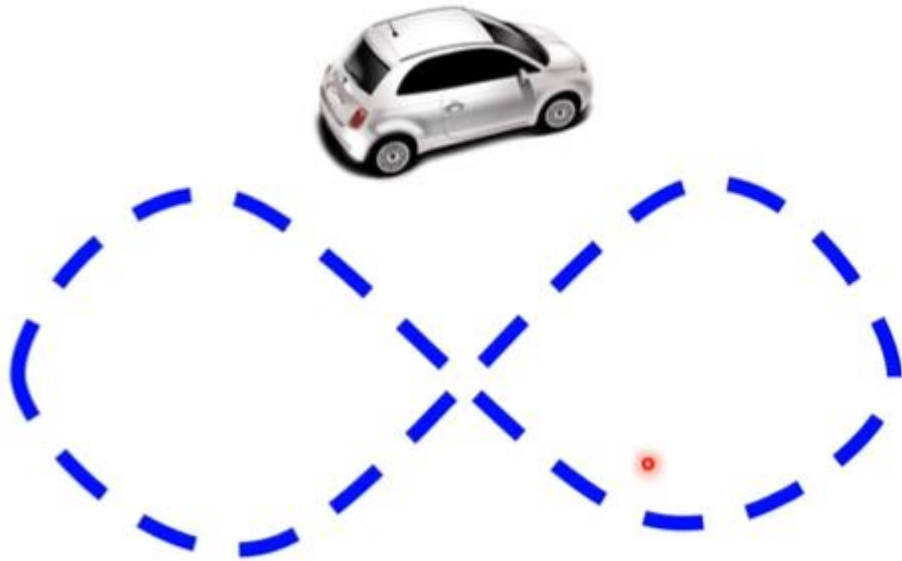


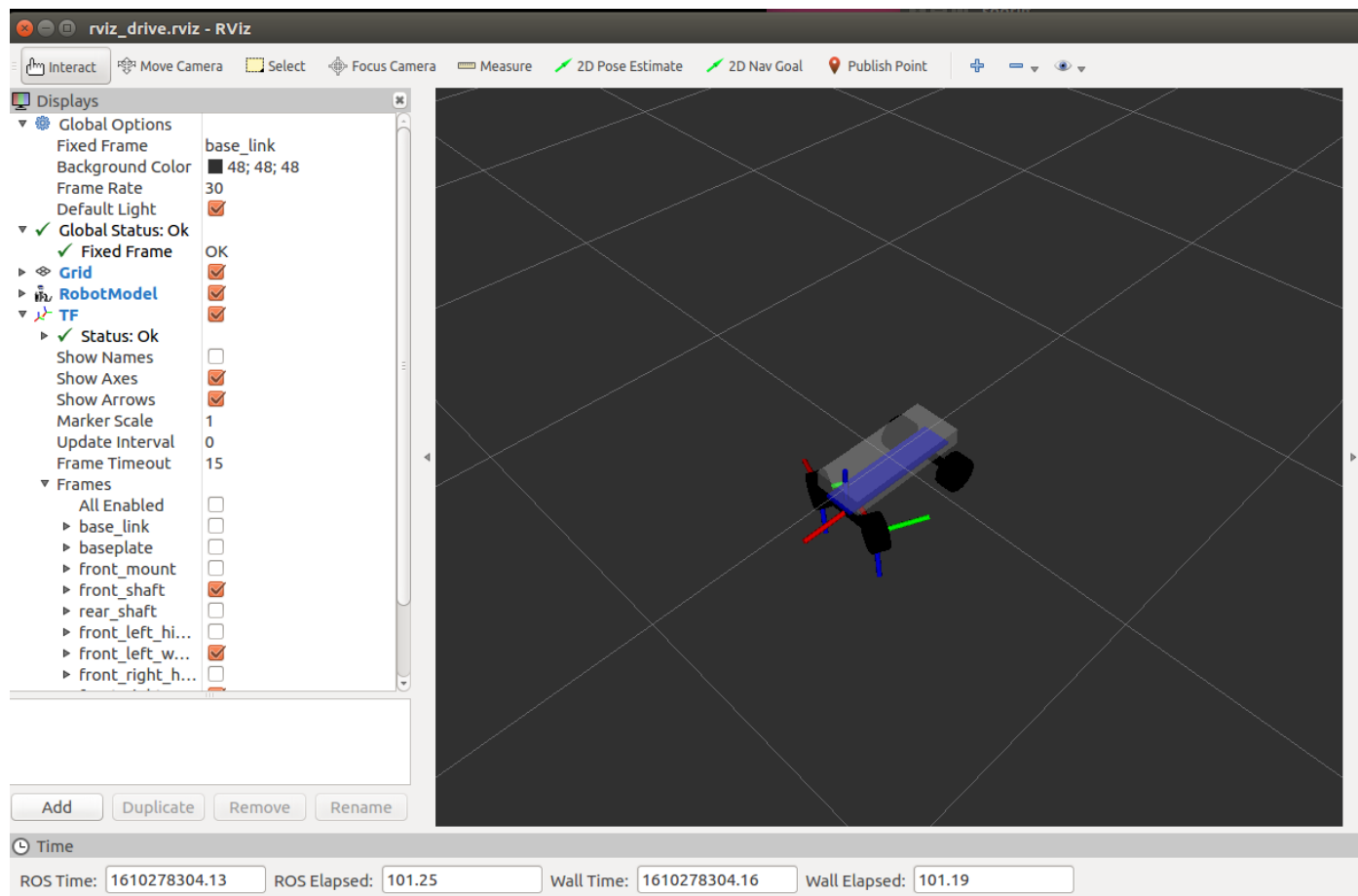
3D 자동차 제어프로그래밍

