

# 보행용 관성 항법 시스템을 위한 HMMS를 통한 걸음 단계 구분

논문
58-5-22

## Gait State Classification by HMMS for Pedestrian Inertial Navigation System

박상경\* · 서영수†  
(Sang-Kyeong Park · Young-soo Suh)

**Abstract** - An inertial navigation system for pedestrian position tracking is proposed, where the position is computed using inertial sensors mounted on shoes. Inertial navigation system(INS) errors increase with time due to inertial sensor errors, and therefore it needs to reset errors frequently. During normal walking, there is an almost periodic zero velocity instance when a foot touches the floor. Using this fact, estimation errors are reduced and this method is called the zero velocity updating algorithm. When implementing this zero velocity updating algorithm, it is important to know when is the zero velocity interval. The gait states are modeled as a Markov process and each state is estimated using the hidden Markov model smoother. With this gait estimation, the zero or nearly zero velocity interval is more accurately estimated, which helps to reduce the position estimation error.

**Key Words** : Pedestrian Navigation System, Zero Velocity Updating; Hidden Markov Mode

### 1. 서론

보행용 관성 항법 시스템은 관성 센서를 이용하여 사람의 위치를 추적한다. 보행용 관성 항법 시스템은 응급 요원이나 안전 요원 그 밖에 여러 부분에서 사용될 수 있다. 예를 들어 소방수가 화재진압 현장에서 임무 중에 자신의 위치를 알리지 못하는 상태가 되었을 때, 항법 시스템으로 부터의 위치 정보는 구조 활동에 유용하게 쓰일 수 있다[1].

실외에서는 GPS를 항법 시스템에 이용할 수 있다. 하지만 GPS 위성신호가 잘 잡히지 않는 실내에서는 항법을 위한 특별한 환경을 필요로 하게 된다. 예를 들어, 건물 안에 자신의 위치를 지속적으로 알려주는 송신기나 RFID 모듈 같은 것을 통하여 항법 시스템을 구성하게 된다. 하지만 사전에 답사나 환경 설정을 하지 못한 장소에는 앞에 소개된 방법들도 사용하기 힘들어 진다.

관성 항법 시스템은 관성센서(3개의 가속도계와 3개의 자이로스코프)를 이용하여 위치를 계산하게 되며, 앞에서 언급 하였던 외부 여건이 필요하지 않게 된다. 앞서 개발되어진 몇몇의 보행용 관성 항법 시스템은 관성센서모듈을 신발위에 위치 시켰다. 이들 관성 시스템 알고리즘 들은 서로 유사하며, 영속도 보정 방법을 사용하고 있다. 영속도 보정 방법은 만약 시스템이 속도가 영(zero)인 구간을 알 수 있으면, 그 구간은 실제 속도(영속도)를 알 수 있게 되므로, 추정된 속도에 대한 오차를 추정할 수 있고, 이를 보상한다. 걷

는 동안 주기적으로 속도 오차를 리셋 시킬 수 있게 되면, 속도 오차는 축적되지 않게 되고, 위치 오차에 대한 부분도 줄일 수 있게 된다[1-4].

영속도 보정을 위해서는 영속도가 되는 부분을 인지하는 것이 중요하다. 영속도 구간을 찾기 위한 방법으로 여러 연구가 되었는데, 가속도 값과 자이로스코프의 값이 일정 값보다 작아지는 시간이 기준 시간보다 길어질 때, 영속도로 인식하는 경우[2]와 3개의 자이로스코프 값의 norm을 이용하여 영속도를 찾은 결과가 있다[3]. 또 다른 형태로는 신발 외부에 압력센서를 부착하고, 압력센서가 압력을 측정하는 동안을 영속도로 추정하였다[4].

본 논문에서는 관성센서만을 이용하여 걸음 단계(gait phase)를 구분하기 위하여 hidden Markov model(HMM)을 제안한다. 보편적인 걸음에서는 각 걸음 단계가 일정한 패턴을 보이는데, 일반적인 칼만 필터로 추정할 경우, 이러한 패턴에 대한 모델링이 쉽지 않으며, 외란 등의 영향에 민감하게 반응하였다. HMM을 사용하면, 각 걸음 상태를 하나의 프로세스로 정의하고, 프로세스 변화에 따른 확률적 요소로 추정을 하기 때문에 걸음 단계를 구분하기 위한 알고리즘으로 적합하다는 결론을 내렸다. HMM을 사용함으로써 단순한 관성 센서 측정치의 기준값에 대한 비교만이 아닌, 현재 상태에 대한 확률의 요소를 추가하였다[5-6].

### 2. Gait Phase

인체학적인 관점에서는 사람이 걸음을 걸을 때 걸음 단계를 다음의 8단계로 구분한다[7].

- 단계 1(initial contact) : 발이 바닥에 닿는 순간 나타나는 단계로, 발뒤꿈치가 닿게 된다.

\* 준 회원 : 울산대학 전기전자정보시스템공학부 박사과정  
† 교신저자, 정회원 : 울산대학 전기전자정보시스템공학부 교수  
E-mail : suh@ieee.org  
접수일자 : 2008년 12월 17일  
최종완료 : 2009년 3월 25일

- 단계 2(loading response) : 발의 다른 부분이 바닥에 닿는 순간까지의 단계.
- 단계 3(mid stance) : 한 발로 땅을 딛고 있는 상태.
- 단계 4(terminal stance) : 발뒤꿈치가 들리기 시작하고, 다른 발은 지면을 닿으려 하는 순간.
- 단계 5(pre-swing) : 스윙하던 맞은편의 다리는 단계 1로 들어간다.
- 단계 6(initial swing) : 발을 들어 올리는 순간.
- 단계 7(mid swing) : 들어 올린 발은 스윙을 하게 되고, 결과적으로 스윙하는 다리가 지면을 딛는 다리보다 앞서게 된다.
- 단계 8(terminal swing) : 발이 땅에 닿기 직전 순간.

영속도 보정을 통한 보행용 관성 시스템에 사용하기 위해 걸음 단계에서, 발의 속도가 0이나 0에 가까운 상태에만 관심이 있다. 이 때문에, 본 논문에서는 위의 8단계의 걸음 단계를 다시 다음의 4가지 상태(gait state)로 구분한다.

- 상태 1 : 걸음 단계 1, 2
- 상태 2 : 걸음 단계 3
- 상태 3 : 걸음 단계 4, 5
- 상태 4 : 걸음 단계 6, 7, 8

여기에서 상태 2가 영속도 구간이다. 상태 1은 발뒤꿈치가 닿는 순간부터 상태 2직전까지, 상태 3은 발뒤꿈치가 떨어지는 순간부터 발이 땅에서 떨어지는 순간까지로 정의한다. 상태 4는 상태 1, 2, 3을 제외한 상태로, 발이 지면에 닿아 있지 않는 순간의 의미한다. 그림 1은 걸음에 대한 네 가지 구분에 대한 도식이다.

