I first read <http://wiki.ros.org/navigation/Tutorials> and try it step by step.   
For paramaters I mainly used turtlebot paramaters , and fine tuned some of them(like footing and robot radius) (most of them were defaults )

I changed the“scale\_angular to value=“0.1   
These are my parameters for gmapping:

<?xml version="1.0"?>

<launch>

<master auto="start"/>

<param name="/use\_sim\_time" value="true"/>

<arg name="scan\_topic" default="/scan" />

<!--- Run gmapping -->

<node pkg="gmapping" name="slam\_gmapping" type="slam\_gmapping" output="screen">

<param name="odom\_frame" value="odom"/>

<param name="base\_frame" value="base\_link"/>

<param name="map\_frame" value="map"/>

<param name="map\_update\_interval" value="5.0"/> <!-- default: 5.0 -->

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

<param name="maxUrange" value="5.0"/>

<!-- The maximum range of the sensor. If regions with no obstacles within the range of the sensor should appear as free space in the map, set maxUrange < maximum range of the real sensor <= maxRange default: 80.0 -->

<param name="maxRange" value="7.0"/>

<!-- The sigma used by the greedy endpoint matching default: 0.05) -->

<param name="sigma" value="0.05"/>

<!-- The kernel in which to look for a correspondence default: 1-->

<param name="kernelSize" value="1"/>

<!-- The optimization step in translation default: 0.05-->

<param name="lstep" value="0.05"/>

<!-- The optimization step in rotation default: 0.05-->

<param name="astep" value="0.05"/>

<!-- The number of iterations of the scanmatcher default: 5-->

<param name="iterations" value="5"/>

<!-- The sigma of a beam used for likelihood computation default: 0.075-->

<param name="lsigma" value="0.075"/>

<!-- Gain to be used while evaluating the likelihood, for smoothing the resampling effects default: 3.0-->

<param name="ogain" value="3.0"/>

<!-- Number of beams to skip in each scan. Take only every (n+1)th laser ray for computing a match (0 = take all rays) default : 0 -->

<param name="lskip" value="0"/>

<!-- TMinimum score for considering the outcome of the scan matching good. Can avoid jumping pose estimates in large open spaces when using laser scanners with limited range (e.g. 5m). Scores go up to 600+, try 50 for example when experiencing jumping estimate issues default: 0.0-->

<param name="minimumScore" value="470"/>

<!-- Odometry error in translation as a function of translation (rho/rho) default: 0.1-->

<param name="srr" value="0.01"/>

<!-- Odometry error in translation as a function of rotation (rho/theta) default: 0.2-->

<param name="srt" value="0.02"/>

<!-- Odometry error in rotation as a function of translation default: 0.1-->

<param name="str" value="0.01"/>

<!-- Odometry error in rotation as a function of rotation default: 0.2-->

<param name="stt" value="0.02"/>

<!-- Process a scan each time the robot translates this far default : 1.0 -->

<param name="linearUpdate" value="0.3"/>

<!-- Process a scan each time the robot rotates this far default: 0.5-->

<param name="angularUpdate" value="0.5"/>

<!-- Process a scan if the last scan processed is older than the update time in seconds. A value less than zero will turn time based updates off. default: -1.0-->

