Pedestrian Inertial Navigation with Gait Phase Detection Assisted Zero Velocity Updating

Young Soo Suh
Dept. of Electrical Engineering
Univ. of Ulsan
Ulsan, Korea 680-749
Email: yssuh@ulsan.ac.kr

Sangkyung Park
Dept. of Electrical Engineering
Univ. of Ulsan
Ulsan, Korea 680-749
Email: damiro76@hotmail.com

***Abstract*—An inertial navigation system for pedestrian position tracking is proposed, where the position is computed using inertial and magnetic sensors on shoes. Using the fact that there is a zero velocity interval in each stride, estimation errors are reduced. 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 gait state is estimated using the hidden Markov model filter. With this gait estimation, the zero velocity interval is more accurately estimated, which helps to reduce the position estimation error.**

I. Introduction

A pedestrian inertial navigation system tracks the location of a person using inertial sensors. The navigation system is useful for emergency responders, security personnel, and wide range augmented reality applications. For example, if a firefighter becomes unconscious in the line of duty, the location transmitted from the navigation system is useful for the rescue.

In the outdoor, GPS can be used for such a navigation system. In the indoor environment, pedestrian navigation systems usually require some infrastructure. For example, to use indoor wireless navigation systems [1], a beacon node must be installed in the building. In the inertial navigation system, the location can be computed from inertial sensors (3 accelerometers and 3 gyroscopes); it does not require any infrastructure in the building.

Pedestrian inertial navigation systems are developed in [2], [3], [4], where inertial sensors are attached on shoes. The inertial navigation algorithms of 3 systems are similar and they employ the same key concept: the zero velocity updating method. Velocity and position of every inertial algorithm inevitably diverges as time goes by. In the zero velocity updating method, the velocity error can be reset if we know the velocity is zero in a certain interval. When a person is walking normally, there is an interval (usually 0.2 - 0.4 seconds) when velocity of a foot is zero. This interval starts when a foot completely contacts the ground and ends when a foot is starting to rise. This zero velocity interval appears in each walking step. Thus the velocity errors are reset almost periodically. Since the velocity error does not accumulate, the position error growth rate can be reduced.

Since the zero velocity updating plays a key role in reducing position errors, it is important to detect the zero velocity interval accurately. In [2], if differences of accelerometer values and gyroscope values are all smaller than the threshold value for longer than the predetermined time, it is assumed that a foot is in the zero velocity. In [3], a similar detection algorithm is used where accelerometer values and gyroscope values are compared with the threshold value instead of differences of differences of accelerometer values and gyroscope values. In [4], the norm of 3 gyroscope values are used for the zero velocity interval detection.

In this paper, we propose a new zero velocity detection algorithm using gait phase detection. The human gait phase is estimated using a hidden Markov model (HMM) filter [5], where inertial sensors and force sensors on shoes are used as outputs to the HMM filter.

II. Inertial Personal Navigation System

The proposed navigation system (see Fig. 1) consists of inertial/magnetic sensors (XSens MTi) and force sensors (Tekscan FlexForce A201). XSens Mti has 3 axis accelerometers, 3 axis gyroscopes, and 3 axis magnetic sensors. The FlexiForce force sensor is a thin and flexible printed circuit sensor. Four force sensors are attached to a shoe as in Fig. 1. It turns out that outputs of force sensor 2 and 3 are almost similar, thus force sensor 3 value is not used.

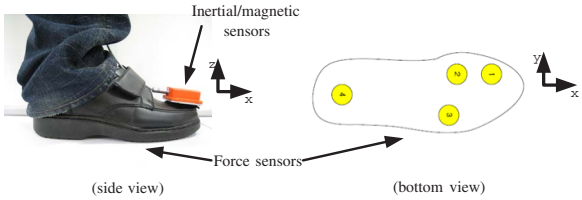

Fig. 1. Proposed navigation system

III. Gait Phase Detection

When a person is walking, it is known that each stride contains eight functional patterns, which are called eight gait phases (the following descriptions of gait phases are from [6]):

- Phase 1 (initial contact) : This phase includes the moment when the foot just touches the floor. Floor contact is usually made with the heel.

