

NDT MCL ROS Node Documentation  
Santosh Thoduka and Aleksandar Mitrevski  
Bonn-Rhein-Sieg University of Applied Science

June 19, 2014

# Contents

<b>1</b>	<b>Introduction . . . . .</b>	<b>3</b>
<b>2</b>	<b>Target Audience . . . . .</b>	<b>3</b>
<b>3</b>	<b>NDT MCL - A Short Introduction . . . . .</b>	<b>3</b>
<b>4</b>	<b>Node Description . . . . .</b>	<b>4</b>
<b>5</b>	<b>Implemented Modifications . . . . .</b>	<b>4</b>
<b>6</b>	<b>Usage . . . . .</b>	<b>5</b>
<b>7</b>	<b>Potential Improvements . . . . .</b>	<b>7</b>
<b>8</b>	<b>Summary and Conclusions . . . . .</b>	<b>7</b>
<b>9</b>	<b>References . . . . .</b>	<b>7</b>

## 1 Introduction

The NDT-MCL (Normal distribution transform Monte Carlo localisation) ROS node is an implementation of the NDT-MCL algorithm for localisation against an NDT map. This document makes an attempt at introducing the node, including implementation and usage details, in a manner that improves the presentation provided by [4]. In order to do that, the rest of the document includes a brief introduction to NDT-MCL in section 3, a description of the node in section 4, our modifications in section 5, some usage details in section 6, as well as suggestions for potential improvements and modifications in section 7.

## 2 Target Audience

This document intends to provide an overview of the NDT-MCL ROS node, focusing not only on its usage, but also on certain internal implementation details. As a result, it might be useful for developers who need to work with, modify, or maintain the node, but also for developers who are developing similar nodes and need an orientation about the complexity of such a development effort.

Ideally, the document will be updated as the node is modified and improved. Its ultimate goal is being a constantly updated development documentation where developers can find all the necessary details regarding the NDT-MCL node.

## 3 NDT MCL - A Short Introduction

Normal distribution transform Monte Carlo localisation (NDT-MCL), described in [1], is a localisation algorithm whose goal is providing a close and consistent estimate to the true position of a robot in a given environment. NDT-MCL uses a discretised, grid-based, representation of an environment; however, unlike other localisation algorithms, which use occupancy grids to represent a map, NDT-MCL parameterises grid cells by Gaussian distributions, employing a so-called NDT representation as suggested by [2].

Given that the algorithm is an instance of the Monte Carlo family, the localisation steps remain the same as in grid-based localisation: calculating a pose likelihood, updating particle weights, and resampling particles. The only difference in NDT-MCL is the likelihood calculation, such that  $L2$ -likelihood is used to express the discrepancy between the estimated and the actual robot pose.

## 4 Node Description

The main activity of the NDT-MCL node consists of processing laser scans in a callback function, called due to the node's subscription to a laser scan topic, and updating the estimated pose of a robot based on the scans and the robot's motion. A UML diagram of the node and some background code that implements a particle filter is shown in the figure below:

