

---

# Mech-Viz Manual

Mech-Mind

2022 년 07 월 01 일

<b>1 기본 개념 소개</b>	<b>5</b>
1.1 로봇	5
1.2 TCP[Tool Center Point] 툴의 센터 포인트	6
1.3 관절 위치	7
1.4 포즈	7
1.5 정방향 운동학	8
1.6 역운동학	8
1.7 피킹-배치	8
1.8 대칭성	8
1.9 피킹 툴 옵셋	11
1.10 충돌 모델 유형	12
<b>2 Mech-Viz 인터페이스 소개</b>	<b>13</b>
2.1 사용자 인터페이스	13
2.2 메뉴 바 소개	26
2.3 태크 페이지 소개	35
<b>3 FAQ</b>	<b>82</b>

### Mech-Viz 지능형 프로그래밍 환경에 대한 기본 소개 :

Mech-Viz 는 메크 마인드 로보틱스가 자체 연구 개발한 산업용 로봇 그래픽 프로그래밍 으로 로봇 프로그래밍 제어 및 시각적 시뮬레이션을 수행하고 로봇 제어 시스템에서 제어 및 시뮬레이션 환경을 제공하며 Mech-Vision Mech-Center 등 소프트웨어와 결합하여 로봇의 비전을 기반으로 하는 지능형 생산을 실현합니다. 그림 1 과 같습니다.

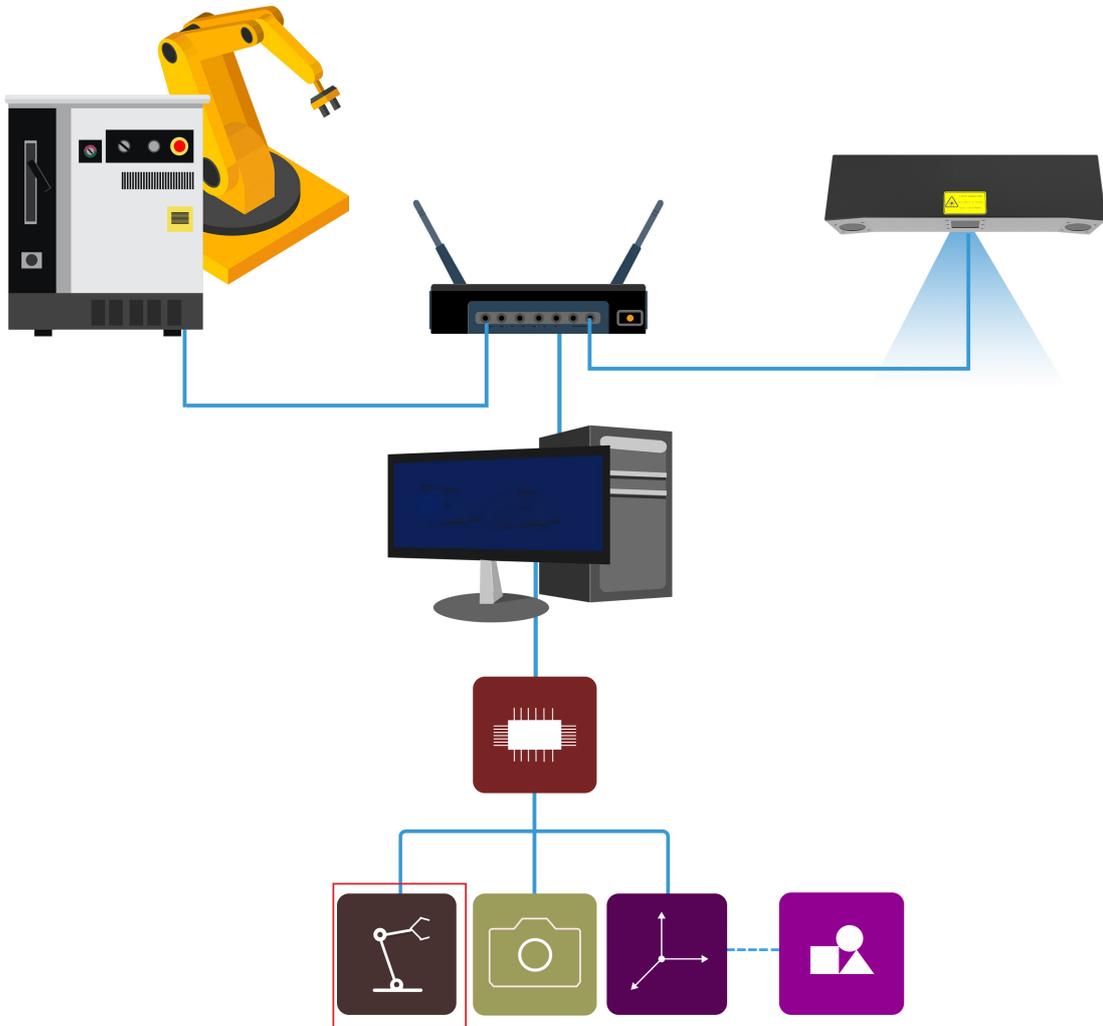


그림 1 Mech-Viz 가 로봇 제어 시스템에서의 역할

그림 2 에 표시하는 바와 같이 Mech-Viz 가 친 사용자적인 인터페이스 및 자체 연구 개발한 고효율적인 비번 솔루션 개발 도구를 갖고 있으며 그래픽 프로그래밍 모델과 시각화 시뮬레이션 모듈을 결합함으로써 사용자들이 로봇 프로그래밍 언어에 관한 전문적인 지식을 습득하지 않아도 로봇 시스템에 대해 간단하고 직접적인 제어를 실현할 수 있어서 로봇 프로그래밍의 소요 시간 및 인건비를 크게 절약할 수 있습니다.

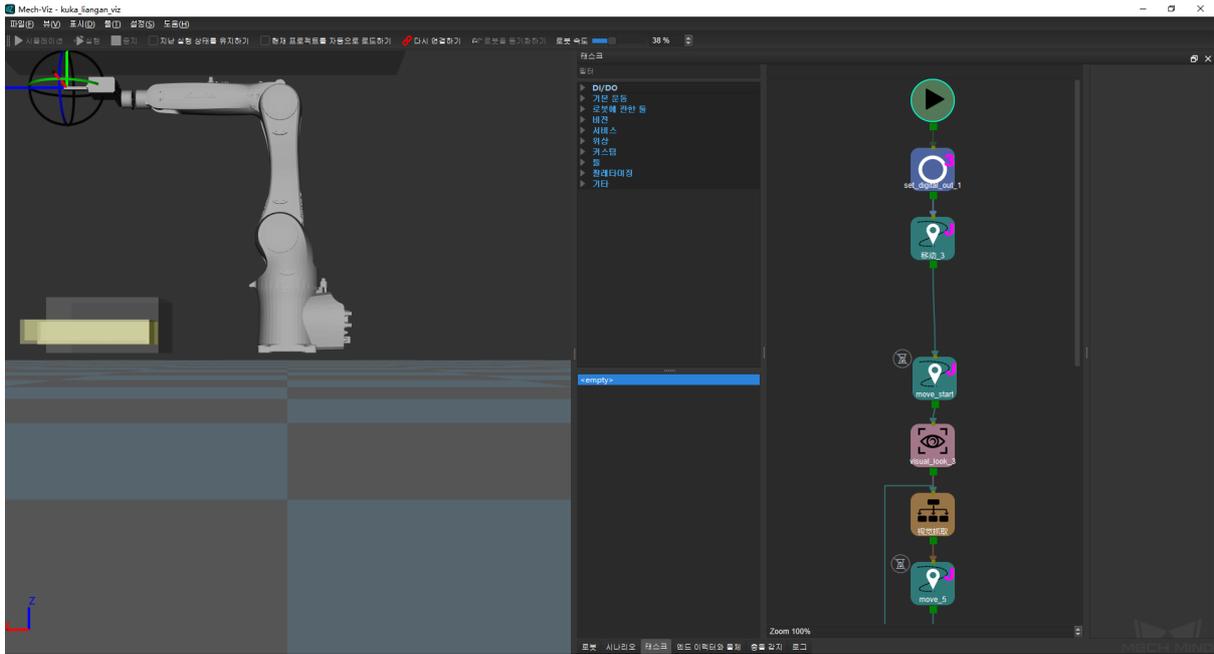


그림 2 Mech-Viz 사용자 인터페이스

그림 3.1 과 그림 3.2 에 표시하는 바와 같이 Mech-Viz 가 동일한 인터페이스를 사용하여 다양한 브랜드 각 모델의 로봇을 제어할 수 있고 지능형 의사 결정, 자율적인 경로 계획, 프로그램 검사 등 지능형 알고리즘이 내장되어 있으며 비전 시스템과 원활하게 통합될 수 있습니다.

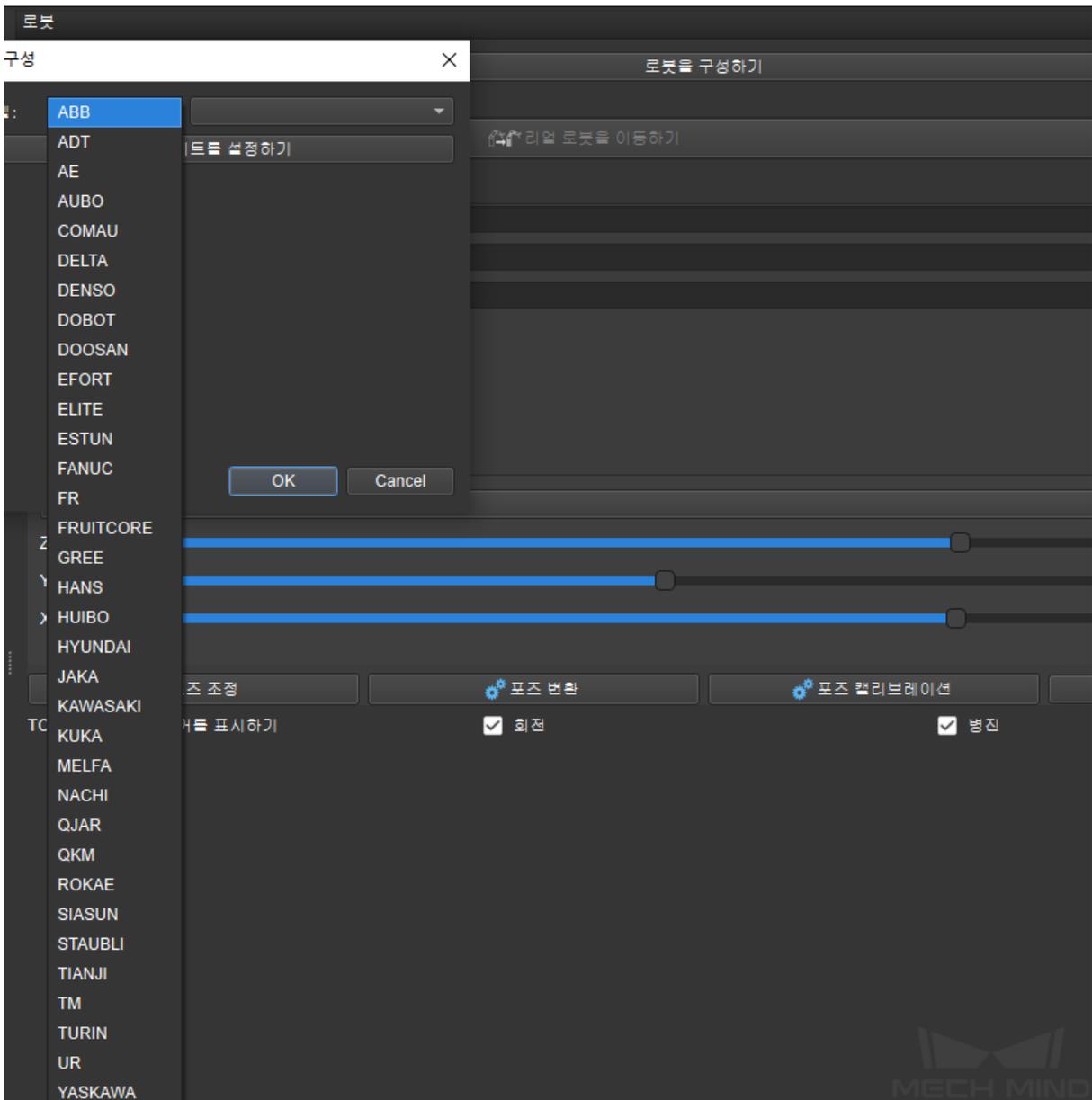


그림 3.1 Mech-Viz 가 로봇 브랜드를 선택하기

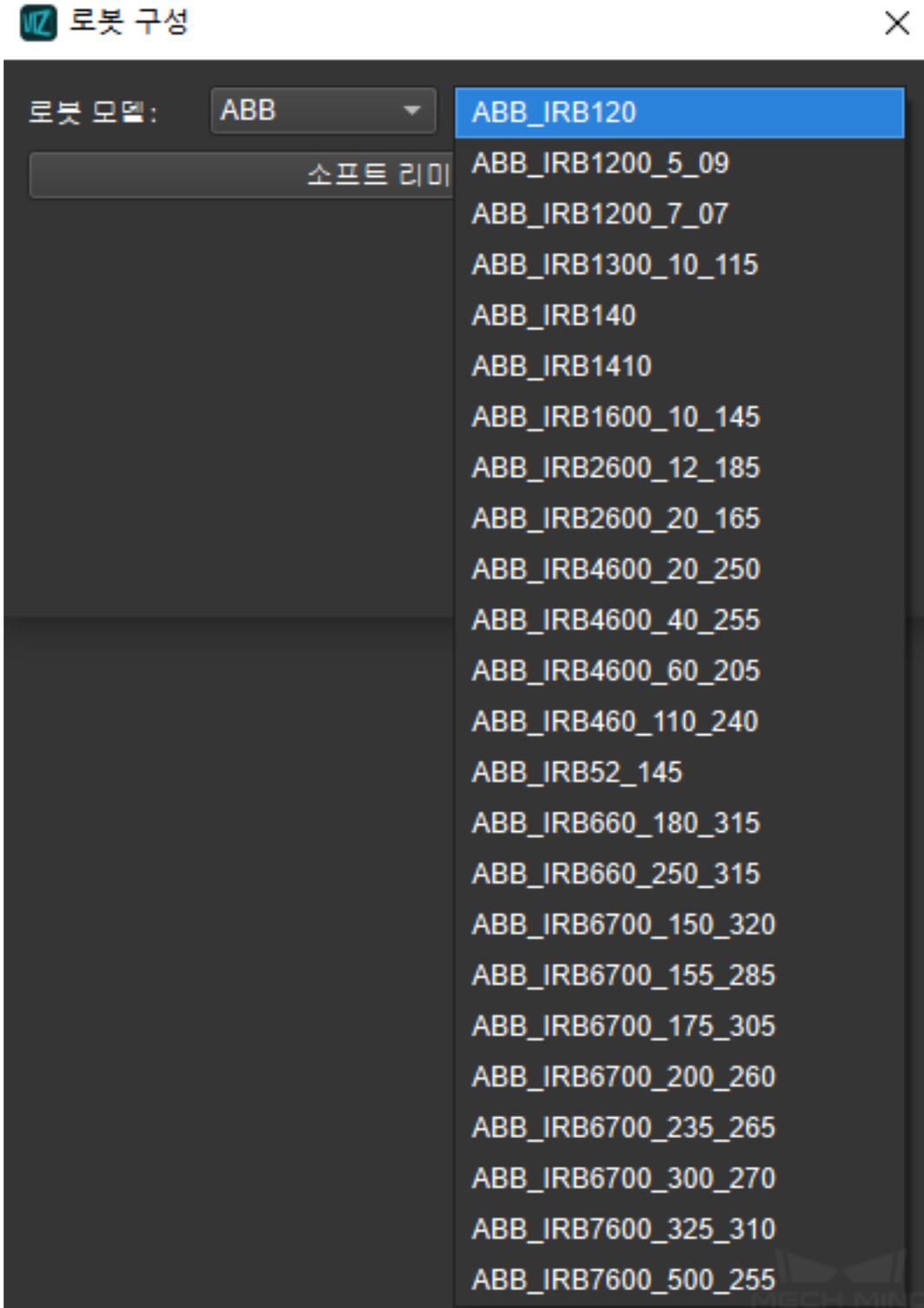


그림 3.2 Mech-Viz 가 로봇 모델을 선택하기

### 1.1 로봇

특별한 설명이 없으면 여기서 말하는 로봇은 관절을 통해 연결된 강체로 구성되는 시스템을 가리킵니다. 아래의 그림과 같습니다:



## 1.2 TCP[Tool Center Point] (툴의 센터 포인트)

일반적으로 로봇에 주변 다른 물체와 상호작용을 실현하는 도구인 엔드 이펙터가 배치되어 있으며 TCP는 바로 엔드 이펙터의 팁 포인트입니다. 물체 피킹 등 태스크를 수행하기 위해 로봇이 공간에 있는 어떤 포인트로 이동하라고 하는 것은 본격적으로 TCP가 이 포인트로 이동하게끔 하는 것입니다.

## 1.3 관절 위치

로봇 각 관절의 위치의 뜻이고 관절 각도라고도 합니다. 관절 위치는 숫자의 세트이고 개수는 관절 축의 수량입니다. 예를 들어 6 축 로봇 하나의 세트에 있는 관절 위치의 수량은 6 입니다.

## 1.4 포즈

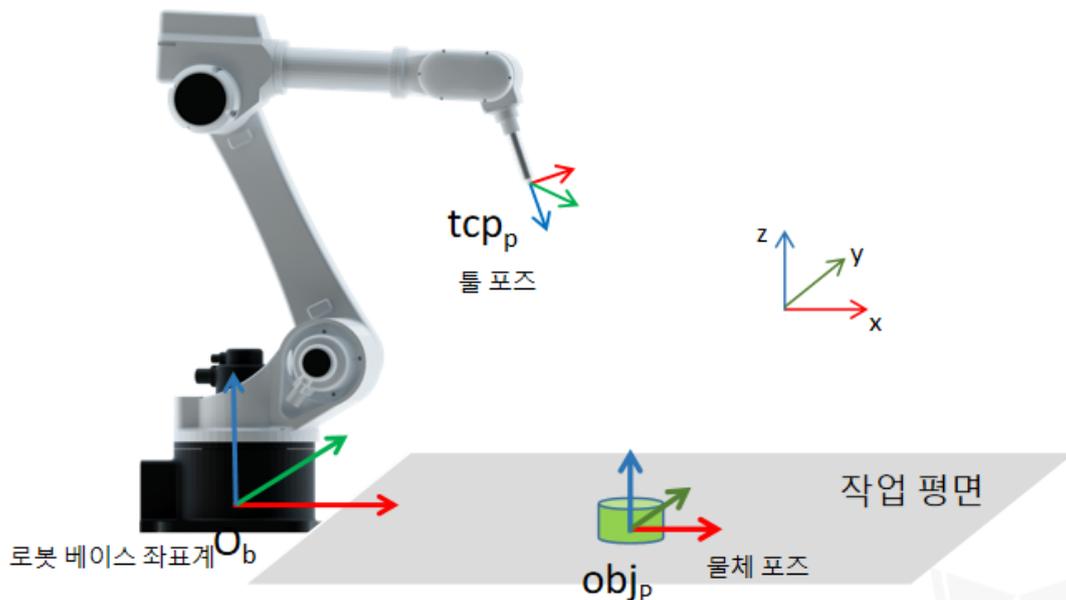
물체가 공간에 있는 위치 및 양상은 일괄적으로 포즈라고 합니다. 포즈는 병진 + 회전 (사원수 혹은 오일러 각) 으로 표시됩니다. Mech-Viz 에서 말하는 포즈가 툴 포즈 및 물체 포즈로 나뉘집니다.

### 1.4.1 툴 포즈

로봇 베이스를 기준으로 하여 TCP 의 포즈를 가리킵니다.

### 1.4.2 물체 포즈

물체 포즈는 로봇 베이스를 기준으로 하여 물체 중심 포인트의 포즈를 가리킵니다. 물체가 로봇의 말단에 붙여 있는 경우 물체의 초기 포즈가 툴 포즈와 일치하며 포즈가 반대라고 여깁니다. **대칭성** 혹은 **피킹 툴** **업셋** 의 경우 물체 포즈와 툴 포즈의 상대적인 변화는 달라질 수도 있습니다.



**주의:** 기본으로 툴 포즈의 Z 축과 세계 좌표계의 Z 축 간의 각도는 둔각입니다. 물체를 피킹할 때 물체 포즈의 Z 축은 툴 포즈의 Z 축의 방향은 반대가 되어야 합니다.

## 1.5 정방향 운동학

관절 위치에 의해서 툴 포즈를 계산하는 것입니다. 로봇의 관절 위치가 정해진다면 툴 포즈는 유일하고 확정된 것입니다.

## 1.6 역운동학

툴 포즈에 의해 해당하는 한 세트의 관절 위치를 계산하는 것입니다. 정해진 툴 포즈가 대응하는 관절 위치는 여러 개 존재할 수도 있으며 존재하지 않을 수도 있습니다.

## 1.7 피킹-배치

피킹-배치는 로봇과 물체 간의 상호작용의 방식입니다. moveTask-label 에 대칭성 응용, 물체 포즈를 툴 포즈로의 전환을 바꾸거나 리셋하는 데에 사용됩니다.

### 1.7.1 피킹

로봇이 물체 포즈로 이동하고 외부 인터페이스 신호를 변화시킴으로써 엔드 클램프를 제어하여 물체를 피킹하는 과정은 피킹이라고 합니다. 《피킹》이 성공되면 물체와 로봇 TCP 가 바인딩되어 있어 소프트웨어가 대칭성 에 의해 로봇 이동 궤적을 계획할 것입니다.

### 1.7.2 쥐고 있음

로봇이 물체를 피킹할 순간부터 배치할 때까지의 상태는 “쥐고 있음”이라고 합니다. 이 상태에 물체와 로봇 간의 바인딩 관계가 여전히 존재합니다. 이 상태는 물체 피킹 후 자동으로 적용되고 소프트웨어에서 선택할 수 없습니다.

### 1.7.3 배치

로봇이 배치점 포즈로 이동하고 외부 인터페이스 신호를 변화시킴으로써 엔드 클램프를 제어하여 물체를 두는 과정은 배치라고 합니다. 《배치》가 성공되면 물체와 로봇 TCP 간의 바인딩 관계가 없어질 것입니다.

## 1.8 대칭성

모든 타겟 물체가 물체 포즈를 갖고 있습니다. 소프트웨어는 물체 포즈에 의해 툴 포즈를 산출하여 로봇 피킹을 제어합니다. 자세한 정보는 포즈 를 참고하세요. 실제 상황에서 흔히 보는 대칭성을 갖는 물체에 대해 로봇은 대칭성 각도 에 따라 다양한 피킹 혹은 배치 방식을 활용하더라도 결과는 똑같습니다.

**대칭성 각도** 물체 혹은 피킹 포즈가 X/Y/Z 축을 중심으로 ° 를 한번 회전하면 이전과 같은 것으로 간주할 수 있습니다. 예를 들어: 정사각형의 ° 는 90°, 직사각형은 180°, 정육각형은 60°, 원 또는 링은 0°, 회전 없는 대칭은 360° 입니다. 아래의 그림과 같습니다:



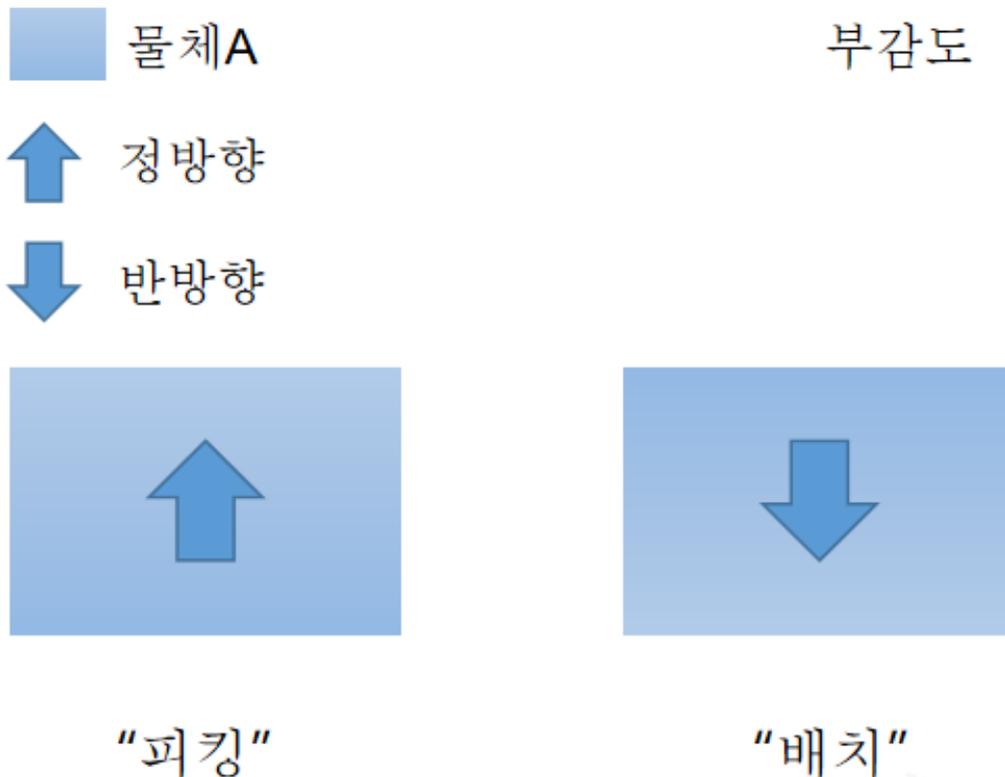
그 중에 Z 축을 중심으로 하는 대칭성은 강한 축 대칭성이라고 하며 X/Y 축을 중심으로 하는 대칭성은 약한 축 대칭성 이라고 합니다.

**약한 축 대칭성:** 일반적으로 대칭성이 더 강한 축을 실제 계산에 쓰이는 약한 축으로 여기버 다른 축의 대칭성을 무시합니다. 예를 들어: X 축은 90° 의 대칭성을 갖고 있으며 Y 축은 0° 의 대칭성을 갖고 있으면 Y 축은 약한 축으로 간주하고 약한 축의 대칭성은 0° 입니다.

사용자들은 물체의 대칭성을 설정하면 소프트웨어가 최적의 피킹 & 배치 궤적을 선택하여 접근성을 향상 시키고 말단의 회전을 줄입니다.

### 1.8.1 물체 대칭성

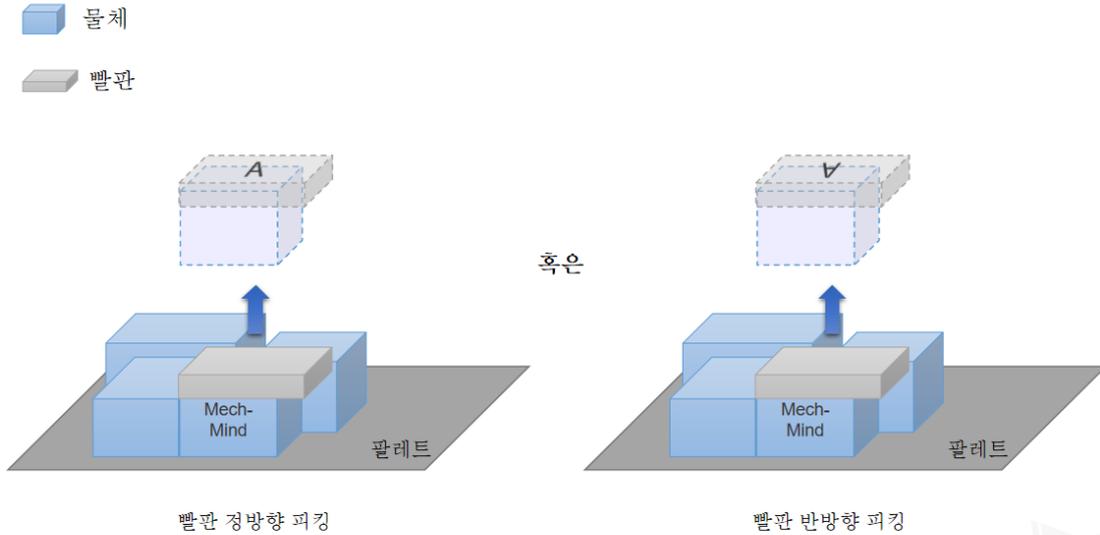
물체 대칭성은 물체 포즈와 관련이 있습니다. 예를 들어 물체의 대칭성이 180° 라면 로봇이 물체를 피킹/배치할 때《장》혹은《반》둘 중의 하나를 선택하여 말단 축의 회전을 줄일 수 있습니다. 아래의 그림과 같습니다:



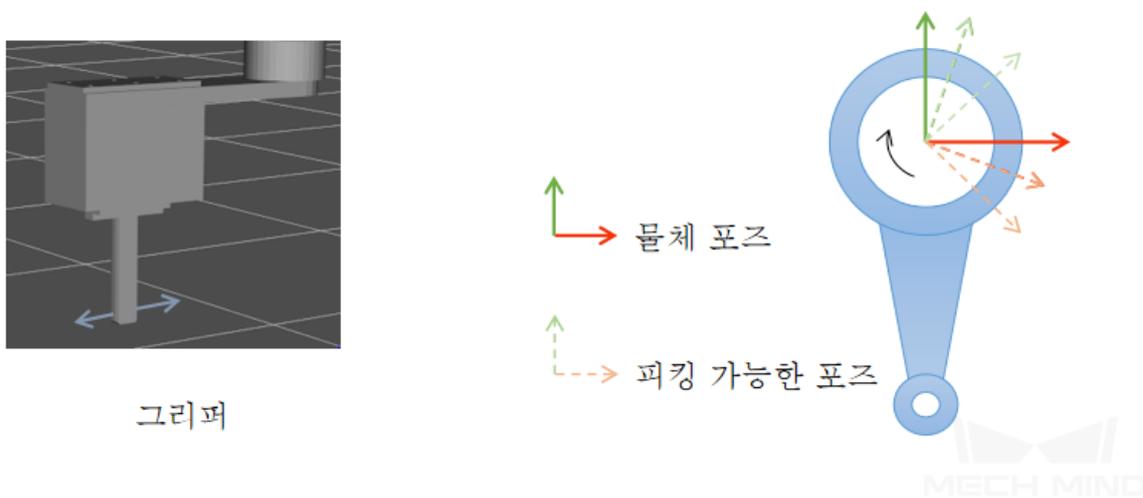
물체의 정 & 반 반향은 같은 상태로 간주할 수 있으니 로봇은 정방향 피킹 또한 반방향 배치 식으로 할 수 있습니다.

### 1.8.2 피킹 포즈 대칭성

피킹 포즈의 대칭성은 피킹 툴이 물체를 피킹하는 방법과 관련이 있으며 최적의 피킹 포즈를 선택하는 데에 사용됩니다. 예를 들어: 물체의 크기가 팔판보다 작은 경우에는 피킹 과정 속에서 주변에 있는 다른 물체와의 충돌을 방지하기 위해 팔판 올셋 식으로 피킹하는 것입니다. 이 때 물체의 대칭성은 사용되지 않으며 피킹 포즈의 대칭성은 180° 입니다. 최초 피킹 포즈에 피킹 툴을 회전시키고《정방향 피킹》혹은《반방향 피킹》을 선택할 수 있습니다. 아래의 그림과 같습니다:



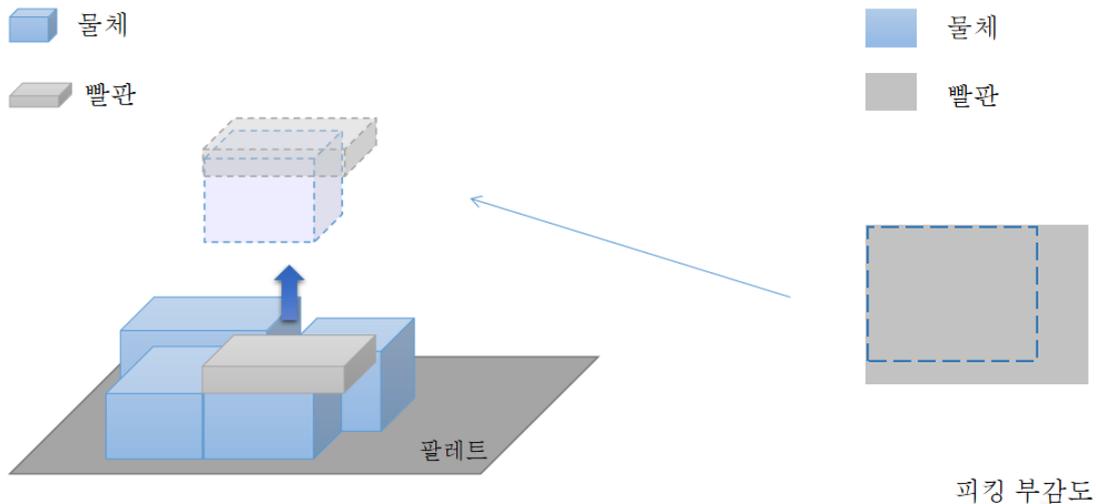
최적의 이동 궤적을 계획하기 위해 피킹 시 소프트웨어는 피킹 포즈의 대칭성 및 물체 대칭성을 동시에 계산합니다. 《쥐고 있음》및 배치 단계에 물체의 대칭성에 의해 물체 포즈를 선택할 것입니다. 예를 들어 타겟 물체가 연결봉인 경우에 그리퍼를 사용하여 링을 밖으로 피킹할 수 있습니다. 아래의 그림과 같습니다:



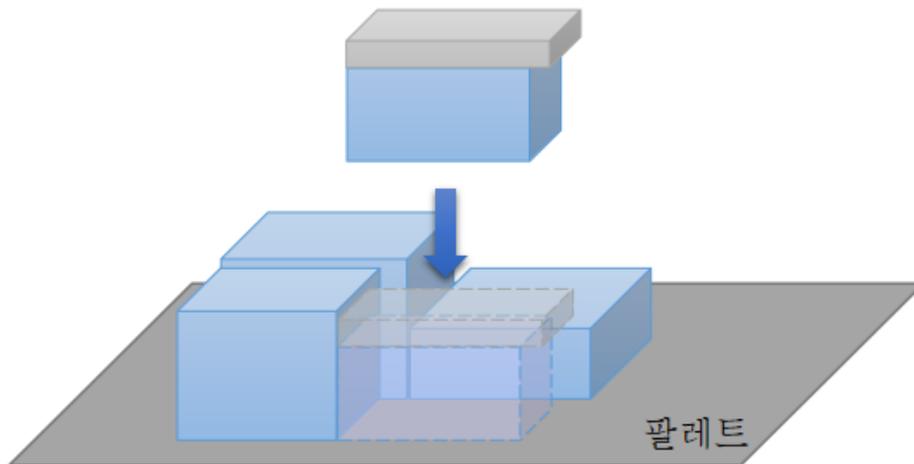
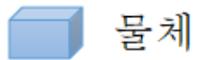
이 때 피킹 포즈의 대칭성은  $0^\circ$  이고 피킹 시 임의의 각도를 선택할 수 있습니다. 또한 피킹 포즈 대칭성은 구체적인 계산에 따라 적당한 스텝 길이 ( $1^\circ$  ~  $10^\circ$  등) 를 설정해야 합니다. 하지만 물체 대칭성이 없어서 설정한 물체 포즈대로 배치하려면 소프트웨어가 피킹 시 피킹 툴 및 물체 간의 상대 변환을 복구할 것입니다.

