# Matlab Code Introduction

## Folder Introduction & Flow Chart

### Folder Introduction:

**Main folder ‘2D-Laser-SLAM’: Is the main code to read scan data from rosbag file and calculate the relative pose ‘Zstate.mat’. To do so, run the main function ‘MainFunc.m’.**

**Subfolder ‘rosbag&data’: To save all the data. Zip file ‘MathResult’ is data generated by MATLAB code, while Zip file ‘RosResult’ is data generated by ROS code. After calculation ‘MainFunc.m’ will save all the data onto this subfolder to update all the ‘.mat’ files(Those not start with ‘R’, ‘.mat’ files start with ‘R’ is generated by ROS code). ‘CheckPureScan.m’ and ‘CheckScanProb’ will read the completed data within this subfolder to plot a global map.**

**Subfolder ‘pose graph optimize’: To do pose graph optimization, read ‘Zstate.mat’ (relative pose between robot pose with submap origin) to get optimizer robot pose to save the data ‘Xsate.mat’ onto the subfolder ‘rosbag&data’.**

### Flow Chart

A close up of text on a white background

Description automatically generated

## Pre-processing

‘SetLidarparameter.m’: Set some lidar parameters for reading scan from rosbag, like max and min range, increment angle etc..

‘SetSubmapParam.m’: Set some parameters for building both the submap and the global map.

Param.resol: grid map resolution（how many cells in 1 meter）.

param.szie: submap size in cell unit.

para.topLeftCorner: the topLeftCorner submap coordiniate in meter.

Param.nscan: number of scans in each submap.

## Which Case

Base on scan index to decide how to update the submap:

%case 1 initialize submap1

%case 2 initialize submap2

%case 3 update both submaps

%case 4 only update submap1

## Store and reset new submap

When is submap is completed, store the submap into ‘map’ and reset this submap.

## Build the submap

### InitializeSubmap

1. function submap = InitializeSubmap(scan, idx, param)
2. %Initialize submap with the first scan in this submap
3. %submap.cell is probability map
4. %submap.grid is binary map
5. %submap.distmap is distance value to represent the distance value (in cell)
6. %to the closet obstacle;

* ProbUpdate: apply the first scan(the first scan in this submap) and their relative pose([0;0;0] for first scan and submap origin) to do probability update for submap.cell value.
* submap.distmap: while calculating submap.distmap the cell whose value is greater than param.hit will be set as 1 (occupied) while others are still 0 (free).

### UpdateSubmap

1. function submap = UpdateSubmap(submap,scan,idx, param)
2. % predict robot pose on submap frame,
3. % update submap with current scan
4. % and update corespoding submap path.

* ScanMatcher:

function [Xstate,Error] = ScanMatcher(DT, scan, Xstate, param)

DT submap.distmap, Xstate predicted robot submap pose.

Apply Non-lease square to do scan to submap matching.

* PopScan :

function submap = PopScan(submap, scan, pose, param, scanIdx)

Populate a scan into this submap with optimized submap pose. And stored its relative pose with submap origin into submap.connection.

## Get Zstate

All the completed submaps will be saved on variable ‘map’. ‘GetZstate.m’ is to reformat these submap relationship between robot poses and their submap origin into specific format for the pose graph optimization code.

## Plot map

‘CheckPureScan.m’: Apply MATLAB code data to generate a scans only global map. Usually for checking whether any obvious mistake after the pose graph optimization. During this process, the global map is generated by simply plot all the scan with their poses without doing any probability update.

‘CheckScanProb.m’: Apply ROS code data to generate a global map. The ROS code will generate around 10% ~ 15% optimized poses of all the robot poses. This function can use these optimized poses to calculate all the optimized robot poses via linear interpolation. With the complete optimized robot poses, do probability update to generate the global map.