

화상 처리에 의한 등곡률반경 방식의 로봇 제어

Vision Steered Micro Robot for MIROSOT

차승엽*, °김병수**, 김경태***

*SeungYup Cha : chasy@solvit. .co.kr

**ByoungSoo Kim : ssing@hitel.net

***GyeongTae Kim : neokim@eeserver.korea.ar.kr

MaroTech

Tel. 82-(0)2-3448-5935

40-26 ChungDamDong KangNamGu Seoul, Korea

Abstracts This paper presents a robot which is steered by vision system. The proposed robot system has an AM188ES CPU(5.3 MIPS) and 2 DC motors with encoder and turns accurately at any speed and shows a movement like a human controlled car using a steering wheel. To the robot only steering angle value is sent without considering the speed. We present how to control this robot using our real time vision system.

1. 서 론

협동로봇 시스템을 구현하고, 구현한 로봇 시스템의 실시간 제어를 목적으로 하는 로봇 축구경기 Mirobot은 비전, 로봇 등의 여러 분야와 관련이 있는 대회이다. 이 대회의 역사가 짧으므로 인하여 이에 관한 연구 자료가 극소수로 존재한다. 따라서, 본 논문에서는 로봇 축구경기 Mirobot에 참가하였던 UFO팀의 협동 로봇 시스템의 실시간 제어기 설계에 관하여 논하고 시스템 구현 사양을 소개하고자 한다.

MIROSOT 시스템의 하드웨어 기본 단위는 비전과 로봇이다. 비전 시스템의 설계 목적은 CCD 카메라가 경기장 위의 2m높이에서 촬영한 영상자료를 기반으로 host 컴퓨터에서 로봇의 위치와 방향을 계산하고 이를 근간으로 로봇이 움직여야할 정보를 무선통신으로 로봇에게 전달하여 경기를 진행하는 것이다. 또한 로봇 시스템의 설계 목적은 비전 시스템의 계산 결과를 이용하여 고속으로 움직이는 여러 개의 로봇을 정확히 제어하는 것이다.

비전 시스템의 연산 속도는 영상 입력을 분석하여 처리하는 계산 시간을 고려한 sampling rate에 의하여 결정되며 고속의 로봇 시스템을 정확히 제어하기 위하여 빠른 비전 시스템의 개발이 요구되어진다. 본 논문에서는 새로운 비전 board를 개발하였으며, 비전 시스템의 연산 속도는 NTSC신호에서 설계 가능한 최고 속도인 초당 60번의 영상 처리를 320x240의 해상도에서 구현하였다. 또한, 설계한 시스템은 320x240의 해상도에서 각 필드의 pixel이 각기 다른 YUV 데이터를 갖고 있으며, 동시에 tracking가능한 Color는 256개의 Color이며 처리 가능한 object의 수는 기본 메모리 사용시 약 50개 정도이다.

로봇 시스템은 빠른 속도로 정확히 움직이는 하드웨어를 설계하였으며, 이를 실험할 수 있는 환경을 구현하였다. 본 논문

에서는 로봇을 제어하기 위한 방법으로써 자동차를 모델로 한 제어 기법에 관하여 논한다.

2. 시스템의 제어 사양

협동 로봇을 제어하기 위한 알고리즘이 여러 연구가들에 의하여 제안되었으며 이를 실제 시스템에 구현하는 연구 또한 활발히 진행되고 있다. 실제로 다수의 Robot Lab에서 이러한 알고리즘을 구현하기 위하여 simulation 및 로봇 제작이 이루어지고 있으며, MIROSOT는 이러한 연구를 실제로 구현하여 연구 교류를 이룰 수 있는 좋은 대회이다.

본 논문에서 구현한 system은 기존의 알고리즘을 구현하여 시스템 외부 상황에 둔감한 하드웨어 설계를 목표로 제작되었다. 시스템 설계시 고려할 제어 사양은 다음과 같다.

2.1 로봇의 관성

로봇을 고속으로 제어하기 위하여 수반되는 문제는 로봇의 관성에 대한 대책이다. 이 문제는 PID제어기법 등과 같은 기존의 제어 알고리즘으로 해결할 수 있는 경우와 그렇지 못한 경우가 있다.

로봇은 2차원에서 한 점이 아닌 두바퀴로 움직이므로 x,y로 나누어 해석하는 방법이 용이하지 않은 점도 발생하며 로봇이 움직이는 경로에서 첨점(미분불능점)이 발생한다. 이에 대한 상황에서 simulation을 통하여 좋은 제어 성능을 얻는다 하더라도 실제 구현시 많은 장애를 받으며 결과적으로 고속의 제어 구현이 어려워진다. 이는 로봇에 사용된 DC모터를 단순한 적분요소로만 생각하여 로봇의 경로를 smoothing하여 해결하면 비효율적인 결과를 얻을 뿐 아니라 로봇이 이동하는데 필요한 곡률반경을 산출해 내기가 매우 어렵게 된다. 이러한 곡률 반경은 로봇의

이동시 짧은 시간동안 관찰하면 로봇은 원운동을 하고 있으므로 로봇의 실제 궤적이 원하는 궤적을 추종하는 성능을 좌우하는 중요한 요소가 된다.

