

# 모션 컨트롤러를 이용한 축구 로봇의 개발

최재일

대한로봇축구협회 사무국

## 1. 서 론

본 팀은 작년 처음 로봇축구를 시작하여 제 2회 대회인 2000년 Fira Korea-Cup에 참가하였다. 로봇축구를 시작한지 불과 1년도 되지 않은 상황에서의 대회출전이라 참가하는데 의의를 두었다. 비록 좋은 성적을 내지는 못하였지만 나름대로 가능성과 성과는 거둘 수 있었다. 그러나 로봇의 성능 면에서 여러 가지 문제점이 대회참가를 통해 발견되었다. 그래서 처음에 논문으로 계획하였던 기술 알고리즘의 개발을 뒤로 미루고 우리 기술로 DC모터를 이용한 축구로봇을 개발하기로 결정하였다.

본 팀은 축구로봇의 개발 시 "최대한 안정한 로봇, 최대한 민첩한 로봇, 최대한 튼튼한 로봇"의 제작을 목표로 세웠다. 로봇축구는 상대로봇 및 경기장의 벽과 충돌하는 경우가 빈번하기 때문에 안정성과 견고성이야말로 가장 최우선으로 고려되어야 할 사항이다. 실제로 대회 중에 CPU보드가 동작을 멈추어 로봇이 경기장 가운데 멈추어 선 경우도 있었고, 기어가 빠지거나 파손되는 경우나 로봇의 바디가 파손되는 경우도 종종 볼 수 있었다. 그리고 로봇의 속도와 순발력 또한 로봇의 성능을 좌우하는 중요사항이기 때문에 이 문제 또한 고려하여 개발하기로 하였다.

우선 로봇의 무게 감량 및 견고성을 위해 바디의 재료로는 가볍고 단단한 듀랄루민 소재를 선택하였고, 모터로는 Minimotor사의 DC모터를 사용하였다. 그리고, DC모터 PID제어를 위해 National Semiconductor사의 Motion Controller LM629를 사용하였다. 물론 소프트웨어적인 디지털 제어기를 설계할 수도 있었으나, 축구로봇에 응용하기에는 주변회로가 차지하는 공간상의 문제와 제어기 설계시의 시간상의 문제, 매 샘플링 시간마다 두 개의 모터를 제어해야하는 CPU의 처리속도 문제 등을 모두 만족하려면 메인 컨트롤러를 DSP수준을 사용해야하는데 컴파일러의 구입에 따르는 비용상의 문제로 인해 보다 값싸고 안정적인 하드웨어적으로 PID제어를 하는 Motion Controller LM629를 사용하여 축구로봇을 제작하게 되었다.

## 2. 본 론

### 2.1 로봇 시스템의 설계

#### 2.1.1 축구로봇의 설계조건

로봇을 만들기 위해서는 항상 고려해야 할 사항들이 있다. 앞서서도 언급했지만 안정성, 견고성 등 가능하면 많은 장점들을 만족하도록 하드웨어를 설계해야한다. 하드웨어 설계 시 고려해야할 사항들을 열거해 보았다.

- 전기적, 물리적으로 안정해야 한다.
- 로봇의 몸체는 강한 충격에도 견딜 수 있는 충분한 견고함이 있어야 한다.
- 가능하면 가볍게 만들어야 제어가 쉽고 전력소비가 적다.
- PCB설계 시 모터에 인가되는 전력라인은 모터에 충분한 전류가 인가될 만큼 넓어야 한다.
- 적은 용량의 건전지로 오래 사용할 수 있을 만큼 회로의 전력소비가 적어야 한다.
- 회로의 전원은 항상 정전압을 유지할 수 있어야 회로의 안정성이 보장된다.
- 회로의 전원의 잡음에 의한 오동작이 없도록 전원노이즈에 대해서도 고려해야 한다.
- 7.5cm×7.5cm×7.5cm의 크기제한이 있으므로 각 부분의 배치도 신중히 고려해야 한다.
- VCC와 GND는 쇼트가 나지 않도록 항상 주의해야 한다.
- 바퀴는 마찰력이 커야 미끄러짐을 막을 수 있어 제어의 오차를 줄일 수 있다.
- 충전지는 착탈이 쉽도록 로봇의 구조를 설계해야 한다.

#### 2.1.2 축구로봇의 전체구성

그림 1은 로봇 시스템의 주요 구성부를 나타낸 것으로, 로봇 시스템은 크게 마이크로컨트롤러, 모터 구동부, 모션 컨트롤러, 통신부, 전원부로 이루어져 있다. 완성된 로봇은 크게 회로기판, 충전지, 모터, 몸체(바퀴 포함)로 구성된다. 로봇의 회로기판은 2개로 분리되어 있으며 밑에 기판에는 모터드라이브와 Motion Controller가 있고 위에 기판에는 마이크로 컨트롤러와 RF통신 모듈, 전원 레귤레이터가 있다. 충전지는 로봇의 모터전원과 로직회로의 전원으로 쓰인다. 모터와 바퀴는 8:1 기어비의 내접 기어로 연결되며 모터는 엔코더 일체형의 소형 DC 모터를 사용한다. 로봇의 특성을 좋게 하는 요인 중에 하나가 모터이며 모터의 반응시간이 짧고 선형특성이 좋으면 로봇의 성능이 뛰어나게 된다.

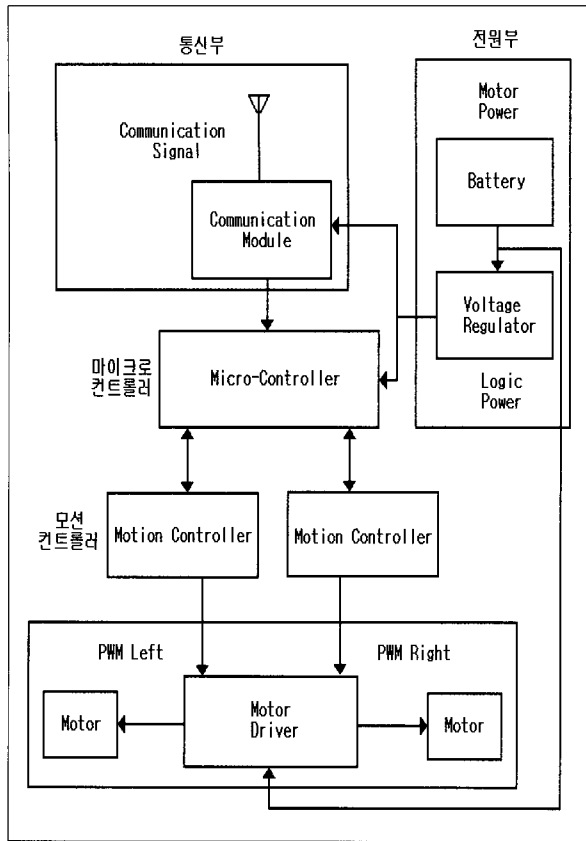


그림 1. 로봇 시스템 주요 구성부

· 다음은 제작하려는 축구로봇 시스템 사양이다.