## 1.9 피킹 툴 옵셋

로봇이 비전 결과의 유도에 따라 피킹할 때 사이즈가 매우 작거나 모양이 복잡한 경우에 다른 타겟 물체와의 충돌을 피하기 위해 피킹 툴의 중심점을 물체의 특정한 점으로 옵셋하여 피킹할 수 있습니다. 피킹 툴의 유형 및 옵셋 전략을 확인한 다음에 Mech-Viz 를 통해 옵셋 값을 계산하고 포인트 클라우드 충돌을 결합하여 피킹 가능 여부를 판단합니다. 아래의 그림과 같습니다:



혼합 팔레타이징 경우에 로봇은 물체를 배치할 때 이미 옆에 배치된 물체와의 충돌을 피해야 하며 피킹 툴 옵셋으로 이 문제를 해결할 수 있습니다.



**참고:** 옵션과 관련된 자세한 정보는 Mech-Viz 사용 가이드 `visual_move` 를 참고하세요.

## 1.10 충돌 모델 유형

Mech-Viz 의 충돌 모델 유형은 주로 네 가지가 있습니다:

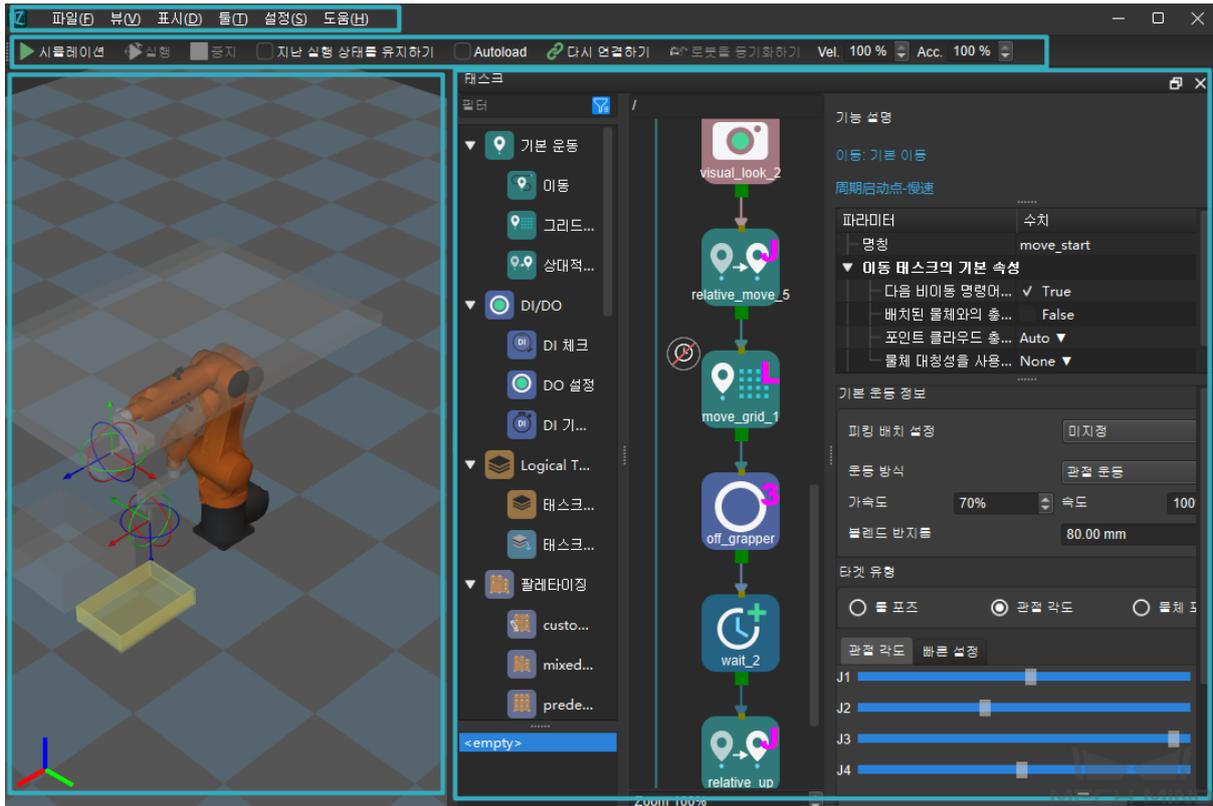
1. 기본 입체 (현재는 입방체만 가능): 이 유형의 모델은 주로 시나리오의 입방체 모델과 충돌 모델을 로드하지 않고 3D 모델만 로드한 엔드 이펙터입니다.
2. Mesh 표면 정보만 감지됨 : 이 유형의 모델은 물체 표면만 대해 충돌 감지를 하고 주요한 모델은 로봇의 관절 및 STL 충돌 모델을 로드한 엔드 이펙터입니다.
3. 볼록 다면체 어셈블리: 이 유형의 모델은 솔리드 물체로 충돌 감지를 수행합니다. 주요한 모델은 OBJ 충돌 모델을 로드한 엔드 이펙터입니다.
4. 팔진트리: 이 유형의 모델은 주로 포인트 클라우드 및 *binvox* 충돌 모델을 로드한 엔드 이펙터와 시나리오입니다.

## 2.1 사용자 인터페이스

다음으로 소프트웨어의 주요 구성 및 기능을 위주로 간단하게 설명하겠습니다.

### 2.1.1 메인 페이지

메인 인터페이스가 아래 그림과 같습니다.



Mech-Viz 시스템 메인 페이지

메인 페이지에는 다음과 같은 부분이 포함되어 있습니다.

1. 메뉴 바
2. 제어 툴 바
3. 3D 시뮬레이션 구역
4. 태스크 편집 구역

총 네 가지의 구역이 포함되어 있습니다. 각 구역의 시너지를 통해 로봇의 시뮬레이션 운동, 실제 제어 및 그래픽 프로그래밍을 실현할 수 있습니다.

### 2.1.2 3D 시뮬레이션 구역

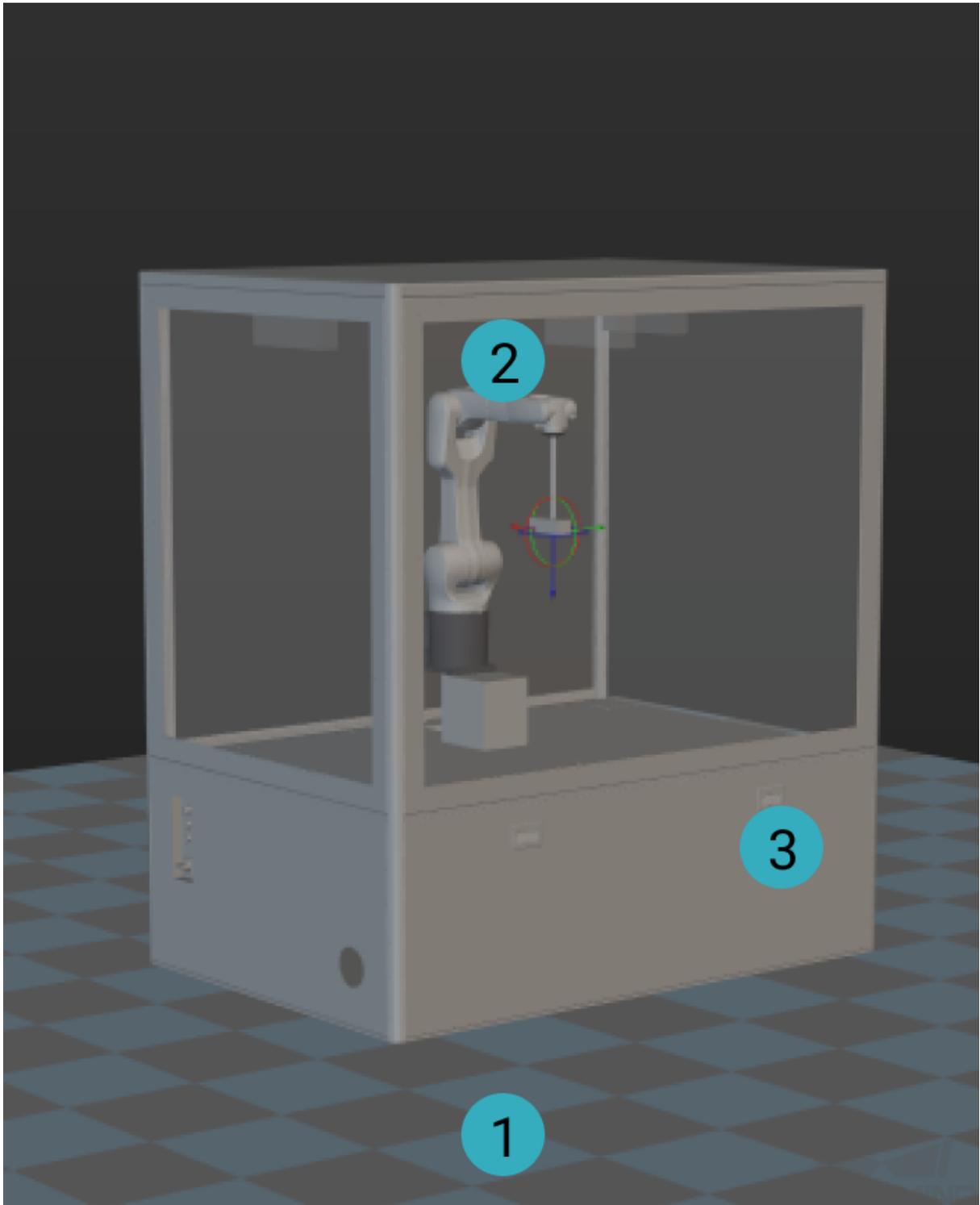
3D 시뮬레이션 구역에는 로봇 이동 궤적 표시, 충돌 예측 강조 표시 및 포인트 클라우드와 피킹 포즈 표시 등 로봇이 리얼 혹은 시뮬레이션 실행 과정속의 상태를 실시간으로 표시합니다.

#### 시나리오 물체를 표시하기

아래의 그림과 같이 3D 시뮬레이션 구역은 주로 세 가지의 부분으로 구성됩니다.

1. 그리드 모양의 바닥
2. 들여온 로봇
3. 들여오거나 새로 만든 시나리오 혹은 엔드 이펙터 모델

마우스의 휠로 표시 구역의 전체를 확대하거나 축소할 수 있습니다. 표시 구역의 임의의 위치를 마우스 왼쪽 버튼으로 클릭하고 마우스 왼쪽 버튼을 누른 채 드래그하면 표시 구역의 시각을 조정합니다. 또한 마우스 가운데 버튼을 누르면 표시 구역의 시각을 평행 이동할 수 있습니다.



시나리오 물체를 표시하기

## 이동 궤적을 표시하기

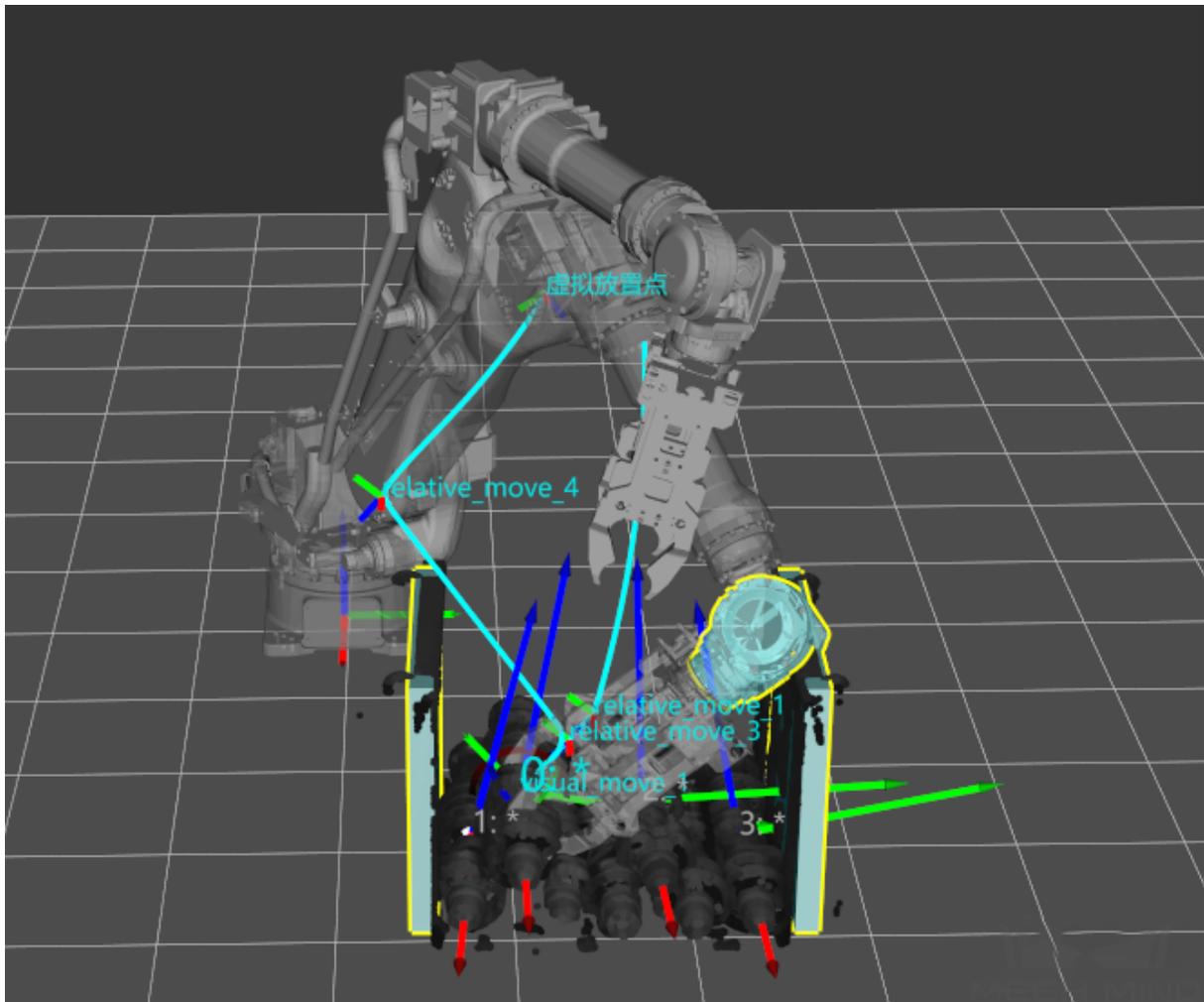
리얼 로봇을 제어하거나 가상 로봇을 시뮬레이션 할 때 3D 시뮬레이션 구역에는 로봇이 이동할 궤적을 표시할 수 있습니다. **실행 과정 시뮬레이션** 과 같이 사용자들은 시뮬레이션 궤적을 통해 로봇의 이동 위치를 더 합리적으로 계획할 수 있습니다.

## 충돌 예측을 표시하기

그림**실행 과정 시뮬레이션** 과 같이 태스크를 수행하는 과정에 viz 가 계획된 궤적에 있는 타겟 물체, 말단 및 자체에 대해 충돌 감지를 할 것입니다. 만약에 충돌 가능성이 있으면 3D 시뮬레이션 구역에 강조 표시 될 것이고 더 합리적인 궤적을 선택하여 로봇 이동을 제어합니다.

## 클라우드와 피킹 포즈를 표시하기

비전 서비스를 실행한 다음에 한번 캡처하면 3D 시뮬레이션 구역에 물체의 포인트 클라우드 및 피킹 포즈의 표시가 업데이트될 것입니다. 구체적인 정보는 피킹 순서, 물체 유형과 정/반방향 등등 있습니다.



실행 과정 시뮬레이션

### 2.1.3 태스크 편집 구역

태스크 편집 구역은 시나리오 (Scene), 태스크 (Mission), 로봇 (Robot) 및 로그 (Log) 의 4 개 탭으로 구성되어 있으며 시나리오에 있는 물체, 로봇 및 엔드 이펙터 모델을 만들 수 있고 그래픽 프로그래밍을 사용하여 이동 프로그램을 작성할 수 있으며 전역 변수를 설정하거나 조정할 수 있습니다. 4 개의 탭 페이지를 드래거하여 표시 순서를 조정하거나 화면에 띄울 수 있습니다. 각 탭의 기능은 아래와 같습니다.

#### 로봇

프로젝트를 만들 때 보통 먼저 해야 할 것은 로봇을 들여오는 것입니다. 로봇 탭은 가상 로봇을 만들고 이미 연결된 리얼 로봇을 제어하는 데에 쓰입니다. 로봇 탭의 기능은 아래의 그림과 같습니다:

1. 로봇을 선택합니다. 사용자들은 이 페이지위에 있는 폴다운 메뉴에서 로봇의 브랜드 및 모델을 선택하여 해당 가상 로봇이 추가되고 3D 시뮬레이션 구역에 표시될 것입니다.
2. 리얼 로봇을 동기화 이동합니다. 리얼 로봇을 이미 연결한 상황에서 리얼 로봇을 가상 로봇의 현재 포즈로 이동하도록 제어할 수 있습니다.
3. 가상 로봇의 포즈를 조정합니다. 4 축 혹은 6 축 로봇은 모두 **관절 각도** 나 **TCP 포즈** 를 바꿈으로써 로봇 포즈를 조정할 수 있습니다.

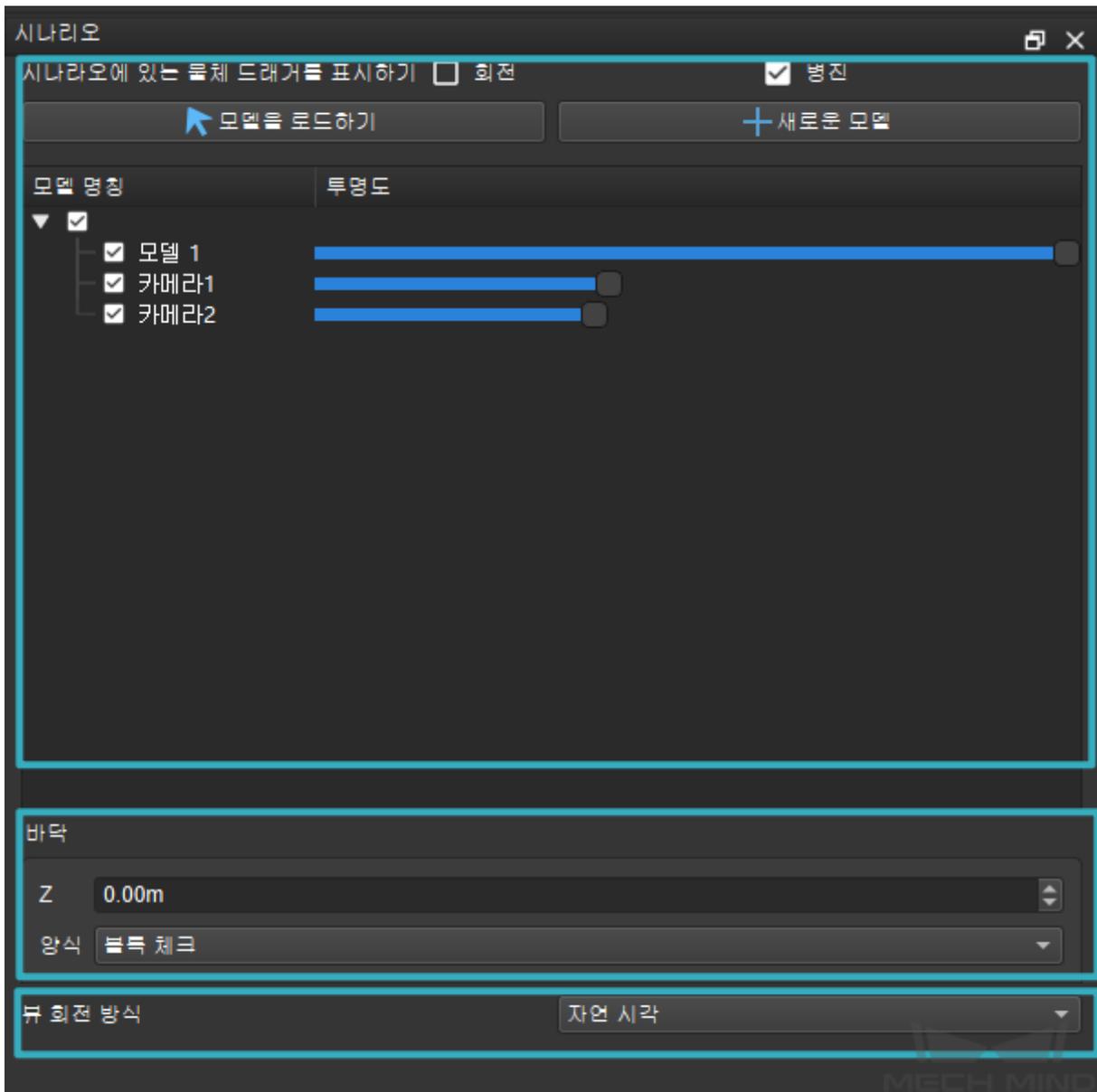


로봇과 관련된 파라미터를 설정하기

## 시나리오

시나리오 탭의 주요 기능은 로봇 작업의 전체 시나리오 (작업대, 작업 공간 등) 를 구축하는 것입니다. 아래의 그림과 같습니다:

1. 시나리오에 있는 물체의 모델을 로드하거나 새로 만들 수 있고 모델의 투명도를 조정할 수 있으며 시나리오 모델을 추출할 수 있습니다.
2. **타겟 물체를 편집하기** 옵션에 피킹 또한 배치 시 최적의 궤적과 최소 충돌 피킹 포즈를 을 계획하기 위해 타겟 물체의 대칭성 및 대칭 범위를 설정할 수 있습니다. 자세한 정보는 **차겟 물체 구성 (궤적 계획에 사용됨)** 를 참고하세요.
3. **지면 높이**: 시나리오의 지면 높이를 조정함으로써 물체 간의 상대 위치 관계를 조정할 수 있습니다.



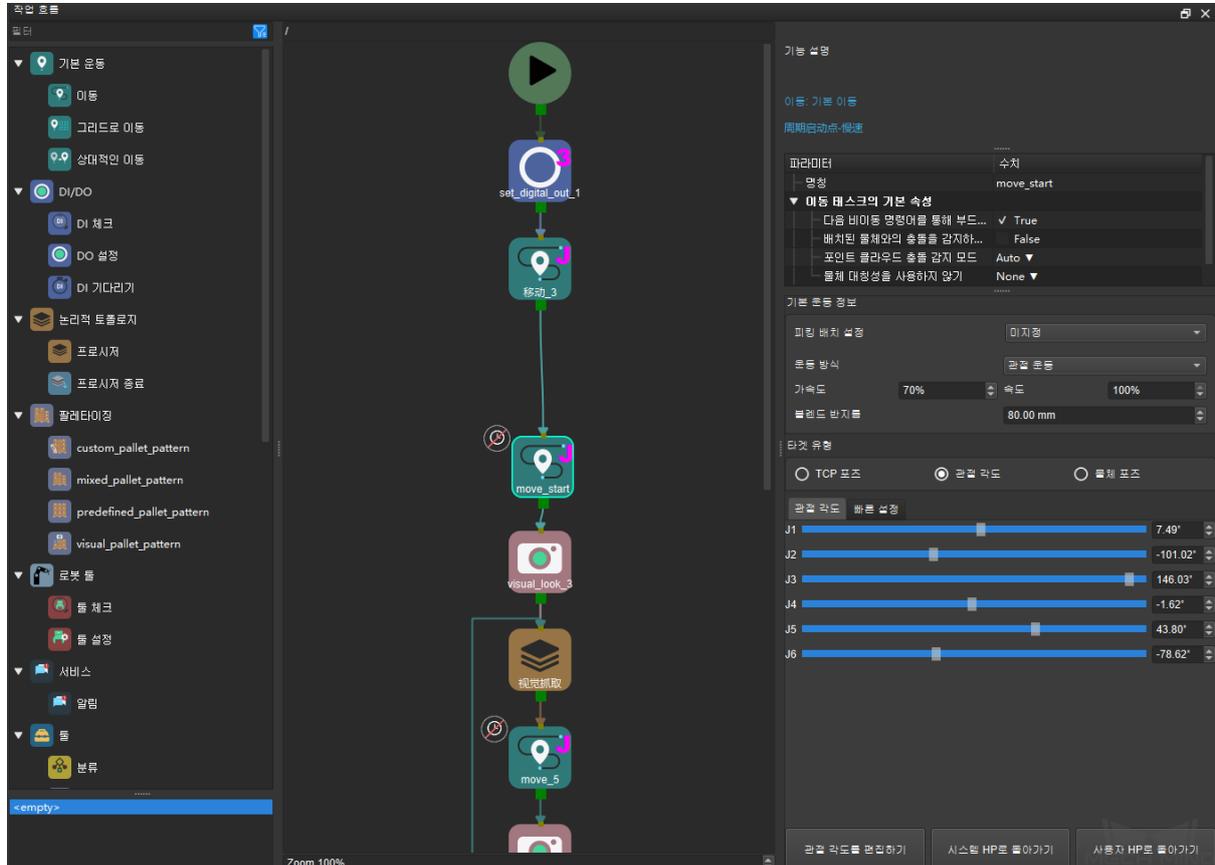
시나리오 옵션의 주요 기능에 대하여

- **모델을 로드하기** 이 옵션을 클릭하여 STL DAE 등 형식의 디지털 모델을 로드할 수 있습니다.
- **새로운 모델** 해당 시나리오에서 간단한 디지털 모델을 직접 만들 수 있습니다.

## 태스크

태스크 탭에는 주로 로봇 프로그래밍과 관련된 내용입니다.

아래의 그림과 같이 모듈 구역, 프로젝트 편집 구역 및 파라미터 설정 구역 총 세 가지의 기능 구역으로 나뉩니다.

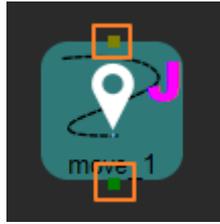


태스크 옵션의 주요 기능에 대하여

1. 모듈 구역에 다양한 기능을 갖는 그래픽 프로그래밍 모듈이 내장되어 있으며 모듈 구역에 있는 프로그래밍 모듈을 프로젝트 편집 구역으로 드래그하고 파라미터 설정 구역에서 해당 모듈의 파라미터를 설정한 다음에 각 프로그래밍 모듈을 연결하면 예상한 프로그램 기능을 실현할 수 있습니다.

**팁:** 모듈 구역 밑에 실행 기록으로 디버깅 과정을 직관적으로 보여줍니다.

1. 프로젝트 편집 구역은 프로그래밍의 실현 구역이고 해당 프로그래밍 모듈을 배열하고 일정한 순서대로 연결하여 프로그램 전체의 작성을 완성합니다. 아래의 그림과 같이 모든 모듈은 입력단 (갈색 사각형) 및 출력단 (녹색 사각형) 을 가지고 있으며 출력단을 클릭하고 연결선을 다른 모듈의 입력단으로 드래그하면 프로그래밍 모듈의 상호 연결을 실현할 수 있습니다.



### 프로그래밍 모듈의 입력 & 출력단

2. 프로젝트 편집 구역에 어떤 모듈을 선택하여 파라미터 설정 구역에서 해당 모듈의 기능을 확인할 수 있고 모듈의 관련 파라미터를 조정할 수 있습니다. 이외에도 파라미터 설정 구역에서 충돌과 특이 점을 피하는 전역 변수를 조정할 수 있습니다. 자세한 정보는 Mech-Viz 사용 가이드 [태크 페이지 소개](#) 를 참고하세요.

### 로그

로그 탭에는 소프트웨어 실행 과정의 로그 정보를 표시하여 디버깅 혹은 신속한 포지셔닝에 편리를 줄 수 있습니다.

```

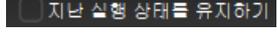
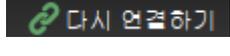
로그
14:23:05.522[I] 서버 UR_5~simu 감시 0.0.0.0:61017
14:23:05.626[W] 서비스 로그인 UR_5~simu failed RPC 실패: registerService 127.0.0.1:5307 Code: 4;
Message: Deadline Exceeded
14:23:05.650[I] Hub 접속 시작 상태: 127.0.0.1:5307
14:23:06.142[W] 서비스 로그아웃 UR_5~simu failed RPC 실패: unregisterService 127.0.0.1:5307 Code:
4; Message: Deadline Exceeded
14:23:06.143[I] 서버 ~simu 감시 0.0.0.0:61019
14:23:06.244[W] 서비스 로그인 ~simu failed RPC 실패: registerService 127.0.0.1:5307 Code: 4;
Message: Deadline Exceeded
14:23:18.465[W] C:/Users/mech-mind_284/Desktop/viz projects/连杆抓取/连杆抓取.viz 열기 실패: 系统找
不到指定的文件。 C:/Users/mech-mind_284/Desktop/viz projects/连杆抓取/连杆抓取.json 열기 실패: 系统
找不到指定的文件。
14:23:25.304[W] 서비스 로그아웃 ~simu failed RPC 실패: unregisterService 127.0.0.1:5307 Code: 14;
Message: Connect Failed
14:23:25.305[I] 서버 KUKA_KR6_R900_SIXX~simu 감시 0.0.0.0:61048
14:23:25.406[W] 서비스 로그인 KUKA_KR6_R900_SIXX~simu failed RPC 실패: registerService
127.0.0.1:5307 Code: 4; Message: Deadline Exceeded
14:23:26.409[I] 열었습니다 C:/Users/mech-mind_284/Desktop/viz projects/连杆抓取/kuka_liangan_viz
14:23:47.936[W] 서비스 로그아웃 KUKA_KR6_R900_SIXX~simu failed RPC 실패: unregisterService
127.0.0.1:5307 Code: 4; Message: Deadline Exceeded
14:23:47.937[I] 서버 KUKA_KR6_R900_SIXX~simu 감시 0.0.0.0:61079
14:23:48.040[W] 서비스 로그인 KUKA_KR6_R900_SIXX~simu failed RPC 실패: registerService
127.0.0.1:5307 Code: 4; Message: Deadline Exceeded
14:23:48.722[I] 열었습니다 C:/Users/mech-mind_284/Desktop/viz projects/连杆抓取/kuka_liangan_viz
14:23:55.405[W] Load icon failed: C:/mmind/Mech-Viz/resource/icons/disc_pallet.png
    
```

글자 크기: 9    최대 블록 카운트: 10000    클리어    내보내기    로그 폴더를 열기  
 로봇    시나리오    태스크    엔드 이펙터와 물체    충돌 감지    로그

로그를 표시하기

### 2.1.4 제어 툴 바

제어 툴 바는 주로 Mech-Viz 프로젝트의 시뮬레이션 및 리얼 로봇과의 통신 & 연결을 제어하는 데 쓰입니다. 주요 기능은 다음과 같습니다:

아이콘	옵션	설명
	시뮬레이션	가상 로봇이 설정한 태스크에 따라 이동하게 만드는 것입니다.
	실행	리얼 로봇이 설정한 태스크에 따라 이동하게 만드는 것입니다.
	중지	현재 실행 중인 태스크를 중지합니다.
	지난 번의 실행 상태를 유지합니다.	팔레타이징 시나리오 디버깅에 쓰입니다. 이미 배치된 위치부터 계속해서 배치합니다.
	다시 연결합니다.	Mech-Center 와 연결하여 로봇과의 데이터 통신을 실현합니다.
	로봇을 동기화하기	가상 로봇의 포즈를 리얼 로봇과 동기화합니다.
	로봇 속도	슬라이드 블록 혹은 입력 박스를 통해 로봇의 속도를 조정합니다.

### 2.1.5 메뉴 바

메뉴 바의 내용은 다음과 같습니다:

옵션	설명
파일	프로젝트를 열기, 프로젝트를 저장하기, 프로젝트를 다른 이름으로 저장하기, 최근에 실행했던 프로젝트, 프로젝트 폴더를 열기 등 다양한 단축키 옵션을 포함합니다.
뷰	각종의 표시 설정과 관련된 옵션을 포함합니다.
표시하기	3D 시뮬레이션 구역의 표시 설정을 포함합니다.
툴	각종의 파라미터 설정과 관련된 옵션을 포함합니다.
설정	Mech-Viz 시스템 설정
도우	언어를 바꾸는 버튼, 버전 정보 및 로그 표시 레벨 등 정보를 포함합니다.

**참고:** 메뉴 바에 관한 자세한 정보는 [메뉴 바 소개](#) 를 참고하세요.

### 2.1.6 사용 방법

마우스

단번 클릭

마우스의 왼쪽 버튼을 눌러 모듈을 선택하거나 활성화시킬 수 있습니다. 어떤 모듈을 클릭하면 모듈의 테두리가 강조 표시되고 현재 모듈의 로봇 포즈와 더미 패턴은 3D 시뮬레이션 구역에서 표시될 것입니다.

## 더블 클릭

- 3D 시뮬레이션 구역에서 모델을 더블 클릭하거나 시나리오 탭에서 모델 명칭을 더블 클릭하면 편집 박스가 나타날 것이고 모델 포즈를 조정할 수 있습니다.
- 다단계 태스크의 경우 마우스 왼쪽 버튼을 더블 클릭하여 하부 인터페이스를 열 수 있습니다.

## 마우스 오른쪽 버튼

- 태스크 탭에 있는 프로젝트 편집 구역에 마우스 오른쪽 버튼을 한번 클릭하면 아래의 그림과 같은 페이지가 나타날 것입니다. 이 페이지는 주로 다단계 태스크의 상위 및 하위 단계 인터페이스를 전환하고 외부 신호 상태를 다시 설정하여 그래픽 모듈을 깔끔하게 정리하는 데 쓰입니다.