978-1-4244-2713-0/09/$25.00 ©2009 IEEE

- Phase 2 (loading response) : The phase begins with initial floor contact and continues until the other foot is lifted for swing.
- Phase 3 (mid stance) : The is the first half of the single limb support interval.
- Phase 4 (terminal stance) : It begins with heel rise and continues until the other foot strikes the ground.
- Phase 5 (pre-swing) : It begins with initial contact of the opposite limb.
- Phase 6 (initial swing) : The foot is lifted.
- Phase 7 (mid swing) : The swinging limb is opposite the stance limb.
- Phase 8 (terminal swing) : The final phase of swing and it ends when the foot strikes the floor.

We are interested with gait phases since identifying gait phases helps identifying the zero velocity intervals. Among gait phases, we are mainly interested with phases where foot velocity is zero or nearly zero. In this regard, identifying whether the gait phase is in phase 6 or phase 7 is not important since in both phases the velocity is not near zero. Thus we simplify eight gait phases into 4 gait states (see Fig. 2) by grouping eight gait phases as follows:

- state 1 : gait phase 1 and 2
- state 2 : gait phase 3
- state 3 : gait phase 4 and 5
- state 4 : gait phase 6, 7, and 8

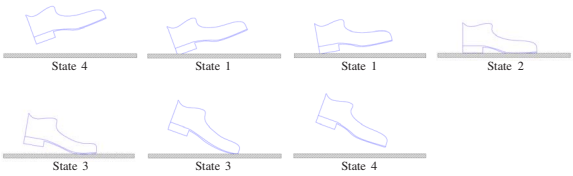

Fig. 2. Gait States

Note that the gait state 2 is a zero velocity interval.

We will identify the gait state using the finite state HMM filter [5]. We assume there is a Markov process $X_0, X_1, \cdots, X_k$ and X_k can assume one of the four gait states (1,2,3, and 4). The state transition probability matrix A (see the state transition diagram in Fig. 3) is assumed as follows:

$$A = \begin{bmatrix} 0.8 & 0 & 0 & 0.15 \\ 0.15 & 0.8 & 0 & 0.025 \\ 0 & 0.15 & 0.8 & 0.025 \\ 0.05 & 0.05 & 0.2 & 0.8 \end{bmatrix} \quad (1)$$

where (i, j)-th element of A is given by

$$A_{ij} \triangleq \Pr\left[X_{k+1} = i \,|\, X_k = j\right].$$

Four sensor outputs (3 force sensors and one y axis gyroscope) are used to define the output process Y_k. The force sensors are quantized into 2 levels and the y axis gyroscope value is quantized into 3 levels as given in Table I.

In Table I, there are $2 \times 2 \times 2 \times 3 = 24$ combinations. Each combination is assigned output value from 1 to 24 as in Table II.

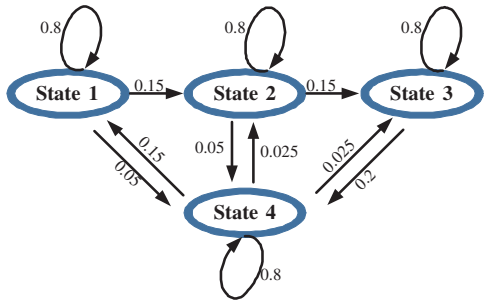

Fig. 3. State transition diagram

sensors	1	2	3
force sensor 1	> 0.5	< 0.5	
force sensor 2	> 1.5	< 1.5	
force sensor 4	> 8.2	< 8.2	
y axis gyroscope	$\lvert\cdot\rvert < 0.15$	> 0.15	< 0.15

TABLE I
SENSOR OUTPUTS QUANTIZATION

The link between the state process X_k and the output process Y_k is defined by a matrix C where $C_{mn} = \Pr\left[Y_k = m \,|\, X_k = n\right]$. In this paper, we use the following C matrix.