<param name="temporalUpdate" value="-1.0"/>

<!-- The Neff based resampling threshold default: 1-->

<param name="resampleThreshold" value="1.0"/>

<!-- Number of particles in the filter. default 30 -->

<param name="particles" value="80"/>

<!-- Initial map size in metres default -100/100 -->

<param name="xmin" value="-50.0"/>

<param name="ymin" value="-50.0"/>

<param name="xmax" value="50.0"/>

<param name="ymax" value="50.0"/>

<!-- Processing parameters (resolution of the map) -->

<param name="delta" value="0.01"/>

<!-- Translational/angular sampling range/step for the likelihood -->

<param name="llsamplerange" value="0.01"/>

<param name="llsamplestep" value="0.01"/>

<param name="lasamplerange" value="0.005"/>

<param name="lasamplestep" value="0.005"/>

<!-- The sigma used by the greedy endpoint matching default: 1-->

<remap from="scan" to="$(arg scan\_topic)"/>

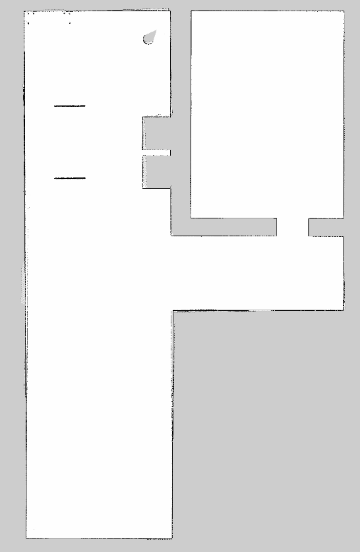
</node>

<!-- Move base -->

<include file="$(find vrep\_teleop)/launch/teleop\_key.launch"/>

</launch>

Video geomapping:   
<https://youtu.be/CennZ0a91r0>

Map:   


My navigation parameters:

amcl.launch

<?xml version="1.0"?>

<launch>

<master auto="start"/>

<param name="/use\_sim\_time" value="true"/>

<!-- Map server -->

<arg name="map\_file" default="/home/alpsark/ros-ws/src/nav\_stack\_tuning/launch/map.yaml"/>

<node name="map\_server" pkg="map\_server" type="map\_server" args="$(arg map\_file)" />

<!-- Localization -->

<node pkg="amcl" type="amcl" name="amcl" output="screen">

<param name="use\_map\_topic" value="true"/>

<remap from="scan" to="/scan"/>

<param name="odom\_frame\_id" value="odom"/>

<param name="base\_frame\_id" value="base\_link"/>

<param name="global\_frame\_id" value="map"/>

<!--Diff for 2 wheel omni for omnidirectional robots either "diff", "omni", "diff-corrected" or "omni-corrected"-->

<param name="odom\_model\_type" value="omni"/>

<!-- Specifies the expected noise in odometry's rotation estimate from the rotational component of the robot's motion. Default="0.2"-->

<param name="odom\_alpha1" value="0.2"/>

<!-- Specifies the expected noise in odometry's rotation estimate from translational component of the robot's motion. Default="0.2"-->

<param name="odom\_alpha2" value="0.2"/>

<!-- Specifies the expected noise in odometry's translation estimate from the translational component of the robot's motion. Default = "0.2" -->

<param name="odom\_alpha3" value="0.2"/>

<!-- Specifies the expected noise in odometry's translation estimate from the rotational component of the robot's motion. Default="0.2"-->

<param name="odom\_alpha4" value="0.2"/>

<!--Transaltiion related noise paramater (only used if model is "omni") Default="0.2"-->

<param name="odom\_alpha5" value="0.1"/>

<!-- Maximum rate (Hz) at which scans and paths are published for visualization, -1.0 to disable. -->

<param name="gui\_publish\_rate" value="10.0"/>

<!-- How many evenly-spaced beams in each scan to be used when updating the filter. Default="30"-->

<param name="laser\_max\_beams" value="80"/>

<param name="laser\_max\_range" value="5.0"/>

<param name="min\_particles" value="500"/>

<param name="max\_particles" value="3000"/>

<!-- Maximum error between the true distribution and the estimated distribution. Default="0.01"-->

<param name="kld\_err" value="0.05"/>

<!--Upper standard normal quantile for (1 - p), where p is the probability that the error on the estimated distrubition will be less than kld\_err. Default="0.99"-->

