

# 아이폰을 이용한 체감형 인터페이스 기반 이동 로봇 제어

## Control of a Mobile Robot Based on a Tangible Interface using iPhone

정하민 · 김동헌

Hahmin Jung and Dong Hun Kim

경남대학교 전기공학과

### 요 약

본 논문에서는 이동로봇의 원격 제어를 위해 조이스틱 혹은 버튼식 원격제어 방법 대신 아이폰을 사용한 체감형 인터페이스 기반의 원격 제어를 제시한다. 제안된 연구는 아이폰의 가속도센서를 사용하여 로봇을 제어하기 때문에 아이폰을 자동차의 핸들로 응용하여 무선으로 로봇 제어가 가능하다. 아이폰을 이용하기 위해 아이폰의 가속도 센서를 측정 및 분석하고 디지털 필터를 사용하여 가속도 센서 값을 안정화 시켰다. 이동 로봇과 아이폰간의 통신을 위해 블루투스 방법이 채택되었다. 본 연구에서는 네가지의 통신 방법을 제시하였고 이 중 가장 적절한 통신방법을 택하였다. 실험 결과로서 제안된 아이폰 기반의 체감형 인터페이스는 쉽고 편리하게 이동 로봇의 원격 제어를 가능하게 해주었다.

**키워드** : 아이폰, 블루투스 통신, 가속도 센서, 로봇, 체감형 인터페이스

### Abstract

In the study, a tangible interface using iPhone is proposed to control a mobile robot, instead of remote control by a joystick or buttons. The robot is controlled by iPhone like a handle bar, since acceleration sensors of iPhone are used in the proposed method. The sensors measure the angles changed on the xyz coordinates of iPhone. And their sensor values are stabilized by digital filters. Bluetooth is chosen for communication between a mobile robot and iPhone. In this paper, four type methods are considered and one of the methods is selected for remote control of a mobile robot. Experimental results show that the robot is easily and conveniently controlled by the tangible interface based on iPhone.

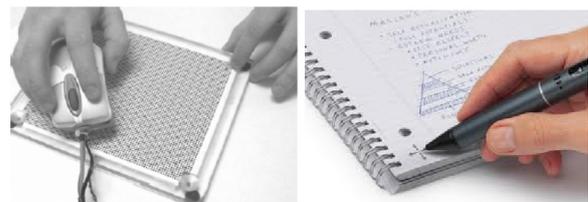
**Key Words** : iPhone, Bluetooth communication, acceleration sensor, robot, tangible interface

## 1. 서 론

게임용, 산업용, 그리고 의료용과 같은 IT기기용 인터페이스사용시 대부분 화면, 소리 등 시각과 청각에 기반을 두고 키보드 마우스나 조이스틱등을 이용한 통상적인 인터페이스를 사용한다[2]. 로봇 제어 역시 단순히 조이스틱을 사용한 연구[1,3]도 있고 게임용처럼 향상된 인터페이스 방법으로 발전된 연구도 있다. 최근에는 햅틱 디바이스를 이용한 로봇 제어[4], 인간의 음성 인식을 이용한 로봇 제어[5], 유저의 뇌파 신호를 이용한 로봇 제어[6] 등 다양하게 변화된 로봇 제어 인터페이스들이 제시 되었다. 이와 유사하게 유비쿼터스(Ubiquitous) 컴퓨팅 환경의 보급과 발전에 의해 모바일 기기를 제어 하는 방법도 단순한 키패드 입력 방식에서 스크롤 버튼의 입력 방법이나 스크린 터치 방식의 입력 방법으로 발전되었다. 즉 모바일 기기의 소형화에 맞추어 센서 입력 방법으로 인터페이스를 확장 하였다. 그 예로서 모바일 기기중 Apple사의 아이폰(iPhone)은 여러 종류의 센서를 활용한 혁신적인 인터페이스를 갖추고 있다.

센서의 활용은 그림 1. (a)처럼 입력된 영상을 마우스로

활용한 비디오 마우스(Video mouse)가 있다. 이 마우스는 특정 패턴이 그려진 패드 위에서 움직임으로써 2차원 좌표를 얻으며, 기울기와 z값을 얻을 수 있다[7]. 그림 1. (b)는 상용화 제품으로 소형의 적외선 카메라를 사용한 Livescribe사의 스마트 펜 이다[8]. 미세하게 점이 찍힌 노트를 인식하며 필기와 음성을 동시에 저장하여 쉽고 빠르게 사용자의 메모를 도와준다. 센서가 사용된 이러한 연구와 제품들은 사용자의 의사를 즉흥적으로 반영하기 때문에 사용자에게 편리성을 더해준다. 이러한 사용자의 편리성을 강조하는 센서 시스템은 모바일 기기나 게임기에 잘 응용되고 있다. 상용화된 제품들은 차별화된 인터페이스 덕분에 사업에서 성공을 거두고 있다. 이러한 결과들은 사용자 편의에 맞춰진 인터페이스 기술이 제품 차별화에 많은 기여를 한다는 것을 보여 준다.