표 1. 축구로봇 시스템 사양

모터	Minimotor2224(006SR)
Encoder	512ppr
기어비	8:1의 내접기어
CPU	i80C196KC
모터 드라이브	L298
모터 제어방식	PID Control
통신 방식	RF module(418&418MHz)
크기	7.5 × 7.5 × 7.5cm
무게	600g
최고속도	2.6m/s
배터리	NiMH(9.6V)

## 2.2 모터 및 모터 구동부

### 2.2.1 모터의 선정

모터 선정기준인 토크, 소비 전력, 회전 속도, 크기 등

을 고려하여 축구 로봇에 사용하기에 적합한 모터를 결정하였다. 소비 전력의 경우는 사용하는 충전지와 직접 관계가 있는데, 충전지는 니켈 수소로 전체 9.6V, 800mA의 평균 공급 전류 중 약 300mA는 회로부에서, 500mA는 기계부인 모터가 사용할 수 있다고 가정한다.

회전 속도는 로봇의 직선 최고 속도 및 바퀴 크기로부터 계산할 수 있으며, 직선 최고 속도가 2m/s, 바퀴의 반지름이 2.5 cm 라고 하면 바퀴 축의 회전 속도  $n$ 은

$$n = \frac{200cm/s}{2 \times 2.5cm \times \pi} = 12.7 \text{ rev/s} = 764 \text{ rpm}$$

이 된다.

토크는 로봇의 무게와 로봇의 바퀴와 경기장면 사이의 마찰 계수와 관계가 있으므로, 로봇의 무게가 0.6kg, 경기장 면과 로봇 바퀴와의 마찰 계수를 대략 0.4, 바퀴 축의 지름을 0.3 cm 라고 하면 최소한의 토크  $\tau_G$ 는

$$\tau_G = \frac{1}{2} \times F \times R = \frac{1}{2} \times \mu \times m \times g \times R_0 = \frac{1}{2}$$

$$\times 0.4 \times 0.6kg \times 9.8m/s^2 \times 0.003m = 3.53 \text{ mNm}$$

이다.

크기는 기어와 모터, 엔코더를 포함하여 3.8cm 이내라면 문제없다.

이러한 기어와 모터의 요구 사양을 표로 정리하면 다음과 같다.

표 2. 기어를 포함한 모터의 설계사양

선정 기준	요구 조건
소비 전력	9.6V, 500mA 이하
회전 속도 (rpm)	764 rpm 이상
토크 ( $\tau_G$ )	3.5 mNm 이상
길이 (L)	38 mm 이하

여기서 회전 속도 및 토크의 경우 기어를 사용하면 곱하거나 나누어주어야 한다.

이 요구 조건에 대하여 실제 쓰일 모터 및 기어를 선정하였다.

Minimotor사의 2224(006SR)모터와 기어의 사양은 각각 표 3, 4와 같다.

표 3. 기어에 관한 사양

Rratio ( $N$ )	8
Efficiency ( $\eta_G$ )	0.73
Length with (L)	10 mm
Mass (m)	4 g

표 4. Minimotor 2224(006SR)

Nominal voltage [ $U_N$ ]	6	Volt
Terminal resistance [ $R$ ]	1.94	$\Omega$
Output power [ $P_{2MAX}$ ]	4.55	W
Efficiency [ $\eta_{MAX}$ ]	82	%
No-load speed [ $n_o$ ]	8200	rpm
No-load current [ $I_o$ ]	0.029	A
Stall torque [ $M_H$ ]	21.2	mNm
Friction torque [ $M_R$ ]	0.2	mNm
Speed constant [ $K_n$ ]	1380	rpm/V
Back-EMF constant [ $K_E$ ]	0.725	mv/rpm
Torque constant [ $K_T$ ]	6.92	mNm/A
Current constant [ $K_M$ ]	0.144	A/mNm
Slope of n-M curve [ $\Delta n / \Delta M$ ]	387	rpm/mNm
Rotor inductance [ $L$ ]	45	$\mu H$
Mechanical time constant [ $\tau_M$ ]	11	ms
Rotor inertia [ $J$ ]	2.7	$gcm^2$
Angular acceleration [ $\alpha_{MAX}$ ]	78	$10^3 rad/s^2$
Thermal resistance [ $R_{th1}/R_{th2}$ ]	5/20	K/W
Thermal time constance [ $\tau_{th1}/\tau_{th2}$ ]	6.8/440	s

요구 조건 중 토크를 기준으로 전류 및 속도를 계산해 보면 다음과 같다. 먼저  $\tau_M$  을 계산하면,

$$\tau_M = \frac{\tau_G}{N \times \eta_G} = \frac{3.53 \text{ (mNm)}}{8 \times 0.9} = 0.490 \text{ (mNm)}$$

이고, 이것은 모터의 스톱 토크 (모터를 멈추기 위해 필요한 토크) 보다 작은 동작 영역에 포함되므로 토크 조건은 만족한다고 할 수 있다.

이 때 공급해야 할 전류  $I$  는,

$$I = \frac{\tau_M}{k_T} + I_0 = \frac{0.490 \text{ (mNm)}}{6.92 \text{ (mNm/A)}} + 29 \text{ (mA)} = 99.81 \text{ (mA)}$$

이다. 두 개의 모터에 대해서는 58.14 mA 가 소비된다.

모터의 직선 속도에 앞서 먼저 회전 속도  $n$  을 계산한다.

$$n = \frac{U - I \times r}{k_{BEMF}} = \frac{6 \text{ (V)} - 99.81 \text{ (mA)} \times 1.94 \text{ (ohm)}}{0.725 \text{ (mV/rpm)}} = 8008.7 \text{ (rpm)}$$

이로부터 직선 최고속도  $v$  는

$$v = \frac{2 \times \pi \times R \times n}{60 \times N} = 262 \text{ (cm/s)}$$

가 된다. 모터 사양 상 무 부하 최고속도는 8200rpm이지만 현재 부하상태의 수치이므로 부하 시 최고속도는

2.6m/s가 됨을 계산할 수 있다. 이상에서 위 조합의 모터와 기어 및 양방향 엔코더는 요구 조건을 모두 만족함을 알 수 있다.

그러나 본 팀은 모터의 성능을 좀더 높이기 위해 충전지를 9.6V로 사용하므로 실제 모터의 회전속도  $n$  은 다음과 같다.

$$n = \frac{U - I \times r}{k_{BEMF}} = \frac{9.6 \text{ (V)} - 99.81 \text{ (mA)} \times 1.94 \text{ (ohm)}}{0.725 \text{ (mV/rpm)}} = 12974 \text{ (rpm)}$$

이로부터 직선 최고속도  $v$ 는

$$v = \frac{2 \times \pi \times R \times n}{60 \times N} = 425 \text{ (cm/s)}$$

가 된다. 하지만 모터정격을 넘어서는 속도이기 때문에 선형성이 깨어져서 이런 속도를 사용할 수는 없다. 따라서 모터가 선형영역에서 동작하도록 최고속도는 8000rpm으로 제한하였다.

### 2.2.2 모터 구동회로

DC 모터를 제어하는 방법에는 여러 가지가 있지만 본 팀은 PWM(Pulse width modulation)방법을 사용하였다. 즉 그림 2와 같이 모터에 인가되는 전압을 ON/OFF 시켜서 모터에 인가되는 평균전압을 변화시킨다.

