

컴퓨터 그래픽스를 이용한 사각보행로봇의 기구적 해석

(Kinematic Analysis of the Quadruped Robot Using Computer Graphics)

崔秉旭,** 林俊弘,*** 鄭明振,* 卞增男*

(Byoung Wook Choi, Joon Hong Lim, Myung Jin Chung and Zeungnam Bien)

要 約

보행로봇에 있어서 다리의 기구적 특성은 안정도, 에너지 효율 및 걷는 영역 등에 영향을 준다. 보행로봇 중에서 안정도 여유가 비교적 큰 사각보행로봇은 보폭이 크다. 그리고 수직운동과 수평운동이 서로 분리되어 있는 팬토그래프 구조의 다리를 갖는 보행로봇은 기구적인 복잡성으로 인하여 움직임을 예상하기가 어렵게 된다. 따라서, 보행로봇의 3차원적인 이동을 해석하기 위하여 그래픽스를 이용한 방법이 중요하게 쓰일 수 있다. 본 논문에서는 서로 다른 조인트 구동방식을 갖는 세개의 사각보행로봇을 모델링하고, 각각에 대하여 몸의 중심으로 표시된 경로로써 작업지시를 하여 걷는 모습을 애니메이션 함으로써 그 기구적인 특성을 해석할 수 있게 하였다. 그리고 회전걸음이 나 옆걸음을 위하여 다리 제어를 하였고, 조인트의 이동에 대한 정보를 위하여 Sun-3 시스템에서의 윈도우를 이용하였다.

Abstract

The stability, energetic efficiency and walking volume are affected by the geometric structure of legs of a walking robot. A quadruped walking robot is considered to have large stability margin among the walking robots and pantograph leg permits large walk stroke and mutually independent vertical and horizontal movements, but the kinematic characteristics are difficult to analyze. Graphical method may be useful to characterize three dimensional legged motion of the pantograph mechanism. We present the modelling method for three different quadruped robots with pantograph legs that have different joints mechanism. The modeled robots are animated by a path that is planned with respect to the center of body. In particular, graphical animation incorporates leg control to rotation and side walking and uses the window of Sun-3 system for displaying joint information.

*正會員, **準會員, 韓國科學技術院 電氣 및 電子工學科
(Dept. of Electrical Eng., KAIST)

***正會員, 韓國航空大學 航空電子工學科
(Dept. of Avionics, Hankook Aviation College)

接受日字 : 1988年 5月 24日

I. 서 론

현재의 로보트 연구는 위험한 환경에서의 응용에 많은 노력이 기울여지고 있다. 이러한 사람이 접근할 수 없는 작업환경으로는 방사능 물질을 다룬다든가, 원자로에서의 작업, 그리고 달 탐사 등이 있다. 이

와 같은 면에서 볼 때 보행로봇은 이동 벨트를 이용한 산업용 로봇이나 바퀴를 갖는 로봇에 비해서 장애물을 오르내린다는가, 험한 표면에서 실질적인 다리의 높이를 조절함으로써 유연한 움직임을 할 수 있는 등의 특별한 장점이 있다.^[1]

이러한 다각보행로봇의 연구는 기구적인 반복 운동의 장난감으로부터 시작되어 Hutchinson과 Smith(1963), Shigley(1967), Ohio 주립대학에서의 ASV (adaptive suspension vehicle), Russell(1983)의 ODEX I 등이 있으며,^[1] 최근에 Hirose(1984)에 의한 PV II와 TITAN III라는 X-Y 테이블을 갖는 팬토프래프 구조의 사각보행로봇을 개발하여 층계에서의 실험과 걸을 때의 에너지 손실에 대해서 연구하였다.^[2] 그리고, McGhee와 A. A. Frank(1966)가 Phony Phony라는 사각보행로봇으로서 crawl과 trot 걸음새에 대하여 연구하였으며,^[1] S. M. Song(1987)이 wave 걸음새에서의 안정도 여유에 대한 연구를 하였다.^[3]

그래픽 시뮬레이터에 관한 연구는 로봇 언어와 함께 로봇 공학의 한 부분으로 연구되어 왔다. 로봇 연구에서 그래픽 시뮬레이터는 로봇과 작업 환경을 모델링하여 로봇 언어와 연결시켜서 위험도 등을 줄이면서 로봇 작업을 컴퓨터 그래픽을 이용한 애니메이션을 통하여 시뮬레이션하는 방식으로 연구되어 왔으며, 일부는 상품화되기도 하였다.^[5~9] McDonell Douglas 회사의 PLACE(positioner layout and cell evaluator),^[7] IGRIP(interactive graphics robot instruction program),^[8] ROBSIM(a robot simulation system)^[9] 등이 있으며, 한국과학기술원에서 1985년에 다이내믹 시뮬레이터와 결합된 그래픽 시뮬레이터가 설계되어 로봇의 기술 방법과 애니메이션 방법 등에 대한 연구가 진행되었고, 1986년에 로봇 언어와 연결된 그래픽 시뮬레이터의 개선이 수행되었다.^[5, 6]

지금까지의 그래픽 시뮬레이터는 기준축이 고정되어 있고 end effector가 이동하는 산업용 로봇에 대한 연구였다. 따라서, 기구학적인 기술이 한 방향으로 이루어짐으로써 애니메이션이 비교적 쉽게 이루어진다. 그러나, 본 논문에서의 연구대상인 보행로봇은 기준축이 계속적으로 변화되며, 산업용 로봇보다는 더 많은 조인트를 갖는다. 또한 팬토프래프 구조의 다리를 이용할 경우 기구적 제한이 존재하는데 애니메이션을 위해서는 수학적으로 모델링되어야 하며, 각각의 다리에 작업지시를 하는 방법이 어렵게 된다.