과제 설명

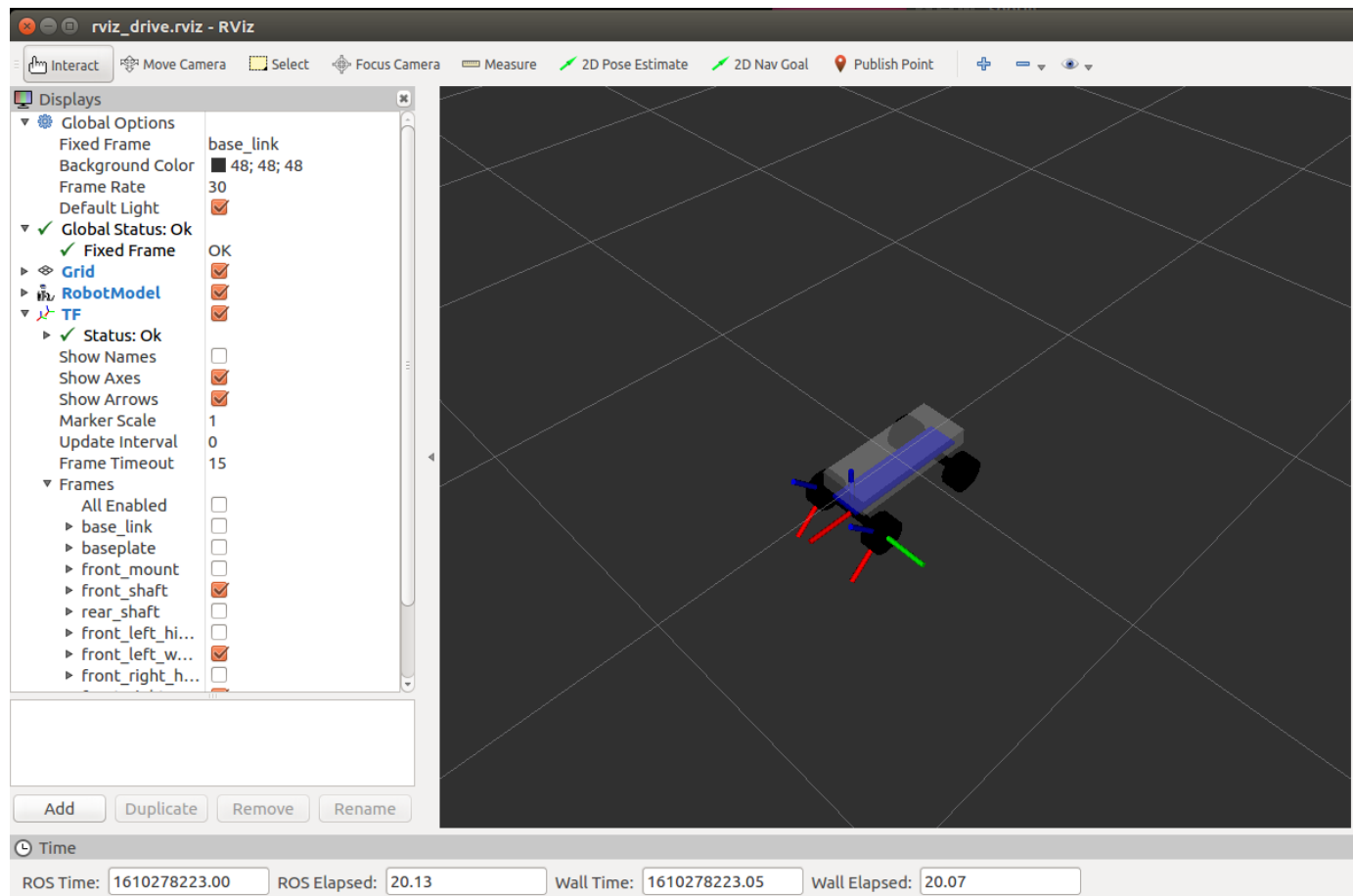
- RVIZ에서 3D자동차 8자 주행 시키기
 - 실제 이동없이 바퀴만 움직이기