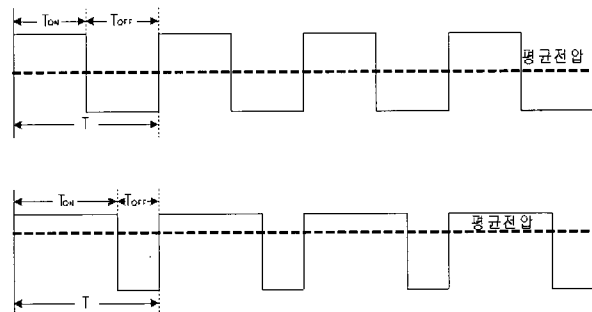


그림 2. PWM 제어 방법

ON시간이 길수록 모터에 가해지는 평균전압은 높아지게 된다. 즉, 같은 주기  $T$  동안에 펄스 폭(pulse width)을 조절하면 모터에 가해지는 평균전압이 그림 2처럼 변하게 된다. DC 모터의 속도는 인가되는 전압에 비례하므로 PWM 방법으로 한 주기 내의 펄스 폭을 조정한다면 속도 조절이 가능하게 된다. 일반적으로 전원 전압이  $V_{입력}[V]$  일 때 모터에 가해지는 평균전압은 다음과 같이 구해진다.

$$V_{\text{평균전압}} = V_{\text{입력}} \times \frac{T_{ON}}{T} [V]$$

PWM신호의 duty 비가 50%가 되면 결국 평균전압이 ON과 OFF 신호의 중간이 되어 모터는 정지해 있지만

모터에 토크는 걸려있게 된다. 마이크로 컨트롤러의 PWM신호를 이용하여 모터를 제어하기 위해서는 적절한 전류 증폭기가 필요하다. PWM 제어기에서 발생한 출력은 논리 회로의 출력으로 저 전류이기 때문에 모터에 직접 인가했을 때 모터가 충분한 토크 및 속도를 발생시킬 수 없다. 따라서 충분한 전류를 공급해 주어야 하는데 이러한 기능을 하는 것이 모터 드라이버다. 보통 전원을 모터의 양단에 직접 연결하면 충분한 전류로 모터를 구동할 수 있는데 이와 같은 원리로 H-브릿지 회로가 구성된다. 이것은 트랜지스터를 스위칭 모드에서 사용함으로써 전원에서 직접 모터로 전류를 흘려 보내준다. H-브릿지 회로에서의 트랜지스터는 스위치로 생각할 수 있다. 이런 구조를 가진 부품이 L298이다. L298은 내부에 2개의 H-브릿지 회로를 가지고 있어서 2개의 모터를 구동할 수 있다. 모터의 응답특성을 더욱 좋게 하려면 위의 H-브릿지 회로에 프리휠링 다이오드를 달아서 모터의 인덕턴스에 남아있는 전류를 제거해야 한다. L298이외에 L293이라는 칩도 브릿지 회로의 모터 구동드라이버이다. L298에 비해 모터 구동전류가 더 작지만 L298에 비해 그 크기가 작으므로 소형의 모터를 구동할 때 이용된다. 만약에 파워가 더 큰 모터를 구동하려면 파워 트랜지스터라는 소자를 이용하여 H-브릿지 회로를 구현해야 한다. 그 외에도 LMD18200같은 모터드라이버도 있는데 L289에 비해 성능은 좋으나 가격이 비싼 단점이 있다. 그러나 본 팀은 소형모터를 제어하기 때문에 L298로도 충분하기에 모터 드라이버로 L298을 사용하였다.

## 2.3 Main Board

### 2.3.1 CPU보드

로봇의 두뇌에 해당하는 부분이 바로 CPU 보드이다. 본 팀 로봇의 CPU보드에서 하는 일은 호스트로부터 날아오는 데이터를 RF통신 회로를 거쳐 받아들인다. 이렇게 읽은 데이터들을 종합하여 로봇에 속도 지령을 내리게 되고, 이에 따라 자신의 움직임을 결정하게 되어 스스로 제어를 하여 움직이게 하는 기능을 한다. 로봇의 가장 중요한 부분 중의 하나가 바로 CPU Board이다. 물론 호스트 중심 시스템인 경우에는 큰 기능은 필요하지 않겠지만, 로봇 중심 시스템을 구현하려 한다면 신중히 고려하여 설계해야 할 것이다. 따라서 CPU 선정에 있어서도 신중해야 한다. CPU의 종류에 따라서 로봇의 연산속도, 기능, 보드의 크기, 특징 등이 결정된다. 본 팀은 로봇의 크기가 걱정이 되어 가장 작게 만들 수 있으면서 알맞은 기능들만을 포함하고 있는 i80C196을 사용하기로 결정하였다. i80C196은 기능 면에서는 알맞지만 속도가 느리다는 단점이 치명적이다. 하지만 DSP와같이 고속의 CPU를 사용하려 하더라도 고가의 컴파일러 구입문제와 새로운 CPU를 사용하는데 따르는 어려움이 있기 때문에

i80C196KC를 사용하였다.

ROM은 256Kbits AT29C256, 120ns를 사용하였고, RAM은 Toshiba사 제품으로 256Kbits로 70ns를 사용하였다. 칩 셀렉터로는 GAL(16V8)을 사용하였고, 11.0592MHz 크리스탈을 사용하였다.

크리스탈을 20MHz를 쓰지 않고 11.0592MHz를 쓴 이유는 Motion Controller LM629의 클럭이 6MHz여서 80C196의 clock out단자에서 CPU 외부클럭의 2분주되어 나오는 클럭을 사용하기 위해서였다. 이렇게 하지 않으면 별도의 오실레이터를 회로에 부가해야하기 때문이다. 이는 PCB설계시 공간상의 문제에 매우 어려운 점으로 작용한다.

CPU보드 설계 시 다른 보드와의 입출력을 고려하여 적절한 커넥터를 적절한 위치에 빼는 것이 중요하고 만일의 경우를 위해 어드레스 버스와 데이터 버스는 커넥터로 빼 놓는 것이 좋다. 또한 안정된 보드를 설계하기 위해서는 각 칩의 VCC, GND 사이에 바이패스 캐패시터를 달아주어 전원 노이즈를 제거하고, 외부와의 전원이 입출력되는 단자에도 큰 캐패시터를 달아주는 것이 좋다. i80C196을 사용할 경우, 회로는 상당히 간단하다는 것을 알 수 있다.

## 2.4 Motion Controller LM629

### 2.4.1 LM629 (Precision Motion Controller)의 특징

- 32비트의 position, velocity, and acceleration register
- 프로그램이 가능한 16비트의 계수들을 가진 디지털 PID 필터
- 프로그램이 가능한 derivative sampling interval
- 8비트 sign-magnitude PWM output data
- 4체배 가능한 up-down 카운터 내장
- internal trapezoidal velocity profile generator
- Velocity, target position, filter parameters may be changed during motion
- Position and velocity modes of operation
- 실시간 프로그램 가능한 host interrupts
- 8비트 병렬 비동기 호스트 인터페이스
- incremental encoder interface with index pulse input
- LM629는 소수점 16비트까지 지원한다. (합이 32bit)
- 4N encoder 는 실제 position 과 direction, velocity 들을 넘겨준다. (encoder에서 넘겨준 속도가 원하는 속도가 되었을 때 그 위치값을 저장한 후 원하는 position에서 현재의 position을 빼주어서 감속을 할 position을 얻는다. 이유는 가속과 감속의 비율은 같기 때문이다.)