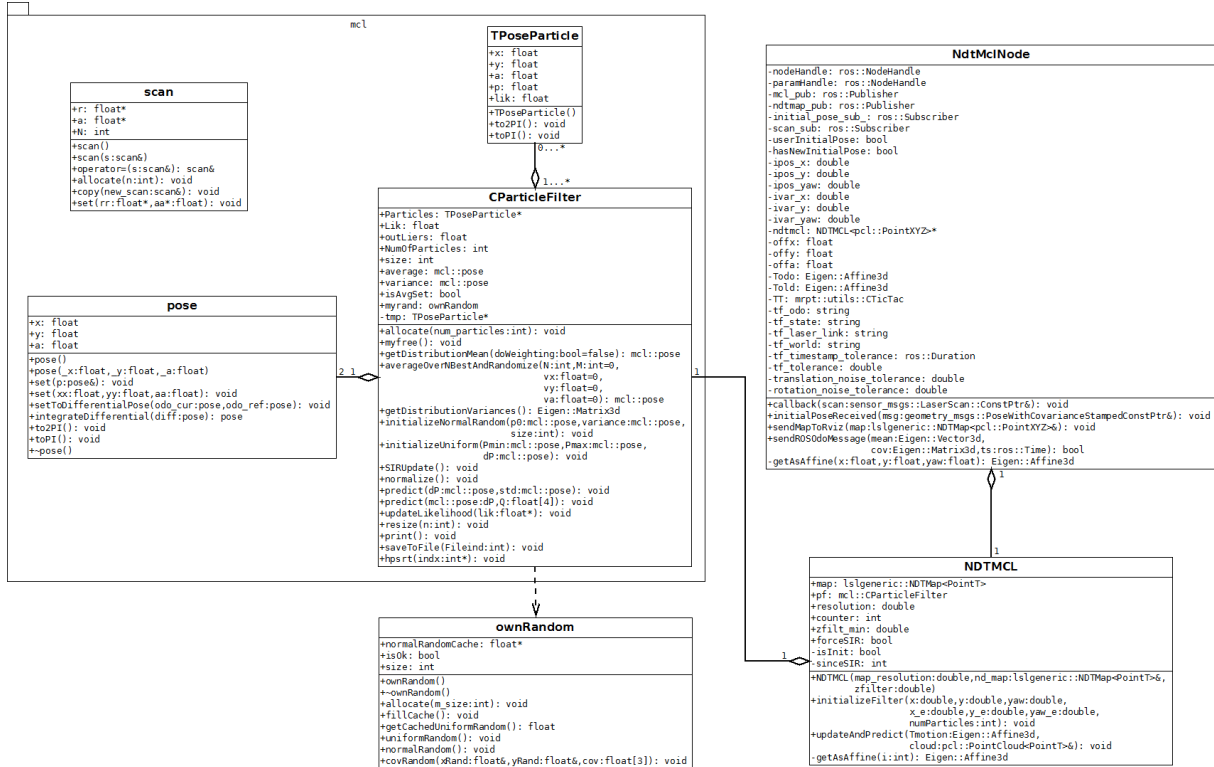


Figure 1: UML diagram of the NDT-MCL node and its dependencies

## 5 Implemented Modifications

The following list of modifications represents our contribution to the node's code:

- As can be seen in figure 1, the node is implemented as a class; this represents a modification of the initial code, which was an unstructured collection of functions and global variables.
- The initial implementation of the node was using a ground truth estimate of the robot's position and the localisation pose estimate was published with respect to the ground truth. The modified version of the node uses the same principle as [6], such that the published pose estimate represents a transform between the localisation frame and a robot's odometry frame.

- Some parameters were added to the launch file used for initialising the node; more details about these parameters are given in the following section.

## 6 Usage

Using the improved NDT-MCL node requires:

- An NDT map of the environment where a robot should be localised; such a map can be developed using the NDT Fuser ROS node. An explanation of NDT Fuser is beyond the scope of this document, but a sufficiently good explanation of the map creation process is given in [3].
- A launch file that will start the localisation node.

An example launch file used for initialising the NDT-MCL node is given below; some of these parameters were also used while working with our robot:

Listing 1: A launch file for initialising NDT-MCL

```
<?xml version="1.0"?>
<launch>
  <node pkg="ndt_mcl" type="2d_ndt_mcl_node" required="true" name="NDTMCL">
    <param name="input_laser_topic" value="scan_front" />
    <param name="tf_base_link" value="base_footprint" />
    <param name="tf_laser_link" value="base_laser_front_link" />
    <param name="tf_odom" value="odom" />
    <param name="tf_world" value="map" />
    <param name="tf_timestamp_tolerance" value="1.1" />
    <param name="translation_noise_tolerance" value="0.1" />
    <param name="rotation_noise_tolerance" value="1.5" />

    <param name="use_pose_offset" value="true" />
    <param name="sensor_pose_x" value="0.3" />
    <param name="sensor_pose_y" value="0.0" />
    <param name="sensor_pose_th" value="-0.03" />

    <param name="load_map_from_file" value="true" />
    <param name="map_file_name" value="map.jff" />

    <param name="set_initial_pose" value="true" />
    <param name="initial_pose_x" value="0.0" />
    <param name="initial_pose_y" value="0.0" />
    <param name="initial_pose_yaw" value="0.0" />

    <param name="save_output_map" value="false" />
    <param name="map_resolution" value="0.3" />
  </node>
</launch>
```

Some of the parameters in the launch file were already present in the original implementation of the node, while others were added additionally in order to increase the node's customisation. For completeness, the following table lists all parameters that can be set through the launch file, a short description of them, and their default values. The parameters that have a \* next to their name are those that were included additionally:

