

## 6. 로봇과 시뮬레이터

### 1. 서브 시스템

- 복잡한 로봇이더라도 하나의 서브 시스템만 고려함으로써 쉽게 설계와 분석이 가능
- 분류
  - 1) 구동(Actuation)
    - : 로봇의 바퀴 또는 팔이 움직이는 방법과 직접 상호 작용하는 시스템
  - 2) 감지(Sensing)
    - : 카메라나 레이저 스캐너와 같은 센서 하드웨어와 직접 상호 작용하는 시스템
  - 3) 계산(Computing)
    - : 로봇이 유용한 작업을 수행할 수 있도록 비교적 지능적인 처리와 함께 액추에이터와 감지를 결합하는 것

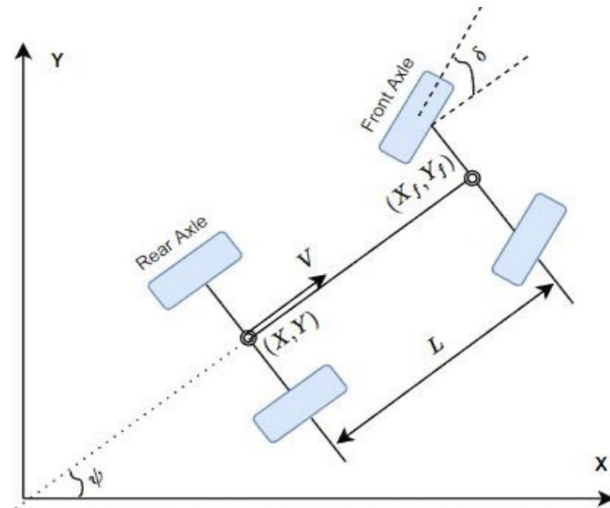
#### 1-1-1. 구동 : 이동 플랫폼

- 로봇의 기본적인 능력은 이동(locomotion)하는 기능
- 이동체(mobile base) : 로봇을 움직이게 하는 액추에이터의 모임
- 차동 구동(differential drive) 플랫폼



- 독립적으로 구동되는 두 개의 바퀴로 구성
- 하나 이상의 캐스터(caster : 로봇의 앞뒤를 지지하도록 자유롭게 회전하는 무전원 바퀴)를 가짐
- 정적으로 안정된 로봇 -> 모델링과 제어가 쉬움
- 동적 안정 / 균형 구동(dynamically stable/balancing) 바퀴
  - 액추에이터가 안정성을 유지하기 위해 지속적으로 움직임
  - 작은 장애물 위로 부드럽게 구동
  - 좁은 영역에서 유용하게 사용 가능
- 미끄럼 조향(skid steering) : 두 바퀴 이상일 때 확장 될 수 있는 역할
  - 기계적인 단순성 유지, 최대 마찰력 제공
  - 정확하게 앞뒤로 구동되지 않을 경우에는 지속적으로 미끄러짐
  - 저속으로 회전을 계속할 경우 엄청난 에너지 소모함

- 애커맨(Ackerman) 플랫폼



- 뒷 바퀴는 항상 앞으로 똑바로 향하고, 앞바퀴들은 함께 회전하는 구조 (자동차 구조)
  - 바퀴를 이동체의 가장자리에 존재하므로 지지하는 다각형의 면적을 최대화하므로 자동차가 뒤집히지 않음
  - 옆으로 운전하기 어려움 -> 정교한 계획과 순차적인 액추에이터 조작이 필요
  - 논홀로노믹(non-holonomic) / 홀로노믹(holonomic) 플랫폼
    - 어느 방향으로든 움직일 수 있는 기능 : 홀로노믹(holonomic), 메카넘 휠(mecanum)
  - *Twist* 메시지
    - : 선 속도, 각 속도를 3차원에서 표현하는 방법
    - 이동 플랫폼과 관련된 ROS 소프트웨어에서 거의 항상 사용하는 메시지
    - 차량의 기구학(kinematics)을 추출 또는 일반화할 수 있음
- \* 고급 소프트웨어 관점에서는 이동 플랫폼의 액추에이터가 차동 구동이든지 애커맨 조종이든지 상관없음