그림 3은 LM629를 이용한 기본적인 모터제어 시스템의 블록다이아그램이다.

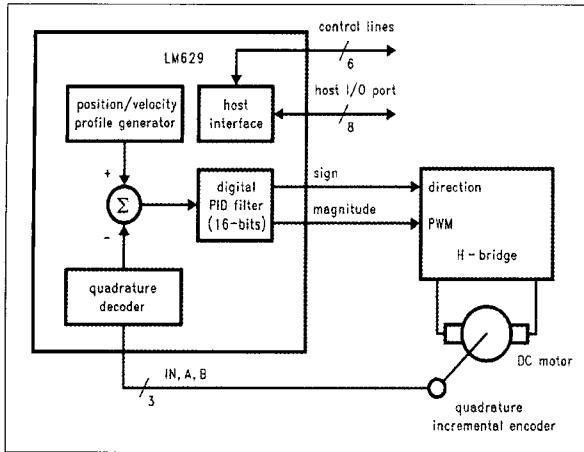


그림 3. LM629를 이용한 기본적인 모터제어 시스템

그림 4는 Quadrature Encoder의 출력펄스를 4체배한 테이블이다. LM629는 4체배가 가능한 16비트 up-down 카운터를 내장하고 있다.

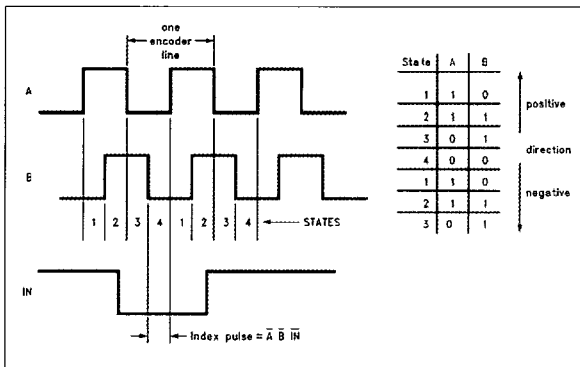


그림 4. Quadrature Encoder 출력신호와 Direction Decoder Table

그림 5는 LM629의 8비트 PWM 출력 파형이다.

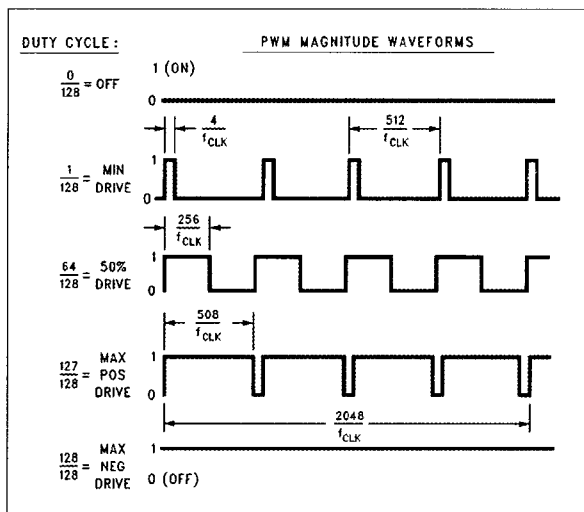


그림 5. LM629의 PWM 출력신호

그림 6은 LM629의 블록다이어그램으로 LM629의 디지털 PID 필터의 구성을 보여주고 있다.

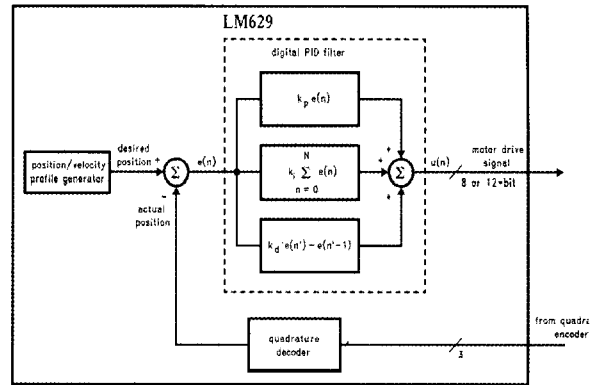


그림 6. LM629 블록다이어그램

마이크로 프로세서를 이용한 PID제어에서의 제어입력  $u$ 는 다음과 같다.

$$u(n) = K_p e(n) + K_i T \sum_{n=0}^N e(n) + K_d / T(s) [e(n) - e(n-1)]$$

모션 컨트롤러 LM629에서의 제어입력  $u$ 는 다음과 같다.

$$u(n) = k_p e(n) + k_i \sum_{n=0}^N e(n) + k_d [e(n) - e(n-1)]$$

LM629에서의 P 및 I 항의 샘플링 타임은  $2048/f_{CLK}$ 이며, D항의 경우 이것의 1배에서 255배까지 조절할 수 있다. 여기서  $f_{CLK}$ 은 LM629의 시스템 클럭 입력이다.

LM629에서의 속도지령 및 가속도 지령은 모두 양수여야 하며 이것은 매 샘플링 시간마다의 계수기 변화를 뜻한다. 이 값은  $2^{16}$ 으로 스케일되어 있으므로 16비트 이하의 비트는 소수의 계수기 값을 표시한다.

다음은 LM629에서의 속도지령을 RPM으로 지령하기 위한 변환이다.

$$VEL_{VAL} = 4 \times NLINE \times Ts \times \text{원하는 속도} [RPM] \times (1/60) \times 65536$$

### 2.4.2 LM629 (Precision Motion Controller)를 이용한 DC모터 제어

이 모션제어기는 엔코더로부터 들어오는 펄스를 누적하여 회전각 또는 각속도를 계산할 수 있게 해주는 계수기와 PWM발생기의 기능을 수행하며, 결론적으로 하드웨어적인 PID제어를 한다. 따라서 모터측에 연결된 엔코더의 A, B 펄스와 전원을 인가하면 원하는 모터의 속도 또는 위치를 제어하기 위한 PWM신호를 출력하고, 이것을 모터 드라이브를 통해 모터에 입력하면 원하는 모터 제어를 할 수 있다. LM629는 소형 마이크로 프로세서라

고도 볼 수 있다. A, B펄스 입력핀과 PWM출력핀은 물론 외부로부터 데이터를 주고 받을 수 있는 8비트 데이터 버스도 가지고 있다. 이 모션 제어기를 이용한 모터제어는 그림 7과 같은 형태의 블록 다이어그램을 갖는다. Main CPU는 모터제어를 위해 매 샘플링 시간마다 엔코더 정보를 이용해서 피드백을 갖지 않고, 단순히 원하는 회전 속도만을 지령하게 되며, 모션 제어기에서 매 샘플링 시간마다 엔코더 신호를 피드백 받아서 PID제어를 하게 된다.