<param name="kld\_z" value="0.99"/>

<!-- Mixture weight for the z\_hit part of the model. Default="0.95"-->

<param name="laser\_z\_hit" value="0.5"/>

<!-- Mixture weight for the z\_short part of the model. Default="0.1"-->

<param name="laser\_z\_short" value="0.05"/>

<!-- Mixture weight for the z\_max part of the model. Default="0.05"-->

<param name="laser\_z\_max" value="0.05"/>

<!-- Mixture weight for the z\_rand part of the model. Default="0.05"-->

<param name="laser\_z\_rand" value="0.5"/>

<!-- Standard deviation for Gaussian model used in z\_hit part of the model. Default="0.2"-->

<param name="laser\_sigma\_hit" value="0.2"/>

<!-- Exponential decay parameter for z\_short part of model. Default="0.1"-->

<param name="laser\_lambda\_short" value="0.1"/>

<!--Which model to use, either beam, likelihood\_field, or likelihood\_field\_prob (same as likelihood\_field but incorporates the beamskip feature, if enabled). -->

<param name="laser\_model\_type" value="likelihood\_field"/>

<!-- < Maximum distance to do obstacle inflation on map, for use in likelihood\_field model. default= 2.0 meters"/> -->

<param name="laser\_likelihood\_max\_dist" value="2.0"/>

<!-- Translational movement required before performing a filter update. Default="0.2"-->

<param name="update\_min\_d" value="0.25"/>

<!-- Rotational movement required before performing a filter update. Default="pi/6.0 rad"-->

<param name="update\_min\_a" value="0.2"/>

<!-- Number of filter updates required before resampling. Default="2"-->

<param name="resample\_interval" value="1"/>

<!-- Time with which to post-date the transform that is published, to indicate that this transform is valid into the future. Default:0.1-->

<param name="transform\_tolerance" value="0.2"/>

<!--Exponential decay rate for the slow average weight filter, used in deciding when to recover by adding random poses. A good value might be 0.001.-->

<param name="recovery\_alpha\_slow" value="0.0"/>

<!--Exponential decay rate for the fast average weight filter, used in deciding when to recover by adding random poses. A good value might be 0.1. -->

<param name="recovery\_alpha\_fast" value="0.0"/>

<param name="initial\_pose\_x" value="0.0"/>

<param name="initial\_pose\_y" value="0.0"/>

<param name="initial\_pose\_a" value="0.0"/>

</node>

</launch>

move\_base.launch

<?xml version="1.0"?>

<launch>

<master auto="start"/>

<param name="/use\_sim\_time" value="true"/>

<!-- Map server -->

<arg name="map\_file" default="/home/alpsark/ros-ws/src/nav\_stack\_tuning/launch/map.yaml"/>

<node name="map\_server" pkg="map\_server" type="map\_server" args="$(arg map\_file)" />

<!-- Move base -->

<node pkg="move\_base" type="move\_base" respawn="false" name="move\_base" output="screen">

<rosparam file="/home/alpsark/ros-ws/src/nav\_stack\_tuning/launch/costmap\_common\_params.yaml" command="load" ns="global\_costmap" />

<rosparam file="/home/alpsark/ros-ws/src/nav\_stack\_tuning/launch/costmap\_common\_params.yaml" command="load" ns="local\_costmap" />

<rosparam file="/home/alpsark/ros-ws/src/nav\_stack\_tuning/launch/local\_costmap\_params.yaml" command="load" />

<rosparam file="/home/alpsark/ros-ws/src/nav\_stack\_tuning/launch/global\_costmap\_params.yaml" command="load" />

<rosparam file="/home/alpsark/ros-ws/src/nav\_stack\_tuning/launch/move\_base\_params.yaml" command="load" />

<rosparam file="/home/alpsark/ros-ws/src/nav\_stack\_tuning/launch/global\_planner\_params.yaml" command="load" />

<rosparam file="/home/alpsark/ros-ws/src/nav\_stack\_tuning/launch/base\_local\_planner\_params.yaml" command="load" />