마우스 오른쪽 버튼 바로 가기 메뉴

- 마우스의 오른쪽 버튼으로 어떤 모듈을 선택할 때 이 모듈에 대해 복사하거나 붙일 수 있습니다.

## 마우스 가운데 버튼

3D 시뮬레이션 구역에 마우스의 가운데 버튼을 길게 누르면 구역 시각의 평행 이동을 실현할 수 있습니다.

## 마우스 휠

- 3D 시뮬레이션 구역에 마우스 휠을 누르면 전체 표시 구역을 확대하거나 축소할 수 있습니다.
- 프로젝트 편집 구역에 *ctrl* + 마우스 휠을 길게 누르면 프로그램 내용의 표시 화면을 확대하거나 축소할 수 있습니다.
- 파라미터 편집 구역에 모든 편집 박스의 내용은 마우스 휠로 통해 신속하게 사이즈를 조정할 수 있습니다.

## 드래거

- 3D 시뮬레이션 구역에 임의의 위치에 마우스 왼쪽 버튼을 누르고 드래거하면 해당 구역 시각을 조정할 수 있습니다. 드래거에서 *ctrl* + 마우스 왼쪽 버튼을 누르고 드래거하면 로봇 포즈를 조정할 수 있습니다;
- 태스크 편집 구역에서 마우스 왼쪽 버튼을 누르면 모듈을 프로젝트 편집 구역의 임의의 위치로 드래거할 수 있으며 연결선을 드래거할 수도 있습니다. 슬라이더 블록을 드래거함으로써 로봇의 각 관절 각도의 크기 및 TCP 좌표계 아래의 오일러 각을 수정할 수 있습니다.

## 키 보드 단축키

키 보드의 단축키는 시스템과 일치합니다.

단축키	기능
F5	시뮬레이션
Ctrl+R	실행
Ctrl+C	복사하기
Ctrl+V	붙이기
Ctrl+Z	돌아가기
Ctrl+Y	취소하기
Ctrl+O	프로젝트를 열기
Ctrl+S	프로젝트를 저장하기
Delete	삭제하기
PgUp	지난 단계
PgDn	다음 단계

## 2.2 메뉴 바 소개

### 2.2.1 파일

《파일》탭은 프로젝트 및 태스크와 관련된 내용을 처리하는 데에 쓰입니다.

옵션	설명	단축키
프로젝트를 열기	원하는 프로젝트 파일을 엽니다.	Ctrl + O
프로젝트를 저장하기	현재까지 프로젝트에 대한 변경 사항을 저장합니다.	Ctrl + S
프로젝트를 JSON 파일의 포맷으로 저장하기	프로젝트를 저장하고 프로젝트 폴더안의.viz 파일을.json으로 저장합니다.	Ctrl + Shift + S
프로젝트를 다른 이름으로 저장하기	프로젝트를 지정한 위치로 저장합니다.	없음
최근에 실행했던 프로젝트	열면 최근에 실행했던 프로젝트의 명칭을 볼 수 있고 프로젝트 명칭을 클릭하면 열 수 있습니다.	없음
프로젝트 폴더를 열기	사용 중인 프로젝트가 있는 폴더를 엽니다.	없음
실행 가능한 파일의 폴더를 열기	현재 실행 중인 Mech-Viz 소프트웨어가 있는 폴더를 엽니다.	없음
백업 프로젝트	실행 중인 프로젝트를 지정한 위치로 백업합니다.	없음
프로젝트를 닫기	Mech-Viz 페이지에서 현재 실행 중인 프로젝트를 닫습니다.	없음
종료	Mech-Viz 를 종료하고 닫습니다.	Ctrl + Q

**참고:** Mech-Viz v1.4.0 버전은 실행한 프로젝트 파일을.viz 형식으로 저장하고 이전 버전의.json 파일에 대응합니다.

## 2.2.2 뷰

《뷰》탭은 인터페이스 표시와 관련된 설정을 수정하는 데에 쓰입니다. 옵션을 선택하면 Mech-Viz 페이지 오른쪽 밑에 대응하는 탭을 표시할 것입니다 (기능 설명 구역은 태그 페이지에 표시될 것입니다).

옵션	설명	단축키
풀 스크린	선택 가능한 것이고 이 옵션을 선택하면 현재 페이지를 풀 스크린으로 볼 수 있습니다.	Ctrl + Shift + F11
시나리오	기본적으로 선택된 것이고《시나리오》탭을 표시하는 것입니다.	없음
태스크	기본적으로 선택된 것이고《태스크》탭을 표시하는 것입니다.	없음
로봇	기본적으로 선택된 것이고《로봇》탭을 표시하는 것입니다.	없음
엔드 이펙터와 물체	기본적으로 선택된 것이고《엔드 이펙터와 물체》탭을 표시하는 것입니다.	없음
충돌 감지	기본적으로 선택된 것이고《충돌 감지》탭을 표시하는 것입니다.	없음
계획 기록	기본적으로 선택된 것이고《계획 기록》탭을 표시하는 것입니다.	없음
기타	기본적으로 선택된 것이고《기타》탭을 표시하는 것입니다.	없음
로그	기본적으로 선택된 것이고《로그》탭을 표시하는 것입니다.	없음
기능 설명	선택 가능한 것이고 이 옵션을 선택하면 태스크 태그 페이지에 선택한 태스크의 기능 설명이 표시될 것입니다.	없음
돌아가기	새웠던 지난 페이지로 돌아갑니다.	Alt + Left
계속하기	다음 사용할 페이지로 다시 들어갑니다. 돌아가기의 역동작입니다.	Alt + Right

### 2.2.3 표시

《표시》탭은 Mech-Viz 왼쪽 3D 모델의 디테일을 표시하는지 여부를 설정하는 데에 쓰입니다.

옵션	설명
접수된 비전 포즈를 표시하기	기본적으로 선택된 것이고 접수된 비전 포즈를 표시할 것입니다.
포인트 클라우드를 표시하기	기본적으로 선택된 것이고 Vision 에서 보내는 포인트 클라우드를 표시할 것입니다.
피킹된 물체를 표시하기	기본적으로 선택된 것이고 사이즈가 이미 계산된 타겟 물체 모델을 표시할 것입니다.
계획 시 충돌을 표시하기	기본적으로 선택된 것이고 3D 모델에 계획 시 감지된 충돌 위치를 표시할 것입니다.
팔진트리를 표시하기	선택 가능한 것이고 선택하면 물체의 포인트 클라우드가 팔진트리의 형식으로 표시될 것입니다.
접수된 상자 모델을 표시하기	기본적으로 선택된 것이고 3D 모델에 접수된 상자의 모델을 표시할 것입니다.
물체 중심을 가리키는 화살표를 표시하기	선택 가능한 것입니다.
물체 포즈를 표시하기	선택 가능한 것이고 3D 모델에 타겟 물체의 포즈를 표시할 것입니다.
표시 설정	더 많은 3D 모델을 설정하여 디테일을 표시합니다.
포즈의 상태 컬러	다른 컬러가 대표하는 포즈 상태를 확인합니다.
이펙터 상태를 표시하기	선택 가능한 것이고 선택하면 태스크 모듈의 아이콘에 이펙터의 상태를 표시할 것입니다.

### 2.2.4 툴

《툴》탭은 소프트웨어의 부가기능을 설정할 수 있으며 디버깅에 많이 쓰입니다.

옵션	설명	단축키
저장된 비전 기록을 사용하기	선택 가능한 것이고 선택하면 저장된 비전 기록을 사용하여 프로젝트를 실행할 것입니다.	없음
비전 기록 설정	디버깅 시 반복되는 문제를 해결하기 위해 비전 기록을 저장하고 호출합니다.	없음
비전 결과를 찾기	3D 모델 페이지에 검색된 번호와 대응하는 비전 결과를 하이라이트로 표시할 것입니다.	없음
로봇에게 보낸 정보를 프린트하지 않기	선택 가능한 것이고 선택하면 로그에 Mech-Viz 에서부터 로봇에게 보낸 정보를 프린트하지 않을 것입니다.	없음
디버깅 파일을 생성하기	개발자들이 문제를 방지 & 해결하는 데에 쓰이는.dmp 파일을 생성할 것입니다.	없음
취소	방금 했던 설정을 취소합니다.	Ctrl + Z 혹은 Alt + T + U
재설정	방금 했던 설정을 다시 합니다.	Ctrl + Y 혹은 Alt + T + R

### 2.2.5 설정

《설정》탭은 자주 사용하는 소프트웨어의 설정을 수정하는 데에 쓰입니다.

옵션	설명	단축키
Mech-Center 주소 설정	Mech-Center 의 IP 주소를 설정합니다.	없음
프로젝트 잠금/잠금 해제	선택 가능한 것이고 선택하면 현재 실행 중인 프로젝트를 잠금할 것입니다.	없음
자주 사용하는 파라미터만 표시하기	기본적으로 선택된 것이고 선택하면 태스크 표시줄에 자주 사용하는 파라미터만 표시할 것입니다.	Ctrl + Shift + P
자주 사용하는 태스크만 표시하기	기본적으로 선택된 것이고 선택하면 태스크 모듈 구역에 자주 사용하는 태스크만 표시할 것입니다.	Ctrl + Shift + T
옵션	소프트웨어의 기본적인 설정 (언어, 단위 등) 을 수정하는 데에 쓰입니다.	없음
로그 레벨	로그 레벨을 설정하고 콘솔은 현재 및 이상 레벨의 로그를 프린트할 것입니다.	없음

### 2.2.6 도움

《도움》탭은 현재 소프트웨어의 버전 업데이트를 확인할 수 있습니다.

옵션	설명
Mech-Viz 에 대하여	현재 사용하는 버전 번호를 확인하는 데에 쓰입니다.
로그 업데이트	로그 업데이트의 인터넷 홈페이지를 엽니다.

#### 계획 기록을 표시하기

이 기능은 소프트웨어 계획의 전반 과정을 완전하고 상세하게 기록하기 때문에 로그에 표시되는 출력 내용보다 계획의 모든 스텝을 상세하게 확인할 수 있습니다 (특히 실패한 스텝). 아래의 그림과 같습니다:

계획 내용	결과	상세 정보	자식 노드 실패한 모든 원인
2022-01-18_16-49-32_315	성공	Start1, Ready_point	
▶ 타겟 포인트의 고정 TCP 포즈의 인...	성공		
▶ 이번 피킹에서부터 다음의 피킹(볼...	성공	Start1, Ready_point	
2022-01-18_16-49-32_536	성공	Ready_point	
▶ 타겟 포인트의 고정 TCP 포즈의 인...	성공		
▶ 이번 피킹에서부터 다음의 피킹(볼...	성공	Ready_point	
2022-01-18_16-49-33_069	성공	Ready_point	
▶ 타겟 포인트의 고정 TCP 포즈의 인...	성공		
▶ 이번 피킹에서부터 다음의 피킹(볼...	성공	Ready_point	
2022-01-18_16-49-33_602	성공	Ready_point	
▶ 타겟 포인트의 고정 TCP 포즈의 인...	성공		
▶ 이번 피킹에서부터 다음의 피킹(볼...	성공	Ready_point	
2022-01-18_16-49-34_148	성공	Ready_point	
▶ 타겟 포인트의 고정 TCP 포즈의 인...	성공		
▶ 이번 피킹에서부터 다음의 피킹(볼...	성공	Ready_point	
▶ 2022-01-18_16-49-55_992	성공	Start1, Ready_point	
▶ 2022-01-18_16-49-56_202	성공	Ready_point	
▶ 2022-01-18_16-49-56_740	성공	Ready_point	
▶ 2022-01-18_16-49-57_279	성공	Ready_point	
▶ 2022-01-18_16-49-57_808	성공	Ready_point	
▶ 2022-01-18_16-51-02_008	성공	Start1, Ready_point	
▶ 2022-01-18_16-51-03_265	성공(실패한 자식 노드 포함)	relative_move_1, relative_move_2, visual_move_1,...	이동 목적에 도달하지 못함.
▶ 2022-01-18_16-51-05_741	성공	move_3, R1, R	
▶ 2022-01-18_16-51-08_533	성공	relative_move_1, relative_move_2, visual_move_1,...	
▶ 2022-01-18_16-51-12_098	성공	Start1, Ready_point	
▶ 2022-01-18_16-51-14_908	성공(실패한 자식 노드 포함)	relative_move_1, relative_move_2, visual_move_1,...	이동 목적에 도달하지 못함.
▶ 2022-01-18_16-51-17_476	성공	move_3, R1, R	
▶ 2022-01-18_16-51-20_267	성공	Ready_point	
▶ 2022-01-18_16-51-22_646	성공	Ready_point	
▶ 2022-01-18_16-51-24_282	성공	Ready_point	
▶ 2022-01-18_16-51-25_803	성공	Ready_point	
▶ 2022-01-18_16-51-41_557	성공	Start1, Ready_point	
▶ 2022-01-18_16-51-42_851	성공(실패한 자식 노드 포함)	relative_move_1, relative_move_2, visual_move_1,...	이동 목적에 도달하지 못함.
▶ 2022-01-18_16-51-45_306	성공	move_3, R1, R	
▶ 2022-01-18_16-51-48_253	성공	relative_move_1, relative_move_2, visual_move_1,...	
▶ 2022-01-18_16-51-51_556	성공	move_3, R1, R	
▶ 2022-01-18_16-51-54_447	성공(실패한 자식 노드 포함)	relative_move_1, relative_move_2, visual_move_1,...	이동 목적에 도달하지 못함.
▶ 타겟 포인트의 고정 TCP 포즈의 인...	성공		
▶ 이번 피킹에서부터 다음의 피킹(볼...	성공	relative_move_1, relative_move_2	
▶ 피킹 계획부터 다음 피킹까지	성공(실패한 자식 노드 포함)	visual_move_1, 相对移动_1, relative_move_3	이동 목적에 도달하지 못함.
▶ 1/1회 피킹 시도	성공(실패한 자식 노드 포함)		이동 목적에 도달하지 못함.
▶ 2022-01-18_16-51-57_451	성공	move_3, R1, R	
▶ 2022-01-18_16-52-00_325	성공	Ready_point	
▶ 2022-01-18_16-52-02_854	성공	Ready_point	
▶ 2022-01-18_16-52-04_482	성공	Ready_point	
▶ 2022-01-18_16-52-06_094	성공	Ready_point	

계획 결과의 전체적인 구조는 트리 구조이고 연속 계획의 오류이든 병행 계획의 오류이든 최종 성공 여부는《값》표시줄에 표시돼서 사용자가 실패한 스텝만 유의하면 됩니다.

- 연속 계획의 오류: 계획 시 일환이 실패되면 전반적인 실패를 초래합니다. 예를 들어 로봇은 이동 태스크를 실행할 때 어떤 관절 각도는 도달할 수 없는 경우입니다.
- 병행 계획의 오류: 계획 시 일환이 실패해도 전반에 영향을 미치지 않으며 성공한 부분만 있으면 전반적으로 성공한다고 여깁니다. 예를 들어 Vision 을 통해 얻은 피킹 포즈 중에 피킹 가능한 것이 하나만 있다면 피킹할 수 있다는 것으로 간주됩니다.

## 루트 노드 설명

- 파라미터: 루트 노드가 시간으로 명명되고 앞에 history 가 붙이지 않은 것은 실시간으로 실행한 결과이며 앞에 history 가 붙이는 것은 페이지 오른쪽 밑에 있는 **계획 기록을 로드하기** 를 클릭하여 로드한 결과 기록입니다.
- 값: 소프트웨어의 계획 결과이며 레벨을 편 후 성공 또는 실패에 대한 구체적인 이유가 표시될 것입니다.
- 설명: 계획의 시작 및 종료 태스크, 그리고 이번 계획 과정속의 태스크의 총 개수를 표시합니다.
- 오류 발생 시 처리 가능 여부: 계획 실패 시 이동 태스크의 두 번째 출구를 통해 오류를 처리할 수 있는지 여부에 대한 설명입니다. 현재까지 두 번째 출구가 있는 이동 태스크는 비전 이동 및 혼합 팔

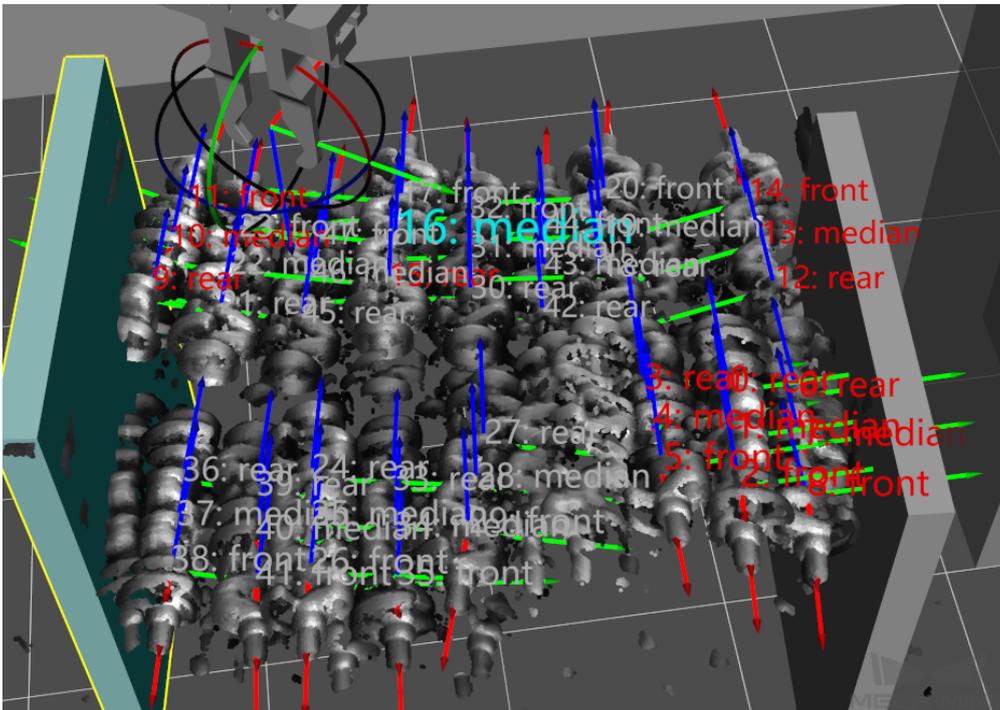
레타이징입니다. 비전 이동의 두 번째 출구는 비전 포즈 혹은 파킹 가능한 포즈가 없는 경우에 사용되며 혼합 팔레타이징의 두 번째 출구는 물체를 이미 피킹을 했지만 기존 더미에 배치하기가 적절한 포즈가 없는 경우에 사용됩니다.

### 계획 기록을 보기

위 그림의 두 번째 계획 기록을 예로 들면 퍼진 후의 구조에 대해 설명해 보겠습니다.

응용 시나리오 소개: 크랭크축 텐딩; 이미지를 한번 캡처하면 여러 개의 크랭크축을 인식할 수 있습니다. 한 크랭크축에 세 개의 피킹 포즈를 가지고 있고 Z 축에는 대칭성이 없지만 X 축 (약한 축) 에는 임의의 각도 대칭성을 가지고 있습니다. 또한 클램프는 180° 혹은 360° 두 가지 가능한 대칭성을 가지고 있습니다.

### 실제 계획 결과



## 기획 결과 전개도

계획 기록의 구체적인 항목을 전개하면 각 피킹 포즈의 구체적인 계획 결과로 신속하게 포지셔닝을 할 수 있습니다.

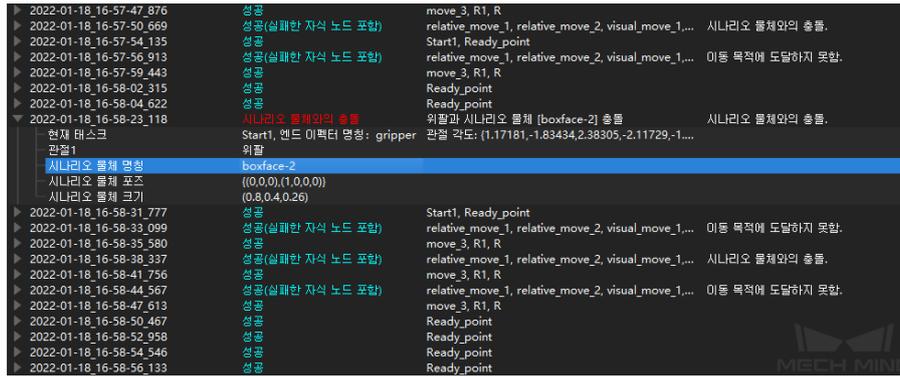
계획 내용	결과	상세 정보	저석 노드 실패한 모든 원동
▶ 2022-02-11_16-04-48_130	성공	移动_2, 拍照点	
▶ 2022-02-11_16-04-55_221	성공	料框上方	
▼ 2022-02-11_16-04-56_143	성공(실패한 저석 노드 포함)	相对移动_20, 视觉移动_2, 相对移动_25, 相对移动_24...	시나리오 문제외의 충돌.
▶ 다섯 포인트의 고정 TCP 포즈의 인...	성공	相对移动_20	
▶ 이번 피킹에서부터 다음의 피킹(불...	성공	视觉移动_2, 相对移动_25, 相对移动_24, 相对移动_23...	시나리오 문제외의 충돌.
▶ 피킹 계획부터 다음 피킹까지	성공(실패한 저석 노드 포함)		시나리오 문제외의 충돌.
▶ 1/24의 피킹 시도	성공(실패한 저석 노드 포함)		시나리오 문제외의 충돌.
▶ "视觉移动_2"에 대 정성/피킹... 대 정성 시도 실패		0° (X축 회전 부분)	시나리오 문제외의 충돌.
▶ "视觉移动_2"에 대 정성/피킹... 대 정성 시도 실패		5° (X축 회전 부분)	시나리오 문제외의 충돌.
▶ "视觉移动_2"에 대 정성/피킹... 대 정성 시도 실패		-5° (X축 회전 부분)	시나리오 문제외의 충돌.
▶ "视觉移动_2"에 대 정성/피킹... 대 정성 시도 실패		10° (X축 회전 부분)	시나리오 문제외의 충돌.
▶ "视觉移动_2"에 대 정성/피킹... 대 정성 시도 실패		-10° (X축 회전 부분)	시나리오 문제외의 충돌.
▶ "视觉移动_2"에 대 정성/피킹... 대 정성 시도 실패		15° (X축 회전 부분)	시나리오 문제외의 충돌.
▶ "视觉移动_2"에 대 정성/피킹... 대 정성 시도 실패		-15° (X축 회전 부분)	시나리오 문제외의 충돌.
▶ "视觉移动_2"에 대 정성/피킹... 대 정성 시도 실패		20° (X축 회전 부분)	시나리오 문제외의 충돌.
▶ "视觉移动_2"에 대 정성/피킹... 대 정성 시도 실패		-20° (X축 회전 부분)	시나리오 문제외의 충돌.
▶ "视觉移动_2"에 대 정성/피킹... 대 정성 시도 실패		25° (X축 회전 부분)	시나리오 문제외의 충돌.
▶ "视觉移动_2"에 대 정성/피킹... 대 정성 시도 실패		30° (X축 회전 부분)	시나리오 문제외의 충돌.
▶ "视觉移动_2"에 대 정성/피킹... 대 정성 시도 실패		-25° (X축 회전 부분)	시나리오 문제외의 충돌.
▶ "视觉移动_2"에 대 정성/피킹... 성공			

그림과 같이 해당 계획 과정에 나타나는 오류는 병행 계획의 오류이고 소프트웨어가 X 축의 대칭성에 의해 총 12 가지의 피킹 포즈를 시도해 봤습니다. 1-11 차례 모두 실패했지만 마지막 차례가 성공했기 때문에 전반적으로 성공한 것으로 여기고 로봇이 정상적으로 피킹할 수 있습니다.

- 대칭성 시도 실패한 경우의 결과 전개도

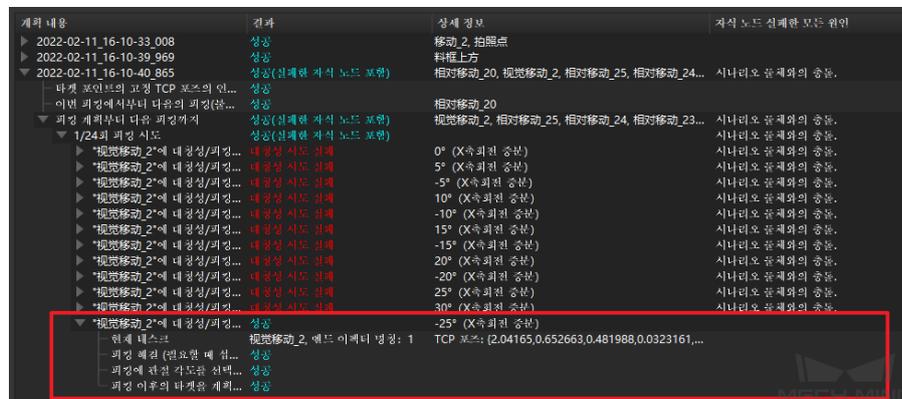
계획 내용	결과	상세 정보	저석 노드 실패한 모든 원동
▶ 2022-02-11_16-04-48_130	성공	移动_2, 拍照点	
▶ 2022-02-11_16-04-55_221	성공	料框上方	
▼ 2022-02-11_16-04-56_143	성공(실패한 저석 노드 포함)	相对移动_20, 视觉移动_2, 相对移动_25, 相对移动_24...	시나리오 문제외의 충돌.
▶ 다섯 포인트의 고정 TCP 포즈의 인...	성공	相对移动_20	
▶ 이번 피킹에서부터 다음의 피킹(불...	성공	视觉移动_2, 相对移动_25, 相对移动_24, 相对移动_23...	시나리오 문제외의 충돌.
▶ 피킹 계획부터 다음 피킹까지	성공(실패한 저석 노드 포함)		시나리오 문제외의 충돌.
▶ 1/24의 피킹 시도	성공(실패한 저석 노드 포함)		시나리오 문제외의 충돌.
▶ "视觉移动_2"에 대 정성/피킹... 대 정성 시도 실패		0° (X축 회전 부분)	시나리오 문제외의 충돌.
▶ 현재 태스크	视觉移动_2, 엔드 이펙터 변경: 1	TCP 포즈: (2.04165, 0.652663, 0.481988, 0.0280149...	시나리오 문제외의 충돌.
▶ "视觉移动_2"의 연결 각도 ...	시나리오 문제외의 충돌	엔드 이펙터 시나리오 문제 [料框] 충돌	시나리오 문제외의 충돌.
▶ 현재 태스크	视觉移动_2, 엔드 이펙터 변경: 1	연결 각도: (0.227215, 0.542611, -0.589674, -0.37994...	시나리오 문제외의 충돌.
▶ 연결1	料框		
▶ 시나리오 문제 변경			
▶ 시나리오 문제 포즈	{(1.87021, 0.793502, 0.476003), (0...		
▶ 시나리오 문제 크기	(1.19, 0.09, 1.04)		
▶ "视觉移动_2"에 대 정성/피킹... 대 정성 시도 실패		5° (X축 회전 부분)	시나리오 문제외의 충돌.
▶ "视觉移动_2"에 대 정성/피킹... 대 정성 시도 실패		-5° (X축 회전 부분)	시나리오 문제외의 충돌.
▶ "视觉移动_2"에 대 정성/피킹... 대 정성 시도 실패		10° (X축 회전 부분)	시나리오 문제외의 충돌.
▶ "视觉移动_2"에 대 정성/피킹... 대 정성 시도 실패		-10° (X축 회전 부분)	시나리오 문제외의 충돌.
▶ "视觉移动_2"에 대 정성/피킹... 대 정성 시도 실패		15° (X축 회전 부분)	시나리오 문제외의 충돌.
▶ "视觉移动_2"에 대 정성/피킹... 대 정성 시도 실패		-15° (X축 회전 부분)	시나리오 문제외의 충돌.
▶ "视觉移动_2"에 대 정성/피킹... 대 정성 시도 실패		20° (X축 회전 부분)	시나리오 문제외의 충돌.
▶ "视觉移动_2"에 대 정성/피킹... 대 정성 시도 실패		-20° (X축 회전 부분)	시나리오 문제외의 충돌.
▶ "视觉移动_2"에 대 정성/피킹... 대 정성 시도 실패		25° (X축 회전 부분)	시나리오 문제외의 충돌.
▶ "视觉移动_2"에 대 정성/피킹... 대 정성 시도 실패		30° (X축 회전 부분)	시나리오 문제외의 충돌.
▶ "视觉移动_2"에 대 정성/피킹... 대 정성 시도 실패		-25° (X축 회전 부분)	시나리오 문제외의 충돌.
▶ "视觉移动_2"에 대 정성/피킹... 성공			

설명을 보기: “원시 Vision-0”이 Mech-Vision 에서부터 보낸 초기 피킹 포즈의 인덱스를 의미합니다. 이 인덱스는 Mech-Viz 가 계획 시 시도한 인덱스와 일치하지 않을 수도 있습니다. 원인은: 크랭크축 피킹 시 반복적으로 같은 위치를 피킹하는 것을 피하기 위해 초기의 피킹 포즈의 우선 순위가 낮아지거나 삭제될 수 있기 때문일 수도 있으며 상자 혼합 다팔레타이징 할 때 Mech-Vision 에서 상지의 피킹 포즈를 산출한 다음에 Mech-Viz 가 자동으로 움직임을 계산할 것이라서 한 상자가 여러 개의 피킹 포즈와 대응할 수 있기 때문일 수도 있습니다. 이러한 원인으로 Mech-Viz 가 시도한 인덱스는 원시 비전 결과와 불일치하는 가중성도 있습니다. 이 인덱스는“실제-0”로 표시되고“총수-48”은 Mech-Viz 가 실제로 시도할 수 있는 피킹 포즈의 총수를 의미합니다.



위 그림에 표시된 실패 원인은 시나리오에 있는 물체와 충돌했기 때문입니다. 충돌이 발생한 관절 인덱스를 확인할 수 있습니다: 6 축 로봇을 예로 들면 인덱스 0 은 로봇의 베이스, 1-6 은 로봇의 1-6 축, 7 은 엔드 이펙터, 8 은 로봇이 피킹한 물체의 뜻입니다. 이 그림은 클램프와 시나리오에 있는 boxface-2 라는 물체와 충돌한다는 뜻입니다.

- 대칭성 시도 성공한 경우의 결과 전개도



계획 결과 유형:

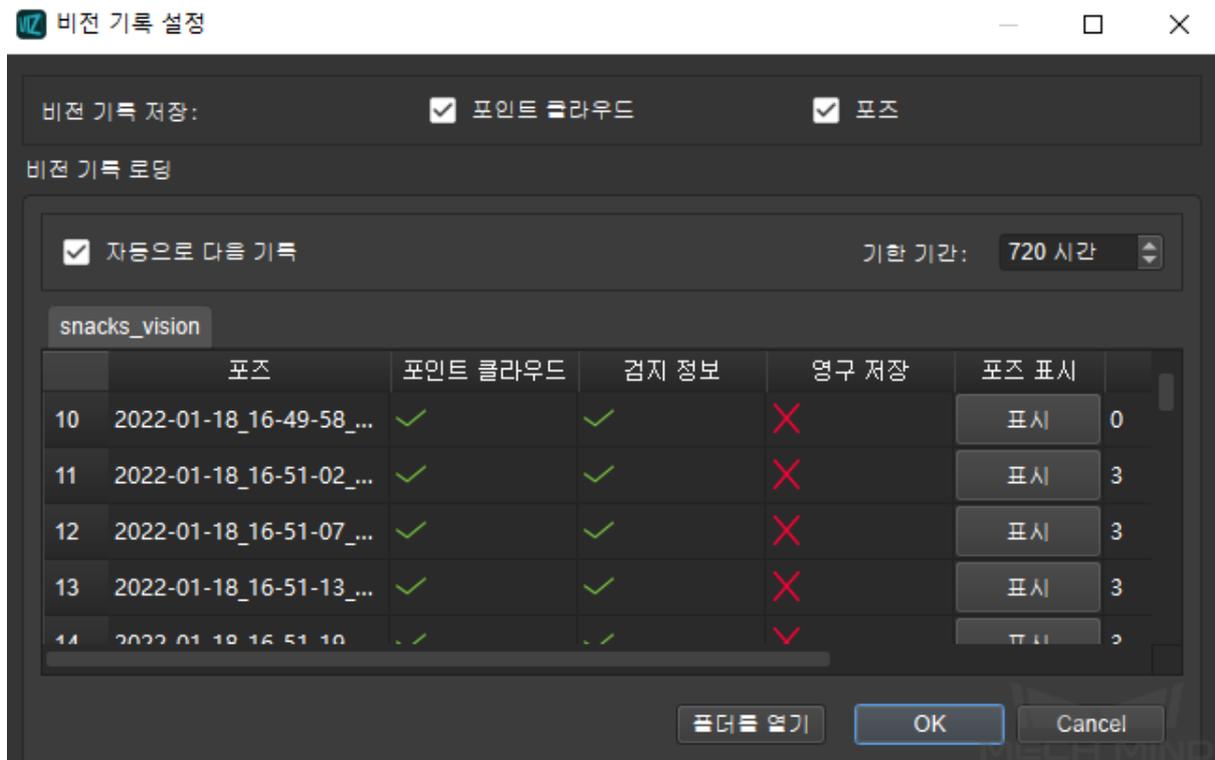
1. 성공
2. 계획될 타겟 없음
3. 최종 태스크는 다음 태스크에 의함
4. 순환 종속
5. 이동 목적에 도달하지 못함
6. 궤적 도달하지 못함
7. 관절 각도 선택 실패
8. 휴리스틱하게 관절 각도 선택 실패
9. 심각한 특이점
10. 직선 운동 계산 결과 불일치
11. 물체 포즈 시도 실패
12. 무효한 피킹 포즈
13. 대칭성 시도 실패
14. 로봇 자체 충돌
15. 시나리오 물체와의 충돌

16. 포인트 클라우드 충돌 포인트수가 제한 초과
17. 포인트 클라우드 충돌 면적 제한 초과
18. 궤적 충돌

성공이 아니라면 모든 상태가 실패로 간주됩니다. 어떤 경우의 실패 원인은 심층적이어서 자세하게 알려면 트리 구조의 더 깊은 노드를 통해 확인해야 합니다.

## 비전 기록 설정

이 기능은 주로 디버깅에 쓰입니다. Mech-Vision 프로젝트 실행 후 보낸 비전 결과를 저장하고 사용자들은 기존의 비전 기록을 직접 사용할 수 있으며 Mech-Vision 프로젝트를 다시 열 필요가 없어서 특정한 위치 및 문제를 찾기가 편리해집니다.