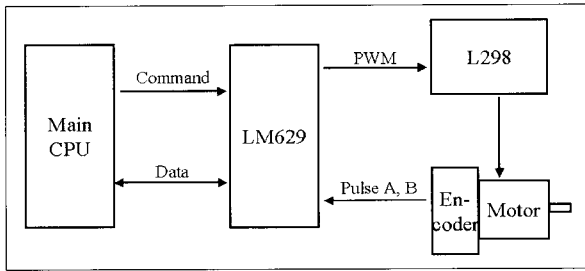


그림 7. LM629를 이용한 DC모터 제어 구성도

다음 그림은 LM629를 이용한 속도 및 위치제어 흐름도이다. 가속도와 속도 및 위치 지령은 동시에 줄 수 있으나 속도(또는 위치)를 변화시키고 싶을 때는 가속도는 유지한 채로 새로운 속도(또는 위치)만을 입력할 수 있다.

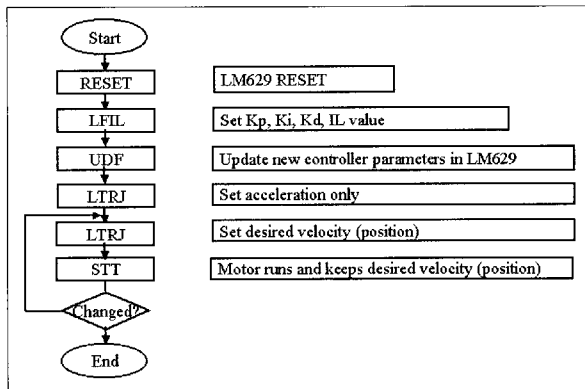


그림 8. LM629 제어 흐름도

## 2.5 무선 통신 시스템의 설계

로봇 축구 경기를 위해서는 컴퓨터에서 처리된 명령을 각 로봇에게 전해주기 위한 무선 통신이 필요하다. 로봇 축구에서 주로 쓰는 통신 방법으로는 RF(Radio Frequency) 통신과 적외선 통신이 있다. 본 팀은 현재 주로 사용되고 있는 상용 무선 통신모듈을 이용해서 호스트 컴퓨터와 로봇간의 통신을 한다. 디지털 통신을 위해 나온 제품이기에 높은 전송 속도와 신뢰성을 보장하지만 국내에서는 비싸다는 것과 주파수가 다양하게 제공되지 않는다는 단점이 있다. 본 팀에서 사용한 상용 통신 모듈은 Radiometrix

사의 BIM-418-F, BIM-433-F로서 40Kbps까지 전송이 가능하다.

### 2.5.1 통신 프로토콜

RF무선 통신 시스템에서 무엇보다 중요한 건 호스트 시스템에서 처리된 데이터를 정확하게 로봇 시스템에게 전달하는 것이다. 그리고 한 주파수를 이용하여 각각 3대의 로봇에게 서로 다른 데이터를 전송해야 하기 때문에 어느 특정한 통신 프로토콜을 가져야 한다.

즉, RS232 시리얼 데이터를 4.8~38.4Kbps 속도로 한 쌍의 상용 BIM433-F나 BIM418-F 모듈간에 반 이중식으로 사용할 수 있다. 데이터는 바이트 사이에 빈 공간(gap)이 없이 패킷화(packetised) 되어야 한다. 즉 데이터 설정시간이 3ms이상의 RX Settling time동안 55h나 aah의 프리앰블을 선행시키고 패킷화된 데이터를 전송시켜야 한다.

표 5의 총 11바이트에서 HEAD(0xfe)는 패킷의 시작을 알리는 데이터이고 모든 속도 데이터의 끝에 있는 checksum을 검사하여 수신된 데이터 패킷의 완전성을 확인한다.

표 5. 통신 프로토콜

데이터	이름	형태	크기	내용
command[0]	HEAD	Unsigned Char	1 Byte	0xfe
command[1]	vel_L (Robot1)	"	"	-128 ~ 127
command[2]	vel_R (Robot1)	"	"	"
command[3]	direction	"	"	0x10 or 0x1C
command[4]	checksum	"	"	"
command[5]	vel_L (Robot2)	"	"	-128 ~ 127
command[6]	vel_R (Robot2)	"	"	"
command[7]	direction	"	"	0x10 or 0x1C
command[8]	checksum	"	"	"
command[9]	vel_L (Robot3)	"	"	-128 ~ 127
command[10]	vel_R (Robot3)	"	"	"
command[11]	direction	"	"	0x10 or 0x1C
command[12]	checksum	"	"	"
command[13]	TAIL	"	"	0xfd

### 2.5.2 RF무선 통신 모듈을 이용한 송신기

실제 로봇 축구 경기에서는 로봇과 호스트 컴퓨터 사이에 양방향 통신을 하지 않는다. 대개의 경우 호스트 컴퓨터에서 로봇이 필요한 정보를 로봇에게 무선 통신으로 보내주기 때문에 송신기는 호스트 컴퓨터에만 필요하다. 호스트 컴퓨터에서 나온 명령 패킷은 RS232의 규격을 갖기 때문에 이를 CMOS/TTL에 넣을 수 있는 신호로 바꿔야 한다. 이를 위해서 MAX232를 사용하였다. 그림 10의 회로는 송신기를 나타낸다. 송신의 속도는 호스트 컴퓨터에서 프로그램으로 제어가 되므로 송신기는 비교적 간단하게 설계가 된다. 아래의 회로는 송신기를 설계

한 것인데, Schmitt trigger not gate 인 74HC14를 사용하여 1비트의 제어 신호로 RXD, TXD 핀을 선택할 수 있고 송신의 신뢰도를 높일 수 있도록 설계하였다.

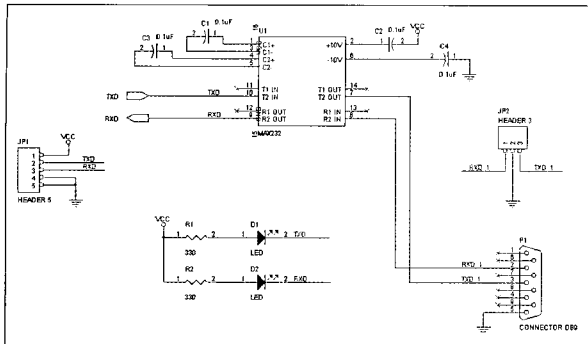


그림 9. RS-232C 통신 설계 회로도

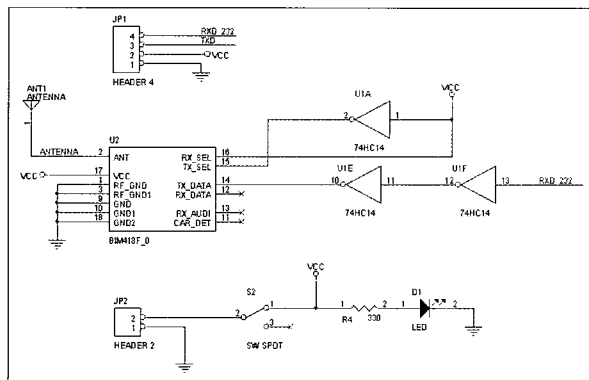


그림 10. RF 무선 송신기 설계 회로도

### 2.5.3 RF무선 통신 모듈을 이용한 수신기

수신기는 로봇의 CPU와 연결하여 설계하였다. 접퍼를 이용해 RXD, TXD 핀을 선택할 수 있고, 송신의 신뢰도를 높이기 위해 반전되어 송신된 데이터를 다시 반전시키도록 송신기와 마찬가지로 74HC14를 사용하였다.

