

PXA-255를 이용한 다족 모바일 로봇의 보행계획생성에 관한 연구

유영국*, 김기우*, 최우창*, 김진걸**

*인하대학교 전기공학과 석사과정, **인하대학교 전자전기공학부 교수

A Study on Walking Trajectory Generation of a Multipied Mobile Robot using the PXA-255

Young-Kuk Yoo*, Ki-Woo Kim*, Woo-Chang Choi*, Jin-Geol Kim**

*School of Electrical Engineering, Inha University, Incheon, South Korea
(Tel : +82-32-860-7384; E-mail: drager007@nate.com)

*School of Electrical Engineering, Inha University, Incheon, South Korea
(Tel : +82-32-860-7384; E-mail: kiwoo1004@hotmail.com)

*School of Electrical Engineering, Inha University, Incheon, South Korea
(Tel : +82-32-860-7384; E-mail: uchang2@gmail.com)

**School of Electrical Engineering, Inha University, Incheon, South Korea
(Tel : +82-32-860-7384; E-mail: john@inha.ac.kr)

Abstract - 본 연구는, 임베디드 리눅스 보드를 기반으로 센서에서 정보를 받아 장애물에 대한 거리 감지를 통해 보행계획을 생성할 수 있는 다족 모바일 로봇의 대해 기술하고자 한다. 현재 임베디드 시스템 분야는 반도체 기술과 소프트웨어 기술의 빠른 발전으로 인하여 시스템의 처리 능력이 급속도로 발전되고 있으며 임베디드 시스템은 적용분야를 점점 넓혀가고 있다. 본 연구는 리눅스 보드의 빠른 처리 능력과 효율적인 시스템 관리 능력을 다족 모바일 로봇에 적용하여 로봇의 안정적인 보행을 꾀하였다. 본 연구에서는 바퀴를 사용할 수 없는 지형에서 보행을 이용한 장애물회피 능력을 실험하고자, HYBUS사의 X-Hyper PXA-255A 보드를 적용하였다. 본 논문에 대해서는 실험을 통해 이를 검증 하였다.

사용 된다.



<그림 1> X-Hyper PXA255A 보드

1. 서 론

바퀴 달린 로봇들은 장애물을 회피하는 등의 원활한 이동을 기초로 손쉽게 개발되어 이용되고 있다.[1] 그러나 이러한 로봇들은 이동하는 경로에서 턱과 같은 장애물을 만나면 그 턱을 넘어가기가 어렵게 되기 때문에 로봇에 움직임에 제약을 갖게 된다. 본 연구에서는 앞에서 언급한 모바일 로봇이 갖는 지형에 따른 제약과 보행 로봇이 갖는 에너지 효율의 단점을 최소화 할 수 있는 다족 모바일 로봇을 제안하며, 환경에 맞는 보행 궤적 생성과 로봇형태를 결정하여 평지에서부터 바퀴구동에 의한 주행이 아닌 장애물이 나타났을 시의 상황에서 보행으로써 효율적인 장애물이 회피가 가능하게 하였다.[2]

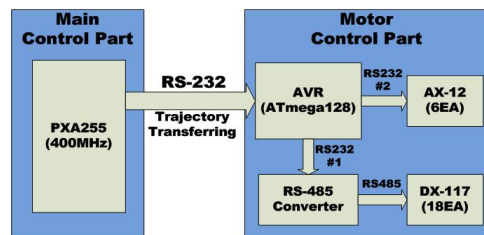
지금 소개되는 다족 모바일 로봇은 기본적으로 6족의 다리를 가지고 있으며 다리 끝에 바퀴 모듈이 장착되어 있어, 평지에서는 빠른 주행과 회전 주행이 가능하고 장애물이나 바퀴로 이동할 수 없는 지형이 나올 시에는 바퀴모듈을 고정시키고 관절을 이용한 보행을 통해 이동하게 된다. 본 연구는 각 상황에 맞는 보행계획을 만들어 주는 프로그램을 만들고 Matlab을 통하여 검증하였다.

메시지 제어기는 Main-Controller로부터 계산된 보행계획을 전송받아 모터로 전송하는 비교적 단순한 기능을 하고 있다. 메시지 제어기는 Atmel사의 ATmega128을 사용하고 있으며, 모터로의 데이터 전송을 위하여 Maxim사의 Max485를 사용하여 RS-232레벨에서 RS-485레벨로 변환해준다. ATmega128은 SPI, UART, PWM 등의 여러 가지 기능을 내장하고 있고 128K의 Flash메모리를 내장하고 있는 것이 특징이며 1Mhz의 클럭당 1MIPS의 속도를 낸다. 각 관절에 사용된 모터는 Robotis사의 DX-117으로 16V에서 약 38kg/Cm의 토크를 갖고 있으며, AX-12는 Endless-Turn기능을 가지고 있어 다리 끝의 구동 바퀴용 모터로 사용하였고, RS-232로 제어가 되어 ATmega128의 2번째 UART를 사용하여 제어 하였다. 관절에 사용된 DX-117모터는 RS-485통신으로 제어가 되며 양방향 통신이 가능하며, 또한 모터에 가해지는 부하 및 온도를 측정하여 전송이 가능하다. <그림 2>은 다족 모바일 로봇의 시스템 구조를 나타낸 그림이다. Main-Controller는 보행계획을 생성하여 약 100ms주기로 보행계획을 내려주며, 메시지 제어기에서는 이 보행계획을 받아 RS-485의 레벨로 변환하여 모터에 전송을 해준다.

