improves the  $\theta$  estimates, as shown in Figure 35, decreasing the error regarding the default parameter values. Note that the default parameters are the corrected default parameters.

## 7 Conclusion

This research extends the analysis of the distinct influence of AMCL ROS package parameters in AGV localization. Besides extending the previous work discussions, its con-



**Fig. 35** Comparison between the externally detected  $\theta$  using the vision-based system and the AMCL estimated  $\theta$ , for each parameter setting. The bottom graph shows the relative error for each parameter setting.

tributions present a vision-based external AGV pose tracking and compare the laser-based estimated localization with a visual-based system. We performed the experiments in a controlled environment, and the AGV ran the same path to enable comparison against the parameter variation. In the publications altogether, seventeen package parameters were tested and discussed. Even so, the work does not intend to establish a tuning method or a better set of parameter values, but to demonstrate its influence, showing each parameter behavior. Also, as seen in the results' presentation, using an external measurement system can help parameter tuning. It is unnecessary to adjust all the seventeen parameters to start using the AMCL ROS package, but the paper increases the source to understand what to tune if need.

To investigate the impact of the parameters in computational resource use is a future work, aiming to define the impact the localization algorithm parameters may have in a lower level positioning control loop of the AGV. Another future work may also discuss the impact of the map resolution, the sensor model, and the geometrical constraints of

the AGV and the environment in the AGV localization and pose estimation.

**Acknowledgements** The authors thank the support of Capes, all the colleagues of TEAR Laboratory, and Dr. Abdeldjallil Naceri. The authors wish to extend special gratitude to Matheus Ungaretti Borges for kindly present the previous paper in ICAR 2019.

### Ethical Approval

Not applicable.

### Consent to Participate

Not applicable.

### Consent to Publish

Not applicable.

## 1 Authors Contributions

2 Conceptualization: Wallace Pereira Neves dos Reis, Kelen  
 3 Cristiane Teixeira Vivaldini; Methodology: Wallace Pereira  
 4 Neves dos Reis, Kelen Cristiane Teixeira Vivaldini, and Orides  
 5 Morandin Junior; Formal analysis and investigation: Wal-  
 6 lace Pereira Neves dos Reis; Software and Visualization:  
 7 Wallace Pereira Neves dos Reis and Guilherme José da Silva;  
 8 Writing - original draft preparation: Wallace Pereira Neves  
 9 dos Reis; Writing - review and editing: Orides Morandin  
 10 Junior; Resources: Kelen Cristiane Teixeira Vivaldini and  
 11 Orides Morandin Junior; Supervision: Kelen Cristiane Teix-  
 12 eira Vivaldini and Orides Morandin Junior.  
 13

## 14 Funding

15 Partial financial support was received from Coordenação de  
 16 Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES)  
 17 - Finance Code 001.

