Inverse Kinematics

-Computer Problem Set 3-

2018. 05. 25.

미래융합기술학과

20187087 조은기

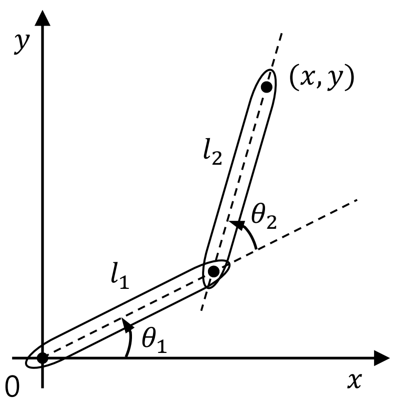
■ Content

1. Introduction
   ① 프로젝트 목적
   ② 사용 Tool 및 라이브러리
2. Algorithm
   ① Manipulator 설계
   ② Inverse Kinematics 수식 유도
3. Demo
4. Source Analysis
5. Impression

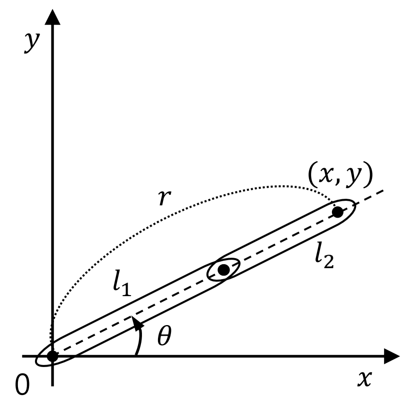
1. Introduction

① 프로젝트 목적

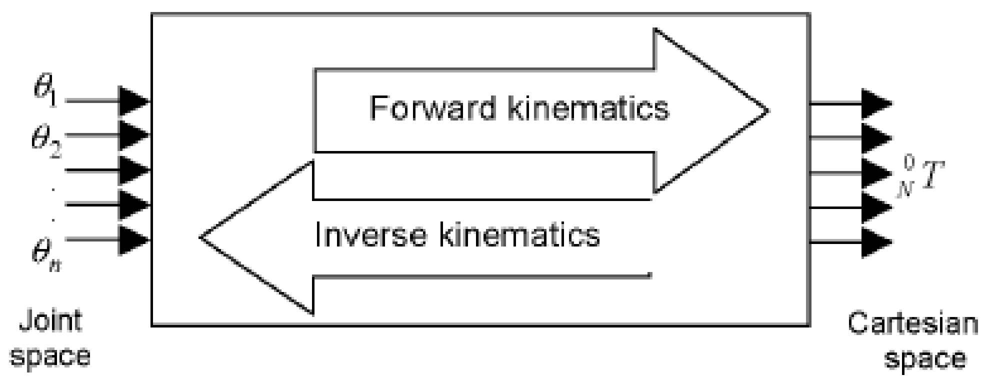
- Manipulator 설계와 Kinematics를 이용한 Control simulation 구현
- 각 Planar Robot에 대한 Inverse Kinematics의 이해