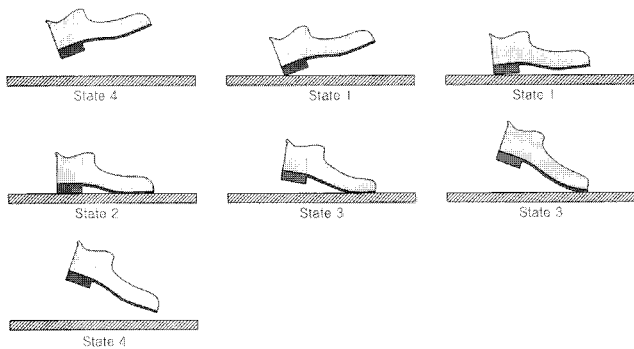


그림 1 걸음 상태.  
Fig. 1 Gait states.

### 3. Hidden Markov model

본 논문에서는 걸음 단계를 추정하기 위하여 hidden Markov model(HMM)을 사용하였다. Markov model은 모델링하는 시스템이 미지의 파라미터를 가진 Markov process 일 것이라고 가정하고, 그 가정에 기초해서 관측된 파라미터로부터 숨겨진 파라미터를 결정하려는 하나의 통계모델이다. HMM은 출력치만 관측되고 상태의 흐름은 관측되지 못하는 경우에 사용하게 된다.

미래의 Process가 현재의 상태에 의해서만 결정되는, 즉 과거와는 독립적인 random process를 Markov process라고

하며, 본 논문에서는 4개의 걸음 상태를 process로 정의하였다. 각 상태에서 다음 상태로의 변화 가능한 상태를 도식으로 표현한 것을 상태 천이도(state transition diagram)라 하고 그림 2이다.

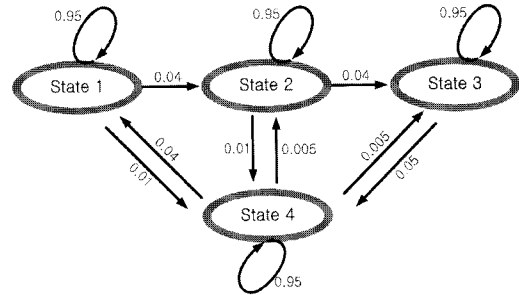


그림 2 상태 천이도  
Fig. 2 State transition diagram

그림 2를 보면 하나의 상태에서 다른 상태로 변화할 확률보다는 계속해서 이전 상태를 유지할 확률이 크기 때문에 자기 자신으로 유지할 확률을 가장 크게 잡았다. 그리고 각 상태에서 다른 상태로 변화할 때, 절대로 변화될 수 없는 상태들이 있다. 예를 들어, 상태 1에서 상태 3, 즉 발 뒤꿈치가 닿은 상태에서 엄지쪽 발바닥만 닿는 상태는 일반적인 걸음에서 나올 수가 없다. 이러한 상태 변화의 확률은 0이 된다. 이러한 원리로 각 상태에서 다른 상태로의, 확률을 그림 2와 같이 정리하였다. 각 수치는 임의로 결정하였고, 여러번 실험을 통하여 조금씩 조정하였다. 그림 2의 상태 천이도룰 토대로 행렬화 한 것이 상태 천이 행렬(state transition probability matrix)이며 (1)과 같다.

$$A = \begin{bmatrix} 0.95 & 0 & 0 & 0.04 \\ 0.04 & 0.95 & 0 & 0.005 \\ 0 & 0.04 & 0.95 & 0.005 \\ 0.01 & 0.01 & 0.05 & 0.95 \end{bmatrix} \quad (1)$$

이 때, A의 각 원소는 다음과 같은 규칙을 가진다.

$$a_{ij} = \Pr [X_{k+1} = i | X_k = j], \quad \sum_{i=1}^N a_{ij} = 1$$

상태 천이 행렬은 현재 상태에서 다음 상태로 변화하는 확률을 의미하며, 현재 상태에서 다음 상태로의 모든 조건에 대한 확률이기 때문에 각 열 행렬의 합은 1이 된다. (1)의 1열에 대한 설명을 해 보면, 상태 1에서 상태를 유지할 확률은 0.95, 상태 2로 변화될 확률은 0.04, 상태 4로 변화될 확률은 0.01이며, 상태 3으로 변화될 확률은 0이 된다. 만약 자기 자신으로 갈 확률이 너무 낮게 되면, 빈번한 상태 변화로 제대로 상태를 추정할 수 없게 되며, 반대로 확률이 너무 높으면 다른 상태로의 변화가 힘들어지기 때문에 적절한 값을 갖는 것이 중요하다.

관성 센서 모듈을 통해 측정되는 값은 가속도 3축, 회전 각속도 3축, 마그네틱 센서를 통한 자기력 3축의 값들이다. 그림 3은 관성 센서 모듈이 신발에 장착된 형태와 센서 모듈에서의 측정 기준이 되는 축인 동체 좌표계와 실제 계산을 위해 사용하는 항법 좌표계를 나타낸 그림이다. 기본적으로 사람이 향하는 방향을 x축, 위로 향하는 축을 z축으로 설정하였다. 하지만 신발의 구조상 관성 센서를 설치하였을 때, 동체 좌표계와 항법 좌표계는 정확히 일치하지 않기 때

문에 자세 계산 시에 보정을 필요로 한다. 센서에서 측정되는 데이터 중에 걸음 상태 추정을 위한 데이터로 z축(지구 중력방향) 가속도와 y축 회전 각속도 성분을 사용하였다. z축 가속도는 발이 땅에 닿는 순간 큰 가속도 결과를 가지게 되며, y축 회전 각속도는 각 상태에 따라 적절히 다른 상태를 보이기 때문에 사용하였다.

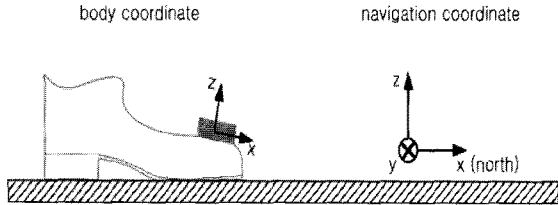


그림 3 동체좌표계와 항법 좌표계  
Fig. 3 Body coordinate and navigation coordinate

그림 4와 5는 3초간 2걸음을 보행한 데이터이며, 각 상태에 대한 z축 가속도와 y축 회전 각속도 성분의 경향성을 보여주고 있다. 실험은 관성 센서 모듈이 부착된 신발을 신고 걷는 동안 신발에 부착된 여러 개의 압력센서를 통하여 각 단계를 구분했다. 이 때, 사용된 압력 센서는 Tekscan사의 FlexiForce A201(그림 6a)을 사용하였으며, 이 센서는 센싱 부위에 힘이 가해지면 저항이 줄어드는 특성을 가지고 있다. 그림 6b는 압력센서를 부착한 위치를 나타낸 그림으로 엄지 발가락, 발 뒤꿈치, 발가락 아래의 센서를 압박할 수 있는 부분에 위치시키며, 센서는 신발의 깔창 아래에 부착하였다. 사람이 걸을 때, 발 뒤쪽, 발바닥 전체, 엄지 발가락 순으로 지면을 닿게 된다. 이를 통해 각 부분의 압력 센서의 출력을 통하여, 걸음 상태를 구분하였다.