<remap from="odom" to="odom"/>

<remap from="scan" to="/scan"/>

<remap from="cmd\_vel" to="/cmd\_vel"/>

<!-- reset frame\_id parameters using user input data -->

<param name="global\_costmap/global\_frame" value="map"/>

<param name="global\_costmap/robot\_base\_frame" value="base\_link"/>

<param name="local\_costmap/global\_frame" value="odom"/>

<param name="local\_costmap/robot\_base\_frame" value="base\_link"/>

<param name="TrajectoryPlannerROS/global\_frame\_id" value="odom"/>

<param name="base\_global\_planner" type="string" value="navfn/NavfnROS" />

<param name="base\_local\_planner" value="base\_local\_planner/TrajectoryPlannerROS"/>

</node>

</launch>

Params files for move\_base:   
move\_base\_params.yaml

# Move base node parameters. For full documentation of the parameters in this file, please see

#

# http://www.ros.org/wiki/move\_base

#

shutdown\_costmaps: false

controller\_frequency: 5.0

controller\_patience: 3.0

planner\_frequency: 1.0

planner\_patience: 5.0

oscillation\_timeout: 10.0

oscillation\_distance: 0.2

# local planner - default is trajectory rollout dwa\_local\_planner/DWAPlannerROS

base\_local\_planner: "base\_local\_planner/TrajectoryPlannerROS"

base\_global\_planner: "navfn/NavfnROS"

base\_local\_planner\_params.yaml

TrajectoryPlannerROS:

# Robot Configuration Parameters

max\_vel\_x: 3.0 # 0.55

min\_vel\_x: 0.0

max\_vel\_y: 0.0 # diff drive robot

min\_vel\_y: 0.0 # diff drive robot

max\_trans\_vel: 3.0 # choose slightly less than the base's capability

min\_trans\_vel: 1.0 # this is the min trans velocity when there is negligible rotational velocity

# trans\_stopped\_vel: 0.1

# Warning!

# do not set min\_trans\_vel to 0.0 otherwise dwa will always think translational velocities

# are non-negligible and small in place rotational velocities will be created.

max\_vel\_theta: 0.5

min\_vel\_theta: -0.5

min\_in\_place\_vel\_theta: -0.314

# rot\_stopped\_vel: 0.05

acc\_lim\_x: 1.0 # maximum is theoretically 2.0, but we

acc\_lim\_theta: 2.0

acc\_lim\_y: 0.0 # diff drive robot

# Goal Tolerance Parameters

yaw\_goal\_tolerance: 0.3 # 0.05

xy\_goal\_tolerance: 0.2 # 0.10

# latch\_xy\_goal\_tolerance: false

# Forward Simulation Parameters

sim\_time: 10.0 # 1.7

vx\_samples: 6 # 3

vy\_samples: 1 # diff drive robot, there is only one sample

vtheta\_samples: 20 # 20

# Trajectory Scoring Parameters

path\_distance\_bias: 64.0 # 32.0 - weighting for how much it should stick to the global path plan

goal\_distance\_bias: 24.0 # 24.0 - swighting for how much it should attempt to reach its goal

occdist\_scale: 0.686 # 0.5 0.01 - weighting for how much the controller should avoid obstacles

forward\_point\_distance: 0.325 # 0.325 - how far along to place an additional scoring point

stop\_time\_buffer: 0.2 # 0.2 - amount of time a robot must stop in before colliding for a valid traj.

scaling\_speed: 0.25 # 0.25 - absolute velocity at which to start scaling the robot's footprint

max\_scaling\_factor: 0.2 # 0.2 - how much to scale the robot's footprint when at speed.

meter\_scoring: true

# Oscillation Prevention Parameters

oscillation\_reset\_dist: 0.05 # 0.05 - how far to travel before resetting oscillation flags