〈 $\theta_1 - \theta_2$ Planar Robot 〉



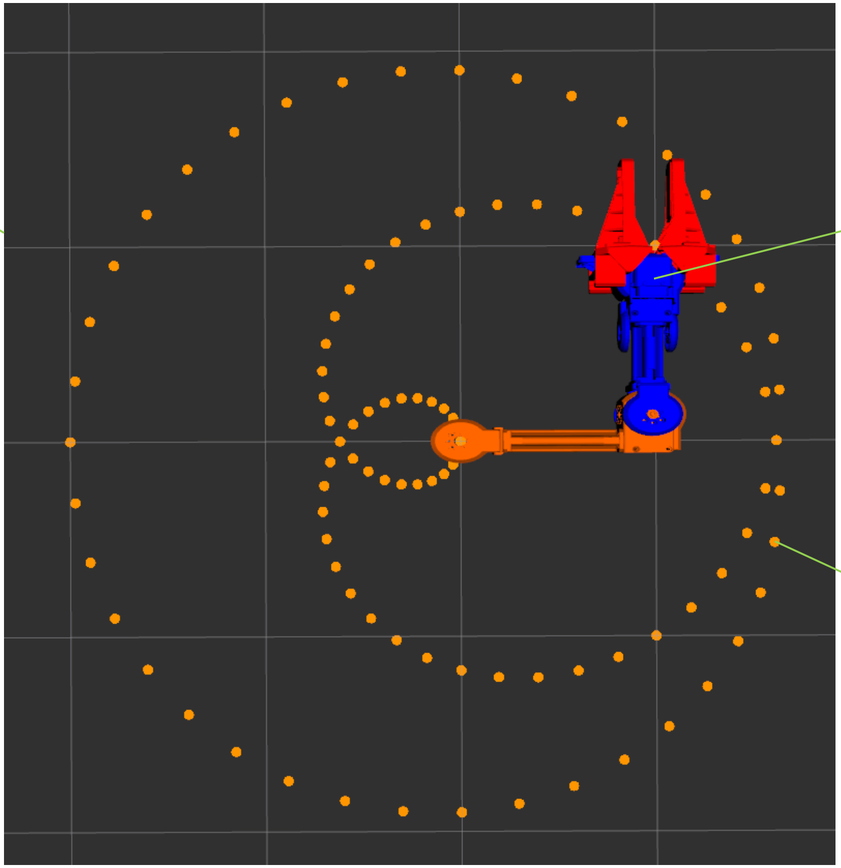
〈 $\theta - r$ Planar Robot 〉



1. Introduction

② 사용 Tool 및 라이브러리

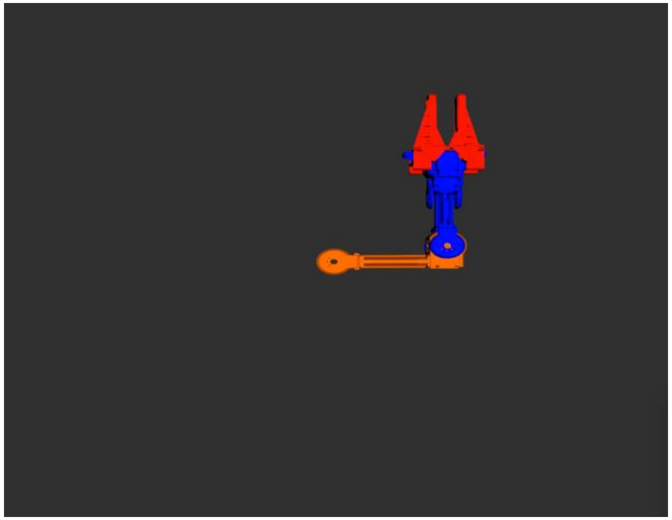
- Ubuntu
- ROS (Robot Operating System)
- GCC Compiler
- PCL (Point Cloud Library)



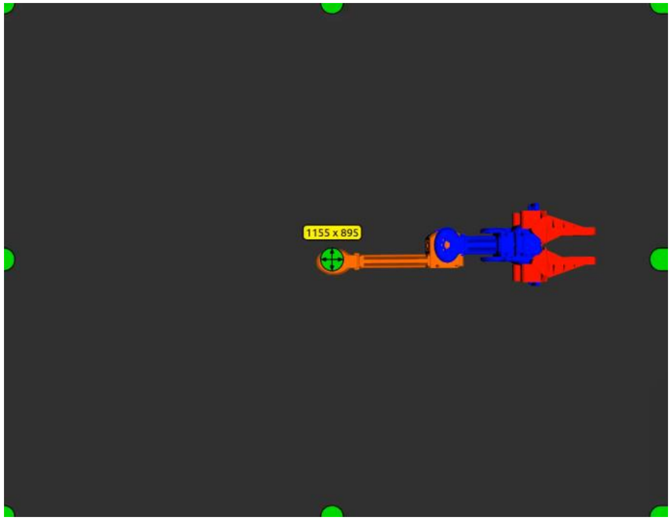
2. Algorithm

① Manipulator 설계

- ROS URDF로 Modeling file을 불러와 2 DOF Manipulator 설계
- 1축 Revolute joint, 2축 Revolute joint로 구성된 $\theta_1 - \theta_2$ Planar Robot
- 1축 Revolute joint, 2축 Prismatic joint로 구성된 $\theta - r$ Planar Robot



< $\theta_1 - \theta_2$ Planar Robot >



< $\theta - r$ Planar Robot >

2. Algorithm

① Manipulator 설계

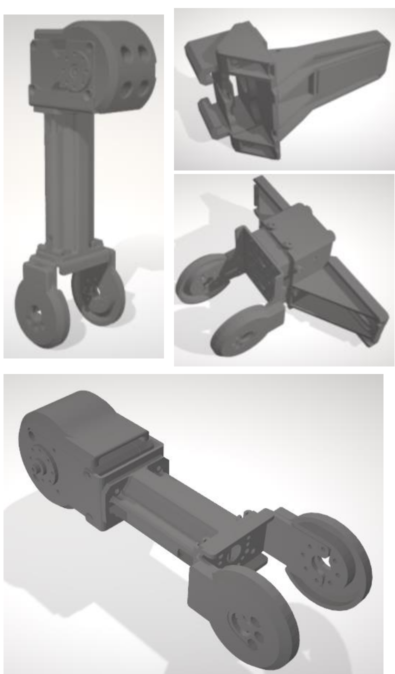
Manipulator 3D Modeling
(create .stl files)

→

Joint / Link Design
(ROS URDF)