마이크로컨트롤러에서 관성 센서 모듈(100Hz)의 데이터와 압력센서의 출력 데이터(100Hz)를 취합하여 실시간으로 컴퓨터로 보내준다. 이를 토대로 각 상태에 따른 두 센서 신호의 경향을 파악하고, 이를 통해 표 1과 같은 조건을 설정하였다.

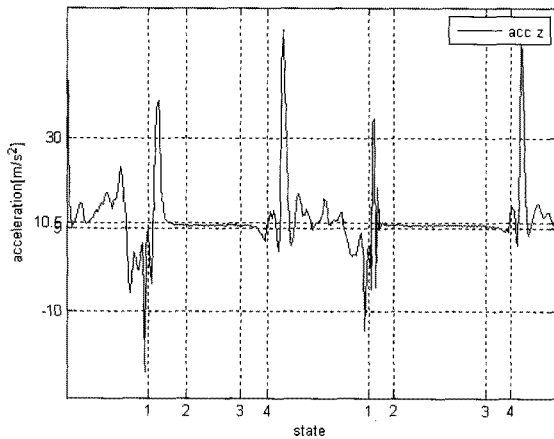


그림 4 z축 가속도계 출력.  
Fig. 4 Accelerometer output of z axis.

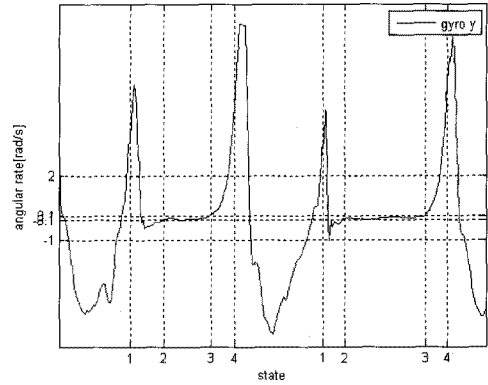


그림 5 y축 자이로스코프 출력.  
Fig. 5 Gyroscope output of y axis.

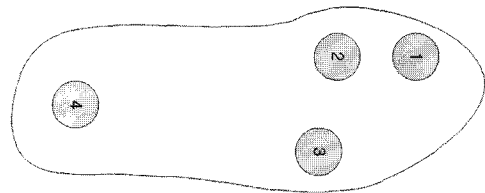
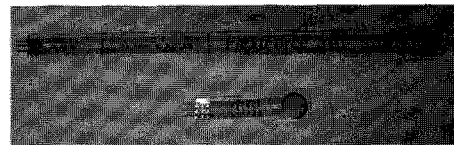


그림 6 FlexiForce와 센서 위치  
Fig. 6 The FlexiForce and sensors position.

표 1의 조건에 의해 데이터를 구분하면 25개의 조합(표 2)이 나오게 된다. 이 조합에 의한 관측행렬(observation matrix) C를 정의할 수 있는데, 행렬의 각 원소는 식 (2)와 같은 조건과 성질을 가진다.

$$c_{mn} = \Pr \{ Y_k = m | X_k = n \}, \sum_{m=1}^M c_{mn} = 1 \quad (2)$$

표 1 센서 출력 구분 조건  
Table 1 Sensor output quantization

	z축 가속도 [m/s <sup>2</sup> ]	y축 회전각속도 [rad/s]
1	$a_z < -10$	$g_y < -1$
2	$-10 \leq a_z < 9$	$-1 \leq g_y < -0.1$
3	$9 \leq a_z < 10.5$	$-0.1 \leq g_y < 0.1$
4	$10.5 \leq a_z < 30$	$0.1 \leq g_y < 2$
5	$a_z \geq 30$	$g_y > 2$

표 2 출력 결과 정의

Table 2 Output process definition

번호	z축 가속도 조건	y축 가속도 조건
1	1	1
2	1	2
3	1	3
4	1	4
5	1	5
6	2	1
7	2	2
⋮	⋮	⋮
25	5	5

관측 행렬을 구하기 위해서 상태 구분을 한 일련의 데이터를 사용하여 표본을 조사하였다. 우선 각 상태에 대하여 어떠한 출력 결과들을 가지는지 조사하였다(표 3의 1단계 참조). 표에 대한 내용은 센서 출력 구분 조건에 의해 나누어진 출력 결과 번호에 대한 각 상태가 가지는 빈도수이다. 예를 들어, 출력 결과 7(조건:  $-10 \leq a_z < 9$ ,  $-1 \leq g_y < -0.1$ )의 경우에는 상태 1의 경우가 4번 있고, 상태 4의 경우가 37번 있다는 의미이다. 표본 조사한 데이터를 그대로 사용하지 않고, 약간의 수정작업을 하였다. 출력 결과 21에서 25의 경우처럼 데이터의 개수가 적은 결과들은 0으로 만들었다. 이러한 경우는 출력 결과가 각 상태에서 우연히 나타나

것으로, 이러한 결과를 그대로 사용하게 되면 각 상태 변화에 대한 경우의 수가 많아지고, 이것은 잦은 상태의 변화로 결과에 악영향을 미치게 된다. 또 번호 10, 17과 같이 거의 비슷한 개수로 나오는 경우는 동일한 결과로 보고 수가 같도록 조종을 하였다(표 3의 2단계 참조).