$$C = \begin{bmatrix} 0 & 0.693 & 0 & 0.012 \\ 0 & 0.001 & 0.025 & 0.012 \\ 0.025 & 0.001 & 0.025 & 0.012 \\ 0 & 0.100 & 0.05 & 0.012 \\ 0.05 & 0.000 & 0.15 & 0.012 \\ 0.05 & 0.000 & 0.25 & 0.012 \\ 0.01 & 0.010 & 0.01 & 0.012 \\ 0.01 & 0.010 & 0.01 & 0.012 \\ 0.01 & 0.010 & 0.01 & 0.012 \\ 0 & 0.000 & 0.18 & 0.012 \\ 0 & 0.000 & 0.1 & 0.012 \\ 0 & 0.000 & 0.19 & 0.012 \\ 0.1 & 0.150 & 0 & 0.012 \\ 0.11 & 0.000 & 0 & 0.012 \\ 0.2 & 0.000 & 0 & 0.012 \\ 0 & 0.025 & 0 & 0.012 \\ 0 & 0.000 & 0 & 0.012 \\ 0 & 0.000 & 0 & 0.012 \\ 0.1 & 0.000 & 0 & 0.012 \\ 0.135 & 0.000 & 0 & 0.012 \\ 0.2 & 0.000 & 0 & 0.012 \\ 0 & 0.000 & 0 & 0.250 \\ 0 & 0.000 & 0 & 0.250 \\ 0 & 0.000 & 0 & 0.250 \end{bmatrix} \quad (2)$$

For example, the first column of C is $\begin{bmatrix} 0.000 & 0.693 & 0.000 & 0.012 \end{bmatrix}$, which states that if output Y_k is 1, then the probability that the gait state is 2 is the greatest (0.693). Since the output 1 corresponds to the case that all force sensor values are large (i.e., floor contacted)

Y_k	force 1	force 2	force 3	gyroscope
1	1	1	1	1
2	1	1	1	2
3	1	1	1	3
4	1	1	2	1
5	1	1	2	2
6	1	1	2	3
7	1	2	1	1
8	1	2	1	2
9	1	2	1	3
10	1	2	2	1
11	1	2	2	2
12	1	2	2	3
13	2	1	1	1
14	2	1	1	2
15	2	1	1	3
16	2	1	2	1
17	2	1	2	2
18	2	1	2	3
19	2	2	1	1
20	2	2	1	2
21	2	2	1	3
22	2	2	2	1
23	2	2	2	2
24	2	2	2	3

TABLE II
OUTPUT PROCESS DEFINITION

and the gyroscope value is small (i.e., no movement), the gait state 2 is most likely.

In the HMM filter, $\Pi_{k|k}$ is a probability vector whose i-th element is given by $\Pr[X_k = i \,|\, Y_0, \cdots, Y_k]$. Similarly, $\Pi_{k+1|k}$ is a probability vector whose i-th element is given by $\Pr[X_{k+1} = i \,|\, Y_0, \cdots, Y_k]$. For example, if $\Pi_{k|k} = [\ 0.1 \quad 0.1 \quad 0.7 \quad 0.1\ ]'$, then given the output $Y_0, \cdots, Y_k$, the probability that the gait state is 3 is the greatest (0.7). Thus we can assume that the gait state is 3.

The HMM filter is given by

$$\begin{aligned} \Pi_{k+1|k} &= A\Pi_{k|k} \\ \Pi_{k+1|k+1} &= \frac{1}{[1\cdots1]C_{Y_{k+1}}\Pi_{k+1|k}} C_{Y_{k+1}}\Pi_{k+1|k} \end{aligned} \tag{3}$$

where $C_{Y_{k+1}} = \mathrm{Diag}(C_{p1}, \cdots, C_{p4})$ when $Y_{k+1} = p$.