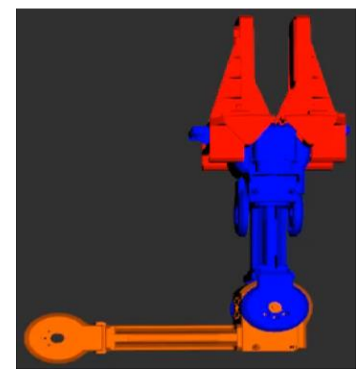
→

Visualization
(ROS Rviz)



```xml
<link name="pan_link">
    <visual>
        <geometry>
            <cylinder length="0.4" radius="0.04"/>
        </geometry>
        <origin rpy="0 0 0" xyz="0 0 0.09"/>
        <material name="red">
            <color rgba="0 0 1 1"/>
        </material>
    </visual>

    <collision>
        <geometry>
            <cylinder length="0.4" radius="0.06"/>
        </geometry>
        <origin rpy="0 0 0" xyz="0 0 0.09"/>
    </collision>

    <inertial>
        <mass value="1"/>
        <inertia ixx="1.0" ixy="0.0" ixz="0.0" iyy="1.0" iyz="0.0" izz="1.0"/>
    </inertial>
</link>

<joint name="tilt_joint" type="revolute">
```

```xml
<joint name="tilt_joint" type="revolute">
    <parent link="pan_link"/>
    <child link="tilt_link"/>
    <origin xyz="0 0 0.2"/>
    <axis xyz="0 1 0" />
    <limit effort="300" velocity="0.1" lower="-4.71239" upper="-1.570796"/>
    <dynamics damping="50" friction="1"/>
</joint>
```



2. Algorithm

① Manipulator 설계

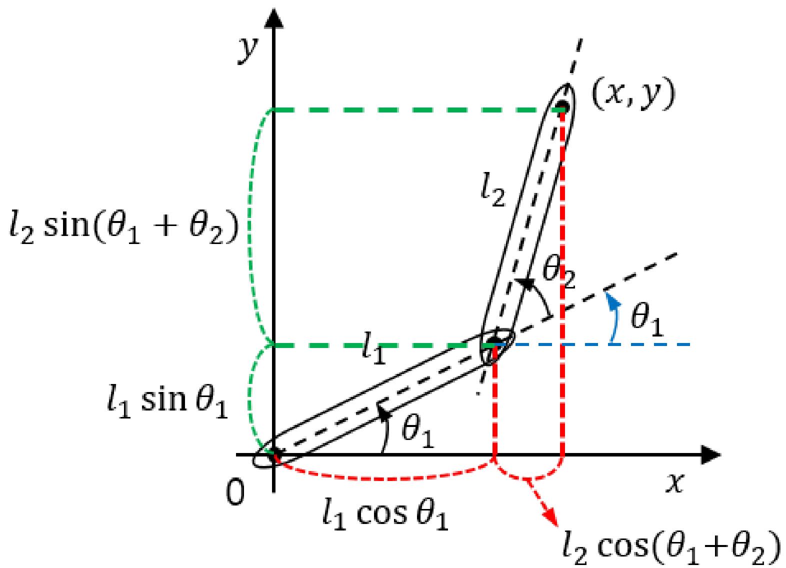
< System Architecture >

- $\theta_1 - \theta_2$ Planar Robot

- $\theta - r$ Planar Robot

2. Algorithm

② Inverse Kinematics 수식 유도 ($\theta_1 - \theta_2$ Planar Robot)



$$P_x = l_1 \cos \theta_1 + l_2 \cos(\theta_1 + \theta_2)$$
$$P_y = l_1 \sin \theta_1 + l_2 \sin(\theta_1 + \theta_2)$$

$$x^2 + y^2 = l_1^2 + l_2^2 + 2l_1 l_2(c_1 c_{12} + s_1 s_{12}) = l_1^2 + l_2^2 + 2l_1 l_2 c_2$$

$$\cos \theta_2 = \frac{x^2 + y^2 - (l_1^2 + l_2^2)}{2l_1 l_2}$$

$$\sin\theta_2 = \pm\sqrt{1 - \cos^2\theta_2}$$

$$\boxed{\theta_2 = \operatorname{atan2}(\sin\theta_2, \cos\theta_2)}$$

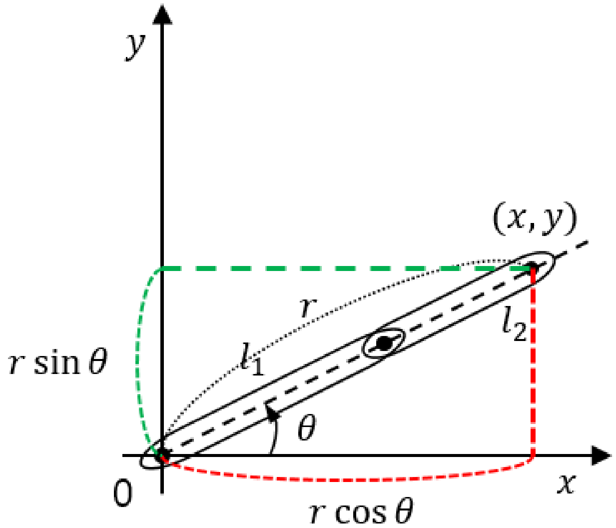
$$\cos \theta_1 = \frac{(l_1 + l_2 c_2)x + l_2 s_2 y}{(l_1 + l_2 c_2)^2 + (l_2 s_2)^2}$$

$$\sin\theta_1 = \frac{-l_2 s_2 x + (l_1 + l_2 c_2)y}{(l_1 + l_2 c_2)^2 + (l_2 s_2)^2}$$

$$\boxed{\theta_1 = \operatorname{atan2}(\sin\theta_1, \cos\theta_1)}$$

2. Algorithm

② **Inverse Kinematics 수식 유도** ($\theta - r$ Planar Robot)



