ROS Resources: Documentation | Support | Discussion Forum | Service Status | Q&A answers.ros.org

ROS ANSWERS

tags users badges

Hi there! Please sign in help

ALL UNANSWERED Search or ask your question ASK YOUR QUESTION

0

Move a certain distance, turn, then move (Odometry topic)

odometry odom movement move forward turn geometry_msgs twist nav_msgs beginner navigation

asked Mar 17 '15
dylankc
11 ●5 ●5 ●7

Currently, I am able to have my robot move forward for a certain distance, turn right by a certain degree, then move forward again. However, it is extremely inaccurate as highlighted in ahendrix's answer here "http://answers.ros.org/question/204973/robot-pose-slightly-off-when-publishing-to-cmd_vel/".

I want to re-make this program by using the odometry topic. This is the current code that I have, I am currently working on the TurtleBot 2.

```
#include <ros/ros.h>
#include <geometry_msgs/Twist.h>
#include <math.h>

int main(int argc, char **argv)
{
const double PI = 3.14159265358979323846;

ros::init(argc, argv, "move_pub");
ros::NodeHandle n;
ros::Publisher movement_pub = n.advertise<geometry_msgs::Twist>("mobile_base/commands/velocity",1); //for sensors the value after , should be higher to get a more accurate result (queued)
ros::Rate rate(10); //the larger the value, the "smoother" , try value of 1 to see "jerk" movement

//move forward
ros::Time start = ros::Time::now();
while(ros::Time::now() - start < ros::Duration(5.0))
{
    geometry_msgs::Twist move;
    //velocity controls
    move.linear.x = 0.1; //speed value m/s
    move.angular.z = 0;
    movement_pub.publish(move);

    ros::spinOnce();
    rate.sleep();
}
//turn right
ros::Time start_turn = ros::Time::now();
while(ros::Time::now() - start_turn < ros::Duration(4.0))
{
    geometry_msgs::Twist move;
    //velocity controls
    move.linear.x = 0; //speed value m/s
    move.angular.z = -2.25;
    movement_pub.publish(move);

    ros::spinOnce();
    rate.sleep();
}
//move forward again
ros::Time start2 = ros::Time::now();
while(ros::Time::now() - start2 < ros::Duration(5.0))
{
    geometry_msgs::Twist move;
    //velocity controls
    move.linear.x = 0.1; //speed value m/s
    move.angular.z = 0;
    movement_pub.publish(move);
```

Question Tools

Follow

1 follower

subscribe to rss feed

Stats

Asked:	Mar 17 '15
Seen:	3,158 times
Last updated:	Aug 28 '17

Related questions

- How to get Position of turtlebot?
- Turning 90degrees using the odometry orientation message
- robot_pose_ekf
- navigation using kinect without laser scan
- twist message: angular velocity unit
- an error waiting for move base server to come up
- TF_OLD_DATA ignoring data from the past for frame odom
- What does Velocity in nav_msgs/Odometry actually mean?
- Obstacle avoidance in move_base package
- navigation for non-holonomic robots

```
    ros::spinOnce();
    rate.sleep();
}

return 0;
}
```

How should I go about replicating this program using the odometry topic?

I am a beginner at ROS, all of your help will be greatly appreciated. I am currently on my school holidays and I am feeling great joy learning something new :)

Comments

http://answers.ros.org/question/20513...
I was able to find a python script in a library book that achieves this. However, I dont know how to replicate as a ROS C++ node, I posted it as a separate question.
dylankc (Mar 17 '15)

add a comment

1 Answer

Sort by » oldest newest most voted

0

Hello,

I started from your code and did some changes in order to use an Odometry message to measure how much the robot moved.

I couldn't format the code properly using this editor, so I created a post: http://www.theconstructsim.com/move-c... But you can see the code below:

answered Aug 28 '17

marcoarruda
216 ●4 ●7
https://www.linkedin.c...