### 1-1-2. 구동 : 로봇 팔

- 관절로 연결된 일련의 단단한 링크로 구성
- 회전 관절(revolute joint, pin), 선형 관절(linear joint, prismatic)
- 자유도(DOF, Degree of Freedom)
  - : 로봇 팔의 크기, 무게, 숙련도, 비용, 신뢰도 등의 가장 중요한 결정 요소 중 하나
- 작업 공간
  - : 로봇 팔이 도달할 수 있는 공간
  - 자유자재 작업 공간(dextrous)
    - : 로봇의 말단 장치(end-effector)가 도달할 수 있는 모든 위치와 방향에 속하는 작업 공간의 일부
  - 로봇의 여섯 관절에서 완전한 운동 범위를 적당한 비용으로 얻는 것은 제약 조건들에 의해 어려움
    - > 7 번째 자유도 (손목의 위치와 방향은 유지한 채로 팔의 관절을 움직이는 데 쓰일 수 있는 자유도)

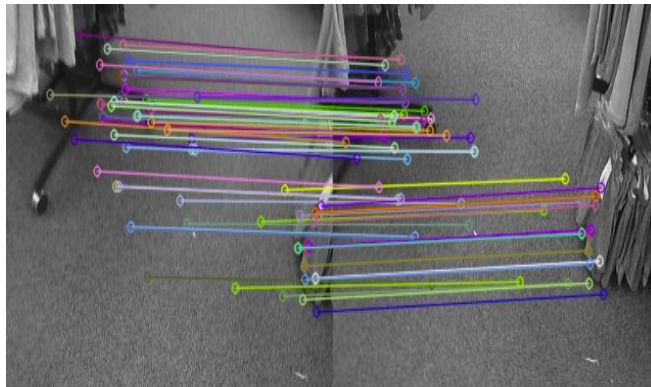
## 1-2. 센서

- 2진(binary) 센서 : on이나 off의 상태를 가지는 기계적인 제어 스위치
  - 광학 제어 스위치 : 광선을 차단하는 기계적인 신호(flag)를 사용
  - 범프 센서(bump sensor) : 기계적인 압력을 비교적 먼 거리에 있는 단일 기계 스위치로 전달
- 수치(scalar) 표시 반환
  - 압력 센서 : 기계적인 압력이나 기압 추정, 감도 범위에 있는 숫자 값 출력
  - 거리 센서 : 소리, 빛 등과 같은 물리적인 현상으로부터 만들어져 범위 안의 숫자 값을 반환
- 각각의 센서는 실제 관점을 왜곡시키는 특이성을 가짐 -> 적절한 센서 처리 알고리즘 조정 필요
- 센서 헤드(sensor head) : 핀/틸드(pan/tilt) 구조물 위에 두어 필요할 때 원하는 각도로 회전 가능
  - > 작업 공간 한가운데의 가시선을 유지하기 위해 물리적인 울타리 내에서 센서들을 통합하는 플랫폼 위에 존재

### 1-2-1. 광학 카메라

: 카메라는 사고 원격 조작에서 유용하기 때문에 로봇 센서 헤드에서 흔히 보임

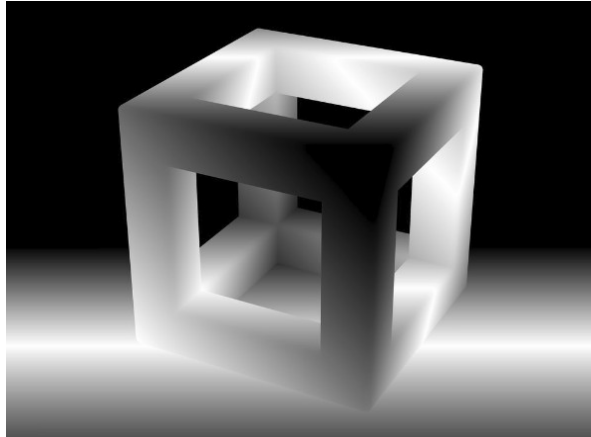
- 3D 데이터
  - 2D 보다 로봇 작업과 환경을 설명하는 것이 수학적으로 훨씬 더 견고
  - 조명, 그림자, 차단 등에 불변하는 특성
- 스테레오 카메라
  - : 두 카메라가 동일한 기계적 구조 위에 고정되어 세워짐