0.0001	0.0001	0.0001	0.0001
0.0001	0.0001	0.0001	0.0001
0.0001	0.0001	0.0001	0.0001
0.0001	0.0001	0.0001	0.0001
0.0001	0.0001	0.0001	0.0001
0	0	0	0.01
0.04	0	0	0.1
0.0001	0.0001	0.0001	0.0001
0	0	0.06	0.2
0.14	0	0.21	0.21
0	0	0	0.01
0.3586	0.07	0	0.04
0.11	0.8586	0	0.02
0.03	0.07	0.7086	0.05
0.0001	0.0001	0.0001	0.0001
0	0	0	0.0186
0.24	0	0	0.24
0.0001	0.0001	0.0001	0.0001
0.0001	0.0001	0.0001	0.0001
0.08	0	0.02	0.1
0.0001	0.0001	0.0001	0.0001
0.0001	0.0001	0.0001	0.0001
0.0001	0.0001	0.0001	0.0001
0.0001	0.0001	0.0001	0.0001
0.0001	0.0001	0.0001	0.0001

(3)

표 3 관측 행렬을 만들기 위한 작업

Table 3 Processes for making observation matrix

번호	1 단계				2 단계			
	상태 1	상태 2	상태 3	상태 4	상태 1	상태 2	상태 3	상태 4
1	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0
4	0	0	0	3	0	0	0	0
5	0	0	0	4	0	0	0	0
6	0	0	0	29	0	0	0	29
7	4	0	0	37	4	0	0	37
8	2	0	0	6	0	0	0	0
9	0	0	6	20	0	0	6	20
10	14	0	20	21	14	0	21	21
11	0	0	0	22	0	0	0	22
12	55	7	0	4	55	7	0	4
13	11	349	0	2	11	349	0	2
14	3	7	53	5	3	7	53	5
15	2	0	3	1	0	0	0	0
16	0	0	0	119	0	0	0	119
17	23	0	0	24	24	0	0	24
18	2	0	0	2	0	0	0	0
19	0	0	0	8	0	0	0	0
20	8	0	2	23	8	0	2	23
21	0	0	0	1	0	0	0	0
22	1	0	0	2	0	0	0	0
23	0	0	0	1	0	0	0	0
24	6	0	0	4	0	0	0	0
25	7	0	0	9	0	0	0	0

식 (2)에서 관측행렬의 경우에는 열 행렬의 합이 1이 되어야 한다. 이를 쉽게 만들기 위해서는 표 3의 2단계에서 각 열의 개수에 대한 비율을 구하면 된다. 하지만 이렇게 구하는 경우, 다른 상태에 대한 비율이 달라지는 문제점이 생기게 된다. 예를 들어 표 3의 2단계에서 상태 1의 경우 열의 합은 119인데 반해, 상태 4의 경우에는 306으로 많은 차이가 있다. 단순한 비율로 계산을 하게 되면 결과 17의 경우 상태 1은 0.202를 가지는 반면 상태 4는 0.078의 값을 가지게 된다. 다시 말하면 결과 17일 경우에는 상대적으로 상태 1로 인식할 확률이 배가 넘게 되는 것이다. 이러한 경우를 방지하기 위하여 일정 값을 곱하여 비율을 만들게 되었다. 하지만 이렇게 하는 경우 각 열 행렬의 합이 1이 될 수가 없다.

이러한 문제를 해결하기 위해 몇 가지 경우로 구분해 보았다. 우선 네 개의 항 모두 0인 경우, 다른 항들은 0이고, 하나의 항에만 값을 가지는 경우, 해당 단계에 여러 가지 상태가 존재하는 경우로 구분을 한 뒤, 각 상태에 따라 차별적으로 값을 부여하였다. 첫 번째, 네 개의 항 모두 0인 경우에는 똑같이 0을 부여할 수도 있지만, 이러한 경우 계산을 통해 에러가 발생하게 된다. 이러한 항들은 상태 변화에 큰 역할을 못하는 경우로, 이전 상태의 유지가 가장 좋은 결과로 볼 수 있다. 이를 위해 네 항 모두 동일한 값을 주어, 상태 천이 행렬에서 이전 상태의 유지를 할 수 있도록 하였다. 두 번째로 하나의 항에만 값을 가지는 경우에는 다른 항의 값이 0이 되므로 값을 가지는 항의 계수가 작은 수라도 결과적으로는 해당 상태로 인식하게 되므로, 적당히 작은 값을 주었다. 세 번째로 여러 상태가 공존하는 경우에는 일정 비율의 값을 사용하였다.

이러한 방식으로 정리하여도, 각 열 행렬의 합이 1이라는 조건을 만족할 수가 없다. 이 조건을 만족시키기 위해 상대적인 우위를 가지는 값들을 더욱 증가시키거나, 충분히 큰 값을 유지하면서 감소 시켜서 상대적인 비율을 맞추는 과정을 거쳐서 식 (3)과 같은 결과를 얻을 수 있다. (3)에서 첫 번째 행([0.0001 0.0001 0.0001 0.0001])과 같이 네 개의 값이 같게 되면, 결과적으로 이전 상태의 유지가 되게 된다. 반면에 여섯 번째 행([0 0 0 0.01])과 같이 네 번째 행의 값만 값을 가질 경우, 그 값이 작은 값이더라도 결과적으로는 상태 4로 결정 된다. 열 번째 행([0.14 0 0.21 0.21])의 결과가 나오는 경우에는 이전 상태에 대한 확률과 상태 천이 확률에 의한 계산을 하게 되며, 다음에 소개 되는 HMM filter에서 소개한다.

### 3.1 HMM filter

HMM filter는 현재까지 출력에 대한 다음 상태를 추정하기 위해 사용된다. 확률 벡터  $\Pi_{k|k}$ 는 처음부터  $k$ 까지 출력 결과를 통한  $k$ 번째의 확률이며 수학적으로 다음과 같이 정의된다[5][6].

$$\Pi_{k|k} \triangleq \Pr[X_k = i | Y_0, \dots, Y_k]$$

유사하게  $\Pi_{k+1|k}$ 는  $\Pi_{k|k}$ 를 통하여  $k+1$ 번째 결과를 유추한 것으로  $\Pr[X_{k+1} = i | Y_0, \dots, Y_k]$ 정의를 가진다. HMM filter는 다음과 같은 식으로 다음 상태를 추정하게 된다.

$$\Pi_{k+1|k} = A\Pi_{k|k} \tag{4}$$

$$\Pi_{k+1|k+1} = \frac{1}{\mathbf{1}_N^T C_{Y_{k+1}} \Pi_{k+1|k}} C_{Y_{k+1}} \Pi_{k+1|k}$$

현재 상태의  $\Pi_{k|k}$ 에 상태 천이 행렬  $A$ 를 이용하여  $\Pi_{k+1|k}$ 를 구한다. 각 센서들을 통해 들어오는 데이터( $Y_{k+1}$ )를 통해 표 2의 출력 결과  $l$ 을 얻게 되고, 이에 해당하는 관측행렬의 열벡터를 이용한 행렬  $C_{Y_{k+1}}$ 를 구한다. 이를 통해 다음 상태의 추정값인  $\Pi_{k+1|k+1}$ 을 구하게 된다. 예를 들어,  $k$ 번째 확률  $\Pi_{k|k} = [0.1 \ 0.2 \ 0.7 \ 0.0]$ 라면 추정되는 상태는 가장 큰 값 (0.7)을 가지는 3이 된다.