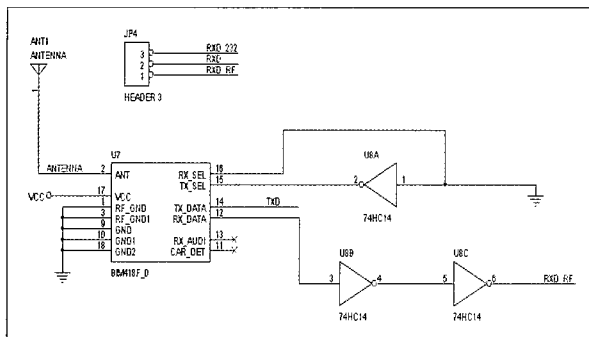


그림 11. RF 무선 수신기 설계 회로도

### 2.5.4 로봇 테스트 프로그램(RF 무선 통신)

그림 12는 Visual C++로 프로그래밍한 로봇 테스트 프

로그램이다. PC에서 RF 모듈을 통해 무선으로 로봇에 속도를 전송시키는 프로그램이다. 상용 RF Module인 BIM-418을 이용하여 38400bps의 전송속도로 무선 통신이 가능하게 하였다. -128 ~ 127까지 속도를 전송할 수 있다. 여기서 속도가 -128 ~ 127인 이유는 속도 지령과 방향을 PWM의 값으로 결정해 주기 위해서이다. -128일 때는 역방향으로 최대속도를 출력하고 127일 때는 정 방향으로 최대속도를 출력한다. 물론 속도지령을 RPM으로도 사용 가능하다.

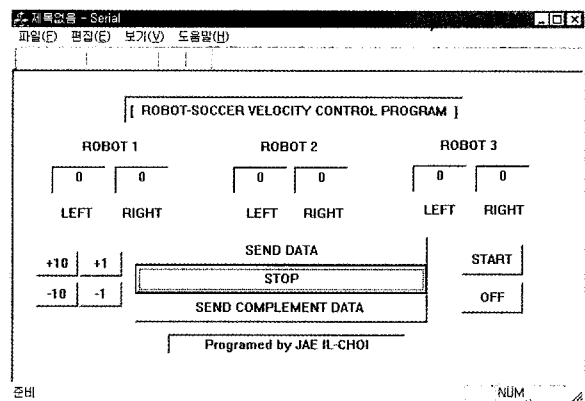


그림 12. 로봇테스트 프로그램

## 3. 결 론

### 3.1 PID 제어 분석

본 장에서는 완성된 축구로봇으로 DC모터 속도제어와 실제 로봇축구 시스템에 적용한 실험들에 대한 결과 및 결과검토에 대해 다룬다.

본 논문에서는 DC모터의 제어를 위해 PID제어기를 사용하였고, PID제어기의 게인 설정은 matlab 시뮬레이션을 바탕으로 실험을 통해 조정하였으며, 속도검출 방법으로 M method(펄스카운트 방식)를 사용하였다. M method는 단위시간당 엔코더에서 발생하는 펄스 수를 카운터로 계수하는 방법으로 속도는 카운터의 계수값에 비례하게 된다. 이 방법은 고속에서는 계수값이 크므로 측정 정밀성이 우수하지만 저속일 경우에는 샘플링 시간이 엔코더 펄스주기의 정수배가 아니므로 발생하는 계수 오차 때문에 측정정밀도가 낮아진다. 이를 해결하려면 1회전당 출력펄스수가 많은 엔코더를 사용하던가 샘플링 시간을 길게 하여야 하는데, 전자의 경우에는 경제적인 부담이 증가되고 후자의 경우에는 제어기의 성능이 저하될 수 있다. 이를 보완하는 방법으로 M/T method가 있지만 구현하기가 그리 쉽지 않고, 축구로봇은 거의 대부분 고속에서 동작하며, PID제어를 하는 motion controller LM629에서도 속도검출 방법으로 M method를 사용하고 있기 때문에 본 논문에서도 속도검출 방법으로 M method

를 사용하였다. M method에 측정된 카운터 계수를 RPM으로 환산하면 다음과 같다.

$$\text{속도} [RPM] = \frac{\text{카운트된 펄스수}}{\text{샘플링 시간}} \div 1\text{회전당엔코더출력펄스수} \times 60$$

이와 같이 모터의 속도는 샘플링 시간과 1회전당 엔코더 출력펄스 수에 영향을 받는다. 따라서 속도측정의 정밀성을 높이기 위해서는 위에서 거론한바와 같이 엔코더의 출력펄스 수를 높여야 하고 샘플링 시간은 늘여야 한다. 이런 이유로 분해능을 높이기 위해 엔코더 출력 펄스를 512PPR을 2048PPR로 4배배하여 사용하였고, 샘플링 시간은 1.48ms로 설정하였다. 샘플링 시간을 더 늘리지 않은 이유는 축구로봇이 주로 고속에서 동작함에 따라 짧은 샘플링 타임에도 충분한 카운터 계수를 얻을 수 있고, 또한 샘플링 시간의 증가로 인한 제어기의 성능이 저하될 것을 염려한 것이 가장 큰 이유였다.

이제 위의 수식에 따라 속도 측정의 분해능을 계산해보자. 속도측정 분해능이란 엔코더에서 출력되는 1개의 펄스로 측정할 수 있는 최소한의 속도를 의미한다.

속도측정 분해능을 계산해보면 다음과 같다.

$$\text{속도측정분해능} = \frac{1}{0.00148} \div 2048 \times 60 \approx 20 [RPM]$$

따라서 샘플링 시간당 1개의 펄스가 카운트되면 20 [RPM]이 되고 2개의 펄스가 카운트되면 40[RPM]이 된다. 그러나 속도측정 분해능이 대략 20[RPM]이라고 하더라도 분해능 이하의 속도를 제어할 수 없다는 뜻은 아니다. M method에 의해 측정된 속도는 일정시간(샘플링 시간)동안 계수한 펄스로부터 얻어지므로 엄밀히 말하자면 순시적인 속도가 아니라 측정주기 내에서의 평균속도가 되기 때문이다. 따라서 만약 속도측정 분해능 이하의 속도지령, 즉 15RPM의 속도지령을 주었다면 모터는 0[RPM]과 20[RPM]의 속도로 번갈아 동작할 것이다. 정확히 말하자면 모터의 회전시간 중 25%동안은 정지해 있고, 나머지 75%동안은 20[RPM]의 속도로 동작할 것이다. 이에 따라 모터는 평균적으로 대략 15[RPM]근처의 속도로 동작할 것이다. 그 이유는 속도제어는 1.48ms마다 한 번씩 수행하는데 반해 모터의 기계적인 시정수는 1.48ms 보다는 훨씬 크기 때문이다. 따라서 엄밀히 말해 20 [RPM]과 0[RPM]의 속도로 번갈아 제어를 하고는 있지만 모터자체의 기계적 시정수가 크기 때문에 이러한 속도 지령을 따라 정확히 동작할 수는 없다. 그렇기 때문에 평균적으로 볼 때 속도측정 분해능 이하의 속도지령에 대해서도 어느 정도의 속도변동률을 가지면서 제어가 가능하다. 만약 이를 M method를 사용하여 속도측정을 하지 않고 비접촉식 RPM미터를 가지고 외부에서 속도를 측정한다면 15[RPM]근처의 속도가 측정될 것이다. 왜냐하면 비접촉식 RPM미터는 평균속도를 측정하고 있기 때문이다.