- 비전 기록 저장: 기본적으로 json 파일 식으로 포즈를 자동으로 저장합니다. 포인트 클라우드 데이터를 저장하려면 포인트 클라우드를 선택하면 됩니다. 단, 주의해야 할 것은 포인트 클라우드 파일이 크고 포인트 클라우드가 존재하는 시나리오에 이미지를 한번 캡처하면 하나의 포인트 클라우드 파일이 생성되고 저장될 것입니다. 따라서 불필요하게 하드 디스크 공간을 낭비하지 않도록 포인트 클라우드 옵션은 테스트 기간에만 선택하고 소프트웨어가 안정적으로 실행할 때 언체크해야 합니다.
- 비전 기록 로딩
  - 자동으로 다음 기록 로딩: 선택하지 않으면 이전에 선택한 기록을 다시 사용할 것입니다. 선택할 때, 현재 기록을 사용한 다음에 자동으로 다음 비전 기록으로 넘어갈 것입니다. 마지막 기록까지 사용하면 자동으로 현재 리스트를 순환하지 않고 새로운 비전 기록이 없다는 것으로 끝날 것입니다.
  - 기한 시간: 기한 시간을 설정합니다. 기본값은 9999 시간인데 설정한 시간을 초과하면 파일은 자동으로 삭제될 것입니다.
  - 폴더를 열기: 비전 결과가 저장되어 있는 폴더를 직접 열 것입니다.

- 영구 저장: 영원히 저장하려고 하는 비전 결과를 선택하여 마우스 오른쪽 버튼을 눌러 영구 저장하면 해당 비전 기록은 기한 시간이 지나도 삭제되지 않을 것입니다.

비전 기록을 사용하여 비전 결과가 없는 경우에 신속하게 문제를 재연합니다.

재연하는 방식	필요한 데이터	데이터 저장 경로
비전 기록을 사용하여 비전 결과가 없는 경우에 신속하게 문제를 재연합니다.	비전 포즈	Mech-Viz 소프트웨어 패키지에 있는 vision_records 폴더 안에 있습니다. 각종의 서비스 폴더 아래의 날짜 폴더 안에 문제를 해결하는 데 필요한 하나 혹은 여러 개의 비전 기록 파일을 선택하여 로드합니다. 폴더 예시 vision_records\visionSimu2019-08-05\2019-08-05_173341869.json
	비전 포즈, 포인트 클라우드 정보	Mech-Viz 소프트웨어 패키지에 있는 vision_records 폴더 안에 있습니다. 각종의 서비스 폴더 아래의 날짜 폴더 안에 문제를 해결하는 데 필요한 하나 혹은 여러 개의 비전 기록 파일을 선택하여 로드합니다. 일반적으로 한 그룹에는 3 개의 파일이 있으며 이 3 개의 파일은 접미사를 제외하고 파일 이름이 같습니다. 각각 다음과 같습니다: 비전 포즈 파일 .json    포인트 클라우드 파일 .cloud    및 포인트 클라우드 포즈 파일 .transform    없는 경우도 종종 있음
vision 서비스를 호출해야 재연할 수 있습니다.	이미지 맵스컬러	vision 프로젝트에 캡처된 이미지를 저장하는 경로

## 2.3 태크 페이지 소개

### 2.3.1 시나리오

그림 1과 같이 이 페이지에서 시나리오의 표시와 관련된 정보는 조정할 수 있습니다.

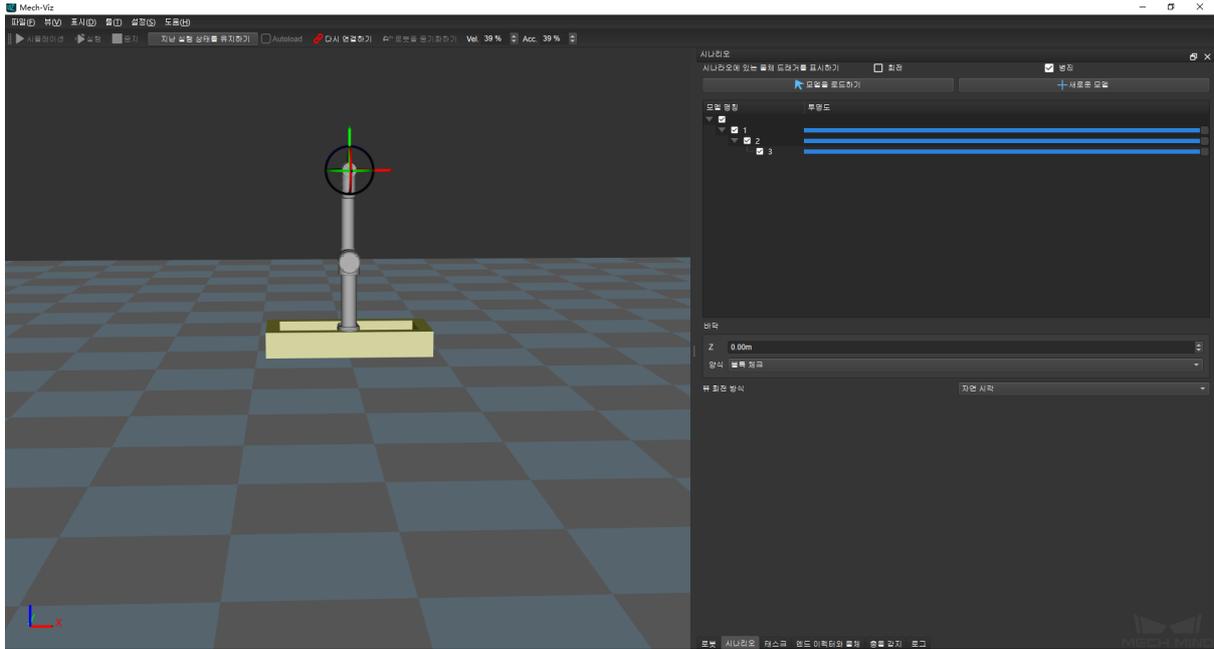


그림 1 시나리오 태그 페이지

### 시나리오에 있는 물체의 드래거를 표시하기 (회전, 병진)

이 옵션을 선택하고 시나리오에 있는 물체를 클릭하면 드래그할 수 있는 좌표축을 표시할 것입니다. 이 때 ctrl 를 눌러 물체를 드래그하면 물체의 병진 혹은 회전을 실현할 수 있습니다. **그림 2**와 같습니다.

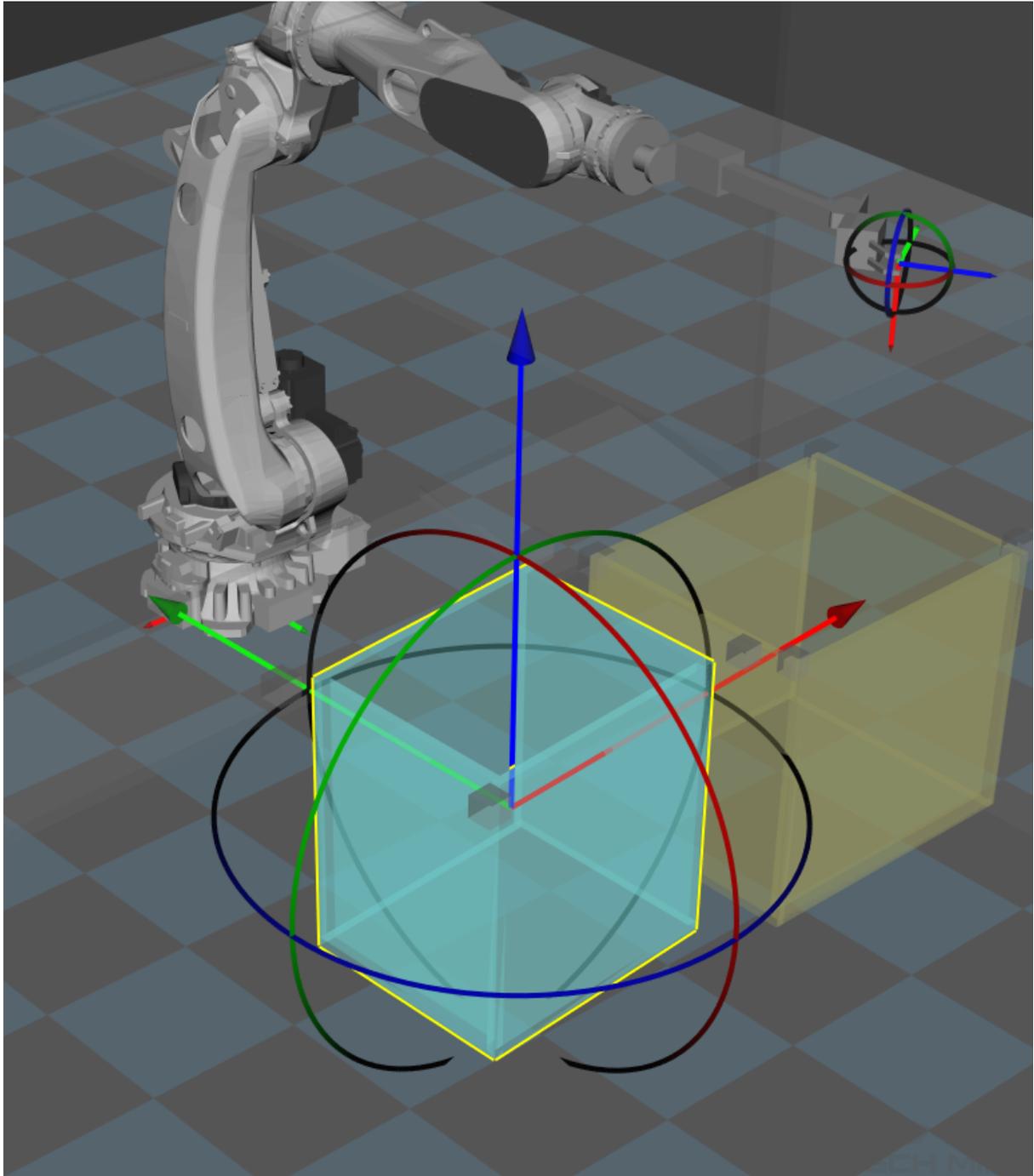


그림 2 드래거

### 모델을 로드하거나 새로 만들기

이 기능은 기존 모델을 로드하거나 새로운 모델을 만드는 데에 쓰이며 새로운 모델을 만들 때 물체의 유형 및 사이즈를 설정할 수 있습니다. 그림 3 과 같습니다.

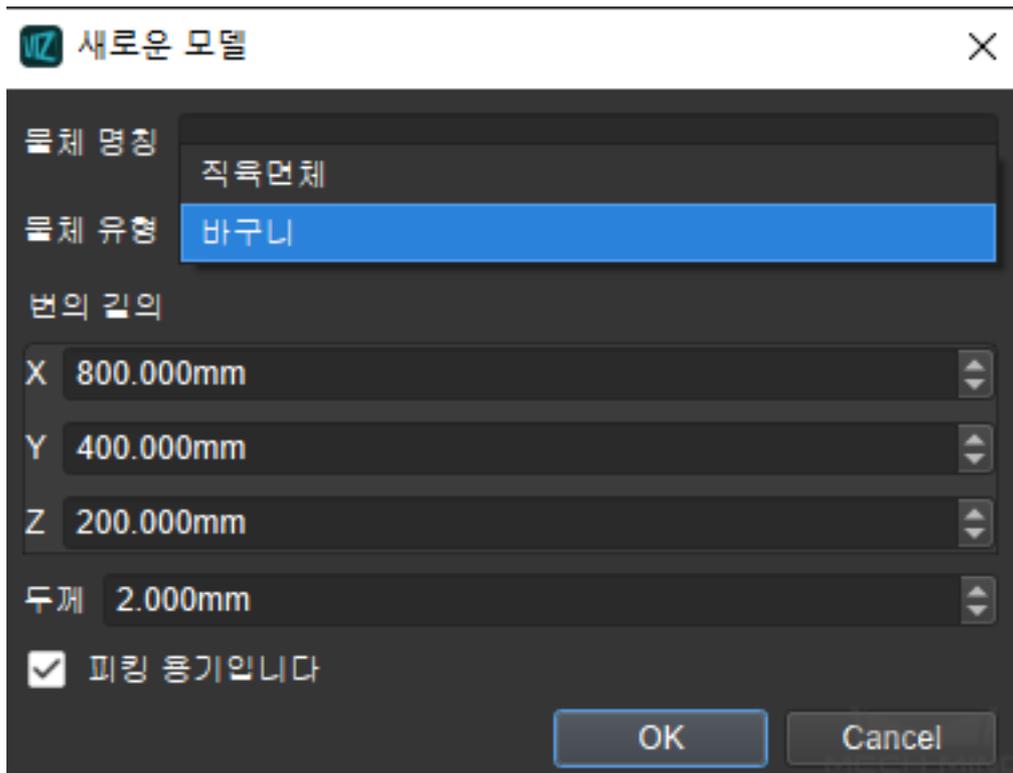


그림 3 새로운 모델

**주의:** 로드된 모델의 사이즈가 시나리오보다 더 큰 경우도 있습니다. 이는 미터와 밀리미터의 단위를 잘 못 전환했기 때문입니다. 이때 해당 모델을 더블 클릭하여 모델 아편에서 모델 스케일을 0.001 로 수정하면 됩니다.

### 시나리오 모델의 복사 & 붙이기

- 마우스 오른쪽 버튼으로 모델 명칭을 클릭하여 Ctrl+C    Ctrl+V 를 통해 복사 또는 일 수 있습니다.
- 하위 클래스는 상위 클래스와 함께 복사하여 붙여 넣습니다.
- 복사 후 여백 공간을 클릭하면 선택이 취소되며 붙여 넣은 후 루트 폴더에 붙여넣을 수 있습니다.
- 복사한 모든 물체 명칭 뒤에 -1 -2 -3 -4 .....를 붙일 것이고 현재 - 뒤에 가장 큰 숫자를 기준으로 하나둘씩 늘어납니다.
- Ctrl 를 눌러 여러 가지를 선택하고 Ctrl+Z 를 눌러 취소하며 Ctrl+Y 를 눌러 복원될 것입니다.



### 투명도 설정

모델을 표시하거나 숨기려면 왼쪽에 있는 작은 사각형을 클릭하면 됩니다. 오른쪽에 있는 가로 막대를 드래그하여 왼쪽에 있는 시나리오에 있는 물체의 투명도를 조정합니다. 그림 4와 같습니다.

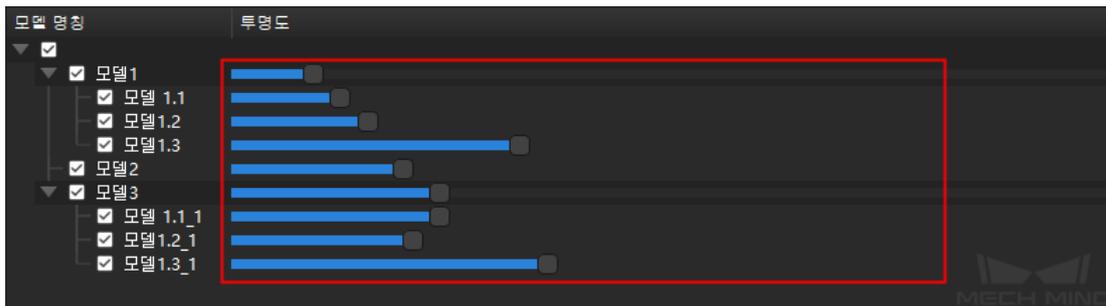


그림 4 투명도 설정

### 바닥 설정

그림 5 와 같이 시나리오에 있는 바닥의 상대 위치 및 모양을 설정할 수 있습니다.

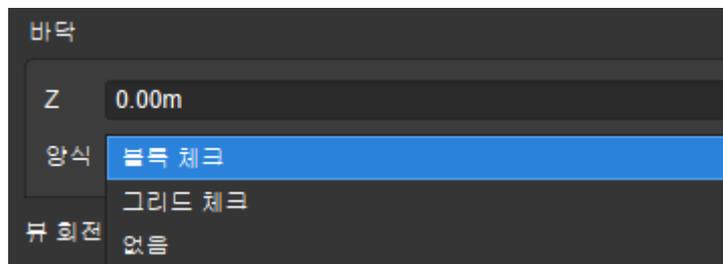


그림 5 바닥 설정

### 시각 회전 방식

자연 시각 및 자유 시각을 설정할 수 있습니다. 두 가지 시각 아래에 마우스 왼쪽 버튼을 통해 시각을 드래그하거나 회전할 수 있으며 시각은 영원히 좌표 원점으로 향하고 있지만 자유 시각은 시각부터 좌표 원점까지의 연결선을 축으로 자유롭게 회전할 수 있습니다.

### 2.3.2 태스크

태스크 페이지는 로봇 프로그래밍을 진행하는 주요 페이지입니다. 그림 1 과 같이 이 페이지는 주로 모듈 구역, 프로세스 기록 구역, 프로젝트 편집 구역 및 속도 설정과 파라미터 편집 구역 등 4 가지의 구역으로 구성됩니다.

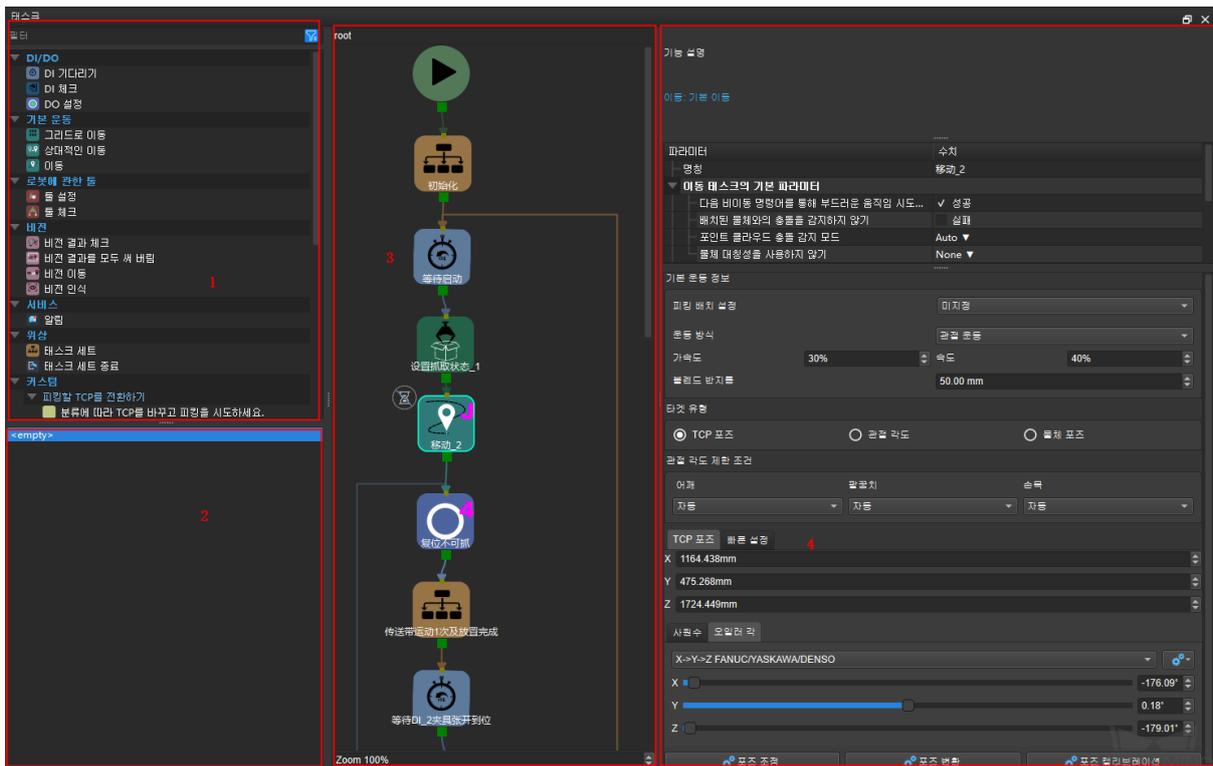


그림 1 태스크 태그 페이지

#### 1. 모듈 구역

모듈 구역에 Mech-Viz 의 다양한 기능을 실현할 수 있는 그래픽 프로그래밍 모듈이 포함되어 있으며 이러한 모듈은 Mech-Viz 에서 로봇 프로그래밍을 하는 필수 조건입니다. 필요한 모듈을 프로젝트 편집 구역으로 드래그하고 관련 파라미터를 설정하여 프로젝트 실행 순서대로 연결하면 다양한 로봇 프로그래밍을 실행할 수 있습니다. 각 모듈의 자세한 정보 및 파라미터 설정 정보는 skills\_guide 를 참고하세요.

#### 2. 프로세스 기록 구역

여기는 프로젝트 편집 구역에서 진행되는 모든 작업을 표시하는 구역입니다. 모듈 혹은 연결선의 추가 및 삭제를 포함하여 어떤 기록을 클릭하면 프로젝트 편집 구역은 해당 기록이 생성된 상태로 돌아갈 수 있습니다.

#### 3. 프로젝트 편집 구역

프로젝트 편집 구역은 각 모듈을 조합하여 로봇 프로그래밍을 실행하는 구역입니다. 중요한 것은 올바른 모듈과 연결선을 선택해야 합니다. 모든 모듈 위에 입력 포트 (갈색 작은 박스)

가 있고 밑에 출력 포트 (녹색 작은 박스) 가 있으며 출력 포트에서부터 입력 포트로 연결선을 통해 연결할 수 있습니다.



여러 개의 모듈을 한 태스크 세트로 조합할 수 있으며 메인 프로그래밍에 하나의 태스크 세트 모듈로 표시되어 프로그래밍이 더 간결하고 명확해질 수 있습니다. 태스크 세트를 두 번 클릭하여 편집할 수 있으며 **지난 페이지** 를 마우스 오른쪽 키로 클릭하여 지난 페이지로 돌아갈 수 있습니다. 또한 특정한 모듈을 빠르게 찾으려면 단축키 Ctrl+F 를 클릭하여 프로젝트 편집 구역 위에 검색창이 나타나서 모듈 명칭을 입력하고 검색하면 됩니다.



4. 속도 설정과 파라미터 편집 구역

모듈이 선택되지 않은 경우 여기에 속도 설정과 관련된 옵션 ( **틀 최대 속도** 및 **전반 속도 비율** ) 만 표시됩니다.

기능 설명

상대적인 이동: 전/후 위치의 상대 위치에 따라 이동하기

---

\*\*\*\*\*

파라미터	수치
명칭	상대적인 이동_1
<b>이동 태스크의 기본 파라미터</b>	
다음 비이동 명령어를 통해 부드러운 움직...	실패
배치된 물체와의 충돌을 감지하지 않기	실패
포인트 클라우드 충돌 감지 모드	Auto ▼
물체 대칭성을 사용하지 않기	None ▼

\*\*\*\*\*

기본 운동 정보

운동 방식: 관절 운동 ▼

가속도: 50% ▼ 속도: 100% ▼

블렌드 반지름: 50.00 mm ▼

(이동 태스크)에 비해

현재  다음으로  옵션

좌표계

TCP  로봇  참고 포인트

좌표 빠른 설정

X 0.000mm ▼

Y 0.000mm ▼

Z 0.000mm ▼

X 잠기기  Y 잠기기  Z 잠기기

≡ 좌표계 아래의 상대적인 회전

사원수 오일러 각

Z->Y'->Z" KAWASAKI/QJAR ⚙️ ▼

Z 0.00° ▼

Y' 0.00° ▼

Z" 0.00° ▼

move until 선택하기



### 2.3.3 로봇

로봇 탭 페이지의 주요 내용은 그림 1 과 같습니다. 다음으로 각 옵션의 기능과 사용 방법을 설명하겠습니다.

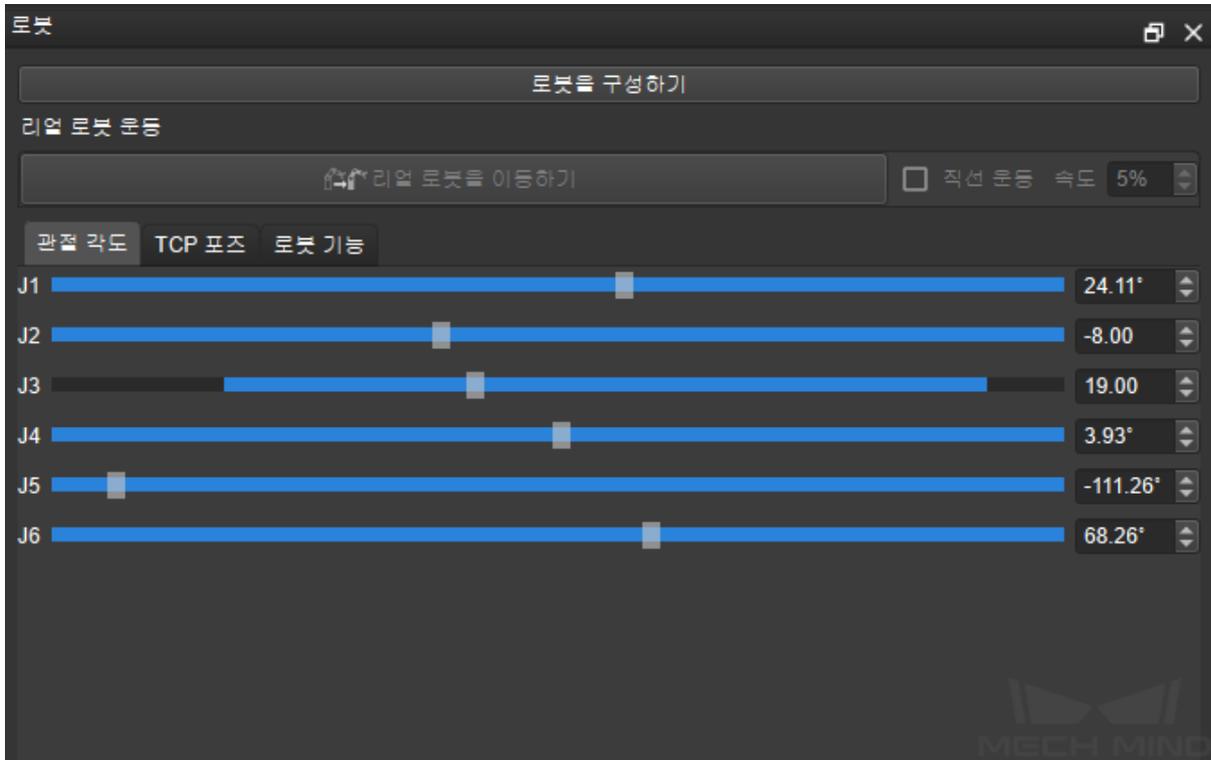


그림 1 로봇 탭 페이지

#### 로봇을 구성하기

로봇을 구성하기 를 클릭한 후 팝업 창에서 로봇의 브랜드 및 모델을 선택하여 로봇 모델을 도입할 수 있습니다. 그림 2 와 같습니다.

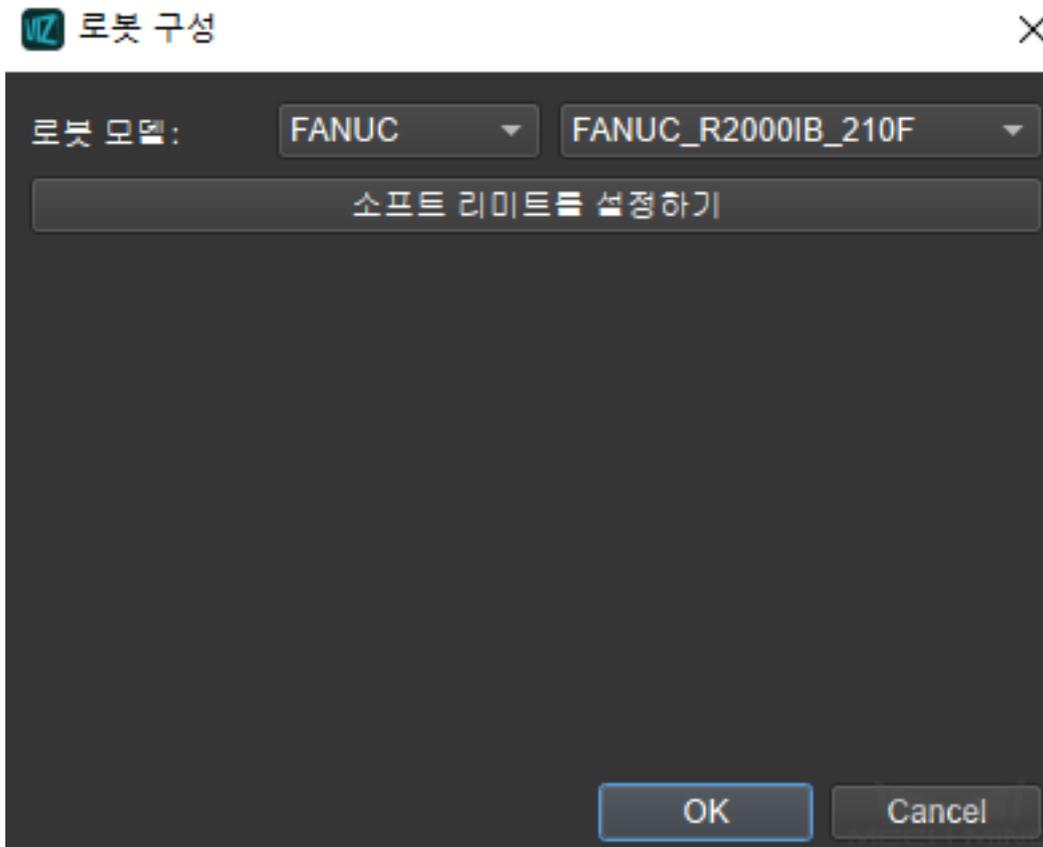


그림 2 로봇을 구성하기

소프트 리미트를 설정하기 를 클릭하여 로봇의 각 축에 소프트 리미트를 설정할 수 있습니다. 소프트 리미트의 수치 범위가 이미 존재한 관절 각도의 수치보다 작으면 영향을 받을 Skill 가 팝업 창에서 표시될 것입니다. 그림 3 과 같습니다.

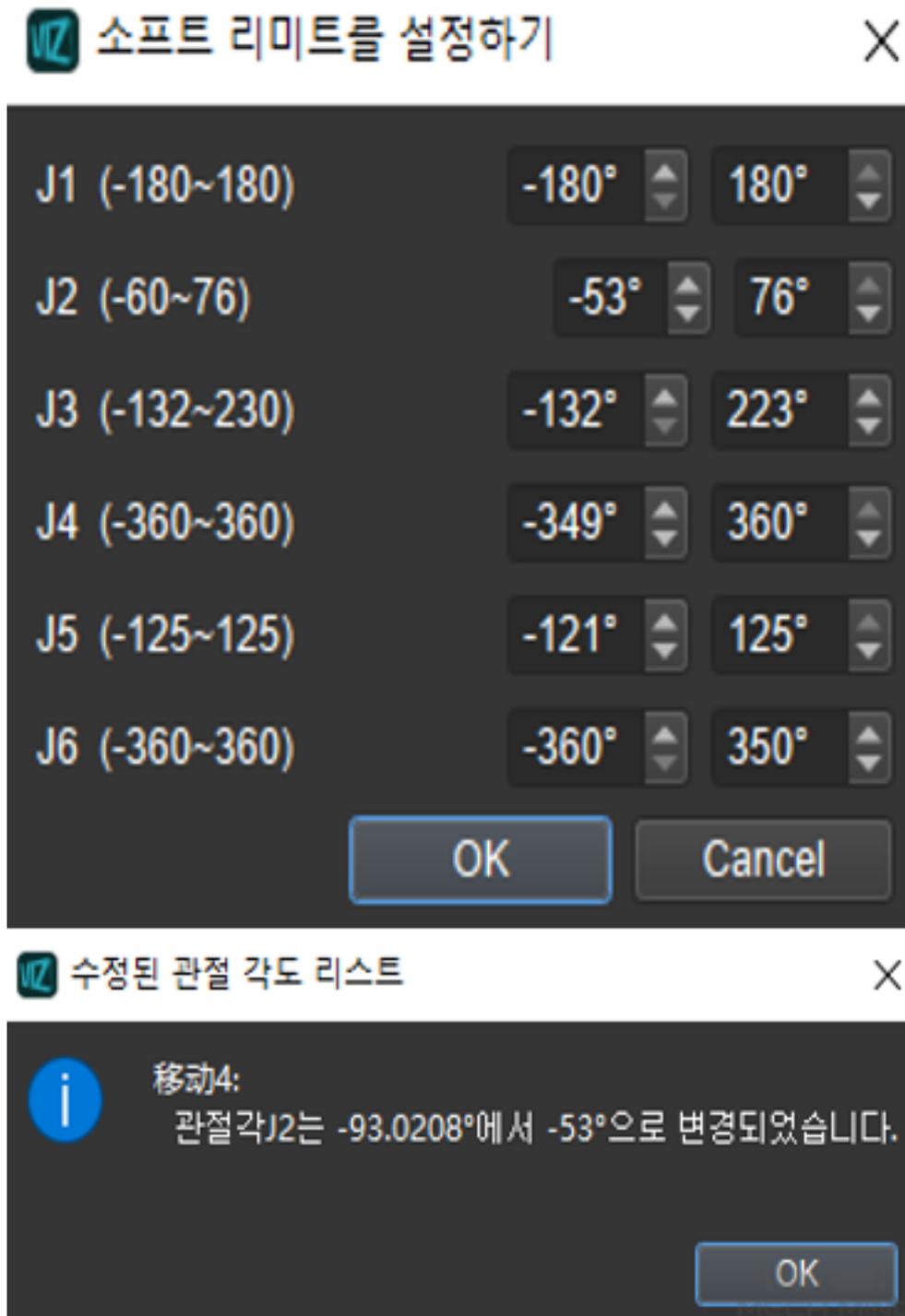


그림 3 소프트 리미트를 설정하기

## 리얼 로봇 운동

리얼 로봇을 이동하기를 클릭하면 로봇은 가상 로봇의 포즈로 이동할 것입니다. 리얼 로봇을 이동하기의 오른쪽에 직선 운동을 선택하여 이동 속도를 설정할 수 있습니다.

## 관절 각도

관절 각도 옵션 밑에 가상 로봇의 현재 관절 각도 데이터가 표시됩니다. 리얼 로봇과 동기화하려면 리얼 로봇이 연결된 상태에서 툴 바에 있는 **로봇을 동기화하기** 버튼을 누르세요. 가상 로봇의 포즈를 조정하려면 각 축의 슬라이드 블록을 조정하거나 오른쪽의 입력 창에서 수치를 직접 입력하면 됩니다. **관절 각도 조정** 버튼을 누르면 팝업 창에서 한번에 각축의 관절 각도 수치를 입력할 수 있습니다. 그림 4 와 같습니다.