IV. INERTIAL NAVIGATION ALGORITHM

The body coordinate coincides with (x, y, z) axis coordinate of the inertial sensors. The navigation coordinate is defined as in Fig. 4; x axis is in the north direction and z axis is upward, which is different from the navigation coordinate used in aeronautics. For a given vector p, we sometimes use subscript b (body) and n (navigation) to emphasize that a vector is expressed in a particular coordinate.

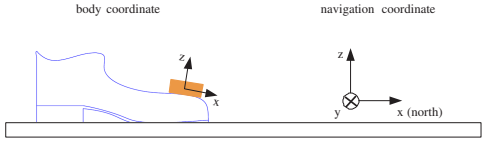

Fig. 4. Body coordinate and navigation coordinate

The quaternion $q = [\ q_0 \quad q_1 \quad q_2 \quad q_3\ ]'$ is a vector that defines attitude (see [7] for details). The relationship between q and the corresponding rotation matrix is given by

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

Note that p_b and p_n are related as follows:

$$p_b = C(q)p_n = q^* p_n q$$

where q^* is the complex conjugate of q.

Let r_n, v_n, ω_b, a_b, and g be defined as follows:

- $r_n \in R^{3\times1}$: position in the navigation coordinate
- $v_n \in R^{3\times1}$: velocity in the navigation coordinate
- $\omega_b \in R^{3\times1}$: body angular rates
- $a_b \in R^{3\times1}$: body acceleration without gravitational acceleration
- $g \in R^{3\times1}$: gravitational acceleration in the navigation coordinate

The basic equations for inertial navigation are given as follows:

$$\begin{aligned} \dot{q} &= \tfrac{1}{2}\Omega(\omega_b)q \\ \dot{v}_n &= a_n = C'(q)a_b \\ \dot{r}_n &= v_n \end{aligned} \tag{4}$$

where

$$\Omega(\omega) = \begin{bmatrix} 0 & -\omega_x & -\omega_y & -\omega_z \\ \omega_x & 0 & \omega_z & -\omega_y \\ \omega_y & -\omega_z & 0 & \omega_x \\ \omega_z & \omega_y & -\omega_x & 0 \end{bmatrix}.$$

In this paper, we compute q, v_n, and r_n using (4) and during the zero velocity updating, errors are compensated using the complementary Kalman filter [8].

A. $\hat{q}$, $\hat{v}_n$, and $\hat{r}_n$ computation

Let $\hat{q}$, $\hat{v}$, and $\hat{r}$ be estimated values of q, v_n, and r_n, respectively. Let y_g, y_a, b_g, v_g, b_a, and v_a be defined as follows:

- $y_g \in R^{3\times1}$: gyroscope output
- $y_a \in R^{3\times1}$: accelerometer output
- $b_g \in R^{3\times1}$: gyroscope bias
- $v_g \in R^{3\times1}$: gyroscope sensor noise
- $b_a \in R^{3\times1}$: accelerometer bias
- $v_a \in R^{3\times1}$: accelerometer sensor noise

Note that y_g and y_a are given as follows:

$$\begin{aligned} y_g &= \omega_b + b_g + v_g \\ y_a &= a_b + b_a + v_a + C(q)g. \end{aligned} \tag{5}$$

From (5) and the first equation of (4), we have

$$\dot{\hat{q}} = \frac{1}{2}\Omega(y_g)\hat{q}. \tag{6}$$

Since $y_g \neq \omega_g$, (6) is not exact. This error and other errors will be compensated later in the complementary Kalman filter. We compute (6) using the second order Taylor approximation:

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

where $\hat{q}_k \triangleq \hat{q}(kT)$.

From (5) and the last two equations of (4), we have

$$\begin{array}{rcl} \dot{\hat{v}} & = & C'(\hat{q})y_a - g \\ \dot{\hat{r}} & = & \hat{v}. \end{array} \tag{7}$$

B. Complementary Kalman filter

Equations (6) and (7) are not exact due to sensor errors in y_g and y_a. Furthermore, since (6) and (7) are approximated using the discrete equations, the overall errors become larger. These errors will be compensated using the complementary Kalman filter.

