2007년도 대한전기학회 하계학술대회 논문집 2007. 7. 18 - 20

옴니휠을 가진 모바일 로봇을 위한 시리얼 서보 동기화에 대한 연구

김대영, 이건영
광운대학교 전기공학과

A Study of Serial Servo Synchronization for Mobile Robot Using Omni-wheel

Dae Young Kim, Keon Young Yi
Dept. Electrical Eng. Kwang woon Univ

Abstract - This paper describes a simple method to reduce rotation angle error of mobile robot using omni-wheel[3](omni-bot). This method can be applied to not only omni-bot, but also other robot with a large number of servo motor. Robot using many servo motor as omni-bot is complicated for hardware and software, each servo motor has difficulty in synchronizing. Three servo motor, three omni-wheel and three serial servo motor controller is used, PC or Micro Processor interface with the serial servo controller through "SSC100" protocol. In order to check the improvement of the proposed serial servo synchronization compared to existing sequential communication method. comparing object is rotation angle error of omni-bot.

The results of this make building of omni-bot system easy and decrease rotation angle error.

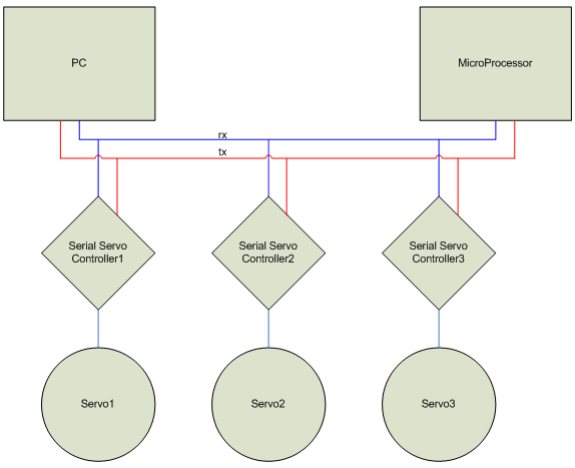
1. 서 론

움직임이 요구되는 모바일 로봇에서 액추에이터(actuator)는 필수적이다. 하나의 서보모터가 아닌 세 개의 서보모터로 구동되는 옴니휠을 가진 모바일 로봇에서 서보모터간의 동기화는 중요한 이슈중의 하나이다. 기존의 방법들은 서보모터의 이동 값의 오차를 검출해내 보간하는 방법 등이었고 제어 프로그램 작성과정에서 상당한 제어지식이 요구되었다. 이 방법들은 하드웨어적으로나 소프트웨어적으로 시스템을 복잡하게 만들고 시스템의 복잡성은 비용의 증가로 이어진다. 이러한 점들을 개선하기 위해 널리 사용되는 RS-232(시리얼통신)방법에 의한 각각의 서보모터를 동기화시키는 방법에 대해 제안하고 실험을 통해서 개선점을 확인한다.

2. 본 론

본 논문에서는 옴니휠을 가진 모바일 로봇을 제어하는데 있어서 보다 정밀한 움직임과 시스템의 복잡성을 피하기 위한 방법으로 시리얼 서보 동기화를 응용한다. 본론에서는 시스템의 구성과 "SSC100" 프로토콜 및 알고리즘을 살펴보고 로봇의 기구학적 해석을 통해 이론적인 회전각을 계산한다. 또한 실험을 통해 기존의 시리얼 통신을 기반으로 한 순차방법과 비교분석해 본다.