$$P_x = r\cos\theta$$

$$P_y = r\sin\theta$$

$$x^2 + y^2 = r^2c^2 + r^2s^2 = r^2(c^2 + s^2) = r^2$$

$$\boxed{r = \sqrt{x^2 + y^2}}$$

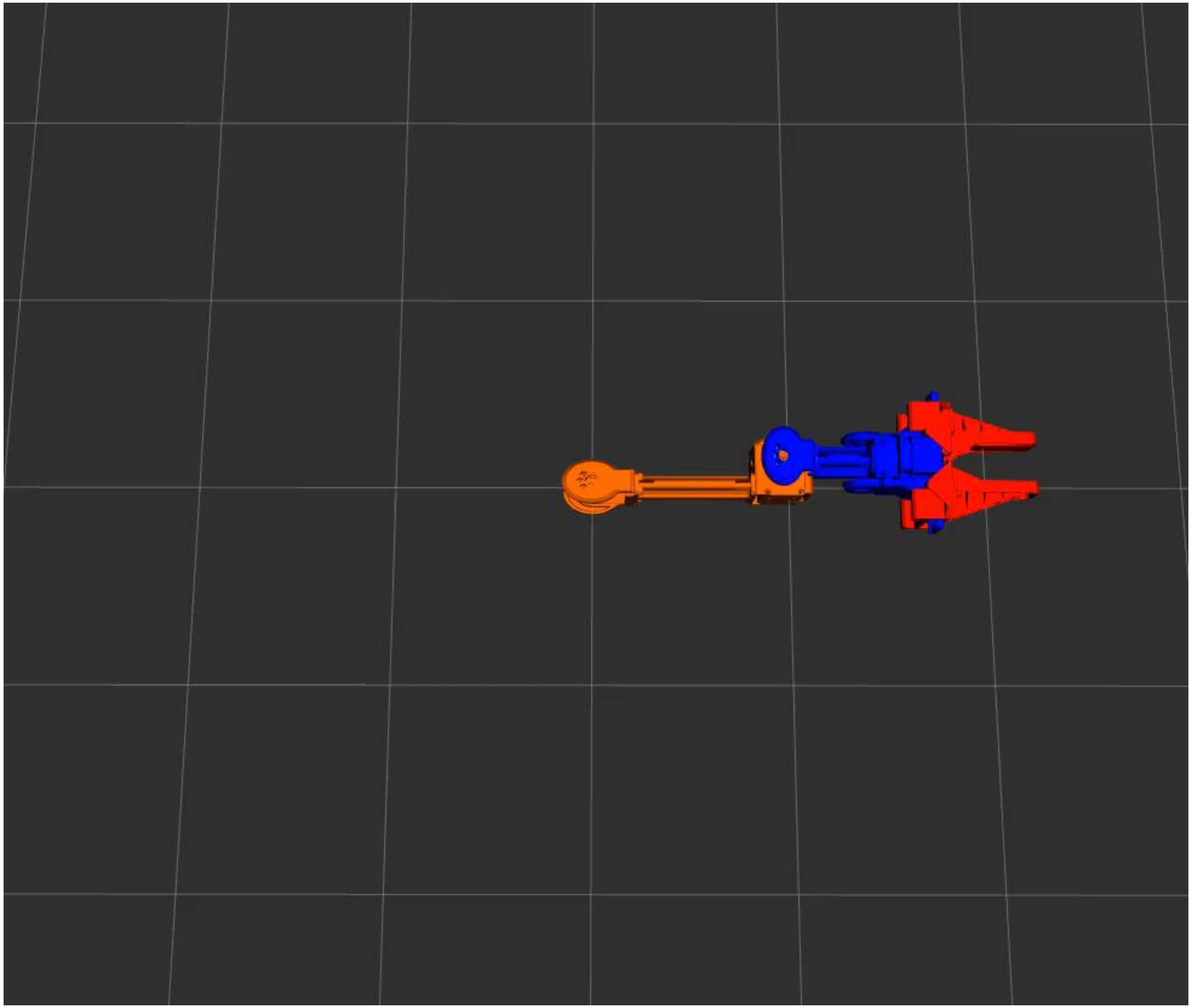
$$\cos\theta = \frac{x}{r} \qquad \sin\theta = \frac{y}{r}$$

