Carry the experiment with default values of parameters as:

position of robot_base with respect to World frame is:

(-0.270409; -0.885734; 0.09)

~map_update_interval (float, default: 5.0) biba_setup: 5.0

- How long (in seconds) between updates to the map. Lowering this number updates the occupancy grid more often, at the expense of greater computational load.

~maxUrange (float, default: 80.0) biba_setup: 30

- The maximum usable range of the laser. A beam is cropped to this value.

  Comment: as ~maxUrange = 0: no detection occur
  as ~maxUrange = 80: detect the map very fast

~sigma (float, default: 0.05)

- The sigma used by the greedy endpoint matching

  Comment: as ~sigma get close to value 0.0. Robot appears to lose its way. It's also noticeable that the map_frame moving relative with the world_frame

~kernelSize (int, default: 1)

- The kernel in which to look for a correspondence

  comment:

~lstep (float, default: 0.05)

- The optimization step in translation
- lstep = 0.0~0.2

~astep (float, default: 0.05)

- The optimization step in rotation
- astep = 0.02~1.2

~iterations (int, default: 5)

- The number of iterations of the scanmatcher
- iterations too low (<3) induces error
- iterations too large induces too slow response and map error
- iterations from 4~20 work fine.

~lsigma (float, default: 0.075)

- The sigma of a beam used for likelihood computation
- lsigma = 0 causes error
- lsigma could be much larger than the default value, even with 2.0, and the result is still fine.

~ogain (float, default: 3.0)

- Gain to be used while evaluating the likelihood, for smoothing the resampling effects
- not allow to be 0. With much higher value (30), the result is still the same.

`~lskip` (`int`) , default = 0

- Number of beams to skip in each scan.
- Should be set to be 0. Increasing this value induces lose of information. For large lskip, for example 100, the map is not constructed correctly.

`~srr` (`float`, default: 0.1)

- Odometry error in translation as a function of translation (rho/rho)
- srr from 0~1. For srr large, =10, the robot base and world frames jump unexpectably.

`~srt` (`float`, default: 0.2)

- Odometry error in translation as a function of rotation (rho/theta)
- srr from 0~1. For srr large, =10, the robot base and world frames jump unexpectably.

`~str` (`float`, default: 0.1)

- Odometry error in rotation as a function of translation (theta/rho)
- srr from 0~1. For srr large, =10, the robot base and world frames jump unexpectably.

`~stt` (`float`, default: 0.2)

- Odometry error in rotation as a function of rotation (theta/theta)
- should be from 0.0~1.0

`~linearUpdate` (`float`, default: 1.0)

- Process a scan each time the robot translates this far
- from 0~0.5 induces some error in the map. The value could be large without inducing error, even with linearUpdate = 1000
- frequent filter updates and limit the search area of the scan-matcher, the distribution is considered to have only a single mode when sampling data points to computer the Gaussian proposal.

`~angularUpdate` (`float`, default: 0.5)

- Process a scan each time the robot rotates this far
- from 0~0.5 induces some error in the map. The value could be large without inducing error, even with angularUpdate = 1000
- frequent filter updates and limit the search area of the scan-matcher, the distribution is considered to have only a single mode when sampling data points to computer the Gaussian proposal.

`~temporalUpdate` (`float`, default: 3.0)

- Process a scan if the last scan proccessed is older than the update time in seconds. A value less than zero will turn time based updates off.
- Robot is sensitive to temporalUpdate < 0.5. For 0.1~0.5, robot loses its way after first loop. For

0~0.1, Robot lose its way at the very beginning of first loop. For negative value, updates is turned off.

`~resampleThreshold` (`float`, default: 0.5)

- The neff based resampling threshold (?)
- not affect the map much.

`~particles` (`int`, default: 5)

- Number of particles in the filter
- small number of particles (<5) induces error and too large number of particle (>100) requires much computations and induces low response.