

그래픽 시뮬레이터와 물리엔진을 이용한 효과적인 4족 보행로봇의 모션티칭 방법

The Efficient Motion Teaching Method of Quadruped Robot Using Graphic Simulator and Physics Engine

류지형, 김지홍, *이창구, **이수영
Jihyoung Ryu, Kim jeehong, Changoo Lee, Sooyeong Yi

Abstract - A graphic simulator is efficient to see what will happen to the target robot. But it is not exactly same as the real world. Because there are so many physical laws to be concerned. In this paper, we propose a simulator with physics engine to create motions for quadruped robot. It is not only to show more real simulations but also to be more efficient for teaching motions to quadruped robot. To solve the quadruped robot's dynamics or inverse kinematics, It takes much times and hard effort. Using physics engine make it easy to setup motions without calculating inverse kinematics or dynamics.

Key Words : simulation, physics engine, quadruped robot

1. 서론

최근 들어 가정용 로봇이 각광을 받으면서 로봇은 새로운 분야를 개척해 나가고 있다. 이전의 산업용 로봇에 비해 가정에서 쓰이는 로봇은 가사를 대신하는 청소기 로봇에서부터 인간과 교감하고 정서적 안정을 줄 수 있는 애완용 로봇에 이르기까지 그 역할이 다양해 졌다. 이러한 애완용 로봇은 주로 인간의 명령과 행동에 다양한 동작을 취함으로써 인간의 감성과 교류한다. 따라서 애완용 로봇이 다양한 동작을 취하고 이를 활용하여 인간과 의사소통을 하기 위해서는 효과적인 지시방법이 필요하다.

시뮬레이션은 실제 시스템이 동작하기 전에 시스템의 안정성이나 동작 상태를 현실과 비슷하게 알 수 있는 이점을 가지고 있다. 또한, 그래픽 시뮬레이터[1]는 그 동작을 단지 수치적으로나 그래프가 아닌 실제 시스템의 동작을 사용자가 직관적으로 알 수 있도록 도움을 준다.

시뮬레이터를 제작함에 있어서는 시스템의 특성은 물론 여러 물리적인 현상을 수학적으로 계산하여 적용하여야 한다. 따라서 이러한 물리적 현상을 충분히 고려하지 못한다면 시뮬레이션은 실제와 많이 다를 수 있다.

본 논문에서는 앞서 설명한 가정용 애완 로봇 중 주류를 이루는 강아지 형태의 4족 보행 로봇의 그래픽 시뮬레이터에 물리엔진을 더하여 이를 활용한 모션티칭 방법을 소개하고 있다. 제안한 그래픽 시뮬레이터는 물리엔진을 탑재하고 타겟인 4족 보행로봇의 기구학을 이용하여 각 다리의 관절 값에 따라서 자세를 취하고 이를 물리적으로 다시 계산하여 이

러한 자세가 안정한지 또는, 불안정하여 넘어질지를 판단 할 수 있다. 또한, 별도의 역기구학이나 동역학의 계산 없이 로봇의 모션을 자유롭게 조정할 수 있다는 장점도 있다.

따라서 2장에서는 시뮬레이션에 쓰일 로봇의 기구학을 소개하고 3장에서는 이를 활용한 시뮬레이터와 물리엔진을 이용하여 로봇에게 동작을 지시하는 방법을 소개한다. 4장에서 결론을 내려한다.

2. 4족 보행로봇의 기구학과 역기구학

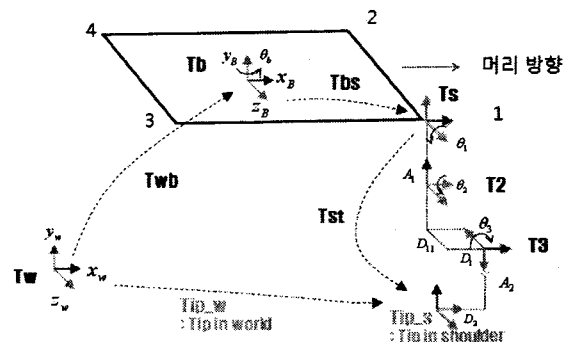


그림 1. 좌표계 설정 및 좌표변환

로봇의 기구학은 로봇의 관절 값들을 제어하여 사용자가 원하는 자세를 만드는데 도움을 준다. 따라서 각각의 로봇에는 각각의 기구학이 존재한다. 이러한 기구학은 대부분 회전 행렬과 이동행렬의 곱으로 표현된다[2].