2.2 Sampling Rate

Sampling rate은 두가지 부분에서 system의 성능을 좌우한다. 첫째는 1장에서 언급한 비전의 속도이며 둘째는 로봇의 바퀴에 장착한 엔코더의 위치 분해능이다.

첫째, 비전 시스템의 속도에 관한 설계 사항은 다음과 같다. NTSC CCD Camera의 속도 한계에서 발생하는 문제로서 97년 현재 초당 30번 이상의 Color영상을 취득하는 CCD Camera를 구입하는 것은 일반적으로 용이하지 않다. 그러나, 취득한 영상을 EVEN field와 ODD field로 나누어 초당 60번의 처리를 하는 것은 가능하다. 또한, MIROSOT의 경기에서 항상 문제가 되는 것은 비전 시스템의 연산 속도보다는 비전 시스템의 오동작이다. 비전 시스템의 오동작은 조명이나 기타 외부 환경에 기인하며 이러한 오동작을 해결하여 외부 환경 환경에 둔감하며 동시에 연산 처리 속도가 빠른 비전 시스템을 구현하는 명확한 방법은 현재까지 제시되지 않고 있는 실정이다. 그러나, 그에 대한 차선책은 색상처리 과정에서 휘도신호를 둔감하게 하고 color만을 추출하기 용이한 YUV신호를 이용하는 방법이다. 이는 NTSC Camera를 사용할 경우 RGB신호를 이용하는 방법보다 더 효과적이다.

둘째, 로봇의 바퀴에 장착한 엔코더의 위치 분해능 문제는 다음과 같다. 로봇에 부착한 바퀴가 한바퀴 회전하는 동안 출력되는 엔코더의 펄스 수는 최대 300개 정도으로써 4cm지름의 바퀴를 가진 로봇이 1m/sec의 등속도로 이동하며 sampling rate이 1msec라고 가정하면

$$1\text{바퀴회전시 거리} = 40\pi = 125.66 \text{ mm}$$

$$1 \text{ 펄스동안 이동거리}$$

$$= 1\text{회전시 거리}/1\text{회전시 발생한 펄스수}$$

$$= 0.419\text{mm}$$

$$\text{거리} = \text{속도} * \text{시간} = 1000\text{mm} * 1\text{msec} = 1\text{mm}$$

$$1 \text{ sampling time동안 발생하는 펄스수} = 2.39 \quad (1)$$

즉, 1m/sec로 로봇이 이동할 경우는 2.3펄스만이 sampling되므로 분해능은 매우 저하됨을 알 수 있다. 이러한 문제를 해결하기 위한 방법은 엔코더의 분해능을 증가하거나 sampling rate time을 증가하는 방법이 있다. 그러나 언급한 두가지 방법 모두 실제 시스템 설계시 용이하지 않다. 따라서, 어느 정도의 분해능이 필요로 하느냐에 관한 정량적인 고찰을 하지 않는다 하더라도 언급한 문제는 분해능면에서 설계 사양으로 중요하게 고려되어야 하며 이를 해결하기 위한 다른 방법을 모색하여야 한다.

2.3 물체의 속도와 위치에 관한 예측(object estimation)

로봇이 다른 로봇과 충돌을 피하여 공을 원하는 방향으로 차려면 미리 공의 위치와 움직이는 방향을 예측해야 한다. 단지 비전 시스템에서 예측한 공의 위치 데이터를 이용하면 공과 로봇이 충돌은 가능하나, 정확한 방향으로 공을 차기가 어렵다. 결국은 공의 위치와 속도를 예측해야만 이것이 가능하며, 그러기 위해서는 다음과 같은 문제를 해결하여야 한다.

경기중에 모든 object가 직선운동을 한다고 가정하면, 공의

위치를 예측하는 것은 쉬운 일이다. 그러나, 실제 경기구로 사용되는 골프공은 표면이 매끄럽지 않아서 공의 속도가 일정 속도보다 작으면 직진성을 잃고 브라운 운동을 하게 된다. 또한, 경기장 벽의 탄성이 예측 가능한 입사·반사 법칙을 따르지 않는 성질을 가지므로 입사각과 반사각이 매우 심한 차이를 보인다. 따라서, 이러한 문제를 해결하기 위하여 실험적으로 입사·반사각을 취득하여 이를 보상하는 알고리즘을 설계하여야만 정확한 볼의 위치와 속도를 예측할 수 있다. 또한, 장애물 회피 문제의 경우도 위치 및 속도에 관한 예측이 매우 중요하는데 그 이유는 로봇 이동 결정 초기에 경로선택에 중요한 영향을 미치기 때문이다.