종합하여 설명하자면 PID 제어가 정확히 사용되었다면 정상상태 오차는 없을 것이다. 왜냐하면 네거티브 피드백 시스템에서 오차 보상기로 PI나 PID제어와 같은 것을 사용하면 기준값과 피드백된 값을 계속 비교하여 제어 입력이 + 또는 -로 변화되면서 출력값을 끊임없이 기준값과 동일해지도록 제어하기 때문이다. 그러나, 실제로 출력값이 기준값과 동일해지면 오차(제어입력)가 0으로 되므로 이런 상태가 지속될 수는 없다. 더구나, 위에서 보듯이 속도측정의 분해능 때문에 피드백된 값이 기준값과 비교될 때 그 비교결과가 정확해지지도 않는다. 그렇지만 이렇게 되는 과정에서 결과적으로 출력은 기준값의 아래위로 약간의 범위에서 진동하면서 레귤레이션 된다. 이것을 steady state로 보면 출력에는 정상상태 오차가 없는 것이다. 다만, 속도 출력값이 기준값을 중심으로 아래위로 변동하기 때문에 속도변동률은 있게 된다. 이 속도변동률은 과도응답 특성과 함께 제어기의 우수성을 판단하는 중요한 요소가 되기 때문에 이를 최소화 해야한다. 속도변동률이 좋아지려면 속도센서의 분해능이 높아야 하는 것은 물론이고, 제어주기 또는 샘플링 시간이 길어져야 하며, PID 제어기의 이득이 적합하게 설정되어야 하는 것이다.

### 3.2 실험결과 및 결과고찰

다음의 결과 파형들은 M method를 사용하여 측정된 데이터를 RPM으로 환산하여 MATLAB을 이용해 그래프로 표시한 것이다.

#### 3.2.1 P(비례)제어 결과

비례 제어는 페루프 시스템의 가장 기본적인고 단순한 구조로 Sampling time 마다 응답에 대한 에러에 비례한 보정 값을 더해줌으로써 시스템의 응답곡선이 원하는 위치나 속도에 수렴할 수 있도록 설계한 구조이다. 비례 제어의 비례 게인(Kp)을 높게 설정하면 정상상태 오차를 줄인 수는 있지만 너무 크면 오버슈트와 함께 진동이 발생된다. 따라서 비례 게인을 무조건 높일 수만은 없다. 반대로 비례 게인은 너무 작게 설정하면 정상상태 오차가 커지게 된다. 이런 이유로 비례 게인은 적절히 설정해야 한다.

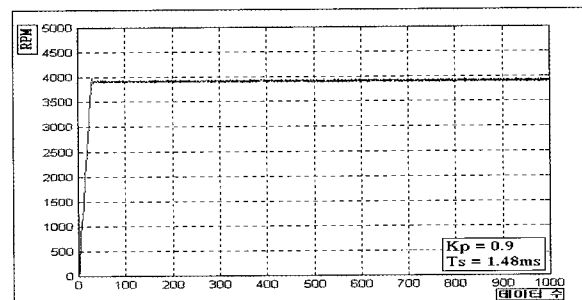


그림 13. KP = 0.9일때 P제어 결과



그림 13은 4000[RPM]의 속도지령 시 비례 제어만을 사용했을 때의 결과이다. 실험결과 Kp가 0.9이상이면 지령속도에서 오실레이션하기 시작하여 Kp가 증가함에 따라 과도한 오실레이션 현상이 발생하였다. 따라서 그림 13과 같이 정상상태 오차를 최소화시킬 수 있는 비례 계인으로 Kp = 0.9로 결정하였다. 그림에서 보듯이 P제어만으로는 정상상태에 도달할 수 없음을 확인할 수 있으며 정상상태 오차는 대략 80[RPM]정도가 됨을 확인할 수 있다.

3.2.2 PI(비례 적분)제어 결과

PI제어는 비례제어에 적분 항을 추가한 것인데 이는 잔류 편차 때문에 생기는 미세한 정상상태 오차를 줄이기 위해서이다. 실제로 모터를 돌려보면 금방 서질 않고 ‘삐’하고 소리가 나게 되는데 이것이 바로 잔류편차가 남아 있기 때문에 아주 미세하게 떨고 있는 것이다. 그러나 I계인을 너무 크게 설정하면 처음에 모터가 동작하며 보정을 할때 심하게 진동하는 현상이 발생한다. 따라서 P계인과 I계인을 적당히 조절하여 설정해야한다.

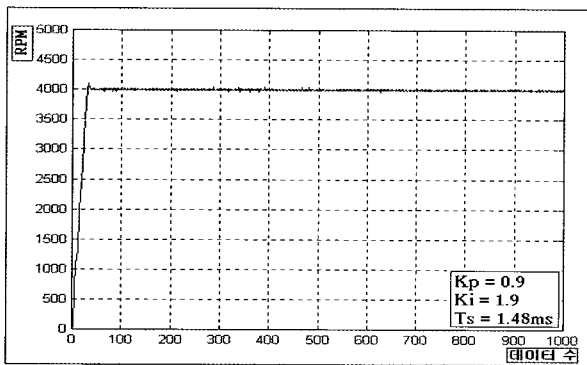


그림 14. Kp = 0.9 와 Ki = 1.9 일 때 PI제어 결과

그림 14는 비례 계인(Kp) = 0.9, 적분 계인(Ki) = 1.9일 때의 PI제어 결과이다. 그림에서 보는바와 같이 약간의 오버슈트가 발생되며 정상상태에 도달함을 알 수 있다. 위의 그림은 1.48ms마다 속도를 검출하여 그래프로 출력한 것이다. 따라서 정상상태 도달시간은 대략  $30 \times 1.48ms = 44.4ms$ 이다.

3.2.3 PID(비례 적분 미분)제어 결과

PID제어는 말 그대로 P제어, PI제어, PD제어등을 다 합쳐놓은 것이다. PI제어에 미분항을 추가함으로써 에러 추정치를 알 수 있게 된다. 이 에러 추정치에 의해서 시스템의 응답은 선형적으로 가려고 하는 특성을 가지게 되는데, 미분 항을 추가함으로써 얼마만큼의 과도한 오실레이션 현상은 줄일 수 있게 되는 것이다. 즉, Kd는 선형특성추정의 속도를 조절하는 기능을 하는 것이다. 그러나 만약 응답곡선이 선형적이라면 Kd는 무의미해진다.

또한 PID계인을 조정할 때에 Kd를 무조건 크게 설정한다고 해서 오실레이션 현상이 감소되는 것이 아니라 구성된 시스템에서의 Sampling Interval과 모터의 현재 가속도 및 속도 그리고 모터의 관성을 고려해서 조정해야 된다.