2.1 시스템의 구성

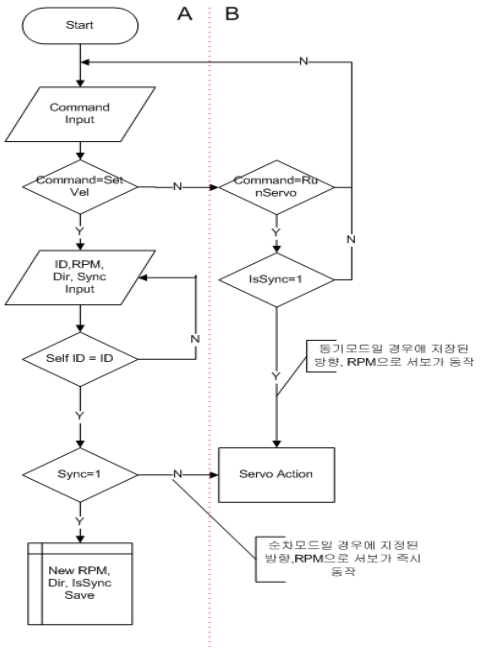


<그림 1> 시스템 블록도

이 로봇은 세 개의 서보모터에 옴니휠이 각각 연결되어 있고 시리얼 서보 컨트롤러에 의해 제어된다. <그림 1>에서 보여 지는 것처럼 시리얼 서보모터 컨트롤러끼리 버스라인(rx, tx)으로 연결되고 버스라인을 통해 명령을 전송할 수 있는 시스템이 구성된다. 명령을 내리는 PC나 마이크로프로세서에 사용할 개수만큼 서보컨트롤러를 연결하는 것으로 하드웨어적 구성은 간단히 완료된다.

2.2 "SSC100 " 프로토콜과 알고리즘 분석

시리얼 서보 동기화를 구현하기 위해서 "SSC100" 프로토콜을 정의하였으며 "SSC100" 프로토콜은 각각의 시리얼 서보 컨트롤러에 내장되어 있어 PC나 마이크로프로세서와 1:N으로 통신할 수 있다. 시리얼 서보 컨트롤러의 내부 파라메타인 통신 속도, ID, PID이득, 엔코더 정보, RPM(revolution per minute), Dir(CW/CCW) 등을 시리얼 통신을 기반으로 한 GUI응용프로그램에서 변경 할 수 있다. 시리얼 서보 컨트롤러는 버스라인을 통해 들어오는 명령이 자신의 아이디와 같으면 처리하고 아닐 경우에는 무시하게 된다.



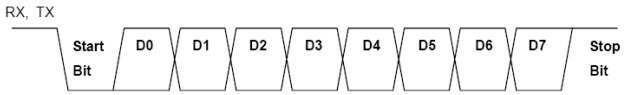
<그림 2> 시리얼 서보 컨트롤러 알고리즘 흐름도

동기모드(Sync=1)에서는 입력된 RPM과 Dir을 "RunServo" 명령이 입력될 때까지 내부 저장 공간에 저장하고 대기한다. "RunServo" 명령이 입력되면 <그림 2>에 B영역처럼 세 개의 시리얼 서보 컨트롤러가 기존에 저장된 RPM, Dir값에 의해 동시에 서보모터를 구동하게 된다. 세 개의 시리얼 서보 컨트롤러에 동시에 "RunServo" 명령이 입력되므로 서보 컨트롤러의 동작시점이 일치하고 처리시간도 같으므로 동기화가 이루어진다.

순차모드에서는 각각의 시리얼 서보 컨트롤러에 "SetVel" 명령이 전송되는 시점에서 서보모터를 구동하게 된다. 각 시리얼 서보 컨트롤러에 순차적으로 명령을 전송하고 통신 속도에 따라서 명령을 전송 받는 시점이 달라져 서보 모터는 동기화가 이루어지지 않고 이동 값의 오차가 생기게 된다.

2.2.1 통신 속도에 따른 동작 시점 지연

순차모드의 경우에 통신 속도에 따른 지연시간을 계산하는 작업이 필요하다. 시리얼 서보 컨트롤러는 RS-232통신 방식이며 1바이트 전송시 필요한 비트수는 <그림 3>과 같이 1 start bit, 8 data bit, 1 stop bit 총 10비트로 구성된다. 통신 속도가 9600bps라고 하면 1비트를 전송하는데 걸리는 시간은 1/9600=104.16us 이며 1바이트의 지연시간은 1041.6 us가 된다. "SSC100" 프로토콜에서 "SetVel" 명령어를 전송하기 위한 바이트는 12바이트이므로 전송에 걸리는 시간은 1041.6usx12 = 12.5ms 이다. 이 결과는 마이크로프로세서의 프로세싱지연시간을 제외한 결과이며 전송시간만큼 각각의 옴니휠 동작의 시점이 지연되게 된다.