(a) 비디오 마우스

(b) 스마트 펜

그림 1. 체감형 인터페이스의 응용의 예  
Fig. 1. Example of tangible interface

접수일자 : 2011년 1월 14일

완료일자 : 2011년 5월 30일

본 연구는 2010학년도 경남대학교 학술연구 장려금 지원으로 이루어졌음.

로봇의 제어에 모바일 기기를 활용한 연구[9]는 홈 네트워크를 통해 로봇을 원격으로 제어한다. 즉 PDA (Personal Digital Assistants)와 유사한 포터블 기기들은 인간과 주변 환경간의 인터페이스를 위한 단말기로서 역할이 가능하다. 기존의 포터블 기기를 이용하면 로봇의 제어를 위해 별도의 하드웨어가 필요가 없고 응용에 더 초점을 맞출 수 있다. 본 연구에서는 [10-12] 처럼 센서활용 하드웨어 제작 없이 성능 높은 스마트 폰을 이용한다. [10]은 아이폰과 닌텐도 Wii의 인터페이스를 응용한 로봇제어 기술들을 평가 및 제시한다. [11]은 인간과 상호 교감 하는 로봇의 제어를 위해 아이폰을 사용하였고 [12]는 네트워크에 저장된 환경정보를 아이팟(iPod, apple)의 무선랜을 통해 접속하여 로봇을 제어하는데 IT단말기를 사용하였다. 본 연구 역시 이러한 기존의 연구들 처럼 IT단말기인 아이폰의 가속도 센서를 활용한 인터페이스 확장으로 로봇을 제어한다. IT기기 중 정전식 터치 방식을 사용하고 계속 버전업된 SDK(Software Development Kit)를 제공하는 Apple사의 아이폰은 다른 스마트기기에 비해 하드웨어 성능이 낮은 반면 최적의 설계와 OS선택으로 사용자들에게 인기가 많다.

본 논문에서 개발하려는 시스템은 [9,12]의 로봇제어를 위해 무선랜 환경에서 사용가능한 WiFi 방식의 통신 대신 실내, 실외 전 지역에서 사용가능한 블루투스 통신을 사용한다. 사람이 앉아서 움직일 수 있는 커다란 이동 로봇이 제어 대상으로 사용되었고, 아이폰의 가속도 센서를 사용하여 아이폰을 무선 핸들처럼 사용 한다. 무선통신을 통해 로봇을 제어할 때 사용자의 제어기 조작은 빠르게 반영 되어야 하기 때문에 적절한 무선통신 제어 방법을 찾는 것이 중요하다. 따라서 본 논문에서는 다양한 패킷을 가지는 네 가지 방법들을 제시하고, 아이폰과 로봇간에 제어시간이 짧은 방법을 찾기 위해 결과들을 분석하려고 한다.

본 논문은 다음과 같이 구성되어 있다. 2장에서 아이폰의 제어 대상인 차등 구동 로봇 제어에 대해 언급 하고 아이폰의 가속도 센서에 대해 설명한다. 3장에서는 아이폰과 PC 간의 통신을 위한 개발 환경 구축 및 로봇과 아이폰의 통신 방법에 대해 제시한다. 4장은 아이폰으로 로봇을 제어하는 실험 내용과 결과를 보여주고, 5장에는 본 논문의 결론을 제시한다.

## 2. 가속도 센서 값과 로봇의 제어

아이폰은 내장된 가속도 센서를 사용하여 아이폰이 정지 상태인지 이동 중인지를 판단할 수 있다. 아이폰은 가속도 센서를 사용하여 자동회전을 처리하고 여러 게임들의 객체를 제어할 때 센서의 값을 사용한다. 따라서 가속도 센서는 흔들기나 갑작스런 움직임을 인식할 때도 사용될 수 있다. 본 논문에서는 아이폰을 컨트롤러로 사용 하여 원격으로 차등 구동 로봇을 제어한다.