따라서, 사각보행로봇의 개발시 기구적 특성, 걸음새에 따른 다리의 움직임, 안정도 연구 및 작업

지시의 방법과 옆걸음이나 회전운동에 대한 연구가 필요하게 되는데 본 논문에서는 보행로봇의 기구적인 형태 중의 하나인 팬토프래프 형태의 다리를 갖는 사각보행로봇의 그래픽 시뮬레이터를 구성하여 그래픽스를 이용한 애니메이션으로서 보행로봇의 제반연구에 이용하고자 한다. 이와 같은 목적에 비추어 작업환경을 2차원 평면으로 가정하여 몸의 중심을 이용한 작업지시를 하였다. 또한 팬토프래프 구조의 복잡한 운동에 따른 다이내믹스는 고려하지 않으며 기구적인 특성만을 고려하기로 한다.^[12]

II. 사각보행로봇의 모델링

1. 개요

본 논문에서는 세가지 방식의 3차원 동작을 위하여 팬토프래프 구조의 사각보행로봇을 모델링하고, 옆걸음이나 회전운동에 대한 기구적 특성을 해석하는 것을 목적으로 한다. 팬토프래프 형태의 다리를 갖는 사각보행로봇의 보행모습은 기구적인 제한에 따라 달라진다. 일반적으로 x와 z방향의 움직임은 주어진 구조에 따라 유도축을 직선적으로 이동하여 몸체에서 볼 때, 수직 방향이나 전진방향은 발끝에서 상수배 증폭의 움직임으로 나타난다. 그러나, 회전을 하거나 옆걸음을 위한 y축 구동방식은, 첫째 다리구조 전체를 z축을 중심으로 회전하는 방법, 둘째 y축으로 다리 전체를 선형적으로 밖으로 움직이는 방법과, 힌지(hinge)를 이용한 사면증폭의 형태로서 옆 걸음시의 이동도 선형적으로 증폭되는 3차원 팬토프래프 구조 등이 있다.^[1, 2, 4] 위의 방식을 그림 3에 나타내었으며, TTR 구조, TTT 구조 및 TT₂ 구조로 구분한다.

2. 기하학적 모델

하나의 구조물을 나타내기 위하여서는 데이터 구조가 필요하게 된다.^[5, 11] 이러한 데이터 구조는 대상물을 나타내기에 용이하여야 하며, 대상물을 계층적으로 나타내었을 때, 조작 및 계산이 편리하여야 한다.

구조물의 조작은 동차 변환(homogeneous transformation)으로 이루어지며, 대상물에서 각 꼭지점까지는 나무 구조를 갖게 된다.^[11] 따라서, 하나의 구조물에 대한 데이터 구조는 아래와 같이 나타내게 된다.

```
Object  Vertex list number
        Polygon definition
        vertex definition
```

여기서 꼭지점과 면은 다음과 같은 구조를 갖는다

vertex definition

coordinate $V_i(X_i, Y_i, Z_i)$

polygon definition

vertex list $P_i(V_1, V_2, \dots, V_{vtx_no})$

3. 링크 및 보행로봇의 모델링

사각보행로봇의 각 링크는 많은 조인트를 갖는 점에 기인하여, 데이터 양이나 애니메이션 속도의 개선을 위하여 충분한 근사화의 방법인 다각기둥으로 모델링하였다.^(5,11) 각 링크간의 결합관계를 나타내기 위해서는 동차 변환을 이용하게 되는데, 각 좌표계간의 좌표 변환은 행렬곱의 형태로 나타난다.⁽¹¹⁾ 그림 1은 사각기둥으로 모델링된 각 링크의 다리에서의 위치와 좌표계를 나타낸다.

제한된 팬토그래프 구조의 다리를 가진 사각보행로봇을 계층적 구조로 기술하며, 애니메이션 속도나 해석적인 측면을 위하여 처음에 그림 1(a)에서 link 5의 기울기가 45°의 위치각을 지니고 있는 하나의 다리에 대하여 모델링 한다. 모든 다리가 로봇 몸체에 대하여 기구적인 대칭을 갖고 있으므로 모델된

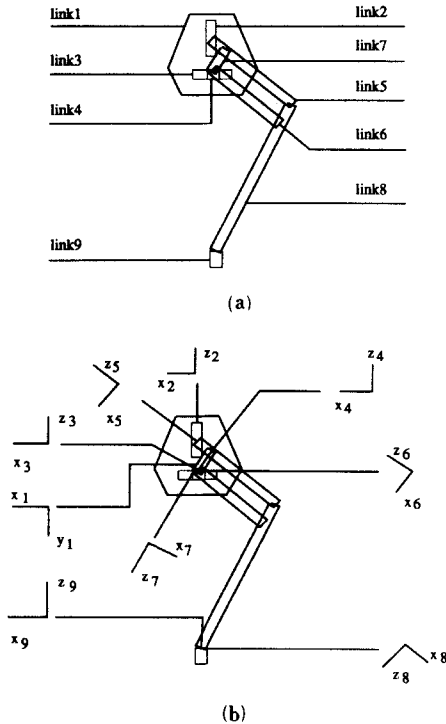


그림 1. (a) 다리에서의 링크의 설정
(b) 다리에서의 링크 좌표계의 설정
Fig. 1. (a) Link configuration of a leg.
(b) Link coordinates of a leg.

하나의 다리로 전체적인 사각보행로봇을 모델링하였다. 그리고, 그림 1(b)에서의 링크의 좌표계는 애니메이션을 쉽게 하기 위한 방법으로 설정하였다. 따라서 link 8의 좌표계는 kinematics에 의하여 계산되는 다리 끝의 운동을 묘사하게 된다.

모든 링크는 계층적 구조를 구성하며, 사각보행로봇의 구조는 이러한 링크들을 하나의 요소로 한다 이와 같은 구조를 갖게함으로써, 처음에는 no_link개의 링크로 몸체와 다리 하나에 대하여 설계를 하고 나머지 다리는 점 대칭과 선 대칭에 의하여 모델링할 수 있다. 따라서, 애니메이션시에는 각 다리에 대한 데이터 구조를 중심으로 계산이 이루어진다. 이러한 사각보행로봇의 데이터 구조는 그림 2와 같은 형태를 갖는다.