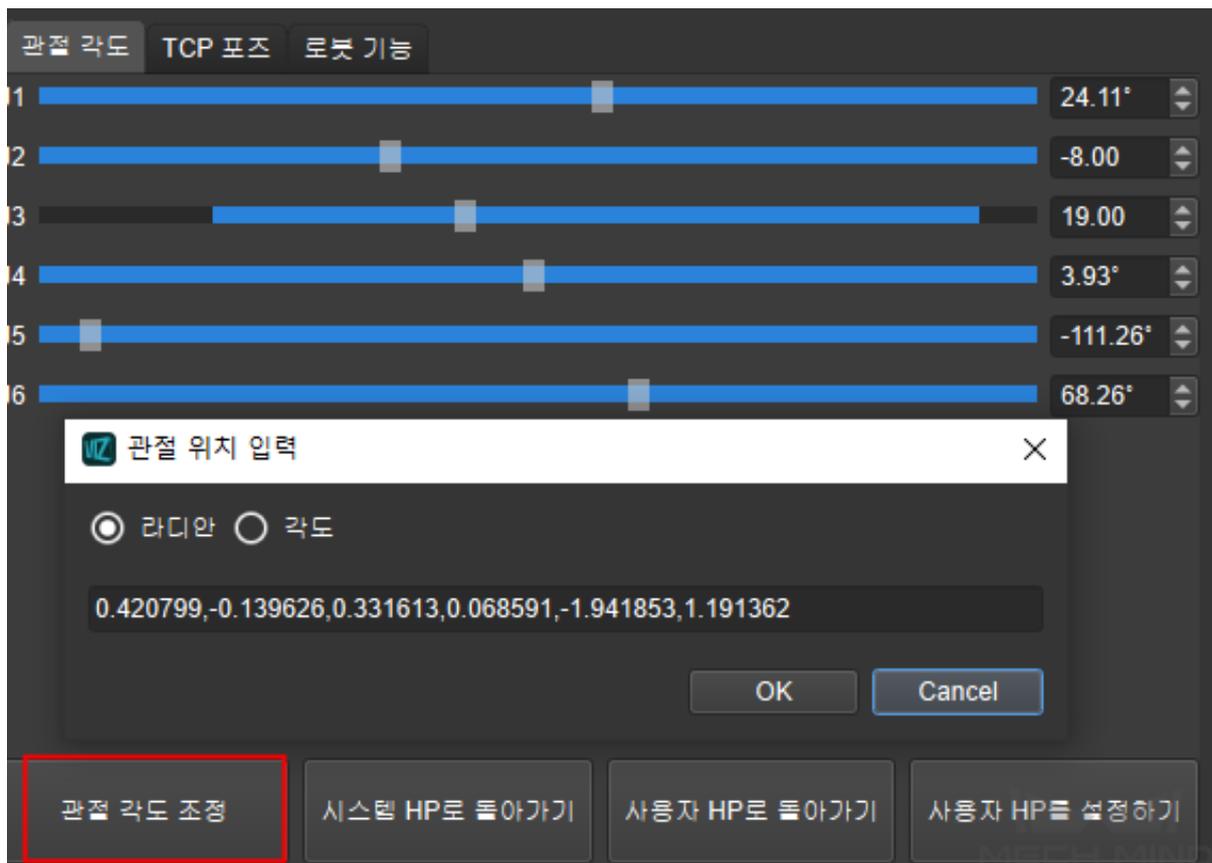


그림 4 관절 각도 조정

시스템 *HP* 로 돌아가기 버튼을 클릭하면 가상 로봇이 로봇 구성 파일에 설정된 초시 위치로 돌아갈 것입니다. 사용자 *HP* 로 돌아가기 버튼을 클릭하면 가상 로봇이 사용자가 설정한 초시 위치로 돌아갈 것입니다. 이 위치는 인터페이스 오른쪽의 사용자 *HP* 를 설정하기 를 통해 설정할 수 있습니다.

## TCP 포즈

TCP 포즈 옵션 밑에 가상 로봇의 현재 툴 포즈 데이터가 표시됩니다. **사원수** 및 **오일러 각** 버튼을 클릭하여 툴 포즈에 대해 조정할 수 있으며 **포즈 조정** 버튼을 클릭하여 툴 포즈를 직접 입력할 수 있습니다. 그림 5 와 같습니다.

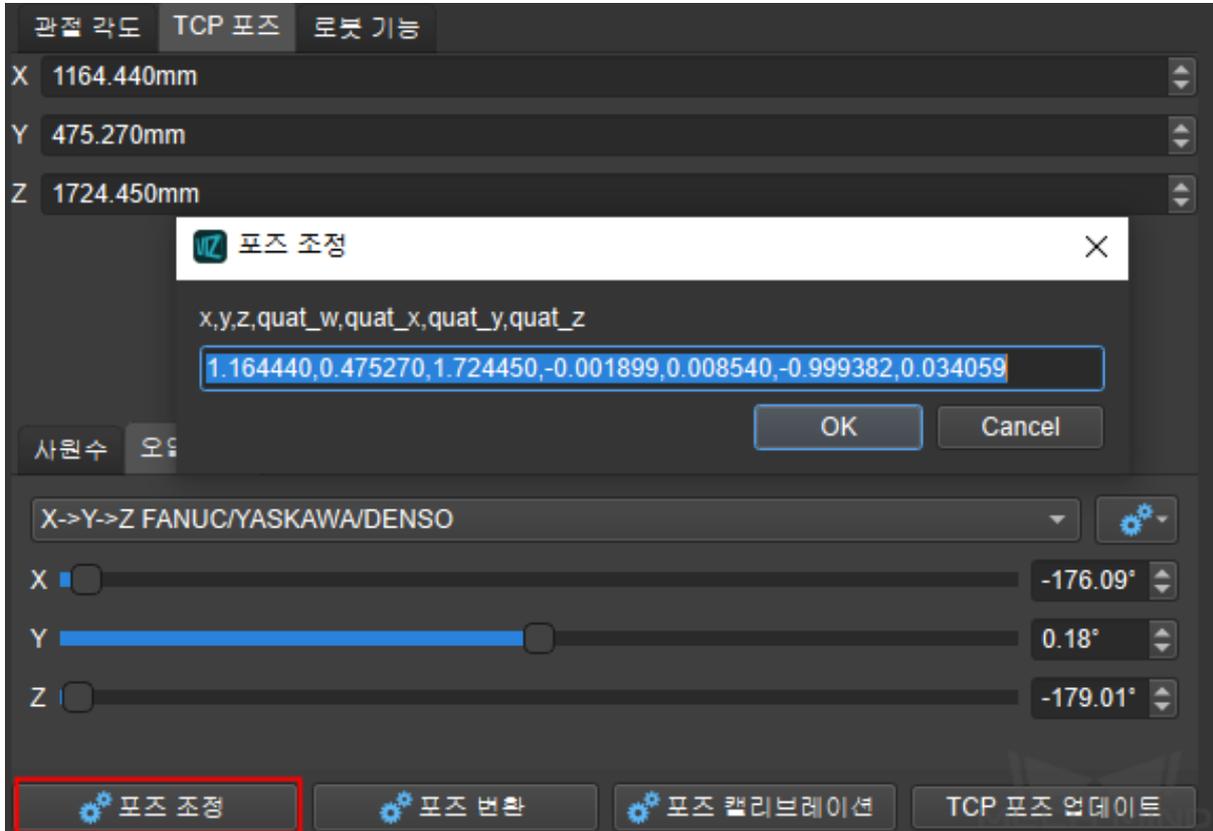


그림 5 포즈 조정

**포즈 변환** 버튼을 클릭하면 변환의 방식에 대해 맞춤형 설정을 하여 현재 포즈를 새로운 포즈로 변환시킬 수 있습니다. 그림 6 과 같이 먼저 참조 좌표계를 선택하여 XYZ 좌표, 사원수, 오일러 각을 설정하거나 **변환 편집** 버튼을 통해 직접 맞춤형 설정을 할 수 있습니다. 가상 로봇에 변환 결과를 실시간으로 표시할 것이고 이 결과를 저장하려면 *Apply* 를 클릭하며 결과를 버리려면 *Cancel* 를 클릭하여 가상 로봇은 원래의 위치로 돌아갈 것입니다.

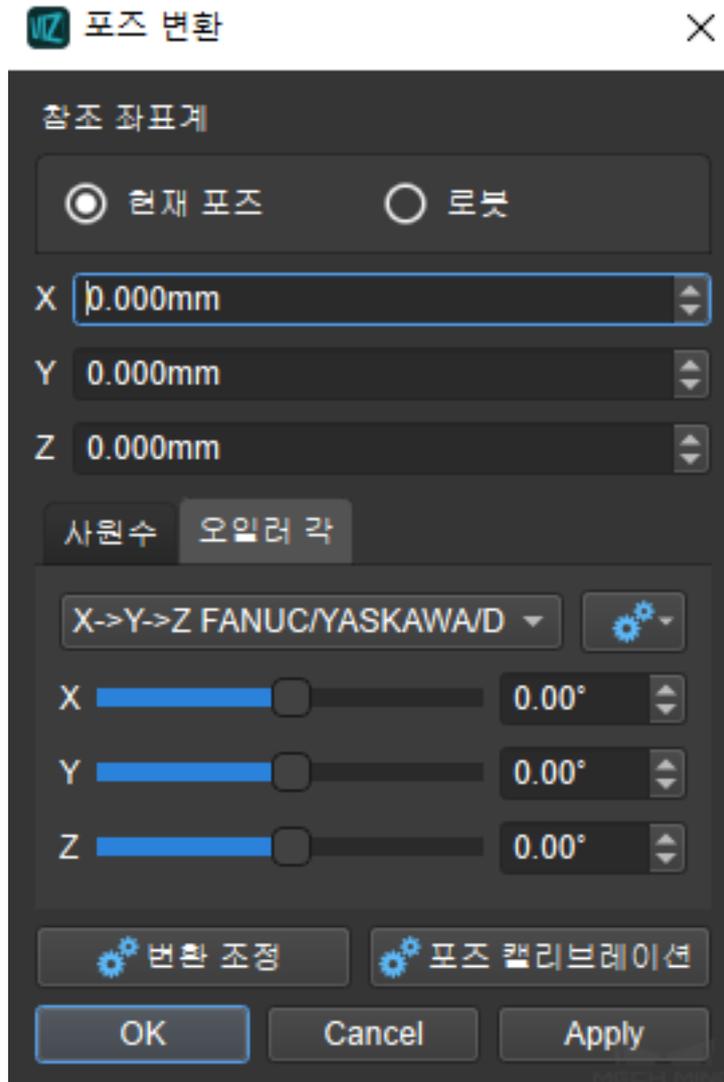


그림 6 포즈 변환

그림 7 과 같이 포즈 캘리브레이션 버튼을 눌러 팝업창 힌트 내용에 따라 TCP 포즈에 대해 캘리브레이션을 진행할 수 있습니다.

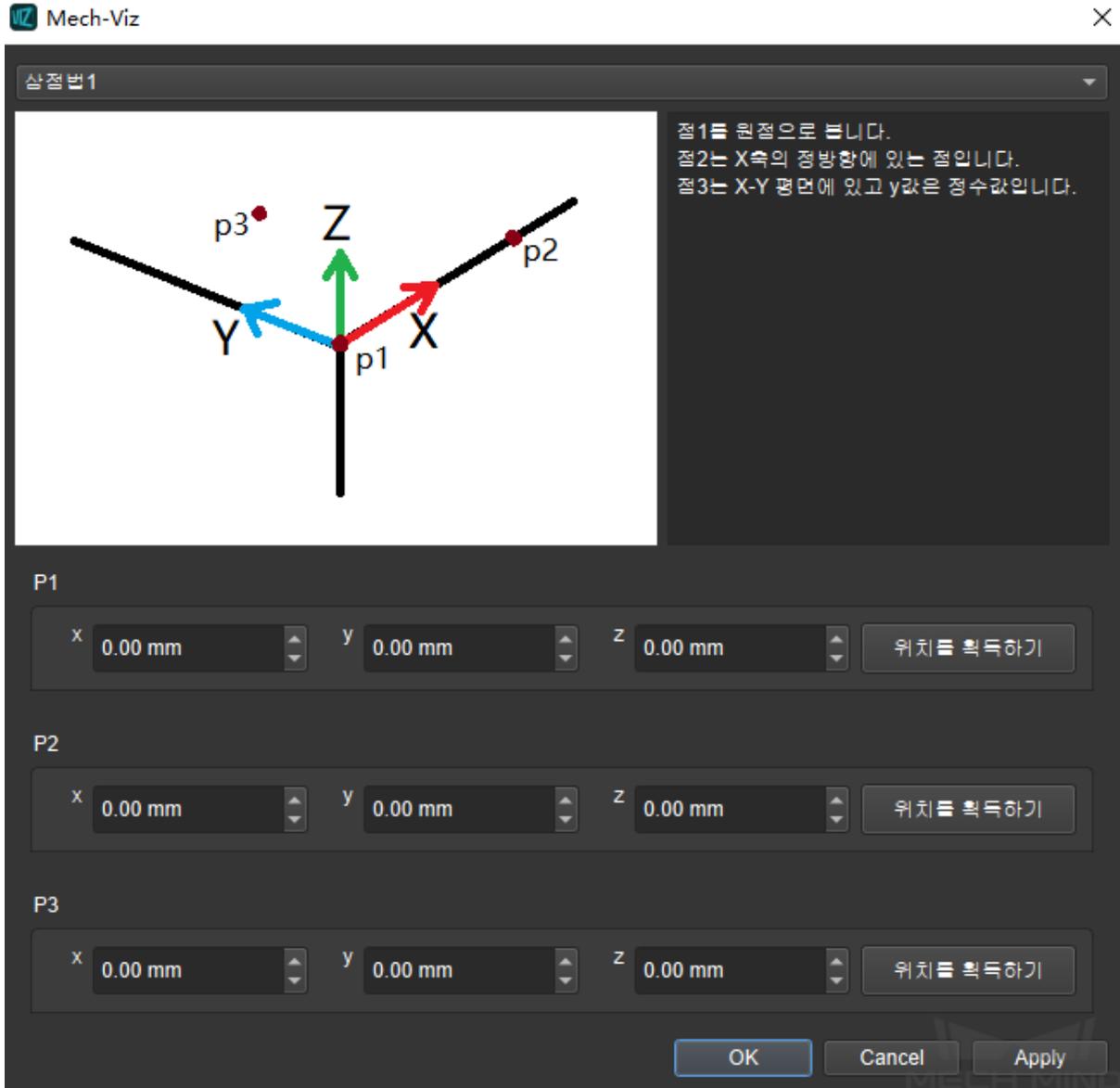
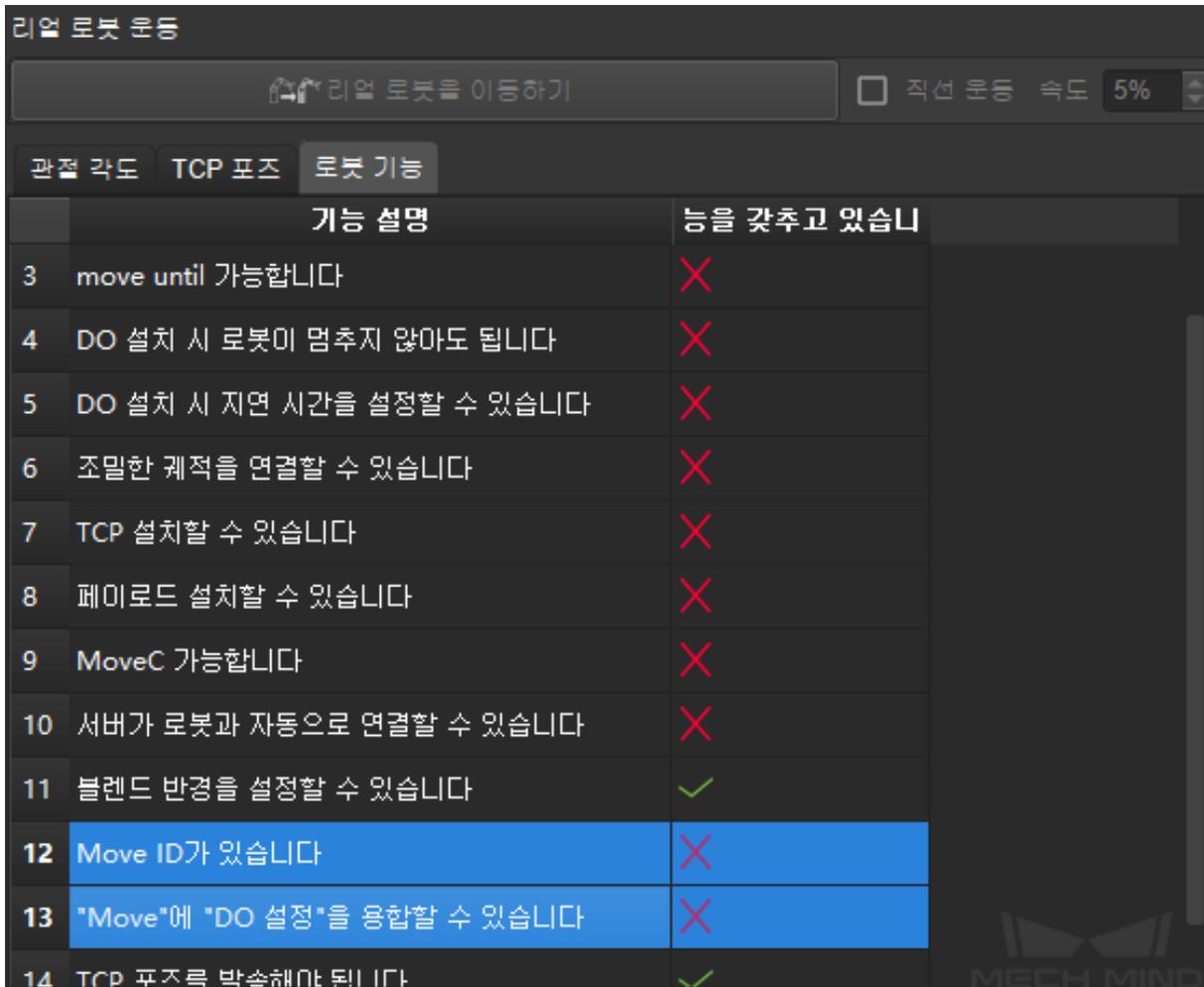


그림 7 포즈 캘리브레이션

로봇이 이미 연결된 상태에서 **툴 포즈 업데이트** 를 클릭하면 Mech-Viz 가 리얼 로봇에서부터 툴 포즈를 획득할 것입니다.

## 로봇 기능

로봇 기능 옵션 밑에 현재 사용 중인 로봇은 해당 기능을 갖추고 있는지가 표시됩니다. 그림 8 과 같습니다.



	기능 설명	능을 갖추고 있습니까
3	move until 가능합니다	✗
4	DO 설치 시 로봇이 멈추지 않아도 됩니다	✗
5	DO 설치 시 지연 시간을 설정할 수 있습니다	✗
6	조밀한 궤적을 연결할 수 있습니다	✗
7	TCP 설치할 수 있습니다	✗
8	페이로드 설치할 수 있습니다	✗
9	MoveC 가능합니다	✗
10	서버가 로봇과 자동으로 연결할 수 있습니다	✗
11	블렌드 반경을 설정할 수 있습니다	✓
12	Move ID가 있습니다	✗
13	"Move"에 "DO 설정"을 융합할 수 있습니다	✗
14	TCP 포즈를 발송해야 됩니다	✓

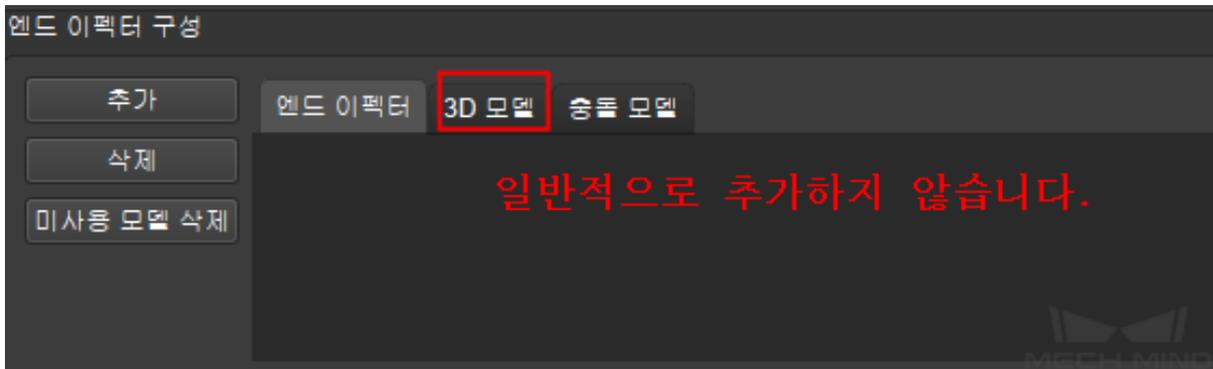
그림 8 로봇 기능

### TCP 드래거를 표시하기

이 옵션은 가상 로봇 엔드 이펙터의 드래거를 가리킵니다. 회전 및 병진을 선택하면 사용자들이 가상 로봇에서 엔드 이펙터를 직접 드래그함으로써 툴 포즈를 더 직관적으로 조정할 수 있습니다.

## 2.3.4 엔드 이펙터와 물체

### 클램프 모델 로딩



포인트 클라우드 충돌 감지를 진행해야 하는 프로젝트에는 반드시 클램프 모델을 로드해야 합니다. 보통 충돌 모델만 추가하고 3D 모델을 추가하지 않습니다.

이유:

1. 충돌 모델의 프로젝트 폴더가 더 작기 때문입니다. 3D 모델의 메모리 용량은 충돌 모델의 약 3 배입니다.
2. 편집하기가 더 쉽기 때문입니다. 클램프가 이동해야 할 때 충돌 모델만 추가하는 경우에 한번만 설정하면 되며 3D 모델과 충돌 모델을 모두 추가하는 경우에 각각 설정해야 합니다.
3. 3D 모델과 충돌 모델 간의 불일치 문제를 방지해서 충돌 감지의 결과가 예상에 부합하도록 할 수 있습니다.

또한 충돌 모델은 최대한 대략적이고 세부 사항은 삭제해야 하며 충돌 감지에 필요한 윤곽선만 사용하면 됩니다. 그렇지 않으면 소프트웨어의 계산 속도를 낮출 수도 있습니다.

### binvox 충돌 모델

포인트 클라우드 충돌 감지에 대해 고정밀도를 요구하는 시나리오에서 binvox 클램프 모델을 사용하는 것이 좋습니다. 가능하면 다른 시나리오에서도 binvox 모델을 사용하길 바랍니다.

SolidWorks 등 CAD 소프트웨어를 사용하여 모델을 STL, wrl 등 형식으로 저장할 수 있습니다.

STL 모델을 통해 전환된 binvox 모델은 윤곽선만 표시되는 충돌 모델입니다.

wrl 파일을 통해 전환된 binvox 모델은 완전한 충돌 모델입니다.

완전한 충돌 모델을 사용하면 충돌 정도 (마찰/충돌) 를 더 정확하게 판단할 수 있지만 계획 시간이 더 오래 걸립니다.

“ binvox.zip” 를 다운로드하고 압축을 푼 후 해당 폴더 아래에 notepad++ 를 통해 generate\_binvoxes.py 파일을 열어 아래의 그림과 같이 폴더 경로를 STL 모델이 저장된 폴더 경로로 수정해야 합니다. 주의해야 할 것은 “r” 를 삭제하면 안됩니다. generate\_binvoxes.py 를 저장한 다음에 두번 클릭하여 실행하면 됩니다.

```

import os
import sys

STL_FOLDER = r"C:\Users\mech-mind-016\Desktop\生成binvox\models"
gridSize = 256

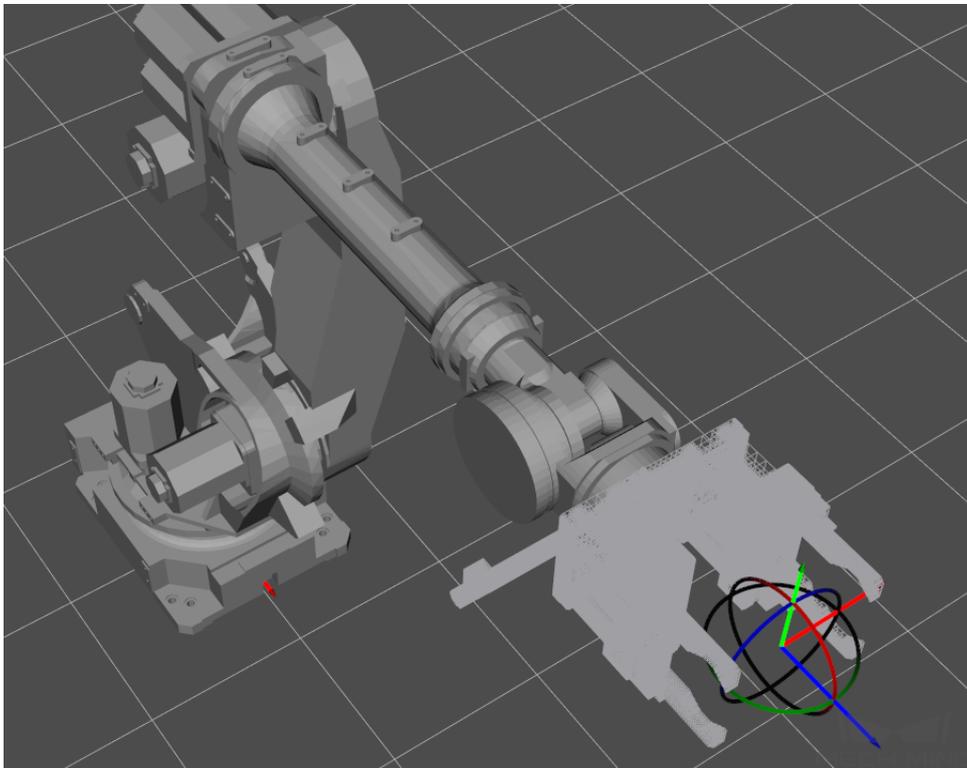
for dirpath, dirnames, files in os.walk(STL_FOLDER):
    for name in files:
        if name.lower().endswith(".stl") or name.lower().endswith(".wrl"):
            print(os.path.join(dirpath,name))
            os.system(r"binvox.exe -c -d:" + str(gridSize) + " " + os.path.join(dirpath,name))
    
```

실행 후 해당 경로 폴더 아래에 있는 모든 STL 와 wrl 모델이 binvox 모델로 전환될 것입니다.

**주의:** 전환될 STL 모델은 ASCII 형식만 지원합니다. 이진법 형식을 사용하면 안 됩니다. 원래의 STL 모델이 이진법 모델이라면 Mech-Viz 를 통해 3D 모델을 추가한 다음에 프로젝트를 저장하면 됩니다. 이렇게 하면 원래의 모델이 프로젝트 폴더에 있는 end\_effectors/3d\_models 폴더에서 ASCII 형식으로 저장될 것입니다. 아래의 그림과 같습니다:

viz工程(3) > end_effectors >	
名称	修改日期
3d_models	2019/11/28 1
collision_models	2019/11/28 1

로딩 후의 binvox 모델:

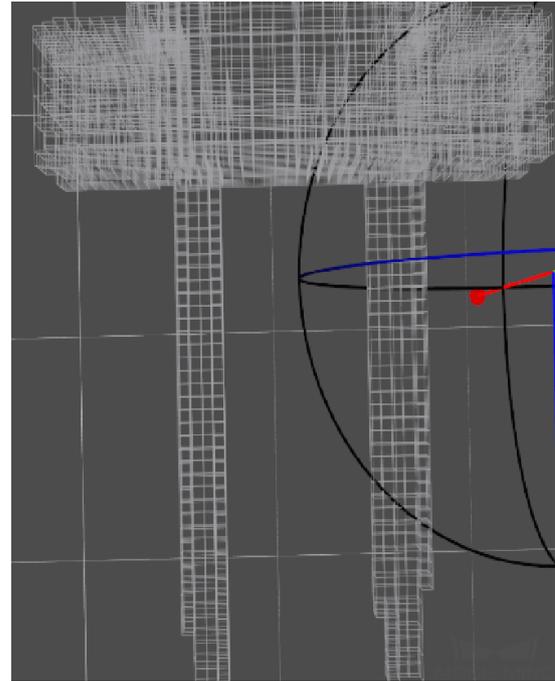
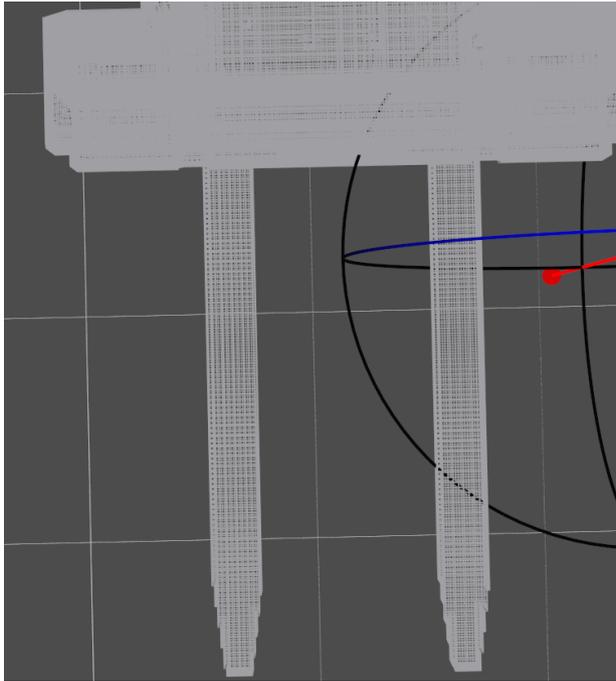


gridSize 를 수정함으로써 binvox 모델의 세밀도를 조정할 수 있습니다:

```
import os
import sys

STL_FOLDER = r"C:\Users\mech-mind-016\Desktop\生成binvox\models"
gridSize = 256

for dirpath, dirnames, files in os.walk(STL_FOLDER):
    for name in files:
        if name.lower().endswith(".stl") or name.lower().endswith(".wrl"):
            print(os.path.join(dirpath, name))
            os.system(r"binvox.exe -c -d " + str(gridSize) + " " + os.path.join(dirpath, name))
```



gridSize = 512 & gridSize = 100

미터와 밀리미터의 단위 전환 문제로 인해 로딩 후 시나리오에 표시된 binvox 모델이 매우 큰 경우도 있습니다. 이 때 notepad++ 를 통해 binvox 파일을 열어 *scale* 및 *translate* 를 1,000 배 축소하고 나서 모델을 다시 로드하면 됩니다.

```
#binvox 1
dim 256 256 256
translate -100 -335 -2.38071e-014
scale 445
data
```



```
#binvox 1
dim 256 256 256
translate -0.1 -0.335 -2.38071e-017
scale 0.445
data
```

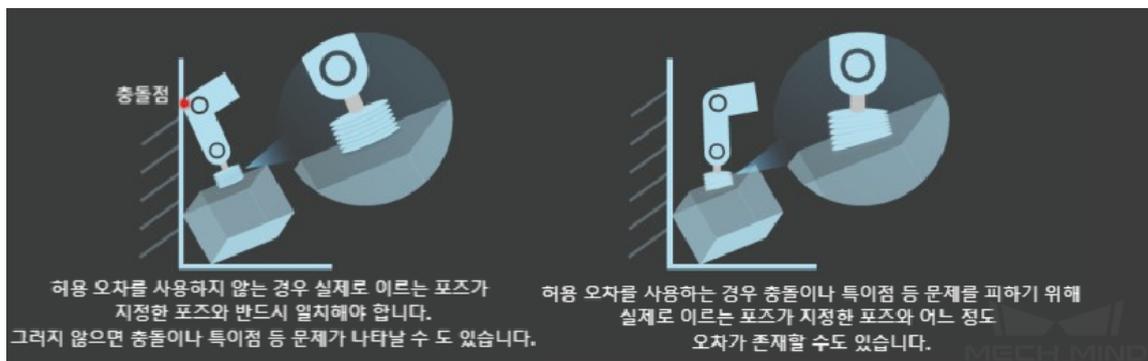
## 차겟 물체 구성 (궤적 계획에 사용됨)

모든 타겟 물체가 해당하는 물체 포즈를 갖습니다. 소프트웨어가 물체 포즈에 의해 툴 포즈를 계산하고 로봇이 피킹하도록 제어합니다.

## 대칭성 및 피킹 허용 오차

**대칭성:** 실제 작업 현장에 나타나는 타겟 물체와 피킹 포즈가 대칭성을 갖는 경우가 많습니다. 대칭성에 관한 자세한 사항은 대칭성을 참고하세요.

**피킹 허용 오차:** 클램프 (특히 발판) 와 물체 자체가 갖는 유연성으로 피킹 시 클램프와 물체 간에 일정한 오차가 존재해도 피킹할 수 있습니다. 피킹 허용 오차를 설정함으로써 로봇이 자발적으로 충돌 혹은 특이점 등 문제를 피할 수 있게 됩니다. 최적의 운동 궤적을 계획하기 위해 피킹 시 Mech-Viz 가 피킹 포즈의 대칭성, 피킹 허용 오차 및 물체 대칭성을 모두 고려하고 배치할 때 물체 대칭성에 의해 포즈를 선택할 것입니다.



