

Lab2: UR robot path planning simulation

Fall, 2021

by Soojin Kim and Youngshik Kim



Intelligent Control & Robotic System (ICRS) lab.

Department of Mechanical Engineering

Hanbat National University

<http://robot.hanbat.ac.kr/>

6축 로봇팔 경로 생성 시뮬레이션 심화 (김영식 교수)

2주차

1. 실험목적

속도를 기반으로 그에 대한 경로를 (적분하여) 구할 수 있다. Homogeneous transformation matrix를 이해하고 이를 활용하여 TCP를 사용자가 원하는 방향으로 로봇의 움직임을 설정할 수 있다.

- (1) Task space와 joint space에서 로봇팔의 TCP(Tool Center Point) 또는 end effector의 경로 생성 방법을 이해한다.
- (1) 3D 공간 (XYZ 좌표)에서 물체의 이동과 회전 운동을 이해한다.
- (2) 로봇 end-effector/tool의 pose(position & orientation)를 이해한다.
- (3) 경로 생성을 위한 프로그램과 명령어를 이해한다.
- (4) 6축의 UR 로봇 경로를 3D 공간에서 Task space와 joint space를 활용하여 생성하고 시뮬레이션한다. 주어진 위치와 회전 정보를 바탕으로 homogeneous transformation matrix를 사용하여 로봇팔의 경로를 프로그램할 수 있다.

2. 기초 지식 및 실험방법

로봇팔의 경로를 생성할 때 task space 또는 joint space를 활용한다. Task space는 일반적인 XYZ 직교좌표 공간을 의미하며 우리가 쉽게 인지할 수 있는 장점이 있다. 하지만 조인트로 구성된 실제 로봇을 구동하기 위해서는 조인트의 모션을 결정해야 한다. 이에 조인트 변수(joint angles)로 구성된 벡터 공간을 joint space라하며 로봇의 모션 생성 및 제어를 위해 흔히 사용된다.

먼저, 로봇의 base에 고정된 reference 좌표계와 각 조인트, tool, end-effector에 상대 좌표계를 설정한다. Task space에서는 XYZ 직교 좌표계를 사용하여 reference로부터 로봇 tool 또는 end-effector의 위치와 회전을 결정한다. 일반적으로 물체 및 좌표계의 직선이동과 회전이동을 동시에 표현하여 위하여 homogeneous transformation matrix를 활용한다. 그리고 시간/상태에 따라 이 위치와 회전을 결정하여 로봇이 이를 tracking하도록 한다.

이를 위해서는 기술적으로 로봇의 inverse kinematics와 dynamics, 그리고 control 등의 고려가 필요하지만, 본 실험 범위를 벗어나므로 여기서는 설명을 생략한다. 본 실험에서 사용하는 URX library에서는 task space motion과 joint space motion에 필요한 정보를 계산 및 변환하는 함수들을 제공하므로 이 함수들을 사용한다. 따라서 본 실험에서는 task space에서 로봇 툴의 경로(위치와 방향)를 계획하고 이를 UR toolbox를 활용하여 시뮬레이션한다.

Homogeneous transformation matrix:

$$\mathbf{T} = \begin{bmatrix} \mathbf{R} & \mathbf{p} \\ \mathbf{0}_{1 \times 3} & 1 \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & x \\ r_{21} & r_{22} & r_{23} & y \\ r_{31} & r_{32} & r_{33} & z \\ 0 & 0 & 0 & 1 \end{bmatrix} \in \mathbb{SE}(3)$$

Base/reference 좌표계, Tool 좌표계는 첨부 그림 참고.

