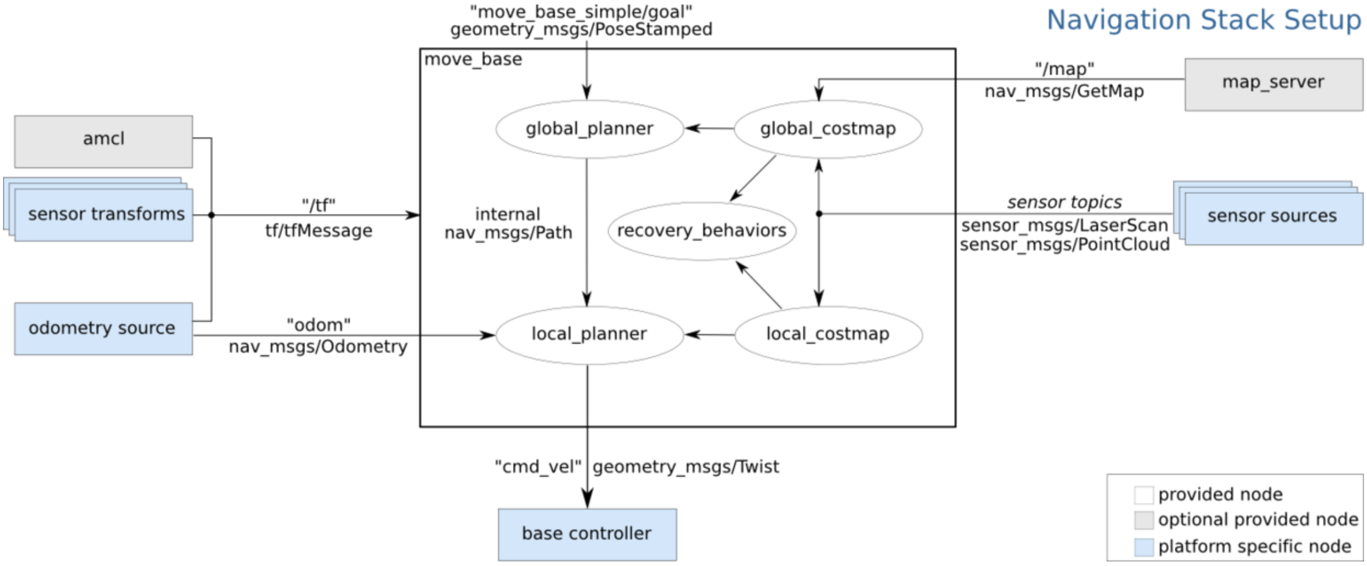
1.概述

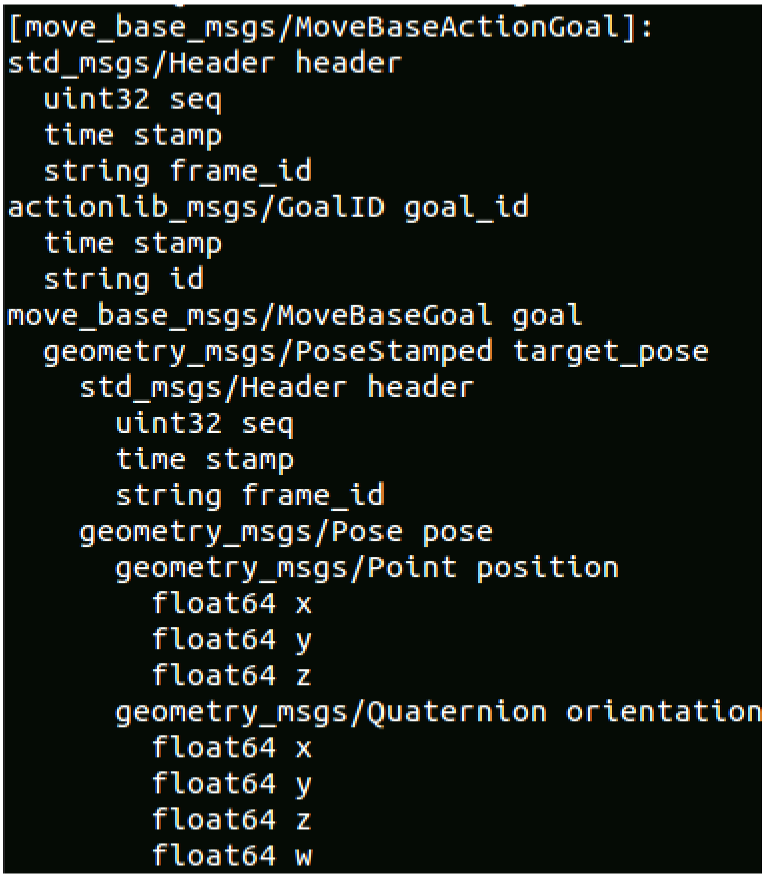
movebase利用actionlib包提供使得我们的机器人到达一个设置的目标点的包 先上一张官方经典的图



- 图中蓝色部分与特定机器人平台有关，灰色部分是可选的，白色部分是必须的
- 可以看到`amcl`和`map_server`都是不必须的
- 机器人平台输入相关传感器、里程计及tf信息
- 输入为一个`goal`(目标点坐标)
- 输出一个`cmd_vel`（速度）

