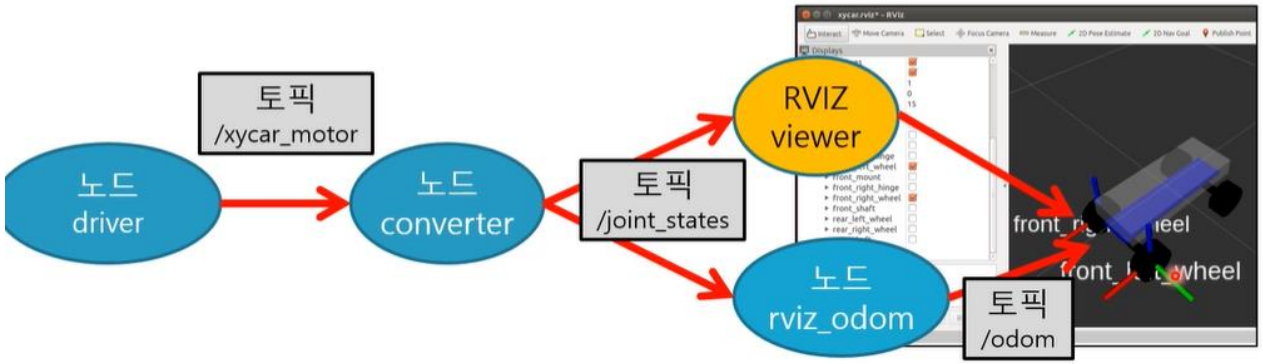
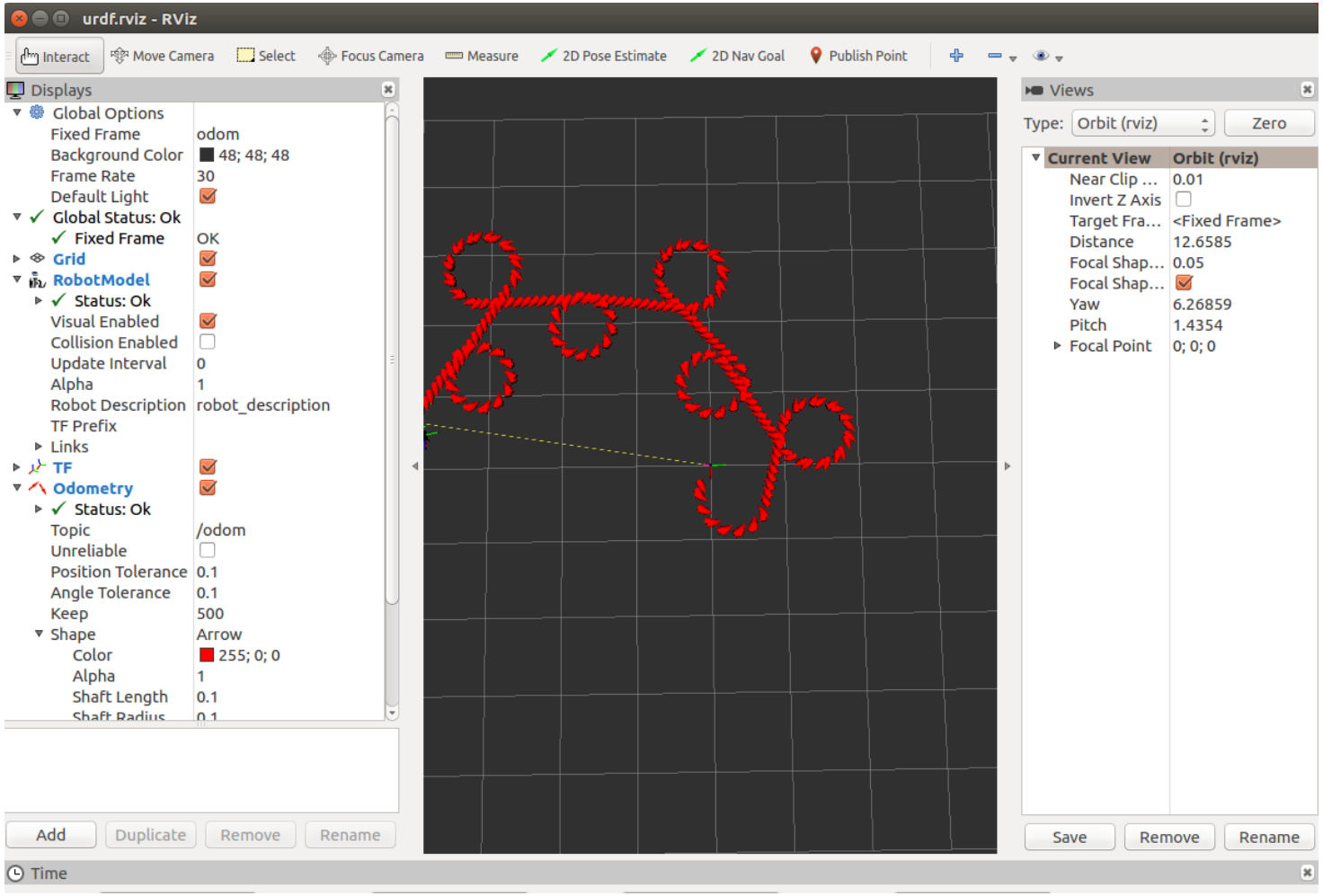
RVIZ 3D자동차 주행프로그래밍