$$\boxed{\theta = \mathrm{atan2}(\sin\theta, \cos\theta)}$$

3. Demo

① $\theta_1 - \theta_2$

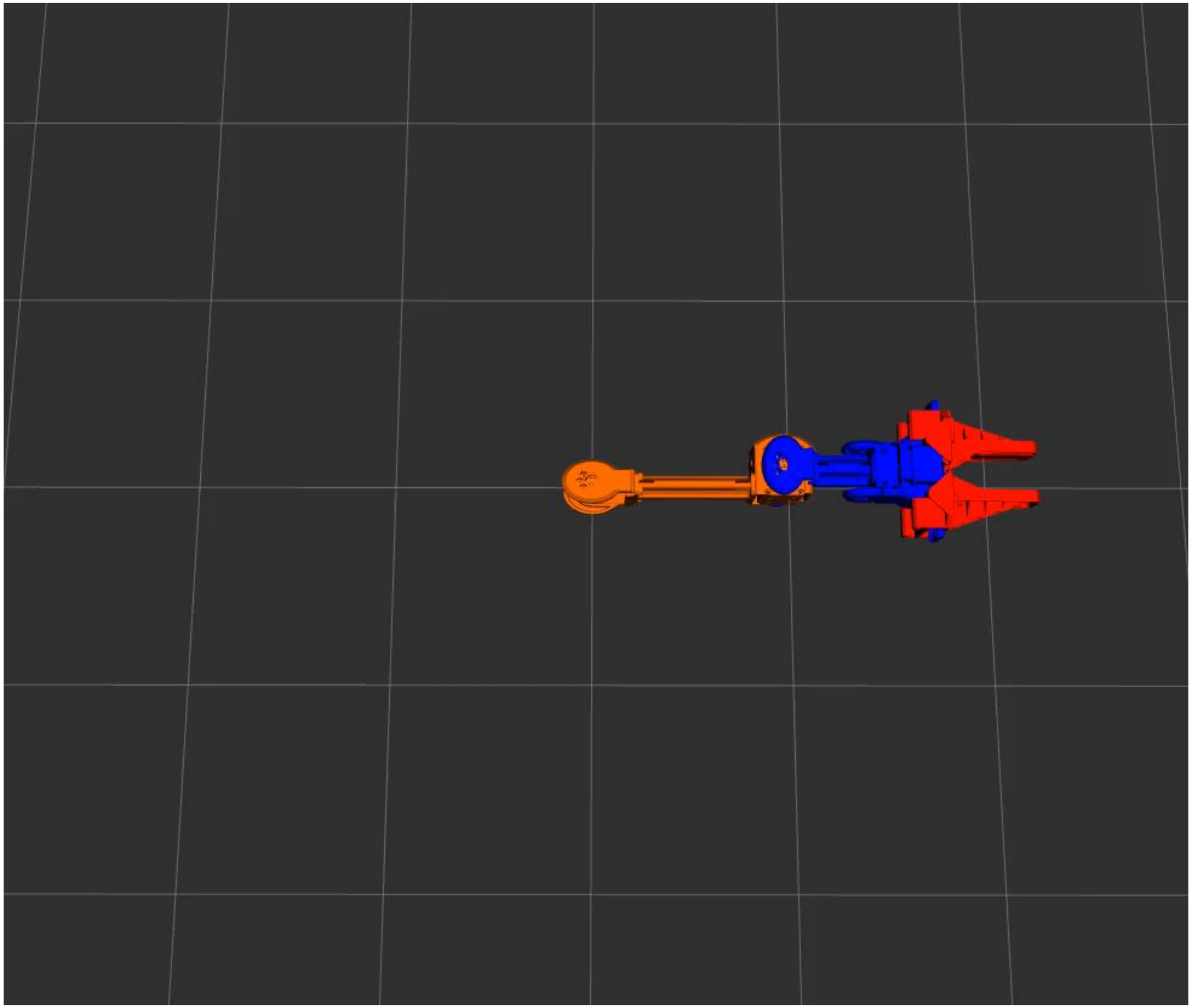
Planar Robot



3. Demo

② $\theta - r$

Planar Robot



4. Source Analysis

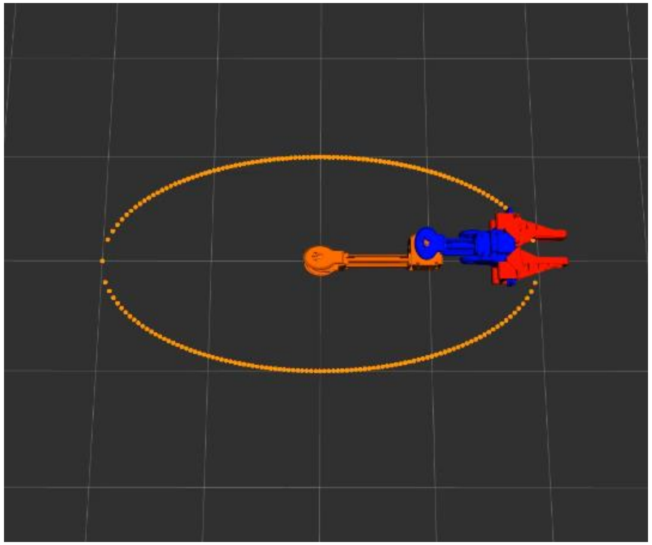
① $\theta_1 - \theta_2$ Planar Robot

```
if(flag == 0) {      // 타원 궤적 그리기
    // (x * x) / (a * a) + (y * y) / (b * b)        // a = 2, b = 1
    float x_ = 2.00 - time2 * 0.04;
    float a = 2, b = 1;

    x[j] = x_;  // 궤적 공식
    float y_ = (float)sqrt(1 - pow(x_, 2) / pow(a, 2));

    if(time_ < 1.00 - 0.01)
        y[j] = y_;
    else
        y[j] = -1 * y_;

    //if(j % 2 == 0){
        pt.x = x[j], pt.y = y[j], pt.z = 0;
        pt.r = 255, pt.g = 150, pt.b = 0;
        position_cluster1.push_back(pt);
    //}

    if(time_ < 2.00 - 0.01) {
        j++;
        time_ += 0.01;
        if(time_ < 1.00) time2++;
        else time2--;
    }
    else { flag++; time_ = 0; j = 0; }
}
```