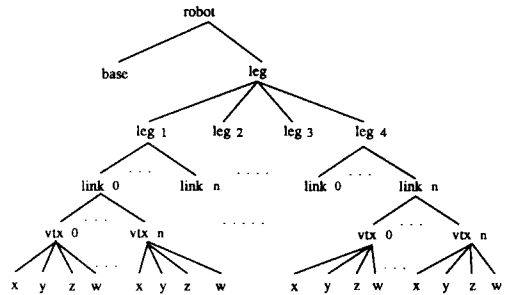


그림 2. 보행로봇의 데이터 구조
Fig. 2. Data structure of a quadruped robot.

III. 사각보행로봇의 해석

1. Kinematics 및 Inverse kinematics

본 논문에서의 대상은 II장에서와 같이 다리에서의 3차원 구동을 위한 세번째 조인트를 기준으로 구분되었다. 따라서, 세가지 방식은 x 및 z 평면에서의 해석은 동일하며, y방향의 동작에 따른 해석만 달라지게 되므로, 먼저 x, z 평면에서 해석을 하고 3차원으로 확장시킨다. 그림 3에서 adbca 루프와 ae-fbca 루프에서 x투영과 z투영을 이용하여 f점에서의 x, z값인 f_x 와 f_z 를 수직운동의 조인트 값인 v_z 와 수평운동의 조인트 값인 h_x 의 함수로 식 (1), (2)와 같이 나타낸다.

$$f_x = \frac{L_{11}}{L_1} h_x = K h_x \tag{1}$$

$$f_z = (1-K)v_z + K h_z \tag{2}$$

여기서 K는 증폭율로써 $K = L_{11}/L_1 = 6$ 으로 나

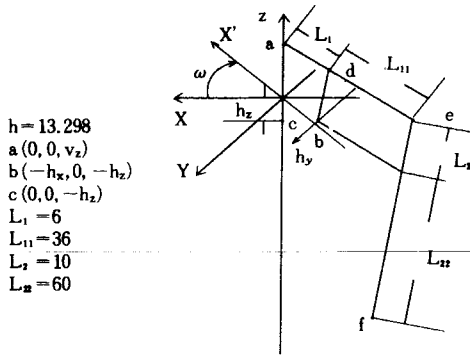


그림 3. 팬토그래프 구조의 기구학
Fig. 3. Kinematics of pantograph mechanism.

타난다. 따라서, x 및 z 방향의 운동이 서로 독립적으로 움직이게 되어 에너지 손실이 적음을 알 수 있다.

그리고 3차원 운동을 위하여 TTT₁ 방식은 y 방향으로 다리전체가 이동하게 되므로 kinematics 를 풀 값은 아래의 식 (3)~(5)와 같다.

$$f_x = K h_x \quad (3)$$

$$f_y = h_y \quad (4)$$

$$f_z = (1 - K) v_z + K h_z \quad (5)$$

TTR 방식은 회전 걸음을 할 때, 제어가 간단해 진다는 장점을 가지고 있으나, 옆걸음을 하는 경우에도 다리전체가 회전해야 하는 단점을 갖는다. 이와 같은 TTR 방식의 3차원 구조의 kinematics 는 식 (6)~(8)과 같이 주어진다.

$$f_x = K h_x \cos(\omega) \quad (6)$$

$$f_y = K h_x \sin(\omega) \quad (7)$$

$$f_z = (1 - K) v_z + K h_z \quad (5)$$

TTT₂ 방식은 3차원 팬토그래프 구조로서 x, y 및 z의 운동이 서로 분리되며, y 방향의 움직임도 조인트값이 다리끝에서 상수배로 증폭이 일어난다. 따라서, 에너지 손실이 적으며 회전운동이나 옆걸음의 이동이 쉽게 이루어지며, 결과는 식 (9)~(11)과 같다.

$$f_x = K h_x \quad (9)$$

$$f_y = K h_y \quad (10)$$

$$f_z = (1 - K) v_z + K h_z \quad (11)$$

기준 좌표계에서 몸의 중심점으로 표시된 경로가 주어져 있을 때, 다른 위치로 가기 위해서는 inverse

kinematics 를 풀어서 조인트 값을 구하여야 한다.^{(1),(10)} 팬토그래프 구조의 다리는 kinematics 가 선형적이어서, inverse kinematics 가 쉽게 이루어진다. TTT₂ 방식의 해만 나타내면 식 (12)~(14)로 주어진다.

$$h_x = \frac{f_x}{K} \quad (12)$$

$$h_y = \frac{f_y}{K} \quad (13)$$

$$v_z = \frac{f_z - K h_z}{1 - K} \quad (14)$$

2. 경로 계획 및 걸음세 제어

보행로봇의 작업지시는 길을 계획하여 그 길을 조건에 맞는 걸음새로 걸게하는 것을 말한다. 길을 2차원 평면으로 가정하였기 때문에 경로는 몸의 중심이 이동하는 2차원 좌표의 연속으로 나타낼 수 있다. 따라서 기준좌표계에서 몸의 중심이 이동하는 2차원 좌표가 된다. 이러한 경로가 직선이 아닌 경우 다리의 보행영역에 의하여 회전 각도가 제한되므로 회전운동을 하여 걸어가고자 할 때 경로를 완만히 할 수 있게 Bezier curve fitting 을 이용하였다.⁽¹¹⁾

본 논문에서는 한번에 하나의 다리가 움직이는 crawl 걸음새를 기본으로 한다. 경로는 기준 좌표계에서의 좌표이므로 몸중심으로의 좌표 변환이 이루어져서 inverse kinematics 를 풀어야 하는데, 좌표 간의 관계는 그림 4와 같으며, p₁, p₂, p₃는 이동하여야 할 좌표들로서, (x₂, y₂, z₂) 좌표계에서 p₃의 위치에 대한 inverse kinematics 를 풀게된다. 이러한 몸중심 좌표계에서 기준 좌표계로의 변환은 T₂=T₁A₂로 되어 기준 좌표계에 대한 보행로봇의 위치를 알게 된다.