과제 설명

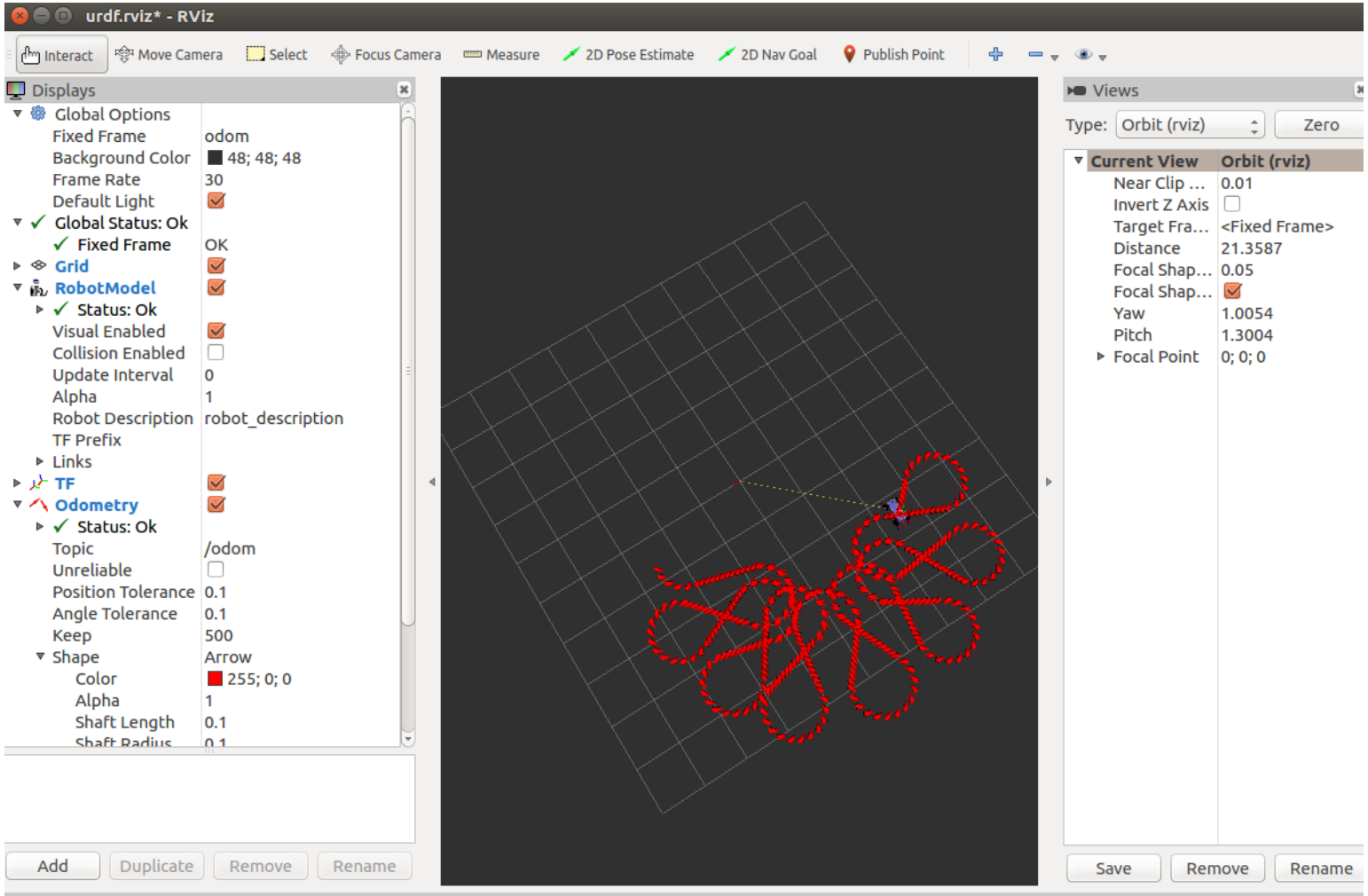
- RVIZ 가상 공간에 있는 3D자동차를 주행 시키기
  - 8자 주행 프로그램이 모터 제어 메시지를 보내면(/xycar_motor토픽)
  - 그것을 받아 converter에서 변환 후 /joint_states로 토픽 발행
  - /joint_states를 다시 오도메트리에서 받아 변환하여 /odom 토픽 발행