Let q_e, r_e, and v_e are defined as follows:

$$\begin{array}{rcl} q_e & \triangleq & \hat{q}^* \otimes q \\ r_e & \triangleq & r - \hat{r} \\ v_e & \triangleq & v - \hat{v} \end{array} \tag{8}$$

where $\otimes$ is the quaternion multiplication. We can consider q_e, r_e and v_e as errors in $\hat{q}$, $\hat{r}$, and $\hat{v}$, respectively. In the complementary Kalman filter, we are trying to estimate q_e, r_e, and v_e instead of directly estimating q, r, and v.

If the error q_e is small, it can be approximated by

$$q_e \approx \begin{bmatrix} 1 \\ \bar{q}_e \end{bmatrix}. \tag{9}$$

Thus in the complementary Kalman filter, $\bar{q}_e$ will be used instead of q_e.

The state $x \in R^{15\times 1}$ for the Kalman filter is defined by

$$x \triangleq \begin{bmatrix} \bar{q}_e \\ b_g \\ r_e \\ v_e \\ b_a \end{bmatrix}.$$

After some approximation (the details are omitted), we can obtain

$$\dot{x}(t) = Ax(t) + \begin{bmatrix} -\frac{1}{2}v_g \\ w_{b_g} \\ 0 \\ -C'(\hat{q})v_a \\ w_{b_a} \end{bmatrix} \tag{10}$$

where A is given by

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

The noise w_{b_g} and w_{b_a} are process noises for compensation of slowly time-varying gyroscope and accelerometer bias. As usual, we assume all noises in (10) are uncorrelated, zero mean white Gaussian.

The velocity measurement y_v is given by

$$\begin{array}{rcl} y_v & = & v + v_v \\ & = & \hat{v} + v_e + v_v \end{array}$$

where v_v is the measurement noise and $E[V_v(t)V_v(s)] = R_v\delta(t-s)$.

When y_v is available, we use $y_v - \hat{v}$ as an output to the complementary Kalman filter, where the output equation is given by

$$\begin{array}{rcl} y_v - \hat{v} & = & v_e + v_v \\ & = & \begin{bmatrix} 0 & 0 & 0 & I & 0 \end{bmatrix} x + v_v. \end{array} \tag{11}$$

C. Zero velocity updating

One of key difficulties in applying the zero velocity algorithm is to know when is the zero velocity time. In this paper, we have already identified the gait state using the HMM filter. So the zero velocity interval detection is easy. We just assume that the zero velocity interval is the gait state 2. Once zero velocity interval is identified, we can apply the Kalman filter to (10) and (11) with $y_v = 0$.

V. Experiments

In the first experiment, a person walked forward and then returned to the starting position. Force sensor outputs and inertial sensor outputs are given in Fig. 5 and Fig. 6, respectively. The gait state estimated using (3) is given in Fig. 7, where we can see that gait states are almost periodically changing such as $2 \rightarrow 3 \rightarrow 4 \rightarrow 1 \rightarrow 2 \rightarrow \cdots$. When the gait state is 2, we applied the zero velocity updating algorithm. The position estimation is given in Fig. 7 and Fig. 8. In Fig. 7, z axis component of the position is given. Since the test is done on the flat floor, at each stride, the z axis position should return to 0. Due to various errors, the z axis position value does not return to 0 but stays near 0. In Fig. 8, the position is given in the 3 dimensional plot. We can see a person is walking forward and then returns to the starting position. In the second experiment (see Fig. 9), a person walked forward, made 90 degree turn, walked forward, and returned to the starting position. The final position value for each experiment is given by $r_{first} = [\ -0.2867 \quad 0.0323 \quad 0.0448\ ]'$ and $r_{second} = [\ 0.2422 \quad -0.6645 \quad 0.3300\ ]$. The unit is meter.