몸의 중심 좌표계에서 다음 위치로 이동을 위한 방향은 그림 5와 같이 구분된다. 이러한 각 방향으로의 이동은 TTT₁, TTT₂ 방법에서 다리가 몸 안으로 들어갈 수 없으므로 표 1의 순서로 이동하게 된다. 옆걸음은 이동시 몸의 중심에 대하여 각 다리의 보폭은 동일한 형태로 나타난다. 따라서, 몸의 위치가 이동할 때 방향은 변하지 아니하고, 다만 각 다리가 오른쪽 왼쪽에 따라 움직이는 순서만 변화될 뿐이다.

그러나, 회전 운동을 할 경우 다리의 이동 순서는 방향에 따라 결정될 것이며, 각 다리의 움직임이 몸체의 중심에 대하여 같은 반경을 유지하여야 한다. 따라서, 거리와 회전각으로 네 다리의 움직임을 묘사해야 하는데, 각 다리의 조인트 값을 구하는 방법은 사각보행로봇이 강체를 이룬다는 점에 착안하

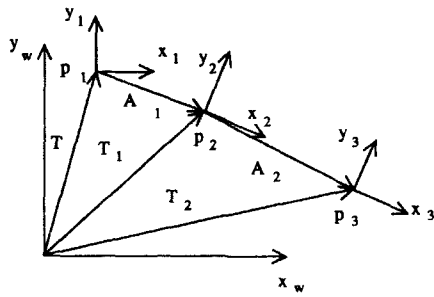


그림 4. 경로의 좌표변환
Fig. 4. Coordinate transformation of path.

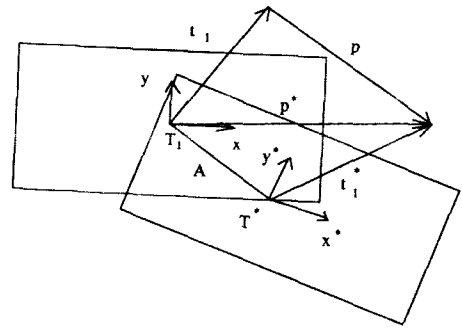


그림 6. 회전이동시의 다리의 이동
Fig. 6. Leg movement on rotating walk.

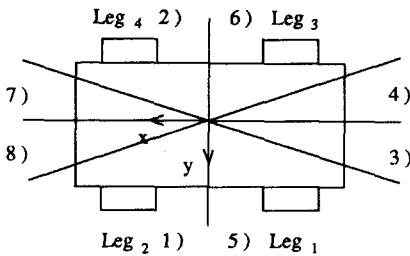


그림 5. 이동방향의 구분
Fig. 5. Partition of path direction.

표 1. 다리의 이동 순서
Table 1. The sequence of leg movement.

영역	다 리 번 호					
1	0	1	2	0	3	4
2	0	3	4	0	1	2
3	0	1	4	0	3	2
4	0	3	2	0	1	4
5	0	2	1	0	4	3
6	0	4	3	0	2	1
7	0	4	1	0	2	3
8	0	2	3	0	4	1

여 각 다리의 궤적을 구할 수 있게 된다. 그림 6에서 다리의 다음 위치는 p로 식 (16)과 같으며, 나머지 다리들도 몸의 중심에서 다리의 위치를 나타내는 고유한 벡터에 의하여 보폭을 결정지을 수 있다.

$$p^* = A t_2 \quad (15)$$

$$p = p^* - t_1 = A t_2 - t_1 \quad (16)$$

3. 애니메이션을 위한 좌표변환

팬토프래프 구조는 기구학적 제한이 존재하고 있다. 이러한 제한은 제어를 하고자 할 때 각 다리의 끝점의 움직임이 기구적인 제한으로 오히려 쉽게 나타난다는 장점을 갖는다. 그러나 그래픽을 통한 애니메이션을 위해서는 팬토프래프를 구성하는 네개의 링크가 각각 분리된 조인트와 같게 된다. 따라서 이러한 기구적 제한에 의한 조인트는 끝점에서의 움직임이 조인트 값에 선형적인데 반해서 비선형성을 갖게 된다. 이러한 기구학적 제한을 수학적으로 모델링할 때, 모든 다리의 링크의 움직임이 3개의 조인트 값으로 명시되어야 한다.

로봇트는 구조에 따라 자유도가 결정되며, 하나의 다리는 x, y 및 z로의 3개의 조인트를 갖게 되는데 그림 7의 2차원 팬토프래프 구조에 대한 수학적 모델링을 3차원 구조로 확장시키면 된다.

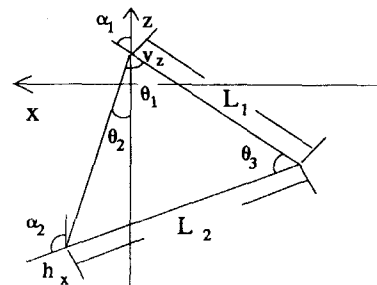


그림 7. 기구적 제한의 수학적 모델링
Fig. 7. Mathematical modelling of mechanical constraints.

식 (17)~(22)은 기구적 제한의 해석으로 α_1 과 α_2 가 2개의 조인트인 h_x 와 v_z 로 표시됨으로 2차원 팬토프래프 구조의 각 링크의 움직임을 명시할 수 있게

된다.

$$\theta_1 = \cos^{-1} \frac{L_1^2 + L_2^2 - L^2}{2LL_1} \quad (17)$$

$$\theta_2 = \tan^{-1} \frac{h_x}{v_z - h_z} \quad (18)$$

$$\theta_3 = \cos^{-1} \frac{L_1^2 + L_2^2 - L^2}{2L_1 L_2} \quad (19)$$

$$\alpha_1 = \theta_1 - \theta_2 \quad (20)$$

$$\alpha_2 = \theta_1 + \theta_3 - \theta_2 \quad (21)$$

여기서 L 은 다음과 같다.

$$L = [(v_z - h_z)^2 + h_x^2]^{\frac{1}{2}} \quad (22)$$

따라서 각각의 링크에 대한 변화는 수직 운동인 v_z 와 수평 운동인 h_x 로 유일하게 규정되어지며, 펜토타그래프구조의 평행사변형 움직임에 대하여 그림 3의 링크 좌표계를 갖는 각 링크는 아래의 표 2와 같은 특성을 갖는다.