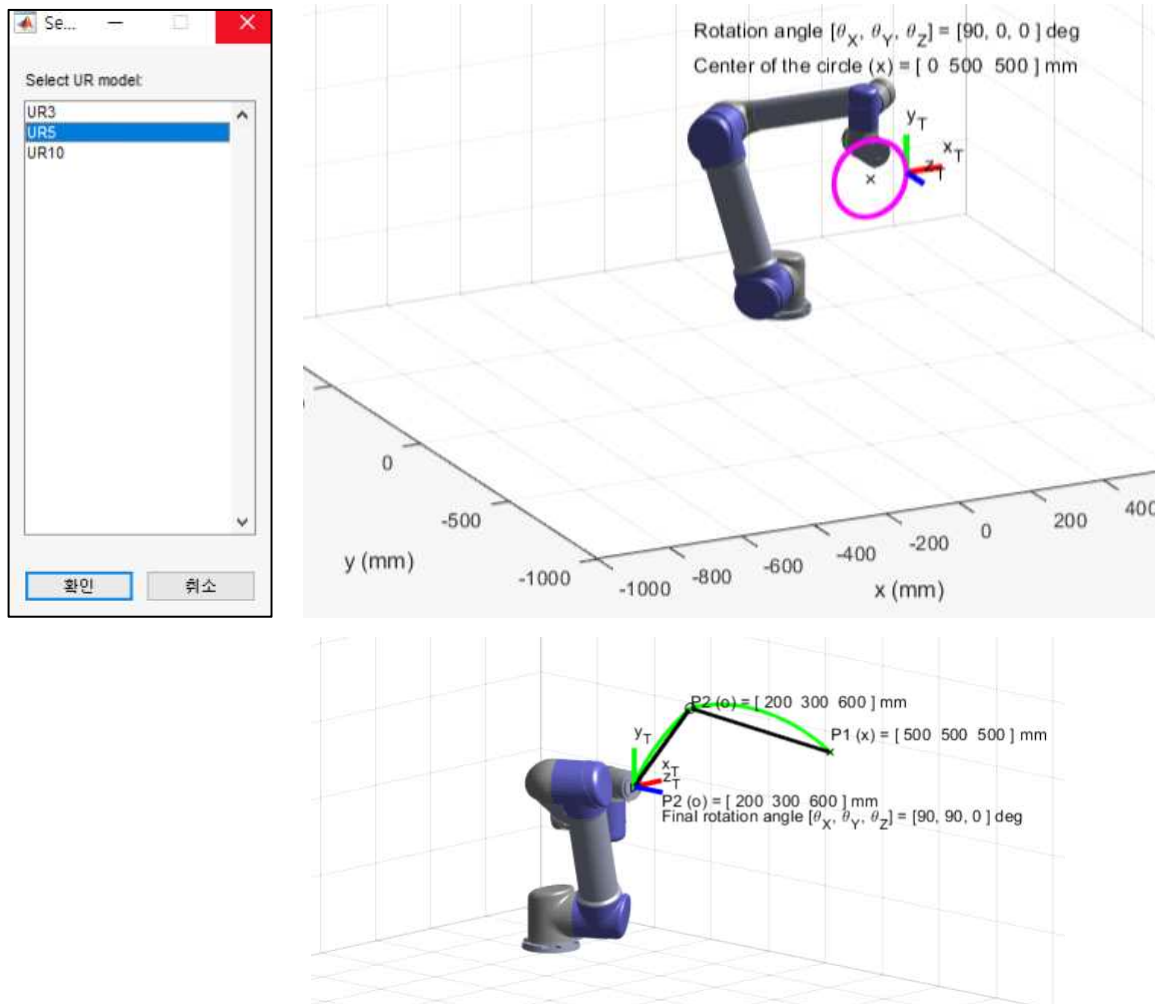
URX에서 Transformation (변환) 사용 예시: $T = T_x(x) * T_y(y) * T_z(z) * R_x(\theta_X) * R_y(\theta_Y) * R_z(\theta_Z)$
여기서 행렬 연산 순서 중요함. 순차적으로 현재/이동 좌표계를 기준으로 변환됨.

3. 예제 프로그램 실행 방법

(1) 2주차 예제 코드(Lab2_Task1_DrawCircle.m, Lab2_Task2.m)를 다운로드 받는다.

https://github.com/icrs-hub/Mech_lab_public

(2) Matlab Editor에서 예제 코드를 불러온 후 F5키 또는 실행 버튼을 눌러 실행한다. 만약 UR 모델 선택 화면이 뜨면, UR5를 선택하고 확인 버튼을 누른다. 이후 아래 그림과 같이 로봇의 경로와 동작을 확인할 수 있다.