<그림 3> RS-232 통신의 신호타이밍, 8 data, 1 stop bit

2.3 로봇 기구학

로봇의 기구학적인 해석을 통해 이론적인 회전각이 계산되며 이 값은 실험에서 측정된 회전각과 비교하는데 이용된다.[1][2]

$$\begin{bmatrix}\dot{\phi}_1\\ \dot{\phi}_2\\ \dot{\phi}_3\end{bmatrix}=\frac{1}{r}\begin{bmatrix}-\sin\theta & \cos\theta & L\\ -\sin(\frac{\pi}{3}-\theta) & \cos(\frac{\pi}{3}-\theta) & L\\ \sin(\frac{\pi}{3}+\theta) & \cos(\frac{\pi}{3}+\theta) & L\end{bmatrix}\begin{bmatrix}\dot{x}\\ \dot{y}\\ \dot{\theta}\end{bmatrix} \quad (1)$$

(1)은 로봇의 기구학 방정식이고, 다시 말하면 방정식(2)와 같다.

$$\dot{\phi}=\frac{1}{r}W\dot{S} \quad (2)$$

$\dot{\phi}$는 바퀴의 각속도이며 $\dot{S}$는 로봇의 선속도와 각속도를 포함한다. L은 로봇의 중심으로부터 바퀴까지의 거리를 의미하며 r의 바퀴의 반지름이다.

로봇의 기구학 방정식(1)을 이용하기 위해 x ,y 그리고 θ는 $\dot{\phi}$ 의해 계산되는 파라메타이다. W의 역행렬을 이용해서 방정식(1)을 풀어보면 방정식(3)과 같다.

$$\dot{S}=rW^{-1}\dot{\phi} \quad (3)$$

방정식(1)을 이용해서 x, y그리고 θ를 바로 계산하기 위해서 방정식(4)과 같이 적분을 취하게 된다.

$$\dot{S}=r\int W^{-1}\dot{\phi}dt \quad (4)$$

$$S=\begin{bmatrix}x\\ y\\ \theta\end{bmatrix}=\begin{bmatrix}x(\phi_1,\phi_2,\phi_3)\\ y(\phi_1,\phi_2,\phi_3)\\ \theta(\phi_1,\phi_2,\phi_3)\end{bmatrix} \quad (5)$$

방정식(4)의 결과는 아래와 같다.

$$x=r\int\frac{1}{3\sin(\frac{\pi}{3})}\begin{bmatrix}(\cos(\frac{\pi}{3}+\theta)-\cos(\frac{\pi}{3}-\theta))\dot{\phi}_1+\\ (-\cos(\theta_0-\cos(\frac{\pi}{3}+\theta))\dot{\phi}_2+(\cos(\theta_c-\cos(\frac{\pi}{3}-\theta))\dot{\phi}_3\end{bmatrix}dt$$

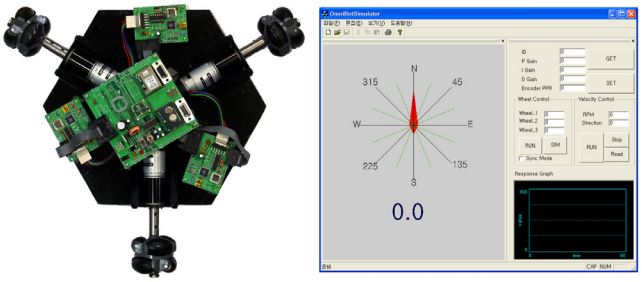
$$y=r\int\frac{1}{3\sin(\frac{\pi}{3})}\begin{bmatrix}(\sin(\frac{\pi}{3}-\theta)-\sin(\frac{\pi}{3}+\theta))\dot{\phi}_1+\\ (-\sin(\theta_i-\sin(\frac{\pi}{3}+\theta))\dot{\phi}_2+(\sin(\theta_s-\sin(\frac{\pi}{3}-\theta))\dot{\phi}_3\end{bmatrix}dt$$

$$\theta=r\int\frac{\dot{\phi}_1+\dot{\phi}_2+\dot{\phi}_3}{3L}dt \quad (6)$$

방정식(6)을 더 간단히 풀어보면 아래 방정식(7)과 같다.