- 두 카메라 간 미세한 차이를 통해 다양한 특징에 대한 거리를 추정하는데 사용
- 카메라의 기계적인 설계의 질, 해상도, 렌즈 종류와 질 등에 의해 성능이 조절
- 대비가 높은 모서리처럼 장면에서 수학적으로 구별 가능한 특징과의 거리만 추정 가능
- `sensor_msgs/Image`
  - : ROS 에서 영상에 대한 표준적인 ROS 메시지 자료형
  - 영상의 크기, 픽셀 부호와 구조, 픽셀 자체만 포함
- `sensor_msgs/CameraInfo`
  - : 렌즈와 센서 정렬로부터 야기된 카메라의 고유한 왜곡을 설명 (OpenCV)
- `cv_bridge` 패키지
  - : OpenCV 와 상호 연동 가능한 ROS 패키지

## 1-2-2. 깊이 카메라

: 텍스처(texture) 패턴을 표면에 비추어 카메라 영상을 얻음



- 물체의 색에 대한 시스템의 민감도를 감소 & 근적외선 파장으로 동작
- 스테레오와 다르게 능동적인 장치
- Kinect (Microsoft)



- 1) 구조광(structured light) 영상 투사
- 2) 장면에 정확하게 알려진 패턴을 투사
- 3) 카메라가 패턴이 장면의 다양한 물체와 표면 위에 떨어질 때 어떻게 왜곡되는지 관찰
- 4) 재생 알고리즘(reconstruction algorithm)을 통해 장면의 3D 구조 추정

### - time-of-flight

- : LED 또는 레이저 발광기를 빠르게 깜빡이면서 날아갔다가 되돌아오는 광 펄스에 대해 요구되는 시간을 추정하기 위해 이미지 센서에 특별하게 설계된 픽셀 구조를 사용
- 비행 시간이 추정되면 추정 값을 깊이 영상으로 변환하는데 쓰임

### - 점군(point cloud)

- : 카메라를 접하는 표면에 놓인 것으로 추정되는 3D점들을 나타냄

### - *sensor\_msgs/PointCloud2*

- 비구조적인 점군 데이터를 허용
- 깊이 카메라가 영상에서 각 픽셀에 대한 유효한 깊이 추정치를 반환할 수 없으므로 때로는 유리

### 1-2-3. 레이저 스캐너

: 초당 10~80번 회전하는 회전 거울판상에서 레이저 빔을 쏘는 장치



- 1) 거울판이 회전할 때 레이저 빛이 빠르게 펄스를 내보냄
- 2) 반사된 파형과 전파된 파형을 상호 연관시킴
- 3) 스캐너 주변을 따라 일련의 각도에 대한 레이저 펄스의 비행 시간을 추정

- *sensor\_msgs/LaserScan*

: ROS에서 레이저 스캔 값을 저장하는 곳

### 1-2-4. 축 인코더

: 축에 표지(marker)를 붙이고, 로봇의 몸통 또는 팔의 이전 링크와 같은 다른 참조 프레임에 상대적인 운동 측정

- 인접한 참조 프레임(frame of reference)에 상대적인 축상 표지의 각 위치(angular position)를 측정

- 주행기록(odometry)

: 단순히 구동 바퀴가 얼마나 많이 회전하였는지 값으로 나타냄 -> 추측 항법(dead reckoning)

- 주행기록에 의한 편차와 바퀴에 대한 불완전성에 의해 완전하지는 않음

- 로봇 팔에서 각각의 회전 관절을 위해 가짐

-> 로봇 팔 설정(manipulator configuration) : 축 인코더를 읽은 값들의 벡터

- *geometry\_msgs/Transform*

: 주행기록 추정 값을 공간 변환(spatial transformation)값으로 전달

공간 변환(spatial transformation) : 다른 참조 프레임에 대해 상대적인 하나의 참조 프레임을 나타냄

주행기록 변환 : 전원을 인가하였을 때나 인코더가 마지막으로 초기화되었을 때의 로봇 위치에 대해 상대적인 축 인코더의 주행기록 추정 값

- *sensor\_msgs/JointState*

: 라디안 단위의 각도와 초당 라디안 단위의 각 속도 벡터를 포함

- 로봇 팔의 상태를 최소한으로 완벽하게 설명 가능

### 1-3. 계산

- ROS는 독립적인 소프트웨어 노드 간 데이터 전달을 위해 동적인 메시지 전달 그래프를 사용

-> POSIX 프로세스 모델 사용

1) 한 노드로부터 오는 메시지를 직렬화

2) 다른 노드로 프로세스 간 또는 네트워크 통신 방식으로 전송

3) 또 다른 노드를 위해 역 직렬화

-> 추가적인 CPU 사이클 필요하나, 빠른 프로토타이핑과 소프트웨어 통합에 대한 장점이 훨씬

큼