표 2. 링크 좌표 변환의 특성

Table 2. The characteristics of link coordinate transformation.

링크	좌 표 변 환		
5	R_{y_4}	T_{z_4}	
6	R_{y_4}	T_{x_5}	
7	R_{y_6}	T_{x_5}	
8	R_{y_6}	T_{x_7}	T_{y_7}
9	T_{x_7}	T_{y_7}	

위의 표에서 링크 6에 대하여 설명하면 R_{y_4} 는 y축에 대한 회전으로 링크 4와 같은 값이며, T_{x_5} 는 x축으로의 이동이며, 링크 5의 x축 이동과 같게 된다.

TTT₁, TTT₂ 및 TTR 방식에 대한 3차원 운동의 수학적 모델링에서 TTT₁ 방식은 각 링크가 y축으로 조인트 값만큼 이동한 것과 같다. 그리고, TTR 방식은 v_z 만큼 이동한 link5 좌표계에 대하여 $R_z(\omega)$ 한 것과 같다. 따라서 표 3과 같은 3차원 특성을 갖게 된다.

그리고 위의 다리의 각 링크의 좌표계에 대한 계산은 다리의 좌표계에서 행하게 됨으로 다리마다의 차이가 존재한다. TTT₂ 방식은 힌지를 이용함으로 x축을 중심으로 y 방향의 회전각 α_3 는 다음과 같다.

$$\alpha_3 = \tan^{-1} \frac{h_y}{v_z - h_z} \quad (23)$$

표 3. 3차원 구동에 따른 다리의 좌표 변환
Table 3. Leg coordinate transformation of 3 dimensional animation.

구 동 방 식	좌 표 변 환
TTT ₁	T_y
TTT ₂	R_x
TTR	R_z

따라서 세개의 조인트 값에 따라 각 링크의 좌표 변환식을 얻는다. 각각의 기준좌표계는 다리 기준좌표계이고, TTT₁, TTT₂ 구동 방법의 각 링크의 움직이는 좌표 변환 관계는 표 4와 표 5와 같다. 그리고, TTR 방식은 다리 전체를 $R_z(\omega)$ 한 것과 같은데 이의 결과는 그림 8과 같은 변환에 의하여 이루어진다.

표 4. TTT₁ 방식의 애니메이션 좌표 변환

Table 4. Coordinate transformation of TTT₁ for animation.

링크	좌 표 변 환			
1	$R_x(-90^\circ)$	$T_x(-37)$	$T_y(37)$	$T_z(-18)$
2	$T_x(-37)$	$T_y(38.5)$	$T_z(-2)$	
3	$T_x(-37)$	$T_y(38.5)$	$T_z(-8+h_z)$	
4	$T_x(-37)$	$T_y(41.5)$	$T_z(-8+h_z)$	
5	$T_z(v_z)$	$T_y(h_y)$	$R_y(\alpha_1)$	
6	$T_x(h_x)$	$T_y(h_y)$	$T_z(h_z+1.5)$	$R_y(\alpha_1)$
7	$T_x(h_x)$	$T_y(h_y)$	$T_z(h_z+1.5)$	$R_y(\alpha_2)$
8	$T_x(Kh_x)$	$T_y(h_y)$	$T_z((1-K)v_z+Kh_z)$	$R_y(\alpha_2)$
9	$T_x(Kh_x)$	$T_y(h_y)$	$T_z((1-K)v_z+Kh_z)$	

표 5. TTT₂ 방식의 애니메이션 좌표 변환

Table 5. Coordinate transformation of TTT₂ for animation.

링크	좌 표 변 환			
1	$R_x(-90^\circ)$	$T_x(-37)$	$T_y(37)$	$T_z(-18)$
2	$T_x(-37)$	$T_y(38.5)$	$T_z(-2)$	
3	$T_x(-37)$	$T_y(38.5)$	$T_z(-8+h_z)$	
4	$T_x(-37)$	$T_y(41.5)$	$T_z(-8+h_z)$	
5	$T_z(v_z)$	$R_x(\alpha_3)$	$R_y(\alpha_1)$	
6	$T_x(h_x)$	$T_y(h_y)$	$T_z(h_z+1.5)$	$R_x(\alpha_3) R_y(\alpha_1)$
7	$T_x(h_x)$	$T_y(h_y)$	$T_z(h_z+1.5)$	$R_x(\alpha_3) R_y(\alpha_2)$
8	$T_x(Kh_x)$	$T_y(Kh_y)$	$T_z((1-K)v_z+Kh_z)$	$R_x(\alpha_3) R_y(\alpha_2)$
9	$T_x(Kh_x)$	$T_y(Kh_y)$	$T_z((1-K)v_z+Kh_z)$	

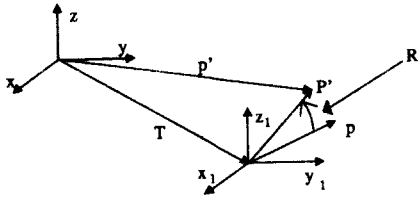


그림 8. TTR 방식의 애니메이션 좌표변환
 Fig. 8. Coordinate transformation of TTR type for animation.

IV. 시스템의 구성 및 수행결과

1. 구성

보행로봇의 시뮬레이션의 목적은 주어진 길을 걷고 있는 모습을 통하여 더많은 기구학적 정보를 얻어야 한다. 따라서 그래픽 시뮬레이터의 기능을 증대시키기 위하여 현실감을 주는 2차원 투영 기술이 요구되며, 기준 좌표계에서의 출력 프리미티브가 화면에 나타나는 관계를 결정지우는 수학적 기술이 필요하게 된다. 그리고, 각다리의 움직임이 기구적 특성을 나타내므로 보행로봇의 애니메이션을 위한 기구적 제한의 수학적 모델링이 이루어져야 한다.

