Record For Gathered Pose Data From PoseStamped

Mingze Chen

February 23, 2021

1 Gathered Data Form

A ros package called *playground* was created and inside the package a **Python** script called *pose.py*:

```
#!/usr/bin/env python
import re
import os.path
import csv
import rospy
from std_msgs.msg import String
from rosbot_ekf.msg import Imu
from nav_msgs.msg import Odometry
from geometry_msgs.msg import PoseStamped


def write_to_csv(filename, dic):
    file_exists = os.path.isfile(filename)
    # print "File exists: ", file_exists
    with open(filename, 'a') as csvfile:
        headers = ['seq', 'secs', 'nsecs', 'pos_x', 'pos_y', 'pos_z',
                   'ori_x', 'ori_y', 'ori_z', 'ori_w', 'time']
        file_is_empty = os.stat(filename).st_size == 0
        header_writer = csv.writer(csvfile, lineterminator='\n')
        writer = csv.DictWriter(csvfile,
                                delimiter=',',
                                lineterminator='\n',
                                fieldnames=headers)

        if file_is_empty:
            header_writer.writerow(headers)
            # writer.writeheader()  # file doesn't exist yet, write a header

        writer.writerow(
            {headers[i]: dic[i]
             for i in range(len(dic))})


def callback(data):
    # 'Imu' object is not iterable
    # rospy.loginfo(rospy.get_caller_id() + " Type of data: %s", type(data))
    rospy.loginfo(rospy.get_caller_id() + "␣I␣heard␣%s", data)
    data = str(data)
    # rospy.loginfo(rospy.get_caller_id() + " Index for linear acceleration is: %s", lin_acc_ind)
    res = re.findall(r"[-+]?\d*\.\d+|\d+", data)
    # res order: seq, secs, nsecs, pos_x, pos_y, pos_z, ori_x, ori_y, ori_z,
    # ori_w
   # print "res[2] = ", res[2]
    res.append(float(res[1]) + float(res[2]) * 10 ** (-9))
    # print "res: ", res
    write_to_csv('data.csv', res)


def listener():
    rospy.init_node('listener', anonymous=True)
    # rospy.Subscriber("mpu9250", Imu, callback)
    # rospy.Subscriber("odom", Odometry, callback)
    rospy.Subscriber("pose", PoseStamped, callback)
    rospy.spin()

if __name__ == '__main__':
    listener()
```

Through running the following command :

```
husarion@husarion:~/pathTo/catkin_ws$ source ./devel/setup.bash
husarion@husarion:~/pathTo/catkin_ws$ roscore
```

start a 2nd. command line window and execute:

```
$ roslaunch rosbot_ekf all.launch rosbot_pro:=true
```

start a 3rd. command line window and execute:

```
$ rosrun playground pose.py
```

From the official website of HUSARION DOCS we can know **PoseStamped** provides direct information about *Position based on encoders*.

The collected data was stored in a csv file (*poseData.csv*) and *pos_x* and *pos_y* are plotted with script *plot.py* and the running process of the robot was also dynamically illustrated, as you can see, figure 1 is a screenshot. Complete and live process can be obtained by running the python script *plot.py*.

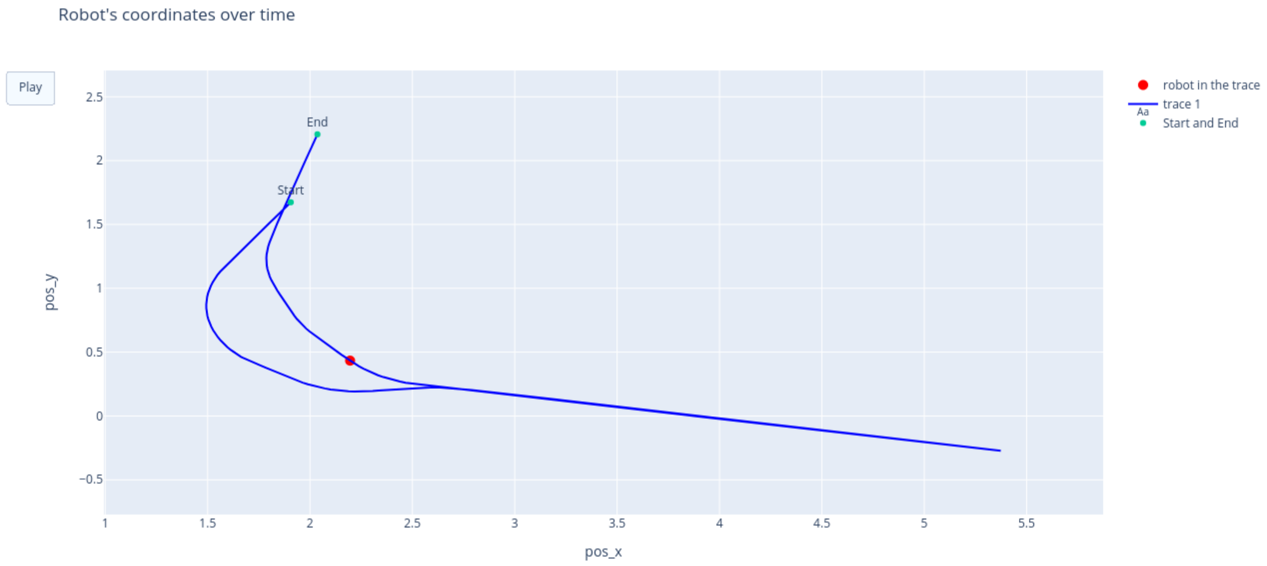

Figure 1: Plot for the running process of the robot

2 Idea for next step

In the internal coordinate system of the robot, (x = 0, y = 0) is where the robot was first turned on. However, the verification of the correctness of the data is still challenging: how correct is the acquired data?