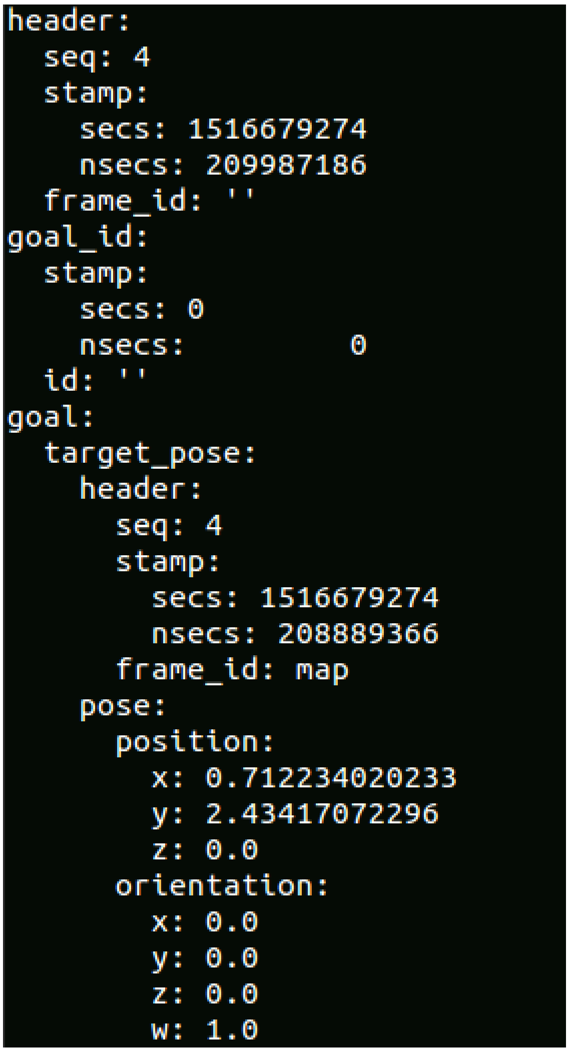
2.指定导航点

首先我们查看下`MoveBaseActionGoal`的定义 `rosmsg show MoveBaseActionGoal`



```
[move_base_msgs/MoveBaseActionGoal]:
std_msgs/Header header
  uint32 seq
  time stamp
  string frame_id
actionlib_msgs/GoalID goal_id
  time stamp
  string id
move_base_msgs/MoveBaseGoal goal
  geometry_msgs/PoseStamped target_pose
    std_msgs/Header header
      uint32 seq
      time stamp
      string frame_id
    geometry_msgs/Pose pose
      geometry_msgs/Point position
        float64 x
        float64 y
        float64 z
      geometry_msgs/Quaternion orientation
        float64 x
        float64 y
        float64 z
        float64 w
```

运行导航逻辑后，`rostopic echo /move_base/goal`, 点击`2D Nav Goal`,输出



```
header:
  seq: 4
  stamp:
    secs: 1516679274
    nsecs: 209987186
  frame_id: ''
goal_id:
  stamp:
    secs: 0
    nsecs:         0
  id: ''
goal:
  target_pose:
    header:
      seq: 4
      stamp:
        secs: 1516679274
        nsecs: 208889366
      frame_id: map
    pose:
      position:
        x: 0.712234020233
        y: 2.43417072296
        z: 0.0
      orientation:
        x: 0.0
        y: 0.0
        z: 0.0
        w: 1.0
```

结合上面2个输出可以看到geometry_msgs/PoseStamped中

- frame_id为参考坐标系
- goal为目标点的位置(pose)与姿态(orientation)

官方有个例子navigation_tutorials

```
#include <ros/ros.h>
#include <move_base_msgs/MoveBaseAction.h>
#include <actionlib/client/simple_action_client.h>
#include <tf/transform_datatypes.h>

#include <boost/thread.hpp>

typedef actionlib::SimpleActionClient<move_base_msgs::MoveBaseAction>
MoveBaseClient;

void spinThread(){
  ros::spin();
}

int main(int argc, char** argv){
  ros::init(argc, argv, "simple_navigation_goals");

  ros::NodeHandle n;

  boost::thread spin_thread = boost::thread(boost::bind(&spinThread));

  MoveBaseClient ac("pose_base_controller");

  //give some time for connections to register
  sleep(2.0);

  move_base_msgs::MoveBaseGoal goal;

  //we'll send a goal to the robot to move 2 meters forward
  goal.target_pose.header.frame_id = "base_link";
  goal.target_pose.header.stamp = ros::Time::now();

  goal.target_pose.pose.position.x = 2.0;
  goal.target_pose.pose.position.y = 0.2;
  goal.target_pose.pose.orientation = tf::createQuaternionMsgFromYaw(M_PI);

  ROS_INFO("Sending goal");
  ac.sendGoal(goal);

  ac.waitForResult();

  if(ac.getState() == actionlib::SimpleClientGoalState::SUCCEEDED)
    ROS_INFO("Hooray, the base moved 2 meters forward");
  else
    ROS_INFO("The base failed to move forward 2 meters for some reason");
  return 0;
}
```

以`base_link`为参考坐标系, 目标就是前面2.0(`x = 2.0`)，左0.2(`y = 0.2`)，方向选择180°(`orientation = tf::createQuaternionMsgFromYaw(M_PI)`)