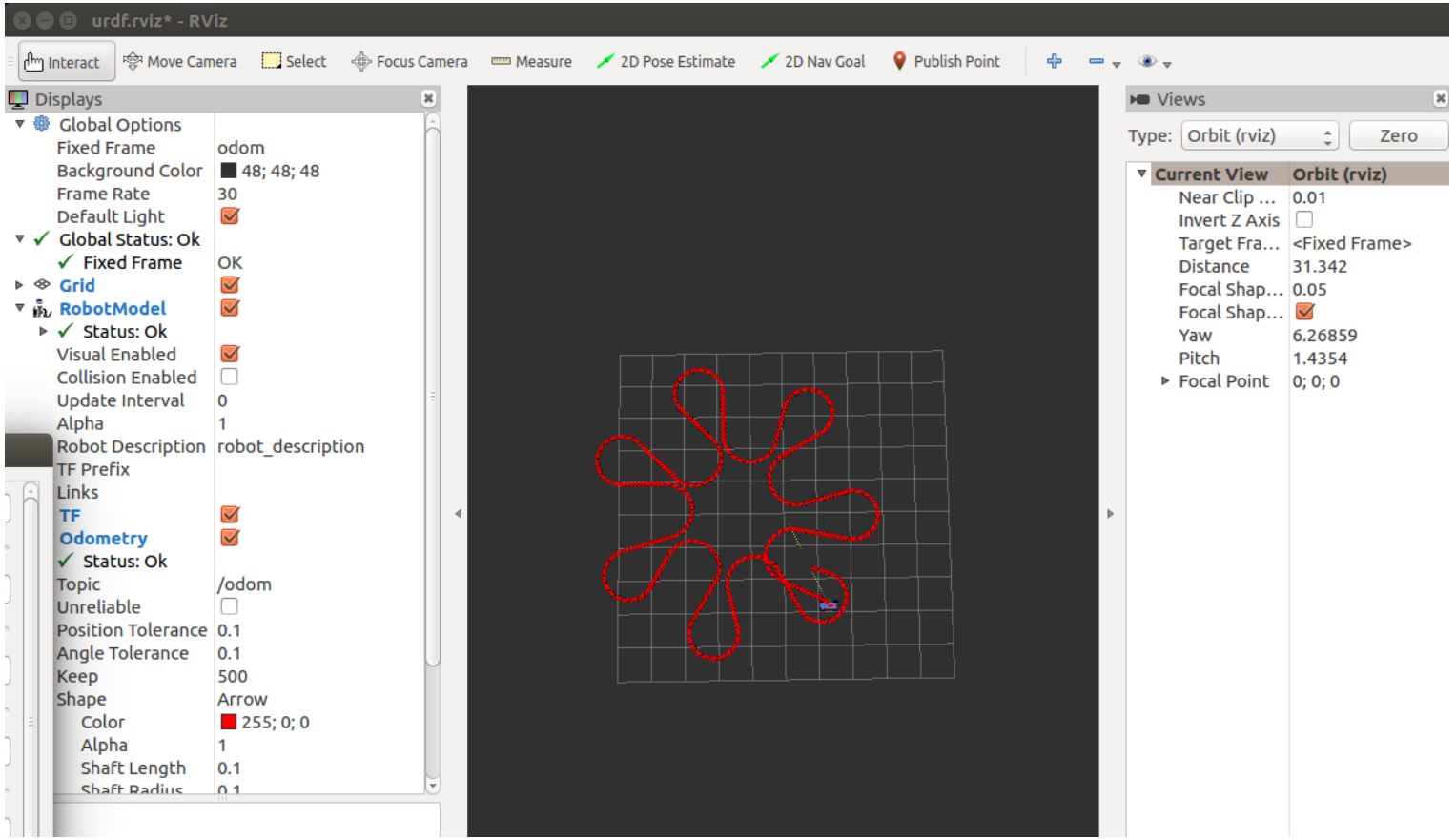
현상 확인 #1


urdf.rviz - RViz
Interact
Move Camera
Select
Focus Camera
Measure
2D Pose Estimate
2D Nav Goal
Publish Point
Displays
Global Options
Fixed Frame odom
Background Color 48; 48; 48
Frame Rate 30
Default Light
Global Status: Ok
Fixed Frame OK
Grid
RobotModel
Status: Ok
Visual Enabled
Collision Enabled
Update Interval 0
Alpha 1
Robot Description robot_description
TF Prefix
Links
TF
Odometry
Status: Ok
Topic /odom
Unreliable
Position Tolerance 0.1
Angle Tolerance 0.1
Keep 500
Shape Arrow
Color 255; 0; 0
Alpha 1
Shaft Length 0.1
Add
Duplicate
Remove
Rename
Views
Type: Orbit (rviz)
Zero
Current View Orbit (rviz)
Near Clip ... 0.01
Invert Z Axis
Target Fra... <Fixed Frame>
Distance 12.6585
Focal Shap... 0.05
Focal Shap...
Yaw 6.26859
Pitch 1.4354
Focal Point 0; 0; 0
Save
Remove
Rename
Time

현상 확인 #2


urdf.rviz* - RViz
Interact
Move Camera
Select
Focus Camera
Measure
2D Pose Estimate
2D Nav Goal
Publish Point
Displays
Global Options
Fixed Frame odom
Background Color 48; 48; 48
Frame Rate 30
Default Light
Global Status: Ok
Fixed Frame OK
Grid
RobotModel
Status: Ok
Visual Enabled
Collision Enabled
Update Interval 0
Alpha 1
Robot Description robot_description
TF Prefix
Links
TF
Odometry
Status: Ok
Topic /odom
Unreliable
Position Tolerance 0.1
Angle Tolerance 0.1
Keep 500
Shape Arrow
Color 255; 0; 0
Alpha 1
Shaft Length 0.1
Add
Duplicate
Remove
Rename
Views
Type: Orbit (rviz)
Zero
Current View Orbit (rviz)
Near Clip ... 0.01
Invert Z Axis
Target Fra... <Fixed Frame>
Distance 21.3587
Focal Shap... 0.05
Focal Shap...
Yaw 1.0054
Pitch 1.3004
Focal Point 0; 0; 0
Save
Remove
Rename