### 2.1 차등 구동 로봇 모델

차등 구동 로봇 모델은 그림 2처럼 구성된다. 차등 구동 로봇모델은 3개의 벡터 로 표현 된다. 여기서 3개의 벡터는 현재 위치와 기준 프레임에 대한 방위각이다. 점에서 로봇의 바퀴가 지면에 대해 미끄러짐이 없다고 가정할 때 수식은 다음과 같다[13].

$$\dot{x}\sin\phi - \dot{y}\cos\phi = 0 \tag{1}$$

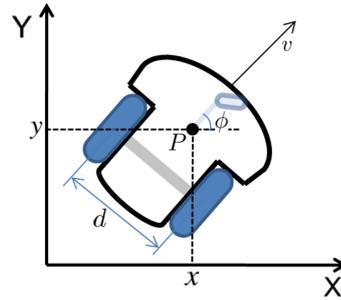


그림 2. 차등 구동 로봇 모델  
Fig. 2. Differential drive robot model

여기서, P는 로봇 무게 중심이고 φ는 로봇의 회전각을 의미 한다. 이 자연적 제한은 적분 불가능하다. 즉 로봇 모델은 논 홀로노믹 제한이 있다. 다시 말해 로봇은 곧 바로 한번에 x 또는 y 방향으로 이동할 수 없다. 로봇의 선속도 v와 각속도 ω는 다음과 같다[13].

$$\dot{x}\cos\phi + \dot{y}\sin\phi = v, \dot{\phi} = \omega \tag{2}$$

따라서, 정리하면 다음과 같다.

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} \cos\phi & 0 \\ \sin\phi & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \tag{3}$$

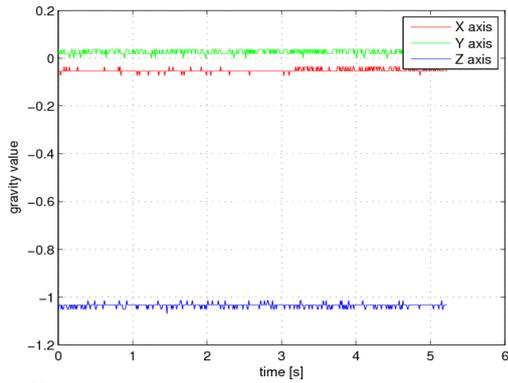
v와 ω는 다음과 같이 구해진다.

$$v = \frac{p_r + p_l}{2}, \omega = \frac{p_r - p_l}{d} \tag{4}$$

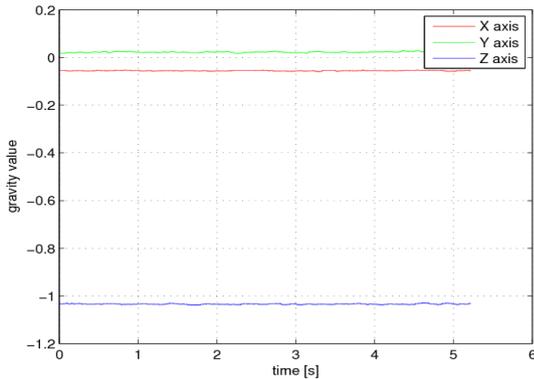
여기서, p<sub>r</sub>과 p<sub>l</sub>은 각각 왼쪽 오른쪽 모터의 pwm값이고 d는 로봇 몸체의 반경이다.

### 2.2 가속도 센서 값 측정

가속도 센서는 중력이 작용하는 범위안에서 측정된 가속도 값을 제공한다. 그래서 가속도 센서의 반환 값이 1.0이면 특정한 방향으로 1g(gravity)의 힘이 작용하고 있음을 나타낸다. 만약 아이폰이 세로 방향으로 세워져 있다면 가속도 센서는 y축 방향으로 1g의 힘이 작용하는 중이며 -1 또는 1의 값을 측정 할 수 있다. 아이폰이 비스듬한 모양으로 정지해 있다면 아이폰의 기울기에 따라 1g의 힘은 분산되어 작용하고 측정 값 또한 분산되었기 때문에 1보다 적은 값을 얻게 된다. 만약 아이폰을 흔들게 된다면 센서값은 1g보다 더 크게 값이 측정될 것이다. 그림 3은 아이폰을 바닥에 놓힌 상태에서 x, y, z축의 계측 결과를 보여준다. 아이폰관련 서적에 의하면 이 상황에서 가속도 센서의 이상적인 값은 x=0, y=0, z=-1이다. 하지만 실제 측정값과는 오차가 있음을 알 수 있다. x, y, z축의 평균값은 각각 -0.0518, 0.0262, -1.0349 이고 값의 범위는 각각 -0.0724~ -0.0362, 0~0.0362, -1.0686~ -1.0143이다. 그림 3. (a)는 필터를 사용하기 전 아이폰에서 측정한 값을 보여준다. 그림 3. (a)-2는 x축값의 분포를 보여준다. 그래프에서 알 수 있듯이 값의 변화가 갑자기 크게 일어나기 때문에 로봇의 입력 값으로 사용하기에 부적합하다. 이를 간단하게 디지털 저역통과 필터를 사용하여 오차와 값의 변화를 줄였다. 디지털 저역통과 필터는 다음식과 같이 구성된다[14]. 로봇의 x, y좌표는 가속도 센서의 y, z축의 값에 대응된다.



