

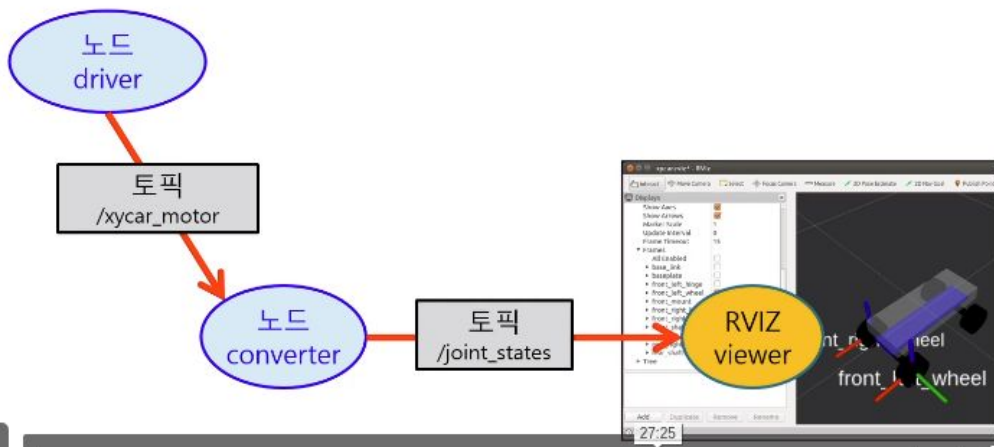
3D 자동차 제어 프로그래밍 과제

프로그래머스 자율주행 코스 1기 조정민

rviz 자동차에 8자 주행 패키지 변환 및 퍼블리시

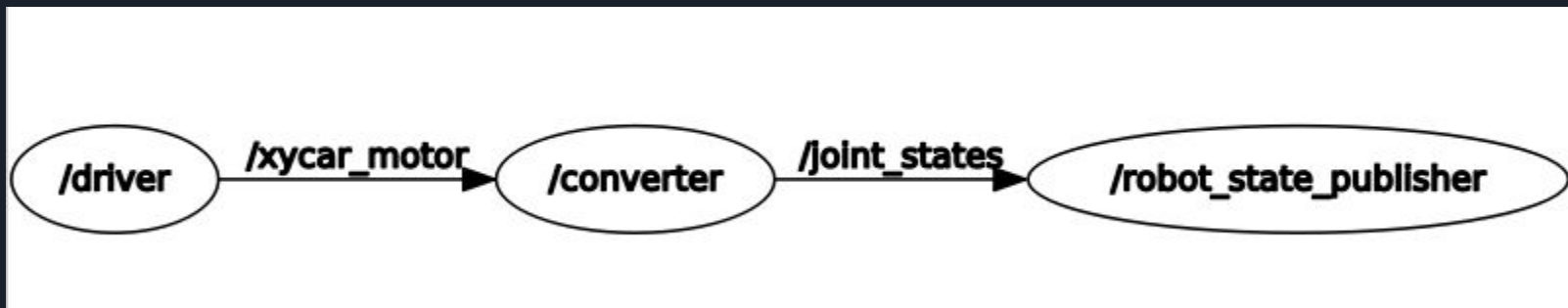
모터제어 토픽을 발행해서 RVIZ 바퀴를 움직이기

- Xycar의 속도와 조향각을 제어하는 메시지를
 - converter 노드가 받아서 포맷을 변경한 후에 RVIZ로 보내면
 - RVIZ상에서 자동차 바퀴가 움직인다



해결 방식

- xycar_motor 토픽을 받는 서브스크라이버 콜백함수에서
 - JointState 토픽 생성
 - xycar_motor.angle 값을 JointState 구조체에 저장
 - 생성한 토픽 퍼블리쉬





구분

```
8 a = -3.14
9 b = -3.14
10
11 def callback(mt_data):
12     global a
13     global b
14     wheel_msg = JointState()
15     wheel_msg.header = Header()
16     wheel_msg.name = ['front_right hinge_joint', 'front_left hinge_joint',
17                       'front_right_wheel_joint', 'front_left_wheel_joint',
18                       'rear_right_wheel_joint', 'rear_left_wheel_joint']
19     wheel_msg.velocity = []
20     wheel_msg.effort = []
21     wheel_msg.header.stamp = rospy.Time.now()
22
23     if a >= 3.14:
24         a = -3.14
25         b = -3.14
26     else:
27         # 0.01 라디안은 약 6도
28         a += 0.01
29         b += 0.01
30
31     # front_*_wheel_joint 만 회전 시키기 위한 것
32     wheel_msg.position = [mt_data.angle / 50, mt_data.angle / 50, a, b, 0, 0]
33     pub.publish(wheel_msg)
34
35 motor_control = xycar_motor()
36 rospy.init_node('converter')
37 sub = rospy.Subscriber('xycar_motor', xycar_motor, callback)
38 pub = rospy.Publisher('joint_states', JointState, queue_size=10)
39 rospy.spin()
```



실행 방법

슬라이더로 바퀴 조절

```
roslaunch rviz_xycar xycar_3d.launch
```

앞바퀴 뱅글뱅글 돌기

```
roslaunch rviz_xycar move_joint.launch
```

과제 실행

```
roslaunch rviz_xycar rviz_drive.launch
```

실행화면