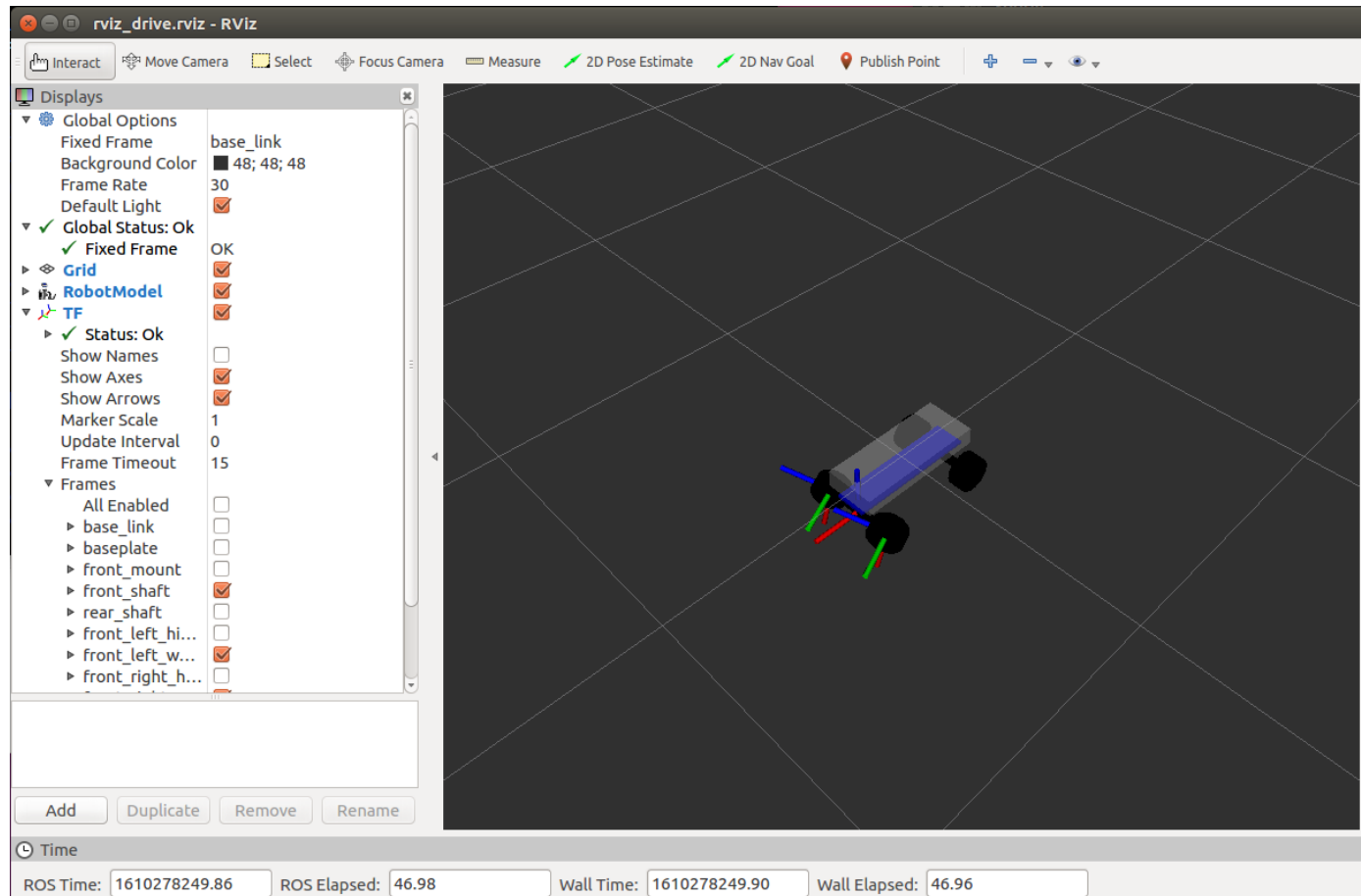
현상 확인 #1



현상 확인 #2



현상 확인 #3



코드 설명 #1(rviz_8_drive.py)

- `motor_control = xycar_motor()`
- `def motor_pub(angle, speed):` *#angle, speed 값을 받기*
 `global pub`
 `global motor_control`

 `motor_control.angle = angle`
 `motor_control.speed = speed`

 `pub.publish(motor_control)` *#받은 값을 xycar_motor()타입으로 맞추어 저장 후 전송*

코드 설명 #2(rviz_8_drive.py)

```
while not rospy.is_shutdown():  
    angle = -50  
    for i in range(60):  
        motor_pub(angle, speed)  
        time.sleep(0.1)
```

종료 전까지 실행

-50 각도로 동그라미 그리기

```
    angle = 0  
    for i in range(35):  
        motor_pub(angle, speed)  
        time.sleep(0.1)
```

각도 0으로 잠깐 정지

```
    angle = 50  
    for i in range(60):  
        motor_pub(angle, speed)  
        time.sleep(0.1)
```

50 각도로(반대) 동그라미 그리기

```
    angle = 0  
    for i in range(35):  
        motor_pub(angle, speed)  
        time.sleep(0.1)
```

각도 0으로 잠깐 정지

코드 설명 #1(convert.py)

```
def callback(data):  
    global msg_joint_states, l_wheel, r_wheel, pub  
    Angle = data.angle  
    msg_joint_states.header.stamp = rospy.Time.now()
```

```
    steering = math.radians(Angle * -0.4)
```

Xycar angle = -50~50 -> -20~ 20도로 변환

```
    if l_wheel > 3.14:  