# Debugging

publish\_traj\_pc : true

publish\_cost\_grid\_pc: true

global\_frame\_id: odom

# Differential-drive robot configuration - necessary?

# holonomic\_robot: false

global\_planner\_params.yaml

GlobalPlanner: # Also see: http://wiki.ros.org/global\_planner

old\_navfn\_behavior: false # Exactly mirror behavior of navfn, use defaults for other boolean parameters, default false

use\_quadratic: true # Use the quadratic approximation of the potential. Otherwise, use a simpler calculation, default true

use\_dijkstra: false # Use dijkstra's algorithm. Otherwise, A\*, default true

use\_grid\_path: false # Create a path that follows the grid boundaries. Otherwise, use a gradient descent method, default false

allow\_unknown: false # Allow planner to plan through unknown space, default true

# Needs to have track\_unknown\_space: true in the obstacle / voxel layer (in costmap\_commons\_param) to work

planner\_window\_x: 0.0 # default 0.0

planner\_window\_y: 0.0 # default 0.0

default\_tolerance: 0.0 # If goal in obstacle, plan to the closest point in radius default\_tolerance, default 0.0

publish\_scale: 100 # Scale by which the published potential gets multiplied, default 100

planner\_costmap\_publish\_frequency: 0.0 # default 0.0

lethal\_cost: 253 # default 253

neutral\_cost: 50 # default 50

cost\_factor: 3.0 # Factor to multiply each cost from costmap by, default 3.0

publish\_potential: true # Publish Potential Costmap (this is not like the navfn pointcloud2 potential), default true

costmap\_common\_params.yaml

max\_obstacle\_height: 0.60 # assume something like an arm is mounted on top of the robot

# Obstacle Cost Shaping (http://wiki.ros.org/costmap\_2d/hydro/inflation)

robot\_radius: 0.2

# distance a circular robot should be clear of the obstacle (kobuki: 0.18)

footprint: [[-0.2, -0.2], [-0.2, 0.2], [0.2,0.2], [0.2, -0.2]]

map\_type: voxel

obstacle\_layer:

enabled: true

max\_obstacle\_height: 0.6

origin\_z: 0.0

z\_resolution: 0.2

z\_voxels: 2

unknown\_threshold: 15

mark\_threshold: 0

combination\_method: 1

track\_unknown\_space: true #true needed for disabling global path planning through unknown space

obstacle\_range: 2.5

raytrace\_range: 3.0

z\_resolution: 0.2

z\_voxels: 2

publish\_voxel\_map: false

observation\_sources: scan

scan:

data\_type: LaserScan

topic: /scan

marking: true

clearing: true

min\_obstacle\_height: 0.25

max\_obstacle\_height: 0.35

#cost\_scaling\_factor and inflation\_radius were now moved to the inflation\_layer ns

inflation\_layer:

enabled: true

cost\_scaling\_factor: 4.0

# exponential rate at which the obstacle cost drops off (default: 10)

inflation\_radius: 1.5 # max. distance from an obstacle at which costs are incurred for planning paths.

static\_layer:

enabled: true

global\_costmap\_params.yaml

global\_costmap:

global\_frame: map

robot\_base\_frame: base\_link

update\_frequency: 1.0

publish\_frequency: 0.5

static\_map: true

transform\_tolerance: 0.5

plugins:

- {name: static, type: "costmap\_2d::StaticLayer"}

- {name: obstacle\_layer, type: "costmap\_2d::VoxelLayer"}

- {name: inflation, type: "costmap\_2d::InflationLayer"}

local\_costmap\_params.yaml

local\_costmap:

global\_frame: odom

robot\_base\_frame: base\_link

update\_frequency: 5.0

publish\_frequency: 2.0

static\_map: false

rolling\_window: true

width: 4.0

height: 4.0

resolution: 0.01

transform\_tolerance: 0.5

plugins:

- {name: obstacle\_layer, type: "costmap\_2d::VoxelLayer"}

- {name: inflation, type: "costmap\_2d::InflationLayer"}

Video navigation:

<https://youtu.be/KO2W7EEY17A>