여기에서,  $\mathbf{1}_N$ 은 1의 값을  $N$ 개 가지는 열 행렬( $[1, \dots, 1]$ )이며,  $Y_{k+1}$ 의 결과값이  $l$ 일 때,  $C_{Y_{k+1}} = \text{Diag}(C_{l1} \ C_{l2} \ C_{l3} \ C_{l4})$ 이다. 그리고,  $\frac{1}{\mathbf{1}_N^T C_{Y_{k+1}} \Pi_{k+1|k}}$ 는 각 데이터의 총합을 1로 만들기 위한 scalar normalizing constant 과정이다.

### 3.2 HMM smoothing

Filter는 지금까지의 결과를 통해 현재 상태에서 다음 상태를 추정하는 반면, smoothing은 현재까지의 결과를 이용하여 과거의 상태를 추정하게 되며 본 논문에서는 여러 smoothing 방법 중에 fixed-lag smoothing method를 사용하였다. HMM smoothing에서 확률벡터  $\tilde{\Pi}_{j|j+\Delta}$  ( $k=j+\Delta$ )는 일정한 시간  $\Delta$ 이후에 추정되는  $j$ 시간의 추정 확률이 되며, 다음과 같이 정의 된다[6].

$$\begin{aligned} \tilde{\Pi}_{j|j+\Delta} &\triangleq \Pr(X_j = i | Y_0, \dots, Y_{j+\Delta}) \\ &= \sum_{i=1}^N \Pr(X_j = i, X_{j+\Delta} = i | Y_0, \dots, Y_{j+\Delta}) \end{aligned}$$

HMM smoothing은 다음과 같이 정리된다.

$$\begin{aligned} \hat{\Pi}_{j,k+1|k} &= \tilde{A} \hat{\Pi}_{j,k|k} \\ \hat{\Pi}_{j,k+1|k+1} &= \frac{1}{\mathbf{1}_N^T \tilde{C}_{Y_{k+1}} \hat{\Pi}_{j,k+1|k}} \tilde{C}_{Y_{k+1}} \hat{\Pi}_{j,k+1|k} \end{aligned} \tag{5}$$

이 때,  $\otimes$ 는 크로네커 곱(Kronecker product)이며,

$$\begin{aligned} \tilde{A} &= I_N \otimes A \\ \tilde{C}_{Y_{k+1}} &= I_N \otimes C_{Y_{k+1}} \end{aligned}$$

확률 벡터  $\Pi_{j|j+\Delta}$ 는 다음과 같이 정리된다.

$$\begin{aligned} \tilde{\Pi}_{j|j+\Delta} &= \begin{bmatrix} 1 & \dots & 1 & 0 & 0 \\ 0 & & & & 0 \\ 0 & & 0 & 1 & \dots & 1 \end{bmatrix} \hat{\Pi}_{j,j+\Delta|j+\Delta} \\ &= (I_N \otimes \mathbf{1}_N') \hat{\Pi}_{j,j+\Delta|j+\Delta} \\ &= (I_N \otimes \mathbf{1}_N') \frac{1}{\mathbf{1}_N^T \tilde{C}_{Y_{j+\Delta}} \hat{\Pi}_{j,j+\Delta|j+\Delta-1}} \tilde{C}_{Y_{j+\Delta}} \hat{\Pi}_{j,j+\Delta|j+\Delta-1} \\ &= (I_N \otimes \mathbf{1}_N') \frac{1}{\mathbf{1}_N^T (I_N \otimes U_{j+1,j+\Delta}) \hat{\Pi}_{j,j|j}} (I_N \otimes U_{j+1,j+\Delta}) \hat{\Pi}_{j,j|j} \end{aligned} \tag{6}$$

이 때,

$$U_{j+1,j+\Delta} = \tilde{C}_{Y_{j+\Delta}} \tilde{A} \tilde{C}_{Y_{j+\Delta-1}} \tilde{A} \dots \tilde{C}_{Y_{j+1}} \tilde{A}$$

$$\hat{\Pi}_{j,j} = \begin{bmatrix} \Pr(X_j=1, X_j=1|Y_0, \dots, Y_j) \\ \Pr(X_j=1, X_j=2|Y_0, \dots, Y_j) \\ \vdots \\ \Pr(X_j=2, X_j=1|Y_0, \dots, Y_j) \\ \vdots \\ \Pr(X_j=N, X_j=N-1|Y_0, \dots, Y_j) \\ \Pr(X_j=N, X_j=M|Y_0, \dots, Y_j) \end{bmatrix} = \begin{bmatrix} \Pi_{j,j}(1) \\ 0 \\ \vdots \\ \Pi_{j,j}(2) \\ \vdots \\ 0 \\ \Pi_{j,j}(N) \end{bmatrix}$$

$\hat{\Pi}_{j,j}$ 를 구하기 위해, HMM filter를 통하여  $\tilde{\Pi}_{j,j}$ 를 구하게 된다. 이를 통해 과거에 대한 확률 정보를 가지게 되며,  $U_{j+1,j+\Delta}$ 를 통해  $j+1$ 부터  $j+\Delta$ , 즉  $k$ 까지의 정보를 추가하게 된다. 이렇게 함으로서 filter의 경우와 비교하여  $\Delta$ 만큼의 미래의 데이터를 이용하게 되고, 전체적인 상태 변화에 대한 안정성을 줄 수 있다.

#### 4. 관성 항법 알고리즘

관성 항법을 위한 계산을 하기 위해서는 우선 좌표축의 정리가 필요하다. 좌표축 정리는 앞에서 설명한 그림 3에서와 같이 구성하였다. 이 때, 항법 좌표계(Navigation Coordinate)는  $x$ 축을 북쪽으로,  $z$ 축을 위의 방향으로 정하는 항공용 좌표축을 따랐다.

자세를 계산하기 위하여 쿼터니언(quaternion)을 사용하였으며, 쿼터니언  $q$ 는 다음과 같이 정의한다[8].

$$q = [q_0 \ q_1 \ q_2 \ q_3]$$

이때  $q$ 와 회전 행렬(rotation matrix)와의 관계는 다음과 같다.

$$C(q) = \begin{bmatrix} 2q_0^2 + 2q_1^2 - 1 & 2q_1q_2 + 2q_0q_3 & 2q_1q_3 - 2q_0q_2 \\ 2q_1q_2 - 2q_0q_3 & 2q_0^2 + 2q_2^2 - 1 & 2q_2q_3 + 2q_0q_1 \\ 2q_1q_3 + 2q_0q_2 & 2q_2q_3 - 2q_0q_1 & 2q_0^2 + 2q_3^2 - 1 \end{bmatrix}$$

주어진 벡터  $\vec{p}$ 가 있을 때, 동체 좌표계(body coordinate)와 항법 좌표계는 각각 약어로  $b, n$ 으로 나타내며, 동체 좌표계와 항법 좌표계 사이의 관계는 다음과 같다.

$$\vec{p}_b = C(q)\vec{p}_n = q^* \vec{p}_n q$$

여기에서 \*는 켈레 복소수(complex conjugate)를 의미한다.