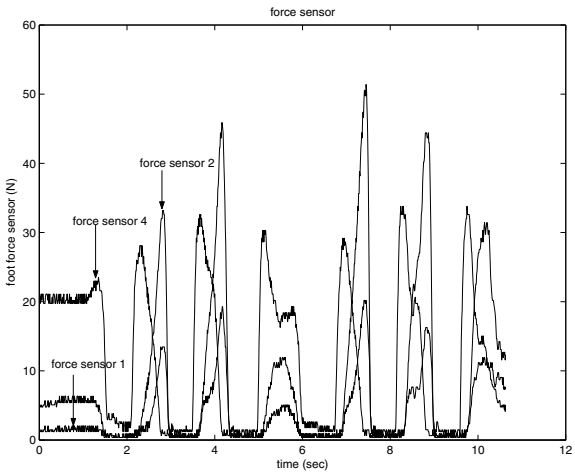

Fig. 5. Foot force sensor outputs

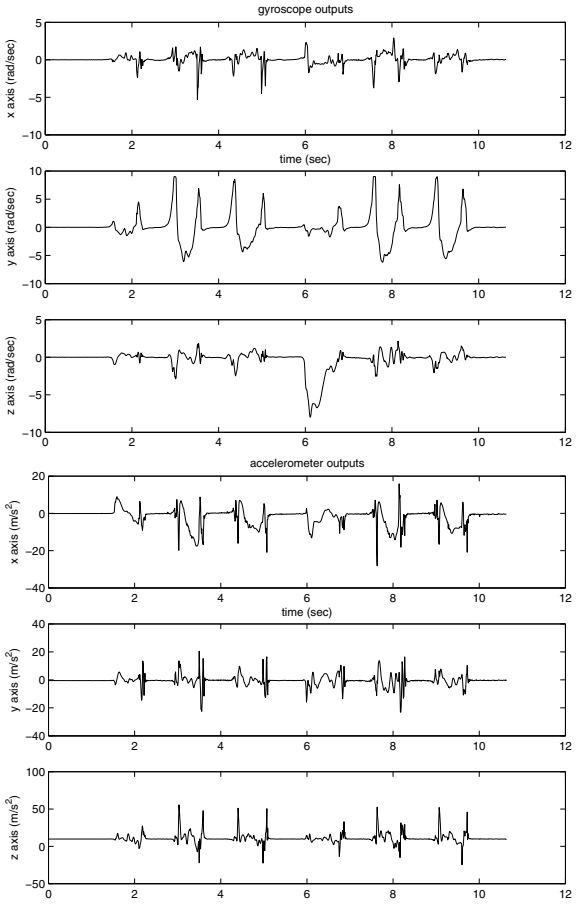

Fig. 6. Inertial sensor outputs

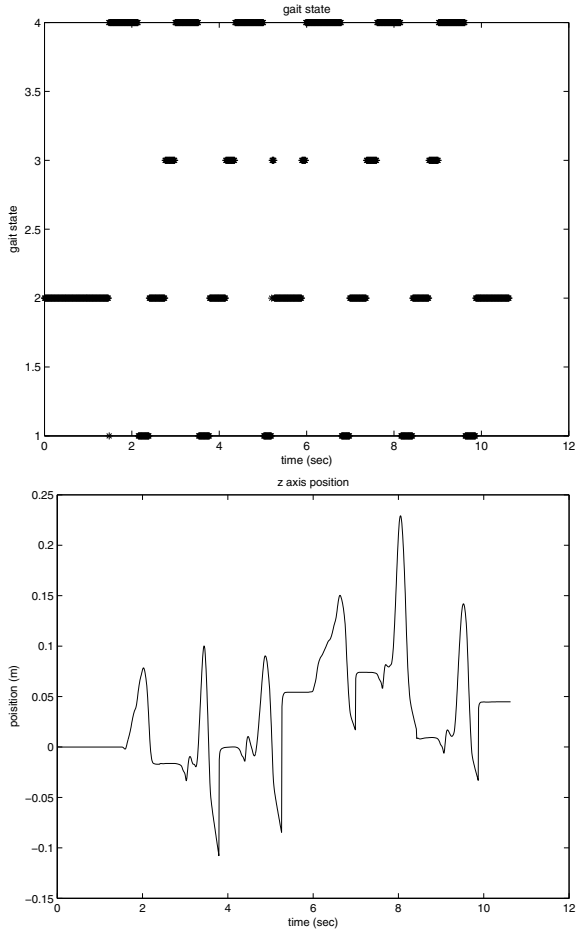

Fig. 7. Gait states and z axis position

VI. Conclusion

In this paper, we have proposed a pedestrian inertial navigation system. A key contribution is that a gait state is modeled as a Markov process and gait states are estimated using the hidden Markov model filter. With this gait state estimation, we can accurately detect zero velocity interval, which plays an important role in reducing the position estimation error.

In the paper, we only used the fact that gait state 2 is zero velocity interval. In the gait state 1, we know that there is relatively small position movement and ankle rotation is dominant movement. If we can use this kind of information effectively, we believe position error could be further reduced. Currently, we are working on this direction.

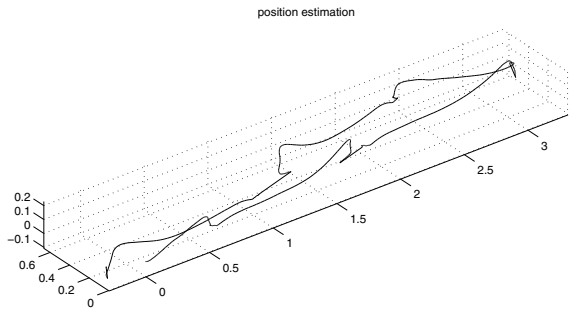

Fig. 8. Position in the navigation coordinate (1st experiment)

Acknowledgment

This work was supported by the Korea Science and Engineering Foundation (KOSEF) grant funded by the Korea government(MOST) (No. R01-2006-000-11334-0) and the Network-based Automation Research Center (NARC) grant funded by Ministry of Knowledge Economy and Ulsan Metropolitan City.