우선 기구학을 구하기 위하여 로봇의 좌표계를 그림 1과 같이 설정하고 이를 이용하여 로봇의 앞오른쪽 다리에 관하여 기구학을 풀었다. 우선 Homogeneous Transformation Matrix는 식 (1)과 같다.

$$T'_s = T_s^2 \cdot T_2^3 \cdot T_3^4 \quad (1)$$

설정된 좌표계에서 어깨와 링크2 사이의 좌표변환은 식

저자 소개

류 지 형 : 전북대학교 전자정보공학부 박사과정

김 지 홍 : 전북대학교 제어계측공학과 박사수료

* 이 창 구 : 전북대학교 전자정보공학부 교수

** 이 수 영 : 서울산업대학교 전기공학과 부교수

(2)와같이 이루어진다.

$$T_2^2 = \begin{bmatrix} -S_1 & 0 & C_1 & 0 \\ C_1 & 0 & S_1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2)$$

또한, 링크2와 링크3인 발목 사이의 좌표 변환은 식 (3)과 같이 이루어진다.

$$T_2^3 = \begin{bmatrix} 0 & -C_2 & S_2 & -A_1C_2 - D_{11}S_2 \\ 0 & -S_2 & -C_2 & -A_1S_2 - D_{11}C_2 \\ 1 & 0 & 0 & D_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3)$$

마지막으로 발목과 발바닥인 Tip사이의 좌표 변환은 식(4)와같이 이루어진다.

$$T_3^t = \begin{bmatrix} S_3 & 0 & C_3 & -D_2C_3 - A_2S_3 \\ -C_3 & 0 & S_3 & -D_2S_3 + A_2C_3 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4)$$

이를 종합하여보면 최종적으로는 식(5)의 Homogeneous Transformation Matrix를 얻을 수 있다. 여기에 쓰인 기호들은 식 (6), (7)에 기술 되어 있다.

$$T_s^t = \begin{bmatrix} C_1S_3 - S_1C_2C_3 & S_1S_2 & C_1C_3 + S_1C_2C_3 & P_x \\ S_1S_3 + C_1C_2C_3 & -C_1S_2 & S_1C_3 - C_1C_2S_3 & P_y \\ S_2C_3 & C_2 & -S_2S_3 & P_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5)$$

$$\begin{aligned} S_1 &= \sin \theta_1 & C_1 &= \cos \theta_1 \\ S_2 &= \sin \theta_2 & C_2 &= \cos \theta_2 \\ S_3 &= \sin \theta_3 & C_3 &= \cos \theta_3 \end{aligned} \quad (6)$$

$$\begin{aligned} P_x &= -C_1 \cdot (D_2C_3 + A_2S_3) - S_1C_2 \cdot (D_2C_3 - A_2C_3) + S_1 \cdot (A_1C_2 + D_{11}S_2) + D_1C_1 \\ P_y &= -S_1 \cdot (D_2C_3 + A_2S_3) + C_1C_2 \cdot (D_2C_3 - A_2C_3) - C_1 \cdot (A_1C_2 + D_{11}S_2) + D_1S_1 \\ P_z &= S_2 \cdot (D_2S_3 - A_2C_3) - A_4S_2 + D_{11}C_2 \\ P_x &= S_1 \cdot \{A_1C_2 + D_{11}S_2 - C_2 \cdot (D_2S_3 - A_2C_3)\} + C_1 \cdot \{D_1 - D_2C_3 - A_2S_3\} \\ P_y &= -C_1 \cdot \{A_1C_2 + D_{11}S_2 - C_2 \cdot (D_2S_3 - A_2C_3)\} + S_1 \cdot \{D_1 - D_2C_3 - A_2S_3\} \\ P_z &= S_2 \cdot (D_2S_3 - A_2C_3 - A_4) + D_{11}C_2 \end{aligned} \quad (7)$$

이렇게 구한 기구학을 기반으로 기하학적인 방법으로는 역기구학을 구하기 힘들었다. 따라서 대수적인 방법으로 역기구학을 구하면 $\theta_1, \theta_2, \theta_3$ 에 대하여 각각 식 (8), (9), (10)의 결과를 얻을 수 있다.

$$\theta_1 = \tan^{-1} \frac{\alpha P_x + \beta P_y}{\beta P_x - \alpha P_y} \quad (8)$$

$$\theta_2 = \tan^{-1} \frac{D_2S_3 - A_2C_3 - A_4}{D_{11}} + \tan^{-1} \frac{\sqrt{D_{11}^2 + (D_2S_3 - A_2C_3 - A_4)^2 - P_z^2}}{P_z} \quad (9)$$

$$\theta_3 = -\tan^{-1} \frac{AD_2 + DA_2}{AA_2 - DD_2} + \tan^{-1} \frac{\sqrt{(AA_2 - DD_2)^2 + (AD_2 + DA_2)^2 - P_z^2}}{(P_x^2 + P_y^2 + P_z^2 - D_1^2 - D_2^2 - D_{11}^2 - A_1^2 - A_2^2)/2} \quad (10)$$

이를 토대로 발바닥의 좌표를 시간에 따라 변화시켜 가면서 궤적을 만들어본 결과는 다음의 그림 3과 같다.