updated Aug 29 '17

```
#include <ros/ros.h>
#include <tf/tf.h>
#include <geometry_msgs/Twist.h>
#include <geometry_msgs/Pose2D.h>
#include <nav_msgs/Odometry.h>

#include <math.h>

geometry_msgs::Pose2D current_pose;
ros::Publisher pub_pose2d;

void odomCallback(const nav_msgs::OdometryConstPtr& msg)
{
    // linear position
    current_pose.x = msg->pose.pose.position.x;
    current_pose.y = msg->pose.pose.position.y;

    // quaternion to RPY conversion
    tf::Quaternion q(
        msg->pose.pose.orientation.x,
        msg->pose.pose.orientation.y,
        msg->pose.pose.orientation.z,
        msg->pose.pose.orientation.w);
    tf::Matrix3x3 m(q);
    double roll, pitch, yaw;
    m.getRPY(roll, pitch, yaw);

    // angular position
    current_pose.theta = yaw;
    pub_pose2d.publish(current_pose);
}

int main(int argc, char **argv)
{
    const double PI = 3.14159265358979323846;

    ROS_INFO("start");

    ros::init(argc, argv, "move_pub");
    ros::NodeHandle n;
    ros::Subscriber sub_odometry = n.subscribe("odom", 1, odomCallback);
    ros::Publisher movement_pub = n.advertise("cmd_vel",1); //for sensors the value after , sh
ould be higher to get a more accurate result (queued)
    pub_pose2d = n.advertise("turtlebot_pose2d", 1);
    ros::Rate rate(10); //the larger the value, the "smoother" , try value of 1 to see "jerk"
 movement

    //move forward
    ROS_INFO("move forward");
    ros::Time start = ros::Time::now();
    while(ros::ok() && current_pose.x < 1.5)
    {
        geometry_msgs::Twist move;
        //velocity controls
        move.linear.x = 0.2; //speed value m/s
        move.angular.z = 0;
        movement_pub.publish(move);

        ros::spinOnce();
        rate.sleep();
    }
    //turn right
    ROS_INFO("turn right");
    ros::Time start_turn = ros::Time::now();
    while(ros::ok() && current_pose.theta > -PI/2)
    {
        geometry_msgs::Twist move;
        //velocity controls
        move.linear.x = 0; //speed value m/s
        move.angular.z = -0.3;
        movement_pub.publish(move);

        ros::spinOnce();
        rate.sleep();
    }
    //move forward again
    ROS_INFO("move forward");
    ros::Time start2 = ros::Time::now();
    while(ros::ok() && current_pose.y > -1.5)
    {
        geometry_msgs::Twist move;
        //velocity controls
        move.linear.x = 0.2; //speed value m/s
        move.angular.z = 0;
        movement_pub.publish(move);

        ros::spinOnce();
        rate.sleep();
    }

    // just stop
    while(ros::ok()) {
        geometry_msgs::Twist move;
        move.linear.x = 0;
        move.angular.z = 0;
        movement_pub.publish(move);

        ros::spinOnce();
        rate.sleep();
    }

    return 0;
}
```

If you want to reproduce my experiment, you can use the Turtlebot 2 public simulation available in RDS (https://rds.theconstructsim.com/). But you can do it in your own computer too, using this robot.

Basically, you have to consider the current position of the robot to have closed loop. The way it's right now, it relays on time and a very good calibration of the robot, what is not recommended if you want to have a node as generic that could be used in different robots and environments.

Don't forget that you have to add in your CMakeLists.txt and package.xml the dependency **nav_msgs** before trying to compile.

Comments

link

This answer, as it is, limited without the source. Please include the source and information from the post in this answer as links can and do go down/die.

jayess (Aug 28 '17)

Also, it is quite easy to format your code in the editor (you've done it in your other answers). Highlight your, code then press the Preformat Text (`101010`) button then your code will be properly formatted.

jayess (Aug 28 '17)

Done. I was trying to click on the code button before adding the code itself. So, the steps are: 1-Add the code, 2-Select it, 3-Code button Thanks for the tip.

marcoarruda (Aug 28 '17)

You can do that too, but from what I've experienced it doesn't work very well for anything longer than very short snippets. It's easier to format the way that you just mentioned.

jayess (Aug 28 '17)

Your code is missing the `include` s

jayess (Aug 28 '17)

Thanks, fixed! It's awkward, I tried to use "& lt;" and "& gt;" but it didn't work. Then I used "<" and ">" and worked. This problem happens if I copy and paste.

marcoarruda (Aug 29 '17)

add a comment

Your Answer

Please start posting anonymously - your entry will be published after you log in or create a new account.

Add Answer

about | faq | help | privacy policy | terms of service | give feedback
Powered by Askbot version 0.7.58