(a)필터를 사용하지 않은 경우(계측결과, x축 가속도 값의 분포)



(b)필터를 사용한 경우(계측결과, x축 가속도 값의 분포)  
그림 3. 아이폰을 바닥에 놓힌 상태에서의 가속도 센서 측정

Fig. 3. Measuring acceleration sensor of iPhone

$$\tilde{A} = \alpha A + (1 - \alpha)\tilde{A} \quad (5)$$

여기서  $\tilde{A}$ 는 추정된 가속도 센서 값이고  $\tilde{A} = [\tilde{x}_a, \tilde{y}_a, \tilde{z}_a]$ 이

다. 그리고  $A$ 는 측정된 가속도 센서 값이고  $A = [x_a, y_a, z_a]$ 이다.  $\alpha$ 는 가중치 상수이고  $\alpha = s_t / (t_c + s_t)$ 이다. 여기서  $s_t$ 는 샘플링 타임이고  $t_c$ 는 시상수이다. 본 논문에서 디지털 필터를 사용하기 위한 파라미터 값으로  $s_t$ 는 0.01이고  $t_c$ 는 0.2로 설정하였다. 필터를 사용 후 x, y, z축의 평균값은 각각 -0.0554, 0.0224, -1.0340 이고 값의 범위는 각각 -0.0600~0.0524, 0.0156~0.0296, -1.0387~-1.0284 이다. 그림 3. (b)는 필터를 사용한 후 아이폰에서 측정된 값을 보여준다. 그림 3 (b)-2 그림은 x축값의 분포를 보여준다. 그래프는 x축 값이 -0.055근처에 균일하게 분포함을 보여준다.

### 3. 로봇의 제어

차등 구동 로봇은 아이폰의 가속도 축 변화에 의해 그림 4처럼 제어된다. 본 연구에서는 차등 구동 로봇의 원격 제어를 위해 아이폰에 내장된 가속도 센서 두축인 y, z축을 사용한다. 아이폰의 화면을 앞으로 숙이게 되면 가속도센서의 z축 값이 -1값으로 점점 변화되고 로봇은 정면으로 이동한다. 그리고 아이폰을 왼쪽으로 세우게 되면 최대 1값으로 변화되고 오른쪽으로 세우게 되면 최대 -1값으로 변한다.

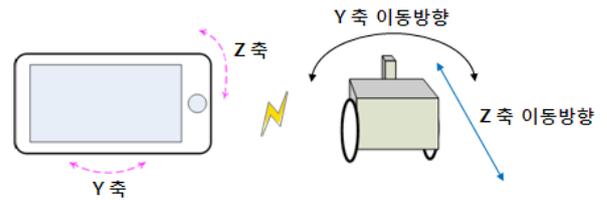


그림 4. 로봇의 제어  
Fig. 4. Robot control

로봇의 모터에 적용되는 각각의 PWM값은 다음처럼 구해진다.

$$\begin{aligned} p_l &= s_m (k_1(\tilde{z}_a + z_i) - k_2 \tilde{y}_a y_r) \\ p_r &= s_m (k_1(\tilde{z}_a + z_i) + k_2 \tilde{y}_a y_r) \end{aligned} \quad (6)$$

여기서,  $k_1, k_2$ 는 비례상수 이고  $k_1 + k_2 = 1$  이다.  $z_i$ 는 아이폰 z축의 초기 가속도 값을  $y_r$ 은 아이폰의 좌우 기울기에 대한 비례값이다.  $s_m$ 은 로봇의 최대 속도이다. 로봇은 아이폰에서 전송된 모드에 따라  $s_m$ 값이 변경된다.

### 3. 아이폰의 개발환경 구축 및 통신 방법

#### 3.1 아이폰의 개발환경 구축

아이폰을 사용한 체감형 인터페이스를 디자인하고 통신 프로그램을 작성하기 위해서는 Mac OS가 필요하고 프로그램 작성을 위해 Apple사에서 제공하는 SDK가 필요하다. SDK는 아이폰에 설치된 센서들을 사용하기 편리하게 프레임워크 라이브러리로 제공 한다. 아이폰의 커넥터는 총 30개의 핀으로 이루어져있고 이 핀들은 시리얼 통신, 영상, 음성 관련 핀들로 구성된다. 본 연구에서 아이폰은 로봇과 블루투스 통신을 통하여 로봇을 제어하기 때문에 RS-232 핀을 그림 5처럼 선택하여 사용한다.

