

1월 1주차 레포트

1. PD Control

(1). BVH Rotation 값 추출

```
Z_angle=self.walkfile.frame_joint_channel(self.frame,i,'ZROTATION')*math.pi/180
X_angle=self.walkfile.frame_joint_channel(self.frame,i,'XROTATION')*math.pi/180
Y_angle=self.walkfile.frame_joint_channel(self.frame,i,'YROTATION')*math.pi/180
>> BVH 파일에서 각각의 ROTATION CHANNEL들의 값들은 각각의 축들에 대한 Degree값
>> Degree값을 Radian으로 변환시켜줘야함
```

(2). Euler Angle Matrix

$$R_x(\phi) = \text{Roll}(\phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & -\sin \phi \\ 0 & \sin \phi & \cos \phi \end{bmatrix}$$

$$R_y(\theta) = \text{Pitch}(\theta) = \begin{bmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ -\sin \theta & 0 & \cos \theta \end{bmatrix}$$

$$R_z(\psi) = \text{Yaw}(\psi) = \begin{bmatrix} \cos \psi & -\sin \psi & 0 \\ \sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

```
Z_ROT=np.array([[np.cos(Z_angle), -np.sin(Z_angle),0],
                 [np.sin(Z_angle), np.cos(Z_angle),0],
                 [0,0,1]])

X_ROT=np.array([[1,0,0],
                 [0,np.cos(X_angle), -np.sin(X_angle)],
                 [0,np.sin(X_angle), np.cos(X_angle)]])

Y_ROT=np.array([[np.cos(Y_angle),0,np.sin(Y_angle)],
                 [0,1,0],
                 [-np.sin(Y_angle),0,np.cos(Y_angle)]])

>> Z,X,Y 순으로 최종 Rotation Matrix 형성

Z_ROT@X_ROT@Y_ROT
```

(3). Torque 계산

- For 3 DOFs rotational joints:

$$\tau = k_p(\log(R^T R_d)) + k_d(\dot{q}_d - \dot{q})$$

```
R^T : vector=np.array(self.Human.getJoint(i).getPositions())
      R = (R.from_rotvec(vector)).as_matrix()
      R^T = np.transpose(R)
      >> "scipy" 라이브러리를 활용하여 Rotation Vector에서
      >> Rotation Matrix 추출
Rd   : Z_ROT@X_ROT@Y_ROT

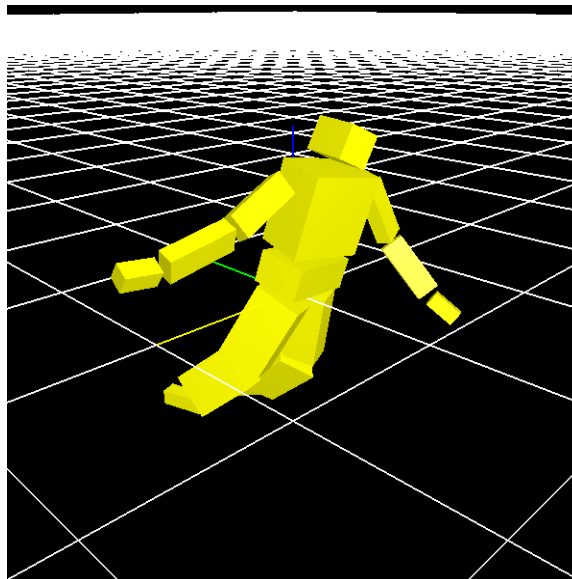
qd'  : 0
      >> 0으로 가정

q'   : vector=np.array(self.Human.getJoint(i).getVelocities())

log  : (R.from_matrix(R^T@Rd)).as_rotvec()

Torque : Kp(상수)*log + Kd(상수)*(0-q')
```

2. Dart를 활용한 구현



(1). 문제점

- >> 전반적으로 걷는 동작의 움직임이 보이지 않음
 - >> 그나마 다행인건 움직일 때의 위아래의 움직임
 - >> 무릎 관절의 굽혀지는 동작이 보여지기는 함
 - >> 팔을 제대로 앞뒤로 움직이지 않음
- >> 팔, 다리를 따로 실행하여도 마찬가지로
 - >> 팔과 다리의 각각 한 부분씩만 돌릴 시, Kp와 Kd를 조절하여
 - >> BVH 파일에서 제공하는 값들처럼 움직임 구현이 가능하나
 - >> 단일 부분에 맞춰진 Kp와 Kd를 활용하여, 팔,다리 전체를 구현할 시
 - >> 움직임이 튕김 (토크나 다른 값들이 너무 커져서 오류 발생)
- >> 주어진 frame 200개에서 걸음동작이 몇번이나 구현되는지 파악필요
 - >> 무릎관절의 각도로 봤을 때,
 - 두 걸음(왼발, 오른발 한번씩)이 총 3번 실행되는 것으로 판단됨
 - >> 팔, 다리를 각각 돌렸을 시에도 총 3번의 걷는 움직임이 구현되어야함

(3). 개선

- >> 전반적으로 걷는 동작의 움직임이 보이지 않음
 - >> Pyqt5의 timer, PD Control, frame이 동시에 업데이트된 것이 문제점
 - >> Pyqt5의 timer, PD Control 같은 frequency로 진행
 - >> frame은 timer가 20번 진행될 시 다음 frame으로 업데이트
- >> 팔, 다리를 따로 실행하여도 마찬가지로
 - >> 전체부분으로 할 시, Kp와 Kd를 축소하는 방향으로 진행
 - >> 위의 frame을 조절하니 보다 움직임이 튕기는 현상 감소
- >> 주어진 frame 200개에서 걸음동작이 몇번이나 구현되는지 파악필요
 - >> BVH 파일에서는 걷기 동작만 연속적으로 존재하는 것 X
 - >> 서있는 자세 - 걷기 준비 자세 - 걷기 (2~3회) - 걷기 종료 준비 자세 - 서있는 자세
 - >> 그리고 처음 파일을 실행할 시, 안정화 시간 필요