$r_n, v_n, \omega_b, a_b, g$ 는 다음과 같이 정의한다.

- $r_n \in R^3$  : 항법 좌표계 상의 위치
- $v_n \in R^3$  : 항법 좌표계 상의 속도
- $\omega_b \in R^3$  : 동체 좌표계 상의 각속도
- $a_b \in R^3$  : 중력가속도를 제외한 동체 좌표계 상의 가속도
- $g \in R^3$  : 항법 좌표계 상에 작용되는 중력 가속도

관성 항법을 계산하기 위한 기본적인 식은 다음과 같다.

$$\begin{aligned} \dot{\hat{q}} &= \frac{1}{2} \Omega(\omega_b) \hat{q} \\ \dot{v}_n &= a_n = C'(q) a_b \\ \dot{r}_n &= v_n \end{aligned} \quad (7)$$

이 때,

$$\Omega([abc]) = \begin{bmatrix} 0 & -a & -b & -c \\ a & 0 & c & -b \\ b & -c & 0 & a \\ c & b & -a & 0 \end{bmatrix}$$

본 논문에서  $q, v_n, r_n$ 을 식 (7)과 영속도 업데이트를 통한 보상 칼만 필터로의 오차 보정[9]을 통하여 계산한다.

#### 4.1 $\hat{q}, \hat{v}_n, \hat{r}_n$ 의 계산

$\hat{q}, \hat{v}_n, \hat{r}_n$ 은 각각  $q, v_n, r_n$ 의 추정된 값을 의미한다. 계산을 위하여 다음과 같이 변수들을 정의한다.

- $y_g \in R^3$  : 자이로스코프 출력
- $b_g \in R^3$  : 자이로스코프의 바이어스
- $v_g \in R^3$  : 자이로스코프의 센서 노이즈
- $y_a \in R^3$  : 가속도 센서의 출력
- $b_a \in R^3$  : 가속도 센서의 바이어스
- $v_a \in R^3$  : 가속도 센서의 센서 노이즈

이 때,  $y_g$ 와  $y_a$ 는 다음과 같이 나타낼 수 있다.

$$\begin{aligned} y_g &= \omega_b + b_g + v_g \\ y_a &= a_b + b_a + v_a + C(q)g \end{aligned} \quad (8)$$

오차가 없거나 아주 작다는 가정 하에 식 (7)의 첫 번째 방정식을 다음과 같이 정리할 수 있다.

$$\dot{\hat{q}} = \frac{1}{2} \Omega(y_g) \hat{q} \quad (9)$$

$y_g$ 와  $\omega_b$ 가 같지 않으므로, 식 (9)는 오차를 가지게 되며, 이 오차는 추후 보상 칼만 필터를 통하여 보상된다. 식 (9)를 테일러의 2차 근사를 통해 정리하면,

$$\begin{aligned} \hat{q}_{k+1} &\approx \hat{q}_k + \dot{\hat{q}}_k T + \frac{T^2}{2} \ddot{\hat{q}}_k \\ &= \left( I + \frac{3}{4} \Omega_k T - \frac{1}{4} \Omega_{k-1} T - \frac{1}{8} \|y_{gk}\|_2^2 T^2 \right) \hat{q}_k \end{aligned}$$

이 때,

$$\begin{aligned} \hat{q}_k &= \hat{q}(kT) \\ \Omega_k &= \Omega(y_{gk}) \end{aligned}$$

식 (8)과 식 (7)의 아래 두 개의 방정식을 통하여

$$\begin{aligned} \dot{\hat{v}} &= C'(q) y_a - g \\ \dot{\hat{r}} &= \hat{v} \end{aligned} \quad (10)$$

#### 4.2 보상 칼만 필터(Complementary Kalman filter)

식(9)와 식 (10)은  $y_g, y_a$ 안에 센서 노이즈 성분이 존재하기 때문에 정확하지 않다. 더구나 이산화를 위한 근사화를 통해 오차성분이 더 커지게 된다. 이 오차들을 보상 칼만 필터를 통해 보상하게 된다.

$\hat{q}, \hat{r}, \hat{v}$ 에 포함된 오차  $q_e, r_e, v_e$ 를 각각 다음과 같이 정의한다.

$$\begin{aligned} q_e &= \hat{q}^* \otimes q \\ r_e &= r - \hat{r} \\ v_e &= v - \hat{v} \end{aligned} \quad (11)$$

여기서  $\otimes$ 는 쿼터니언 곱셈(quaternion multiplication)을 의미한다. 보상 칼만 필터에서  $q, r, v$ 를 직접 추정하지 않고,  $q_e, r_e, v_e$ 를 추정하게 된다.

만약  $q_e$ 가 작다면, 다음과 같이 근사화 시킬 수 있다.

$$q_e \approx \begin{bmatrix} 1 \\ q_e \end{bmatrix}, \quad q_e \in R^3 \quad (12)$$

$q_e$  대신에  $\bar{q}_e$ 를 사용하면, 추정하고자 하는 state를 다음과 같이 구성할 수 있다.

$$x = \begin{bmatrix} q_e \\ b_g \\ r_e \\ v_e \\ b_a \end{bmatrix}, \quad x \in R^{15}$$

미분 방정식으로 유도하여 근사화를 시키면 다음과 같다.

$$\dot{x}(t) = Ax(t) + \begin{bmatrix} -\frac{1}{2}v_g \\ w_{b_g} \\ 0 \\ -\dot{C}(\hat{q}) \\ w_{b_a} \end{bmatrix} \quad (13)$$

이 때,

$$A = \begin{bmatrix} -[y_g] & -\frac{1}{2}I & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & I & 0 \\ -2\dot{C}(\hat{q})[y_a] & 0 & 0 & 0 & -\dot{C}(\hat{q}) \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

행렬 내부의 [ ]는 외적 계산을 위해 변환된 형태의 행렬을 의미하며, 다음과 같다.

$$A \times B = [A]B = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}$$

노이즈 성분인  $w_{b_g}$ ,  $w_{b_a}$ 는 보상되어지는 프로세스 노이즈로 시간에 따라 느리게 변화하는 자이로스코프와 가속도 센서의 바이어스 오차이다. 식 (13)에 나오는 모든 에러는 uncorrelate, 영평균 white gaussian 으로 간주한다.

