

# 컴퓨터 애니메이션

## 실습 보고서

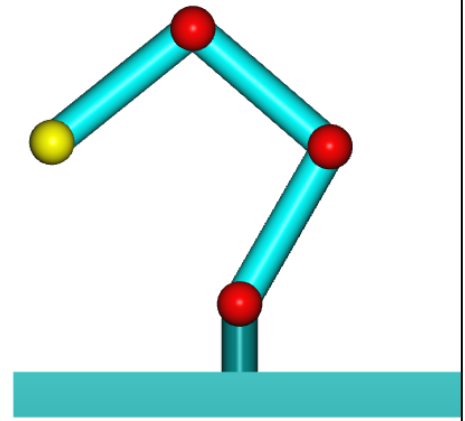


Self-Scoring Table

	P1	P2	P3	E1
Score	1	1	1	1

## P1 - Jacobian transpose method

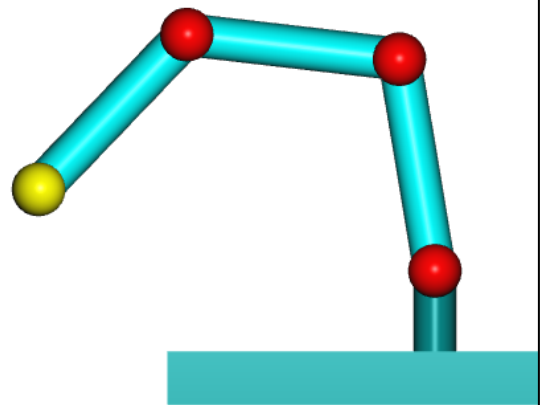
```
Status: Monitor 344mm x194mm  
Status: Screen 1440x 810  
Status: Framebuffer 1440x 810  
Status: Renderer AMD Radeon  
Status: Vendor ATI Technologies Inc.  
Status: OpenGL4.6.14800 Core Profile  
reshape(1440, 810) with success  
Keyboard Input: [1:3] for the  
Keyboard Input: left/right  
Keyboard Input: j for Jacobian  
Keyboard Input: p for the  
Keyboard Input: d for the  
Mouse Input: desired position  
IL iterations = 33
```



Jacobian transpose method 방법으로 IK를 구현했을 때, iteration의 횟수가 다른 방법보다 많다.

## P2 - Pseudoinverse method

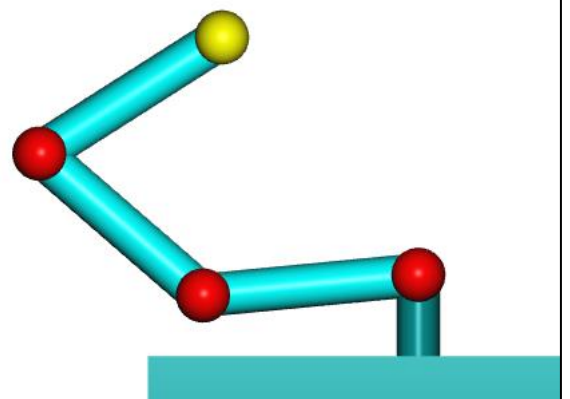
```
IL iterations = 2  
IL iterations = 3  
IL iterations = 2  
IL iterations = 6  
IL iterations = 3  
IL iterations = 6  
IL iterations = 6
```

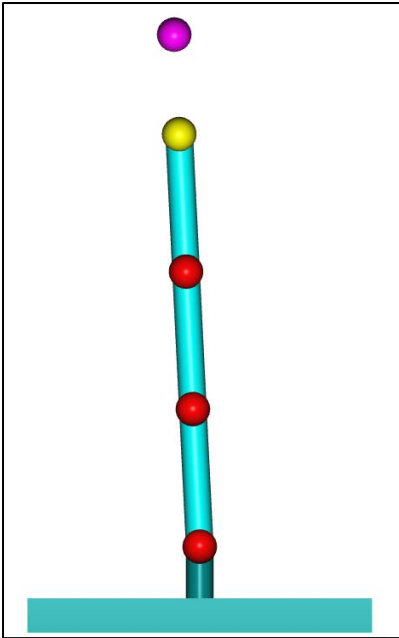


Pseudoinverse method로 IK를 구현했을 때, Jacobian method 보다 iteration이 줄어 들었다.

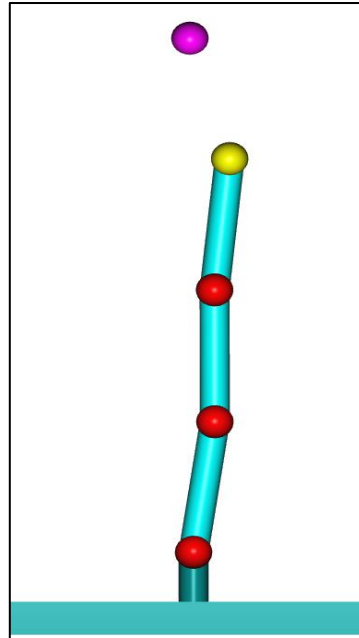
## P3 - Damped least squares method

```
Mouse Input: desired position  
IL iterations = 11
```





[ Damped least squares method ]



[ Pseudo Inverse method ]

Damped least squares method를 사용하면 out of reach 상태일 때, 튕기지 않는다. 사진에 안담겼지만 Pseudo Inverse method를 사용하면 out of reach일 때, 불안정하다.

## E1 - Determine whether a solution is found in the “out of reach” state

[구현 코드]

```

407 float dthetaSum = 0.0f;
408 // Update the joint angles
409 for (int j = 0; j < nLinks; j++)
410 {
411     // Clamping dtheta[j] so that it becomes less than MAX_ANGLES_CHANGE
412     float l = fabs(dtheta[j]);
413     if (l > MAX_ANGLE_CHANGE)
414     {
415         dtheta[j] *= MAX_ANGLE_CHANGE / l;
416         l = fabs(dtheta[j]);
417     }
418     // Joint angle within [-M_PI, M_PI]
419     jointAngle[j] += dtheta[j];
420
421     if (jointAngle[j] > M_PI) jointAngle[j] -= 2 * float(M_PI);
422     if (jointAngle[j] < -M_PI) jointAngle[j] += 2 * float(M_PI);
423
424     dthetaSum += l;
425 }
426

```

line 407 : 각도들의 변화 합을 구할 변수를 선언한다.

line 412 ~ 416 : 각도변화량을 절대값을 취해준다.

line 424 : dthetaSum에 변화량을 누적 합 해준다.

```

428 // Exercise: Check to see if a solution is found when the ee_goal is out of reach.
429 // Hint: You can consider the changes of the joint angles.
430 // theta의 합이 변화가 없으니, 이는 out of reach...
431 if (dthetaSum < 0.005)
432 {
433     cout << "IL iterations = " << numIterations << endl;
434     return true;
435 }
436 return false;
437 }

```

line 431~ 435 : 변화량 합이 0.005보다 작으면, IK Solve를 멈춘다.



out of reach상태이지만 iteration이 끝나서 출력하는 모습이다.