1 Configuration of Amcl parameters

The results so far (see update document of 12_11) did show deviations in the pose, drifting by some value as soon as the end of a straight line has been reached. This is probably due to errors in the motion model, which needs to be compensated by tuning proper parameters. **Amcl** does infact strongly rely on **odometry** and **sample motion model**, hence if errors in the odometry occurs (the error in the odometry is always presence due to the noise), this will just accumulate.
The **Kinematic motion** model in use will be **odometry motion model**; there is also a velocity motion model, not in use by AMCL. The latter tend to be less accurate (for all details see book Probabilistic Robotics p. 96).

Table 1: AMCL parameters

alpha1	Specifies the expected noise in odometry's rotation estimate from the rotational component of the robot's motion, has an effect on Figure 1.b
alpha2	same as alpha1, but for the translational component
alpha3	Specifies the expected noise in odometry's translation estimate from the rotational component of the robot's motion, Figure 1.c
alpha4	same as alpha3, but for the translational component

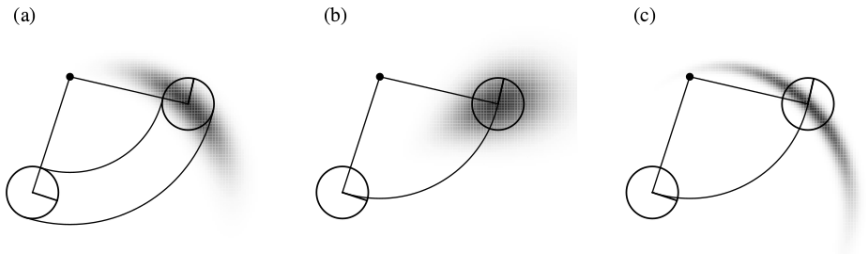

Figure 1: Velocity motion models for different noise parameters settings. Figure .b shows large translational error and small angular error, Figure .c large angular errors and small translational error
.

1.1 AMCL with GMapping map

Due to a small deviation (1/2 cm over 37 meters) between the $odometry/filtered$ and the markers, we can assume the EKF to be our ground thruth. We also did some wrong considerations for tuning, please see **Table5** for the new tuning.

See this link for an answer: https://answers.ros.org/question/227811/tuning-amcls-diff-corrected-and-omni-corrected-odom-models/

Table 2: alfa parameters and results

Parameters	**Results**
alfa4=0.05	Just changing this alone did not produce considerable results at all
alfa4=0.05, alfa3=0.05	Also this combinations does not have an effect on the bearing error. The alfas here have been changed to a factor of 20.
alfa3,4=default=2, alfa1,2=0.05	Also no result in here
all alfas to 0.1	Also did not see nothing relevant
alfa1=0.005, alfa2=0.005, alfa3=0.010, alfa4=0.005	No satisfactory results
alfas=0.0001	works now, I loose global localization property with such a small values, if calling the global localization service.

Table 3: Starting from the corrected alfas = 0,0001, higher progressively alfa3, alfa4 till the optimal

Parameters	**Results**
alfa3,4=0,001 (try a factor of 10 higher)	Still get considerably good results
alfa3,4=0,01 (try a factor of 10 higher)	With those value the error is not anymore bounded to the 5 centimeters, hence we keep as a good result the value of before with alfas3,4=0,001.
minparticle = 6000, maxparticle= 9000	The tuned ones

For the update of min_ and min_d see the folder update_min_a_d under bagfiles/amcl_localization.

Table 4: Update min a/d

Parameters	**Results**
Keep min_d 0.2 and min_a = 0.25 as default values	Plot the difference in changing their values.

Table 5: Final tuning Amcl

Parameters	**Results**
min_particles= 10 max_particles= 9000	min is the end number of particles after filter update, max is the spread number at the beginning
min=10, max=9000, alfas=0.2 (default)	Here I am expecting the large error value, infact. **This works**
Curious to see how the error changes with an max= 12000	Interesting, with too high particles unable to get convergence, but multimodal distribution. This can be explained by: http://roboticsknowledgebase.com/wiki/state-estimation/adaptive-monte-carlo-localization/. Infact, as the robot moves forward we apply the odom to all the samples (particles), a weight needs to be adressed to all particle, and more resampling steps are needed for convergence. This cause a delay in the filter convergence.
maxparticle 10000	Converging wrongly
max particle= 9000, alfas.1	No difference with previous case
mapart= 9000, alfa1,2,3=0.1, alfa4=0.005 (error is translation due to rotation)	No results
maxpart = 9000, alfa1,2=0.1 alfa3,4=0.005	Converges correctly but no final result
max part = 9000 alfa1,2 = 0.005, alfa3,4=0.1	Results in wrong convergence after global localization, lost global loc
max part = 9000, alfa1,2 = 0.1, alfa3,4 = 0.0005	Still 20 cm error
max part = 9000, alfa1,2 = 0.1, alfa3,4 = 0.0001	Still 20 cm error
alfa1,2,3 = 0.1 alfa 4= 0.0001	If this does not work I keep alfas 3,4 low = 0.0005 and go back with alfa 1,2 till I loose global loc. This does not work
alfa1,2 = 0.08, alfa3,4 = 0.0005 (get alfas 1,2 to a lower value progressively)	No result in lowering progressively alfas 1,2
alfa 1,2 = 0.1 and alfa3,4 = 0.00005	No change