현상 확인 #3


urdf.rviz* - RViz
Interact
Move Camera
Select
Focus Camera
Measure
2D Pose Estimate
2D Nav Goal
Publish Point
Displays
Global Options
Fixed Frame odom
Background Color 48; 48; 48
Frame Rate 30
Default Light
Global Status: Ok
Fixed Frame OK
Grid
RobotModel
Status: Ok
Visual Enabled
Collision Enabled
Update Interval 0
Alpha 1
Robot Description robot_description
TF Prefix
Links
TF
Odometry
Status: Ok
Topic /odom
Unreliable
Position Tolerance 0.1
Angle Tolerance 0.1
Keep 500
Shape Arrow
Color 255; 0; 0
Alpha 1
Shaft Length 0.1
Views
Type: Orbit (rviz)
Zero
Current View Orbit (rviz)
Near Clip ... 0.01
Invert Z Axis
Target Fra... <Fixed Frame>
Distance 31.342
Focal Shap... 0.05
Focal Shap...
Yaw 6.26859
Pitch 1.4354
Focal Point 0; 0; 0

코드 설명 #1 (rviz_odom.py)

```
global Angle

def callback(msg):
        global Angle
        Angle = msg.position[msg.name.index("front_left_hinge_joint")]

rospy.Subscriber('joint_states', JointState, callback)
```

Joint_states토픽 구독하면 해당 데이터에서 angle값 추출

코드 설명 #2 (rviz_odom.py)

```
odom_pub = rospy.Publisher("odom", Odometry, queue_size=50)
odom_broadcaster = tf.TransformBroadcaster()

current_time = rospy.Time.now()
last_time = rospy.Time.now()

r = rospy.Rate(30.0)

current_speed = 0.4
wheel_base = 0.2
x_ = 0
y_ = 0
yaw_ = 0

Angle = 0
```

Odom 토픽 발행 준비

1초에 30번 반복

속도는 초속 40cm, 축간 거리는 20cm(앞 바퀴와 뒷바퀴 중심 간 거리

코드 설명 #3 (rviz_odom.py)

```
while not rospy.is_shutdown():
    current_time = rospy.Time.now()

    # compute odometry in a typical way given
    dt = (current_time - last_time).to_sec()

    current_steering_angle = Angle
    current_angular_velocity = current_speed * math.tan(current_steering_angle) / wheel_base

    x_dot = current_speed * cos(yaw_)
    y_dot = current_speed * sin(yaw_)
    x_ += x_dot * dt;
    y_ += y_dot * dt;
    yaw_ += current_angular_velocity * dt

    # since all odometry is 6DOF we'll need a quaternion created fro
    odom_quat = tf.transformations.quaternion_from_euler(0, 0, yaw_)

    # first, we'll publish the transform over tf
    odom_broadcaster.sendTransform(
        (x_, y_, 0.),
        odom_quat,
        current_time,
        "base_link",
        "odom"
    )
```

이전 계산 이후 경과 시간

각속도 계산

이동 계산

오일러 값을 쿼터니언 값으로 변환

위치 정보에 대한 발행 준비, odom과 base_link 연결

코드 설명 #4 (rviz_odom.py)

```
# next, we'll publish the odometry message over ROS
odom = Odometry()
odom.header.stamp = current_time
odom.header.frame_id = "odom"

# set the position
odom.pose.pose = Pose(Point(x_, y_, 0.), Quaternion(*odom_quat))

# set the velocity
odom.child_frame_id = "base_link"
#odom.twist.twist = Twist(Vector3(vx, vy, 0), Vector3(0, 0, yaw_))

# publish the message
odom_pub.publish(odom)

last_time = current_time
r.sleep()
```

Odometry 메시지 헤더 만들기

Position 값 채우기

속도 값 채우기

/Odom 토픽 발행

시간 값 변경