Table 1: Parameters used by the NDT-MCL node

Parameter name	Description	Default value
<i>input_laser_topic</i>	Name of the topic where laser scans are published	<i>/base_scan</i>
<i>tf_base_link</i>	Name of a robot's base link frame	<i>/base_link</i>
<i>tf_laser_link</i>	Name of a laser frame; the frame should be expressed in terms of the base link frame	<i>/hokuyo1_link</i>
<i>tf_odom*</i>	Name of the robot's odometry frame	<i>odom</i>
<i>tf_world*</i>	Name of the localisation frame	<i>map</i>
<i>tf_timestamp_tolerance*</i>	A number describing a discrepancy between the times when a transform is published and requested	<i>1.0</i>
<i>translation_noise_tolerance*</i>	Translation noise tolerance (in meters) used for deciding whether a robot has really moved	<i>0.01</i>
<i>rotation_noise_tolerance*</i>	Rotation noise tolerance (angle in degrees) used for deciding whether a robot has really rotated	<i>0.5</i>
<i>sensor_pose_x</i>	X coordinate of a laser with respect to the base frame	<i>0.0</i>
<i>sensor_pose_y</i>	Y coordinate of a laser with respect to the base frame	<i>0.0</i>
<i>sensor_pose_th</i>	Rotation angle of a laser with respect to the base frame	<i>0.0</i>
<i>load_map_from_file</i>	A boolean value indicating whether to load an environment map from a file	<i>false</i>
<i>map_file_name</i>	Name of a .jff file containing an NDT map; used if <i>load_map_from_file</i> is set to <i>true</i>	<i>basement.ndmap</i>
<i>forceSIR</i>	A boolean value indicating whether to re-sample particles at each iteration	<i>false</i>
<i>set_initial_pose</i>	Indicates whether an initial pose is available and should be assigned	<i>true</i>
<i>initial_pose_x</i>	X coordinate of an initial pose	<i>0.0</i>
<i>initial_pose_y</i>	Y coordinate of an initial pose	<i>0.0</i>
<i>initial_pose_yaw</i>	Rotation angle of an initial pose	<i>0.0</i>

<i>map_resolution</i>	Resolution of the map in which a robot is going to be localised	0.2
-----------------------	---	-----

In order to work with a complete navigation system, the navigation of the robot should be adapted to the NDT representation used by an environment's map.

## 7 Potential Improvements

## 8 Summary and Conclusions

The NDT-MCL (Normal distribution transform Monte Carlo localisation) ROS node is an implementation of the NDT-MCL algorithm for localisation against an NDT map. Being developed recently, the node has multiple potential areas of improvements: some of them were addressed during our four-week software development project, while others remain open.

We hope that this documentation will be a good starting point for developers who need to work with the node, helping them to use and improve NDT-MCL effectively.

## 9 References

- [1] J. Saarinen, H. Andreasson, T. Stoyanov, and A. J. Lilienthal, "Normal Distributions Transform Monte-Carlo Localization (NDT-MCL)," in *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, Tokyo, Japan, 2013, pp. 382-389.
- [2] P. Biber, "The Normal Distributions Transform: A New Approach to Laser Scan Matching," in *Intelligent Robots and Systems, 2003. (IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on*, Las Vegas, NV, 2003, pp. 2743-2748.
- [3] T. Stoyanov. (2013). *Using NDT Fuser to create an NDT map* [Online]. Available: [http://wiki.ros.org/perception\\_oru/Tutorials/Using%20NDT%20Fuser%20to%20create%20an%20NDT%20map](http://wiki.ros.org/perception_oru/Tutorials/Using%20NDT%20Fuser%20to%20create%20an%20NDT%20map)
- [4] T. Stoyanov. (2014). *Running NDT-MCL examples* [Online]. Available: [http://wiki.ros.org/perception\\_oru/Tutorials/Running%20NDT-MCL%20examples](http://wiki.ros.org/perception_oru/Tutorials/Running%20NDT-MCL%20examples)
- [5] T. Stoyanov. (2014). *NDM MCL source code* [Online]. Available: [https://github.com/tstoyanov/perception\\_oru-release/tree/release/hydro/ndt\\_mcl](https://github.com/tstoyanov/perception_oru-release/tree/release/hydro/ndt_mcl)

[6] B. Gerkey. (2014). *AMCL source code* [Online]. Available: [https://github.com/ros-planning/navigation/blob/hydro-devel/amcl/src/amcl\\_node.cpp](https://github.com/ros-planning/navigation/blob/hydro-devel/amcl/src/amcl_node.cpp)