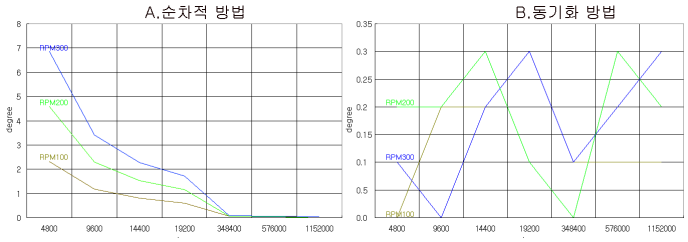
$$\theta=\frac{r}{3L}\left[\int\dot{\phi}_1dt+\int\dot{\phi}_1dt+\int\dot{\phi}_3dt\right]=\frac{r}{3L}(\dot{\phi}_1+\dot{\phi}_2+\dot{\phi}_3) \quad (7)$$

2.4 순차방법과 동기화방법에 대한 실험 및 분석



<그림 4>옴니봇과 GUI응용프로그램

이 실험은 순차방법과 동기화방법을 적용했을 때 회전각의 오차를 알기 위해 각 모터의 RPM을 같도록 하였고 동일한 규격의 옴니휠과 모터를 사용했다. 구동 명령을 받은 시점에서 1초 후에 0.1 degree 분해능을 가진 전자콤파스를 이용해 로봇의 회전각을 측정했다. 시리얼 서보 동기화의 개선점을 기존의 순차방법과 비교하기 위해 <그림 4>와 같이 로봇을 구성하고 GUI응용프로그램을 제작하였으며 이론적인 수치에 대한 회전각의 절대오차 값을 획득했다.



<그림 5> 순차방법과 동기화방법일 때 오차 그래프

<그림 5> A에서 보여 지는 것처럼 순차방법으로 세 개의 모터를 구동하는 경우에는 RPM이 높을수록 통신 속도가 낮을수록 오차가 커지는 것을 그래프를 통해 확인 할 수 있다. 이때 절대오차는 2.3°~7°이다. 시리얼 서보 동기화 방법을 이용할 때는 <그림 5> B처럼 서보모터의 RPM과 통신 속도에 독립적이고 절대오차는 0.0°~0.3°이다. 제어할 대상의 서보모터 수만큼 시리얼 서보 컨트롤러를 연결하고 고유의 ID를 부여한 후에 몇 개의 명령만으로 로봇을 원하는 위치와 방향으로 이동할 수 있었다. 이 과정은 서보모터 컨트롤러의 처리시간, 통신 속도 및 명령을 내리는 PC나 마이크로프로세서의 사양을 전혀 고려하지 않고 할 수 있었다.

3. 결 론

단일의 마이크로프로세서로 다수의 서보를 제어할 때는 마이크로프로세서의 프로세싱 지연시간에 의한 오차가 발생하게 되며 이러한 처리 때문에 마이크로프로세서의 처리부담은 커지고 시스템은 복잡하게 된다. 시스템의 복잡함을 줄이기 위해 통신을 기반으로 한 서보 컨트롤러를 이용하게 되면 통신 속도에 따른 지연시간 때문에 오차가 발생하게 된다.

본 연구에서는 "SSC100" 프로토콜을 기반으로 한 시리얼 서보 컨트롤러를 옴니휠을 가진 모바일 로봇에 적용해봄으로써 2가지 개선점을 확인했다. 옴니휠을 가진 모바일 로봇의 구동부를 간단히 구축할 수 있었고 통신 속도와 프로세서 처리시간의 고려 없이 서보모터의 동기화를 구현하였다.

[참 고 문 헌]

[1] Moballegh, H.R. Amini, P. Pakzad, Y. Hashemi, M. Nannia ni, M. "An Improvement of Self-Localization for Omnidirection al Mobile Robots Using a New Odmetry Sensor and Omnidire ctional Vision, IEEE, 2337-2340, 2004

[2] Ould-Khessal, N. Inf. Technol. & Telecommun. Dept., Vaasa Polyt ech., Finland;., "Design and Implementation of a Robot soccer Team b ased on Omni-directional Wheels", IEEE, 2005

[3]Acroname Robotics, http://www.acroname.com/robotics/parts/R277-PPRK-BS-2-BLACK.html