전체적인 하드웨어 구성은 그림 9와 같다. Sun-3 110 LC는 68881 floating point coprocessor를 갖고 있으며, 주 CPU는 68020으로 32 bit machine의 구조를 가지고 있다. 그래픽 모니터는 1152×900의 해상도를 갖으며 RGB가 0-255의 조합으로 color lookup table을 구성하게 되어 있는데, 16가지의 색으로 구현하였다.

그래픽 시뮬레이션을 위한 소프트웨어 구조로는 그림 10과 같이 구분 지을 수 있다. 그림 11은 main menu로서 mouse인 pick device를 이용하여 segment에 주어진 pick_id를 선택하게 된다. 로봇 modeller는 각각의 링크의 꼭지점 값들을 사각보행로봇의

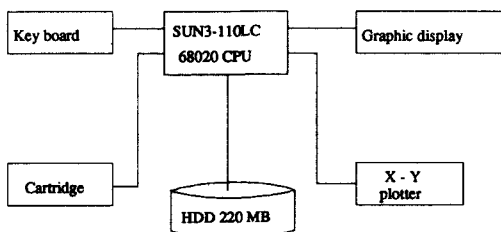


그림 9. 전체 하드웨어 구성
 Fig. 9. Hardware structure of total system.

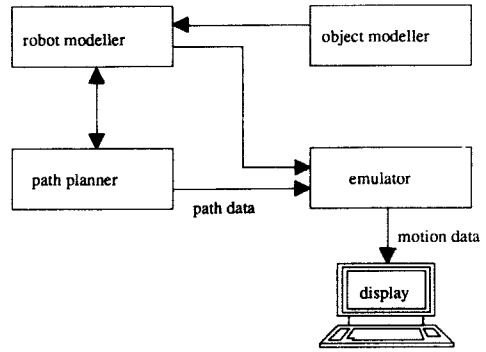


그림 10. 전체 소프트웨어 구성
 Fig. 10. Software structure of total system.

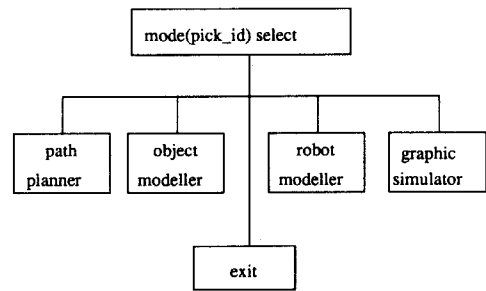


그림 11. Main menu의 구성
 Fig. 11. Main menu description.

데이터 구조로 구성하는 것을 말한다. 이러한 모델된 보행로봇에 작업 지시의 방법으로 경로를 계획하게 되며, emulator에서는 이러한 경로에 따라 세가지의 3차원 운동 구조를 갖는 사각보행로봇에 맞는 inverse kinematics를 풀어서 조인트 값을 얻게 되는데, 이러한 조인트 값으로 각 다리의 구성 링크의 좌표계를 계산하여 애니메이션 데이터를 만든다. 위와 같은 작업을 계속적인 경로에 대해서 행하게 되며, emulator에서 주어진 경로를 가기 위한 조인트 값과 다리의 순서를 정하여 준다. 따라서 이러한 조인트 값을 이용하여 모델링된 로봇의 움직임이 그래픽 프리미티브를 통하여 그래픽 화면에 나타나게 된다.

2. 그래픽 기능

그래픽 프리미티브로는 ACM SIGGRAPH core system에 기초하여 level 3C(dynamic output with 3D scaling, rotation and translation)의 출력 프리미티브와 level 2(complete input)의 입력 프리미

티브로된 Sun core를 이용하였다. Sun core에서 3차원 동작은 view surface에서 그림의 일부를 나타내는 segment 단위로 나타내게 된다.^{13,14)}

출력을 위하여 몸체와 다리 네 개가 이름을 갖는 다섯 개의 retained segment로 구성하였다. 따라서 다리 하나가 움직일 때 다른 segment는 계산을 하지 않으며 움직이는 segment에 대해서만 3차원 조작이 이루어진다. 이렇게 함으로써 다리 하나가 이동할 경계 나머지 다리는 고정되어 사각보행로봇의 걷는 모습을 좀더 현실감있게 나타낼 수 있었다. 또한 사각 보행로봇의 현실감있는 표현을 위하여 원근 투영을 하였고, 각 링크에 대한 숨은선 제거를 통하여 기구적 특성의 관찰을 증대시켰다.^{5,11)}

사용자의 편의를 위하여 sun core의 입력 프리미티브 중에서 논리적 입력 소자로 locator, pick 및 button을 mouse로 구현하였다. 또한 키보드로서 string input을 받아 들이게하여 응용 프로그램에서 변수 변환에 사용되어진다. Locator는 NDC(normalized device coordinate) 좌표계에서의 위치를 나타내며 dingbat으로 echo등의 위치를 나타내거나 rubber band로서 경로 계획에서 로봇 중심의 이동점들의 연속된 선을 그리고자 할 때 이용된다. 그리고, pick device는 segment나 segment내의 프리미티브를 선택할 때 쓰여지며 pointing finger로서 위치를 나타낸다. 그리고 키보드 입력을 위한 시뮬레이터의 포시는 "Enter"라는 말로 시작하게 하였다.^{13,14)}

또한 다음과 같은 필요에 의하여 윈도우를 변화시켜서 viewport로의 변환을 통하여 화면에 그려지는 기준 좌표계의 위치를 변화시킬 필요가 있다.

- 1) 한 부분을 zooming 시켜서 관찰할 필요성
- 2) 보행로봇이 윈도우 밖으로 나갔을 경우의 모습을 보고자 할 경우

따라서, 위와 같은 목적을 위하여 윈도우를 조절하는데 mouse를 locator로 이용하여 mouse에 의하여 NDC에서의 좌표를 얻는다. 이러한 NDC좌표를 이용하여 윈도우에서의 좌표를 구할 수 있으므로 locator 위치에 맞게 윈도우 조절을 하였다. 따라서 마우스 버튼의 숫자와 NDC에서의 위치에 따라 윈도우가 변환되어 사각보행로봇을 시뮬레이션할 때 좀더 자세히 관찰할 수 있다.