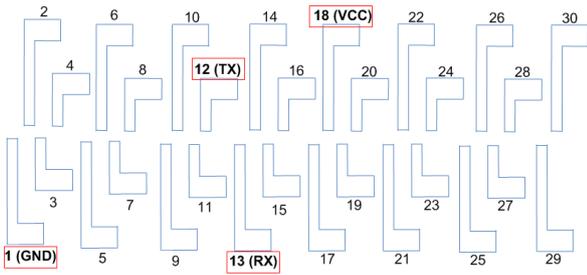


그림 5. 시리얼 통신을 위한 아이폰 커넥터  
Fig. 5. iPhone connector for serial communication

### 3.2 아이폰과 로봇의 통신 방법

본 연구에서 아이폰과 로봇의 통신 방법으로 다음과 같은 네 가지방법을 사용하였다. 그림 6, 7, 8, 9는 이때 사용한 통신패킷을 보여준다. 여기서 사각형 한 개는 1byte를 의미한다.

#### ① 패킷 전체를 위한 시작문자를 주는 방법

그림 6과 같은 데이터 패킷을 아이폰에서 만든후 로봇에게 전송한다. 이 패킷은 NMEA 데이터를 보내는 GPS수신기가 사용하는 방법과 유사하다. 패킷 전체를 위한 시작 문자 '#'과 구분자 ','를 정하여 많은 양의 데이터를 한번에 보내는 장점이 있다. 하지만, 전체 패킷의 크기에 시작 문자와 구분자가 포함되기 때문에 패킷의 길이가 긴 단점이 있다. 이 방법은 총 데이터를 15byte를 전송한다. 로봇은 전체 데이터를 버퍼에 저장한 후 데이터를 분석하여 로봇의 제어 입력으로 사용한다. 따라서 로봇이 데이터를 분석하는 동안에는 아이폰의 다음 데이터를 받을 수 없게 되고 시간 지연이 발생하게 된다. 이 방법은 아이폰과 로봇의 통신간 시간 지연이 발생하기 때문에 실시간 제어가 힘들다.



그림 6. 방법①의 통신 패킷  
Fig. 6. Communication packet of the ① method.

#### ② 각각의 데이터를 위한 시작문자를 각각 주는 방법

① 방법과 다르게 ②방법은 전체 패킷을 위한 '#'문자가 제외 되었다. 대신 로봇에서 사용하기 위한 각각의 데이터, 모드값, y축값, z축값의 시작과 끝을 각각 준다. 그림 7은 ②방법의 통신 패킷을 보여준다. 여기서 'm', 'y', 'z'는 각각 값의 시작을 의미하고 ','는 값의 종료를 의미한다. 로봇은 시작 문자만 확인하여 용도에 맞게 데이터를 저장하여 이용한다. 이 방법은 ①방법에 보다 빠른 제어가 가능하다. 하지만 아이폰에서 mode값, y축값, z축값 순서로 데이터를 계속 보내기 때문에 로봇이 통신 패킷을 분석하는 동안 다른 패킷을 지나치고 값 갱신이 늦어지는 모습을 보여주었고 실시간 제어가 힘들었다.



그림 7. 방법②의 통신 패킷  
Fig. 7. Communication packet of the ② method.

#### ③ 요청에 의한 데이터 전송방법

①, ②방법이 데이터를 무조건 전송하고 로봇이 분석하는 방법인 반면 ③방법은 로봇의 요청이 있을 때 값을 전송하는 방법이다. 그림 8은 ③방법을 쉽게 잘 보여준다. 아이폰은 로봇의 요청이 있을 때까지 기다린다. 만약 데이터가 'y' 이라면 아이폰은 로봇에게 y축 값을 전송한다. 그리고 아이폰은 다음 데이터 요청을 기다린다. 여기서 아이폰의 데이터 전송은 안정하다는 가정을 하여 전송 후 로봇의 완료 데이터는 다시 받지 않는다. ③방법은 다른 ①, ②방법보다 적은 크기의 데이터를 사용하여 통신이 가능하다. 하지만 여전히 y축값과 z축값의 데이터 크기가 크다. 따라서 데이터를 획득 후 계산을 위한 데이터 형 변환이 필요하고 로봇의 제어가 힘든 단점이 존재한다. 이를 해결하기 위해 패킷의 크기도 줄이면서 로봇의 연산시간도 줄이기 위한 방법이 필요 하다.

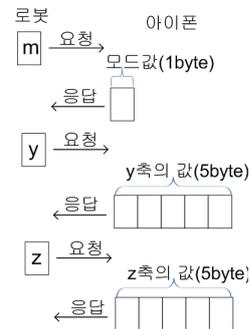


그림 8. 방법③의 통신 패킷  
Fig. 8. Communication packet of the ③ method.

#### ④ 최소의 데이터를 사용한 로봇 제어 방법.

로봇의 연산시간을 최대한 줄이고 로봇의 실시간 제어에 용이 하도록 아이폰에서 로봇의 왼쪽, 오른쪽 모터PWM값을 계산한다. 그리고 그림 9와 같은 데이터 패킷을 전송한다. 여기서 '#'은 데이터의 시작을 의미한다. PWM 데이터는 8개의 비트로 이루어지고 최상위 여덟 번째 비트는 모터의 회전 방향을 의미한다. 예를 들어 0이면 시계방향 회전이고 1이면 반시계방향 회전을 의미한다. 나머지 1~7번째 비트는 데이터를 의미하고 0~127의 값을 가진다.