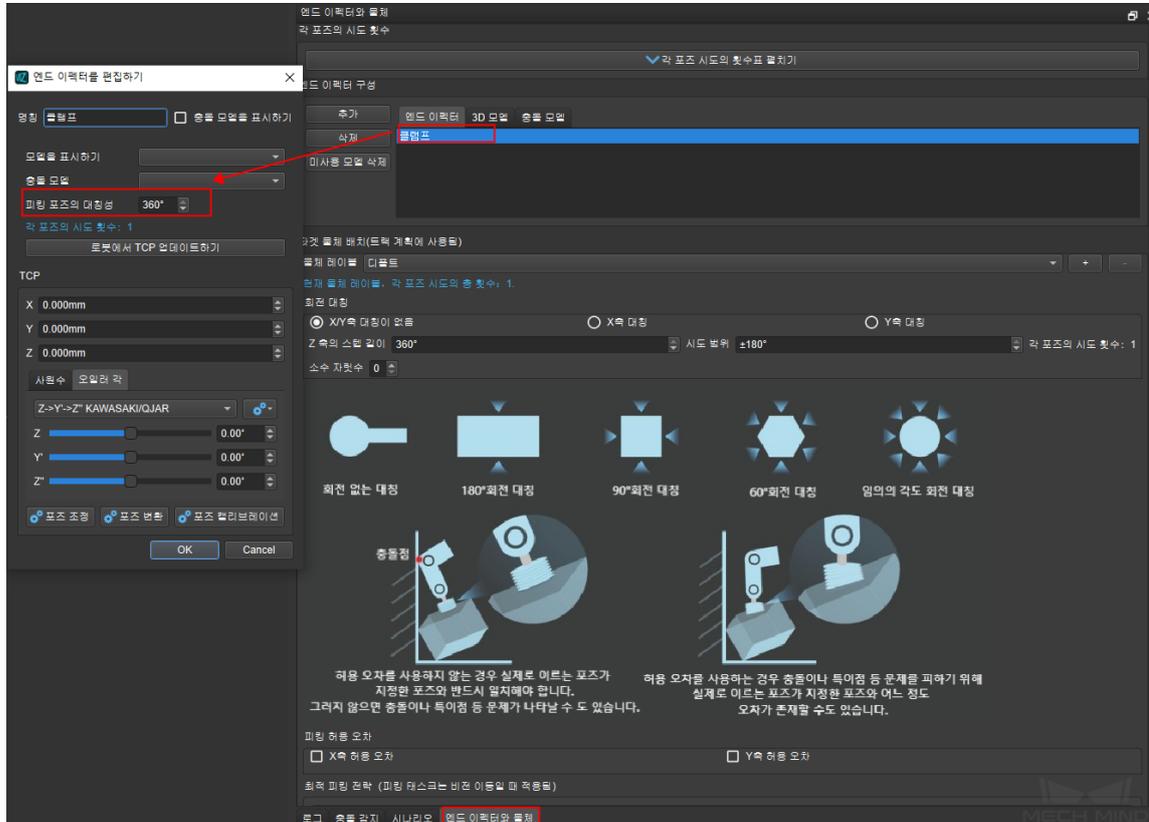
**주의:** 피킹 포즈의 대칭성과 피킹 허용 오차에 대해 구분할 필요가 있는지가 Mech-Vision 프로젝트의 출력 결과에 달려 있습니다. Mech-Vision 프로젝트에서 “많은 피킹 포즈에 매핑하기”란 스텝이 사용되는 경우에만 두 개념을 구분해야 하고 단순히 포즈만 출력하는 Mech-Vision 프로젝트에는 피킹 포즈의 대칭성과 피킹 허용 오차가 같은 효과를 갖습니다.



## 대칭성 및 피킹 허용 오차에 대한 설명

### 1. 피킹 포즈의 대칭성

피킹 포즈의 대칭성은 툴을 사용하여 물체를 피킹하는 방식과 관련이 있으며 최적의 피킹 포즈를 선택하는 데 사용됩니다. 아래 그림에 표시되는 바와 같이 설정하면 됩니다.



더 많은 정보는 **피킹 포즈 대칭성** 를 참고하세요.

### 2. 물체의 대칭성 및 피킹 허용 오차

**물체 레이블** visual\_move 스텝에서 출력된 포즈가 레이블을 갖는 경우 여기서 대응하는 레이블을 검색하고 설정된 해당 유형 물체의 대칭성을 사용합니다. 대응하는 레이블을 못 찾는 경우 “디폴트” 레이블과 대응하는 대칭성을 사용할 것입니다. “디폴트” 레이블을 삭제하면 안 되고 다른 레이블은 사용자가 자체 추가하거나 삭제할 수 있습니다.

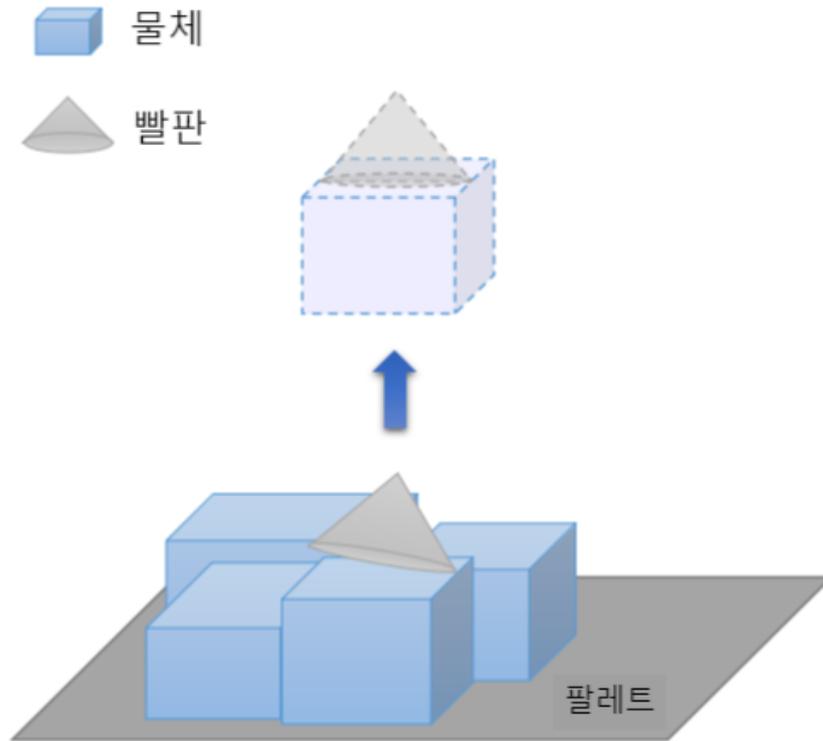
**회전 대칭** X 축과 Y 축 대칭 둘 중 하나만 설정할 수 있습니다 (Z 축은 이 설정의 영향을 받지 않음). 일반적으로 X 축과 Y 축 대칭을 모두 가지고 있는 물체가 구이며 실제로 이러한 경우가 거의 없기 때문입니다. 구에 대해 대칭성을 설정하려면 대칭성이 아닌 피킹 허용 오차를 통해 설정해야 합니다.

**피킹 허용 오차** 피킹 허용 오차는 X 축과 Y 축의 허용 오차를 동시에 설정할 수 있으며 실제 응용 시 따로 적용됩니다. 즉 기준 포즈가 X 축 혹은 Y 축을 중심으로 회전할 수 있지만 동시에 두개의 축을 중심으로 회전하지 않을 것입니다. 또한 피킹 허용 오차와 물체 대칭성의 대칭축을 선택할 때 같은 축을 선택하면 안됩니다. 예를 들어 아래의 그림에 표기되는 바와 같이 X 축 대칭성을 선택하면 X 축 허용 오차를 선택하지 못합니다.



### 3. 물체 X/Y 축 대칭성 및 피킹 허용 오차의 차이

- 물체 X/Y 축 대칭성은 엔드 이펙터와 물체가 X/Y 축에 있는 상대 위치가 변하면 안되고 물체를 배치할 때 물체 포즈에 대해 명확한 요구가 있는 시나리오에 사용됩니다. 예를 들어 크랭크 축 피킹 시나리오에 크랭크 축을 배치할 때 X 축의 회전 각도에 대해 특정한 요구가 있는 경우가 많습니다. 이때 크랭크 축이 배치 시 요구되는 각도에 의해 정확하게 배치될 수 있도록 X 축 대칭성을 설정해야 합니다.
- 피킹 허용 오차는 엔드 이펙터와 물체가 X/Y 축에 있는 상대 위치가 변해도 되는 시나리오에 사용됩니다. 예를 들어 팔판으로 물체를 피킹하는 시나리오에서 팔판과 물체 간에 사소한 각도 차이가 있어도 팔판의 유연성으로 여전히 물체를 성공적으로 피킹할 수 있습니다. 아래의 그림과 같습니다:

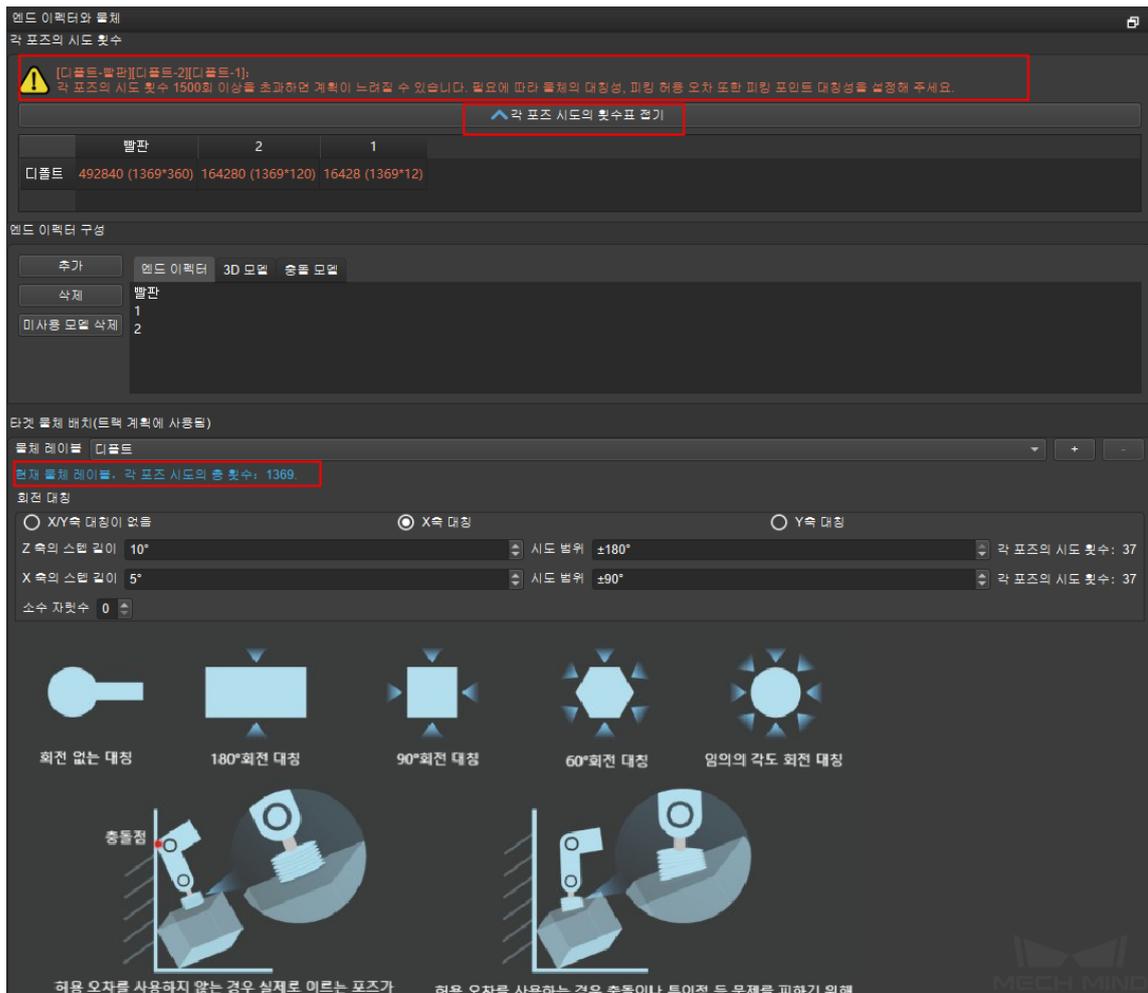


빨판으로 작은 각도 음셋 피킹

클램프로 부품을 피킹하는 시나리오도 마찬가지입니다. 예를 들어 크랭크 축 혹은 체인 링크 피킹 시 X/Y 축에서 작은 각도 편차가 허용되면 피킹 시 타겟 물체가 클램프와 딱 맞춰 물체를 피킹할 수 있습니다.

4. 설정이 정당하지 않으면 계획 시간이 오래 걸릴 수도 있습니다.

피킹 포즈 대칭성, 물체 대칭성 및 피킹 허용 오차를 동시에 사용하면 최종의 시도 횟수는 이 세 가지 파라미터의 설정된 수치의 곱이기 때문에 잘못된 설정으로 인해 시도 횟수가 많아지고 계획 시간이 너무 길어져서 Mech-Viz 가 응답하지 않을 수 있습니다. 시도 횟수가 1500 회를 초과해서 Mech-Viz 에서 경고 알림이 나타나면 아래 그림과 같이 각 포즈의 시도 횟수표를 펼치기를 클릭하여 자세한 정보를 확인할 수 있습니다.



### 최적 피킹 전략 (피킹 태스크가 비전 이동일 때 적용됨)

현재 디폴트, 최소 전체 회전 그리고 최소 비전 포즈와의 회전 차액 등 세 가지의 모드를 선택할 수 있습니다.

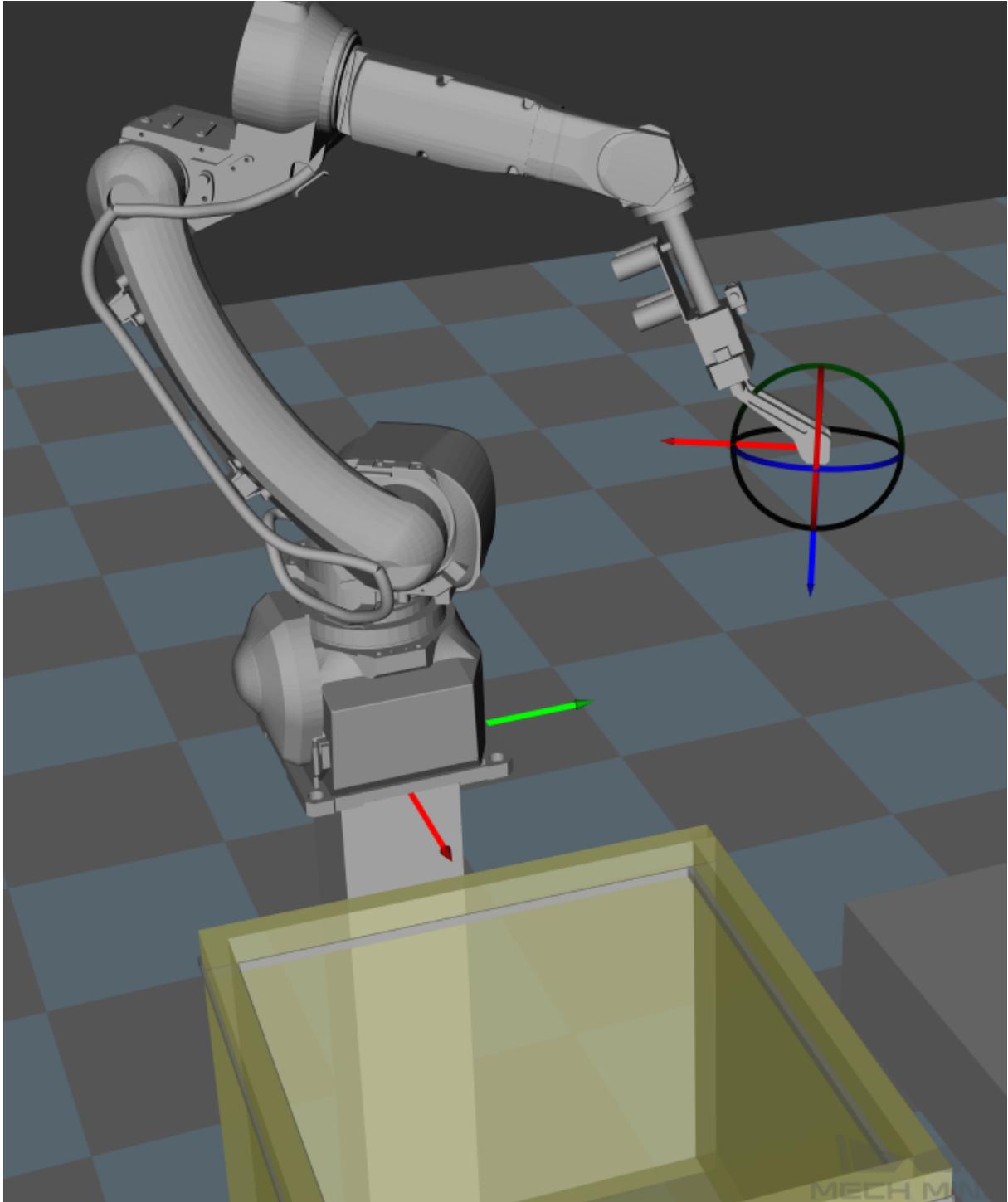
**주의:** 1.4.0 버전에서《디폴트》와《최소 전체 회전》의 효과가 같습니다.

**최소 전체 회전** 이 모드를 선택하면《피킹-배치》전 과정에서 피킹 포즈의 최소 Z 축 회전 원칙에 의해 대칭성을 설정할 것입니다. 이 모드의 장점은 물체를 피킹한 후 로봇의 불필요한 회전 혹은 타겟 물체가 떨어질 위험을 피할 수 있다는 것입니다.

**최소 비전 포즈와의 회전 차액** 어떤 특별한 시나리오에서 더 나은 피킹 효과를 얻기 위해 Mech-Vision가 엔드 이펙터의 회전을 유도해야 합니다.

로봇의 클램프가 아래의 그림과 같습니다. 타겟 물체는 임의의 각도 대칭을 갖는 원기둥체입니다. 간섭이 없는 경우에 클램프가 임의의 Z 축 각도로 물체를 피킹할 수 있습니다. 하지만 깊은 피킹 용기의 테두리 부분에 위치하는 물체를 피킹할 때“최소 전체 회전”모드를 선택하면 실제 피킹 시 클램프가 용기 벽에 붙어서 피킹하여 충돌 가능성이 매우 큽니다. 이때 Mech-Vision 을 통해 실제 피킹 시의 Z 축 각도를 계산할 수 있습니다. 예를 들어 Mech-Vision 프로젝트에서 피킹 용기의 테두리 부분에 위치하는 물체의 X 축은 모두 용기 중심점으로 가리키도록 설정함으로써 최소 비전 포즈와의 회전 차액을 유지하여 클램프가 물체를 피킹할 때 용기에서 최대한 정도 멀리 떨어져 있어 안전성을 크게 향상시킬 수 있습니다.

위에서 언급한 시나리오에 대해 피킹 결과를 최적화하려면 Mech-Vision 프로젝트에서 용기 중심과 내벽에 대해 서로 다른 레이블을 지정하여 Mech-Viz 에서 이 두 가지의 레이블에 대해 따로 편집합니다. 용기 중심에 위치하는 물체는 **최소 전체 회전** 원칙에 의해 불필요한 회전을 줄이며 내벽에 위치하는 물체는 **최소 비전 포즈와의 회전 차액** 원칙에 의해 Mech-Vision 유도에 따라 보다 안전한 피킹 포즈를 선택합니다.



## 타겟 물체의 사이즈

물체의 레이블과 상관없이 전반에서 이 파라미터는 하나만 있습니다. 피킹 태스크가 비전 이동이 아닌 경우에 적용되고 피킹 물체의 사이즈를 결정하는 것으로 개발자 테스트에 많이 사용됩니다.

### 2.3.5 충돌 감지

이 부분에는 주로 충돌 감지와 관련된 설정 및 주의 사항을 소개하겠습니다. Mech-Viz 에서 충돌 감지에 대한 설정은 주로 충돌 모델 구성 및 충돌 감지 파라미터 설정 두 개의 스텝이 포함됩니다.

#### 충돌 모델 구성

Mech-Viz 에서 충돌 감지를 받는 모델 유형은 로봇 관절, 시나리오 물체, 엔드 이펙터 (표면), 포인트 클라우드, 감지된 비 직육면체, 피킹된 비 직육면체 그리고 배치된 비 직육면체 등 있습니다. 유형이 다른 충돌 모델에 대한 감지의 계산 과정과 방식도 다릅니다. 충돌 모델 유형에 관한 자세한 정보는 [충돌 모델 유형](#) 를 참고하세요.

각종의 충돌 모델의 구성 방식이 서로 다릅니다. 구체적인 구성 방식은 다음과 같습니다:

**로봇 관절** 로봇 에서 대응하는 로봇 브랜드 및 모델 번호를 선택하여 로봇 모델을 추가할 수 있습니다. [그림 1](#) 과 같습니다.



그림 1 로봇 관절 모델 구성

시나리오 물체 시나리오에서 새로운 모델을 클릭하여 물체 유형에 따라 직육면체 혹은 정사각형 바구니 등 모델을 새로 만들 수 있으며 모델을 로드하기를 클릭하여 기존의 시나리오 모델 (STL OBJ DAE 등 형식 모두 가능함) 을 로드할 수도 있습니다. 그림 2 와 같습니다.

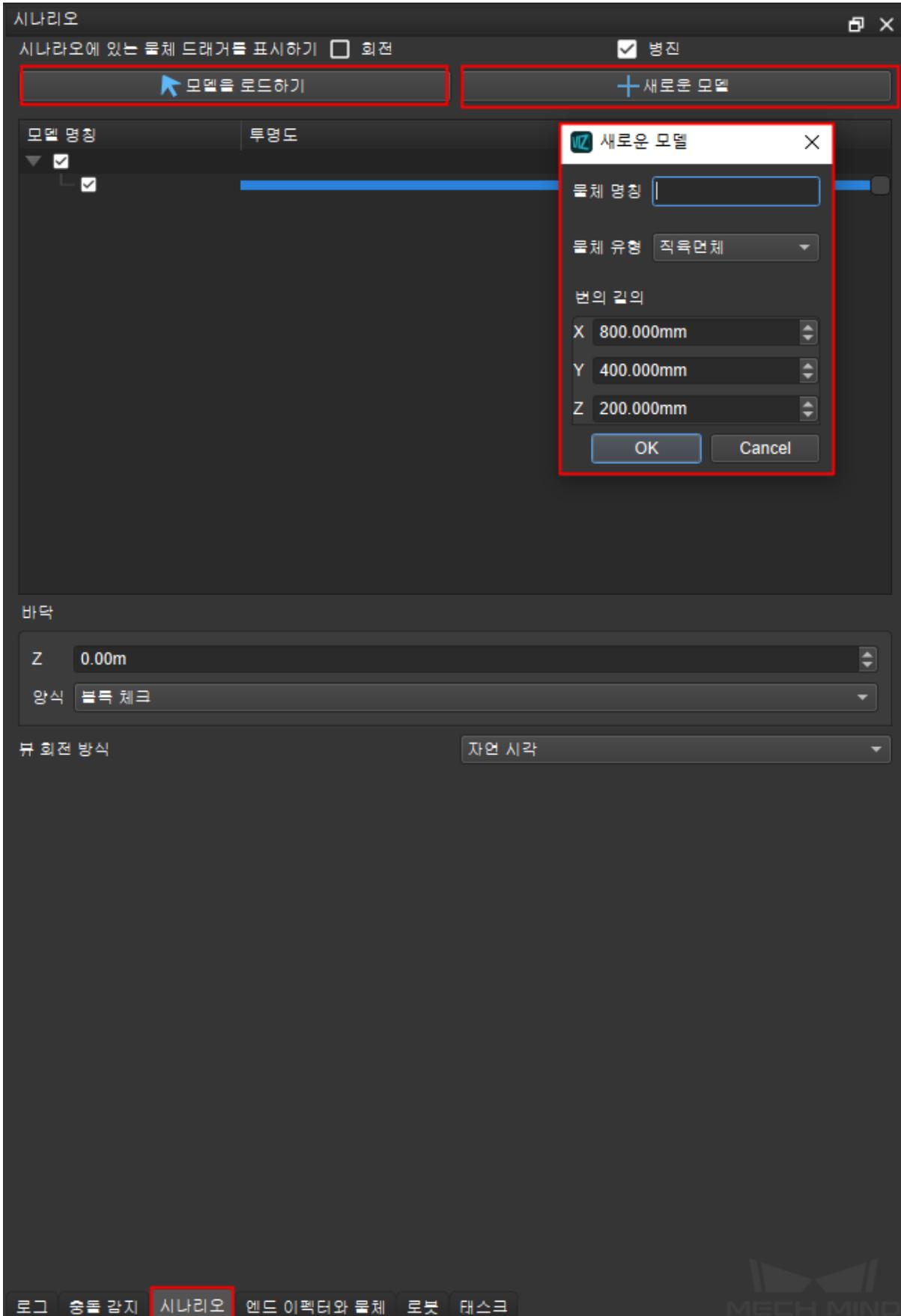
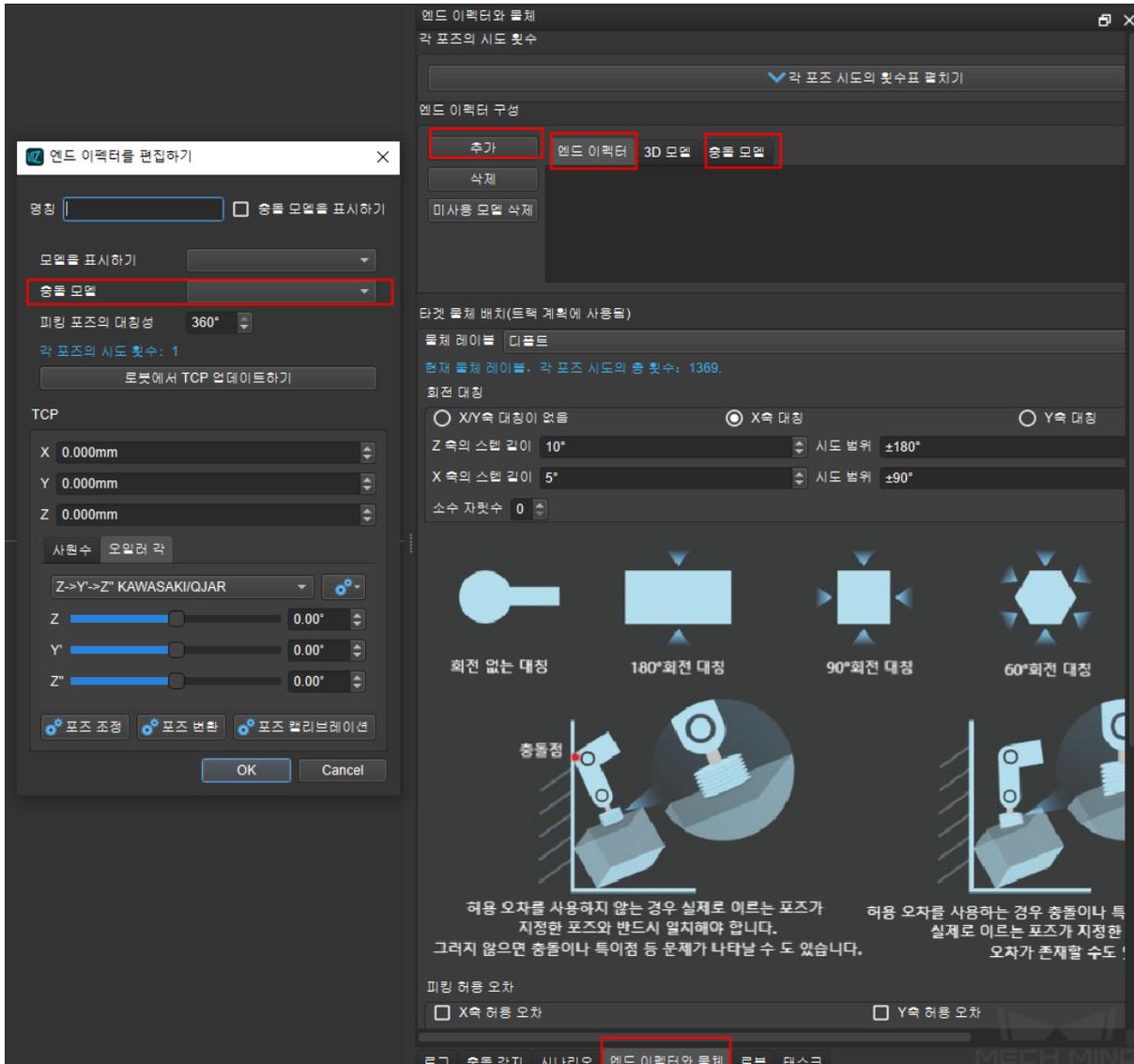


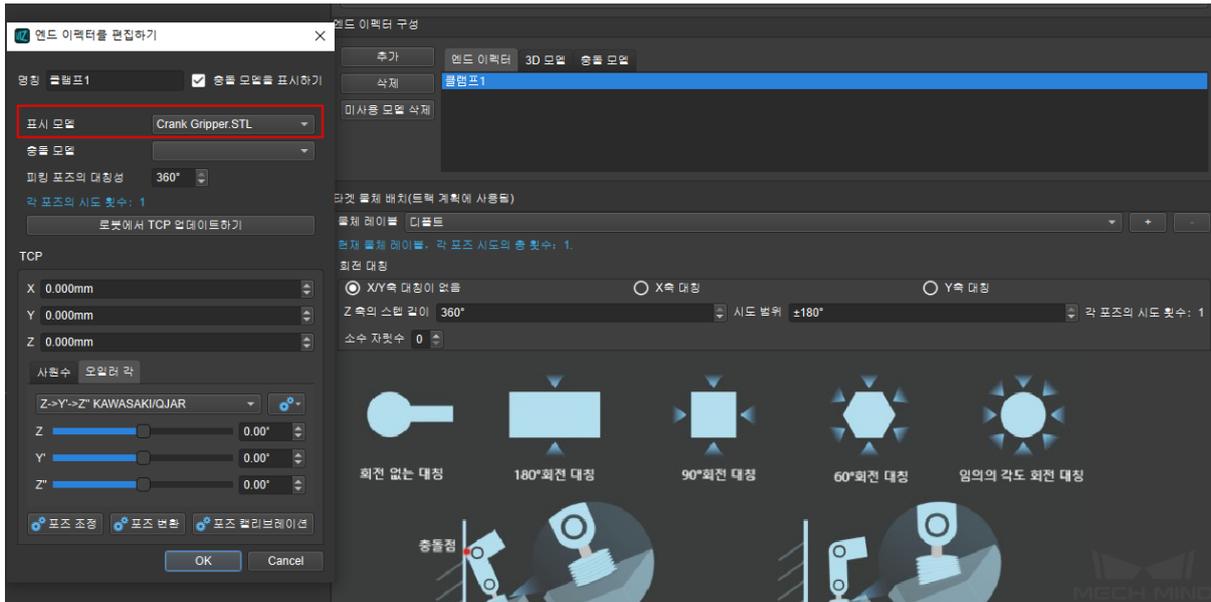
그림 2 시나리오 물체 모델 구성

**주의:** 로드된 모델은 부가 파라미터의 **충돌을 감지할 필요가 있습니다** 를 선택해야 실제로 충돌 감지 태스크로 추가될 수 있습니다. 그렇지 않으면 표시 모델만으로 나타날 것입니다.

**엔드 이펙터** **엔드 이펙터와 물체** 에서 충돌 모델을 선택하고 왼쪽에 있는 **추가** 를 클릭하여 원하는 모델을 추가하세요. 하지만 이때 충돌 모델이 실제 효과를 발휘하지 않았습니다. 아래의 그림과 같이 엔드 이펙터에서 해당 충돌 모델을 선택해야 합니다.

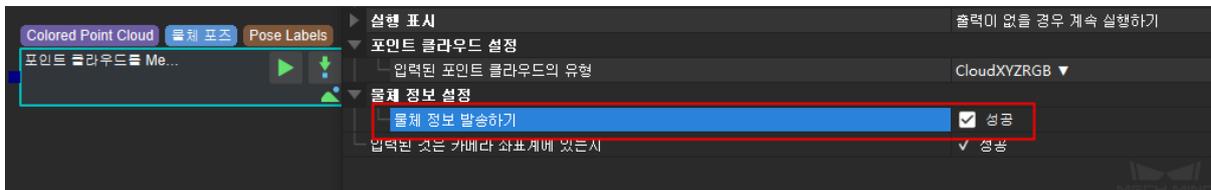


**팁:** 표시 모델만 있는 경우에 기본적으로 경계 박스 (기본 기학체 유형) 를 충돌 모델로 사용하여 충돌 감지를 실행합니다. 경계 박스 충돌 모델을 관찰하려면 엔드 이펙터에서 표시 모델을 도입한 뒤 **충돌 모델을 표기하기** 를 선택해야 합니다. 아래의 그림과 같습니다.

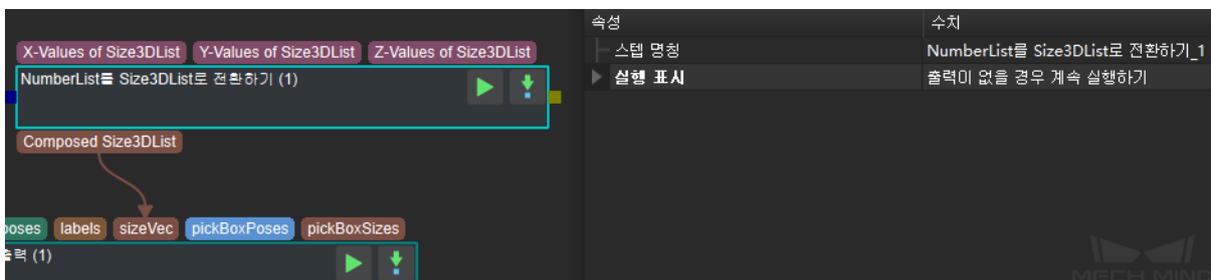


**포인트 클라우드** 대부분 포인트 클라우드 모델은 프로젝트에서 처리해야 할 물체인데 Mech-Vision 에서 send\_point\_cloud\_to\_external\_service 스텝을 통해 도입해야 합니다.

**감지된 물체** 충돌을 감지할 필요가 있는 물체의 STL 모델과 binvox 모델을 프로젝트의 collision\_models 폴더에 추가해야 합니다. 다음으로 아래의 그림과 같이 Mech-Vision 에서 send\_point\_cloud\_to\_external\_service 스텝을 사용하고 파라미터 설정에 있는 **물체 정보를 보내기** 물체 포즈 레이블과 포인트 클라우드 등 를 True 로 설정하여 도입해야 합니다. binvox 모델의 전환 방법은 **binvox 충돌 모델** 를 참고하세요.



**피킹된 물체** 주로 직육면체와 비직육면체로 나뉘집니다. 비직육면체 모델의 구성 방식은 감지된 물체의 모델과 같고 직육면체 모델을 구성할 때 Mech-Vision 에서 “NumberList 를 Size3DList 로 전환하기” 스텝을 통해 물체의 사이즈를 계산하여 “출력” 스텝으로 입력해야 합니다. 그 다음에 “출력” 스텝에서 Size3DList(물체의 사이즈) 를 Mech-Viz 로 보내고 Mech-Viz 가 자동으로 해당하는 충돌 모델을 생성할 것입니다. 아래의 그림과 같습니다.