        l_wheel = -3.14  
        r_wheel = -3.14  
    else:  
        l_wheel += 0.01  
        r_wheel += 0.01
```

3.14이면 한바퀴 완성
-3.14부터 다시 회전

```
    msg_joint_states.position = [steering, steering, r_wheel, l_wheel,  
                                r_wheel, l_wheel]  
    pub.publish(msg_joint_states)
```

해당 값을 positio에 채워서 publis하기(바퀴 회전)

코드 설명 #2(convert.py)

```
def start():  
    global msg_joint_states, l_wheel, r_wheel, pub  
    rospy.init_node('converter')  
    pub = rospy.Publisher('joint_states', JointState, queue_size=10)  
  
    msg_joint_states = JointState()  
    msg_joint_states.header = Header()  
    msg_joint_states.name = ['front_right_hinge_joint',  
                             'front_left_hinge_joint',  
                             'front_right_wheel_joint',  
                             'front_left_wheel_joint',  
                             'rear_right_wheel_joint',  
                             'rear_left_wheel_joint']  
  
    msg_joint_states.velocity = []  
    msg_joint_states.effort = []  
    l_wheel = -3.14  
    r_wheel = -3.14  
    rospy.Subscriber("xycar_motor", xycar_motor, callback)  
    rospy.spin()
```

Joint_states로 토픽 발행

Xycar_motor 토픽 구독, callback함수 호출