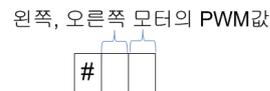


그림 9. 방법④의 통신 패킷  
Fig. 9. Communication packet of the ④ method.

## 4. 실험

### 4.1 네 가지 통신 방법에 따른 통신 실험 결과

본 연구에서는 아이폰의 가속도 센서를 사용하여 체감형 인터페이스로 만든 후, 로봇을 편리하고 쉽게 제어하는데 목적을 두고 있다. 가속도 센서는 아이폰의 기울기에 반응하여 x축, y축, z축들의 값을 변화 시킨다. 세축의 값중에서 두축을 이용한다. 로봇의 전진을 위해 z축을 사용하고, 로봇의 좌우 회전을 위해 y축을 사용한다. 그림 10은 아이폰의 GUI화면이다. 버튼은 총 두 개로 구성하였고 화면을 보지 않고 터치화면이 가능하게 터치 완료시 확인음이 재생된다. START 버

튼은 로봇과의 통신포트를 열어주고 동일 버튼을 한번 더 누르게 되면 로봇은 정지 한다. 다른 Mode 버튼을 누르게 되면  $s_m$  값이 변화되고 로봇의 이동 속도가 변경된다. 초기 값은 저속이고 버튼을 한번 더 누르게 되면 고속으로 설정된다. 제안된 네가지 방법을 사용하여 실험을 한 후 그림 11과 같은 결과를 얻었다. 그림 11은 각 방법을 사용하여 로봇을 제어할 때, 필요한 최소 요구 샘플링 시간과 0.01초의 샘플링 시간을 갖는 아이폰 데이터 500개를 전송한 후 얻는 데이터 수의 비교를 보여준다. 여기서 최소 요구 샘플링 타임은 아이폰에서 로봇 제어를 위한 명령을 빠르게 전송하였을 때 로봇이 이 명령을 획득한 후 제어가 가능한 시간을 의미한다. 그리고 0.01초의 샘플링 시간을 갖는 아이폰 데이터 500개는 아이폰으로부터 5초 분량의 데이터를 각 방법에서 제시한 패킷 모양으로 만든 것을 의미한다. 그림 11에 의해 방법 ④를 사용하였을 때, 로봇이 아이폰과 통신을 위한 샘플링 시간이 가장 짧고, 아이폰에서 데이터를 전송 후 로봇에서 얻는 데이터의 수가 가장 많다는 것을 보여준다. 아이폰에서 총 500개의 데이터를 보내지만 로봇에서 받은 데이터 수가 적은 이유는 다음과 같다. 아이폰은 0.01초의 샘플링 시간으로 데이터를 보내고 데이터를 받는 아이폰은 0.01초 보다 느린 샘플링 시간으로 데이터를 받기 때문이다. 따라서 획득한 데이터 수가 줄어들게 되었다. 만약 아이폰의 샘플링 시간을 로봇이 사용할 수 있는 샘플링 시간 보다 더 느리게 설정 한다면 로봇은 아이폰의 데이터를 손실 없이 획득 할 수 있다. 따라서, 그림 12에서와 같이 아이폰을 사용한 로봇제어는 방법 ④를 사용하였다.

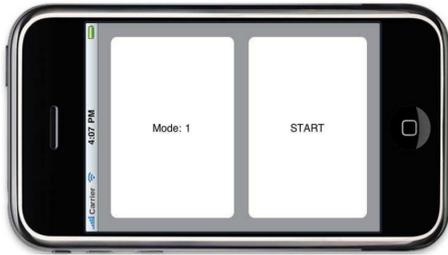


그림 10. 아이폰 GUI 환경  
Fig. 10. GUI environment of iPhone

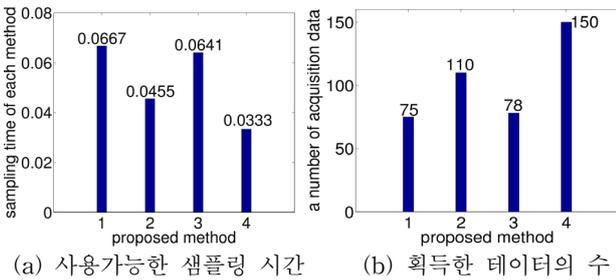


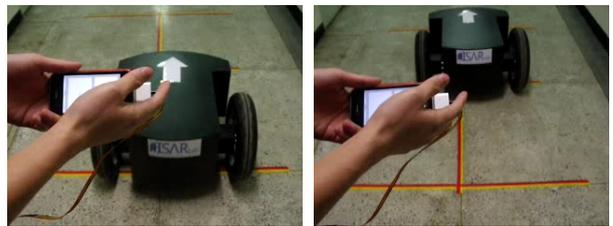
그림 11. 각 방법의 사용가능 샘플링 시간과 획득한 데이터의 수  
Fig. 11. sampling time of each method and a number of acquisition data

