Scan Matching 2D Toolbox

The Scan Matching 2D Toolbox is a collection of 2D ScanMatching functions for range finder sensors. It contains a set of 12 algorithms plus a simple random world generator.
It lets the user test algorithms with different parameters and different world and it is important to understand how ScanMatching and SLAM works. The user will be able to work and modify an open source MATLAB code for its own purposes. This document explains how the toolbox can be executed and defines the internal structure so that the user can modify the code.

Folder Tree

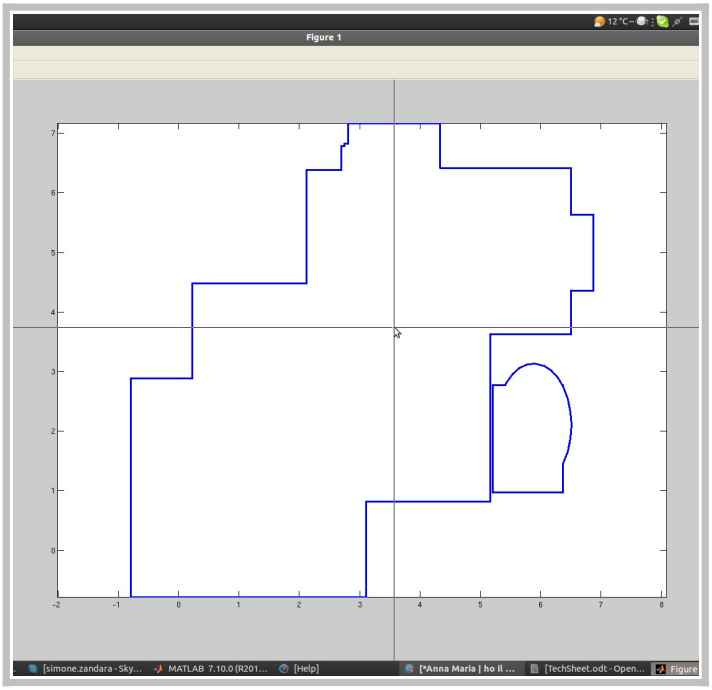
- Bin/ executing, testing and generating worlds
- EKF/ ekf related functions
- Error/ error estimation
- Etc/ configuration files
- Graphics/ plotting functions
- Init/ simulator initialization structures
- Kinematics/ odometry simulation
- Map/ mapping functions
- Math/ math libraries
- ScanMatching/ the 12 scan matching algorithms
- Sensors/ sensors simulation

Simple Execution

To execute the file, simply include the root directory in your MATLAB path, then, type (or run)

```
slam
```

At this point a random map will be generated and displayed on the screen and the user will have to draw the robot trajectory with the mouse then press Enter to start the computation. The default parameters will generate a random structured world and will correct the trajectory using an EKF based SLAM and ICP.



Configuration

`input_traj = 1`	`Specify whether the user is going to select the robot trajectory or using a default one.`
`world = wall_generator(0.1,1);`	The world generation changing the object wall_generator. The first parameter controls how many ellipses will define the world. More ellipses will generate a less shaped world. If the first parameter is 0, then, the world will be unstructured, ie. It will simulates outdoor natural enviroments (caves, rocks). The second parametrs control whether the user has to insert the trajectory. Do not change, use the previous variable to work modify this.
`DEBUG`	This struct contains an entry for each algorithm and control whether to plot the algorithm as it is functioning.
Opt.map	Controls the mapping function for the algorithms. Used by some of them (please refer to the paper). The parameter Opt.map.resolution controls the resolution of the working space for some of the algorithm and must be changed according to the optimal parameters (listed below). Vary this parameter to test how the algorithms behave.
Opt.plot	Controls the plotting. Leave unchanged except Opt.plot.robot_scale: increase if the robot figure is not visible or too big.
Opt.filter	Opt.filter.usefilter says whether to use a stochastic filter or not. Opt.filter.type tells which filter. (only ekf2d availabe).
Opt.scanmatcher	Controls most of the scanmatching parameters.
- usematcher	Do scan matching?
- motionamount	Travel X distance before doing another scanmatching (m)
- distancethreshold	unused
- handle	Tells which algorithm to use for scan matching. Upon changing this, remember to change the

	parameters as in the list below.
- associate	Specify which association method to use. Only if 'handle' is set to 'icpbSM'.
- register	Specify which registration method to use. Only if 'handle' is set to 'icpbSM'.
- rejection rule	Only value @x84. Uses x84 rjection rule during association. Only for ICP based methods
- projfilter	Use projection filter to remove points not visible. Available for all
- iterations	Max number of itarations. Only for iterative algorithms
- Br	Scan Matching Parameters for most of algorithms. Br(1) is the angular threshold and Br(2) is the radial error.
- convalue	Convergence accuracy. The higher is this value the slower will take to converge. (might never converge if too high, but will stop when the max number is reached)
-niterconv	How many iterations before checking if a minimum has been reached (ie. Convergence is achieved)
- chival	Not used
Opt.scan	Controls the scan forming. The only availabel options tells the maximum number of points before completing a full scan. Generally leave unchanged
Opt.error	Tells whether to display the SLAM result

Optimal parameters

	Angular threshold	Radial threshold	Grid resolution	Number of minimum iterations
Parameter	Opt.scanmatcher.Br(1)	Opt.scanmatcher.Br(2)	Opt.map.resolution	Opt.scanmatcher.niterconv
ICP	-	0.3 + σ	-	3
IDC	0.1 + σ	0.3 + σ	-	3
MBICP	-	0.5 + σ	-	3
PIC	-	-	-	3
NDT	-	-	3	3
LF-SOG	-	0.3 + σ	-	3
GENETIC	0.2 + σ/2	0.5 + σ/2	0.1	5
MONTECARLO	0.2 + σ/2	0.7+σ	0.1	4
GMAPPING	0.2 + σ	2+2σ^2	0.1	5
ANGLEHISTOGRAM	-	-	1	-
FOURIER	-	4+2σ^2	0.5	-
HOUGH	-	1+σ	4	-

'-' indicates that that parameter has no effect on the behaviour. σ is the expected error in motion. (angular and radial).