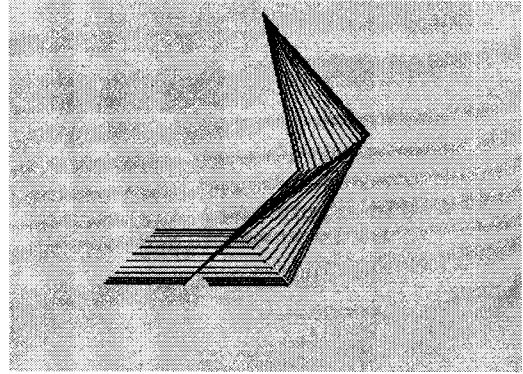


그림 2. 역기구학을 바탕으로 그린 다리의 궤적

또한, 이때의 $\theta_1, \theta_2, \theta_3$ 과 Tip의 위치 변화는 그림 3과 같이 얻었다.

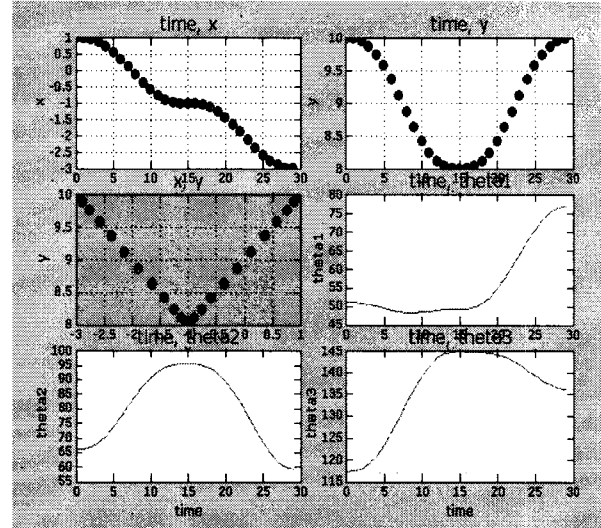


그림 3. 시간에 따른 Tip위치와 각 관절의 변화량

3. 그래픽 시뮬레이터를 이용한 모션티칭

앞에서 구한 기구학적인 지식을 기반으로 그래픽 시뮬레이터를 작성하였다. 대표적인 그래픽 라이브러리인 OpenGL과 오픈소스 물리엔진인 ODE(Open Dynamics Engine)을 이용하여 그림 4.와 같이 그래픽 시뮬레이터 겸 모션티칭 어플리케이션을 개발하였다.

티칭 모드는 두 가지를 지원하고 전방향 행렬식(Forward Matrix)을 이용한 동작과 자세 명령어 방식을 이용한 그중 하나인 조인트 티칭 모드에서는 사용자가 각 관절의 값을 입력하면 시뮬레이터는 이들 값을 물리엔진에 기술된 기구학에 입력을 한다. 물리엔진에서는 현재의 관절 값들로 부터 사용자가 입력한 관절의 값 즉, 목표 값까지의 시간에 따른 변화를 물리법칙을 적용하면서 보여준다. 이에 따라서 사용자가

입력한 관절의 값이 만약 불안정한 자세를 지시하고 있다면 시뮬레이터 상에서도 그림 5. 과 같이 넘어지게 된다.

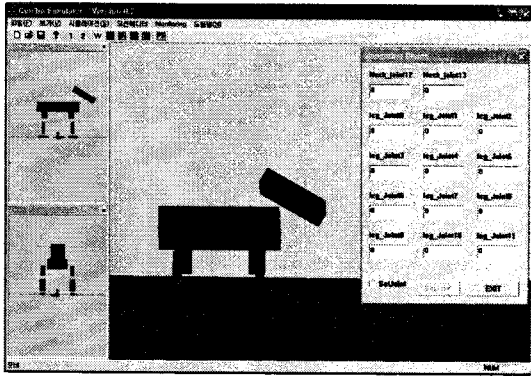


그림 4. 4족 보행로봇의 그래픽 시뮬레이터(조인트 티칭)

또 다른 모드인 카티션 티칭 모드에서는 그림 5. 의 우측과 같이 로봇의 몸체나 링크에 사용자가 직접 힘을 가하여 로봇의 자세를 변화시키는 방법이다.