2.4 시스템 개발 환경

시스템의 개발 환경은 Windows95나 NT등을 사용하는 것이 일반적이나 로봇을 화면에 나타내면서 초당 60번의 비전 시스템 알고리즘을 수행하는 것은 Windows 환경에서 시간적으로 매우 어려운 일이다. 더우기 PC의 수행과제가 Hardware와 매우 밀접한 관련이 있으므로 이번 시스템에서는 DOS를 사용하였다. 또한, DOS를 사용하므로써 PC내부의 82C54나 16C550을 직접 access할 수 있을 뿐아니라 Interrupt를 사용할 수 있다. 특히, 82C54를 직접 access함으로써 일반적인 OS가 제공하는 timer보다 훨씬더 정밀한 timer를 사용할 수 있다.

로봇의 전지를 자주 교환하는 것 또한 적지않은 제한 요건이다. 이러한, 전지 교환 어려움을 해결하기 위하여 니켈메탈수소 전지 1200mAh의 대용량 충전지를 사용하였다. 또한, 로봇 자체에서 Power save mode를 지원하도록 하였으므로 이럴 경우 한번 충전에 2시간정도 동작이 가능하다.

3. 로봇제어 알고리즘의 설계

로봇의 제어 알고리즘 설계시 2장에서 기술한 문제점들을 고려하여야 한다. 또한, 전체 시스템 알고리즘을 효율적으로 적용하기 위하여 로봇의 제어 알고리즘은 고정밀성이 요구된다. 이러한 고정밀성의 로봇 제어 알고리즘을 설계하기 위하여 다음과 같은 사양을 고려하였다.

첫째, 정확한 방향으로 공을 차기위한 문제는 다음과 같다. 로봇의 주행속도를 감소하는 경우는 공을 차기 위하여 timing을 결정하기 위한 방법으로만 이용하여야 하며, 회전을 용이하게 하기 위한 방법으로는 이용하지 않는다. 회전을 위하여 속도를 감소하는 방법은 고속 정확한 주행(여기서의 정확한 주행이란 단순히 경로를 정확히 추종하는 것이 아니며, 특정점을 통과할 때의 시간이 정확한 것을 의미)을 하기 어렵게 된다. 이러한 문제를 해결하기 위해 로봇이 주행에 필요한 최소의 곡률반경에서 낼 수 있는 속도의 한계점을 계산하고 계산된 속도를 로봇의 최고 속도로 규정한다. UFO 로봇의 경우, 알고리즘의 원활한 수행을 위해서 필요로 하는 최소 곡률반경은 6.5cm로 정하였으며 속도한계는 1m/sec이다. 로봇이 직선 주행시에는 속도한계보다 더 고속으로 주행할 수 있다 하더라도 결정된 속도한계 이상의 속도는 갖지 않는다.

둘째, 로봇 제어에서 꼭 필요한 사항은 속도에 상관없이 동일한 반경의 원을 그릴 수 있어야 한다는 것이다. 즉 일정한 원을 따라 주행을 할 경우 속도를 증가하여도 주행하는 원의 반경은

변하지 않아야 한다. 이를 해결하기 위한 방법으로 자동차의 steering wheel이 기능하는 방식을 모방하여 로봇을 제어하였다.

위에 언급한 사항들을 만족시키기 위해 로봇은 주행중에 많은 연산을 수행하여야 한다. 이를 구현하기 위하여 5.3MIPS 성능의 16bit CPU를 사용하여 4개의 interrupt routine을 처리하였다.

3.1 Steering Method

위에서 언급했듯이 로봇을 제어하는 기본 방식으로 steering wheel을 modeling하였으며 이는 측구라는 환경과 밀접한 연관이 있기 때문이다.

측구는 장애물 회피뿐만 아니라 정확히 공을 원하는 방향으로 차야 하기 때문에 공과 만나는 지점과 방향과 시간을 예측하여야 한다. 이러한 상황을 2차원에서 질량(관성)을 가진 물체를 가지고 구현한다는 것은 대단히 어려운 일이다. 그리고, 물체들은 점도 아니며 omni-directional control도 불가능한 두 바퀴를 가진 물체이므로 시스템 구현이 어렵다. 그러나, 주어진 상황을 일차원상에서 해결한다면 문제는 그렇게 어렵지 않다. 즉, 2차원 상에서 이동하는 자동차가 움직이는 물체를 의도한 방향으로 차는 것은 어렵지만 1차원 상에서 이동하는 기차가 물체를 특정시간에 차는 것은 2차방정식만으로 해결이 가능하기 때문이다. 이러한 관점에서 문제를 해결하기 위하여 로봇 제어 알고리즘의 설계는 올바른 방향으로 rail을 깔아주고 로봇은 기차처럼 움직이는 역할을 수행하게 하면 되는 것이다. 여기서, 로봇이 기차처럼 움직인다는 것은 어떤 의미는 로봇이 특정한 회전 반경으로 이동할 때 속도가 변화해도 그 회전 반경이 일정한 것을 의미한다.