4.2 아이폰을 통한 로봇 제어 실험 결과

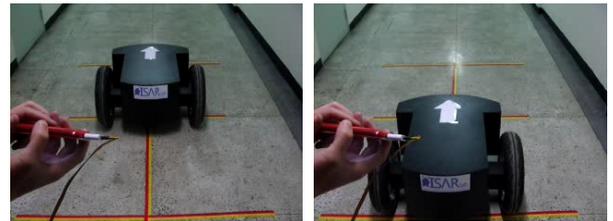
그림 12는 아이폰을 핸들로 사용하여 모바일 로봇을 제어하는 장면을 보여준다. 실험에 사용된 파라미터 값은  $s_m = 40$ ,  $k_1 = 0.6$ ,  $k_2 = 0.4$ ,  $z_i = 0$ ,  $y_r = 0.1$  이다. 그림 12. (a)는 아이폰을 앞으로 눕혀서 로봇이 전진하는 장면이다. 그

림 12. (b)는 아이폰을 뒤로 눕혀서 로봇이 후진하는 장면이다. 그림 12. (c)는 아이폰을 왼쪽으로 기울여서 로봇이 왼쪽으로 회전하는 장면이다. 그림 12. (d)는 아이폰을 오른쪽으로 기울여서 로봇이 오른쪽으로 회전하는 장면이다.

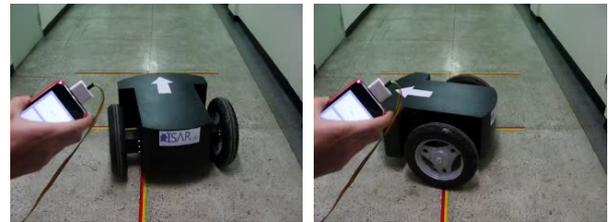
본 연구에서 로봇이 제어될 때, 제어 입력으로 아이폰에 부착된 가속도 센서 값을 사용한다. 아이폰의 가속도 센서를 측정된 결과, 센서 값에는 고주파의 잡음 성분이 많이 섞여있다. 로봇을 제어할 때, 사용자는 아이폰을 핸들처럼 이용하여 좌, 우, 앞, 뒤로 움직이기 때문에, 고주파 잡음은 사용자 의도와 다르게 로봇 제어에 영향을 미친다. 이를 제거하기 위해 식 (5)와 같은 디지털 저주파 통과 필터를 사용하였다. 그 결과 값을 사용하여 로봇을 제어하기 위해 식 (6)을 이용해 오른쪽과 왼쪽 모터의 입력값을 구하였다. 본 연구의 동영상은 [15]에서 볼 수 있다.



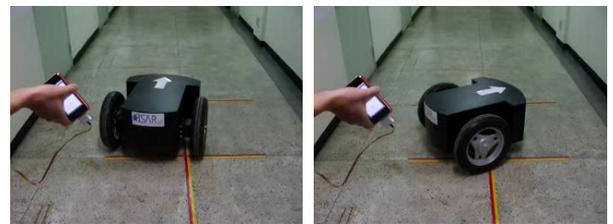
(a) 전진 이동 (左: 전, 右: 후)



(b) 후진 이동 (左: 전, 右: 후)



(c) 왼쪽 회전 (左: 전, 右: 후)



(d) 오른쪽 회전 (左: 전, 右: 후)

그림 12. 아이폰을 사용한 로봇 제어[15]  
Fig. 12. Control of a robot using iPhone[15]

5. 결 론

본 논문에서는 아이폰의 가속도 센서를 이용하여 체감형 인터페이스를 구성한 후 로봇을 제어 한다. 아이폰을 자동차

의 핸들처럼 응용하게 되면 일반적인 조이스틱 혹은 버튼식 원격제어 방법과 다르게 쉽고 편리하게 로봇의 원격 제어가 가능하다. 로봇의 무선 제어는 블루투스 통신을 사용하기 때문에 WiFi를 사용할 때와 달리 실내 혹은 실외 전 지역에서 사용이 가능하다. 본 연구에서는 아이폰과 로봇간의 무선 통신 제어를 생기는 커다란 시간 지연을 줄이기 위해 네 가지의 통신 방법을 제시하였다. 각각에 대하여 실험한 후 결과가 가장 좋은 네 번째 방법을 적용하였다. 결과로서, 로봇의 실시간 제어를 위해 로봇과 iPhone의 무선 통신에서 생기는 입력 시간 지연을 충분히 제어 가능한 수준으로 줄였다. 아이폰을 사용하여 만든 체감형 인터페이스는 편리하고 쉽게 로봇을 제어 가능하게 해준다. 실험의 결과로 제안된 체감형 인터페이스는 즉흥적이고 쉽게 사용자의 의도를 반영하고 이동로봇의 원격제어에 적합함을 보여준다.