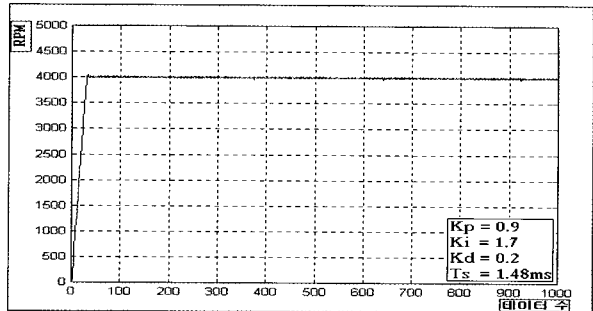


그림 15. Kp = 0.9 와 Ki = 1.7, Kd = 0.2 일 때 PID제어 결과

그림 15를 보면 정상상태 오차는 0이며, 오버슈트 및 오실레이션 현상도 거의 없음을 알 수 있다. 다만 속도변동률이  $\pm 20[RPM]$ 으로 나타나고 있는데 이는 속도 측정 분해능이 20[RPM]인데서 비롯된 문제이며, 속도변동률을 개선하기 위해서는 속도센서의 분해능이 높아야 하는 것은 물론이고, 제어주기 또는 샘플링 시간이 짧아야 하며, PID 제어기의 이득이 적합하게 설정되어야 한다. 따라서 속도변동률을 줄이기 위해 샘플링 시간 3.33ms로 늘려보았다.

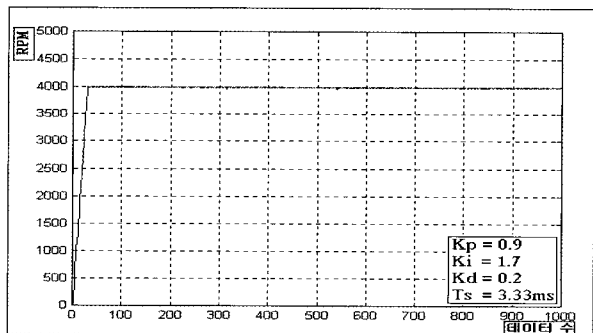


그림 16. 샘플링 시간에 따른 속도변동률의 변화

샘플링 시간이 3.33ms일 때의 속도측정 분해능을 계산해보면 다음과 같다.

$$\text{속도측정분해능} = \frac{1}{0.00333} \div 2048 \times 60 \approx 9 [RPM]$$

위의 결과 파형을 살펴보면 샘플링 시간이 1.48ms일 때보다 속도변동률이 적어진 것을 확인할 수 있으며, 이때의 속도변동률은 약 9[RPM]이 됨을 확인할 수 있다. 이때 주의할 점은 속도변동률을 줄이기 위해 샘플링 시간을 과도하게 증가시키면 제어기의 성능이 저하되므로 적당히 조절해야 한다.



축구로봇에서 DC모터의 속도제어에 있어서 원하는 속도에 정확히 추종하는 것도 중요하지만 속도의 변화에 빠르게 추종할 수 있는 것이 오히려 더욱 중요하다. 로봇축구 경기의 특성상 상대편의 로봇보다 빠르고 정확히 움직일수록 공에 접근하는 시간이 늘어나게 되어 좋은 결과를 가져올 수 있다. 또한 공의 움직임이 예측 가능한 경우가 많기 때문에 공의 움직임에 로봇이 빠르고 정확하게 속도를 변화시킬 수 있어야만 한다.

그림 17은 4000[rpm]과 3000[rpm]의 속도를 반복하여 지령하였을 때 동적 응답 특성을 보여주고 있다.

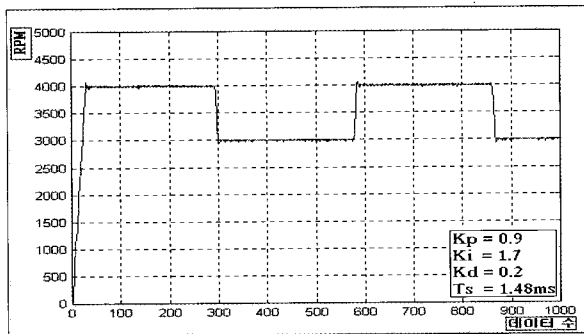


그림 17. 반복적인 속도지령에 대한 동적 응답

그림 17에서 보여지듯이 반복적인 속도지령에 대해서도 지령속도를 잘 추종하고 있음을 알 수 있다.

### 3.3 결론

Motion Controller LM629를 이용하여 DC모터 PID Control을 이용한 축구로봇을 완성하였다. 모터제어가 정확히 되는지 여부를 판단하기 위해 엔코더에서 출력되는 데이터를 RPM으로 환산한 뒤 시리얼통신을 이용하여 PC로 전송하여 그래프화하는 방식으로 확인하였는데 실험결과 20[RPM]정도의 속도변동률이 발생하기는 하였지만 축구로봇의 구동에는 거의 영향이 없어 더 이상 속도변동률을 줄이지는 않았다. 물론 샘플링 시간을 더욱 증가시키면 속도변동률을 줄일 수는 있으나 이는 제어기

의 성능저하를 유발시키기 때문에 적당한 샘플링 시간으로 1.48ms로 설정하였고, 이에 따라 속도변동률은 대략 20[RPM]이 되었다. 한가지 문제점은 속도검출 방법을 M method를 사용하였기 때문에 정확도가 떨어지는 문제점이 발생되었는데 이를 해결하기 위한 방법으로 M/T method가 있는 점에도 불구하고 축구로봇이 주로 고속 영역에서 동작하는 이유로 M/T method를 사용하지 않은 점이 미흡했다.

앞으로도 축구로봇을 구동시키기 위한 호스트 프로그램 및 알고리즘 개발 등의 많은 과제가 있다. 현재 만족할 만한 성능의 축구로봇의 개발을 마친 상태여서 앞으로는 전체적인 로봇축구 시스템의 성능향상을 위해서 비전 시스템의 성능 향상, 공격과 수비 알고리즘의 보완, 새로운 전술 알고리즘의 개발 등에 주력할 계획이다.

### 참고 문헌

1. 김종환, "로봇 축구 시스템", <http://vivladi.kaist.ac.kr/>, 대한 로봇 축구 협회 배포 자료, 1999
2. 김종환 심현식 이선우 김홍수, "MIROSOT에 대한 소개", <http://vivladi.kaist.ac.kr/MIROSOT>, 1996
3. "서보모터 제어이론과 실습", 아경 산업 자동화연구소 편저, 1999.
4. "UHF FM Transceiver Manual", <http://www.radiometrix.co.uk/>, 1999
5. 이상락 박귀태, "C언어로 쉽게 쓰는 80C196KC", 대영사, 1994
6. 차영배, "80196을 이용한 MICRO MOUSE", 다다미디어, 1999
7. 윤덕용, "어셈블리와 C언어로 익히는 80C196KC마스터", Ohm사, 2000
8. 이상엽, "Visual C++ Programming Bible ver.6.x", 영진출판사, 1999
9. "LM628/LM629 Precision Motion Controller Manual", <http://www.national.com/>, 1995