4. 실험방법 (Task1)

- (1) 먼저 다음 테이블을 이용하여 원의 중심점과 툴의 Z_T 축을 선택한다. 원의 반지름은 100 mm이다. 그리고 로봇 end effector/TCP(tool center point)의 변환을 사용하여 위치와 회전을 결정한다.

Case	1	2	3	4	5	6	7	8	9	0
Xc	200	100	200	-200	200	200	-200	100	300	200
Yc	200	200	100	200	-100	100	-200	200	-300	-300
Zc	100	200	200	600	400	500	100	200	400	200
Tool 의 Z_T 축 방 향	-Y	-X	-Z	Y	X	Y	-Z	-X	-Y	X

- Case 번호는 본인 학번의 끝자리 수로 결정함.
- X, Y, Z 는 로봇 base/reference의 좌표축임.
- Xc, Yc, Zc 는 base 기준 로봇 Tool 좌표계의 원점 위치로, 여기서는 원의 중심으로 사용됨. (단위: mm)
- 로봇 Tool의 Z_T 축 방향을 base의 축을 기준으로 표현할 수 있음. 여기서는 Z_T 축은 경로가 생성되는 평면에 수직 아래 방향으로 사용됨.
- 문제를 단순화하기 위하여 회전 변환 시 x, y, z 축 중 하나만 사용하여 회전하는 것으로 제한함.

- (2) 예제 코드, Lab2_Task1_DrawCircle.m을 수정하여 시뮬레이션하고 결과를 확인한다.

5. 실험방법 (Task2)

- (1) Task space와 joint space를 사용하여 경로를 생성한다. 먼저 다음 테이블을 사용하여 점 P1, P2, P3의 위치와 툴 축의 방향을 결정한다.

Case	1	2	3	4	5	6	7	8	9	0
X	500	100	400	-200	200	400	-200	100	300	-200
Y	400	200	600	200	-100	300	-200	200	-300	-300
Z	400	200	100	600	400	500	200	500	300	200
X_T ,	-Y,	-X,	-Z,	Y,	-X,	Y,	-Z,	-X,	-Y,	Z,
Y_T ,	-X,	-Y,	Y,	X,	Y,	Z,	-X,	Y,	Z,	Y,
Z_T	-Z	Z	X	-Z	-Z	X	Y	-Z	-X	-X

- 점 P1, P2, P3는 본인 학번의 끝자리 수에 각각 0, 1, 2 더한 Case 번호로부터 결정함. (예) 학번 끝자리 수가 5이면, Case 5, 6, 7의 좌표를 순차적으로 선택. 만약 숫자가 10이 넘어가면 끝자리 수만 고려함.
- X, Y, Z는 로봇 base/reference 좌표를 기준으로 한 툴의 위치임 (단위: mm)
- X_t , Y_t , Z_t 는 각 점의 위치에서 Tool의 좌표계 축을 나타냄.
- X , Y , Z 는 base/reference의 좌표계 축을 나타냄.
- 문제를 단순화하기 위하여, 툴 좌표축의 회전은 현재 툴 좌표의 x축, y축, z축 순서대로 회전한다고 가정한다. 이때 회전변환은 $T_x * T_y * T_z$ 와 같이 표현된다. 여기서 각 회전 변환은 기준 좌표계가 아닌 현재 로컬 좌표(여기서는 툴 좌표)를 기준으로 회전한다는 점에 주의할 것.

(2) Task space에서 점 P1, P2, P3를 통과하는 직선 경로를 생성한다. 여기서는 툴의 회전 방향을 고려하기 위하여 Transformation 연산을 수행한다.

(3) 예제 코드 Lab2_Task2.m를 수정하여 P1, P2, P3를 순차적으로 통과하는 직선 경로와 조인트를 사용하여 생성한 곡선 경로를 모두 생성한다. 아래 내용을 참고할 것.