**배치된 물체** 대부분 경우에는 구성할 필요가 없으며 피킹된 물체 모델 혹은 감지된 물체 모델에 따라 구성됩니다. 이러한 모델의 충돌 감지는 시나리오 충돌 감지입니다.

## 충돌 감지 파라미터 설정

**주의:** 충돌 감지 파라미터를 설정하기 전에 대응하는 충돌 모델을 구성해야 합니다. 그렇지 않으면 충돌 감지를 진행하지 못하거나 오류가 발생할 것입니다.

Mech-Viz 충돌 감지는 각종의 충돌 모델을 둘씩 결합하여 진행되는 것이므로 포인트 클라우드와 다른 물체의 충돌, 로봇과 다른 물체의 충돌 등 다양한 유형의 충돌 감지 조합이 있습니다. Mech-Viz의 충돌 리스트를 통해 현재 실행 중인 프로젝트에서 이미 감지된 충돌 모델 조합 유형을 확인할 수 있습니다. 아래의 그림과 같이 충돌 리스트에 있는 색깔이 다른 블록은 충돌 감지의 다른 판단 조건을 뜻합니다.

	로봇 관절	시나리오 물체	엔드 이펙터 (표면)	포인트 클라우드	감지된 비 직육면체	피킹된 비 직육면체	배치된 비 직육면체
로봇 관절							
시나리오 물체							
엔드 이펙터 (표면)							
포인트 클라우드	250 mm <sup>3</sup>		1000, 8 cm <sup>2</sup>				
감지된 비 직육면체			27000 mm <sup>3</sup>				
피킹된 비 직육면체		0		2000 mm <sup>3</sup>	2000 mm <sup>3</sup>		
배치된 비 직육면체							

**충돌 리스트**

- 충돌을 감지하지 않거나 충돌이 존재하지 않습니다.
- 접촉만 있으면 모두 충돌로 판정됩니다.
- 설정된 역치보다 더 작은 충돌만 허용합니다.
- 충돌을 감지할 때 설정된 직육면체의 하부 안전 거리를 사용합니다.
- 충돌 혹은 접촉은 계획에서 감스 추세를 보이면서 설정된 역치보다 큰 경우도 허용되며 역치보다 작은 경우도 허용합니다.

아래의 그림과 같이 충돌 감지 구성 (전반) 에 포인트 클라우드 구성, 엔드 이펙터 구성, 로봇 관절 구성 및 피킹된 물체 구성 등 네 가지의 구성이 포함됩니다. 충돌 감지에 대해 올바른 설정을 하면 불필요한 충돌 감지를 피하고 프로젝트의 실행 효율을 크게 향상시킬 수 있습니다.

	로봇 관절	시나리오 물체	엔드 이펙터 (표면)	포인트 클라우드	감지된 비 직육면체	피킹된 비 직육면체	배치된 비 직육면체
로봇 관절							
시나리오 물체							
엔드 이펙터 (표면)							
포인트 클라우드	250 mm <sup>3</sup>		1000, 8 cm <sup>2</sup>				
감지된 비 직육면체			27000 mm <sup>3</sup>				
피킹된 비 직육면체		0		2000 mm <sup>3</sup>	2000 mm <sup>3</sup>		
배치된 비 직육면체							

**포인트 클라우드 구성**

포인트 클라우드의 다른 물체의 충돌을 감지하기

포인트 클라우드를 알려진 충돌 모델로 판정하는 역치: 3.0 mm

8mm      2mm

**엔드 이펙터 구성**

STL 표면      OBJ 표면

포인트 클라우드와의 충돌을 감지하기

포인트 클라우드와의 충돌을 감지하기

점차 선택할 때의 최적 모드: XYZ축 최단 범위나 충돌이 가장 작은 모드

충돌 표면 역치: 1000 mm

충돌 면적 역치: 8 cm<sup>2</sup>

**로봇 관절 구성**

포인트 클라우드와의 충돌을 감지하기

포인트 클라우드와의 충돌을 감지하기

충돌 부피 역치: 250 mm<sup>3</sup>

**피킹된 물체 구성**

피킹된 물체의 다른 물체의 충돌을 감지하기

피킹된 물체 유형

직육면체      비 직육면체

포인트 클라우드와의 충돌을 감지하기

포인트 클라우드와의 충돌을 감지하기

충돌 부피 역치: 2000 mm<sup>3</sup>

위에서 언급된 네 가지의 구성에 대해 아무 설정도 하지 않으면 Mech-Viz 가 로봇 관절, 시나리오 물체 및 엔드 이펙터에 대해만 충돌 감지를 진행하기 때문에 이때의 시나리오는 기본적인 충돌 감지 시나리오이고 이때 진행되는 충돌 감지는 시나리오 충돌과 자체 충돌에 대한 감지입니다. 아래의 그림과 같습니다.

	로봇 관절	시나리오 물체	엔드 이펙터 (표면)	포인트 클라우드	감지된 비 직육면체	피킹된 비 직육면체	배치된 비 직육면체
로봇 관절	충돌 감지	충돌 감지	충돌 감지	충돌 감지	충돌 감지	충돌 감지	충돌 감지
시나리오 물체	충돌 감지	충돌 감지	충돌 감지	충돌 감지	충돌 감지	충돌 감지	충돌 감지
엔드 이펙터 (표면)	충돌 감지	충돌 감지	충돌 감지	충돌 감지	충돌 감지	충돌 감지	충돌 감지
포인트 클라우드	충돌 감지	충돌 감지	충돌 감지	충돌 감지	충돌 감지	충돌 감지	충돌 감지
감지된 비 직육면체	충돌 감지	충돌 감지	충돌 감지	충돌 감지	충돌 감지	충돌 감지	충돌 감지
피킹된 비 직육면체	충돌 감지	충돌 감지	충돌 감지	충돌 감지	충돌 감지	충돌 감지	충돌 감지
배치된 비 직육면체	충돌 감지	충돌 감지	충돌 감지	충돌 감지	충돌 감지	충돌 감지	충돌 감지

### 포인트 클라우드 구성

프로젝트 실행 시 포인트 클라우드와 관련된 충돌 감지가 필요하면 포인트 클라우드 구성을 선택하여 관련 파라미터를 설정해야 합니다. 구체적인 스텝은 다음과 같습니다:

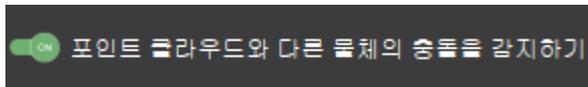
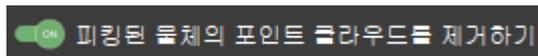


그림 1. **포인트 클라우드와 다른 물체의 충돌을 감지하기** 를 클릭하여 포인트 클라우드 충돌 감지 기능 사용 여부를 선택하세요. 주의해야 할 것은 포인트 클라우드 충돌 감지 기능을 사용하려면 Mech-Vision 에서 send\_point\_cloud\_to\_external\_service 스텝을 먼저 사용해야 합니다.

스텝 2. 포인트 클라우드를 팔진트리 충돌 모델로 전환하는 픽셀을 설정하세요. 여기서 말하는 픽셀은 포인트 클라우드를 복셀 입방체로 전환할 때 복셀마다의 변 길이를 가리킵니다. 같은 포인트 클라우드의 픽셀이 낮을 수록 전환된 복셀 입방체의 수량이 더 많고 계산 결과가 더 정확하지만 시간이 더 오래 걸립니다. 반면에 픽셀이 높을 수록 전환된 복셀 입방체의 수량이 더 적고 계산 결과가 그리 정확하지는 않지만 시간을 절약할 수 있습니다.

**주의:** 포인트 클라우드 충돌 감지를 할 때 사실은 엔드 이펙터와 포인트 클라우드의 포인트가 아닌 복셀 입방체와의 충돌을 감지하는 것입니다.



스텝 3. **피킹된 물체의 포인트 클라우드를 제거하기** 를 통해 피킹된 물체의 포인트 클라우드를 삭제할 지를 선택하세요.

스텝 4. **XY 평면에 추가로 제거된 길이** 와 **Z 축 정방향 추가로 제거된 길이** 의 파라미터를 설정하세요. 여기서 말하는 길이는 포인트 클라우드를 제거할 때 설정된 파라미터 길이 범위 내에 있는 포인트 클라우드를 추가로 제거하는 것을 말합니다.

#### 팁:

1. **XY 평면에 추가로 제거된 길이** 의 파라미터는 너무 큰 값으로 설정하면 주변에 충돌 감지가 필요한 포인트 클라우드도 함께 제거할 수 있으니 너무 크게 설정하지 마세요.
2. 포인트 클라우드의 파동으로 인한 이상한 상황을 피하기 위해 **Z 축 정방향 추가로 제거된 길이** 의 파라미터를 적당히 크게 설정해도 됩니다.

설정이 완료된 후 생성된 충돌 감지 시나리오에는 비전 결과가 포함됩니다. 이 때 진행되는 충돌 감지는 시나리오의 충돌, 자체 충돌 및 포인트 클라우드의 충돌입니다. 아래의 그림과 같습니다:

	로봇 관절	시나리오 물체	엔드 이펙터 (표면)
로봇 관절			
시나리오 물체			
엔드 이펙터 (표면)			
포인트 클라우드	0 mm <sup>3</sup>		0, 0 cm <sup>2</sup>
검출된 비 직육면체			
피킹된 비 직육면체			
놓아두는 비 직육면체			

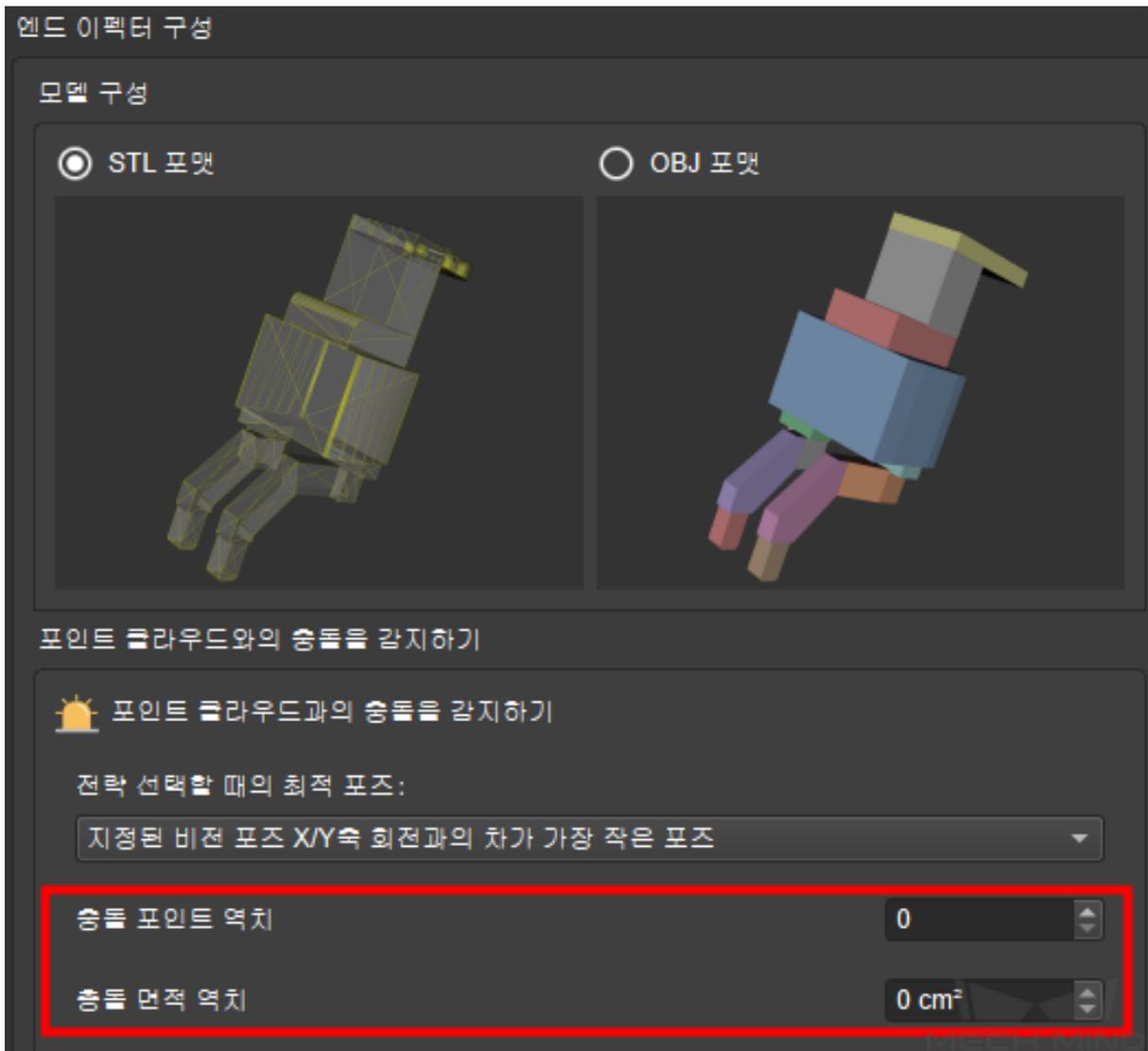
여기까지는 전반에서 포인트 클라우드 충돌 감지를 설정하는 방법을 소개하였습니다. Mech-Viz 에서 일부 이동 태스크 모듈에 대해 포인트 클라우드 충돌 감지를 따로 설정할 수 있습니다. 자세한 정보는 MoveProperty 를 참고하세요.

### 엔드 이펙터 구성

엔드 이펙터의 충돌 감지 기능은 기본적으로 항상 켜져 있습니다. 엔드 이펙터의 구성은 주로 STL 혹은 OBJ 파일 포맷을 선택하고 해당 파라미터에 대해 설정하는 것입니다.

STL 와 OBJ 모델의 차이점은 다음과 같습니다: STL 포맷은 3 차원 물체의 표면 모양만 설명하므로 이 모델을 사용하는 경우에 Mech-Viz 는 모델 표면과 포인트 클라우드 간의 충돌만 감지합니다. OBJ 포맷은 여러 개의 솔리드 볼록다면체의 결합체라서 Mech-Viz 는 모델 표면 및 내부와 포인트 클라우드 간의 충돌을 감지합니다.

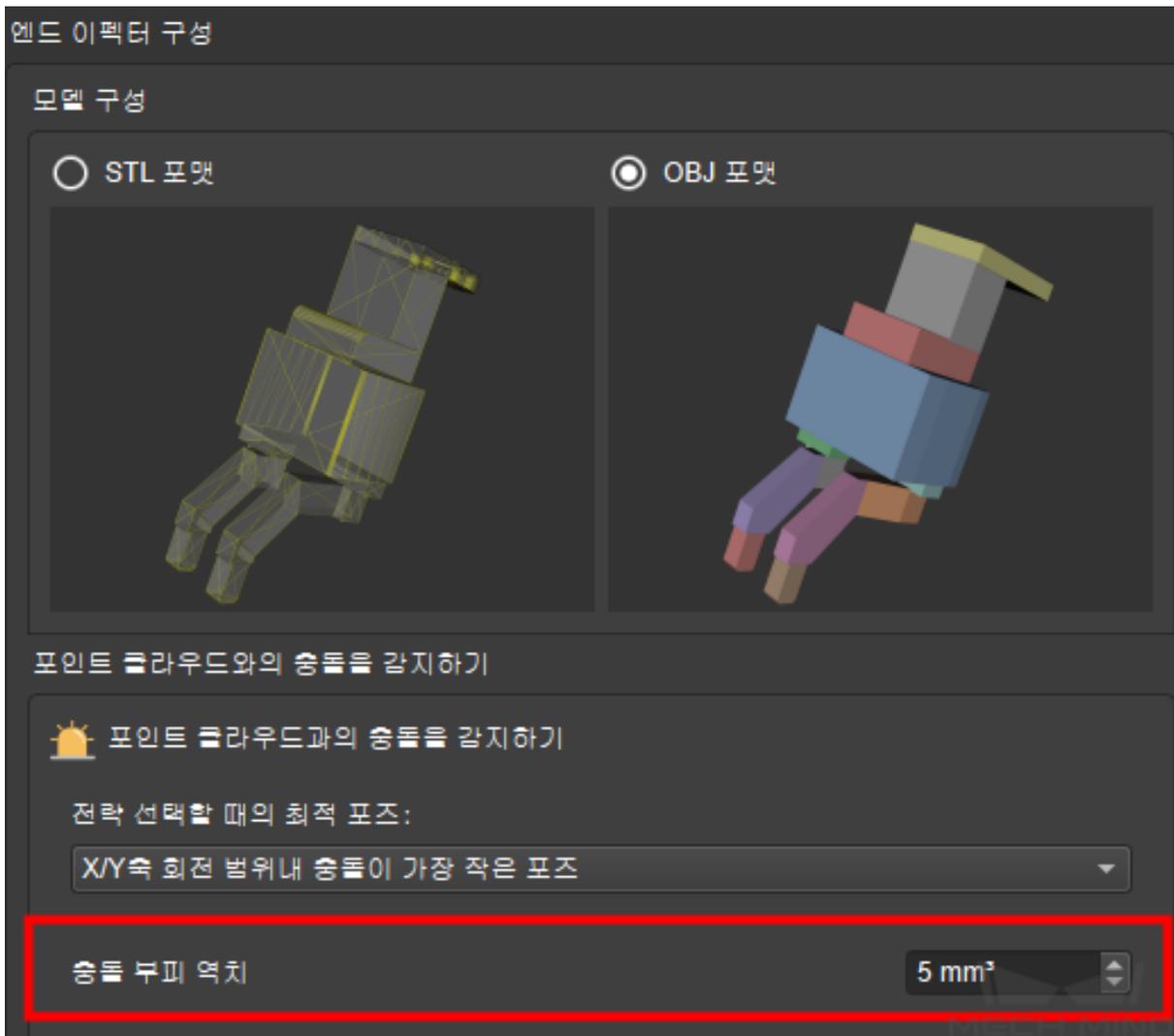
- 아래의 그림과 같이 STL 포맷 의 엔드 이펙터는 포인트 클라우드와의 충돌 포인트 역치 및 포인트 클라우드와의 충돌 면적 역치 설정해야 합니다.



포인트 클라우드와의 충돌 포인트 역치는 피킹 시 모델 표면에서 충돌을 발생하는 포인트 클라우드 수량의 역치를 가리킵니다.

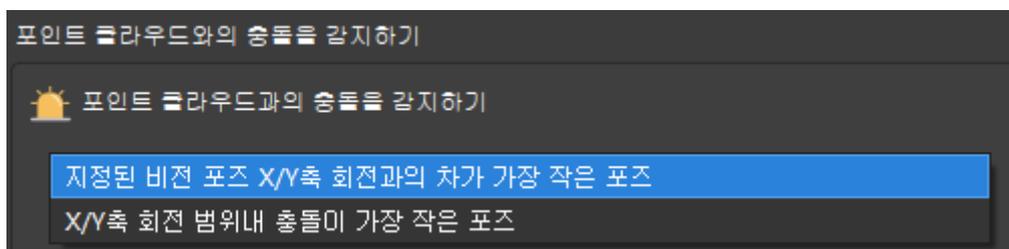
포인트 클라우드와의 충돌 면적 역치는 포인트 클라우드가 한 궤적에서 모델 표면을 휩쓸어 가는 면적의 역치를 가리킵니다.

- 아래의 그림과 같이 OBJ 포맷의 엔드 이펙터는 포인트 클라우드와의 충돌 부피 역치만을 설정해야 합니다.



포인트 클라우드와의 충돌 부피 역치는 충돌 감지 시 엔드 이펙터와 포인트 클라우드간의 충돌 부피의 역치를 가리킵니다.

전략 선택할 때의 최적 포즈는 STL 와 OBJ 엔드 이펙터가 모두 설정해야 하는 파라미터입니다. 아래의 그림과 같이 지정된 비전 포즈 X/Y 축 회전과의 차가 가장 작은 포즈 및 X/Y 축 회전 범위내 충돌이 가장 작은 포즈 등 두 가지의 포즈를 선택할 수 있습니다.



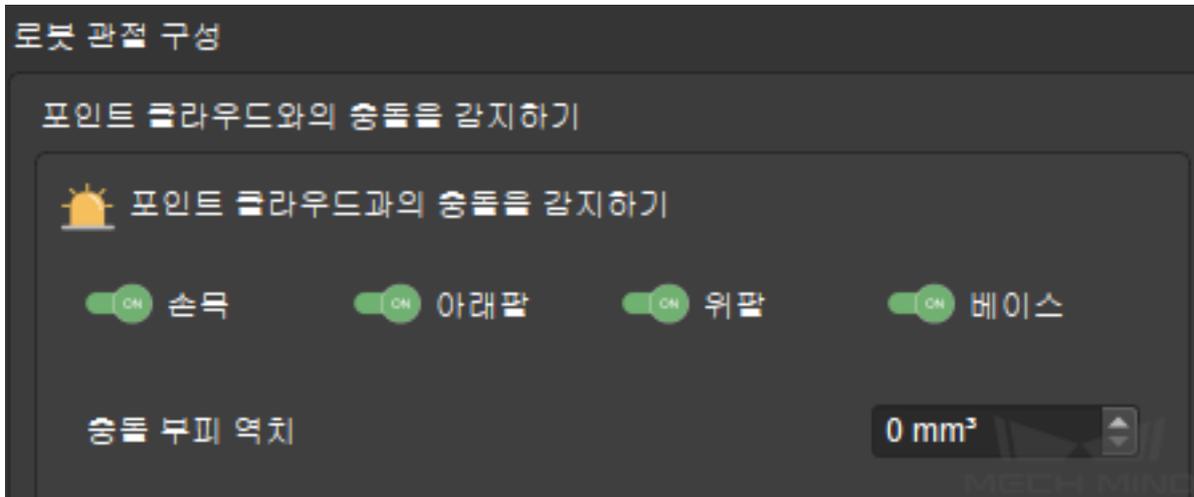
지정된 비전 포즈 X/Y 축 회전과의 차가 가장 작은 포즈란 비전 이동을 통해 출력한 순서가 정해진 포즈가《지정된 포즈》로 여기고 지정된 포즈의 X/Y 축 회전 각도가 가장 작은 포즈에 대해 충돌 감지를 진행하는 것입니다.

X/Y 축 회전 범위내 충돌이 가장 작은 포즈란 X/Y 축 대칭성을 갖는 모든 포인트 클라우드의 충돌을 계산하고 충돌 정도를 작은 것부터 큰 것으로 순서를 배열하는 것입니다.

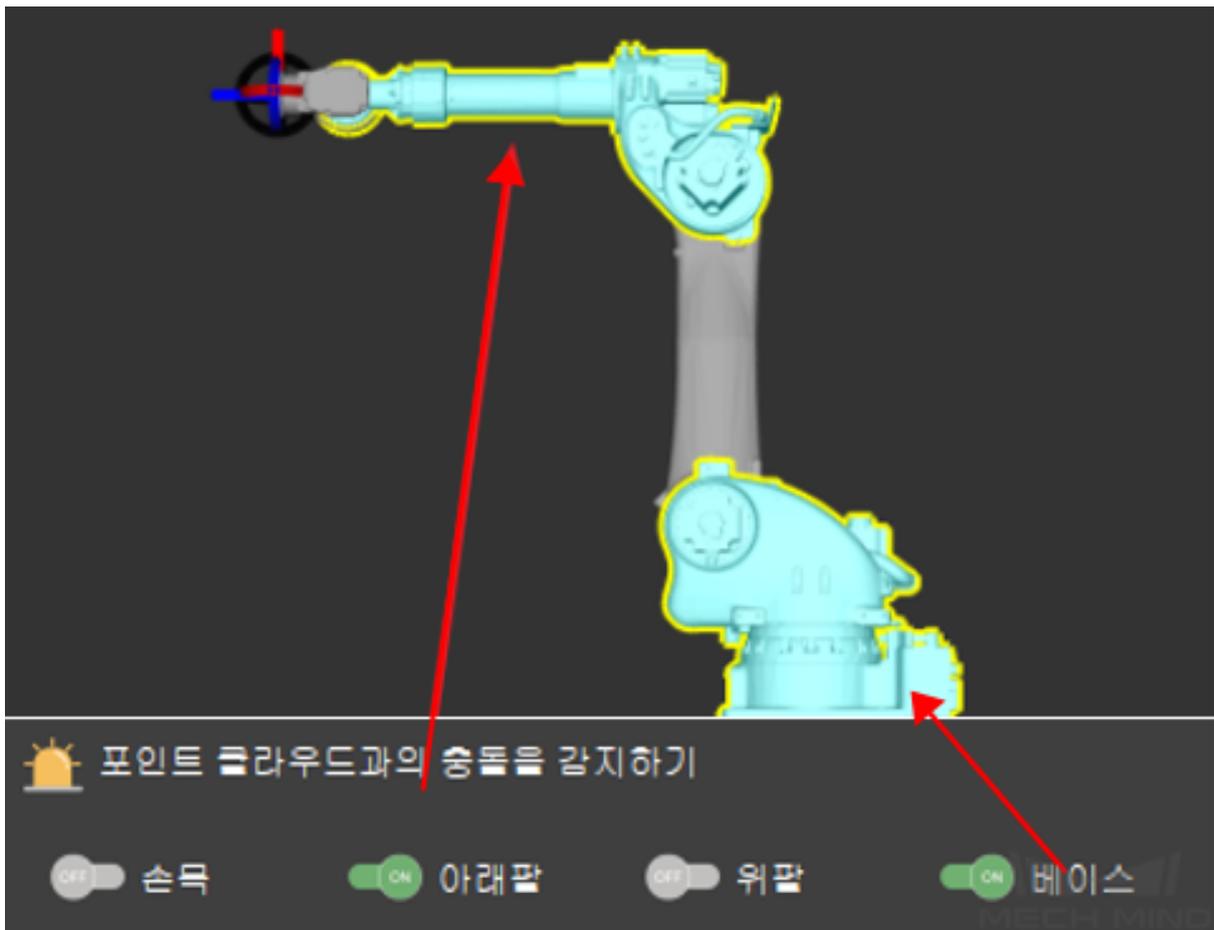
**주의:** 위에서 언급된 두 가지의 파라미터는 모두 포즈의 대칭성을 전제로 하는 것입니다.

### 로봇 관절 구성

로봇 관절의 충돌 감지 기능은 기본적으로 항상 켜져 있습니다. 로봇 관절 구성은 주로 로봇 관절가 포인트 클라우드와의 충돌 감지 파라미터에 대해 설정하는 것입니다. 아래의 그림과 같습니다.



- 로봇은 손목, 위팔, 아래팔 및 베이스로 구성되어 있습니다. 사용자들은 충돌 감지 페이지에서 포인트 클라우드와 충돌 감지를 해야 하는 분위기를 선택할 수 있으며 선택된 분위는 하이라이트로 표시 될 것입니다.

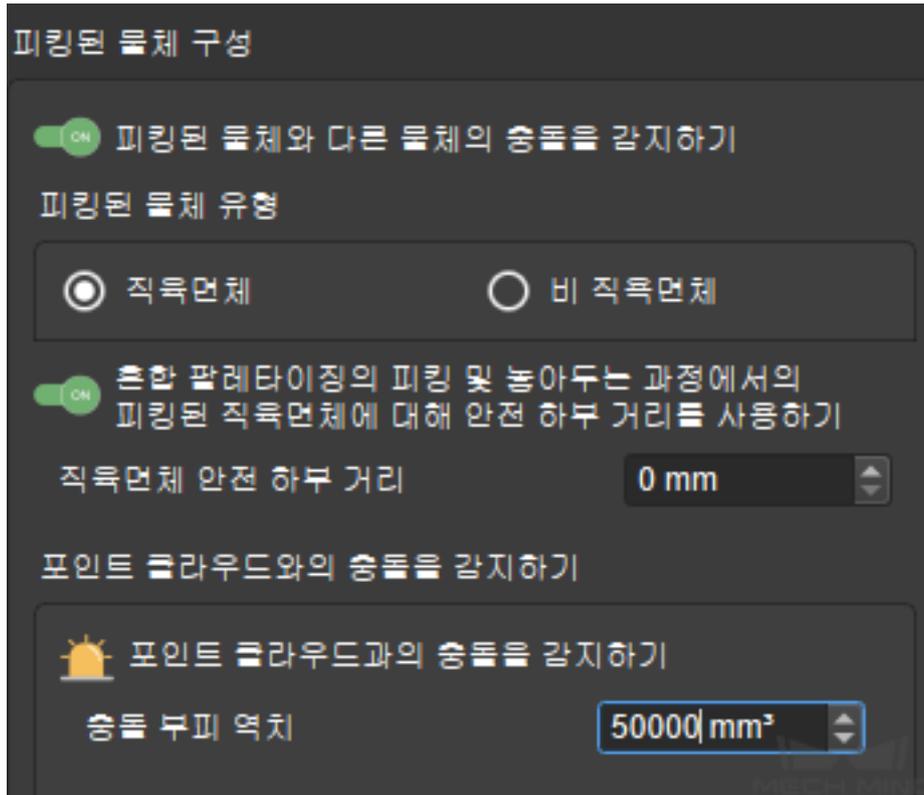


- 충돌 부피 역치 는 충돌 감지 시 로봇이 포인트 클라우드와의 충돌 부피 역치를 가리킵니다. 하지만 로봇은 STL 포맷의 모델이라서 모델 표면과 포인트 클라우드 간의 충돌만 감지하기 때문에 여기서 말하는 충돌 부피는 로봇 표면과 복셀의 충돌 포인트의 개수 및 복셀 부피가 곱한 결과입니다.

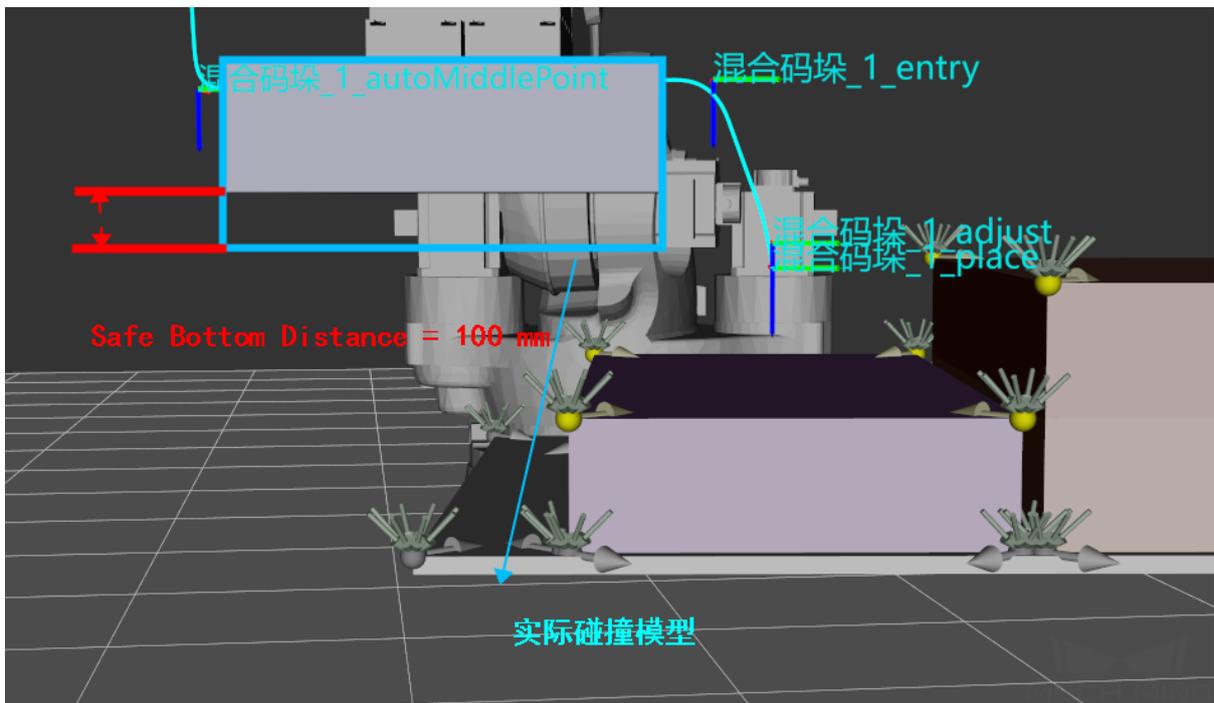
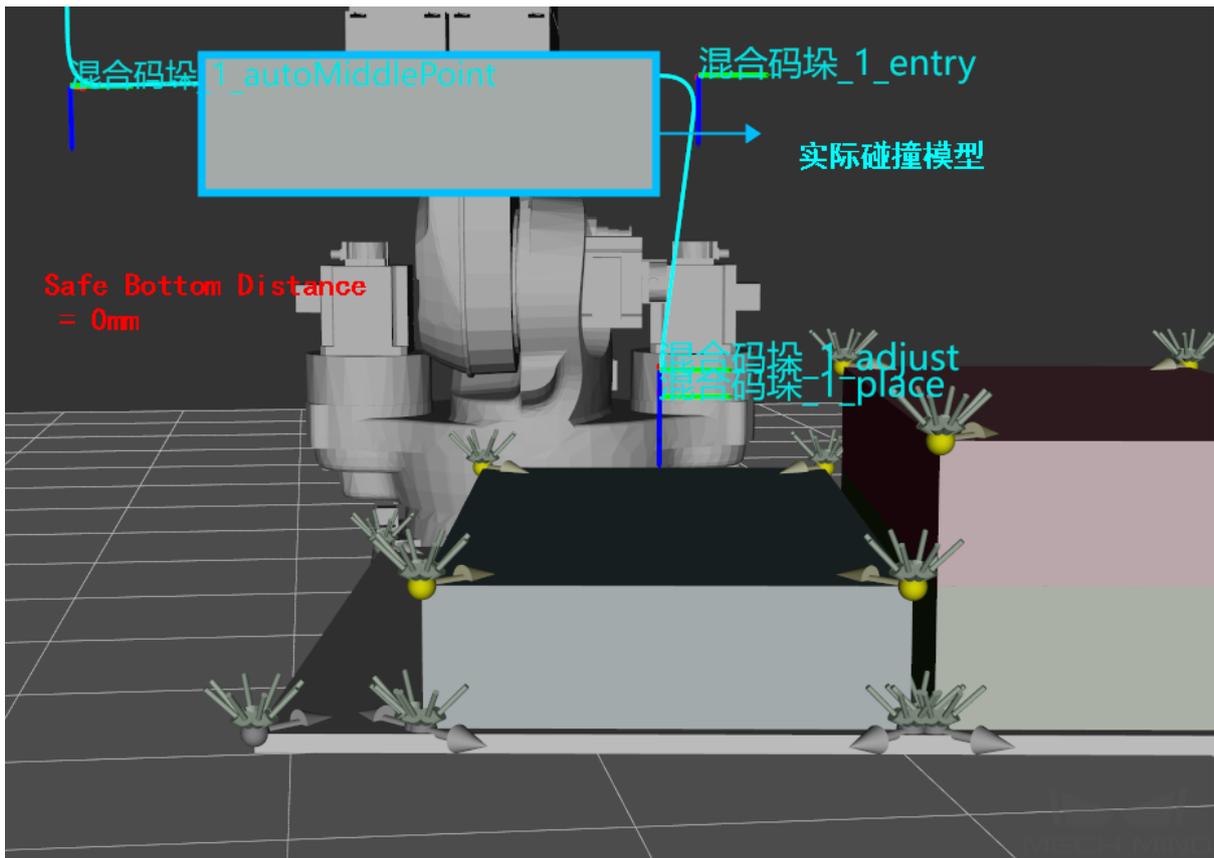
### 피킹된 물체 구성

충돌 감지에 타겟 물체 유형은 주로 직육면체 및 비 직육면체로 나뉩니다. 완전히 다른 물체 유형이지만 설정해야 할 파라미터가 비슷합니다. 단, 두 가지 물체의 모델 및 정보 도입 과정은 많이 다릅니다. 모델 구성에 관한 내용은 **충돌 모델 구성** 을 참고하세요.