그리고, 대상물 설계시에 두 개의 윈도우를 만들어서 데이터 조작과 그래픽화면을 분리시켜서 나타내었다. 또한 경로 계획을 할 때, 화일을 받는 경우와 마우스를 locator로 이용할 수 있게 하였으며, 시뮬레이션할 때 보행로봇의 위치를 2차원 평면으로 투영하여 polymarker로써 나타내었다.

3. 수행 결과

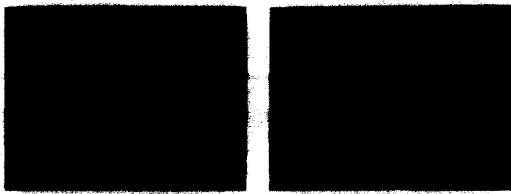
본 연구에서의 전체적인 응용 프로그램은 "C" 언어로 구현하였으며, 출력 프리미티브가 DYNAMICC이고, SYNCHRONOUS INPUT까지 제공되는 Sun core를 그래픽 프리미티브로서 이용하였다.^{13,14)}

시뮬레이션시의 애니메이션 속도의 증가를 위하여 retained segment를 이용하여 다리의 이동을 묘사하였으며, 현실감을 위하여 retained_segment_visibility를 이용함으로써 화면을 지우지 않고 그림을 변화시켰다. 한 보폭에 대하여 몸의 이동이 한번 일어나며, 다리의 움직임은 x, y 및 z로 한번의 인터폴레이션을 하게되어 모두 8번 일어난다. 이러한 하나의 보폭에 대하여 약 0.75초가 소요되며, 원근법이 나 숨은선 제거를 고려한 경우도 거의 같은 속도를 갖는다.

그림 12는 TTT₂ 방식의 로봇에 대해서 이동할 곳이 (515.8, 480.0)에서 (578.6 469.1)으로 변화할 경우 가장 현실감을 높일 수 있는 semi-hidden perspective mode에서 실험한 결과이다. 여기서 몸의 기준 좌표계에서 본 기울기는 -9.858° 이며, 관찰자의 위치를 (1000, 2000, 400)으로 하였을 경우로 데이터에 대한 scale 값은 2이다. 이와 같은 몸의 중심에서의 회전각이 $-y$ 방향이므로 걷는 다리의 순서가 달라지는데, 이 경우는 1,4,3,2의 순서로 이동하여 회전시 다리와 몸체의 간섭을 피하게 되며, 회전 이동이므로 각각의 다리의 inverse kinematics 값이 서로 다르며 다리의 이동에 따른 설명을 그림 아래에 나타내었는데, 다리의 번호와 그때의 조인트 값으로서 h_x 는 전진 방향의 이동이며, h_y 는 허프에서 옆방향으로의 이동값을 나타낸다. 그리고, v_z 는 z축으로 다리가 들어지는 값을 나타내는데, -1.0 으로서 조인트 인터폴레이션을 하였다.

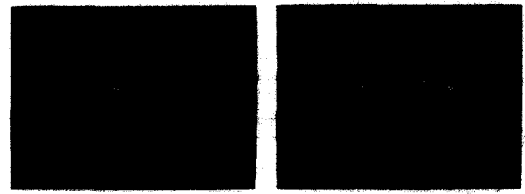
그림 13은 TTT₁ 방식인데 기구의 제한에 의하여 옆걸음을 하게 된다. 이 경우에 회전 방향의 변화는 존재하지 않으며, 한번의 보폭 이동에서 각각의 다리에서의 조인트 값은 같게 된다. 그림에서 보듯이 이동을 위하여 팬토그래프 구조의 기구외에도 다리의 보조 기구까지 같이 h_y 값으로 이동하므로 에너지 손실이 생기게 된다. 또한 다리와 몸체의 간섭에 의하여 회전이동이 어렵게 된다.

그림 14는 TTR 방식의 이동에 대한 결과이며, ω 는 다리 전체가 z축을 중심으로 회전하는 양을 나타낸다. 그림에서 다리의 회전으로 몸체의 설계가 어렵게 됨을 알 수 있다. 또한 옆걸음의 경우도 다리 전체를 회전하여야 하며 움직임이 다리의 형태에 의한 불규칙성을 가져오게되어 정적 안정도에도 문제



(a) Body: $h_x=0.00$
 $h_y=0.00$ $v_z=0.00$

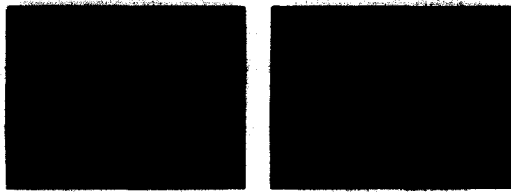
(b) Leg 1: $h_x=-2.34$
 $h_y=-0.17$ $v_z=-1.00$



(a) Leg 1: $h_x=-2.18$
 $h_y=21.83$ $v_z=0.00$

(b) Leg 4: $h_x=1.96$
 $h_y=-24.54$ $v_z=0.00$

그림 13. TTT₁ 방식의 시뮬레이션 결과
Fig. 13. Simulation results of TTT₁ type.



(c) Leg 1: $h_x=-4.69$
 $h_y=-0.33$ $v_z=0.00$

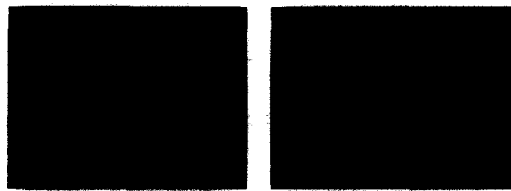
(d) Leg 4: $h_x=-2.89$
 $h_y=-0.74$ $v_z=-1.00$



(a) Leg 2: $h_x=-2.84$
 $v_z=-1.00$ $\omega=-5.83$

(b) Leg 4: $h_x=-5.82$
 $v_z=0.00$ $\omega=18.76$

그림 14. TTR 방식의 시뮬레이션 결과
Fig. 14. Simulation results of TTR type.