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1$$

$$total\ time = 2\ second$$

$$sequence = 0.01$$



4. Source Analysis

① $\theta_1 - \theta_2$ Planar Robot

$$\cos\theta_2 = \frac{x^2 + y^2 - (l_1^2 + l_2^2)}{2 l_1 l_2}$$

$$\sin\theta_2 = \pm\sqrt{1 - \cos^2\theta_2}$$

$$\theta_2 = \mathrm{atan2}(\sin\theta_2, \cos\theta_2)$$

```
else if(flag == 1) {          // 상박 제어
    float c2 = (pow(x[j], 2) + pow(y[j], 2) - 2) / 2;
    float s2 = sqrt(1 - pow(c2, 2));
    float theta2 = atan2(s2, c2);

    float c1 = ((1 + c2) * x[j] + s2 * y[j]) / (pow((1 + c2), 2) + pow(s2, 2));
    float s1 = (-1 * s2 * x[j] + (1 + c2) * y[j]) / (pow((1 + c2), 2) + pow(s2, 2));
    joint_position[0] = atan2(s1, c1);

    pcl::PointCloud<pcl::PointXYZRGB> position_cluster;
    position_cluster.header.frame_id = "base";

    if(time_ < 2.00 - 0.01) {
        j++;
        time_ += 0.01;
    }
    else { flag++; time_ = 0; j = 0; }
}
```

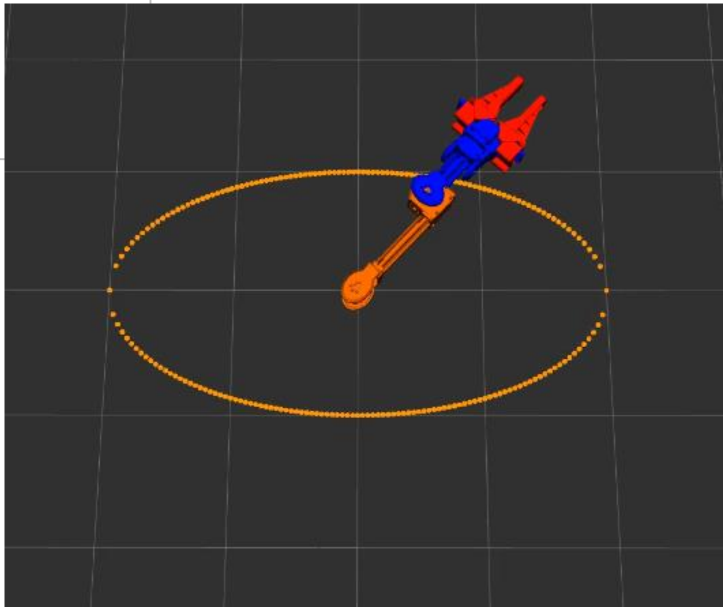
$$total\ time = 2\ second$$

$$sequence = 0.01$$

$$\cos\theta_1 = \frac{(l_1 + l_2 c_2)x + l_2 s_2 y}{(l_1 + l_2 c_2)^2 + (l_2 s_2)^2}$$

$$\sin\theta_1 = \frac{-l_2 s_2 x + (l_1 + l_2 c_2)y}{(l_1 + l_2 c_2)^2 + (l_2 s_2)^2}$$

$$\theta_1 = \mathrm{atan2}(\sin\theta_1, \cos\theta_1)$$



4. Source Analysis

① $\theta_1 - \theta_2$ Planar Robot

```
else if(flag == 2) {          // 상박+하박 제어
    float c2 = (pow(x[j], 2) + pow(y[j], 2) - 2) / 2;
    float s2 = sqrt(1 - pow(c2, 2));
    joint_position[1] = atan2(s2, c2);

    float c1 = ((1 + c2) * x[j] + s2 * y[j]) / (pow((1 + c2), 2) + pow(s2, 2));
    float s1 = (-1 * s2 * x[j] + (1 + c2) * y[j]) / (pow((1 + c2), 2) + pow(s2, 2));
    joint_position[0] = atan2(s1, c1);

    x[j] = ((l*cos(joint_position[0])) + (l*cos(joint_position[0] + joint_position[1])));
    y[j] = ((l*sin(joint_position[0])) + (l*sin(joint_position[0] + joint_position[1])));

    //if(j % 2 == 0) {        // 궤적 2
        pt.x = x[j], pt.y = y[j], pt.z = 0;
        pt.r = 0, pt.g = 0, pt.b = 255;
        position_cluster2.push_back(pt);
    //}

    if(time_ < 2.00 - 0.01) {
        j++;
        time_ += 0.01;
    }
    else { flag++; time_ = 0; j = 0; }
}
```