- 직육면체: 포인트 클라우드와의 충돌을 감지할 때 상자가 기본 기하체로서 완전한 충돌을 감지하기가 가능합니다. 아래의 그림과 같이 이런 경우에 타겟 직육면체가 이동 과정에서 발생한 포인트 클라우드와의 충돌을 감지하며 충돌 부피에 대해 역치 파라미터를 설정해야 합니다.



피킹된 물체 유형 아래에 하부 안전 거리를 설정할 수 있습니다. 주요 목적은 다음 그림과 같이 상자가 이동할 때 이미 배치된 상자와의 충돌을 피하기 위해 더 안전한 배치 궤적을 선택할 수 있도록 피킹된 상자의 충돌 모델의 하부를 두껍게 만드는 것입니다.



현재 이 기능은 혼합 팔레타이징 시나리오에만 사용되고 반드시 켜져야 합니다. 설정 방식은 매우 간단하

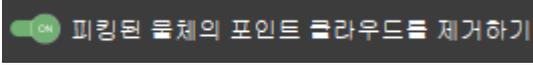
고 **ON** 혼합 팔레타이징의 피킹 및 배치 과정에서의 피킹된 직육면체에 대해 하부 안전 거리를 사용하기 를 클릭하여 하부 안전 거리를 설정하는 기능을 켜고 원하는 값으로 설정하면 됩니다. 추천 값:80mm-120mm .

설정이 완료된 후 생성된 충돌 감지 시나리오에는 피킹된 직육면체가 포함됩니다. 이 때 진행되는 충돌 감지는 시나리오의 충돌, 자체 충돌, 포인트 클라우드와의 충돌 및 피킹된 직육면체와의 충돌입니다. 아래이 그림과 같습니다:

	로봇 관절	시나리오 물체	엔드 이펙터 (표면)	포인트 클라우드	검속된 직육면체	피킹된 직육면체	놓아두는 직육면체
로봇 관절							
시나리오 물체							
엔드 이펙터 (표면)							
포인트 클라우드	0 mm <sup>2</sup>		0, 0 cm <sup>2</sup>				
검속된 직육면체							
피킹된 직육면체	90 mm	90 mm		50000 mm <sup>2</sup>			
놓아두는 직육면체						90 mm	

- **비 직육면체 충돌 부피 역치** 를 설정해야 합니다. 하지만 주의해야 할 것은 이 역치는 피킹된 비직육면체와 인식된 다른 물체 간의 충돌을 판단하는 데도 사용되기 때문에 인식되지 못하는 물체의 포인트 클라우드와의 충돌 부피를 역치로 설정해야 합니다.

위에서 언급된 두 가지 물체는 모두 인식된 물체의 포인트 클라우드를 제거해야 하며 이 기능은 **포인트 클라우드 구성** 에서



를 클릭함으로써 구성할 수 있습니다.

설정이 완료된 후 생성된 충돌 감지 시나리오에는 피킹된 비직육면체가 포함됩니다. 이 때 진행되는 충돌 감지는 시나리오의 충돌, 자체 충돌, 포인트 클라우드와의 충돌 및 피킹된 비직육면체와의 충돌입니다. 아래이 그림과 같습니다:

	로봇 관절	시나리오 물체	엔드 이펙터 (표면)	포인트 클라우드	검속된 비 직육면체	피킹된 비 직육면체	놓아두는 비 직육면체
로봇 관절							
시나리오 물체							
엔드 이펙터 (표면)							
포인트 클라우드	0 mm <sup>2</sup>		0, 0 cm <sup>2</sup>				
검속된 비 직육면체			0 mm <sup>2</sup>				
피킹된 비 직육면체		0		50000 mm <sup>2</sup>	50000 mm <sup>2</sup>		
놓아두는 비 직육면체							

## 충돌 감지 시각화

충돌 감지에 아래와 같이 충돌 감지 시각화와 관련된 설정을 할 수 있습니다.

### 충돌 감지 계산 및 기록 방식을 설정하기

주로 **충돌 계산** 및 **충돌 기록** 의 다른 방식을 선택함으로써 충돌 감지 시각화를 설정합니다. 그림 1 과 같습니다.

충돌 감지	
계산 설정	
충돌을 계산하기	충돌 정도는 역치를 초과하면 남은 계산을 건너뜁니다. (연속 작동에 적용)
충돌을 기록하기	전략 선택에 대응하는 완전한 충돌 접촉을 계산하기(파라미터를 조정하고 충돌 시각화를 편리화하기 위해)
	계획 기록에 저장하지 않기 (연속 작동에 적용)
	계획 기록에 저장하기(느리게 해서 파라미터 조정과 시각화를 편리화할)

그림 1 충돌 감지 시각화

- **충돌 계산** 에서 전략 선택에 대응하는 완전한 충돌을 계산하기 파라미터 조정과 충돌 감지 시각화 위해 및 충돌 정도가 역치를 초과하면 남은 계산을 뛰어넘기 지속적 실행을 위해 등 두 가지 방식을 선택할 수 있습니다.
- **충돌 기록** 에서 계획 기록에 저장하기 느낌, 파라미터 조정과 충돌 감지 시각화 위해 및 계획 기록에 저장하지 않기 지속적 실행을 위해 등 두 가지 방식을 선택할 수 있습니다.

둘 씩 결합하여 다음과 같은 네 가지의 경우를 얻을 수 있습니다:

		충돌 계산	
		전략 선택에 대응하는 완전한 충돌을 계산하기	충돌 정도가 역치를 초과하면 남은 계산을 뛰어넘기
충돌 기록	계획 기록에 저장하기	일반적으로 디버깅에 사용되고 모든 완전한 결과를 계산합니다. 시각화에 사용될 수 있지만 속도가 상대적으로 느립니다.	역치를 초과하면 남은 계산을 진행하지 않고 여전히 시각화 가능하지만 일부 충돌만 표시됩니다. 또한 충돌을 허용할 수 없는 시나리오에서 한 포인트만 존재하는 시각화 결과가 거의 없습니다.
	계획 기록에 저장하지 않기	계획 기록에서 완전한 충돌을 표시할 수 있지만 충돌의 구체적인 위치를 시각화할 수 없습니다.	일반적으로 안정된 생산 시나리오에 적용되고 충돌 정도가 역치를 초과할 때까지 계산하고 계획 기록에 저장하지 않습니다. 계산 속도가 빠릅니다.

일반적으로 지속적 실행에 적용되는 두 가지 방식을 동시에 선택하거나 파라미터 조정과 충돌 감지 시각화를 위한 두 가지 방식을 동세에 선택하는 것이 좋습니다.

### 계획 실시 표시

메뉴 바에서 **표시** → **계획시 충돌을 표시하기** 를 선택해야 합니다. 그리고 계획 기록 아래에 충돌 지속 표시 시산 및 만료 시간을 설정할 수 있습니다.

### 계획 기록 표시

계획 기록에서 해당 사항을 클릭하면 현재 로봇의 포즈를 표시할 것입니다. 더 많은 정보를 알려면 **계획 기록** 를 참고하세요.

### 주의 사항

**주의:** 포인트 클라우드와의 충돌 감지만을 통해 고정된 물체와의 충돌을 완전히 피할 수 없습니다. 고정된 물체는 시나리오에서 직접 모델을 추가하고 더 많은 제한 조건을 설정해야 합니다.

이는 설정된 포인트 클라우드와의 충돌 포인트 역치, 면적 역치 및 부피는 보통 오차 (예를 들어 노이즈, 인식 오차, 클램프 오피셋 등) 를 허용하는 데 사용되기 때문입니다. 또한 인식된 포인트 클라우드가 부족하거나 노이즈가 심각하면, 또한 빛 반사로 인한 포인트 클라우드가 많이 없어지면 포인트 클라우드 충돌 감지의 효과가 그리 좋지 않을 것입니다.

예를 들어 아래와 같은 상황에서 포인트 클라우드 충돌 감지만을 통해 충돌을 피하면 안 됩니다.

1. 크랭크 축의 응용 시나리오에서 충돌 포인트의 역치는 100 인데 100 은 보통 크랭크 축의 포인트 클라우드 오차를 허용하는 것입니다. 만약에 피킹 용기의 테두리 부분에 있는 크랭크 축과 플램프가 충돌하지 않지만 클램프와 피킹 용기가 98 개의 포인트와 충돌하는 경우 피킹 용기의 모델을 추가하지 않았기 때문에 충돌을 감지하지 못합니다. 그리고 포인트 클라우드와의 충돌 포인트 수가 역

치보다 적어서 소프트웨어가 충돌이 없다는 결론을 도출하여 피킹을 계획할 것입니다. 이렇게 하면 클램프가 피킹 용기와 충돌할 것입니다.

- 대부분 시나리오에서 포인트 클라우드는 가장 윗 표면에 대한 것입니다. 그림 1 과 같이 클램프가 위치하는 곳에 피킹 용기의 포인트 클라우드가 없기 때문에 클램프와 피킹 용기 간의 충돌을 감지할 수 없습니다.

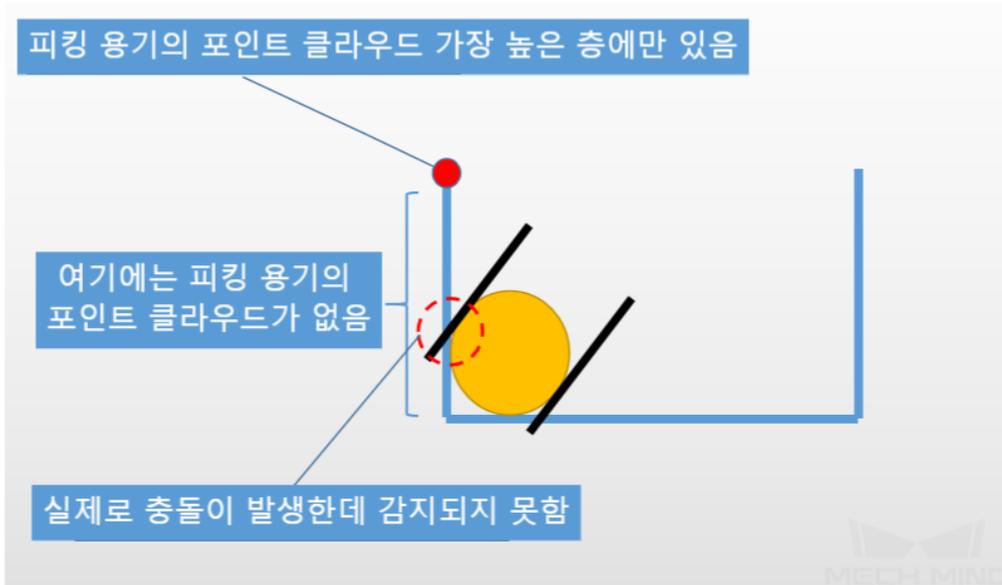


그림 1 충돌을 감지하지 못하는 경우

### 2.3.6 계획 기록

이 페이지에서 프로젝트 실행 시의 계획 내용 및 오류 설명이 표시되어 디버깅에 도움을 줄 수 있습니다. 아래의 그림과 같습니다:

계획 내용	결과	상세 정보	자식 노드 실패한 모든 원인
2022-02-11_14-58-46_575	성공	Start1, Ready_point	
▶ 타겟 포인트의 고정 TCP 포조의 인...	성공		
▶ 이번 피킹에서부터 다음의 피킹(블...	성공	Start1, Ready_point	
2022-02-11_14-58-46_705	성공	Ready_point	
▶ 타겟 포인트의 고정 TCP 포조의 인...	성공		
▶ 이번 피킹에서부터 다음의 피킹(블...	성공	Ready_point	
2022-02-11_14-58-47_250	성공	Ready_point	
▶ 타겟 포인트의 고정 TCP 포조의 인...	성공		
▶ 이번 피킹에서부터 다음의 피킹(블...	성공	Ready_point	
2022-02-11_14-58-47_763	성공	Ready_point	
▶ 타겟 포인트의 고정 TCP 포조의 인...	성공		
▶ 이번 피킹에서부터 다음의 피킹(블...	성공	Ready_point	
2022-02-11_14-58-48_279	성공	Ready_point	
▶ 타겟 포인트의 고정 TCP 포조의 인...	성공		
▶ 이번 피킹에서부터 다음의 피킹(블...	성공	Ready_point	

**계획 내용:** 각 항목 구체적인 시간의 계획 기록을 볼 수 있으며 항목을 펴면 각 레이어의 계획 내용을 확인할 수 있습니다.

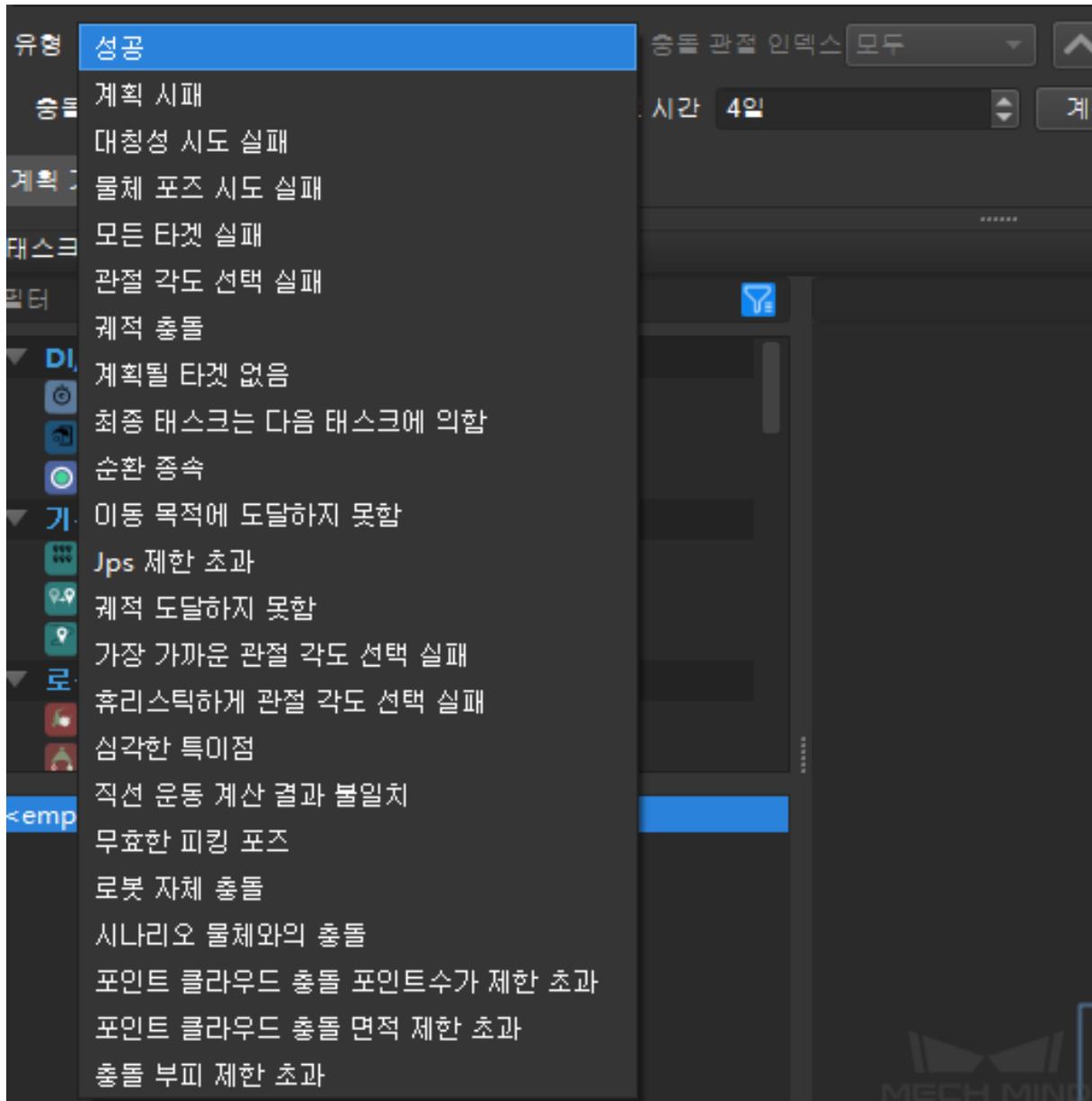
**계획 기록에 있는 충돌 결과의 그래픽 표시:**

- 시나리오의 충돌
  - 충돌이 발생하는 태스크 명칭을 클릭하면 모든 충돌 물체가 하이라이트로 표시될 것입니다.
  - 충돌 물체를 클릭하면 하이라이트로 표시될 것입니다.

- 포인트 클라우드와의 충돌
  - 충돌이 발생하는 태스크 명칭을 클릭하면 모든 충돌 물체가 하이라이트로 표시될 것입니다.
  - 충돌 부피를 클릭하면 충돌이 발생하는 포인트 클라우드가 하이라이트로 표시될 것입니다.
  - 포인트 클라우드와의 충돌을 감지할 때 충돌 포인트를 기록하지 않으면 나머지 충돌 물체 (로봇/엔드 이펙터/타겟 물체) 만 하이라이트로 표시될 것입니다.
- 피킹된 물체의 충돌
  - 충돌이 발생하면 피킹된 물체가 하이라이트로 표시될 것입니다.

**결과** 이번 계획이 성공 여부를 표시할 것입니다. 실패하면 빨간색으로 표시되고 실패한 항목만 펴서 구체적인 원인을 확인할 수 있습니다.

해당 항목을 선택하여 Ctrl+F 를 누르면 바로 다양한 유형의 계획 기록을 볼 수 있습니다. 아래의 그림과 같습니다:



아래의 그림과 같이 이지 밑에서 계획 기록의 충돌 지속 표시 시간, 만료 시간이 표시되고 계획 기록을 로드하거나 클리어할 수도 있습니다.



### 2.3.7 기타

#### 실행 설정

##### 기타



**근사한 Move ID 를 사용하기 (Move ID 를 보내지 못하는 로봇에 대해)** Move ID 를 보내지 못하는 로봇을 위해 근사한 Move ID 를 계산하는 것입니다. 이를 통해 이미 발송한 Move 가 이미 도달한지를 최대한 정확하게 알아내고 Move 발송을 간능한 한 끊기지 않게 만들어 로봇의 이동을 더 유창하게 만들 수 있습니다. 기본적으로 선택되지만 이 기능은 불안정할 때도 있어서 실제 사용 시 문제가 발생할 경우도 종종 있습니다. 문제가 나타나면 이 기능을 닫으세요.

**TCP 포즈를 보내기** Mech-Viz 에서 보내온 포즈가 JPS 인지 혹은 TCP 포즈인지 전환하는 것입니다.

**로봇 서비스 타임 아웃** 이동을 빼고 다른 요소가 로봇과 통신하는 데 시간 제한을 초과한 시간입니다. set\_do check\_di 등 있습니다.

**기본적인 물체 계획 구성만 사용하기** 이 기능을 켜면 다종의 물체 레이블이 있는 경우에도 여전히 기본적인 물체의 대칭성을 사용합니다.

## 태스크 수집

**이동 태스크 최대 수집 개수, 비 이동 태스크 최대 수집 개수** Mech-Viz 가 무한 루프가 되어 너무 많은 포즈를 보내는 것을 방지하면서 매번 발신하는 명령 수에 대해 제한합니다.

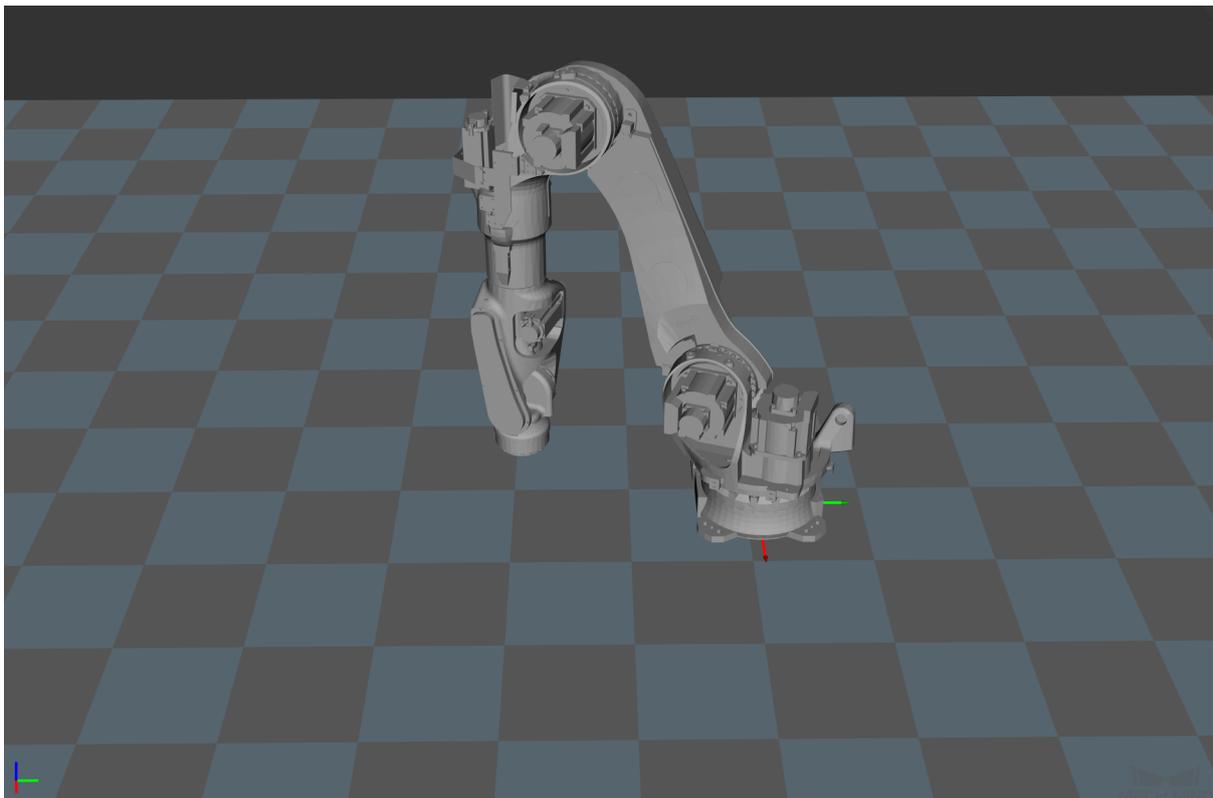
## 특이점

**특이점 역치** 궤적을 계획할 때 특이점 역치 (관절의 최대 각속도 - 라디안 법) 에 도달했는지 여부를 판단합니다. 일반적으로 디폴트 설정을 사용합니다. 특이점 감지에 보다 민감한 로봇을 사용하는 경우 (예를 들어 소프트웨어 시뮬레이션 시의 궤적이 정상 상태인데도 불구하고 로봇이 실제적으로 그 궤적에 따라 이동하면 특이점 에러가 나옴) 이 값을 적절히 줄여 주세요.

**특이점 감속비** 특이점이 나타나고 감속비가 이 값보다 큰 경우에 속도를 낮춤으로써 특이점 문제를 해결할 수 있다고 봅니다. 이 값을 적절히 작게 설정하면 “심각한 특이점”으로 인해 계획 실패 가능성을 낮출 수 있습니다.

## 전반 관절 제한

**어깨/팔꿈치/손목을 뒤집지 않기** 물체를 쥐고 있을 때 혹은 전반 실행 과정 두 가지 경우에 로봇의 불필요한 뒤집기를 줄일 수 있습니다. 하지만 어깨/팔꿈치/손목 모두 “뒤집지 않기”로 설정하는 것은 최적인 선택이 아닐 수도 있습니다. 예시는 다음과 같습니다:



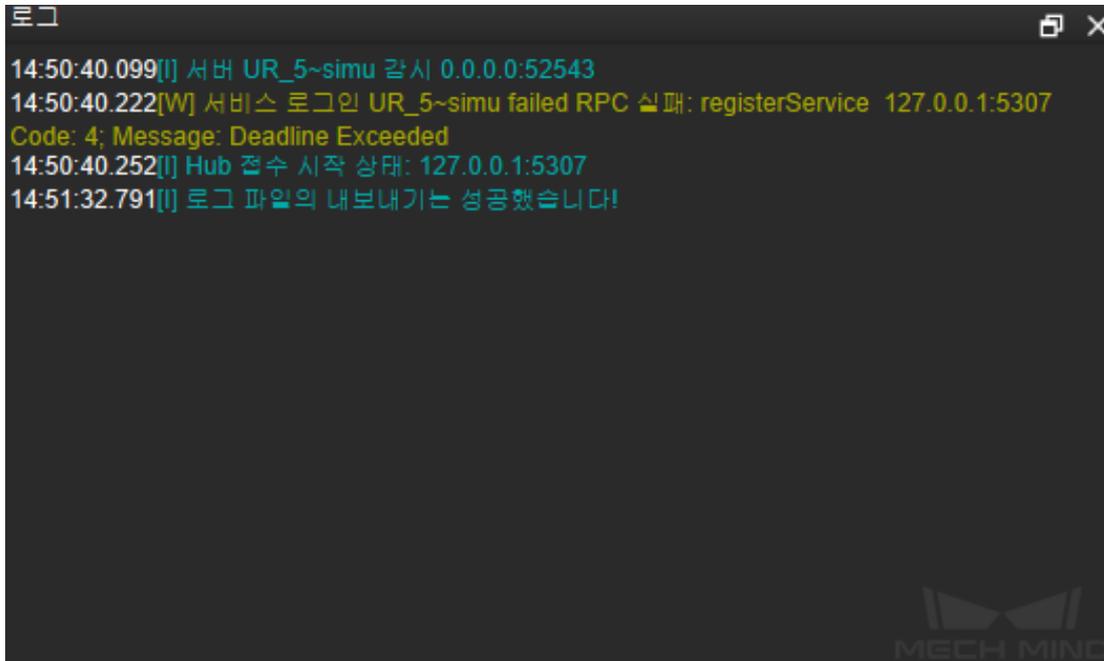
로봇이 이동 중에 위의 자세가 나타나면, 로봇의 어깨/팔꿈치/손목이 뒤집힐 때 각 축 간에 상호 관계가 존재하므로 강제로 손목을 뒤집지 않도록 설정하면 로봇이 보다 큰 각도로 뒤집힐 수도 있습니다. 따라서 이런 경우에 어깨를 뒤집지 않기 및 팔꿈치를 뒤집지 않기 만 설정하는 것이 좋습니다.

**참고:** 이동 태스크의 JpsConstraint 의 우선 순위가 이 때의 전반 설정보다 높기 때문에 이동 태스크에서

설정을 하지 않으면 전반 설정이 발효될 것입니다.

### 2.3.8 로그

그림 1 과 같이 이 페이지에서 Mech-Viz 의 실행 상태를 확인할 수 있습니다.



```

로그
14:50:40.099[I] 서버 UR_5~simu 감시 0.0.0.0:52543
14:50:40.222[W] 서비스 로그인 UR_5~simu failed RPC 실패: registerService 127.0.0.1:5307
Code: 4; Message: Deadline Exceeded
14:50:40.252[I] Hub 접속 시작 상태: 127.0.0.1:5307
14:51:32.791[I] 로그 파일의 내보내기는 성공했습니다!
    
```

그림 1 로그 페이지

로그의 레벨은 낮은 레벨부터 높은 레벨로 결함 수정 (Debug), 정상 정보 (Info), 경고 (Warning), 핵심 오류 (Critical) 및 치명적 오류 (Fatal) 로 나뉘집니다. **설정** → **로그 레벨** 에서 원하는 로그 레벨을 선택할 수 있습니다. **그림 2** 와 같이 어떤 레벨을 선택한 후 선택된 로그 및 더 높은 레벨의 로그를 출력할 것입니다.

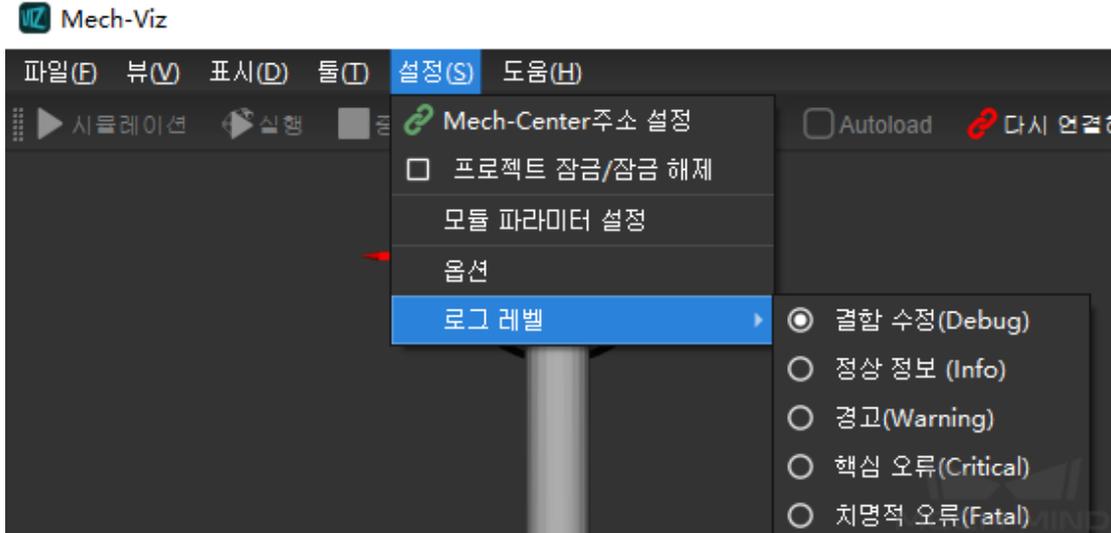


그림 2 로그 레벨

그림 3 과 같이 일반 설정에서도 로그와 관련된 내용을 설정할 수 있습니다.

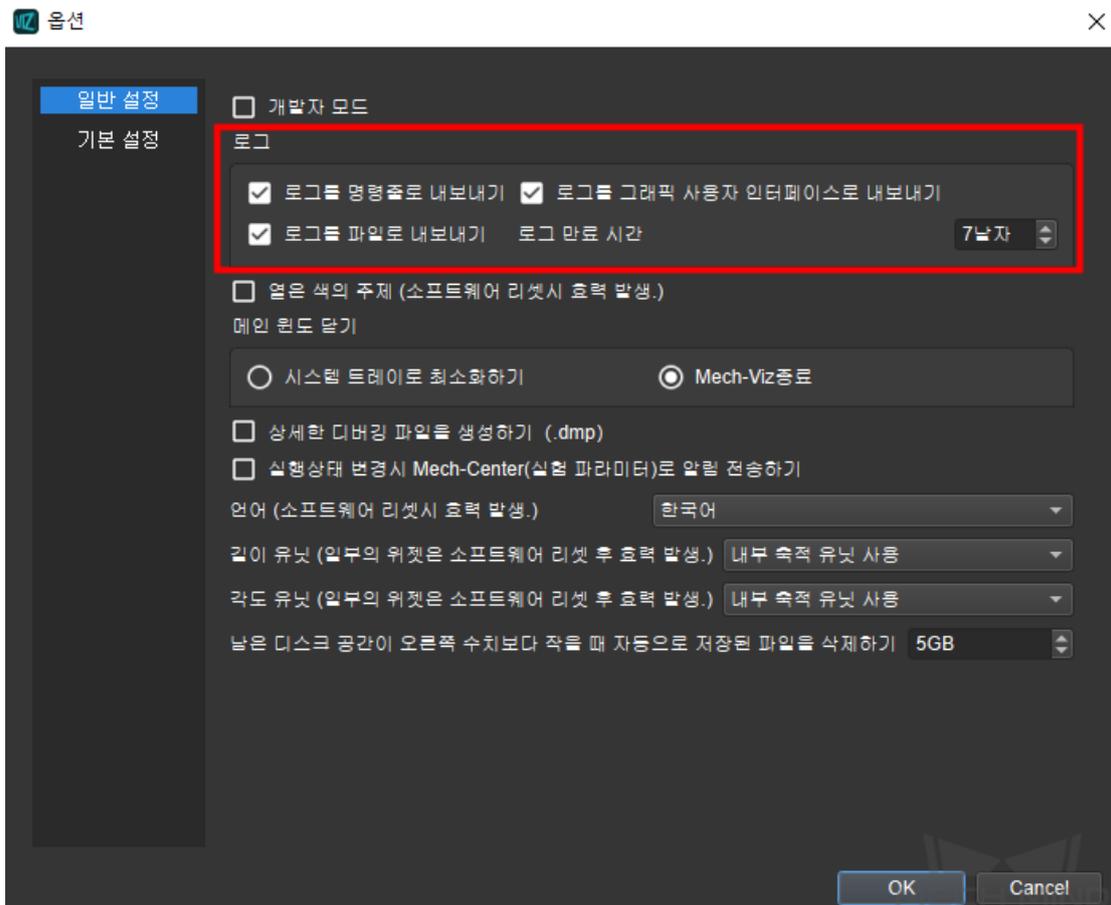


그림 3 로그 설정

1. 환경 데이터 모델을 들여올 수 있습니까? 예. 들여올 수 있습니다. 자세한 정보는 [시나리오](#) 를 참고 하세요.