로봇이 속도에 관계없이 일정한 원운동을 하기 위해서는 좌측 모터속력과 우측 모터속력의 비가 항상 일정하게 유지되어야 한다. 이렇게 왼쪽 모터와 오른쪽 모터의 속도 비를 일정하게 유지하기 위하여 다음과 같은 제어 알고리즘을 설계하였다.

로봇이 곡률반경을 유지하면서 turn을 하는 방법중 가장 쉽게 생각할 수 있는 것은 양쪽 바퀴에 일정속력을 가감하는 방법이다. 즉, 좌측 모터의 속력을 ls (left motor speed), 우측 모터의 속력을 rs (right motor speed), 실제 속력을 os (origin speed)라 정의하고, 로봇이 좌측으로 회전한다고 가정하면, 식(2)와 같이 모터의 속력을 결정할 수 있다.

$$\begin{aligned} ls &= os - k \\ rs &= os + k \end{aligned} \quad (2)$$

(k: constant number)

그러나, 이 방법에서는 os 가 변화하면 양쪽바퀴의 속도 비율이 달라져서 곡률반경이 변화하게 된다. 이 비율을 일정하게 유지하기 위하여 식(3)과 같이 실제 속력에 상수를 곱한 후 가감한다.

$$\begin{aligned} ls &= os - os * k \\ rs &= os + os * k \\ ls : rs &= (os - os * k) : (os + os * k) \end{aligned} \quad (3)$$

만약, 여기서 n 배의 속도를 낸다면 (n 은 임의의 실수) 이것은

os 가 $n*os$ 로 변화했음을 의미한다.

$$\begin{aligned} (n*ls) / (n*rs) &= (n*os - n*os*k) / (n*os + n*os*k) \\ &= n(os - os*k) / n(os + os*k) \\ &= (os - os*k) / (os + os*k) \\ &= ls/rs \end{aligned} \quad (4)$$

즉 식(3)과 같이 속도를 정의하면 임의의 속도에 대해 항상 왼쪽 모터의 속도와 오른쪽 모터의 속도의 비율이 같게 되어 속도에 무관한 곡률반경을 얻을 수 있다.

$$\begin{aligned} \text{Left motor speed} &= \text{Origin speed} - \text{Origin speed} * \text{Turn Rate} \\ \text{Right motor speed} &= \text{Origin speed} + \text{Origin speed} * \text{Turn Rate} \end{aligned}$$

그러나, 위에 제시한 방법을 이용하여 원하는 속력에 비례하는 전류를 모터에 공급해서는 모터를 정확히 회전할 수 없으므로 원심력, 지면과의 운동마찰력과 접지력, 그리고 가속과 감속시에 필요한 힘의 계산을 모두 합하여 PID제어방법으로 모터를 구동하는 알고리즘을 설계하였다.

3.2 기타 로봇제어

UFO 로봇의 경우는 바퀴가 0.325mm를 이동할 때마다 sampling하여 이를 속도로 연산한다. 이것은 일정 시간 간격마다 sampling하는 것이 실제 micro robot의 Hardware에서 구현하기 어려우므로 일정거리마다 sampling하여 속도를 계산하는 것으로서, 이로 인하여 발생하는 오차를 보상하기 위하여 제어 알고리즘에 보상 알고리즘을 추가하였다. 이때에 1m/sec의 속도에서 입력되는 timer값은 200을 넘는 결과를 얻을 수 있으며 (40MHz crystal 사용시), 기존의 발상보다 충분한 분해능을 얻을 수 있었다.

4. 결 론

본 논문에서는 로봇 축구경기 Mirosoft에 참가하였던 UFO팀의 협동 로봇 시스템의 실시간 제어기 설계와 시스템 구현 사양을 설명하였다. 설계된 시스템에서는 로봇의 고속 고정밀 제어 성능을 얻는 하드웨어를 설계하였으며, 이를 실험하는 환경을 구현하였다. 또한, 로봇을 제어하기 위한 방법으로써 자동차를 모델로 한 제어 기법을 이용하였다. UFO팀의 비전 시스템의 성능을 향상시키기 위하여 새로운 비전 board를 개발하였으며, 비전 시스템의 연산 속도는 NTSC신호에서 설계 가능한 최고 속도인 초당 60번의 영상 처리를 320x240의 해상도에서 구현하였다.