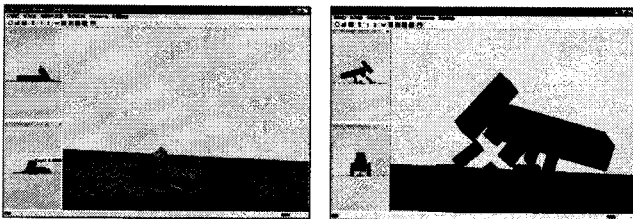


그림 5. 잘못된 지시로 넘어진 로봇그림(좌측), 4족 보행로봇의 그래픽 시뮬레이터(카티션 티칭)

로봇의 몸체상의 한 점에 임의의 힘을 가하면 그 점에 대한 관성력과 힘의 작용선 그리고 회전 모멘트를 무게중심으로 구할 수 있는데 이러한 수식적인 해석은 매우 복잡하여 계산과 해석이 쉽지 않다. 또한 만약 실제의 로봇이 자세를 유지하고 있는 상태에서 힘을 가하려면 로봇의 관절을 구동하는 모터에 무리를 주게 되어 로봇의 수명을 저하시킬 수 있다. 하지만 시뮬레이터 상에서는 그러한 위험 없이 로봇에게 자유롭게 힘을 가하여 로봇의 자세를 변화시킬 수 있다.

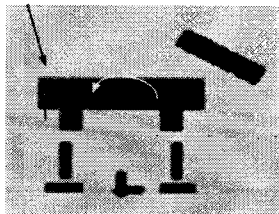


그림 6. 무게중심에 의한 단순화된 관절 표현과 특정지점에서의 외력(F_E 검은색)과 반력(R 빨간색) 그리고 회전력(M_W 파란색)의 벡터표현

$$M_W = F_E \times D + M_F, \quad M_F = \sum_{n=1}^4 R_{Fn} \times D_{Fn} \quad (11)$$

$$F_E = R_x + R_y$$

본 시뮬레이터는 ODE를 이용하여 물리적인 힘의 관계를 기구의 형태와 배치에 따라 각 연결부의 기계적 계수들을 설정하여 자연상태의 동작을 모방함으로써 몸체의 무게중심으로부터 미리 설정된 몇 군데의 지점에 대한 거리와 실제와 유사하게 가정된 무게중심과 무게를 이용 회전력과 우력

(couple force)을 단순계산하게 함으로 외력에 대한 반력과 회전력을 간단히 계산할 수 있도록 하였다. 또한, 이에 따른 로봇의 물리적 반응을 눈으로 바로 확인할 수 있어 매우 효과적으로 동작을 지시할 수 있다.

4. 결 론

그래픽 시뮬레이터는 시뮬레이션 결과를 미리 알려줄 수 있을 뿐 아니라 실제 시스템이 어떻게 동작할지를 한눈에 보여줄 수 있는 장점이 있다. 본 논문에서는 이러한 그래픽 시뮬레이터에 물리엔진을 결합하여 사용자가 보다 쉽고 효과적으로 로봇에게 동작을 지시할 수 있는 방법을 소개하였다.

로봇에게 사용자의 의도대로 동작을 취하게 하는 방법이 그치지 않고 사용자의 명령 동작에 의해 만들어진 동작이 실세계 즉 물리적인 의미를 포함하여 동작하도록 함으로써 동작의 궤적으로 인한 중력의 작용을 시뮬레이션 동작으로 알아볼 수가 있음을 보였다.

앞으로 가정용 로봇의 수요가 증가하는 추세를 보면 PC와 마찬가지로 이제 각 가정에도 애완용 로봇이나 가사 작업용 로봇이 한 대쯤은 존재하게 될 것이다. 실제 애완동물이 그러하듯 이러한 로봇들은 사람과의 감정의 교류에 동작과 반응을 이용한다. 따라서 본 논문에서 소개한 방법을 이용한다면 애완용 로봇에게 더욱 다양한 동작을 쉽고 빠르게 그리고 효과적으로 지시할 수 있을 것이다.

참 고 문 헌

- [1] 강택기, 이수영, "로커_보기 구조를 갖는 전방향 이동로봇 시스템 개발", 제어 자동화 시스템학회 논문지, 7호, 제 12권, pp. 679-685, 2006. 7.
- [2] B.W.Choi, M.J.Chung, Z.Bien, "Development of Software Environment for The Kinematic Analysis of A Quadruped Robot", International Workshop on Intelligent Robot and Systems, pp. 193-198, 1989.
- [3] H. Kimura, I. Shimoyama, and H. Miura. "Dynamics of a quadruped walking robot(in Japanese)." Jour. of Robotics Society of Japan, vol. 6, no. 5, pp. 3-14, 1988.