- 로봇 툴이 지나가는 점 P1과 P2 사이, 그리고 점 P2와 P3 사이의 중간 경로점의 개수는 각각 50개로 한다.
- 각 경로점 마다 0.5초 동안 일시정지 명령을 사용한다: pause(0.5)
- 여기서 사용되는 길이의 단위는 mm이고 각도의 단위는 radian이다.
- 필요 시 계산은 소수점 둘째 자리까지 한다. (예: 0.01 mm)

(4) 코드를 수정하여 시뮬레이션하고 결과를 확인한다.

6. 제출물

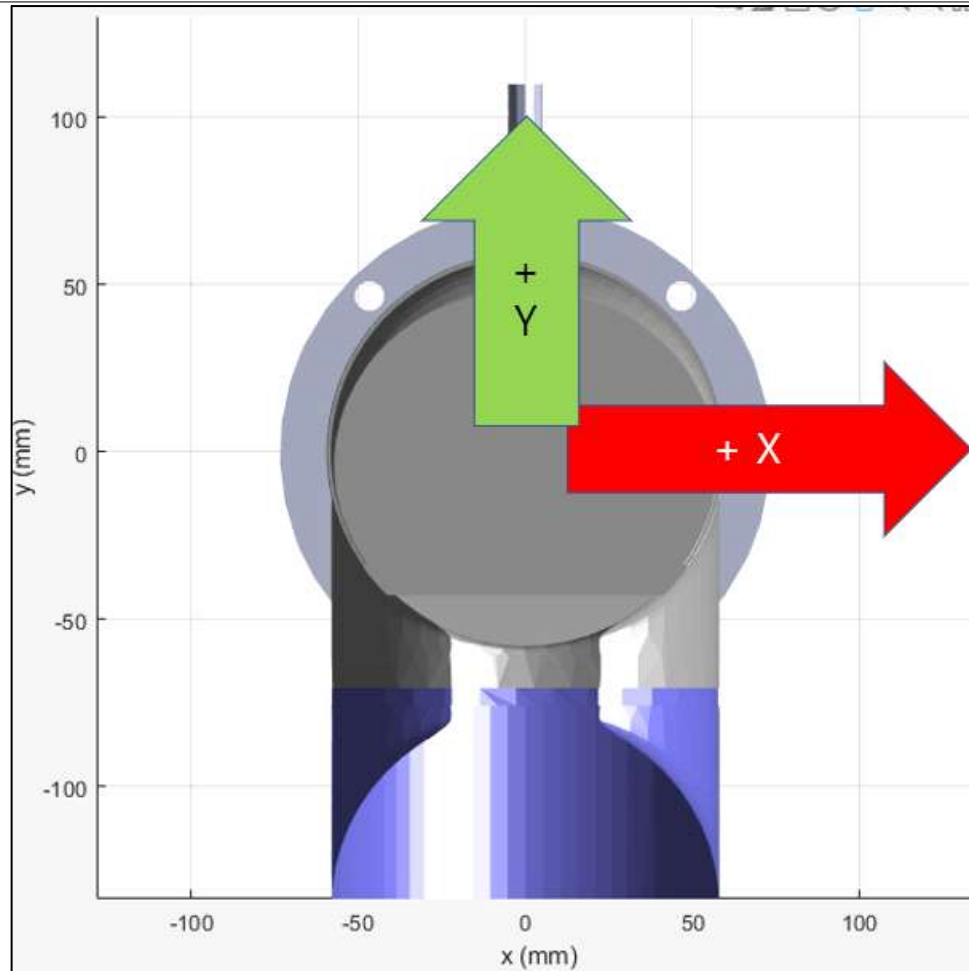
(1) 보고서: 실험_주차_보고서_이름_학번.pdf,

ex) "실험_1주차_보고서_홍길동_0123456.pdf"