2. 본 론

2.1 전체 구성

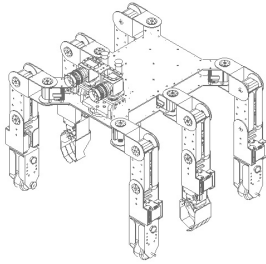
로봇 시스템은 보행계획 생성을 담당하는 중앙 처리기와 보행계획 데이터를 전송, 센서정보 취득, 환경설정 저장을 담당하는 메시지 컨트롤러로 구성되어 있다. Main-Controller는 Hybus사의 X-Hyper Pxa-255A 보드를 사용하였으며, 이 보드는 X-scale Pxa-255A(400Mhz)의 CPU를 사용 함으로써 빠른 처리속도를 갖으며, 32M의 SD RAM과 16M의 Flash Memory를 갖고 있어 넓은 확장성을 보장한다. 또한 보드 내부에는 USB 및 Serial Port등의 장치들이 탑재되어 있어 다른 기기와의 통신이 용이하며, 여러 가지 Daughter Board를 장착 할 수 있는 외부 핀이 나와 있다. 그리고 저전력을 소모하여 배터리로 동작하는 모바일로봇에 있어서는 상당한 유리하게 작용된다. 다족 모바일 로봇에 적용된 X-Hyper PXA-255A보드에는 Embedded Linux가 탑재되어 있으며 Kernel ver는 2.4.18이다. 개발 장비로는 GNU TOOL(gcc, glibc, binuti 1...) Compiler를 사용하였다. 아래의 그림 1은 다족 모바일 로봇에 적용된 Embedded Linux 보드인 X-Hyper Pxa255A의 사진이다. X-Hyper Pxa255A는 저전력을 소모하는 특징을 갖고 있어, 휴대용 기기에 많이



<그림 2> 시스템구조

본 연구에 적용된 다족 모바일 로봇인 ITIBO는 각 다리에 4개의 DOF, 총24개의 자유도를 가지고 있으며 각 관절은 약 200도의 자유 각도를 갖고 있다. 다리 끝에 바퀴모듈이 장착되어 있어 바퀴의 위치를

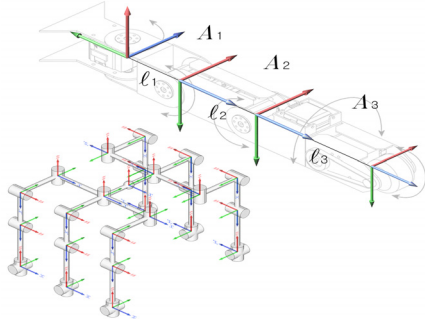
자유롭게 조정할 수 있게 하였다. <그림 3>는 6족 모바일 로봇 ITIBO의 3D모델링과 각 관절의 구조를 나타내고 있다.



<그림 3> 6족 모바일 로봇 ITIBO

2.2 기구적 해석

6족 모바일 로봇인 ITIBO의 기구학 해석을 위해 좌표계를 구성하였다. 로봇의 특성상 6개의 다리가 같은 형태로 되어 있어 하나의 다리 관절의 해석으로 전체 로봇의 구조를 해석할 수 있다. <그림 4>는 ITIBO 시스템의 좌표계이다.[3-4]



<그림 4> ITIBO의 좌표계 시스템

다족 모바일 로봇인 ITIBO의 기구학적 해석을 위해 강체의 링크 사이의 관계를 표시하는 <표 1>의 DH Parameter를 이용하여 세 부분의 단순 기구 연쇄로 표시 하였다.

<표 1> D-H Parameter

link	a	D	α	θ
First link	λ1	-90°	0	-90°
Second link	λ2	0°	0	0°
Third link	λ3	0°	0	0°

밑에 수식은 ITIBO의 Inverse Kinematics를 풀 수식으로써 총 3개의 θ를 구하는 식이다.

$$\begin{aligned}
 -\sin \theta_1 \cdot p_x + \cos \theta_1 \cdot p_y &= 0 \\
 a_3 \sin \theta_1 &= -p_z \\
 -\sin \theta_1 \theta_2 \cdot p_x + \cos \theta_1 \theta_2 \cdot p_y + \sin \theta_1 a_2 &= 0 \\
 \cos \theta_1 \theta_2 \cdot p_x + \sin \theta_1 \theta_2 \cdot p_y + \cos \theta_1 a_2 - a_1 &= a_3 \cos \theta_3
 \end{aligned}$$