$$\cos\theta_2 = \frac{x^2 + y^2 - (l_1^2 + l_2^2)}{2l_1 l_2}$$

$$\sin\theta_2 = \pm\sqrt{1-\cos^2\theta_2}$$

$$\theta_2 = \mathrm{atan2}(\sin\theta_2, \cos\theta_2)$$

$$\cos\theta_1 = \frac{(l_1+l_2c_2)x + l_2s_2y}{(l_1+l_2c_2)^2+(l_2s_2)^2}$$

$$\sin\theta_1 = \frac{-l_2s_2x + (l_1+l_2c_2)y}{(l_1+l_2c_2)^2+(l_2s_2)^2}$$

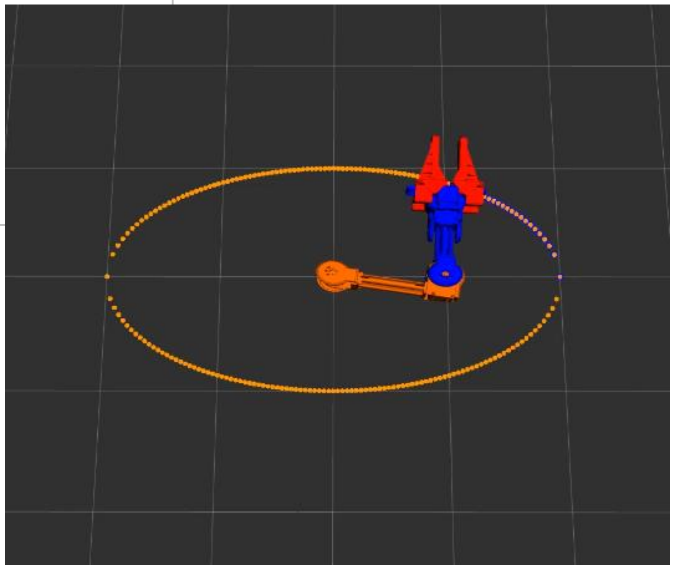
$$\theta_1 = \mathrm{atan2}(\sin\theta_1, \cos\theta_1)$$

$$P_x = l_1\cos\theta_1 + l_2\cos(\theta_1+\theta_2)$$

$$P_y = l_1\sin\theta_1 + l_2\sin(\theta_1+\theta_2)$$

$$total\ time = 2\ second$$

$$sequence = 0.01$$



4. Source Analysis

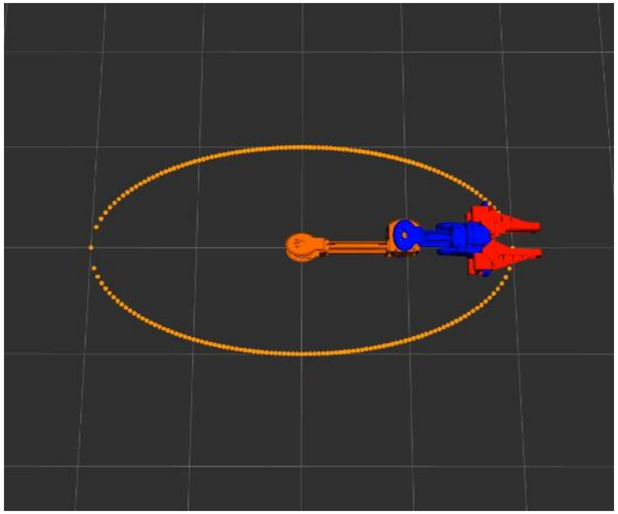
② $\theta - r$ Planar Robot

```
if(flag == 0) {      // 타원 궤적 그리기
    // (x * x) / (a * a) + (y * y) / (b * b)        // a = 2, b = 1
    float x_ = 2.00 - time2 * 0.04;
    float a = 2, b = 1;

    x[j] = x_;  // 궤적 공식
    float y_ = (float)sqrt(1 - pow(x_, 2) / pow(a, 2));

    if(time_ < 1.00 - 0.01)
        y[j] = y_;
    else
        y[j] = -1 * y_;

    //if(j % 2 == 0){
        pt.x = x[j], pt.y = y[j], pt.z = 0;
        pt.r = 255, pt.g = 150, pt.b = 0;
        position_cluster1.push_back(pt);
    //}

    if(time_ < 2.00 - 0.01) {
        j++;
        time_ += 0.01;
        if(time_ < 1.00) time2++;
        else time2--;
    }
    else { flag++; time_ = 0; j = 0; }
}
```

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1$$

$$total\ time = 2\ second$$

$$sequence = 0.01$$



4. Source Analysis

② $\theta - r$ Planar Robot

```
else if(flag == 1) {          // 상박 제어
    float r = (float)sqrt(pow(x[j], 2) + pow(y[j], 2));
    float c = x[j] / r;
    float s = y[j] / r;
    joint_position[0] = atan2(s, c);

    pcl::PointCloud<pcl::PointXYZRGB> position_cluster;
    position_cluster.header.frame_id = "base";

    if(time_ < 2.00 - 0.01) {
        j++;
        time_ += 0.01;
    }
    else { flag++; time_ = 0; j = 0; }
}
```