(2) 매틴랩 코드: Lab2_Task1_DrawCircle_학번.m, Lab2_Task2_학번.m

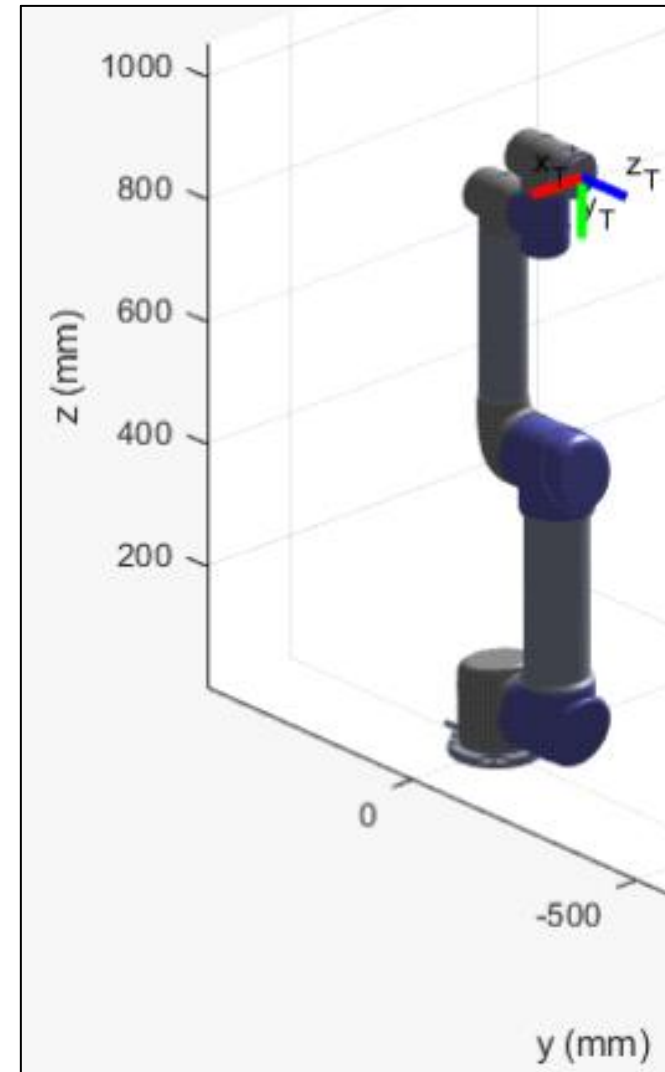
2-1. Base 좌표계, Tool 좌표계

Base 좌표계



선 튀어나온 곳이 +Y 방향,
현재 그림 시점에서 오른쪽 방향이 +X 방향

Tool 좌표계



로봇팔 시뮬레이션 결과보고서 2주차 Task1 작성예시

이름 : 김한발 학번 : 20211329 분반: 03 날짜 : 2021년 8월 31 일

1. Task1 실험을 위해 사용한 원의 중심 좌표와 로봇 툴의 방향 및 회전 변환 각도를 표시 하시오.

원의 중심 좌표, [Xc, Yc, Zc] (mm)	툴의 Z_T 축 방향	Base에 대한 회전 변환 각도, [thX, thY, thZ] (rad)
[0, 500, 500]	-Y 축	[pi/2, 0, 0]