### 참 고 문 헌

[1] P. Abolmaesumi, S. E. Salcudean, W. H. Zhu, S. P. Di Maio, and M. R. Sirouspour, "A User Interface for Robot-Assisted Diagnostic Ultrasound," *Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on*, vol. 2, pp. 1549-1554, 2001.

[2] H. A. Tijsma, F. Liefhebber, and J. L. Herder, "A Framework of Interface Improvements for Designing New User Interfaces for the MANUS Robot Arm," *Rehabilitation Robotics, 2005. ICORR 2005. 9th International Conference on*, pp. 235-240, 2005.

[3] H. Soonshin and L. JangMyung, "Tele-operation of a Mobile Robot Using a Force Reflection Joystick with a Single Hall Sensor," *Robot and Human interactive Communication, 2007. RO-MAN 2007. The 16th IEEE International Symposium on*, pp. 206-211, 2007.

[4] H. Kyung-Lyong, C. O. Kyu, L. In, H. Inwook, L. Jin S., and C. Seungmoon, "Design and Control of Omni-Directional Mobile Robot for Mobile Haptic Interface," *International Conference on Control, Automation and Systems 2008*, pp. 1290-1295, 2008.

[5] D. Shilpi, A. Dutta, A. Mukerjee, and P. Kulkarni, "Development of A Speech Interface for Control of a Biped Robot," *Robot and Human Interactive Communication, 2004. ROMAN 2004. 13th IEEE International Workshop on*, pp. 601-605, 2004.

[6] C. R. Hema, M. P. Paulraj, S. Yaacob, A. H. Adom, and R. Nagarajan, "Robot Chair Control using an Asynchronous Brain Machine Interface," *Signal Processing and Its Applications (CSPA), 2010 6th International Colloquium on*, pp. 1-4, 2010.

[7] K. Hinckley, M. Sinclair, E. Hanson, R. Szeliski, and M. Conway, "The VideoMouse: a camera-based multidegree of freedom input device," *In UIST '99: Proceedings of the 12th annual ACM symposium on User interface software and technology*, pp. 103-112, 1999.

[8] <http://www.livescribe.com>

[9] P. J. Hwan, K. K. Oh, and J. J. Wook, "PDA based user interface management system for remote control robots," *SICE-ICASE International Joint Conference 2006*, pp. 417-420, 2006.

[10] R. Pierre, B. Jérôme, and O. Pierre-Yves, "A comparison of three interfaces using handheld devices to intuitively drive and show objects to a social robot: the impact of underlying metaphors," *Robot and Human Interactive Communication, 2009. RO-MAN 2009. The 18th IEEE International Symposium on*, pp. 1066-1072, 2009.

[11] N. Kubota, W. Sato, "Robot Design Support System based on Interactive Evolutionary Computation using Boltzmann Selection," *Evolutionary Computation (CEC), 2010 IEEE Congress on*, pp. 1-8, 2010.

[12] N. Kubota, S. Wakisaka, and A. Yorita, "Tele-operation of Robot Partners through iPod touch," pp. 75-80, 2009.

[13] D. H. Kim and S. H. Han, "Robust Self-organization for Swarm Robots," *International Conference on Control, Automation and Systems 2008*, pp. 1228-1233, 2008.

[14] [http://en.wikipedia.org/wiki/Low-pass\\_filter](http://en.wikipedia.org/wiki/Low-pass_filter).

[15] <http://www.kyungnam.ac.kr/~dhkim> > pds.

### 저 자 소 개



**정하민(Hahmin Jung)**

2009년 : 경남대학교 전기전자 공학부 졸업  
2009년~현재 : 동 대학원 첨단공학과 박사 과정

관심분야 : 스웸 시스템, 지능제어&로봇, 디지털 신호처리, 영상처리



**김동헌(Dong Hun Kim)**

2001년 : 한양대학교 전기공학과(공학 박사)  
2001년~2003년 : 미국 듀크 대학교 연구원  
2003년~2004년 : 미국 보스턴 대학교 연구원  
2004년~2005년 : 일본 동경대학교 박사후 과정  
2005년~현재 : 경남대학교 전기공학과 부교수

관심분야 : 스웸 시스템, 이동로봇 경로계획, 지능 로봇 제어, 비선형 적응제어

Phone : 055-249-2629

Fax : 055-249-2839

E-mail : dhkim@kyungnam.ac.kr