## 18 Competing Interests

19 The authors have no relevant financial or non-financial in-  
 20 terests to disclose.

## 21 Availability of Data and Materials

22 All the data and material are available upon request to the  
 23 corresponding author.

## 24 References

1. R. Guzman, R. Navarro, M. Beneto, D. Carbonell, in *Robot Operating System (ROS)* (Springer, 2016), pp. 253–288
2. R. Guzmán, R. Navarro, M. Cantero, J. Ariño, in *Robot Operating System (ROS)* (Springer, 2017), pp. 419–447
3. A. Howard, B. Gerkey. Adaptive Monte-Carlo Localization (AMCL) package. Robot Operating System (ROS) (2002). URL <http://wiki.ros.org/amcl>
4. J.P.S. Azevedo, Automatic parameter tuning of algorithms using optimization. Master's thesis, Instituto Superior Técnico Lisboa - Lisbon, Portugal (2017)
5. W.P.N. dos Reis, O. Morandin, K.C.T. Vivaldini, in *2019 19th International Conference on Advanced Robotics (ICAR)* (IEEE, 2019), pp. 302–307
6. P. Astolfi, A. Gabrielli, L. Bascetta, M. Matteucci, IFAC-PapersOnLine **51**(11), 704 (2018)
7. A. Kudriashov, T. Buratowski, M. Giergel, Mechanisms and Machine Science **73**, 2819 (2019)
8. S. Xu, R. Chen, Y. Yu, G. Guo, L. Huang, IEEE Access **7**, 95140 (2019)
9. S. Xu, W. Chou, H. Dong, Sensors **19**(2), 249 (2019)
10. G. Li, J. Meng, Y. Xie, X. Zhang, Y. Huang, L. Jiang, C. Liu, Sensors **19**(15) (2019)
11. B. Zhang, J. Liu, H. Chen, in *2013 IEEE International Conference on Information and Automation (ICIA)* (IEEE, 2013), pp. 822–827
12. Y. Zhang, D. Chen, H. Lin, L. Zhao, in *2018 13th World Congress on Intelligent Control and Automation (WCICA)* (IEEE, 2018), pp. 727–732
13. K.T. Song, Y.H. Chiu, L.R. Kang, S.H. Song, C.A. Yang, P.C. Lu, S.Q. Ou, in *2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC)* (IEEE, 2018), pp. 1833–1838
14. M.Á. Muñoz-Bañón, I. del Pino, F.A. Candelas, F. Torres, Applied Sciences **9**(10), 1997 (2019)
15. P. Anderson, A. Shrivastava, J. Truong, A. Majumdar, D. Parikh, D. Batra, S. Lee, arXiv preprint arXiv:2011.03807 (2020)
16. N. Charabaruk, S. Nokleby, Transactions of the Canadian Society for Mechanical Engineering **40**(2), 169 (2016)
17. Z. Li, Y. Xiong, L. Zhou, in *2017 10th International Symposium on Computational Intelligence and Design (ISCID)*, vol. 2 (IEEE, 2017), vol. 2, pp. 132–135
18. U. Maniscalco, I. Infantino, A. Manfre, in *2017 IEEE International Symposium on Robotics and Intelligent Sensors (IRIS)* (IEEE, 2017), pp. 19–24
19. S.P. Thale, M.M. Prabhu, P.V. Thakur, P. Kadam, in *ITM Web of Conferences*, vol. 32 (EDP Sciences, 2020), vol. 32, p. 01011
20. L.O. Russo, S. Rosa, M. Maggiore, B. Bona, Sensors **16**(8), 1255 (2016)
21. A.W. Alhashimi, R. Hostettler, T. Gustafsson, in *2014 11th International Conference on Informatics in Control, Automation and Robotics (ICINCO)*, vol. 2 (IEEE, 2014), vol. 2, pp. 498–505
22. P. Javierre, B.P. Alvarado, P. de la Puente, in *2019 Third IEEE International Conference on Robotic Computing (IRC)* (IEEE, 2019), pp. 246–249
23. T. Stahl, A. Wischnewski, J. Betz, M. Lienkamp, in *E3S Web of Conferences*, vol. 95 (EDP Sciences, 2019), vol. 95, p. 04002
24. M.Á. de Miguel, F. García, J.M. Armingol, Sensors **20**(11), 3145 (2020)
25. I. Wasisto, N. Istiqomah, I.K.N. Trisnawan, A.N. Jati, in *2019 International Conference on Computer, Control, Informatics and its Applications (IC3INA)* (IEEE, 2019), pp. 187–192
26. K. Zheng. ROS navigation tuning guide. [http://kaiyuzheng.me/documents/papers/ros\\_navguide.pdf](http://kaiyuzheng.me/documents/papers/ros_navguide.pdf) (2016). Last access: 2020-03-23
27. O. Lima, R. Ventura, in *2017 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC)* (IEEE, 2017), pp. 43–48
28. S. Thrun, D. Fox, W. Burgard, F. Dellaert, Artificial intelligence **128**(1-2), 99 (2001)
29. S. Thrun, W. Burgard, D. Fox, *Probabilistic robotics* (MIT press, 2005)
30. D. Puppim de Oliveira, W. Pereira Neves dos Reis, O. Morandin Junior, Sensors **19**(19), 4111 (2019)