(e) Leg 4: $h_x=-5.78$
 $h_y=-1.49$ $v_z=0.00$

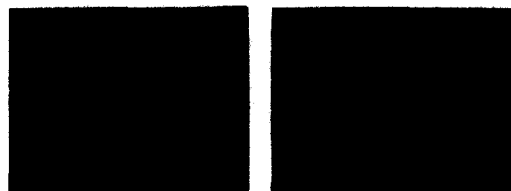
(f) Leg 3: $h_x=5.87$ $h_y=0.43$ $v_z=0.00$
Leg 2: $h_x=4.60$ $h_y=1.39$ $v_z=0.00$

가 발생함을 알 수 있게 된다.

V. 결 론

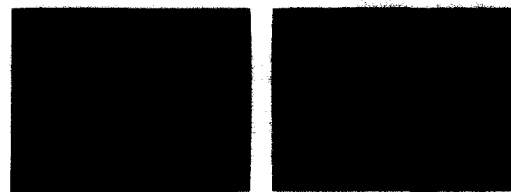
본 논문에서는 지금까지의 로봇틱스에 있어서의 그래픽 시뮬레이터의 대상이었던 산업용 로봇과는 다른 펜토그래프 구조의 다리를 갖는 사각보행로봇으로서 3차원 운동을 갖는 세가지 방법의 로봇트에 대하여 모델링하였다. 그리고, 이러한 기구적 제한을 수학적으로 나타내어 애니메이션 시키는 방법에 대하여 연구하였다. 또한, 세가지 방식의 보행로봇에 대하여 2차원 평면을 대상으로 견제하여, 그 움직임을 애니메이션 시킴으로써 기구학적 특성을 관찰해 보았다. 그리고, 보행로봇의 개발에 필요한 기구적 특성과 걷는 방법에 대한 다리 제어의 고려사항 등을 기초적으로 제시하였다. 따라서, 보행로봇에 관련된 제반 알고리즘의 개발에 응용될 수 있을 것이다.

앞으로 작업 환경을 모델링하고 다리의 제어에 대한 연구 및 다이나믹스에 대한 연구가 진행되어 본 연구에 추가 된다면 사각보행로봇의 해석 기능이



(g) Leg 3: $h_x=2.94$
 $h_y=0.22$ $v_z=-1.00$

(h) Leg 3: $h_x=0.00$
 $h_y=0.00$ $v_z=0.00$



(i) Leg 2: $h_x=2.30$
 $h_y=0.69$ $v_z=-1.00$

(k) Leg 2: $h_x=0.00$
 $h_y=0.00$ $v_z=0.00$

그림 12. TTT₂ 방식의 시뮬레이션 결과
Fig. 12. Simulation results of TTT₂ type.

더욱 강화될 것이다.

參 考 文 獻

- [1] D.J. Todd, *Walking Machines-A Introduction to Legged Robots*, Anchor Press, 1985.
- [2] S. Hirose, "A Study on Design and Control of a Quadruped Walking Vehicle," *Int. J. of Robotics Research*, vol. 3, no. 2, pp. 113-133, 1984.
- [3] 박성혁, 황승구, "다각보행로봇의 제어 방법에 관한 연구," '87한국자동제어학술회의, vol. 1, pp. 69-73, 1987.
- [4] S-M. Song and K.J. Waldron, "An Analytical Approach for Gait Study and Its Applications on Wave Gaits," *Int. J. of Robotics Research*, vol. 6, no. 2, pp. 60-71, 1987.
- [5] 장원, "컴퓨터 그래픽스를 이용한 로봇 시뮬레이터의 설계," 한국과학기술원, 석사학위논문, 1986.
- [6] 김형수, "로봇트 교시를 위한 오프라인 프로그래밍 시스템의 설계," 한국과학기술원, 석사학위논문, 1987.
- [7] D.L. Henderson, "Off-line Programing Cuts Costs, Saves Time," *Robotics Today*, pp. 23-24, Aug, 1986.
- [8] J.P. Harrison and R. Marahan, "The IGRIP Approach to Off-line Programming and Workcell Design," *Robotics Today*, pp. 25-26, Aug, 1986.
- [9] D.W. Wloka, "ROBSIM-A Robot Simulation System," *Proc. IEEE Int. Conf. on Robotics and Automation*, vol. 3, pp. 1859-1864, 1986.
- [10] C.S.G. Lee, R.C. Gonzalez, and K.S. Fu, *Tutorial on Robotics*, *IEEE Computer Society*, 1984.
- [11] J.D. Foley and A. Van Dam, *Fundamentals of Interactive Computer Graphics*, Addison-Wesley, 1982.
- [12] M. Vukobratovic and V. Potkonjak, *Scientific Fundamentals of Robotics-Applied Dynamics and CAD of Manipulation Robots*, vol. 6, Springer-Verlag, Berlin, Heidelberg, 1985.
- [13] *Sun Core Reference Manual*, Sun micro-systems, 1986.
- [14] *Sunview Programmer's Guide*, Sun micro-systems, 1986. *

著 者 紹 介

崔 秉 旭 (準會員)

1963年 2月 13日生. 1982年 ~ 1986年 2月 한국항공대학 항공전자공학과 공학사 취득. 1986年 ~ 1988年 2月 한국과학기술원 전기 및 전자공학과 공학석사학위 취득. 현재 한국과학기술원 전기 및 전자공학과 박사과정 재학중. 주관심분야는 로보틱스 등임.

자공학과 박사과정 재학중. 주관심분야는 로보틱스 등임.

林 俊 弘 (正會員)

第25卷 第4號 參照.

현재 한국항공대학 항공전자공학과 조교수. 주관심분야는 로보틱스 등임.

卞 墻 男 (正會員) 第25卷 第10號 參照

현재 한국과학기술원 전기전자공학과 교수

鄭 明 振 (正會員) 第25卷 第8號 參照

현재 한국과학기술원 전기 및 전자공학과 조교수.