

前文 [18.如何在实际项目中应用ROS导航相关\(1\)](#) 定点导航使用程序启动一个 `pibot_simulator`, 并且完成一个定点导航, 本文对其修改完成一个多点导航的例子

直接贴出代码 `navigation_multi_demo.launch`

```
<launch>
  <node pkg="pibot" type="navigation_multi_goals.py" respawn="false"
name="navigation_multi_goals" output="screen">
    <param name="goalListX" value="2.0, 1.0" />
    <param name="goalListY" value="2.0, 3.0" />
    <param name="goalListYaw" value="0.0, 90.0" /><!--degree-->
  </node>
</launch>
```

分别导航至[2.0, 2.0, 0]、[1.0, 3.0, 90]

[x, y, yaw] x, y为目标坐标, yaw为目标姿态yaw(角度)

`navigation_multi_goals.py`

```
#!/usr/bin/env python
from launch_demo import launch_demo
import rospy

import actionlib
from actionlib_msgs.msg import *
from move_base_msgs.msg import MoveBaseAction, MoveBaseGoal
from nav_msgs.msg import Path
from geometry_msgs.msg import PoseWithCovarianceStamped
from tf_conversions import transformations
from math import pi

class navigation_demo:
    def __init__(self):
        self.set_pose_pub = rospy.Publisher('/initialpose',
PoseWithCovarianceStamped, queue_size=5)

        self.move_base = actionlib.SimpleActionClient("move_base", MoveBaseAction)
        self.move_base.wait_for_server(rospy.Duration(60))

    def set_pose(self, p):
        if self.move_base is None:
            return False

        x, y, th = p

        pose = PoseWithCovarianceStamped()
        pose.header.stamp = rospy.Time.now()
        pose.header.frame_id = 'map'
        pose.pose.pose.position.x = x
```

```

    pose.pose.pose.position.y = y
    q = transformations.quaternion_from_euler(0.0, 0.0, th/180.0*pi)
    pose.pose.pose.orientation.x = q[0]
    pose.pose.pose.orientation.y = q[1]
    pose.pose.pose.orientation.z = q[2]
    pose.pose.pose.orientation.w = q[3]

    self.set_pose_pub.publish(pose)
    return True

def _done_cb(self, status, result):
    rospy.loginfo("navigation done! status:%d result:%s"%(status, result))

def _active_cb(self):
    rospy.loginfo("[Navi] navigation has be activated")

def _feedback_cb(self, feedback):
    rospy.loginfo("[Navi] navigation feedback\r\n%s"%feedback)

def goto(self, p):
    rospy.loginfo("[Navi] goto %s"%p)

    goal = MoveBaseGoal()

    goal.target_pose.header.frame_id = 'map'
    goal.target_pose.header.stamp = rospy.Time.now()
    goal.target_pose.pose.position.x = p[0]
    goal.target_pose.pose.position.y = p[1]
    q = transformations.quaternion_from_euler(0.0, 0.0, p[2]/180.0*pi)
    goal.target_pose.pose.orientation.x = q[0]
    goal.target_pose.pose.orientation.y = q[1]
    goal.target_pose.pose.orientation.z = q[2]
    goal.target_pose.pose.orientation.w = q[3]

    self.move_base.send_goal(goal, self._done_cb, self._active_cb,
self._feedback_cb)
    result = self.move_base.wait_for_result(rospy.Duration(60))
    if not result:
        self.move_base.cancel_goal()
        rospy.loginfo("Timed out achieving goal")
    else:
        state = self.move_base.get_state()
        if state == GoalStatus.SUCCEEDED:
            rospy.loginfo("reach goal %s succeeded!"%p)
    return True

def cancel(self):
    self.move_base.cancel_all_goals()
    return True

if __name__ == "__main__":
    rospy.init_node('navigation_demo', anonymous=True)
    goalListX = rospy.get_param('~goalListX', '2.0, 2.0')
    goalListY = rospy.get_param('~goalListY', '2.0, 4.0')

```

```
goalListYaw = rospy.get_param('~goalListYaw', '0, 90.0')

goals = [[float(x), float(y), float(yaw)] for (x, y, yaw) in
zip(goalListX.split(","),goalListY.split(","),goalListYaw.split(","))]
navi = navigation_demo()

r = rospy.Rate(1)
r.sleep()

for goal in goals:
    navi.goto(goal)

while not rospy.is_shutdown():
    r.sleep()
```