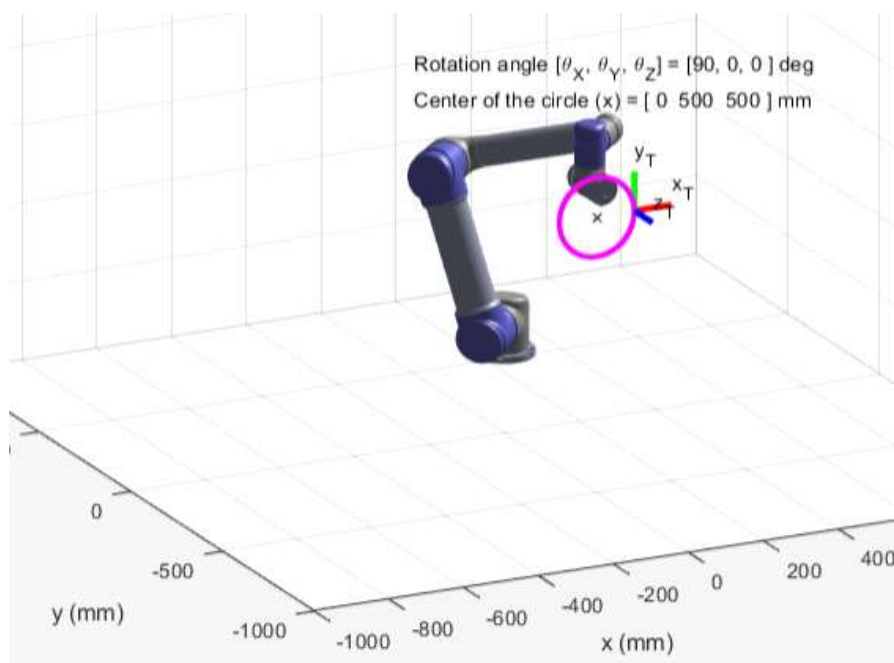
2. Task1 실험을 위해 적용한 reference로부터 툴의 변환 행렬, T를 URX 함수와 위의 테이블 값을 이용하여 표현하시오.

$$T = T_y(500) * T_z(500) * R_x(\pi/2)$$

3. URX 함수와 homogeneous coordinates을 사용하여 툴 좌표계에서 주어진 원의 좌표점, $[X_T, Y_T, Z_T]$ 를 reference에 대한 좌표점, $[X, Y, Z]$ 로 변환하는 명령어를 작성하시오. 계산 필요 없음.

$$[X; Y; Z; 1] = T_y(500) * T_z(500) * R_x(\pi/2) * [X_T; Y_T; Z_T; 1]$$

4. 시뮬레이션 결과를 그림으로 첨부하시오. 실습에서 주어진 원의 중심점의 좌표와 툴의 회전 방향이 맞는지 확인할 것.



로봇팔 시뮬레이션 결과보고서 2주차 Task 2 작성예시

이름 : 김한발 학번 : 20211329 분반: 03 날짜 : 2021년 8월 31 일

1. Task2 실험을 위해 사용한 점 P1, P2, P3 좌표와 툴의 회전 변환 각도를 적으시오.

	Base 기준 P1, P2, P3의 좌표 (mm)	Base 축을 사용한 각 점에서 툴의 축 방향, $[X_T; Y_T; Z_T]$	각 점에서 Base에 대한 툴의 회전 변환 각도 $[thX, thY, thZ]$ (rad)
P1	[400, 500, 400]	$[X; Z; -Y]$	$[pi/2, 0, 0]$
P2	[200, 300, 600]	$[Y; Z; X]$	$[pi/2, pi/2, 0]$
P3	[100, 200, 400]	$[-Y; Z; -X]$	$[pi/2, -pi/2, 0]$