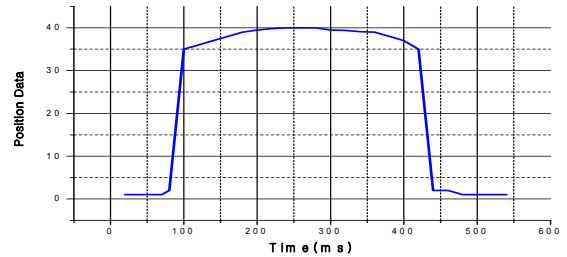
여기까지 수식은 θ를 구하기 위한 Homogeneous Transform을 연립해서 정리한 것이다. (1)번은 θ₁을 (2)번은 θ₂를 (3)번은 θ₃를 구하고 있다.

$$\begin{aligned}
 (1) \quad a \cos \theta + b \sin \theta &= 0 \Rightarrow A \tan 2(a, -b) \\
 \Rightarrow p_y \cdot \cos \theta_1 + (-p_x) \cdot \sin \theta_1 &= 0 \\
 (2) \quad a \cos \theta + b \sin \theta &= 0 \Rightarrow A \tan 2(b, a) \pm A \tan 2(\sqrt{a^2 + b^2 - c^2}, c) \\
 \Rightarrow p_y \cdot \cos \theta_1 \theta_2 + (-p_x) \cdot \sin \theta_1 \theta_2 &= -a_2 \cdot \sin \theta_1 \\
 \theta_{12} - \theta_1 &= \theta_2
 \end{aligned}$$

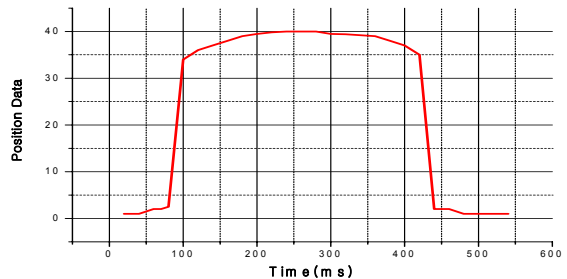
$$(3) \quad a_3 \cdot \sin \theta_3 = -p_z \Rightarrow a \sin \left(\frac{-p_z}{a_3} \right)$$

2.3 실험

Host PC에서 전진보행, 후진보행, 회전보행 등 몇가지 Position을 Forward로 산출하고 Main-Controller인 X-Hyper PXA-255A보드에 전송하였다. <그림 5>는 Inverse Kinematics를 풀어서 나온 θ값을 X-Hyper PXA-255A보드에서 만든 궤적 생성 코드에 적용하여 하나의 발이 움직일 때의 궤적을 그래프로 나타낸 것이다. <그림 6>은 위의 과정에서 나온 θ값을 Matlab에서 Forward Kinematics를 계산해서 나온 포지션 값을 계산해 나타낸 것이다. Matlab 검증을 통해 궤적 값을 확인해 본 결과 1.75%오차가 생기는 것을 알 수 있었는데, 이것은 시뮬레이션 상에서는 소수점 하위 자리에서 일정소수점까지 내려가면 반올림하는 반면, Matlab에서는 모든 소수점 하위 자리까지 그대로 계산해서 나오는 오차였다. 하지만 아주 미세한 오차이기에 궤적을 생성에 로봇이 걸음을 옮기는 데에는 거의 지장을 주지 않았다.



<그림 5> Inverse Kinematics 를 풀었을 때의 궤적 변화



<그림 6> Matlab으로 검증했을 때의 궤적변화

3. 결 론

본 논문에서는 임베디드 리눅스 보드인 X-Hyper PXA-255A보드에서 궤적을 생성하여 다족 모바일 로봇에 적용하는 것을 실험하였다. 향후 연구 목표는 장애물과 불규칙적인 지형에서 비전과 센서데이터에 의한 자율적인 판단을 하여 융통성 있게 극복할 수 있는 지능을 가진 로봇을 만들도록 연구하여야 한다. 그러기 위해서는 X-Hyper PXA-255보드 보다 조금 더 나은 성능의 보드가 필요하며, 로봇의 센서 적용과 비전처리에 대한 연구가 지속적으로 진행되어야 될 것이다.

후 기

이 논문은 2007년도 산업자원부의 지원에 의하여 기초전력공학공동연구소(R-2007-2-059) 주관으로 수행된 과제임.

[참 고 문 헌]

[1] Gen ENDO 외, "Study on Roller-Walker", Internationa Congerence on Robotics & Automation, pp. 2032~2037, 1999
 [2] Katsuhiko Tabata 외, "Development of a Transformational Mobile Robot to search Victims under Debris and Rubble -2nd report-Improvement of mechanism and Interface-" International Workshop on Safety, Security and Rescue Robotics, pp 19~24, 2005
 [3] K.S.Fu 외, "Robotics-Control, Sensing, Vision, and intelligence", pp 12~76, pp 149~198, 1987
 [4] Saeed b. Niku, "Introduction to Robotics-Analysis, Systems, Applications" pp 147~170, 2001