References

[1] H. Liu, H. Darabi, P. Banerjee, and J. Liu, "Survey of wireless indoor positioning techniques and systems," *IEEE Tr. on Systems, Man, and Cybernetics, Part C*, vol. 37, no. 6, pp. 1067–1080, 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. M. IV, and J. Calusidan, "Self-contained position tracking of human movement using small inertial/magnetic sensor modules," in *Proceedings of IEEE International Conference on Robotics and Automation*, pp. 2526–2533, 2007.

[4] L. Ojeda and J. Borenstein, "Non-GPS navigation for security personnel and first responders," *Journal of Navigation*, vol. 60, no. 3, pp. 391–407, 2007.

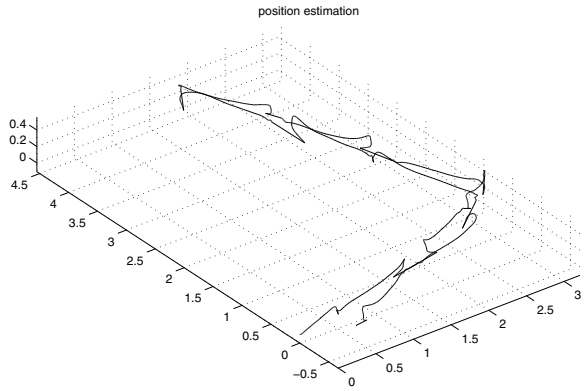

Fig. 9. Position in the navigation coordinate (2nd experiment)

[5] B. D. O. Anderson, "From Wiener to hiden Markov models," *IEEE Control Systems*, vol. 19, no. 3, pp. 41–51, 1999.
[6] J. Perry, *Gait Analysis: Normal and Pathological Function*. SLACK Incoporated, 1992.
[7] J. B. Kuipers, *Quaternions and rotation sequences: a primer with applications to orbits, aerospace, and virtual reality*. New Jersey: Princeton University Press, 1999.
[8] R. G. Brown and P. Y. C. Hwang, *Introduction to Random Signals and Applied Kalman Filtering*. New York: John Wiley & Sons, 1997.