속도 측정 성분인  $y_v$ 는 다음과 같이 정의한다.

$$y_v = v + v_v \\ = \hat{v} + v_e + v_v$$

이 때,  $v_v$ 는 측정 오차이며,  $E[V_v(t)V_v(s)] = R_v\delta(t-s)$ 이다.  $y_v$ 가 사용가능한 값일 때, 다음 관계를 이용한다.

$$y_v - \hat{v} = v_e + v_v \\ = [0 \ 0 \ 0 \ I \ 0]x + v_v \quad (14)$$

### 4.3 Zero velocity updating

영속도 구간에 대한 추정은 이미 앞의 장에서 HMM smoothing을 통해서 설명되었다. 이를 통해 영속도 구간(상태 2)에서는 칼만 필터의 (13)과 (14)에서  $y_v$ 의 값이 0이 된다.

## 5. 실험

실험을 위해서 장착한 관성 센서로는 XSens의 MTi 모델을 사용하였다. 이 제품은 3축의 가속도계와 3축의 자이로스코프, 3축의 자기 센서가 있으며, 센서 데이터는 100Hz로 업데이트 된다. 사용된 센서의 간단한 사양은 표 4와 같다.

표 5 Mti 모듈의 사양

Table 4 Specification of using MTi module

	rate of turn	acceleration	magnetic field
Dimensions	3 axes	3 axes	3 axes
Full scale	$\pm 300 \text{ }^\circ/s$	$\pm 50 \text{ m/s}^2$	$\pm 750 \text{ mGauss}$
Bias stability	$5 \text{ }^\circ/s$	$0.02 \text{ m/s}^2$	0.5 mGauss
Bandwidth	40Hz	30Hz	10Hz

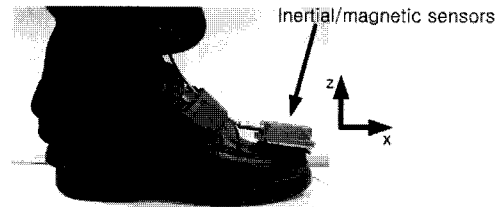


그림 7 센서 부착 위치

Fig. 7 Sensors position

실험은 초기 1초간 가만히 서 있다가 약 3미터의 거리를 'ㄱ'자 형태로 이동하였다. 그림 8과 9는 가속도계와 자이로스코프의 출력이다.

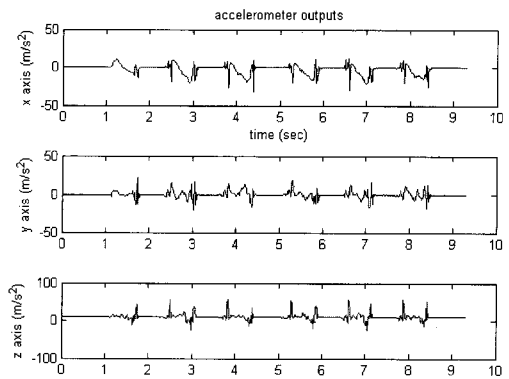


그림 8 가속도계 출력

Fig. 8 Accelerometer outputs

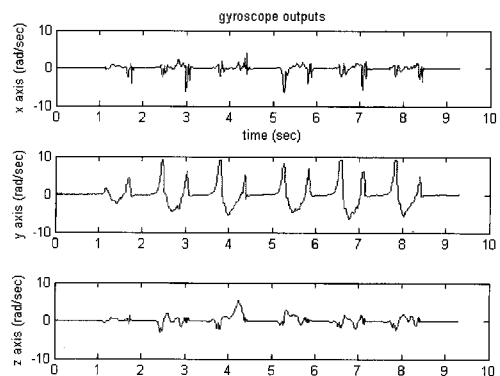


그림 9 자이로스코프 출력

Fig. 9 Gyroscope outputs

그림 10에는 관성센서모듈의 출력을 통한 HMM filter와 smoothing을 이용한 걸음 상태를 추정하였다. 그림 10a는 filter를 이용한 상태 추정 결과이며, 그림 10b는 smoothing을 이용하여 상태 추정한 결과이다. 처음 1초 정도 서 있다가(상태 2) 걷기 시작하여 일정 걸음 후에 정지하였으므로, 그 상태는 2-3-4-1-2...-1-2의 형태로 나타나게 된다. filter의 경우 몇몇 구간에서 다른 상태로 넘어갔다가 다시 돌아오는 결과를 확인할 수 있는 반면에 smoothing의 경우에는 Filter의 결과보다 좀 더 안정적인 상태 추정 하는 것을 확인할 수 있다.

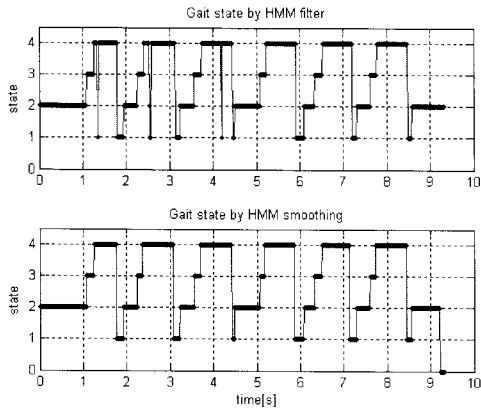


그림 10 걸음 상태 결과 1  
Fig. 10 Gait state outputs 1

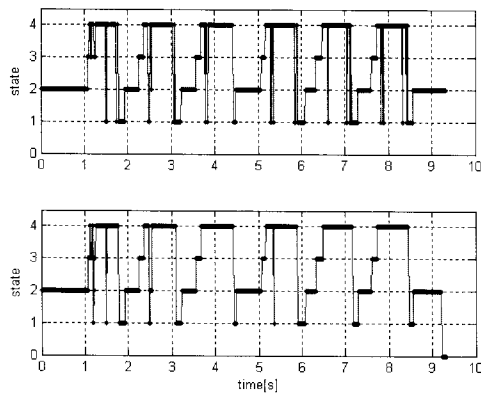


그림 11 걸음 상태 결과 2  
Fig. 11 Gait state outputs 2

그림 11은 상태 변환 행렬을 식(1)에서 식 (15)와 같이 수정하였을 때의 결과이다. 이 값은 3장에서 설명한 방식대로 결정 되었으며, 다만 자기 자신으로 돌아가는 확률에 대한 다른 상태로의 변화에 대한 확률을 높여주었다.

$$A = \begin{bmatrix} 0.6 & 0 & 0 & 0.3 \\ 0.3 & 0.6 & 0 & 0.05 \\ 0 & 0.3 & 0.6 & 0.05 \\ 0.1 & 0.1 & 0.4 & 0.6 \end{bmatrix} \quad (15)$$

식 (1)에 비하여 이전 상태로 유지되는 확률이 낮아진 반

면, 다른 상태로 가는 확률을 높였다. 그로인한 다른 상태로의 잦은 변화를 보이는 것을 확인할 수 있다. 상태 변환 행렬은 구성되는 시스템의 특성에 맞게 구서하면 된다. 본 시스템과 같이 안정적인 결과를 원할 때는 이전 상태를 유지하는 항의 값을 높게 잡고, 좀 더 빠르고 정확한 상태 변화를 원할 때는 전체적인 확률 값을 비슷하게 잡으면 된다.