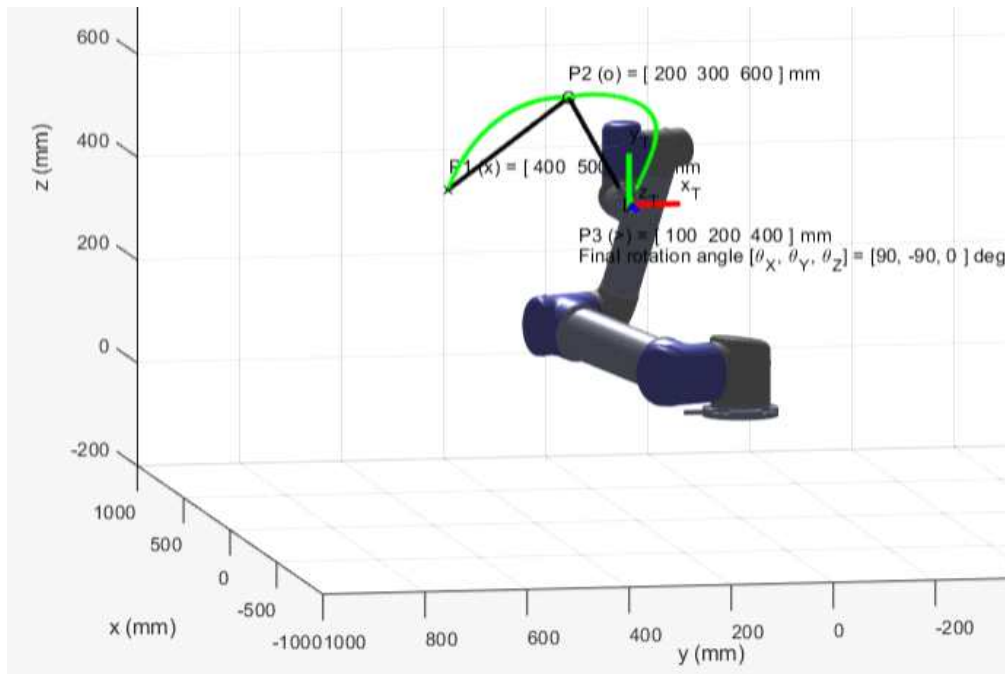
2. 로봇 reference/base 좌표를 점 P2로 평행 이동 변환(translation)하고, 점 P2에서 회전 변환(rotation)을 적용한 툴의 좌표 변환 행렬(coordinate transformation matrix), T를 URX 함수와 위의 테이블 값을 이용하여 Matlab 코드로 표현하시오. 그리고 변환 행렬 T의 값을 계산하시오.

```
T=Tx(200)*Ty(300)*Tz(600)*Rx(pi/2)*Ry(pi/2) % T의 크기=4x4
=[ 0.0000      0      1.0000  200.0000
   1.0000     0.0000  -0.0000  300.0000
  -0.0000     1.0000   0.0000  600.0000
      0         0         0      1.0000 ]
```

3. 그리고 2번에서 구한 변환으로부터 조인트의 각도 벡터, q를 구하는 Matlab 명령어를 표현하시오. 그리고 조인트 각 벡터 q를 구하시오.

```
simObj.ToolPose=Tx(200)*Ty(300)*Tz(600)*Rx(pi/2)*Ry(pi/2);
q=simObj.Joints % 조인트 각 벡터 q의 크기=6x1
=[ 1.5425,  -1.4671,  -1.1907,  -3.6255,  -0.0283,    0 ] rad
rad2deg(q) =[ 88.3759,  -84.0567,  -68.2200,  -207.7233,  -1.6241,    0 ] deg
```

4. 시뮬레이션 결과를 그림으로 첨부하시오. 실습에서 주어진 원의 중심점의 좌표와 툴의 회전 방향이 맞는지 확인할 것.



5. Task space 경로와 joint space 경로에 대하여 비교하고, 특징, 장단점, 활용 방법, 문제점 등 적절히 논의하시오.

Task space 경로의 특징은 ~~~이다. 또한 장점은 ~~~이고, 단점은 ~~~이다.

Task space 경로에 비하여 joint space 경로의 장점은 ~~~이고, 단점은 ~~~이다. 또한 특징은 ~~~이다.

~~~~~

6. (1) 점 P1에서 점 P3까지를 균등하게 50번 나누어 생성한 툴의 위치 벡터 성분,  $[X, Y, Z]$  을 생성하라. (2) 초기 P1에서 Tool의 방향은 reference와 같다. P3에서 방향은 1번 테이블의 변환 각도 값과 같을 때, 회전각의 변화를 50번 나누는 벡터 성분,  $[thX, thY, thZ]$ 를 생성하라. (3) 이 결과를 이용하여 Task space에서 로봇 툴의 좌표 변환, T와 조인트 각도, q를 구하고, 경로 시뮬레이션 하는 Matab 명령어를 작성하시오.

(예시) 여러 가지 방법 존재

(1) 위치 벡터 성분,  $[X, Y, Z]$

$X = \text{linspace}(400, 100, 50);$

$Y = \text{linspace}(500, 200, 50);$

$Z = \text{linspace}(400, 400, 50);$

(2) 회전 변환 각도 성분,  $[thX, thY, thZ]$

$thX = \text{linspace}(\pi/2, \pi/2, 50);$

$thY = \text{linspace}(0, -\pi/2, 50);$