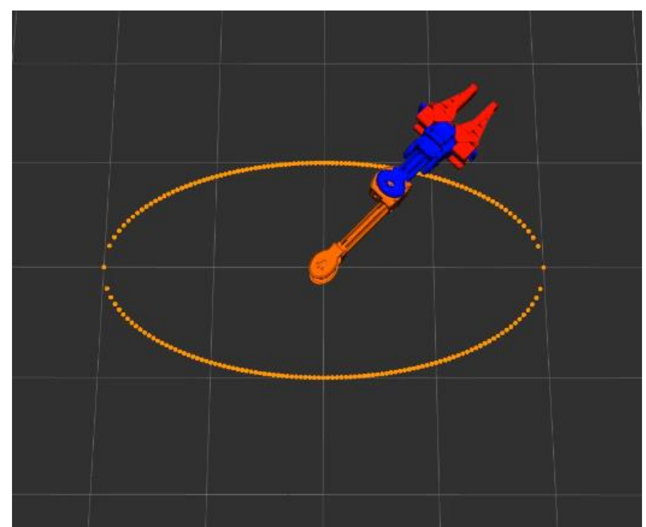
$$r = \sqrt{x^2 + y^2}$$

$$\cos\theta = \frac{x}{r} \qquad \sin\theta = \frac{y}{r}$$

$$\theta = \text{atan2}(\sin\theta, \cos\theta)$$

$$total\ time = 2\ second$$

$$sequence = 0.01$$



4. Source Analysis

② $\theta - r$ Planar Robot

```
else if(flag == 2) {          // 상박+하박 제어
    float r = (float)sqrt(pow(x[j], 2) + pow(y[j], 2));
    float c = x[j] / r;
    float s = y[j] / r;
    joint_position[0] = atan2(s, c);
    joint_position[1] = r - 2;

    x[j] = (2.0 + joint_position[1]) * cos(joint_position[0]);   // 궤적 공식
    y[j] = (2.0 + joint_position[1]) * sin(joint_position[0]);

    //if(j % 2 == 0) {        // 궤적 2
        pt.x = x[j], pt.y = y[j], pt.z = 0;
        pt.r = 0, pt.g = 0, pt.b = 255;
        position_cluster2.push_back(pt);
    //}

    if(time_ < 2.00 - 0.01) {
        j++;
        time_ += 0.01;

        if(time_ < 1.00 - 0.01)
            time2 += 0.01;
        else
            time2 -= 0.01;
    }
    else { flag++; time_ = 0; j = 0; }
}
```

$$r = \sqrt{x^2 + y^2}$$

$$\cos\theta = \frac{x}{r} \qquad \sin\theta = \frac{y}{r}$$

$$\theta = \mathrm{atan2}(\sin\theta, \cos\theta)$$

$$P_x = r\cos\theta$$

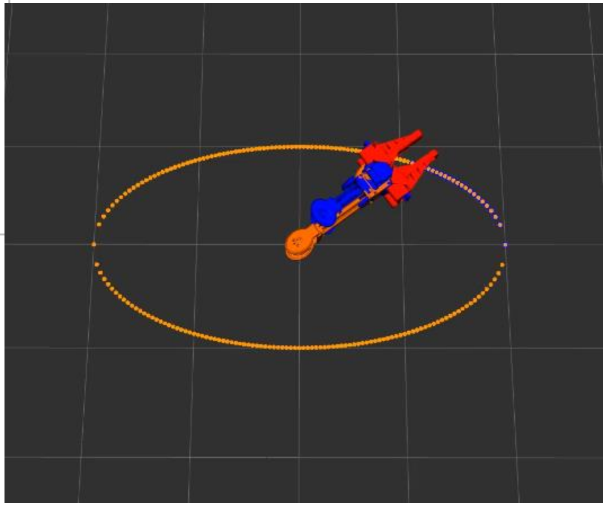
$$P_y = r\sin\theta$$

$$total\ time = t_1 = 2\ second$$

$$sequence = 0.01$$

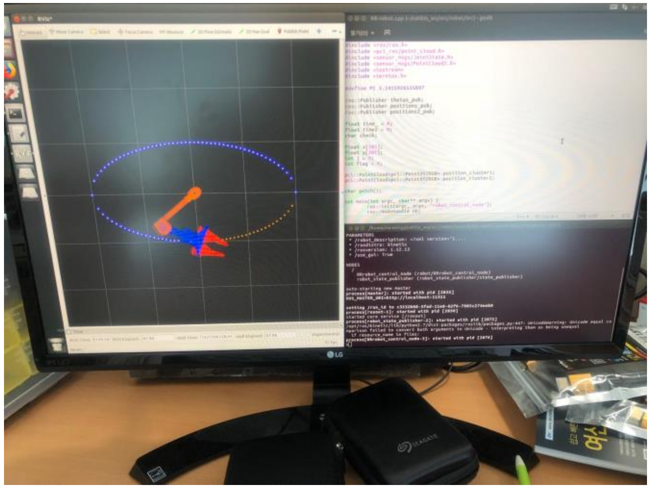
$$t_2 = 0 \rightarrow 1\ (t_1 = 0 \sim 1)$$

$$t_2 = 1 \rightarrow 0\ (t_1 = 1 \sim 2)$$



5. Impression

Manipulator를 3D로 직접 설계하고 확인해 봄으로써 Joint, Link의 구조에 대해 정확하게 이해할 수 있었습니다. 또한 simulator에 Inverse Kinematics를 적용시켜 봄으로써 가야 되는 좌표가 주어졌을 때 Joint들의 제어로 End-effector의 위치를 이동시킬 수 있는 방법에 대해 알 수 있었습니다.



Q & A

감사합니다.