그림 12와 13은 HMM smoothing을 통해 구한 걸음 상태를 이용하여 영속도 보정을 통한 위치 추정 결과이다. 그림 12는 결과 중에 z축의 결과만을 나타낸 것이고, 그림 13은 전체적인 궤적을 3차원으로 나타낸 것이다. 실험을 평평한 장소에서 한 만큼 z축 결과의 최저점이 일정 값으로 유지되어야 함에도 각 걸음마다 그 결과가 조금씩 변하는 것을 확인할 수 있지만, 일반적인 사람의 보폭이 1m 근방인데 비해, 수렴되는 z축의 변화는 최대 0.1m 정도로 비교적 적은 오차라 할 수 있다. 이러한 오차의 원인은 장착되어 있는 관성센서가 발가락의 움직임이나 발이 땅에 닿는 순간 센서의 흔들림에 의한 것으로 생각되어진다. 그림 13를 통한 전체적인 궤적은 비교적 정확하게 나온 것으로 볼 수 있다. 표 5는 10여초 동안 일정 거리를 이동 후에 원래의 자리로 돌아갔을 때, 생기는 오차의 RMS를 정리한 것이다. 평균적으로 약 0.1216m의 오차를 보였다. 하지만 이동 후에 정확히 처음 지점으로 돌아간다는 보장을 할 수 없기에, 이 정도의 오차는 크지 않다고 판단한다.

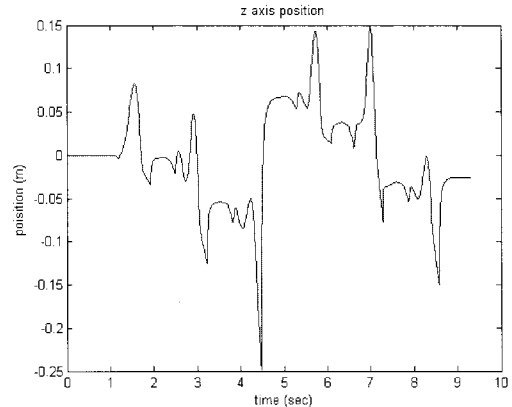


그림 12 z축 결과  
Fig. 12 Z axis position

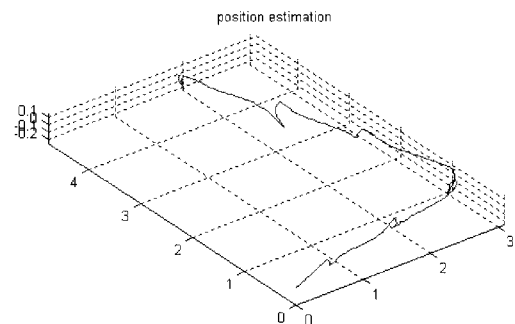


그림 13 위치 추정 결과  
Fig. 13 Position estimation result



표 6 실험 결과

Table 5 Result of experiment

	RMS 오차 [m]
1	0.1919
2	0.0509
3	0.2456
4	0.0783
5	0.1123
6	0.0456
7	0.1091
8	0.1992
9	0.0757
10	0.1079

### 6. 결 론

본 논문에서는 보행용 관성항법 장치에 대한 영속도 보정을 위한 알고리즘들을 제안하였다. 걸음 상태를 구분하기 위하여 HMM smoothing을 사용하였고, 이를 통해 영속도 구간을 추정할 수 있었다. 이를 토대로 보상 칼만 필터를 통한 오차 추정을 통해 이를 보상함으로써 최종적으로 실제 이동한 위치를 추정할 수 있었다.

현재는 영속도 구간(상태 2)에 한해서만 오차 보정을 하고 있지만, 상태 1과 3의 경우에 발의 한 부분이 땅에 닿아 있기 때문에 실질적으로 위치의 변화는 없는 상태에서 약간의 자세의 변화만을 보이게 된다. 이러한 부분을 위한 알고리즘 개발을 통해 좀 더 정확한 보행용 관성 항법 장치를 개발하는 것이 앞으로의 연구 과제이다.

#### 감사의 글

이 논문은 2007년 울산대학교의 연구비에 의하여 연구되었습니다.

#### 참 고 문 헌

[1] L. Ojeda, and J. Borenstein, "Non-GPS Navigation for Security Personnel and First Responders," The Journal of Navigation, vol. 60, no. 3, pp. 391-407, 2007.

[2] E. Foxlin, "Pedestrian Tracking with Shoe-mounted Inertial Sensors," IEEE Computer Graphics and Applications, vol. 25, no. 6, pp. 38-46, 2005.

[3] X. Yun, E. R. Bachmann, H. Moore IV, and J. Calusdian, "Self-contained Position Tracking of Human Movement Using Small Inertial/magnetic Sensor Modules," in Proc. IEEE Int. Conf. Robotics and Automation, pp. 2526-2533, 2007.

[4] K. Kong and M. Tomizuka, "Smooth and Continuous Human Gait Phase Detection Based on Foot Pressure Patterns," in Proc. IEEE Int. Conf. Robotics and Automation, pp. 3678-3683, 2008.

[5] B. D. O. Anderson, "From Wiener to Hidden Markov Models," IEEE Control Systems Magazine, vol. 19, no. 3, pp. 41-51, 1999.

[6] L. Shue, B. D. O. Anderson, and S. Dey, "Exponential Stability of Filters and Smoothers for Hidden Markov Models," IEEE Trans. Acoustics, Speech, and Signal Processing, vol. 46, no. 8, pp. 2180-2194, 1988.

[7] J. Perry, Gait Analysis: Normal and Pathological Function. SLACK Incorporated, 1992.

[8] J. B. Kuipers, Quaternions and rotation sequences: a primer with applications to orbits, aerospace, and virtual reality. New Jersey: Princeton University Press, 1999.

[9] R. G. Brown and P. Y. C. Hwang, Introduction to Random Signals and Applied Kalman Filtering, 3rd ed. New York: John Wiley & Sons, 1997.

### 저 자 소 개



#### 박 상 경 (朴 賞 卿)

1976년 9월 28일생. 2002년 울산대 공대 제어계측공학과 졸업. 2004년 동 대학원 전기전자정보시스템공학과 졸업(석사). 2006년~현재 동 대학원 전기전자정보시스템공학과 박사과정

E-mail : damiro76@hotmail.com



#### 서 영 수 (徐 榮 秀)

1967년 10월 6일생. 1990년 서울대 공대 제어계측과 졸업. 1992년 동 대학원 제어계측과 졸업(석사). 1997년 동경대학교 계측공학과 졸업(공학박사). 2000~현재 울산대학교 전기전자정보시스템공학부 교수

E-mail : suh@ieee.org