```
thZ=linspace(0, 0, 50);
```

(3) Task space에서 툴의 좌표 변환, T와 조인트 각도, q, 경로 시뮬레이션

```
for l=1: length(X)
```

```
T=Tx(X(i))*Ty(Y(i))*Tz(Z(i))*Rx(thX(i))*Ry(thY(i))*Rz(thZ(i));
```

```
simObj.ToolPose=T;
```

```
q_task=simObj.Joints;
```

```
pause(0.5)
```

```
end
```

## 로봇팔 시뮬레이션 결과보고서 2주차 Task1

이름 : \_\_\_\_\_ 학번 : \_\_\_\_\_ 분반: \_\_\_\_\_ 날짜 : 2021년 \_\_\_\_\_ 월 \_\_\_\_\_ 일

1. Task1 실험을 위해 사용한 원의 중심 좌표와 로봇 툴의 방향 및 회전 변환 각도를 표시하시오.

| 원의 중심 좌표,<br>[Xc, Yc, Zc] (mm) | 툴의 $Z_T$ 축 방향 | Base에 대한 회전 변환<br>각도, [thX, thY, thZ] (rad) |
|--------------------------------|---------------|---------------------------------------------|
| [                      ]       |               | [                      ]                    |

2. Task1 실험을 위해 적용한 reference로부터 툴의 변환 행렬, T를 URX 함수와 위의 테이블 값을 이용하여 표현하시오.

3. URX 함수와 homogeneous coordinates을 사용하여 툴 좌표계에서 주어진 원의 좌표점,  $[X_T, Y_T, Z_T]$ 를 reference에 대한 좌표점,  $[X, Y, Z]$ 로 변환하는 명령어를 작성하시오. 계산 필요 없음.

4. 시뮬레이션 결과를 그림으로 첨부하시오. 실습에서 주어진 원의 중심점의 좌표와 툴의 회전 방향이 맞는지 확인할 것.

## 로봇팔 시뮬레이션 결과보고서 2주차 Task 2

이름 : \_\_\_\_\_ 학번 : \_\_\_\_\_ 분반: \_\_\_\_\_ 날짜 : 2021년 \_\_\_\_\_ 월 \_\_\_\_\_ 일

1. Task2 실험을 위해 사용한 점 P1, P2, P3 좌표와 툴의 회전축을 적으시오.

|    | Base 기준 P1, P2, P3의 좌표 (mm) | Base 축을 사용한 각 점에서 툴의 축 방향, $[X_T, Y_T, Z_T]$ | 각 점에서 Base에 대한 툴의 회전 변환 각도 $[thX, thY, thZ]$ (rad) |
|----|-----------------------------|----------------------------------------------|----------------------------------------------------|
| P1 | [                      ]    | [                      ]                     | [                      ]                           |
| P2 | [                      ]    | [                      ]                     | [                      ]                           |
| P3 | [                      ]    | [                      ]                     | [                      ]                           |

2. 로봇 reference/base 좌표를 점 P2로 평행 이동 변환(translation)하고, 점 P2에서 회전 변환(rotation)을 적용한 툴의 좌표 변환 행렬(coordinate transformation matrix), T를 URX 함수와 위의 테이블 값을 이용하여 Matlab 코드로 표현하시오. 그리고 변환 행렬 T의 값을 계산하시오.

3. 그리고 2번에서 구한 변환으로부터 조인트의 각도 벡터, q를 구하는 Matlab 명령어를 표현하시오. 그리고 조인트 각 벡터 q를 구하시오.

4. 시뮬레이션 결과를 그림으로 첨부하시오. 실습에서 주어진 원의 중심점의 좌표와 툴의 회전 방향이 맞는지 확인할 것.

5. Task space 경로와 joint space 경로에 대하여 비교하고, 특징, 장단점, 활용 방법, 문제점 등 적절히 논의하시오.